

IBM® DB2® Universal Database



관리 안내서: 구현

버전 7

IBM® DB2® Universal Database



관리 안내서: 구현

버전 7

이 책의 정보와 지원하는 제품을 사용하기 전에 반드시 633 페이지의 『부록M. 주의사항』을 읽으십시오.

이 책에는 IBM의 특허 정보가 나와 있습니다. 이 정보는 사용권 계약하에서 제공되며, 저작권법으로 보호받습니다. 이 책에 있는 정보는 어떠한 제품도 보증하지 않으며, 이 책에 제공된 어떤 내용도 이와 같이 해석되어서는 안됩니다.

책에 대한 주문은 IBM 영업 대표나 IBM 해당 지역 사무소를 통해 하시기 바랍니다.

IBM으로 정보를 보내면, IBM은 적절한 방식으로 이를 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

목차

이 책에 관하여	xi
이 책의 사용자	xii
이 책의 구성 방법	xii
관리 안내서의 다른 볼륨에 대한 간단한 개요	xiv
관리 안내서: 계획	xiv
관리 안내서: 성능	xvi

제1부 제어 센터를 사용하여 관리 1

제1장 GUI 도구를 사용한 DB2 관리	3
관리 도구	4
공통 도구 기능	7
SQL 표시 및 명령 표시	7
관련 내용 표시	8
DDL 생성	9
필터	11
도움말	12
제어 센터	12
제어 센터의 주 요소	13
OS/390용 DB2에서 사용자 정의된 제어	
센터 사용	15
관리될 수 있는 시스템	15
관리될 수 있는 오브젝트	15
제어 센터에서 시스템 표시	17
OS/390용 DB2 오브젝트 관리	18
OS/390용 DB2 서브시스템 추가	18
게이트웨이 연결 관리	19
제어 센터에서 수행할 수 있는 기능	19
새로운 오브젝트 작성	20
기존 오브젝트에 대한 작업	21
오브젝트 위치(OS/390용 DB2 전용)	21
위성 관리 센터	22
명령 센터	24
스크립트 센터	24

스크립트 센터에서 기존 스크립트 사용	25
저장된 명령 스크립트를 수행하도록 스케줄	26
저널	26
작업에 대한 작업	26
사용권 센터	27
경보 센터	28
클라이언트 구성 지원 프로그램	28
성능 모니터	29
이벤트 모니터	30
모니터 도구 사용	31
특정 시점에서 성능 모니터	34
사전 정의된 모니터	36
경보 센터에 오브젝트가 나타날 때 필요한	
조치	38
기간 동안 이벤트 분석	38
이벤트 분석기	39
SQL문 분석	42
조회 성능 향상	43
단순 동적 SQL문 분석	44
원격 데이터베이스 관리	45
사용자 관리	47
권한 및 특권의 권한 부여와 권한 취소	47
데이터 이동	48
저장영역 관리	51
테이블 및 색인 크기 계산	51
테이블 공간에서 사용 가능한 공간 점검	52
테이블 공간으로 공간 추가	52
문제점 해결	53
데이터 복제	54
LDAP(Lightweight Directory Access	
Protocol) 사용	56
Java 제어 센터 사용	56
Java 애플릿으로서 제어 센터 수행	57
관리용으로 Java 기반 도구 사용	57

제2부 설계 구현	59
I 제2장 데이터베이스를 작성하기 전에	61
I 데이터베이스를 작성하기 전에	62
I DB2 시작	62
Windows NT에서 DB2 UDB 시작	63
데이터베이스 관리 프로그램의 다중 인스턴스 사용	64
스키마별 오브젝트 구성 및 그룹화	65
I 병렬 처리 작동	66
데이터 파티션 작동	69
I DB2 중지	72
I 데이터베이스에서 세부사항 작성	73
I 데이터베이스의 논리적 특성 및 물리적 특성 설계	73
I 인스턴스 작성	73
I 사용권 관리	84
I 환경 변수 및 프로파일 레지스트리 설정	84
I DB2 관리 서버(DAS)	94
I 노드 구성 파일 작성	115
I 데이터베이스 구성 파일 작성	117
I 응답 파일을 사용한 구성 정보 복제	118
I FCM 통신 작동	118
제3장 데이터베이스 작성	121
초기 노드 그룹의 정의	123
초기 테이블 공간의 정의	123
시스템 카탈로그 테이블의 정의	125
데이터베이스 디렉토리의 정의	126
지역 데이터베이스 디렉토리	126
시스템 데이터베이스 디렉토리	127
노드 디렉토리	127
DCE 디렉토리 서비스	128
LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스	129
노드 그룹 작성	129
데이터베이스 복구 로그의 정의	130
유틸리티를 데이터베이스에 바인딩	131

데이터베이스 카탈로그화	131
테이블 공간 작성	132
I 시스템 임시 테이블 공간 작성	136
I 사용자 임시 테이블 공간 작성	136
I 노드 그룹에 테이블 공간 작성	137
I 원시 입출력	137
I 스키마 작성	138
I 스키마 설정	139
I 테이블 작성 및 데이터 수록	140
I 대형 오브젝트(LOB) 컬럼 고려사항	143
I 제한조건 정의	145
I 생성된 컬럼을 새 테이블에 정의	151
I 사용자 정의 임시 테이블 작성	152
I 식별 컬럼을 새 테이블에 정의	153
I 입력된 테이블 작성	154
I 입력된 테이블에 데이터 수록	155
I 계층구조 테이블	155
I 다중 테이블 공간에 테이블 작성	155
I 파티션된 데이터베이스에서 테이블 작성	156
I 트리거 작성	158
I 트리거 종속성	160
I 사용자 정의 함수(UDF) 및 메소드 작성	160
함수 맵핑 작성	162
함수 템플릿 작성	163
I 사용자 정의 유형(UDT) 작성	164
I 사용자 정의 구별 유형 작성	165
I 사용자 정의 구조화 유형 작성	166
I 유형 맵핑 작성	166
I 뷰 작성	167
입력된 뷰 작성	170
I 요약 테이블 작성	170
I 별명 작성	173
I 래퍼 작성	175
I 서버 작성	176
서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화	177
I 별명 작성	184
I 별명 및 데이터 소스 오브젝트 참조	185

별명 및 데이터 소스 오브젝트와의 작업	185	래퍼 삭제	240
기존의 별명 및 데이터 소스 식별	185	서버 변경 또는 삭제	241
색인, 색인 확장 또는 색인 스펙 작성	187	별명 변경 또는 삭제	241
색인 사용	191	색인, 색인 확장 또는 색인 스펙 삭제	243
CREATE INDEX문 사용	192	오브젝트 변경시의 명령문 종속성	244
사용자 정의 확장 색인 유형 작성	194		
색인 유지보수의 세부사항	195	제3부 데이터베이스 보안	247
색인 검색의 세부사항	195		
색인 검색 세부사항	196	제5장 데이터베이스 액세스 제어	249
색인 확장을 정의하는 시나리오	198	설치를 위한 사용자 ID 및 그룹 선택	250
제4장 데이터베이스 변경	201	Windows NT 플랫폼 고려사항	251
데이터베이스를 변경하기 전에	201	UNIX 플랫폼 고려사항	252
논리 및 물리 설계 특성 변경	201	일반 규칙	252
사용권 정보 변경	201	서버에 대한 인증 방식 선택	253
인스턴스 변경	201	원격 클라이언트에 대한 인증 고려사항	259
환경 변수 및 프로파일 레지스트리 변수 변경	206	파티션된 데이터베이스 고려사항	260
노드 구성 파일 변경	206	사용자 인증을 위한 DCE 보안 서비스 사용	260
데이터베이스 구성 변경	206	DCE용 DB2 사용자 설정 방법	261
데이터베이스 변경	208	DCE를 사용하기 위한 DB2 서버 설정 방법	263
데이터베이스 삭제	208	DCE를 사용하기 위한 DB2 클라이언트 인스턴스 설정 방법	265
노드 그룹 변경	209	DCE 보안을 사용한 DB2 제한사항	265
테이블 공간 변경	209	연합 데이터베이스 인증 처리	267
스키마 삭제	215	인증 설정값	267
구조 및 내용에서 테이블 수정	216	ID 및 암호를 데이터 소스로 전달	269
사용자 정의 구조화 유형 변경	231	연합 데이터베이스 인증 예	271
입력된 테이블의 행 삭제 및 갱신	231	특권, 권한 및 권한 부여	273
기존 테이블 이름 바꾸기	231	시스템 관리 권한(SYSADM)	276
테이블 삭제	233	시스템 제어 권한(SYSCTRL)	277
사용자 정의 임시 테이블 삭제	235	시스템 유지보수 권한(SYSMAINT)	278
트리거 삭제	235	데이터베이스 관리 권한(DBADM)	279
사용자 정의 함수(UDF), 유형 맵핑 또는		LOAD 권한	280
메소드 삭제	236	데이터베이스 특권	281
사용자 정의 유형(UDT) 또는 유형 맵핑 삭제	236	스키마 특권	283
뷰 변경 또는 삭제	237	테이블 공간 특권	284
요약 테이블 삭제	239	테이블 및 뷰 특권	284
		별명 특권	287

서버 특권	288	제8장 데이터베이스 복구	347
패키지 특권.	289	시스템 손상 복구.	348
색인 특권	290	일관성 있는 데이터베이스 확보	348
데이터베이스 오브젝트에 대한 액세스 제어	290	파티션된 데이터베이스 환경의 트랜잭션	
특권 권한 부여	291	오류 복구	349
특권 취소	292	실패한 데이터베이스 파티션 서버 식별	354
오브젝트 작성 및 삭제에 의한 내재적 권		복구 방법: 버전 복구	355
한 부여 관리	294	데이터베이스 백업	355
플랜 또는 패키지의 소유권 설정	294	데이터베이스 복원	365
패키지를 통한 간접 특권 허용	295	복구 방법: 롤 포워드 복구.	372
별명이 들어 있는 패키지를 통한 간접 특		백업 고려사항	373
권 허용	295	복원 고려사항	376
뷰를 사용하여 데이터에 대한 액세스 제어	296	데이터베이스의 롤 포워드 변경사항	380
감사 기능을 사용하여 데이터에 대한 액세스		복구 실행기록 파일 정보	406
스 모니터링.	300	가비지 콜렉션	409
타스크 및 필수 권한 부여.	301	DB2 Data Links Manager 고려사항	413
시스템 카탈로그 사용	302	시스템 손상 복구 고려사항	413
권한 부여된 특권이 있는 권한 부여 이름		백업 유틸리티 고려사항.	414
검색	303	복원 및 롤 포워드 유틸리티 고려사항	415
DBADM 권한이 있는 모든 이름 검색	303	롤 포워드 없이 오프라인 백업에서 데이터	
테이블에 액세스하기 위해 권한 부여된 이		베이스 복원.	417
름 검색	303	데이터베이스 및 테이블 공간 복원 및 로	
사용자에게 권한 부여된 모든 특권 검색	304	그 끝까지 롤 포워드.	418
시스템 카탈로그 뷰 보안	305	데이터베이스 및 테이블 공간 복원 및 특	
제6장 DB2 활동 감사	307	정 시점까지 롤 포워드	418
감사 기능 활동	309	DB2 Data Links Manager 및 복구 상호	
감사 기능 사용 시나리오	312	작용	419
감사 기능 메시지.	316	Datalink_Reconcile_Not_Possible 상태에	
감사 기능 레코드 레이아웃	317	서 테이블 제거	422
감사 기능 추가 정보 및 기술.	335	데이터 링크 조정	423
DB2 감사 기능 활동 제어.	336	Tivoli 저장영역 관리 프로그램	425
제4부 데이터 이동.	341	UNIX 기반 플랫폼용 Tivoli Storage	
제7장 데이터 이동을 위한 유틸리티.	343	Manager 클라이언트 설정	425
제5부 복구	345	기타 플랫폼용 Tivoli Storage Manager	
		클라이언트 설정	427
		Tivoli Storage Manager 사용에 대한 고	
		려사항	429

호스트에서 2단계 확약중 이상 실패 트랜잭션 복구	436	ATTACH 명령	475
DB2 Connect에 DB2 동기 지점 관리 프 로그래밍 구성된 경우의 복구	436	CONNECT문	476
DB2 Connect가 DB2 동기 지점 관리 프 로그래밍 사용하지 않는 경우의 복구	438	DCE 디렉토리 정보를 임시로 대체	478
<hr/>		디렉토리 서비스 TASK	478
제6부 부록 및 끝머리	441	DCE 관리자 TASK	478
		데이터베이스 관리자 TASK	479
		데이터베이스 사용자 TASK	481
		디렉토리 서비스의 제한사항	482
부록A. 이름 지정 규칙	443	부록C. 데이터베이스 복구를 위한 User Exit	485
데이터베이스 이름	443	OS/2의 개요	485
데이터베이스 및 데이터베이스 별명	444	UNIX 기반 운영 체제의 개요	486
사용자 ID 및 암호	445	User Exit 프로그램 호출	487
스키마 이름	446	User Exit 프로그램 샘플	487
그룹 및 사용자 이름	446	OS/2 샘플 User Exit 프로그램	488
오브젝트 이름	447	UNIX 기반 운영 체제용 샘플 User Exit 프로그램	489
연합 데이터베이스 오브젝트 이름	448	호출 형식	490
연합 시스템에서 대소문자 구별 값이 보존 되는 방법	450	OS/2용 호출 형식	490
부록B. DCE 디렉토리 서비스 사용	453	UNIX 기반 또는 Windows NT 운영 체 제용 형식	492
디렉토리 오브젝트 작성	454	아카이브 및 검색 고려사항	493
데이터베이스 오브젝트	454	백업 및 복원에 대한 고려사항(OS/2용 DB2 전용)	496
데이터베이스 위치 지정자 오브젝트	456	오류 조절	497
경로지정 정보 오브젝트	457	부록D. 다중 데이터베이스 파티션 서버로 명 령 발행	501
각 오브젝트 클래스의 속성	459	명령	502
각 속성에 관한 세부사항	460	명령 설명	502
디렉토리 서비스 보안	465	수행될 명령 지정	503
구성 매개변수 및 레지스트리 변수	467	UNIX 기반 플랫폼에서 병렬에 있는 명령 수행	504
CATALOG 및 ATTACH 명령 및 CONNECT문	469	UNIX 기반 플랫폼에서 rah 프로세스 모 니터링	505
CATALOG GLOBAL DATABASE 명 령	469	추가 rah(모든 호스트 수행) 정보(Solaris 및 AIX 전용)	506
CONNECT문	469	접두부 순서	508
ATTACH 명령	470	머신 목록 지정	511
클라이언트를 데이터베이스에 연결하는 방법	470		
같은 셀의 데이터베이스에 연결	472		
서로 다른 셀의 데이터베이스에 연결	474		
디렉토리 검색 방법	475		

머신의 목록에서 중복 항목 제거	511		IBM Netfinity SP Switch 사용에 대한
rah 명령 제어	512		전제조건
UNIX 기반 플랫폼의 \$RAHDOTFILES	514		VI를 사용하여 고속 상호연결
Windows NT에 기본 환경 프로파일 설정	514		가상 인터페이스(VI) 하드웨어 설정
UNIX 기반 플랫폼에서 rah 문제점 판별	515		VI를 사용하여 DB2 수행

부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법 519

서버 인증에 대한 샘플 시나리오 520

클라이언트 인증 및 Windows NT 클라이언트 머신에 대한 샘플 시나리오 521

클라이언트 인증 및 Windows 95 클라이언트 머신에 대한 샘플 시나리오 522

DB2로 백업 도메인 제어기 사용 523

| Windows NT용 DB2 사용자 인증 524

| 사용자 이름 및 그룹 이름 제한사항 524

| Windows NT용 DB2 보안 서비스 524

| 백업 도메인 제어기에서 DB2 설치 524

| 그룹 및 도메인 보안 인증 526

부록F. Windows NT 성능 모니터 사용 531

Windows NT 성능 모니터로 DB2 등록 531

DB2 성능 정보에 원격 액세스 작동 532

DB2 및 DB2 Connect 성능 값 표시 533

원격 DB2 성능 정보 액세스 534

DB2 성능 값 재설정 535

부록G. Windows 2000 데이터베이스 파티션 서버 및 Windows NT 작업 537

인스턴스에서의 데이터베이스 파티션 서버 나열 537

인스턴스에 데이터베이스 파티션 서버 추가 538

| 데이터베이스 파티션 서버 구성 변경 540

인스턴스에서 데이터베이스 파티션 서버 삭제 541

부록H. 다중 논리 노드 구성 543

| **부록I. 고속 노드간 통신 545**

| TCP/IP를 사용하여 고속 상호연결 546

부록J. LDAP(Lightweight Directory Access Protocol) 등록부 서비스 559

LDAP 클라이언트 및 서버 구성 지원 560

| Windows 2000에서 사용 중인 디렉토리 지원 561

| 사용중인 디렉토리로 DB2 구성 561

| IBM LDAP 환경에서 DB2 구성 562

| LDAP 사용자 작성 563

| DB2 응용프로그램용 LDAP 사용자 구성 564

설치 후 DB2 서버의 등록 564

DB2 서버에 대한 프로토콜 정보 갱신 566

ATTACH에 대한 노드 별명 카탈로그화 567

DB2 서버 등록해제 567

데이터베이스의 등록 567

원격 서버에 접속 568

데이터베이스 등록 해제 569

지역 데이터베이스 및 노드 디렉토리에 있는 LDAP 항목 새로 고침 569

검색 571

호스트 데이터베이스 구성 571

사용자 레벨에서 DB2 레지스트리 변수 설정 573

설치 완료 후 LDAP 지원 작동 기능 574

LDAP 지원 작동 불가능 575

| LDAP 지원 및 DB2 연결 575

| 보안 고려사항 576

| Windows 2000에서 사용중인 디렉토리 구성 고려사항 577

| DB2 오브젝트 클래스 및 속성이 있는 디렉토리 스키마 확장 578

| IBM eNetwork Directory 버전 2.1에 대해 디렉토리 스키마 확장 578

	Windows 2000 Active Directory 버전				
	2.1에 대해 디렉토리 스키마 확장 . . .	579			
	Windows 2000 Active Directory에 있는				
	DB2 오브젝트.	581			
	DB2가 사용하는 오브젝트 클래스 및 속				
	성	581			
	부록K. 제어 센터 확장	595			
	성능 고려사항	595			
	팩 고려사항.	595			
	인터페이스 설명	595			
	CCExtension	596			
	CCObject	597			
	CCMenuItem	600			
	CCToolBarAction	601			
	사용 시나리오	601			
	MyExtension.java	602			
	MySample.java	603			
	MyDatabaseActions.java	604			
	MyInstance.java	604			
	MyDB2.java	605			
	MyDatabases.java	606			
	MySYSPLAN.java	607			
	MyTable.java	607			
	MyDBUser.java	608			
	MyToolBarAction.java	609			
	MyAlterAction.java	610			
	MyAction.java	610			
	MyDropAction.java	610			
	MyCascadeAction.java	611			
	MyCreateAction.java	611			
	부록L. DB2 라이브러리 사용	613			
	DB2 PDF 파일 및 인쇄된 책	613			
	DB2 정보	613			
	PDF 책 인쇄	623			
	인쇄된 책 주문	623			
	DB2 온라인 문서.	625			
	온라인 도움말 액세스	625			
	정보 온라인 보기.	627			
	DB2 마법사 사용.	629			
	문서 서버 설정	630			
	정보 온라인 검색.	631			
	부록M. 주의사항	633			
	등록 상표	636			
	색인	639			
	IBM에 문의	655			
	제품 정보	655			

이 책에 관하여

새 개의 볼륨의 관리 안내서에서는 DB2* 관계형 데이터베이스 관리 시스템 (RDBMS) 제품을 사용하고 2000년 문제를 관리하는 데 필요한 정보를 제공합니다. 내용은 다음과 같습니다.

- 데이터베이스 설계에 대한 정보(*관리 안내서: 계획에 있음*)
- 데이터베이스의 구현 및 관리에 대한 정보(*관리 안내서: 구현에 있음*)
- 성능 개선을 위한 데이터베이스 환경의 구성 및 조정에 대한 정보(*관리 안내서 : 성능에 있음*)

이 책에서 다루는 여러 타스크는 서로 다른 인터페이스를 사용하여 수행될 수 있습니다.

- 그래픽 인터페이스로부터 데이터베이스를 액세스하고 조작할 수 있도록 해주는 명령 프롬프트. 이 인터페이스에서, SQL문과 DB2 유틸리티 기능을 실행할 수도 있습니다. 이 책의 예 중 대부분이 이 인터페이스를 사용하여 설명되어 있습니다. 명령행 처리기 사용에 관한 자세한 정보는 *Command Reference*에서 자세한 내용을 참조하십시오.
- 응용프로그램 내에서 DB2 유틸리티 기능을 실행할 수 있도록 해주는 **Application Programming Interface(API)**. *Administrative API Reference*에서 자세한 내용을 참조하십시오.
- 시스템 구성, 디렉토리 관리, 시스템 백업 및 복구, 작업 스케줄 작성, 미디어 관리와 같은 관리 타스크를 그래픽으로 수행할 수 있도록 해주는 제어 센터. 제어 센터에는 시스템간에 그래픽으로 데이터 복제를 설정할 수 있는 복제 관리가 있습니다. 그리고 제어 센터는 그래픽 사용자 인터페이스를 통해 DB2 유틸리티 기능을 실행할 수 있게 합니다. 사용자 플랫폼에 따라 제어 센터를 호출하는 여러가지 방법이 있습니다. 예를 들어, 명령행에서 db2cc 명령을 사용하거나, (OS/2에서) DB2 폴더에서 제어 센터 아이콘을 선택하거나 Windows 플랫폼에서 시작 패널을 시작하십시오. 좀더 자세한 내용을 보려면, 제어 센터 창의 도움말 풀다운 메뉴에서 시작하기를 선택하십시오. 제어 센터에서 **Visual Explain** 및 성능 모니터 도구가 호출됩니다.

관리 작업을 수행하는 데 사용할 수 있는 기타 도구가 있습니다. 도구는 다음과 같습니다.

- 스크립트라고 불리는 작은 응용프로그램을 저장하는 스크립트 센터. 이 스크립트는 운영 체제 명령뿐 아니라 SQL문과 DB2 명령도 수록하고 있습니다.
- 다른 DB2 조작에서 기인한 메시지를 모니터하는 경고 센터
- 제어 센터, 경고 센터, 복제 관리의 설정값을 변경하는 도구 설정값
- 작업이 자동으로 수행되도록 스케줄을 잡는 저널
- 웨어하우스 오브젝트를 관리하는 Data Warehouse Center.

이 책의 사용자

이 책은 기본적으로 지역 또는 원격 클라이언트에 의해 액세스될 데이터베이스를 설계, 구현 및 관리해야 하는 데이터베이스 관리자, 시스템 관리자, 보안 관리자 및 시스템 조작용을 대상으로 합니다. 또한, 프로그래머나 DB2 관계형 데이터베이스의 관리 및 조작을 이해해야 하는 기타 다른 사용자에게도 도움이 됩니다.

이 책의 구성 방법

이 책에는 다음 주요 주제에 대한 정보가 들어 있습니다.

제어 센터를 사용하여 관리

- 제1장 GUI 도구를 사용한 DB2 관리에서는 데이터베이스 관리에 사용되는 그래픽 사용자 인터페이스(GUI) 도구를 소개합니다.

설계 구현

- 제2장 데이터베이스를 작성하기 전에는 데이터베이스와 데이터베이스 내의 오브젝트를 작성하기 전에 전제조건에 대해 설명합니다.
- 제3장 데이터베이스 작성에서는 데이터베이스와 데이터베이스 내의 오브젝트를 작성하는 것과 연관된 TASK에 대해 설명합니다.
- 제4장 데이터베이스 변경에서는 데이터베이스와 데이터베이스 내의 오브젝트를 변경하고 삭제하는 것과 연관된 전제조건과 TASK에 대해 설명합니다.

데이터베이스 보안

- 제5장 데이터베이스 액세스 제어에서는 데이터베이스 자원에 대한 액세스를 제어하는 방법에 대해 설명합니다.
- 제6장 DB2 활동 감사에서는 원하지 않거나 예상하지 않은 데이터 액세스를 검출하고 모니터링하는 방법에 대해 설명합니다.

데이터 이동

- 제7장 데이터 이동을 위한 유틸리티에서는 로드, AutoLoader, 가져오기 및 내보내기 유틸리티에 대해 설명합니다.

복구

- 제8장 데이터베이스 복구에서는 데이터베이스 또는 테이블 공간의 백업 및 복구, 롤 포워드 복구 방법 사용을 포함하여 데이터베이스 및 테이블 공간 복구 방법을 선택할 때 고려할 사항을 설명합니다.

부록

- 부록A. 이름 지정 규칙에서는 데이터베이스 및 오브젝트를 이름 지정할 때 지켜야 할 규칙을 제공합니다.
- 부록B. DCE 디렉토리 서비스 사용에서는 DCE 디렉토리 서비스 사용 방법에 대한 정보가 제공됩니다.
- 부록C. 데이터베이스 복구를 위한 User Exit에서는 데이터베이스 로그 파일로 User Exit 프로그램을 사용하는 방법 및 몇몇 User Exit 샘플 프로그램이 제공됩니다.
- 부록D. 다중 데이터베이스 파티션 서버로 명령 발행에서는 파티션된 데이터베이스 환경의 모든 파티션으로 명령을 송신하는 *db2_all*과 *rah* 쉘 스크립트를 사용하는 방법을 설명합니다.
- 부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법에서는 DB2가 Windows NT 보안 기능을 사용하여 작업하는 방법을 설명합니다.
- 부록F. Windows NT 성능 모니터 사용에서는 Windows NT용 DB2 사용자에게 사용 가능한 두 개의 성능 모니터, DB2 성능 모니터 및 Windows NT 성능 모니터를 설명합니다.

- 부록G. Windows 2000 데이터베이스 파티션 서버 및 Windows NT 작업에서는 파티션된 데이터베이스 서버에서 작업하는 Windows NT 및 Windows 2000에 의해 사용되는 유틸리티를 설명합니다.
- 부록H. 다중 논리 노드 구성에서는 파티션된 데이터베이스 환경에서 다중 논리 노드를 구성하는 방법을 설명합니다.
- 부록I. 고속 노드간 통신에서는 Windows NT 환경에서 DB2 Enterprise - Extended Edition으로 가상 인터페이스 아키텍처를 사용하는 방법에 대해 설명합니다.
- 부록J. LDAP(Lightweight Directory Access Protocol) 등록부 서비스에서는 LDAP 등록부 서비스 사용 방법에 대한 정보가 제공됩니다.
- 부록K. 제어 센터 확장에서는 새 조치를 포함한 새 도구 막대 버튼, 새 오브젝트 정의 및 새 조치 정의를 추가하여 제어 센터를 확장하는 방법에 대한 정보를 제공합니다.
- 부록L. DB2 라이브러리 사용에서는 wizard, 온라인 도움말, 메시지 및 책을 포함하여 DB2 라이브러리 구조에 대한 정보를 제공합니다.

관리 안내서의 다른 볼륨에 대한 간단한 개요

관리 안내서: 계획

*관리 안내서: 계획*은 데이터베이스 설계와 관련됩니다. 이 책은 논리적 및 물리적 설계 문제와 높은 사용 가능성 주제에 대해 설명합니다. 관리 안내서, 성능에 있는 특정 장 및 부록에 대해 다음에서 간략히 설명합니다.

DB2 Universal Database의 세계

- "DB2 Universal Database 관리"에서는 DB2 Universal Database의 소개 및 개요를 나타냅니다.

데이터베이스 개념

- "기본 관계형 데이터베이스 개념"에서는 복구 오브젝트, 저장 오브젝트 및 시스템 오브젝트를 포함하여 데이터베이스 오브젝트의 개요를 나타냅니다.

- "연합 시스템"에서는 단일 명령문에서 두 개 이상의 DBMS나 데이터베이스를 참조하는 SQL문을 제출하는 사용자 및 응용프로그램을 지원하는 DBMS(database management systems)인 연합 시스템을 설명합니다.
- "병렬 데이터베이스 시스템"에서는 DB2에서 사용 가능한 병렬 처리 유형에 대한 소개를 제공합니다.
- "데이터 웨어하우스 정보"에서는 데이터 웨어하우스 및 데이터 웨어하우스 타스크의 개요를 제공합니다.
- "Spatial Extender 정보"에서는 Spatial Extender의 목적을 설명하고 처리하는 데이터를 설명하여 Spatial Extender를 소개합니다.

Database Design

- "논리적 데이터베이스 설계"에서는 논리 데이터베이스 설계에 대한 개념 및 지시사항을 설명합니다.
- "물리 데이터베이스 설계"에서는 데이터 저장영역과 관련된 고려사항을 포함하여 물리 데이터베이스 설계에 대한 지시사항을 설명합니다.

분산 트랜잭션 프로세싱(DTP)

- "분산 데이터베이스 설계"에서는 단일 트랜잭션에서 다중 데이터베이스에 액세스하는 방법을 설명합니다.
- "트랜잭션 관리 프로그램 설계"에서는 CICS와 같은 분산트랜잭션 프로세싱 환경에서, 데이터베이스를 사용하는 방법에 대해 설명합니다.

시스템의 고가용성 보장

- "높은 사용 가능성 설계"에서는 DB2가 제공하는 높은 사용 가능성 실패에 대한 개요를 나타냅니다.
- "AIX의 높은 사용 가능성 클러스터 멀티 프로세싱, 향상된 확장성(HACMP ES)"에서는 AIX에서 높은 사용 가능성 실패 복구의 DB2 지원을 설명합니다.
- "Windows NT 환경에서의 높은 사용 가능성"에서는 Windows NT에서의 높은 사용 가용성 실패 복구에 대한 DB2 지원을 설명합니다.
- "Sun Cluster 2.2에서는 DB2 및 높은 사용 가능성"에서는 Sun Solaris 운영 체제에서의 높은 사용 가능성 실패에 대한 DB2 지원을 설명합니다.

부록

- "데이터베이스 이주 계획"은 버전 7로 데이터베이스를 이주하는 정보를 제공합니다.
- "릴리스간의 비호환성"에서는 버전 7을 포함하여 릴리스간에 도입된 비호환성에 대해 설명합니다.
- "자국어 지원(NLS)"은 DB2 자국어 지원(국가, 언어 및 코드 페이지 정보)을 소개합니다.

관리 안내서: 성능

관리 안내서: 성능은 성능 문제와 관련됩니다. 즉, 이 주제 및 문제는 사용자 응용프로그램과 DB2 Universal Database 제품 자체의 성능을 설정, 테스트 및 개선하는 것과 연관됩니다. 관리 안내서, 성능에 있는 특정 장 및 부록에 대해 다음에서 간략히 설명합니다.

성능 개요

- "성능 요소"에서는 DB2 UDB 성능의 관리 및 개선에 대한개념 및 고려사항을 소개합니다.
- "아키텍처 및 처리 개요"에서는 주요 DB2 Universal Database 아키텍처 및 프로세스에 대해 소개합니다.

응용프로그램 성능 조정

- "응용프로그램 고려사항"에서는 응용프로그램을 설계할 때 데이터베이스 성능을 향상시키기 위한 몇몇 기술에 대해 설명합니다.
- "환경상의 고려사항"에서는 데이터베이스 환경을 설정할 때 데이터베이스 성능을 향상시키기 위한 몇몇 기술에 대해 설명합니다.
- "시스템 카탈로그 통계"에서는 사용자 데이터에 대한 통계를 수집하고 이를 사용하여 최적의 성능을 보장할 수 있는 방법을 설명합니다.
- "SQL 컴파일러 이해"에서는 SQL 컴파일러로 컴파일할 때 SQL문에 발생하는 현상에 대해 설명합니다.
- "SQL Explain 기능"에서는 SQL 컴파일러가 데이터에 액세스하기 위해 취하는 선택항목을 살펴볼 수 있는 Explain 기능에 대해 설명합니다.

시스템의 조정 및 구성

- "조작적 성능"에서는 데이터베이스 관리 프로그램이 메모리를 사용하는 방법 및 런타임 성능에 영향을 주는 기타 고려사항에 대한 개요를 제공합니다.
- "조정자(governor) 사용"에서는 조정자를 사용하여 데이터베이스 관리의 일부 양상을 제어하는 방법에 대해 설명합니다.
- "구성 조정"에서는 데이터베이스 시스템의 크기 증가에 관련된 일부 고려사항 및 TASK에 대해 소개합니다.
- "데이터베이스 파티션에 데이터 재분배에서는 파티션"된 데이터베이스 환경에서 파티션에 데이터를 재분배하는 데 필요한 TASK에 대해 설명합니다.
- "벤치마크 테스트"에서는 벤치마크 테스트의 개요 및 벤치마크 테스트 수행 방법을 제공합니다.
- "DB2 구성"에서는 데이터베이스 관리 프로그램 및 데이터베이스 구성 파일과 구성 매개변수에 대한 값에 대해 설명합니다.

부록

- "DB2 레지스트리 및 환경 변수"에서는 프로파일 등록 값 및 환경 변수를 설명합니다.
- "Explain 테이블 및 정의"에서는 DB2 Explain 기능에서 사용되는 테이블 및 해당 테이블 작성 방법에 대한 정보를 제공합니다.
- "SQL Explain 도구"에서는 DB2 explain 도구, 즉 db2expln 및 dynexpln의 사용에 대한 정보를 제공합니다.
- "db2exfmt — Explain 테이블 형식 도구"는 DB2 Explain 도구를 사용하여 Explain 테이블 데이터를 형식화하는 방법에 대한 정보를 제공합니다.

제1부 제어 센터를 사용하여 관리

제1장 GUI 도구를 사용한 DB2 관리

DB2 Universal Database에서는 그래픽 사용자 데이터베이스(GUI) 도구를 제공하여 사용자가 『제어 센터』라는 중앙 위치에서 지역 및 원격 데이터베이스를 쉽게 관리할 수 있도록 합니다.

이 장에서는 사용할 수 있는 DB2 Universal Database 관리 도구에 대한 개요를 제공하고 이를 사용하여 쉽고 효율적으로 작업을 처리할 수 있는 방법에 대해 설명합니다. Java 제어 센터에 대한 요약과 제어 센터를 사용자 정의하여 사용자 Java 기능 도구를 포함시키는 방법에 대해서도 설명합니다.

이 장에는 다음과 같은 내용이 들어 있습니다.

- 4 페이지의 『관리 도구』
- 7 페이지의 『공통 도구 기능』
- 12 페이지의 『제어 센터』
- 22 페이지의 『위성 관리 센터』
- 24 페이지의 『명령 센터』
- 24 페이지의 『스크립트 센터』
- 26 페이지의 『저널』
- 27 페이지의 『사용권 센터』
- 28 페이지의 『경보 센터』
- 28 페이지의 『클라이언트 구성 지원 프로그램』
- 29 페이지의 『성능 모니터』
- 45 페이지의 『원격 데이터베이스 관리』
- 47 페이지의 『사용자 관리』
- 48 페이지의 『데이터 이동』
- 51 페이지의 『저장영역 관리』
- 53 페이지의 『문제점 해결』

- 54 페이지의 『데이터 복제』
- 56 페이지의 『LDAP(Lightweight Directory Access Protocol) 사용』
- 56 페이지의 『Java 제어 센터 사용』
- 57 페이지의 『관리용으로 Java 기반 도구 사용』

관리 도구

DB2 관리 도구는 DB2 Universal Database 제품 각각에 대해 선택 가능한 구성 요소인 관리 클라이언트의 일부입니다. 관리 클라이언트는 DB2를 사용할 수 있는 모든 운영 체제에 대한 관리 클라이언트가 들어 있는 CD-ROM 세트에서도 사용할 수 있습니다. 이것으로 모든 워크스테이션에 관리 클라이언트를 설치하고 사용할 수 있습니다. 데이터베이스 서버 가지역인지 원격인지 또는 데이터베이스 서버가 수행되고 있는 운영 체제가 무엇인지에 관계없습니다. 도구를 사용하면, 명령행 처리기에서 사용할 수 있는 것과 같은 기능을 그래픽 사용자 인터페이스에서도 수행할 수 있습니다. 이러한 기능에는 DB2 명령, SQL문 또는 시스템 명령 입력이 포함됩니다. 그러나, 도구를 사용하면 복잡한 명령문 또는 명령을 기억할 필요가 없고 추가 지원을 받게 됩니다.

주: 관리 클라이언트는 설치 옵션입니다.

다음 도구는 제어 센터 도구 모음에서 사용할 수 있습니다.

- 제어 센터. 제어 센터는 데이터베이스를 관리하기 위한 주 DB2 그래픽 도구입니다. 제어 센터에서, 지역적으로 등록된 모든 시스템 및 데이터베이스 오브젝트에 대한 명확한 개요를 얻을 수 있습니다.
- 위성 관리 센터. 위성 관리 센터에서 DB2 위성 서버를 관리할 수 있습니다.
- 명령 센터. 명령 센터에서는 DB2 데이터베이스 명령, SQL문 및 운영 체제 명령, 이전 명령 재호출, SQL 조회를 위한 액세스 플랜 화면이동 등을 발행할 수 있습니다.
- 스크립트 센터. 스크립트 센터에서는 운영 체제 레벨의 명령, DB2 명령 스크립트 및 SQL문 스크립트를 작성, 수행 및 스케줄할 수 있습니다.
- 정보 센터. 정보 센터에서는 설정한 임계값을 초과할 때 또는 다중 노드 환경에서 노드가 더 이상 응답하지 않을 때를 알려줍니다.

- 저널. 저널을 사용하면 작업 상태를 보고, 복구 실행 기록 로그와 메시지 로그를 볼 수 있습니다.
- 정보 센터. 정보 센터에서는 DB2 제품 매뉴얼 및 샘플 프로그램에 있는 정보에 빠르게 액세스할 수 있고, 웹에 있는 다른 DB2 정보 소스에도 액세스할 수 있습니다.
- 사용권 센터. 사용권 센터에서는 적합한 사용권 모니터링을 하도록 시스템을 구성할 뿐만 아니라 사용권의 상태도 표시합니다.

GUI 도구로 수행할 수 있는 몇 가지 기능에서는 Wizard 사용 옵션이 제공됩니다. Wizards는 제어 센터에 있는 팝업 메뉴에서 호출됩니다. 이들은 수행 중인 타스크에 필요한 정보를 채우는 방법에 대해 단계별로 프롬프트를 표시하고, 사용자가 제공한 정보를 근거로 계산 및 권장사항을 결정함으로써 보다 높은 레벨의 도움말을 제공합니다. Wizards는 새로운 데이터베이스 관리자 또는 데이터베이스를 가꿈 관리하는 사용자인 경우에 매우 유용합니다.

DB2 Universal Database에는 다음 Wizards가 있습니다.

- 데이터베이스 백업. 이것은 데이터베이스에 있는 데이터, 데이터베이스 가용성 및 복구 가능성 요구사항에 대한 기본적인 질문을 합니다. 그런 다음, 백업 플랜을 제안하고, 작업 스크립트를 작성한 다음 스케줄합니다. 백업 데이터베이스 Wizard를 호출하려면, 백업하려는 데이터베이스를 나타내는 아이콘을 선택하고, 오른쪽 마우스 버튼을 클릭한 후, 백업 → Wizard를 사용한 데이터베이스를 선택하십시오.
- 데이터베이스 작성. 이 Wizard를 사용하면 데이터베이스 작성, 저장영역 지정 및 기본 성능 옵션 선택을 수행할 수 있습니다. 데이터베이스 마법사를 호출하려면, 오브젝트 트리 분할창에서 데이터베이스 아이콘을 선택하고, 오른쪽 마우스 버튼을 클릭한 후, 작성 → 마법사를 사용한 데이터베이스를 선택하십시오.
- 테이블 작성. 이 Wizard는 사전 정의된 컬럼 템플릿을 사용하여 컬럼 설계, 테이블에 대한 기본 키 작성 및 하나 이상의 테이블 공간으로 테이블 지정을 수행할 수 있습니다. 마법사를 호출하려면, 오브젝트 오른쪽 마우스 버튼을 클릭하고, 작성 → 마법사를 사용한 테이블을 선택하십시오.

- 테이블 공간 작성. 이 Wizard로 새로운 테이블 공간을 작성하고 기본 저장영역 성능 옵션을 설정할 수 있습니다. 마법사를 호출하려면, 오른쪽 마우스 버튼을 클릭하고, 작성 → 마법사를 사용한 테이블을 선택하십시오.
- 색인 Wizard. Wizard 색인을 사용하여 제공된 SQL문에 대해 작성하거나 삭제할 색인을 결정하십시오. 권장사항은 지정한 작업 로드 에 따라 결정됩니다. 색인 마법사를 호출하려면, 색인 폴더를 선택하고, 오른쪽 마우스 버튼을 클릭하고, 작성 → 마법사를 사용한 색인을 선택하십시오.
- 성능 구성. 이 Wizard는 데이터베이스, 데이터 및 시스템 목적에 관한 정보를 요청하여 데이터베이스를 조정하는 데 도움이 됩니다. 그런 다음, 데이터베이스 및 인스턴스에 대한 새로운 구성매개변수를 작성하고, 원하면 이를 자동으로 적용하도록 합니다. 이 마법사를 호출하려면, 데이터베이스의 아이콘을 선택하고, 오른쪽 마우스 버튼을 누르고 마법사를 사용하여 구성을 선택하십시오.
- 데이터베이스 복원. 이 Wizard는 데이터베이스 복구 프로세스를 진행합니다. 마법사를 호출하려면, 데이터베이스의 아이콘을 선택하고, 오른쪽 마우스 버튼을 클릭하고 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.
- 다중 사이트 갱신 구성 마법사. 이 Wizard에서는 모든 사이트에 있는 데이터가 일치해야 하는 것이 중요한 경우, 응용프로그램이 여러 사이트를 동시에 갱신할 수 있도록 데이터베이스를 구성할 수 있습니다. 이 마법사를 호출하려면, 인스턴스의 아이콘을 선택하고, 오른쪽 마우스 버튼을 클릭하고 다중 사이트 갱신 → 마법사를 사용한 구성을 선택하십시오.

주: OS/390용 DB2 서브시스템에 대한 Wizards는 없습니다.

그래픽 도구는 제어 센터 도구 모음에서 호출할 수 있는 반면, 일부 추가 GUI 도구는 제어 센터 도구 모음에서 직접 호출할 수 없습니다.

- 성능 모니터. 성능 모니터는 인스턴스, 데이터베이스, 테이블, 테이블 공간 및 연결과 같은 DB2 오브젝트를 모니터링하는 도구입니다. 이 도구를 사용하여 성능 문제점을 발견하고 최적의 성능을 위해 데이터베이스를 조정합니다. 성능 모니터는 제어 센터에 있는 팝업 메뉴의 선택사항으로서 호출됩니다.
- 이벤트 모니터. 이벤트 모니터는 특정 이벤트가 발생한 시간에 데이터베이스의 상태를 기록하여 자원 이용도를 분석할 수 있도록 하는 도구입니다. 이벤트 모니터는 DB2 명령행에서 db2emcrt를 입력하여 작성됩니다.

- 이벤트 분석기. 이벤트 분석기는 이벤트 모니터가 수집한 데이터를 분석할 수 있도록 하는 도구입니다. 이벤트 분석기는 DB2 명령행에서 db2evmon을 입력하여 호출됩니다.
- Visual Explain 함수. Visual Explain 함수를 사용하면 보다 나은 성능을 위해 SQL 조화를 조정할 수 있도록 SQL문에 대한 액세스 플랜을 그래프로서 볼 수 있습니다. 버전 6 이전에는, 액세스 플랜을 보는 데 Visual Explain 도구를 사용했습니다. 이제, Visual Explain은 더 이상 별개의 도구가 아닙니다. 함수는 제어 센터에 있는 다양한 데이터베이스 오브젝트의 팝업 메뉴에서 사용할 수 있으며, 명령 센터에서도 사용할 수 있습니다.

이러한 도구 이외에, 제어 센터의 일부가 아닌 데이터베이스 관리에 유용한 또다른 도구는 클라이언트 구성 지원 프로그램입니다. 클라이언트 구성 지원 프로그램은 사용자가 원격 서버와 통신하도록 돕는 마법사가 들어 있는 도구입니다.

이러한 모든 도구는 나중에 자세히 설명합니다. 다음 절에서는 도구에서 볼 수 있는 기능에 대한 개요를 제공합니다.

공통 도구 기능

다음 기능은 여러 도구에서 사용할 수 있습니다.

- SQL 표시 및 명령 표시
- 관련 내용 표시
- 도움말
- DDL 생성
- 필터

SQL 표시 및 명령 표시

도구가 SQL문을 생성하면, 도구 인터페이스에서 **SQL 표시** 누름 버튼을 사용할 수 있습니다. 마찬가지로, DB2 명령을 생성하는 도구에서도 **명령 표시** 누름 버튼을 사용할 수 있습니다 이들 누름 버튼 중 하나를 클릭하면 다음을 수행할 수 있습니다.

- 그래픽 인터페이스에서 작성한 선택사항을 근거로 도구가 생성하는 SQL문 또는 DB2 명령을 볼 수 있습니다. 이 정보는 인터페이스가 작동하는 방법을 이해하는 데 도움이 됩니다.
- 나중에 다시 사용할 수 있도록 명령문이나 명령을 스크립트로서 저장합니다. 이 기능으로 같은문 또는 명령을 다시 수행하려는 경우, SQL문 또는 DB2 명령을 다시 입력하지 않아도 됩니다. 일단 SQL문 또는 DB2 명령이 스크립트에 저장되면, 스크립트를 스케줄하거나, 스크립트를 변경하거나, 명령문 또는 명령을 다시 입력하지 않고 비슷한 스크립트를 작성할 수 있습니다.

SQL문 또는 DB2 명령을 표시하려면, 다음을 수행하십시오.

1. 제어 센터에서 오브젝트에 대해 작업하는 창 또는 노트북으로 가십시오.
2. **SQL 표시** 또는 **명령 표시** 누름 버튼을 클릭하십시오. 해당 창이 열립니다.

SQL문 및 DB2 명령을 저장하면 SQL문 또는 DB2 명령이 복잡한 경우 매우 유용합니다.

명령 표시 또는 **SQL 표시** 기능을 사용하는 경우, 나중에 편집할 수 있는 새로운 스크립트를 작성하거나, 그렇지 않으면 대화 상자를 닫고 원래 대화 상자로 리턴하여 변경할 수 있습니다. 스크립트 작성 누름 버튼을 클릭하면, 새 명령 스크립트 창이 나타납니다. 스크립트를 저장하기 전에 SQL문 또는 DB2 명령을 편집할 수 있습니다.

관련 내용 표시

관련 내용 표시에서는 테이블, 색인, 뷰, 별명, 트리거, 테이블 공간, 사용자 정의 함수 및 사용자 정의 유형의 관계를 보여줍니다. 예를 들어, 테이블을 선택하고 관련 뷰를 표시하도록 선택하면, 테이블을 직접 근거로 하는 뷰만을 보게 됩니다. 관련 뷰를 근거로 하는 뷰는 볼 수 없습니다. 이러한 관련 뷰는 테이블에서 직접 작성되지 않기 때문입니다.

관련 오브젝트 표시는 다음을 수행하는 데 도움이 됩니다.

- 데이터베이스 구조 이해
- 테이블에 대해 이미 존재하는 색인 결정
- 테이블 공간에 저장된 오브젝트 결정

- 한 오브젝트와 연관되어 수행하려는 조치에 의해 영향을 받는 다른 오브젝트 알기. 예를 들어, 종속 뷰가 있는 테이블을 삭제하려는 경우, 관련 내용 표시에서는 동작하지 않게 될 뷰를 보여줍니다.

관련 내용 표시 기능을 사용하려면, 다음을 수행하십시오.

- 제어 센터의 내용 패널에서 오브젝트를 선택한 후 오른쪽 마우스 버튼을 클릭하십시오.
- 관련 내용 표시를 선택하십시오.
- 탭을 눌러 원하는 관련 오브젝트에 대한 페이지를 여십시오. 선택한 탭에 따라 다른 관련 오브젝트가 나열됩니다. 선택한 오브젝트와 직접 관련되어 있는 오브젝트만이 표시됩니다.

선택된 페이지의 관련 오브젝트에서 오른쪽 마우스 버튼을 클릭하고, 팝업 메뉴에서 관련 내용 표시를 선택할 수 있습니다. 선택된 페이지는 최근 선택사항과 관련된 오브젝트를 표시하도록 변경됩니다. 또한, 선택된 오브젝트 옆에 있는 아래쪽 화살표를 눌러 관계를 표시하기 위해 사용자가 이전에 선택한 오브젝트 목록을 표시할 수 있습니다.

- 닫기를 클릭하여 관련 내용 표시 노트북을 닫은 후 제어 센터로 돌아가십시오.

DDL 생성

DDL 생성 함수를 사용하면 DDL 및 SQL문, 그리고 다음에 대한 통계를 다시 작성하여 스크립트 파일에 저장할 수 있습니다.

- 데이터베이스 오브젝트
- 권한 부여문
- 테이블 공간, 노드 그룹 및 버퍼풀
- 데이터베이스 통계

이것으로 다음을 수행할 수 있습니다.

- DDL을 저장하여 동일하게 정의된 테이블, 데이터베이스 및 다른 데이터베이스에서 색인을 작성합니다(예: 데이터베이스 웨어하우스 응용프로그램).
- DDL을 사용하여 테스트 환경에서 제품 환경으로 또는 하나의 시스템에서 또 다른 시스템으로 데이터베이스 복사
- DDL을 편집하여 유사한 오브젝트 작성

DDL 생성 누름 버튼을 클릭하면, **db2look** 유틸리티가 생성한 명령문과 함께 명령 표시 창이 나타납니다. 명령 표시 창에서, **스크립트 저장** 누름 버튼을 눌러 명령문을 저장할 수 있습니다. 명령문은 스크립트가 됩니다. **생성** 버튼을 클릭하면, 스크립트 수행 창이 열립니다.

주: DDL 명령문을 생성하는 것은 시스템 390의 제어 센터와 작업할 때 다릅니다. 이러한 차이점에 대한 특성에 대해서는 도움말 정보를 참조하십시오.

선택된 스키마 또는 데이터베이스 내에 있는 모든 스키마에 대해 DDL문을 생성할지 여부를 선택할 수 있습니다. 그런 다음, 제품 환경에서 스크립트를 사용하기 전에 변경하려는 경우 스크립트를 편집할 수 있습니다. 생성된 DDL문을 사용하여 같은 데이터베이스를 작성하려면, 간단히 사용자가 생성한 스크립트를 사용하여 새로운 환경에서 수행하십시오.

DDL문을 생성하려면, 다음을 수행하십시오.

- DDL문을 생성하려는 오브젝트를 강조한 후 오른쪽 마우스 버튼을 클릭하십시오.
- DDL** 생성을 선택하십시오. 스크립트 수행 창이 나타납니다.
- 사용자 ID와 암호를 입력한 후 **확인**을 클릭하십시오. **db2look** 명령의 내용이 들어 있는 작업이 작성됩니다. DB2 메시지 창이 새로운 작업의 작업 ID와 함께 나타납니다.
- 확인**을 눌러 메시지 창을 닫으십시오.
- 저널 노트북의 작업 실행기록 페이지를 사용하여 작업 결과를 본 후 해당 작업과 연관되어 저장된 스크립트의 내용을 보십시오.
- 작업을 선택한 후 오른쪽 마우스 버튼을 클릭하십시오. 팝업 메뉴에서 **결과 표시**를 선택하십시오. 작업 결과 창이 열립니다. **db2look** 명령의 결과는 작업 결과 패널에 표시됩니다.
- 스크립트 작성**을 선택하여 결과의 스크립트를 작성하십시오. 새 명령 스크립트 창이 나타납니다.
- 이를 다시 사용하려는 경우 새로운 스크립트를 저장하십시오.

필터

제어 센터에서, 사용자는 내용 패널에 표시된 정보를 거르거나, 실제 결과 세트로서 테이블에서 검색된 정보를 거를 수 있습니다. 하나 이상의 오브젝트에 대해 필터를 작성하여 리턴된 오브젝트 수나 표시된 오브젝트 수를 제한할 수 있습니다. 일단 필터를 설정하면, 트리에 있는 모든 오브젝트를 다시 표시하려는 경우 필터를 제거하거나 삭제해야 합니다.

표시장치 필터링

좀더 효율적인 관리를 위해 내용 패널에 나타나는 오브젝트 수를 줄이려면, 다음을 수행하십시오.

1. 제어 센터 맨 아래에 있는 내용 패널 도구 모음에서 필터 아이콘을 선택하거나 뷰 메뉴 막대에서 필터를 선택하십시오.
2. 오브젝트 수를 줄이기 위해 사용되는 기준을 선택하십시오.
3. 필터 작동 선택란을 선택하여 필터를 활성화하십시오.

나중에 그 내용을 볼 오브젝트를 선택하면, 오브젝트와 연관된 필터가 이전에 설정한 기준에 따라 뷰를 제한합니다.

검색된 데이터 필터링

조회에서 리턴되는 행 수를 줄이고 응답 시간을 향상시키기 위해, 오브젝트를 선택할 때 내용 패널에 표시되는 출력이나 결과 세트를 정의할 수 있습니다.

1. 트리에서 폴더 오브젝트를 선택한 후 오른쪽 마우스 버튼을 클릭하십시오.
2. 팝업 메뉴에서 필터를 선택하십시오. 필터 창이 열립니다.
3. 필터 기능을 사용하여 해당 오브젝트에 속하는 행을 검색하는 기준 세트를 정의하십시오.

데이터의 특정 세트를 검색하기 위한 필터 정의

특정 세트의 데이터를 검색하도록 필터를 정의하려면 다음을 수행하십시오.

1. 제어 센터에서 플랫폼에 종속되는 데이터베이스 또는 서브시스템 폴더를 확장하십시오.
2. 필터를 정의하려는 오브젝트를 선택하십시오. 해당 오브젝트에서 오른쪽 마우스 버튼을 클릭하십시오.
3. 팝업 메뉴에서 필터를 선택하십시오. 필터 노트북이 열립니다.

4. 위치 페이지에서, 선택된 오브젝트의 이름 또는 다른 기술 필터 기준을 지정하십시오. 필터 결과는 제어 센터의 내용 패널에 표시되는 선택된 오브젝트와 연관된 결과 세트입니다.
5. 위치 페이지에서, 단일선택 버튼을 선택하여 위치 페이지에 있는 필드에서 선택된 모든 조건에 부합하게 할지 또는 적어도 하나의 조건에 부합하게 할지 여부를 지정하십시오.
6. 필터 노트북의 고급 페이지에서, 리턴된 행 수를 제한하기 위해 표시되는 텍스트를 편집하여 추가 기준을 사용할 수 있습니다.
7. 확인을 클릭하여 정의된 필터 기준을 사용하십시오.

행 수를 기준으로 이 필터 노트북을 자동으로 호출하려면, 메뉴 막대에서 도구를 선택한 후, 팝업 메뉴에서 도구 설정값을 선택하십시오. 행 수를 초과할 때의 필터링 선택 선택란을 사용하면 선택사항에서 리턴된 행의 임계값을 미리 정의할 수 있습니다. 임계값에 도달하면, 정의된 범주를 기준으로 현재 검색을 제한할 수 있도록 필터 노트북이 나타납니다. 이것은 테이블이 예상치 못하게 커지고 이전에 필터링이 수행되지 않은 경우 특히 유용합니다. 플랫폼 및 데이터에 따라, 행의 부속 집합만이 필요한 경우, 수 많은 행을 리턴할 수도 있습니다.

도움말

확장된 도움말 정보가 관리 도구에서 제공됩니다. 도움말 버튼은 메뉴 도구 모음에서 뿐만 아니라 모든 대화 상자에도 있습니다. 필드를 채우고 작업을 수행하는 방법에 대한 도움말뿐만 아니라, 일반 도움말도 얻을 수 있습니다. 도움말 메뉴에서, 용어 색인 또는 참조 정보와 제품 매뉴얼에서 제공하는 정보에도 액세스할 수 있습니다.

제어 센터

관리의 주 지점으로서 제어 센터를 사용하여 시스템, DB2 인스턴스, 데이터베이스, 그리고 테이블, 뷰 및 사용자 그룹과 같은 데이터베이스 오브젝트를 관리하십시오. 또한, 제어 센터를 사용하여 OS/390용 DB2 서브시스템에 액세스할 수 있습니다. 모든 DB2 데이터베이스는 제어 센터에 나타나기 전에 카탈로그되어야 합

니다. 그림1에서는 제어 센터의 1차 기능을 보여줍니다 운영 체제의 차이로 인해, 시스템에 있는 제어 센터가 그림과 다르게 표시될 수 있습니다.

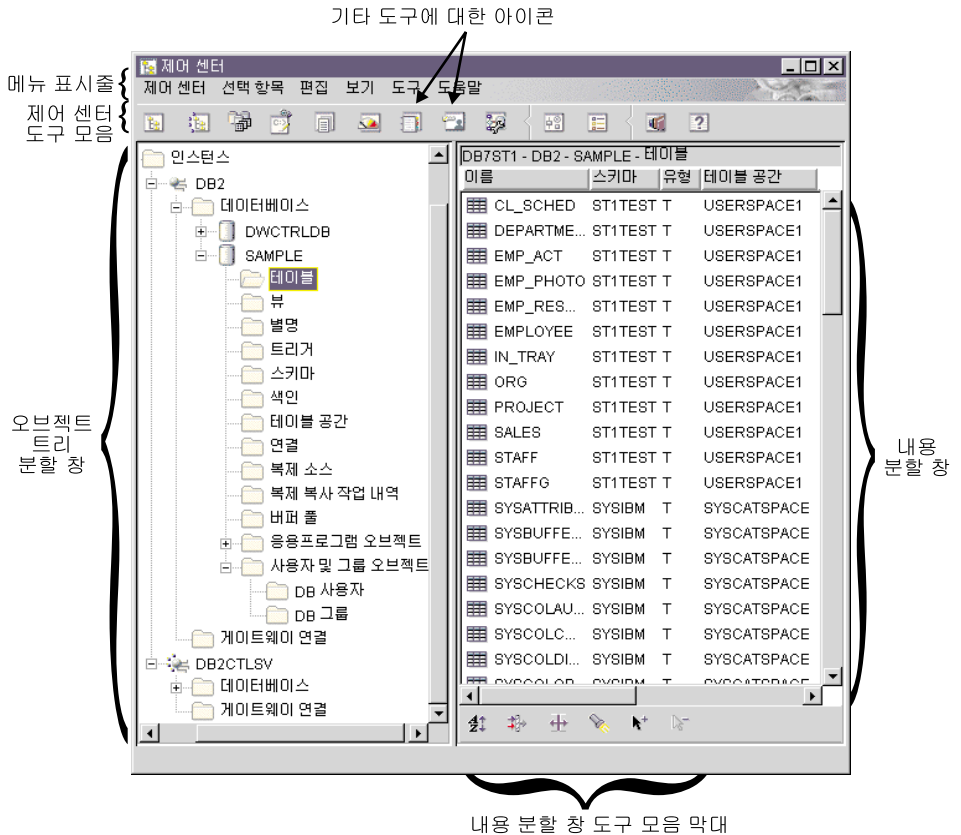


그림1. 제어 센터 기능

제어 센터의 주 요소

제어 센터의 주 요소는 다음과 같습니다.

- 메뉴 막대. 메뉴 막대는 화면의 맨 위에 있습니다. 메뉴 막대에서 메뉴를 선택 하면 DB2 도구 종료, 그래픽 도구 액세스, 온라인 도움말 및 제품 정보에 대한 액세스 확보 같은 많은 기능을 수행할 수 있습니다. 메뉴 막대에 있는 각 항목을 눌러 이러한 기능에 익숙해져야 합니다.

- 제어 센터 도구 모음. 제어 센터 및 기타 도구에 대한 아이콘은 제어 센터 도구 모음에 있습니다. 호버(hover) 도움말은 커서가 아이콘 위에 있을 때 각 아이콘을 식별합니다.

제어 센터 도구 모음에서 도구 설정값 아이콘을 선택하여 이러한 도구에 대한 설정값을 변경할 수 있습니다.

- 오브젝트 트리. 오브젝트 트리는 화면의 왼쪽 패널에 있습니다. 여기에서는 제어 센터에서 관리할 수 있는 모든 데이터베이스 서버와 오브젝트에 대한 아이콘을 표시합니다. 먼저 오브젝트 트리 패널에 나타나기 전에 원격 데이터베이스 서버를 카탈로그해야 합니다. 오브젝트 트리 패널에 있는 일부 오브젝트에는 기타 오브젝트가 들어 있습니다. 오브젝트의 왼쪽에 있는 더하기 부호(+)는 오브젝트가 접기되었음을 나타냅니다. 더하기 부호를 눌러 이를 확장할 수 있습니다. 빼기 부호(-)는 확장될 때 오브젝트의 왼쪽에 나타납니다. 오브젝트를 접으려면, 빼기 부호를 클릭하십시오.
- 내용 패널. 내용 패널은 화면의 오른쪽 패널에 있습니다. 이 패널에서는 오브젝트 트리 패널에서 선택된 오브젝트에 들어 있는 모든 오브젝트를 표시합니다. 예를 들어, 오브젝트 트리 패널에서 테이블 폴더를 선택하면, 모든 테이블이 내용 패널에 표시됩니다. 데이터베이스 폴더를 선택하면, 내용 패널이 모든 데이터베이스를 표시하기 위해 변경됩니다. 내용 패널 도구 모음에 있는 필터 아이콘을 클릭하고 필요한 정보를 지정하여 내용 패널에 나타나는 컬럼을 필터링하거나, 도구 모음에서 도구를 선택하고 도구 설정값을 선택하여 오브젝트를 필터링할 수 있습니다. 필터 작동 선택란이 내용 패널 필터 대화상자에서 선택되어 있는지 확인해야 합니다.
- 내용 패널 도구 모음. 이 도구 모음은 내용 패널의 맨 아래에 나타납니다. 이것으로 내용 패널에 있는 정보를 조정할 수 있습니다. 이 도구 모음은 제품 전체에서 가장 자세한 뷰의 맨 아래 또는 측면에 나타나는 공통 제어입니다.

제어 센터에서 작업할 때 일부 필드는 굵은 빨간색 테두리로 강조표시됩니다. 이 테두리는 사용자가 입력해야 하는 필수 필드임을 나타냅니다. 값을 선택하거나 값을 입력하고 나면, 빨간색 테두리가 사라집니다.

OS/390용 DB2에서 사용자 정의된 제어 센터 사용

사용자가 정의한 관리 지점으로서 OS/390용 DB2 플랫폼에서 사용자 정의된 제어 센터를 사용하여 테이블, 뷰 및 데이터베이스사용자와 같은 데이터베이스 오브젝트, 서브시스템 및 데이터베이스를 관리하십시오. 이 사용자 정의된 제어 센터를 사용하여 사용자가 정의한 OS/390용 DB2 오브젝트에 액세스할 수 있습니다.

사용자 정의된 제어 센터의 주 요소는 기본 제어 센터에 대해 나열한 것과 같습니다. 사용자 정의된 제어 센터에서는 개인화된 제어 센터에 포함시키려는 오브젝트를 지정할 수 있습니다. 이 사용자 정의 트리는 저장된 후 DB2 오브젝트를 관리하기 위해 호출될 수 있습니다. 사용자 정의 트리는 모든 사용자에게 대한 기본값인 제어 센터 트리를 대체하지는 않지만, 제어 센터가 호출될 때마다 같은 방식으로 오브젝트 세트에 액세스하려는 경우에는 유용합니다. 필요한 만큼 사용자 정의된 트리를 작성할 수 있습니다. 각각에는 다른 세트의 오브젝트가 들어 있으며, 이들은 사용자가 지정한 방식으로 정렬될 수 있습니다.

사용자 정의된 트리를 사용하면 DB2 오브젝트의 고정 계층 구조를 탐색해야 하는 노력이 줄어들고, 관련 오브젝트를 그룹화하는 방법도 제공합니다. 예를 들어, 급여 정보가 있는 테이블만 포함하는 트리를 정의할 수도 있습니다.

관리될 수 있는 시스템

제어 센터에서, OS/2, Windows 및 DB2 UNIX 플랫폼용 Universal Database 제품군에 대한 데이터베이스 오브젝트를 관리할 수 있습니다. 빠른 시작 책에서 플랫폼 특유의 설치 및 설정 정보를 참조하십시오.

또한 AS/400용 DB2, VSE 및 VM 시스템용 DB2, OS/390용 DB2에서 DB2 Universal Database 제품군으로 데이터를 복제할 수 있습니다. 복제 안내 및 참조서 매뉴얼에서 제품 간의 복제에 대한 자세한 내용을 참조하십시오.

관리될 수 있는 오브젝트

제어 센터에서 오브젝트를 관리하려는 경우, 이를 오브젝트 트리로 추가해야 합니다. 데이터베이스를 제거하거나 제어 센터 외부로 이를 카탈로그 해제한 다음, 제어 센터를 사용하여 이에 대해 작업을 수행하려는 경우, 이를 오브젝트 트리로 추가해야 합니다.

제어 센터에서 관리할 수 있는 DB2 Universal Database 오브젝트는 다음과 같습니다.

- 시스템
- 인스턴스
- 테이블
- 뷰
- 색인
- 트리거
- 사용자 정의 유형
- 사용자 정의 함수
- 패키지
- 별명
- 복제 오브젝트
- 사용자 및 그룹

제어 센터에서 관리할 수 있는 OS/390용 DB2 버전 5 오브젝트는 다음과 같습니다.

- 버퍼 풀
- 뷰
- 카탈로그 테이블
- 저장영역 그룹
- 별명
- 유사어
- DB2 사용자
- 위치
- 응용프로그램 오브젝트(컬렉션, 패키지, 플랜, 프로시저어)
- 데이터베이스
- 테이블
- 테이블 공간

- 색인
- 복제 소스
- 복제 복사 작업 내역

OS/390용 DB2 버전 6의 경우, 제어 센터에서 관리할 수 있는 오브젝트는 버전 5용으로 언급한 오브젝트 모두와, 다음과 같은 것이 있습니다.

- 스키마
- 트리거
- 사용자 정의 함수
- 구별 유형

이러한 오브젝트 각각에 대해 수행할 수 있는 조치를 보려면, 오브젝트 패널에서 오브젝트를 선택한 후 오른쪽 마우스 버튼을 클릭하십시오. 팝업 창이 함수를 나열하며 나타냅니다.

제어 센터에서 시스템 표시

시스템에 카탈로그된 모든 시스템과 설치된 DB2를 표시하려면, 다음을 수행하십시오.

1. 시스템 옆에 있는 더하기 부호(+)를 클릭하여 오브젝트 트리를 확장하십시오. 지역 머신 및 원격 머신을 나타내는 아이콘이 표시됩니다. 지역 시스템은 지역 아이콘에 의해 표현됩니다. 지역 아이콘은 지역 머신이 DB2 서버인 경우에만 나타납니다. 지역 아이콘에서 오른쪽 마우스 버튼을 클릭하면, 팝업 메뉴에 있는 옵션 중 하나에 관리 서버에 접속이 있습니다. 관리 서버를 사용하면, 성능 모니터 및 스케줄링과 같은 기능을 이용할 수 있습니다. 이는 DB2 서비스 요청을 충족시키기 위해 DB2 관리 도구에 의해 사용되며, 자동으로 작성되고 시작됩니다. DB2 관리 서버(DAS)용 기본 이름은 플랫폼별로 다양합니다. 예를 들어, Windows 및 OS/2 플랫폼에서, 『DB2DAS00』가 사용되며, AIX에서 『db2as』가 사용됩니다.
2. 지역 아이콘을 확장하십시오. 지역 머신에서 DB2의 인스턴스는 트리 구조로 표시됩니다.

OS/2, Windows 및 지원되는 DB2 UNIX 기반의 시스템에서, 데이터베이스 관리 프로그램 코드의 각 사본을 별개의 인스턴스로서 간주할 수 있으며, 이것

은 사용자 머신의 디렉토리에 저장됩니다. OS/390용 DB2에서, 인스턴스는 서브시스템을 나타냅니다. 기본 지역 인스턴스는 DB2를 설치할 때 작성됩니다. 단일 시스템에 여러 인스턴스가 있을 수 있습니다. 이러한 인스턴스를 사용하여 개발 환경을 제품 환경과 구분하거나, 관련 정보를 특별한 그룹에게만 한정할 수 있습니다. 또한, 특별한 환경에 맞도록 인스턴스를 조정할 수 있습니다.

3. 인스턴스 아이콘을 확장하십시오. 존재하는 각 데이터베이스에 대해 아이콘과 이름이 표시됩니다.

OS/390용 DB2 오브젝트 관리

제어 센터를 사용하면, 기존 OS/390용 DB2 버전 5 및 OS/390용 DB2 UDB 버전 6 제품의 많은 기능, 즉 데이터를 재구성하거나 로드하는 수행 유틸리티뿐만 아니라 오브젝트 작성, 변경 및 삭제와 같은 기능을 수행할 수 있습니다. 그러나, 제어 센터에서 OS/390용 DB2 서브시스템을 관리하기 전에, 이에 대한 연결을 구성하여 오브젝트 트리로 추가해야 합니다.

OS/390용 DB2 서브시스템 추가

클라이언트 구성 지원 프로그램이 설치되어 있으면, 이를 사용하여 OS/390용 DB2 시스템에 쉽게 연결을 구성할 수 있습니다. 클라이언트 구성 지원 프로그램이 설치되어 있지 않으면, 명령행 처리기(CLP)를 사용하여 OS/390용 DB2에 대한 연결을 수동으로 구성해야 합니다.

클라이언트 구성 지원 프로그램을 사용하여 LAN에서 클라이언트가 사용할 수 있는 OS/390용 DB2 시스템 모두에 대한 네트워크를 검색하십시오. OS/390용 DB2 시스템 중 하나를 추가하려는 경우, 데이터베이스 추가 마법사를 사용하여 시스템을 추가하거나, 프로파일을 사용하여 연결을 가져오거나 연결을 수동으로 추가할 수 있습니다.

네트워크를 검색하도록 선택한 경우, 네트워크에 있는 DB2 Connect 제품에 시스템에 대해 정의된 연결이 있어야 합니다. 액세스 프로파일을 사용하도록 선택한 경우, 프로파일에서 시스템을 나타내는 DB2 Connect 연결을 선택할 필요가 있습니다. 수동으로 연결을 구성해야 하는 경우, 호스트 이름과 TCP/IP에 대한 포트 번호 또는 SNA용 기호 대상 이름과 같은 통신 프로토콜 매개변수, 시스템 이름 및

통신 프로토콜을 알아야 합니다. OS/390용 DB2 시스템을 추가하고 나면, DB2 Connect 서버 연결용 오브젝트가 제어 센터의 지역 시스템에 표시됩니다.

OS/390용 DB2 버전 5 이후 시스템을 추가하면, 이것은 제어 센터 오브젝트 트리의 자체 섹션에 나타납니다. 특정 시스템에 상주하는 OS/390용 DB2 및 기타 데이터베이스 오브젝트를 보려면, OS/390용 DB2 시스템을 나타내는 OS/390용 DB2 시스템 아이콘에서 오브젝트 트리를 확장하십시오.

특정 오브젝트에 대해 수행할 수 있는 조치 목록을 보려면, 오브젝트 트리에 나타나 있는 오브젝트를 선택한 후 오른쪽 마우스 버튼을 클릭하십시오. 팝업 메뉴가 나타나 해당 오브젝트에 대해 수행할 수 있는 사용 가능한 조치를 보여줍니다. 예를 들어, 내용 보기, 이에 대한 특권 수정 및 이와 연관된 기타 오브젝트 목록 표시뿐만 아니라 뷰를 작성, 변경 또는 삭제할 수 있습니다. 수행할 수 있는 기타 기능에 대해서는 OS/390용 DB2 오브젝트에 대한 온라인 도움말을 참조하십시오.

게이트웨이 연결 관리

DB2 Connect 서버가 카탈로그되면, 게이트웨이 연결 폴더가 지역 시스템의 인스턴스 오브젝트 아래에 있는 제어 센터 오브젝트 트리에서 표시됩니다. 게이트웨이 연결 폴더에는 지역적으로 카탈로그화된 호스트 및 AS/400 데이터베이스로의 연결을 관리하는 데 사용되는 오브젝트 계층 구조가 들어 있습니다. 이러한 연결 관리 오브젝트와 연관된 조치는 호스트와 AS/400 데이터베이스의 연결을 나열하고, 강제 수행하고, 모니터링하는 데 사용될 수 있습니다.

게이트웨이 연결 폴더 아래에 있는 오브젝트 트리는 호스트와 AS/400 데이터베이스간의 연결을 관리하는 데 사용되지만 데이터베이스 관리 태스크에 대해서는 사용되지 않습니다. 그러나, 지역 시스템에 호스트 또는 AS/400 데이터베이스를 추가, 변경 또는 제거해야 할 경우에는, 클라이언트 구성 지원 프로그램을 사용하십시오.

제어 센터에서 수행할 수 있는 기능

제어 센터에서, 다음을 수행할 수 있습니다.

- 데이터베이스 오브젝트 관리. 데이터베이스, 테이블 공간, 테이블, 뷰, 색인, 트리거, 스키마를 작성, 변경 및 삭제할 수 있습니다. 또한, 사용자를 관리할 수도 있습니다.

- 데이터 관리. 데이터를 로드, 가져오기, 내보내기, 재구성하고, 통계를 수집할 수 있습니다.
- 작업 스케줄. 작업은 스크립트 실행을 보류, 수행 또는 완료할 수 있습니다. 특정 시간에 시작되도록 작업을 스케줄할 수 있습니다.
- 데이터베이스를 백업 및 복원하여 예방 유지보수 수행
- 성능 모니터 및 문제점 해결 수행
- 데이터 복제
- 인스턴스와 데이터베이스 구성 및 조정
- DB2 Connect 서버 및 서브시스템과 같은 데이터베이스 연결관리. 응용프로그램 관리
- Explain SQL을 사용하여 조회를 분석하고 액세스 플랜 참조
- 제어 센터를 통해 메뉴 및 텍스트를 표시하는 데 사용되는 글꼴을 변경하십시오. 사용 가능한 글꼴 중 하나로 변경하고, 글꼴의 크기 및 표시된 색상을 변경할 수 있습니다. 제어 센터의 변경사항이 유효하려면 재시작되어야 합니다.
- 기타 도구 시작. 예를 들어, 위성 관리 센터 또는 명령 센터를 시작할 수 있습니다.

오브젝트에 대해 수행할 수 있는 모든 조치를 보려면, 간단히 오브젝트 트리 패널 또는 내용 패널에서 오브젝트를 선택한 후 오른쪽 마우스 버튼을 클릭하십시오. 팝업 메뉴가 나타나고 해당 유형의 오브젝트에 대해 수행할 수 있는 모든 기능을 표시합니다. 예를 들어, 테이블 폴더를 선택하면, Wizard의 도움과는 관계없이 새로운 테이블을 작성하고, 테이블 성능을 모니터하며, 내용 패널에 표시되는 테이블을 필터링할 수 있습니다. 수행할 수 있는 기능은 선택한 오브젝트에 따라 다릅니다.

내용 패널에 있는 오브젝트에서 오른쪽 마우스 버튼을 클릭하여 특정 오브젝트에 대한 추가 기능을 수행하십시오. 예를 들어, 내용 패널에서 테이블 중 하나를 선택하여 오른쪽 마우스 버튼을 클릭하면, 팝업 창이 해당 테이블에 대해 사용할 수 있는 기능을 표시합니다.

새로운 오브젝트 작성

새로운 오브젝트를 작성하려면, 다음을 수행하십시오.

1. 데이터베이스 폴더를 확장하십시오. 오브젝트 유형이 폴더 아이콘으로 표시됩니다.
2. 오브젝트에 대한 폴더 아이콘에서 오른쪽 마우스 버튼을 클릭하십시오. 예를 들면, 테이블 아이콘을 클릭하십시오. 팝업 메뉴가 표시됩니다. 일부 오브젝트의 경우, 기능을 수행할 두 개의 옵션이 있습니다. 하나의 옵션은 마법사를 사용하는 것입니다. 수행할 수 있는 모든 기능에 대한 마법사가 있는 것은 아닙니다.
3. 작성을 선택하십시오. 테이블을 작성하는 마법사가 있으므로, 두 개의 옵션을 갖게됩니다. 하나는 마법사를 사용하는 테이블을 작성하는 것입니다. 마법사 옵션을 선택하면, 정보를 입력하도록 프롬프트가 표시되고 작성해야 하는 선택사항에 대한 제안사항이 제공됩니다. 마법사는 특히 데이터베이스 오브젝트를 자주 작성하지 않는 사용자 또는 새로운 사용자에게 특히 유용합니다.

기존 오브젝트에 대한 작업

오브젝트 트리 패널에서 테이블 폴더와 같은 오브젝트를 클릭하면, 기존의 모든 테이블이 내용 패널에 나타납니다. 그런 다음, 작업하려는 테이블을 선택하고, 오른쪽 마우스 버튼을 클릭하여 해당 특정 테이블에 대해 수행하려는 기능을 호출할 수 있습니다.

제어 센터 사용에 관한 자세한 정보는, 도움말 메뉴로 가거나 제어 센터에 어디에서든 F1을 누르면 온라인 도움말을 얻을 수 있습니다.

오브젝트 위치(OS/390용 DB2 전용)

위치 노트북을 사용하여 데이터베이스 또는 서브시스템 오브젝트를 쉽게 검색할 수 있습니다. 이것으로 다음을 수행할 수 있습니다.

- 제어 센터의 트리 구조를 탐색하지 않고도 오브젝트를 찾을 수 있습니다. 오브젝트는 데이터베이스 또는 서브시스템, 테이블 공간 또는 데이터베이스와 테이블 및 지원 오브젝트 내에 있을 수 있습니다.
- 서브시스템 내에 있는 여러 데이터베이스에서 오브젝트(테이블 공간, 테이블 및 색인)를 찾을 수 있습니다.

위치 노트북의 위치 페이지를 사용하여 검색 기준을 지정하십시오. 위치 노트북의 고급 페이지를 사용하여 검색을 사용자 정의하십시오. 고급 페이지에서 제공되는 텍스트를 편집하고 검색 기준을 추가 또는 수정하십시오.

데이터베이스 또는 OS/390용 DB2 서브시스템 내에 정의된 오브젝트를 찾으려면, 다음을 수행하십시오.

1. 제어 센터에서, 오른쪽 마우스 버튼으로 오브젝트를 클릭하십시오. 팝업 메뉴에서 위치를 선택하십시오. 위치 노트북이 열립니다.
2. 오브젝트 유형 필드에서, 검색할 데이터베이스 오브젝트의 유형을 선택하십시오. 사용 가능한 목표 오브젝트 목록은 검색을 시작하는 오브젝트에 따라 다릅니다.
3. 위치 페이지에서 검색 기준으로 채우십시오. 적어도 하나의 검색 기준을 입력해야 하고, 와일드카드를 사용하면 검색에 도움이 됩니다. 문자는 유효한 분리 문자를 사용하여 소문자 또는 확장된 문자 세트를 묶지 않으면 대문자로 전환됩니다.
4. 위치 페이지에서, 단일선택 버튼을 선택하여 위치 페이지에 있는 필드에서 선택된 모든 조건에 부합하게 할지 또는 적어도 하나의 조건에 부합하게 할지 여부를 지정하십시오.
5. 확인을 클릭하여 검색 기준을 사용하십시오. 검색 결과는 위치 결과 창에 표시됩니다. 출력 테이블의 형식은 검색한 오브젝트 유형에 따라 달라집니다.
6. 같은 기준 또는 다른 기준으로 검색을 반복하려면, 적용을 클릭하십시오.
7. 위치 결과 창에 나타나는 행을 선택하고, 해당 행을 오른쪽 버튼으로 클릭하여 수행할 수 있는 추가 조치가 있는 팝업 메뉴를 볼 수 있습니다.

위성 관리 센터

위성 관리 센터는 DB2 제어 센터에서 사용할 수 있는 도구 세트입니다. 이를 사용하면 중앙 지점에서 DB2 서버 컬렉션을 설정하고 관리할 수 있습니다. 하나의 그룹에 속하는 DB2 서버를 위성이라고 합니다. 중앙 위치에서 위성을 관리한다는 것은 DB2가 DB2 위성을 사용하는 사용자에게 보이지 않으므로 사용자는 데이터베이스 관리에 대해 배우지 않아도 됨을 의미합니다.

그룹을 사용하여 DB2 서버에서 수행되는 응용프로그램이나, 응용프로그램을 지원하는 데이터베이스 구성과 같은 특성을 공유하는 DB2 서버를 구성하십시오. DB2 서버는 데이터베이스 구성, 이용 및 목적면에서 비슷합니다.

DB2 서버를 함께 그룹화함으로써, DB2 서버들을 각각 개별적으로 관리하는 대신 DB2 서버 그룹을 관리할 수 있습니다. 기존 DB2 서버 그룹과 같은 기능을 제공하는 추가 DB2 서버를 확보하면, 위성 관리 센터를 사용하여 해당 그룹에 이를 추가할 수 있습니다.

위성 관리 센터에서, 그룹, 위성, 응용프로그램 버전, 일괄처리 및 인증을 작성할 수 있습니다. 또한 성공 코드 세트를 정의하고, 위성 환경의 관리와 연관된 기타 기능을 수행할 수도 있습니다. 위성 환경에 관한 정보는 위성 제어 데이터베이스라고 하는 중앙 데이터베이스에 저장됩니다. 이 데이터베이스는 환경에 어떤 위성이 있는지, 각 위성이 어느 그룹에 속하는지와 위성이 수행중인 일반 사용자 응용 프로그램이 무슨 버전인지를 기록합니다. 이 데이터베이스는 DB2 제어 서버라고 하는 DB2 서버상에 있습니다.

위성 관리 센터의 기능을 수행할 수 있으려면, 먼저 위성 제어 데이터베이스 (SATCTLDB)를 제어 센터에 카탈로그화해야 합니다. 사용 가능하게 되면, 위성 제어 센터를 사용하여 위성이 자체 응용프로그램 버전을 동기화할 때 실행하는 위성, 그룹 및 일괄처리를 설정하고 유지보수할 수 있습니다.

데이터베이스 구성을 설정하고 유지보수하기 위해, 각 위성은 위성 제어 데이터베이스로 연결하여 일반 사용자 응용프로그램 버전에 해당되는 일괄처리를 다운로드 합니다. 위성은 이들 일괄처리를 지역적으로 실행한 다음, 결과를 다시 위성 제어 데이터베이스로 보고합니다. 이러한 일괄처리를 다운로드하고, 실행한 다음, 일괄처리 실행의 결과를 보고하는 프로세스를 동기화라고 합니다. 위성은 그룹에 속하는 다른 위성과의 일관성을 유지보수하고, 같은 버전의 일반 사용자 응용프로그램을 수행합니다.

명령 센터

도구 모음에서 명령 센터 아이콘을 클릭하여 제어 센터에서 명령 센터를 시작할 수 있습니다.

명령 센터를 사용하면 다음을 수행할 수 있습니다.

- 결과 창에서 하나 이상의 SQL문 및 DB2 명령의 결과 출력을 보십시오. 결과를 화면이동하고 보고서를 생성할 수 있습니다.
- 명령 스크립트를 작성하고, 스크립트 센터에 저장하십시오. 명령 스크립트를 편집하여 새로운 스크립트를 작성할 수 있습니다. 스크립트 센터에서, 명령 스크립트는 사용자가 지정한 언제든지 작업으로서 수행되도록 스케줄될 수 있습니다.
- SQL문, DB2 명령 및 운영 체제 명령을 수행하십시오. 명령 센터에서 DB2 명령을 수행할 때, **DB2**가 명령 앞에 올 필요는 없습니다. REXX와 같은 지원되는 운영 체제 스크립트 언어로 운영 체제 명령을 수행할 수 있으며, 이 앞에는 느낌표(!)가 옵니다. 명령 센터를 사용하여 명령 및 명령문을 수행하면 각각의 명령을 개별적으로 입력하고 수행하지 않고도 많은 명령을 한 번에 실행할 수 있습니다.
- 주 도구 모음에서 제어 센터와 같은 DB2 관리 도구에 대한 빠른 액세스를 확보하십시오.
- 실행하기 전에 SQL문과 연관된 액세스 플랜 및 통계를 참조하십시오.

스크립트 센터

제어 센터 도구 모음에서 아이콘을 선택하여 스크립트 센터를 시작할 수 있습니다. 스크립트 센터는 명령 및 명령문 세트를 기록하여 스크립트를 작성할 수 있도록 하는 도구로서, 사용자가 필요할 때마다 수행하도록 스케줄할 수 있습니다. 이전에 작성한 스크립트 또는 명령 센터에 저장된 스크립트를 가져올 수 있습니다. 저장된 스크립트 세트에서 스크립트를 선택하거나, 기존 스크립트를 편집하여 새로운 스크립트 작성, 스크립트 복사 또는 스크립트 제거를 수행할 수 있습니다.

사용자의 편집기를 사용하여 스크립트 센터 내부 또는 스크립트 센터 외부에서 스크립트를 편집할 수 있습니다. 스크립트 센터 내에서 스크립트를 수행하는 경우, 저널에 결과가 기록되도록 하는 장점이 있습니다.

스크립트 센터에 있는 스크립트에서 운영 체제 명령을 수행하려면, 다음을 수행하십시오.

1. 스크립트 → 새로 작성을 선택하십시오 새로운 명령 스크립트 창이 나타납니다.
2. 스크립트 유형의 경우, OS 명령 단일선택 버튼을 선택하십시오.
3. 스크립트 이름, 설명 및 작업 디렉토리를 입력하십시오.
4. 명령을 입력하십시오.
5. 확인을 클릭하십시오.

스크립트 센터에서, 시스템이 알고 있는 모든 명령 스크립트에 대한 설명 및 스크립트 유형과 같은 정보를 본 후, 다음 작업을 수행할 수 있습니다.

- DB2 및 운영 체제 명령이 들어 있는 명령 스크립트 작성
- 저장된 명령 스크립트를 즉시 수행
- 나중에 또는 정기적으로 수행되도록 스크립트 스케줄. 예를 들어, 여러 테이블에 대한 통계를 수집하는 스크립트를 작성할 수도 있습니다. 그런 다음, 작업이 밤에 수행되도록 작업을 스케줄할 수 있습니다. 작업을 수행하려는 시, 일, 주, 월, 여러 요일 또는 날짜를 지정하여 지정된 간격으로 무인상태에서 수행되도록 작업을 스케줄할 수 있습니다. 작업은 스크립트를 스케줄하거나 스크립트를 수행할 때마다 작성됩니다.
- 도구 모음에서 저널에 액세스하여 특정 스크립트를 사용하는 작업 열람 및 스케줄된 모든 작업의 상태 열람
- 저장된 명령 스크립트 편집

스크립트 센터에서 기존 스크립트 사용

스크립트 센터에서 작성되지 않은 기존의 스크립트에 대해 스크립트 센터를 사용하려면, 다음을 수행하십시오.

1. 제어 센터 도구 모음에서, 스크립트 센터 아이콘을 클릭하십시오. 스크립트 센터가 열립니다.
2. 스크립트 → 가져오기를 선택하십시오 파일 브라우저 창이 열립니다.

3. 기존 스크립트 파일을 선택한 후 확인을 클릭하십시오. 새로운 명령 스크립트 창이 나타납니다. 스크립트 편집기인 창의 맨 아래에 스크립트가 표시됩니다. 인스턴스, 스크립트 이름, 스크립트 설명 및 작업 디렉토리 필드를 완성한 후, 스크립트 유형을 선택하십시오.
4. 확인을 클릭하십시오. 스크립트 센터에서 스크립트가 작성됩니다.

저장된 명령 스크립트를 수행하도록 스케줄

스크립트를 스케줄하려면, 다음을 수행하십시오.

1. 제어 센터 도구 모음에 있는 스크립트 센터 아이콘을 클릭하십시오. 스크립트 센터가 열립니다.
2. 실행하도록 스케줄하려는 스크립트를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 스케줄을 선택하십시오. 스케줄러 창이 열립니다.
3. 작업 빈도와, 완료 메시지 또는 시작할 다른 명령 스크립트 같은 완료 조치를 선택하십시오.
4. 확인을 클릭하십시오. 이것으로 저널에서 추적할 수 있는 보류된 작업이 시작됩니다.

저널

제어 센터 도구 모음에서 아이콘을 선택하여 저널을 시작할 수 있습니다. 저널을 사용하면 작업을 모니터링하고 결과를 검토할 수 있습니다. 저널에서, 복구 실행기록 및 DB2 메시지를 표시할 수도 있습니다. 저널을 사용하면 보류중인 작업, 수행중인 작업 및 작업 실행기록을 모니터링하고, 결과를 검토하며, 복구 실행기록 및 경보 메시지 표시, DB2 메시지 로그를 표시할 수 있습니다.

작업에 대한 작업

저널을 사용하여 작업을 수행하십시오. 저널을 열려면, 다음을 수행하십시오.

1. 스크립트 센터 도구 모음에서 저널 아이콘을 클릭하십시오. 저널이 열립니다.
2. 나중에 수행되도록 스케줄된 작업을 보려면, 보류중인 작업 누름 버튼을 클릭하십시오. 보류중인 작업 목록에서 작업을 보게 됩니다. 또한, 작업에 대한 모든 정보도 볼 수 있습니다. 보류중인 작업에 대해 재스케줄, 작업과 연관된 스

크립트 표시 또는 즉시 수행 등과 같은 조치를 수행할 수 있습니다. 저장된 스크립트가 수정되면, 이에 종속되는 모든 작업은 새로 수정된 동작을 상속합니다.

저널에서, 현재 수행중인 작업과 작업 실행기록을 볼 수도 있습니다.

저널 창에 있는 다른 페이지는 다음과 같습니다.

- 복구 페이지. 이 페이지에서는 복구 실행기록(백업, 복원 조작 및 로드 조작에 대한 세부사항)을 표시하고, 복구 로그를 복원할 수 있도록 합니다.
- 경보 페이지. 이 페이지에서는 모든 경보를 표시합니다.
- 메시지 페이지. 이 페이지에서는 DB2 관리 도구를 통해 발행된 모든 메시지를 표시합니다.

저널에 대한 온라인 도움말에서는 작업 및 로그에 대한 작업의 세부 단계를 제공합니다.

사용권 센터

사용권 센터를 사용하여 시스템에 설치된 DB2 제품에 대한 사용권 상태와 이용 정보를 표시합니다. 사용권 센터를 사용하여 사용권 모니터링에 적합하도록 시스템을 구성할 수도 있습니다. 사용권 센터에서는 다음을 수행할 수 있습니다.

- 새로운 사용권 추가
- 제품의 시운용 사용권에서 영구 사용권으로 업그레이드
- 사용권의 세부사항 보기

사용권 정보에 대한 세부사항을 보면 다음을 알 수 있습니다

- 제품 이름
- 버전 정보
- 만기일
- 등록된 사용자
- 동시 사용자
- 책임 있는 사용자 수
- 동시 사용자 수

- 강제 정책
- 프로세서의 수(Enterprise Edition 및 Enterprise – Extended Edition의 경우).

경보 센터

경보 센터는 시스템을 모니터링하여 잠재적인 문제점을 알려주는 도구입니다. 임계값을 초과하여 경고 상태에 있는 모니터링된 오브젝트를 표시하기 위해 경보 센터가 자동으로 열리도록 설정할 수 있습니다. 성능 모니터를 사용하여 임계값을 설정하며, 이것은 제어센터에서 호출됩니다. 아이콘 색은 경고의 심각도를 나타냅니다. 빨간 아이콘은 경보를 나타냅니다. 노란 아이콘은 경고를 나타냅니다. 성능 변수에서 리턴된 데이터가 표시됩니다. 데이터를 분석하는 방법에 대해서는 온라인 도움말을 참조하십시오.

클라이언트 구성 지원 프로그램

클라이언트 구성 지원 프로그램(CCA)은 원래 지역 또는 원격 DB2 서버에 클라이언트를 설정하도록 돕는 마법사가 들어 있는 도구입니다. 그러나, 또한 DB2 Connect 서버 구성을 쉽게 도울 수 있도록 사용될 수 있습니다.

클라이언트 구성 지원 프로그램은 응용프로그램이 연결할 수 있는 데이터베이스의 목록을 유지보수하게 합니다. 클라이언트 구성 지원 프로그램은 이 타스크의 본래의 복잡성으로부터 사용자를 보호하는 동안 노드 및 데이터베이스를 카탈로그화합니다.

클라이언트 구성 지원 프로그램에서 다음과 같은 타스크를 수행할 수 있습니다.

- 데이터베이스 연결 항목을 추가, 수정, 삭제하십시오.
- 선택한 데이터베이스에 대한 연결을 테스트하십시오.
- 데이터베이스 관리 프로그램 구성 매개변수를 구성하십시오.
- CLI/ODBC 설정을 구성하십시오.
- 선택된 데이터베이스에 DB2 유틸리티 및 기타 응용프로그램을 바인드하십시오.
- 구성 정보를 가져오기/내보내기하십시오. 이것은 새로운 머신을 구성하기 위해 이전에 구성된 머신에서 기존 구성을 사용하게 합니다.

- 선택한 데이터베이스에 연결하기 위해 사용하는 사용자 ID의 암호를 변경하십시오.

클라이언트 구성 지원 프로그램은 새로운 데이터베이스 연결 항목을 추가할 때 도움이 되도록 다음과 같은 메소드를 제공합니다.

- 프로파일을 사용하십시오. 프로파일은 이전에 구성된 머신에서 내보내져서, 새로운 머신을 구성하기 위해 사용될 수 있습니다. 서버 프로파일은 제어 센터에서 내보내기될 수 있으며, 클라이언트 또는 서버 프로파일은 CCA에서 내보내기될 수 있습니다.
- 네트워크를 검색하십시오. CCA는 수행 중인 관리 서버를 갖는 DB2 시스템의 네트워크를 검색할 수 있습니다. 검색 및 알려진(또는 직접) 발견 모드가 제공됩니다. 각 발견 모드는 네트워크 구성 제한사항을 따릅니다. 보통 네트워크 라우터는 검색 발견 기능 요청이 전송되도록 허용하지 않습니다. 알려진 발견 기능은 원하는 서버 시스템을 찾기 위해 몇가지 정보만을 요구합니다. 이전에 게이트웨이에 정의된 AS/400 시스템이나 호스트도 찾을 수 있습니다.
- 데이터베이스 연결을 수동으로 구성합니다. 모든 정보가 제공되어야 하지만 마법사가 작업을 간단히 만들기 위해 시작됩니다.

성능 모니터

성능 모니터는 DB2 Universal Database 상태 및 이것이 제어하는 데이터에 대한 정보를 제공합니다. 이것은 데이터베이스 환경에 맞도록 조정될 수 있는 그래픽 유틸리티입니다. 성능 모니터가 수집한 값이 허용 범위를 벗어나면 트리거가 경고하거나 정보를 보내는 임계값이나 영역을 정의할 수 있습니다.

오브젝트 트리 패널 또는 내용 패널에서 오브젝트를 선택하고 오른쪽 마우스 버튼을 클릭하여 인스턴스, 데이터베이스, 테이블, 테이블 공간 및 연결과 같은 DB2 오브젝트를 모니터할 수 있습니다. 여기에서, 모니터 활동을 시작하도록 선택할 수 있습니다.

오브젝트가 모니터되고 있는 경우에는, 아이콘 색상이 초록색, 노란색 또는 빨간색으로 표시되어 모니터 상태를 나타냅니다. 색상은 사용자가 설정한 임계값으로 정의된 문제점의 심각도를 나타냅니다. 초록색은 모니터가 수행중이며 모든 것이 양호함을 나타냅니다. 노란색은 경고가 있으며, 모니터가 설정된 임계값에 도달하

고 있음을 나타냅니다. 빨간색은 경보를 나타내고, 모니터가 임계값에 도달했음을 나타냅니다. DB2와 함께 제공되는 사전 정의된 모니터를 사용하거나, 자체 모니터를 작성할 수 있습니다.

성능 모니터가 수집한 정보를 보려면, 오브젝트를 오른쪽 마우스 버튼으로 클릭한 후 팝업 창에서 **모니터 활동** 표시를 선택하십시오.

성능 모니터에 있는 정보를 사용하여 다음을 수행하십시오.

- 성능 문제점 발견
- 최적의 성능을 위한 데이터베이스 조정
- 성능 추세 분석
- 데이터베이스 응용프로그램의 성능 분석
- 문제점 발생 예방

성능 모니터를 사용하면 디스크 활동, 버퍼 풀 이용, 프리페치 양, 잠금 이용 및 특정 간격의 레코드 블로킹과 같은 데이터베이스 디스크의 시각적 표현을 작성하여 추세를 분석할 수 있습니다.

기본 문제점을 모니터해야 하거나 시스템 성능을 관찰하려는 경우 도구를 사용합니다. 이를 사용하면 특정 시점의 데이터베이스 활동의 스냅샷과 성능 데이터를 확보할 수 있습니다. 이러한 스냅샷은 시간별 비교에 사용됩니다. 그래프에 있는 각 지점은 데이터 값을 나타냅니다. 스냅샷을 확보하는 과정은 34 페이지의 『특정 시점에서 성능 모니터』에서 제공됩니다. 이 정보는 잠재적인 문제점을 식별하고 분석하거나, 설정한 임계값을 근거로 예외 조건을 식별하는 데 도움이 될 수 있습니다. 특정 시점에서 데이터베이스 관리 프로그램 및 데이터베이스 응용프로그램의 성능을 알아야 하는 경우 성능 도구를 사용하고 시간별 추세를 관찰하십시오. 또한, 이를 사용하여 경보 상태에 있는 요소의 시각적 개요를 얻을 수도 있습니다. 이는 조정이 필요한 매개변수를 식별하는 데 도움이 됩니다. 그런 다음, 요소에 대해 설정된 매개변수를 면밀히 조사하고 이를 변경하여 성능을 향상시킬 수 있습니다.

이벤트 모니터

특정 시점의 스냅샷을 취하는 것과는 반대로, 이벤트 모니터는 일정 기간 동안 데이터베이스 활동에 대한 정보를 수집합니다. 이렇게 수집된 정보는 특정 데이터베

이스 이벤트(예: 데이터베이스 연결 또는 SQL문)에 대한 활동 요약을 제공합니다. 이벤트 모니터는 특정 이벤트가 발생하는 시간에 데이터베이스의 상태를 기록합니다. 이것으로 데이터베이스에서 활동 추적을 확보할 수 있습니다. 이벤트 모니터 레코드는 데이터를 캡처한 다음 저장되고 분석됩니다. 트랜잭션이 수행되는 기간(예를 들면, 사용된 CPU 및 SQL문의 양)을 알아야 하는 경우 이벤트 모니터를 사용하십시오. 그런 다음, 이벤트 모니터에 기록된 데이터를 읽기 위해 이벤트 분석기를 사용하십시오.

데이터베이스 연결 각각의 경우, 하나의 연결 이벤트 레코드가 생성됩니다. 해당 연결에서 수행되는 명령문의 경우, 명령문 레코드가 생성됩니다. 각 연결 이벤트 레코드는 이벤트 분석기의 연결 뷰 창에 있는 하나의 행으로 맵됩니다. 이 창에서는 모니터링되는 기간 동안 연결된 각각의 응용프로그램에 대한 정보를 표시하며, 여기에는 다음이 포함됩니다.

- 응용프로그램 이름
- 실행 ID
- 연결 시간
- 총 CPU 시간
- 잠금 대기 시간
- 총 정렬 시간
- 교착 상태
- 연결해제 시간
- 응용프로그램 ID

각각의 명령문 이벤트 레코드는 이벤트 분석기의 명령문 뷰 창에 있는 하나의 행으로 맵됩니다.

모니터 도구 사용

성능 모니터 및 이벤트 분석기를 사용하면 다음과 같은 이점이 제공됩니다.

- 광범위하고, 유연성 있는 데이터 컬렉션. 200개가 넘는 성능 변수가 지원되며, 여기에는 버퍼 풀 및 입출력, 잠금 및 교착 상태, 정렬, 통신, 에이전트 및 로그 정보가 포함됩니다. 데이터베이스 관리 프로그램, 데이터베이스, 테이블 공간, 테이블, 버퍼 풀, 연결, 트랜잭션 및 SQL문에 대한 데이터가 표시됩니다.

- 사용하기 쉽고 직관적인 뷰. 데이터는 논리 그룹으로 편리하게 구성된 읽기 쉬운 그래프 또는 텍스트를 사용하여 실시간으로 볼 수 있습니다. 세부사항 및 요약 뷰 모두가 제공되며, 좀더 자세한 정보에 액세스하는 기능을 가지고 있습니다.
- 견고한 경고 기능. 성능 측정을 위해, 임계값을 지정하여 예외 조건을 정의할 수 있습니다. 임계값은 성능 그래프에 있는 특정 영역에서 측정치 점을 찍어 성능 측정치가 임계값에 도달하거나 초과했음을 시각적으로 표시하는 데 사용됩니다. 임계값에 도달하면, 다음 조치 중 일부 또는 전부가 발생하도록 지정할 수 있습니다.
 - 경고 센터를 통해 알 수 있습니다.
 - 들을 수 있는 경보를 받습니다.
 - 프로그램이 수행됩니다.
 - 메시지가 표시됩니다.
 - 통지는 제공되지 않습니다.

33 페이지의 그림2에서는 모니터가 함께 동작하는 방법에 대해 설명합니다.

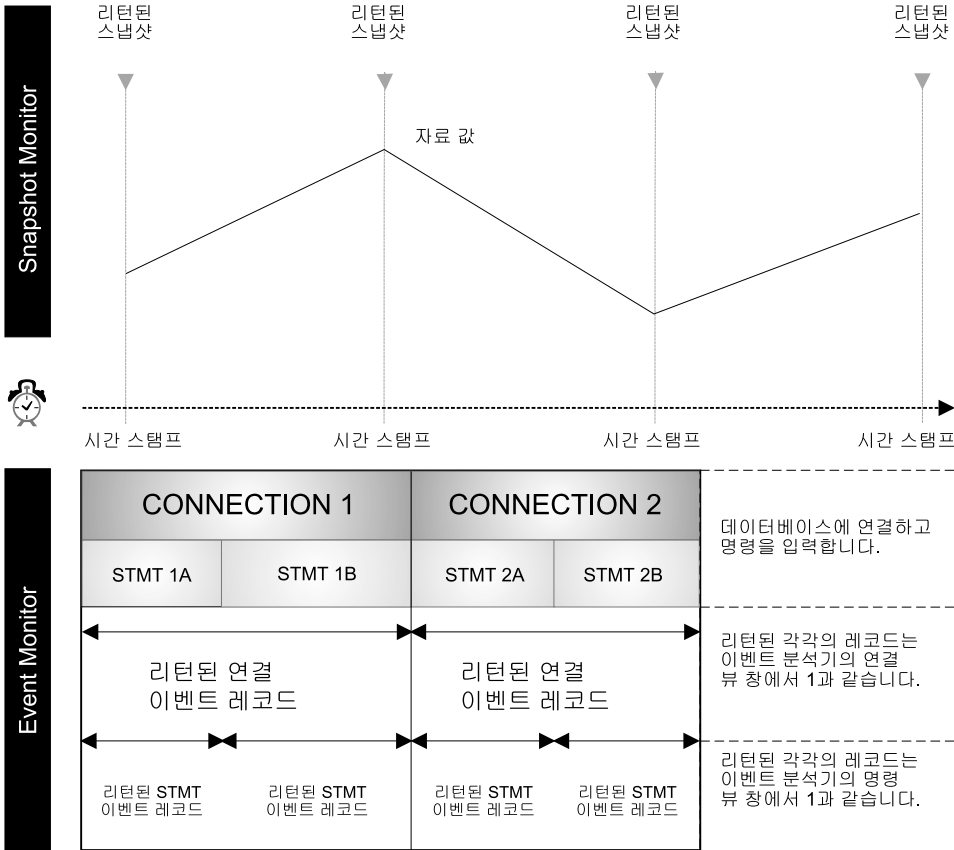


그림 2. 비교: 스냅샷 확보 및 이벤트 모니터(이벤트 모니터, 이벤트 분석기)

데이터베이스 모니터 및 조정을 위한 고려사항

데이터베이스를 모니터하고 조정하기 전에, 다음을 수행해야 합니다.

- 목표를 정의하십시오. 예를 들어, 데이터베이스 동시성이 특수 응용프로그램이 시작될 때 줄어드는지 점검할 수 있도록 특정 시점에 인스턴스 레벨에서 응용프로그램이 자원을 사용하는 방법을 이해하고자 할 수도 있습니다. 또는, 응용프로그램이 수행될 때 발생하는 인스턴스 레벨 이벤트, 예를 들면 특정 응용프로그램이 수행될 때 전체적인 성능이 불량한지 여부를 알고 싶을 수도 있습니다.
- 분석할 정보를 결정하십시오. 예를 들어, 하드웨어와 병목현상이 관련되어 있는지 알아보기 위해, 스냅샷을 사용하여 데이터베이스 연결 활동 또는 테이블 공

간, 버퍼 풀 및 입출력 활동을 모니터링하려는 경우가 있습니다. 병목현상이 환경에 관련된 것인지 알아보기 위해, 이벤트 분석기를 사용하여 다음의 경우를 모니터링합니다.

- 너무 많은 데이터베이스 타스크가 최대 시점에서 스케줄되는지의 여부
 - 사용자 연결 수가 많은지
 - 데이터베이스 파티션(하드웨어 로그 균형)이 제대로 최적화되지 않았는지
 - 서버가 둘 이상의 데이터베이스 서버에 대해 사용되고 있는지
- 몇몇 눈에 띄는 효과가 있습니다. 예를 들면,
- 조회/응답 시간이 늦습니다.
 - 스케줄된 타스크가 제때 완료되지 않습니다.
 - 응용프로그램이 시간종료되었습니다.
- DB2와 함께 사용할 수 있는 사전 정의된 모니터를 사용할 것인지 또는 자체 모니터를 작성할 것인지 여부를 결정하십시오.

다음 절에서는 스냅샷을 확보하는 방법과 경보 센터를 사용하여 성능과 관련된 문제점을 추적하는 방법에 대해 설명합니다.

특정 시점에서 성능 모니터

복잡한 데이터 콜렉션을 수행하고 데이터를 분석하여 잠재적인 문제점을 정확하게 지적하려면, 성능 모니터를 사용하여 시스템의 스냅샷을 확보하고 시간별 성능 데이터의 변화를 관찰하십시오.

도구를 사용하면, 다음을 수행할 수 있습니다.

- 성능 정보 그래프화
- 성능 변수 정의
- 성능 스냅샷의 캡처 빈도 설정
- 성능 계산 결과 보기
- 임계값 및 임계 조치 정의
- 경보 생성 및 저장
- 요약 정보 보기(예: 모든 데이터베이스)

다음 유형의 정보가 캡처됩니다.

- 수명이 긴 활동에 대한 정보(예: 응용프로그램이 완료하는 데 너무 오랜 시간이 소요될 때 데이터베이스 활동).
- 활동의 동시 레벨에 대한 정보를 추적하는 카운터(예: 데이터베이스에 대한 오픈 커서의 수).
- 데이터베이스 활동에 대한 정보 누적(예: 데이터베이스 인스턴스가 활성화되어 있는 동안 작성된 최대 연결 횟수 또는 특정 데이터베이스에 대해 실행된 SQL 문의 총 수).

사전 정의된 간격으로 스냅샷을 확보하면 데이터베이스 관리 프로그램과 응용프로그램의 활동에 대한 현재 상태를 나타내는 그림이 제공됩니다. 이 정보는 다음을 수행하는 데 사용될 수 있습니다.

- 성능 문제점 발견
- 성능 추세 분석
- 데이터베이스 관리 프로그램 및 데이터베이스 구성 매개변수 조정
- 데이터베이스 응용프로그램의 성능 분석

성능 정보는 다음 데이터베이스 오브젝트에 대해 사용할 수 있습니다.

- 인스턴스
- 데이터베이스
- 테이블
- 테이블 공간
- 데이터베이스 연결

각각에 대해 다양한 성능 변수를 모니터할 수 있습니다. 스냅샷 모니터 창의 도움말 메뉴에서 사용할 수 있는 성능 변수 참조 도움말에서는 모든 성능 변수에 대해 설명합니다. 이러한 변수는 범주로 구성됩니다. 기본적으로, 모든 성능변수들이 모니터되지만 범주는 관리 도구에 의해 켜지거나 꺼질 수 있습니다. 기본적으로 다음 범주가 설정됩니다.

- 인스턴스: 에이전트, 연결, 정렬
- 데이터베이스: 잠금과 교착 상태, 버퍼 풀 및 입출력, 연결, 정렬, SQL문 활동

- 테이블: 테이블
- 테이블 공간: 버퍼 풀 및 입출력
- 데이터베이스 연결: 버퍼 풀과 입출력, 잠금과 교착 상태, 정렬, SQL 커서, SQL 문 활동

제어 센터에서는 한 번에 하나의 데이터베이스 관리 프로그램의 인스턴스에서 스냅샷을 확보할 수 있습니다. 이것은 스냅샷 정보를 확보하기 위해 사용되는 API가 데이터베이스 관리 프로그램에 있는 모니터링되는 모든 데이터베이스 오브젝트에 대해 한번만 발행됨을 의미합니다. 이것으로 데이터베이스 관리 프로그램에 대한 오버헤드가 줄어듭니다.

스냅샷 생성 방법에 관한 자세한 정보는 온라인 도움말을 참조하십시오.

사전 정의된 모니터

DB2 성능 모니터에는 사전 정의된 모니터 세트가 들어 있으며, 이를 있는 그대로 사용하거나 요구사항에 맞도록 복사 및 수정할 수 있습니다. 광범위한 성능 계산 세트도 제공합니다. IBM이 제공하는 성능 모니터의 이름, 공식, 또는 텍스트 설명을 변경할 수 없지만, 임계값과 경고 조치는 변경할 수 있습니다. 사전 정의된 모니터를 사용하여 성능 모니터링에 대해 이해하고 사전 정의된 모니터를 복사하고 사본에서 성능 변수를 추가하거나 제거하여 사용자 고유의 모니터를 작성할 수 있습니다.

DB2와 함께 제공되는 사전 정의된 모니터는 다음과 같습니다.

- 모니터 기능. 이 모니터를 사용하여 시스템 용량에 대한 정보를 확보하십시오. 이 모니터는 정기적으로 점검되어 시스템의 전체적인 이용도를 볼 수 있습니다.
- 정렬. 이 모니터를 사용하여 정렬 힙(heap)과 정렬 힙(heap) 임계값 매개변수가 제대로 설정되어 있는지 확인하십시오. 이 모니터는 시스템을 처음으로 시작할 때, 활동이 가장 많은 시간에 또는 응용프로그램이 변경될 때 수행되어야 합니다.
- 잠금. 이 모니터를 사용하여 시스템에서 발생하는 잠금의 양과, 잠금 목록 매개변수가 제대로 설정되어 있는지 결정하십시오.

- 캐쉬. 이 모니터를 사용하여 캐쉬 이용도를 최적화하십시오. 최대기간 동안 이들 값을 모니터함으로써, 캐쉬 크기를 증가시켜야 하는지 여부를 결정할 수 있습니다.
- 버퍼 풀. 소형 테이블에서 이 모니터를 사용하여 자체 버퍼 풀이 필요한지 여부를 결정하십시오.
- 교착 상태. 이 모니터를 사용하여 응용프로그램이 교착 상태가 되는지 여부를 결정하십시오.
- 고속 통신 관리 프로그램. 이 모니터를 사용하여 노드 사이에서 정보를 교환하는 데 사용되는 메모리 비율을 알 수 있습니다.
- 프리페처. 이 모니터를 사용하여 시스템에 충분한 프리페처가 정의되었는지를 결정하십시오.
- 디스크 성능. 이 모니터를 사용하여 입출력을 관찰하십시오. 이 모니터에는 데이터베이스 및 테이블 공간 레벨에서 디스크 성능에 초점을 두는 성능 변수가 들어 있습니다.
- 전역 메모리. 이 모니터를 사용하여 응용프로그램 메모리 사용을 관찰하십시오.
- 장기 수행 메모리. 이 모니터를 사용하면 조회를 완료하는 데 오랜 시간이 소요되는 이유를 판별하는 데 도움이 됩니다.
- 게이트웨이 연결. 이 모니터를 사용하여 DB2 Connect 서버 연결을 관찰하십시오.

사전 정의된 모니터 사용법에 대한 예는 성능 모니터링을 위해 제공된 온라인 도움말을 참조하십시오.

사용 가능한 모니터 목록을 보려면, 제어 센터에서 오른쪽 마우스 버튼으로 시스템 폴더를 클릭하고, 팝업 메뉴에서 모니터 목록을 선택하십시오. 모니터 목록 창이 열립니다. 현재 연결된 JDBC 서버에 저장된 모니터를 나열합니다. 각 모니터에 대해, 모니터 이름, 설명, 상태, 기본 모니터가 있는지, 그리고 모니터 작성자를 알 수 있습니다. 모니터 상태는 지역 시스템에서 모니터의 상태를 나타내지만, JDBC 서버에서는 나타내지 않습니다. 레벨에 대한 기본값은 인스턴스, 데이터베이스, 테이블, 테이블 공간 또는 연결 레벨에서 기본 모니터를 나타냅니다. 사전 정의된 모니터의 경우, 작성자 컬럼에는 **NULLID**가 들어 있습니다. 창 오른쪽에는

사용자가 모니터에 대해 다양한 작업을 수행할 수 있도록 하는 누름 버튼이 들어 있습니다. 57 페이지의 『Java 애플릿으로서 제어 센터 수행』에서 JDBC 서버에 대한 자세한 내용을 참조하십시오.

오브젝트에 대한 기본 모니터로서 시작되는 모니터를 선택할 수 있습니다.

일단 성능 모니터를 시작하면, 도구 모음에 있는 경고 센터 버튼을 눌러 모니터링하는 오브젝트의 상태를 보고, 임계값에 도달하여 경고 상태에 있는 오브젝트도 볼 수 있습니다. 이들은 임계값을 초과한 동안에만 나타납니다.

모니터중인 오브젝트를 면밀히 관찰하려는 경우, 경고 센터를 열거나 요약 페이지에서 모니터 표시 창을 열어 빨간색 또는 노란색 항목을 찾을 수 있습니다. 또한, 새로운 경고 또는 정보가 추가된 경우 경고 센터가 자동으로 열릴 수 있도록 제어 센터 설정값을 수정할 수도 있습니다. 경고 센터에서, 모니터를 계속하는 동안 경보를 임시로 중지시킬 수도 있습니다.

경보 센터에 오브젝트가 나타날 때 필요한 조치

경보 또는 경고 상태에서(즉, 임계값을 초과한 경우) 모니터링되는 오브젝트를 표시하기 위해 자동으로 경고 센터가 열리도록 설정할 수 있습니다. 도구 설정값 창에서 이 기본값을 변경할 수 있습니다.

경보 센터에서 오브젝트를 볼 경우, 오브젝트를 오른쪽 마우스 버튼으로 클릭하고 성능 모니터 → 모니터 표시를 선택하여 해당 데이터베이스 오브젝트에 대한 성능 세부사항을 보십시오.

데이터 분석 방법에 대해서는 성능 모니터 창의 도움말 메뉴에서 사용할 수 있는 온라인 도움말을 참조하십시오.

기간 동안 이벤트 분석

이벤트 분석기는 또하나의 DB2 성능 도구입니다. 발생한 이벤트에 대한 정보를 진단하려는 경우 이 도구를 사용하십시오. 이벤트 분석기를 이벤트 모니터와 함께 사용하십시오. 예를 들어, 데이터베이스를 사용하고 있는 동안 이벤트 모니터를 사용하여 연결, 트랜잭션, 명령문 및 교착 상태와 같은 데이터베이스 활동을 추적할 수 있습니다. 이벤트 모니터는 응용프로그램이 데이터베이스에서 연결해제될 때 로

그된 누적 성능 데이터도 기록합니다. 이벤트 모니터가 이벤트 모니터 파일을 작성한 후, 이벤트 분석기를 사용하여 성능 정보를 볼 수 있습니다.

이벤트 모니터 도구를 사용하면, 다음을 수행할 수 있습니다.

- 이벤트 모니터를 작성하여 관심 있는 데이터베이스 이벤트 유형을 모니터하십시오.
- 이벤트 모니터를 활성화하여 이벤트 데이터 수집을 시작하십시오. 데이터는 파일에 저장됩니다.
- 이벤트 데이터를 수집하지 않도록 이벤트 모니터를 정지시키십시오.
- 이벤트 모니터가 생성한 추적 유형별 요약 정보를 보십시오.
- 더 이상 필요가 없으면 이벤트 모니터를 제거하십시오. 추적 파일을 제거하는 옵션도 제공됩니다.
- 데이터베이스와 연관된 이벤트 모니터 목록을 표시하십시오.
- 이벤트 모니터 정의를 보십시오.

이벤트 분석기를 사용하면 다음 이벤트 유형에 대해 이벤트 모니터가 생성한 데이터를 볼 수 있습니다.

- 데이터베이스 연결 활동(연결 및 연결해제 동안의 기간)
- 트랜잭션(작업 단위)
- SQL문 실행
- 교착 상태 활동 발견

이벤트 분석기

다음 이벤트 유형에 대해 하나의 이벤트 모니터를 작성한 다음, 이벤트 분석기를 사용하여 수집된 정보를 볼 수 있습니다. 그러나 다음에 대해 생성된 데이터를 보려면 db2evmon 실행파일을 사용하십시오(*Command Reference* 및 *시스템 모니터 안내* 및 참조서 참조).

- 교착 상태
- 데이터베이스 활동
- 테이블 공간 활동
- 테이블 활동

- 명령문 활동

이벤트 모니터 및 이벤트 분석기를 사용하여 이벤트 데이터를 분석하려면, 다음 단계를 수행하십시오. 이들은 연결 및 명령문 이벤트에 대한 이벤트 모니터를 작성하는 방법의 예만을 나타냅니다. 이벤트 모니터를 작성하려면, 다음을 수행하십시오.

1. 명령 센터의 명령행에서, **db2emcrt**를 입력하십시오. 이벤트 모니터 창이 열립니다.
2. **이벤트 모니터**를 클릭한 후 메뉴에서 작성을 선택하십시오. 이벤트 모니터 작성 창이 열립니다.
3. 필드에서, 작성하는 이벤트 모니터에 대한 이름을 지정하십시오. 이 새로운 이벤트 모니터의 이름은 기존 모니터와 같아서는 안됩니다. 이름에서 공백은 허용되지 않습니다.
4. Enterprise - Extended Edition 제품 전용에서, 이벤트 모니터 파일이 있는 노드를 노드 드롭다운 목록에서 선택하십시오.
5. Enterprise - Extended Edition 제품 전용에서, 이벤트 모니터에 대한 범위를 선택하십시오. 기본적으로, 범위는 전역입니다.
6. 하나 이상의 선택란을 선택하여 모니터링하려는 이벤트 유형을 나타내십시오. 교착 상태 이벤트 유형이 기본 선택임을 주의하십시오.
7. 이 모니터를 시작하려는 시기를 나타내십시오. 지금 시작이 기본 선택사항임을 주의하십시오.
8. 이러한 레벨에서 모니터링을 제어할 연결, 명령문 또는 트랜잭션에 대해 하나 이상의 조건을 정의하십시오.
9. 모니터가 이벤트 데이터 파일을 기록할 경로(디렉토리 이름)를 식별하십시오.
10. 옵션을 클릭하여 이벤트 모니터 파일 지정 옵션 창을 여십시오. 이러한 옵션은 모니터 출력이 처리되는 방법을 결정하므로, 이벤트 모니터의 성능에 영향을 줄 수 있습니다.
11. 확인을 클릭하여 모니터를 작성하거나, 취소를 클릭하여 모니터를 작성하지 않고 나가십시오.
12. 이벤트 모니터를 오른쪽 마우스 버튼으로 클릭한 후 팝업 메뉴에서 **이벤트 모니터 중지**를 선택하여 이벤트 모니터링을 중지시키십시오.

이렇게 하면 이벤트 모니터가 추적 파일을 기록하도록 강제됩니다. 모니터가 꺼지지 않으면, 버퍼가 가득차거나 모든 연결이 종료될 때에만 정보가 디스크로 기록됩니다. 이벤트 모니터 창에서, 작성한 이벤트 모니터를 오른쪽 마우스 버튼으로 클릭한 후 팝업 메뉴에서 **이벤트 모니터 파일 보기**를 선택하여 그 결과 발생한 이벤트 데이터를 볼 수 있습니다. 모니터 기간 보기 창이 열립니다.

이벤트 분석기에서 이벤트 데이터에 액세스하려면, 다음을 수행하십시오.

1. 명령 센터의 명령행에서, **db2eva**를 입력하여 이벤트 분석기를 시작하십시오. 이벤트 분석기 창이 열립니다.
2. 경로 필드에서 데이터 파일이 저장된 경로(디렉토리 이름)를 식별하십시오. 파일이 이동되지 않았으면, 이벤트 모니터가 작성될 때 지정된 경로가 됩니다. 파일을 지정한 뒤, 해당 디렉토리를 지정하십시오. ...을 클릭하여 기존 디렉토리를 표시할 수 있습니다.

주: 데이터 파일이 원격으로 저장되면, 이들을 볼 수 있도록 파일을 지역 머신으로 FTP해야 합니다. 파일 크기에 따라, 이 전송에서 다소 시간이 걸릴 수도 있습니다. 파일은 어느 지역 경로로도 전송될 수 있습니다. 작성될 때 사용된 것과 같은 경로를 선택하지 않아도 됩니다.

3. **확인**을 클릭하여 디렉토리에 들어 있는 데이터 파일에 액세스하거나, **취소**를 클릭하여 나가십시오. 모니터 기간 보기 창이 열립니다.
4. 모니터 기간을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **열기** → **연결**을 선택하십시오. 연결 뷰 창이 열립니다. 이 창에서는 이벤트 모니터 세션 동안 작성된 연결 목록을 보여줍니다(둘 이상의 연결이 나열될 수 있습니다. 관심 있는 연결이 목록에서 첫번째에 없을 수도 있습니다).
5. 연결을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **열기** → **명령문**을 선택하십시오. SQL문 뷰 창이 열립니다. 이 창은 선택된 연결에 대한 모든 명령문을 표시합니다. 정보 컬럼은 다음을 포함한 각각의 명령문에 대해 제공됩니다.
 - 조작
 - 패키지 이름
 - 작성자

- 시작 시간
- 경과 시간
- 총 CPU 시간
- 텍스트

이벤트 모니터 및 이벤트 분석기에 대한 온라인 도움말에서는 이벤트 모니터를 작성하고 그 결과 발생한 이벤트 데이터를 보는 상세한 내용을 설명합니다.

SQL문 분석

그래프로 설명된 SQL문에 대한 액세스 플랜을 보고 이 정보를 사용하여 보다 나은 성능을 위해 SQL 조회를 조정할 수 있습니다.

액세스 플랜 그래프에서는 다음에 대한 세부사항을 보여줍니다.

- 테이블(및 해당 연관 컬럼) 및 색인
- 연산자(예: 테이블 스캔, 정렬 및 조인)
- 테이블 공간 및 함수

버전 6 이전에는, Visual Explain이라는 도구를 사용하여 액세스 플랜을 보았습니다. 이제, 명령행에서 별도의 도구로서 Visual Explain을 호출하지 않지만, 제어 센터에 있는 여러 데이터베이스 오브젝트에서, 그리고 명령 센터에서 Visual Explain 함수를 호출할 수 있습니다. 이 절에서는 *Visual Explain* 함수 용어가 사용됩니다.

Visual Explain 함수를 사용하여 다음을 수행할 수 있습니다.

- 최적화시 사용되는 통계 뷰. 그런 다음, 이러한 통계를 현재 카탈로그 통계와 비교하여 패키지 리바인딩이 성능을 향상시킬 수 있을지 여부를 결정하는 데 도움을 줄 수 있습니다.
- 테이블 액세스에 색인이 사용되는지 여부 결정. 색인이 사용되고 있지 않으면, Visual Explain 함수는 색인화하여 이득을 얻는 컬럼을 결정하는 데 도움이 됩니다.
- 하나의 조회에 대한 액세스 플랜 그래프의 이전 및 이후 버전을 비교함으로써 다양한 조정 기법 수행의 효과를 보십시오.

- 해당 액세스 플랜에서 각 조작에 관한 정보(총 추정 비용 및 수신된 행 수 포함)를 얻으십시오.

조회 성능 향상

Visual Explain 함수를 사용하여 SQL문을 분석하고 조정을 돕습니다. 설명된 SQL문에 대한 액세스 플랜의 그래픽 뷰를 나타냅니다. 테이블과 색인, 그리고 이에 대한 각각의 조작은 노드로서 표현되고, 데이터 흐름은 노드간의 링크로서 표현됩니다. 이 그래프에서 사용할 수 있는 정보를 사용하여 보다 나은 성능을 위한 SQL 조회 조정 방법을 알 수 있습니다.

Visual Explain 함수는 SQL문이 컴파일되는 방법에 대한 정보를 캡처합니다. 이 정보를 사용하면 SQL문의 플랜 및 잠재적인 실행 성능을 알 수 있습니다.

이 정보는 다음을 수행하는 데 도움이 될 수 있습니다.

- 응용프로그램 설계
- 데이터베이스 설계
- 두 테이블이 조인되는 방법. 즉, 방법, 순서, 소트 발생 및 정렬 유형 이해
- SQL문의 성능을 향상시키는 방법 결정(예를 들면, 새로운 색인을 작성하여)
- 최적화시 사용되는 통계 뷰. 그런 다음, 이러한 통계를 현재 카탈로그 통계와 비교하면 패키지 리바인드가 성능을 향상시키는지 여부를 결정하는 데 도움이 될 수 있습니다. 또한, 수집된 통계가 성능을 향상시킬 수 있는지 여부를 결정하는 데에도 도움이 됩니다.
- 테이블 액세스에 색인이 사용되는지 여부 결정. 색인이 사용되지 않는 경우, Visual Explain 함수는 조회 성능을 향상시키기 위해 색인에 포함될 수 있는 컬럼을 결정하는 데 도움이 될 수 있습니다.
- 조회를 위한 액세스 플랜 그래프 버전의 이전과 이후를 비교하여 성능을 향상시키기 위한 다양한 조정 기술의 수행 효과 보기
- 액세스 플랜에서 각 조작에 대한 정보 확보, 총 추정 비용과 검색된 행 수 포함

Explain된 SQL문의 액세스 플랜을 이해하기 위해 Visual Explain을 사용한 후에, 색인이 해당 조회의 성능을 향상시킬 수 있는지 판별할 수 있습니다. 조회의

권장 색인을 받으려면 색인 마법사를 사용해야 합니다. 또는 RECOMMENDED_INDEXES EXPLAIN-mode를 사용할 수 있습니다. 색인 마법사에 대한 자세한 내용은, 제어 센터로 가서 정보 센터로 들어 가십시오.

관리 안내서: 성능에서 RECOMMENDED_INDEXES EXPLAIN-mode에 대한 자세한 내용을 참조하십시오.

단순 동적 SQL문 분석

이 절에서는 동적 SQL 조회를 분석하는 방법에 대한 간단한 예를 제공합니다.

1. 제어 센터에서, SAMPLE 데이터베이스를 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **SQL 설명**을 선택하십시오. SQL문 설명 창이 열립니다.
2. **SQL 텍스트 필드**에서, 다음 SQL문을 입력하십시오.

```
select * from staff order by name
```
3. **확인**을 클릭하십시오. 액세스 플랜 그래프 창이 열립니다. 그래프는 조회 결과를 제공하기 위해 가장 효과적인 것으로 최적화 알고리즘이 선택한 경로를 나타냅니다.
4. **선택적: 노드**를 두 번 클릭하십시오(예: RETURN 연산자 노드). 연산자 세부 사항 창이 열리며, 해당 연산자에 대한 세부사항을 표시합니다.

설명된 SQL문이 자동으로 저장됩니다. 이를 나중에 보려면, 다음을 수행하십시오.

1. 제어 센터에서, SAMPLE 데이터베이스를 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **Explain된 명령문 실행기록 표시**를 선택하십시오. 설명된 명령문 실행기록 창이 열립니다.
2. 원하는 항목을 찾으십시오. **SQL 텍스트 컬럼**을 찾아 이전에 설명한 SQL문을 볼 수 있습니다.
3. 항목을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **액세스 플랜 표시**를 선택하십시오. 액세스 플랜 그래프 창이 열립니다.

Visual Explain에 대한 도움말(도움말 메뉴에서 사용할 수 있음)에서는 SQL문의 성능을 향상시키기 위해 액세스 플랜 그래프 창을 해석하는 방법에 대한 세부사항을 제공합니다. 온라인 도움말에는 Visual Explain 사용 방법을 배우는 데 도움이 되는 상세한 예도 들어 있습니다.

원격 데이터베이스 관리

다음 절에서는 다음을 수행하는 방법에 대해 보여줍니다.

- 원격 시스템 추가
- 해당 시스템에 대해 작업하려는 인스턴스 추가
- 해당 인스턴스에서 작업하려는 데이터베이스 추가

DB2는 먼저 노드 디렉토리(데이터베이스 클라이언트가 연결할 수 있는 모든 서버에 대한 항목과, 연결에 사용된 통신 프로토콜이 들어 있음)를 점검하여, 원격 시스템이 이미 알려져 있는지를 확인합니다. 원격 시스템을 모르면, 시스템, 인스턴스 또는 원격 시스템상의 데이터베이스와 함께, 사용자는 자신을 원격 시스템에 대한 클라이언트로서 설정해야 합니다.

DB2를 설치한 후, 클라이언트 구성 지원 프로그램을 사용하여 시스템, 인스턴스 및 데이터베이스를 네트워크에서 검색하고 이들간의 통신을 구성할 수 있습니다. 그런 다음, 원격 시스템을 카탈로그화하여 이를 추가합니다. 이것으로 해당 인스턴스 및 데이터베이스가 알려질 수 있도록 노드 디렉토리에 시스템에 대한 항목이 작성됩니다. 그런 다음, 인스턴스 및 데이터베이스를 카탈로그화하여 노드 디렉토리 및 데이터베이스 디렉토리에 항목을 작성함으로써 시스템에 대한 인스턴스와 데이터베이스를 추가해야 합니다. 이것으로 노드 디렉토리와 데이터베이스 디렉토리에서 각각에 대한 항목이 작성됩니다. 구성이 완료되면, 원격 시스템은 사용자가 이에 대해 작업할 수 있도록 제어 센터에 표시됩니다.

원격 시스템을 추가하려면, 다음을 수행하십시오.

1. 제어 센터에서 시스템 오브젝트를 오른쪽 마우스 버튼으로 클릭한 후 추가를 선택하십시오. 시스템 추가 창이 열립니다.
2. 시스템 이름 필드에 시스템 이름을 입력하십시오.

인스턴스에 대한 **Discover** 구성 매개변수가 **search**로 설정되고 **discover comm** 구성 매개변수가 공백이 아니면, 새로 고침을 선택하여 원격 시스템 목록을 확보할 수 있습니다. 그런 다음, 시스템 이름 필드 아래에 있는 목록에서 시스템 중 하나를 선택할 수 있습니다.

3. 원격 인스턴스 이름 필드에 원격 인스턴스 이름을 입력하십시오.
4. 운영 체제 목록에서 원격 시스템에 대한 운영 체제 유형을 선택하십시오.

5. 원격 위치와의 통신에 사용되는 프로토콜을 선택하십시오. 지역 시스템의 경우, 지역이 자동으로 선택되고 유일하게 유효한 프로토콜입니다. 원격 시스템의 경우, 가능한 프로토콜은 다음과 같습니다.

- APPC
- IPX/SPX
- NetBIOS
- TCP/IP
- Named Pipe(Windows NT 및 Windows 9x 운영 체제 전용)

목록 상자에 나타나도록 컴퓨터에 현재 설정된 프로토콜만 해당

6. 적합한 프로토콜 매개변수를 입력하십시오.
7. 시스템과 연관될 주석을 입력하십시오.
8. 적용을 클릭하여 노드 디렉토리로 시스템을 추가하십시오.

그런 다음, 해당 시스템에 대해 작업하려는 인스턴스를 추가하십시오.

1. 제어 센터에서, 방금 추가한 시스템에 속하는 인스턴스 오브젝트를 오른쪽 마우스 버튼으로 클릭하십시오.
2. 추가를 선택하십시오. 인스턴스 추가 창이 열립니다.
3. 필드에 필요한 값을 입력하십시오.
4. 새로 고침 누름 버튼을 클릭하면 기존 인스턴스 목록이 표시됩니다.
5. 작업하려는 인스턴스를 선택하십시오.
6. 적용 누름 버튼을 클릭한 다음, 닫기 누름 버튼을 클릭하십시오.

마지막으로, 해당 인스턴스에서 작업하려는 데이터베이스를 추가하십시오.

1. 제어 센터에서 데이터베이스 오브젝트를 오른쪽 마우스 버튼으로 클릭하십시오.
2. 추가를 클릭하십시오. 데이터베이스 추가 창이 열립니다.
3. 데이터베이스 이름, 통신 프로토콜 유형, 그리고 선택적으로 별명을 입력하십시오. 이 경우, 별명은 데이터베이스를 식별하는 데 사용되는 대체 이름입니다.
4. 새로 고침 누름 버튼을 클릭하여 해당 인스턴스에 대한 기존 데이터베이스 목록을 표시하십시오.
5. 데이터베이스를 선택하십시오.

6. 적용 누름 버튼을 클릭한 다음, 닫기 누름 버튼을 클릭하십시오.

사용자 관리

데이터베이스 관리자로서, 데이터에 대해 사용자가 갖는 액세스 유형을 제어하거나, 이들의 데이터 뷰를 제한해야 하는 경우도 있습니다. 다음 정보는 관리 도구를 사용하여 데이터베이스 오브젝트에 대한 데이터베이스 권한 및 특권을 관리하는 방법에 대해 알려줍니다.

데이터베이스 권한에는 데이터베이스에 대한 조치가 전체적으로 포함됩니다. 데이터베이스가 작성되면, 일부 권한이 데이터베이스에 액세스하는 사용자에게 자동으로 권한 부여됩니다. 예를 들어 CONNECT, CREATETAB, BINDADD 및 IMPLICIT_SCHEMA 권한은 모든 사용자에게 권한 부여됩니다. 데이터베이스 특권에는 데이터베이스 내의 특정 오브젝트에 대한 조치가 포함됩니다. 데이터베이스가 작성되면, 일부 특권이 데이터베이스에 액세스하는 사용자에게 자동으로 권한 부여됩니다. 예를 들어 SELECT 특권은 카탈로그 뷰에 권한 부여되고, 각각의 바운드 유틸리티에 대한 EXECUTE 및 BIND 특권은 모든 사용자에게 권한 부여됩니다.

이와 함께, 특권 및 권한은 인스턴스와 해당 데이터베이스 오브젝트에 대한 액세스를 제어하도록 동작합니다. 사용자는 적합한 권한이 부여된, 즉 필수 특권 또는 권한을 가진 오브젝트에만 액세스할 수 있습니다.

권한 및 특권의 권한 부여와 권한 취소

DB2 관리 도구를 사용하여 데이터베이스, 테이블 공간, 테이블, 뷰 및 스키마에 대한 특권을 사용자와 그룹에 권한 부여하고 권한 취소할 수 있습니다.

1. 제어 센터에서, 특권을 권한 부여하거나 권한 취소하려는 데이터베이스, 테이블, 뷰, 스키마 또는 색인을 오른쪽 마우스 버튼으로 클릭하십시오. 팝업 메뉴에서 권한 또는 특권을 선택하십시오. 권한 창이나 특권 창이 열립니다.
2. 사용자 페이지를 선택하여 사용자 권한 또는 특권에 대해 작업하거나, 그룹 페이지를 선택하여 그룹 권한 또는 특권에 대해 작업하십시오.
3. 하나 이상의 사용자 또는 그룹을 선택하십시오. 사용자 또는 그룹을 목록에 추가하려면, 사용자 추가 또는 그룹 추가 누름 버튼을 클릭하십시오.

4. 창 맨 아래에서 각 권한 또는 특권에 대해 예, 아니오 또는 권한 부여를 선택하십시오. 권한 부여는 유효한 옵션이 있는 오브젝트에 대해서만 표시됩니다.
5. 완료하면, 적용 누름 버튼을 클릭하십시오.

특별한 사용자에게 권한이 부여되는 오브젝트를 검토하거나 변경하려는 경우, 사용자를 선택한 후 오른쪽 마우스 버튼을 클릭하여 오브젝트에 대한 권한을 추가 또는 변경하거나, 권한을 제거하십시오.

데이터 이동

DB2에서는 기존 소스에서 테이블로 데이터를 이동시키는 데 도움이 되도록 가져오기 및 로드 유틸리티를 제공합니다. 이 절에서 제공하는 정보는 데이터 이동에 관한 간단한 개요입니다. 데이터 이동에 관한 자세한 정보는 [데이터 이동 유틸리티 안내 및 참조서 매뉴얼](#)을 참조하십시오.

가져오기 유틸리티는 입력 파일에서 데이터를 선택하고 이를 테이블 또는 뷰로 삽입합니다. 이 경우, 입력 파일에는 기존 데이터 소스에서 추출한 데이터, 예를 들면, Lotus 1-2-3 파일 또는 ASCII 파일이 들어 있습니다. 또한, 가져오기 유틸리티를 사용하면, 내보내기 유틸리티를 사용하여 저장한 테이블 또는 뷰를 다시 작성할 수 있습니다. 다음 정보는 데이터를 가져오는 방법을 알려줍니다.

일단 지원되는 형식으로 입력 파일을 사용할 수 있으면, 가져오기 노트북을 사용하여 파일에서 기존 테이블로 데이터를 삽입하십시오. 이 테이블에 이미 데이터가 들어 있으면, 기존 데이터를 파일에 있는 데이터로 대체하거나 첨부할 수 있습니다.

또한, 가져오기 노트북을 사용하여 입력 파일이 들어있는 새로운 테이블을 작성하거나 선택된 테이블에 있는 기존 행을 삭제하고, 입력 파일의 데이터를 사용하여 이를 다시 포함시킬 수 있습니다.

기존 테이블로 파일을 가져오려면, 다음을 수행하십시오.

1. 가져오기 노트북의 파일 페이지를 여십시오.
2. 선택적. 가져오기 노트북을 지정하십시오.
3. 선택적. 대형 오브젝트(LOB)를 검색하십시오.

4. 선택적. 컬럼 가져오기 옵션을 지정하십시오.
5. 확인을 클릭하십시오.

가져오기 노트북의 파일 페이지를 열려면, 다음을 수행하십시오.

1. 제어 센터에서, 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 클릭하십시오. 기존 테이블은 내용 패널에 표시됩니다.
3. 내용 패널에 있는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 가져오기를 선택하십시오. 가져오기 노트북은 파일 페이지를 표시하며 열립니다.

파일 옵션을 지정하려면, 다음을 수행하십시오.

1. 파일 페이지의 가져올 파일 필드에서, 가져오려는 데이터가 들어 있는 파일 이름을 입력하십시오.
2. 다음 중 하나를 선택하여 가져올 파일 유형을 지정하십시오.
 - 컬럼 식별자가 없는 ASCII 형식(ASC)

컬럼 식별자가 없는 ASCII 데이터는 컬럼별로 정렬되는 데이터입니다.
 - 컬럼 식별자가 있는 ASCII 형식(DEL)

컬럼 식별자가 있는 ASCII 데이터는 보통 컬럼 값이 콤마와 같이 사용자가 정의한 분리 문자로 구분되는 데이터를 저장하는 방식에 사용됩니다.
 - 워크시트 형식(WSF)
 - 통합 교환 형식(IXF)

PC/IXF는 데이터베이스 테이블 또는 뷰에 대한 구조적인 설명입니다. PC/IXF 형식으로 내보내기된 데이터는 다른 Universal Database 제품 데이터베이스로 가져오기되거나 로드될 수 있습니다.

지원되는 특정 제품 및 릴리스에 대해서는 온라인 도움말을 참조하십시오.
3. 선택적: 해당 옵션 누름 버튼을 클릭하여 파일 유형 수정자를 지정하십시오. 해당 형식의 옵션 창이 열립니다.
4. 가져오기 모드를 선택하십시오. 사용 가능한 반입 모드는 선택한 파일 유형에 따라 달라집니다.
5. 선택적: 레코드 요약 필드에서, 변경사항을 요약하기 전에 가져올 레코드 수를 입력하십시오.

6. 선택적: 재시작 필드에서, 가져오기 조치를 시작하기 전에 통과할 파일에 있는 레코드 수를 입력하십시오.
7. 선택적: 복합 텍스트 필드에, 실행될 SQL문의 수를 지정하는 숫자를 입력하십시오(실행 가능한 블록에서).
8. 선택적: 십진 데이터에 내재된 소수점 삽입(IMPLIEDECIMALPOINT) 선택란을 선택하십시오.
9. 메시지 파일 필드에, 가져오는 동안 발생한 경고 및 오류 메시지가 들어갈 파일 이름을 입력하십시오.

별개의 파일에서 대형 오브젝트(LOB)를 검색하려면, 가져오기 검색의 대형 오브젝트 페이지를 사용하여 LOB 파일이 저장된 경로에서 대형 오브젝트(LOB)를 검색하십시오.

1. 대형 오브젝트(LOB)를 별도의 파일에서 검색(LOBINFIL) 선택란을 클릭하여 대형 오브젝트(LOB) 페이지에서 옵션을 사용 가능하게 하십시오.
2. 추가 누름 버튼을 클릭하여 LOB 경로 목록 상자에 별도의 LOB 파일 위치를 지정하십시오. 이러한 경로에서는 입력 파일의 LOB 컬럼에서 지정된 LOB 파일을 검색합니다(LOB 경로 목록 상자에 표시되는 순서대로).
3. 확인을 눌러 다른 노트북 페이지의 기본값을 수용하고, 가져오기 프로세스를 시작하십시오.

컬럼 가져오기 옵션을 지정하십시오. 가져오기 노트북의 컬럼 페이지를 사용하여 컬럼 가져오기 옵션을 지정하십시오.

1. 컬럼 포함 방법 상자에 있는 단일선택 버튼 중 하나를 클릭하여 데이터 파일 컬럼을 테이블로 가져오는 데 사용될 컬럼 방법을 지정하십시오. 사용 가능한 방법은 파일 유형과, 파일 페이지에서 선택한 모드에 따라 달라집니다.
2. 선택적: 변경 누름 버튼을 클릭하여 가져오기 파일 컬럼 속성을 지정하거나 변경하십시오.

이 옵션은 기본값(방법 D) 단일선택 버튼을 선택할 경우에는 사용할 수 없습니다.

저장영역 관리

데이터베이스 관리자로서, 테이블과 색인의 크기를 추정하고, 테이블 공간이 가득 차면 기존 테이블 공간에 보다 많은 공간을 추가하여 테이블 공간에서 사용할 수 있는 공간량을 점검해야 합니다.

이 절에서는 다음 방법에 대해 설명합니다.

- 테이블 및 색인 크기 계산
- 테이블 공간에서 사용할 수 있는 공간량 점검
- 가득 찬 채로 시작될 때 기존 테이블 공간에 공간 추가

테이블 및 색인 크기 계산

크기 계산 대화 상자를 열어 새로운 또는 기존 테이블에 또는 색인에 필요한 저장영역 공간의 양을 추정할 수 있습니다. 개별 테이블 및 색인을 선택하고 이를 오른쪽 마우스 버튼으로 클릭하거나, 테이블 작성 및 색인 작성 창에서 크기 계산을 선택하여 이 대화 상자를 엽니다. 크기는 특별한 테이블 및 종속 색인 정의에서 추정될 수 있습니다. 추정치는 테이블에 지정된 수의 행이 있을 때 사용되는 계획된 저장영역 공간의 양입니다. 가변 길이 필드의 최소 및 최대 크기를 근거로 최소 및 최대 공간도 추정됩니다. 크기 계산 대화 상자는 테이블 또는 색인에서 열릴 때, 테이블 스펙으로 미리 채워지며, 테이블 및 모든 종속 색인에 관련된 숫자가 들어 있습니다. 새로 고침 누름 버튼을 클릭하면, 추정 크기, 최소 크기 및 최대 크기가 새 행의 총 수 및 새 행의 평균 길이 필드에 입력한 숫자를 근거로 갱신됩니다.

테이블 또는 색인 크기를 측정하면, 다음을 수행하려는 경우 유용합니다.

- 새로운 테이블을 작성하고 테이블 공간의 크기를 알고자 합니다.
- 기존 테이블의 크기 계산치를 근거로 새로운 테이블을 작성합니다.
- 시스템에 저장영역 공간이 부족하므로 다른 테이블 및 색인 오브젝트가 사용하는 공간량을 알아봅니다.
- 데이터를 로드하기 전에 계획된 테이블 크기를 추정합니다.

주: Enterprise-Extended Edition 제품에 대해 크기 계산을 사용하는 경우, 크기 계산치는 데이터베이스 파티션이 아닌 테이블에 있는 데이터의 논리 크기를 근거로 합니다.

테이블에 대한 통계를 갱신하지 않았으면, 통계 수행 누름 버튼을 클릭하여 선택된 테이블에 대한 통계를 갱신할 수 있습니다. 색인을 선택한 다음 통계 수행 버튼을 누르면, 관련된 테이블에 대해 통계가 수행됩니다.

테이블 크기를 추정하려면, 다음을 수행하십시오.

- 크기 계산 창을 여십시오.
- 새 행의 총 수에서 다른 값을 선택하거나 기본값을 사용하십시오.
- 새로 고침을 클릭하여 새로운 값에 대한 크기 계산치를 보십시오.
- 새 행의 평균 길이에서 다른 값을 선택하거나 기본값을 사용하십시오.
- 새로 고침을 클릭하여 새로운 값에 대한 크기 계산치를 보십시오.

테이블 공간에서 사용 가능한 공간 점검

DMS 테이블 공간에서 사용 가능한 공간량을 점검하려면, 다음을 수행하십시오.

1. 제어 센터에서 테이블 공간을 두 번 클릭하십시오. 모든 테이블 공간 목록이 내용 패널에 나타납니다.
2. 제목이 할당된 크기, 사용된 크기 및 사용 비율인 컬럼으로 화면이동하여 테이블 공간에서 사용할 수 있는 공간량과 관련된 세부사항을 보십시오. 공간은 1 페이지가 4KB인 페이지 단위로 측정됩니다.

내용 패널 맨 아래에 있는 컬럼 사용자 정의 아이콘을 사용하여 컬럼 순서와 표시되는 컬럼을 사용자 정의할 수 있습니다.

SMS 테이블 공간에서 사용할 수 있는 공간량을 점검하려면, 운영 체제가 제공하는 기능을 사용하여 공간 이용도를 모니터링하고, 테이블 공간에 대한 디렉토리에서 사용 가능한 공간이 부족하지 않도록 합니다.

테이블 공간으로 공간 추가

DMS 테이블 공간에 대한 용량은 테이블 공간에 할당된 컨테이너의 총 크기입니다. DMS 테이블 공간이 용량에 도달하면(테이블 공간의 이용도에 따라, 90%가

가능한 임계값임), 좀더 많은 공간을 추가해야 합니다. 데이터베이스 관리 프로그램은 사용 가능한 모든 컨테이너 사이의 DMS 테이블 공간에 있는 테이블을 자동으로 재조정합니다. 재조정하는 동안, 테이블 공간에 있는 데이터는 계속 액세스할 수 있습니다.

용량에 도달한 DMS 테이블 공간의 경우, 다른 컨테이너를 추가할 수 있습니다.

1. 제어 센터에서, 컨테이너를 추가하려는 내용 패널에 있는 테이블 공간을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **변경**을 선택하십시오. 테이블 공간 변경 창이 열립니다.
2. **추가**를 클릭하십시오. 컨테이너 추가 창이 열립니다.
3. **파일** 또는 **원시 장치** 단일선택 버튼을 선택한 후, 필드를 채우십시오. 도움은 온라인 도움말을 참조하십시오.
4. **확인**을 클릭하십시오.

일반적으로, SMS 용량은 파일 시스템에서 사용 가능한 공간과, 운영 체제가 지원하는 파일의 최대 크기에 따라 다르므로 SMS 테이블 공간의 크기를 쉽게 확장할 수는 없습니다. 그러나 운영 체제에 따라 운영 체제 기능을 사용하여 파일 시스템 크기를 증가시킬 수도 있습니다. UNIX 기반 시스템에 있는 SMS 테이블 공간의 경우, 적합한 UNIX 기반 시스템 명령을 사용하여 테이블 공간의 크기를 증가시킬 수 있습니다. 수행 중인 UNIX 기반 시스템에 대한 문서를 참조하십시오. SMS 테이블 공간이 들어 있는 파일 시스템에 DB2 이외의 파일이 들어 있는 경우, 이들 파일을 다른 파일 시스템으로 이동시켜 DB2가 사용할 수 있도록 파일 시스템에서 사용 가능한 공간을 확보할 수 있습니다. 또한, 백업된 것보다 많은 수의 컨테이너로 테이블 공간 복원이 포함된 경로 재지정 복원을 수행할 수도 있습니다. 데이터베이스 복원 노트북에서 경로 재지정된 복원을 수행할 수 있습니다. 복원하려는 데이터베이스의 팝업 메뉴에서 **복원** → **데이터베이스**를 선택하십시오.

문제점 해결

DB2에서는 DB2 서버 및 클라이언트를 위한 기술 지원 담당자용으로 문제점 진단 매뉴얼을 제공합니다. 이것은 다음을 수행하는 데 도움이 됩니다.

- 간단한 방식으로 문제점 또는 오류 식별
- 증후를 근거로 문제점 해결

- 사용 가능한 진단 도구 사용
- 일상적인 DB2 조작을 위해 문제점 해결 방식 개발

문제점 해결 안내서에서는 다음과 같은 기본적인 문제점 해결 주제에 대해 다룹니다.

- 문제점 해결 실제 연습
- 서버에서의 문제점 해결
- 클라이언트에서의 문제점 해결
- 호스트 통신의 문제점 해결
- 문제점 해결 응용프로그램
- 문제점 해결 및 문제점 판별

문제점 해결 안내서에서는 다음과 같은 고급 문제점 해결 주제에 대해 다룹니다.

- DB2 프로세스 모델
- 로그된 정보 사용
- 추적 사용
- UNIX 기반, OS/2, Microsoft 및 Windows 운영 체제용 진단 도구

최신 소식과 기술 문서는 <http://www.software.ibm.com/data/db2/library/> 주소의 월드 와이드 웹에서 사용할 수 있습니다.

IBM 문의 방법에 대해서는 이 책 끝의 마지막 절을 참조하십시오.

데이터 복제

복제는 소스 서버에서 데이터베이스 로그에 저장된 변경사항을 취하여 이를 목표 서버에 적용하는 프로세스입니다. 복제를 사용하여 기업 전체 데이터에 대한 복사 조작을 정의, 동기화, 자동화 및 관리할 수 있습니다. 호스트 시스템에서 목표 사이트로 데이터를 자동 전달할 수도 있습니다. 예를 들어, 데이터와 응용프로그램을 지점 사무실, 소매 아울렛, 심지어 영업 담당자의 노트북에까지 복사할 수 있습니다.

복제에 있는 두 개의 동작 구성요소는 Capture와 Apply입니다. Capture 구성요소는 데이터베이스 로그를 읽음으로써 복제를 위해 정의된 소스 테이블에서 데이터의 변경사항을 캡처합니다. Apply 구성요소는 이전에 캡처되어 변경 데이터 테이블에 저장된 변경된 데이터를 읽고, 이를 목표 테이블에 적용합니다.

제어 센터를 사용하면, 복제 소스로서 정의 및 복사 작업 내역 정의 조치에 필요한 설정을 수행할 수 있습니다. 복제 구성요소 Capture와 Apply는 DB2 관리 도구 외부에서 수행됩니다.

복제 관리자는 제어 센터에서 다음 조치를 수행할 수 있습니다.

- 복제 소스 정의
- 복제 복사 작업 내역 정의
- 적용 프로세스 동안 데이터를 향상시키도록 SQL 지정

데이터 복제에 대한 고급 단계들은 다음과 같습니다. 복제 안내 및 참조서에서 자세한 내용을 참조하십시오.

1. 복제 시나리오 설계(소스 및 목표 테이블 맵)
2. 복제 소스 정의(캡처 조치와 관련됨)

복제 소스를 정의하려면, 다음을 수행하십시오.

1. 캡처할 소스 컬럼을 지정하십시오.
2. 복제 옵션을 선택하십시오.
3. 복제 복사 작업 내역 정의를 정의합니다(Apply 조치와 관련됨).
4. 소스 테이블을 데이터 캡처 변경 옵션으로 변경하십시오.
5. Capture를 시작하여 데이터 변경사항을 읽고 저장하십시오.
6. Apply를 시작하여 목표 테이블로 변경사항을 복제하십시오.

복제 복사 작업 내역 정의를 정의하려면, 다음을 수행하십시오.

1. 복사 작업 내역 정의 세트의 이름을 지정하십시오.
2. 데이터베이스 및 목표 테이블을 지정하십시오.
3. 목표 컬럼을 지정하십시오.
4. 행 선택사항을 지정하십시오.

5. 런타임 처리를 위해 SQL을 지정하십시오.
6. 복사 작업 내역 시간을 설정하십시오.

LDAP(Lightweight Directory Access Protocol) 사용

클라이언트 구성 지원 프로그램(CCA)을 사용하여 LDAP 서버에서 항목을 추가하고 삭제할 수 있습니다. LDAP 서버에 등록된 모든데이터베이스 인스턴스는 자동으로 클라이언트에 카탈로그화(캐쉬)됩니다. 이들은 경로탐색 트리에서 일반 노드로서 제어 센터에 표시됩니다. 이러한 데이터베이스는 사용자가 머신에 등록한 다른 데이터베이스와 같은 방식으로 관리될 수 있습니다(이 릴리스에서 ADD DATABASE 옵션을 아직 사용할 수 없는 경우는 예외입니다).

LDAP 데이터베이스를 관리하기 위해, 데이터베이스를 선택하고 오른쪽 마우스 버튼을 클릭하십시오. 팝업 창에서 수행할 수 있는 함수를 나열합니다. LDAP에 대해서는 559 페이지의 『부록J. LDAP(Lightweight Directory Access Protocol) 등록부 서비스』에서 자세한 내용을 참조하십시오.

Java 제어 센터 사용

제어 센터를 Java 응용프로그램으로서 또는 웹 서버를 통해 Java 애플릿으로서 수행할 수 있습니다. 어느 경우에도, 제어 센터를 수행하려면 지원되는 JVM(Java Virtual Machine)이 머신에 설치되어야 합니다. 제어 센터를 Java 응용프로그램으로서 런타임하려면, 정확한 JRE(Java Runtime Environment)가 설치되어야 합니다. JVM(Java Virtual Machine)은 응용프로그램을 런타임하기 위해서는 JRE(Java Runtime ntvrontent)가, 애플릿을 수행하기 위해서는 Java 가능 브라우저가 될 수 있습니다.

Java 응용프로그램은 머신에서 다른 응용프로그램처럼 수행되며, 정확한 JRE가 설치되도록 합니다. Java 애플릿은 Java 가능 브라우저 내에서 수행되는 프로그램입니다. 제어 센터 애플릿 코드는 원격 클라이언트에 상주하고, 웹 서버를 통해 클라이언트의 브라우저에 제공됩니다. 제어 센터를 Java 애플릿으로서 수행하는 경우, Windows 32비트 또는 OS/2 운영 체제에서 수행중인 지원되는 Java 가능 브라우저를 사용해야 합니다. UNIX 운영 체제에 대해 지원되는 브라우저는 없습니다.

제어 센터 JDBC 애플릿 서버는 애플릿 서버가 있는 머신에 대해 관리자 권한을 가진 사용자 계정으로 시작되어야 합니다. 제어 센터 JDBC 애플릿 서버를 시동 시 자동으로 시작하도록 설정할 수 있습니다.

Java 애플릿으로서 제어 센터 수행

제어 센터를 Java 애플릿으로서 수행하려면, 제어 센터 애플릿 코드와 제어 센터 JDBC 애플릿 서버가 들어 있는 머신에서 웹 서버를 설정해야 합니다. 웹 서버는 sqllib 디렉토리로의 액세스를 허용해야 합니다. 가상 디렉토리를 사용하도록 선택하면, 홈 디렉토리에 대해 이 디렉토리를 대체하십시오. 예를 들어, 가상 디렉토리의 이름을 temp로 지정하면, sqllib/temp를 사용해야 합니다. DB2는 OS/2용 FAT 드라이브에서 제어 센터 설치를 지원하지 않습니다. OS/2 FAT 드라이브가 Java에 필요한 긴 파일 이름을 지원하지 않기 때문입니다. Java 응용프로그램 또는 Java 애플릿으로 제어 센터를 설치하고 구성하는 자세한 내용은 사용자 플랫폼의 빠른 시작 매뉴얼을 참조하십시오.

관리용으로 Java 기반 도구 사용

DB2에는 사용자가 제어 센터의 기능을 확장할 수 있는 Java 인터페이스 세트가 포함됩니다. Java 인터페이스에서는 사용자가 다음을 수행할 수 있습니다.

- 오브젝트에 대해 작업할 때 추가 항목을 메뉴에 추가할 수 있습니다.
- 제어 센터 도구 모음에 버튼을 추가할 수 있습니다.

이 기능을 사용하려면, 적합한 레벨의 Java 소프트웨어가 설치되어 있어야 합니다. 595 페이지의 『부록K. 제어 센터 확장』에서 이 기능 사용에 대한 자세한 내용을 참조하십시오.

제2부 설계 구현

제2장 데이터베이스를 작성하기 전에

데이터베이스의 설계를 결정한 후에, 데이터베이스 및 오브젝트를 작성해야 합니다. 이들 오브젝트에는 스키마, 노드 그룹, 테이블 공간, 테이블, 뷰, 랩퍼, 서버, 별칭, 유형 맵핑, 함수 맵핑, 별명, 사용자 정의 유형(UDT), 사용자 정의 함수(UDF), 트리거, 제한조건, 색인 및 패키지가 포함됩니다. 이들 오브젝트는 제어 센터(Windows 및 OS/2 운영 체제)에서 또는 응용프로그램의 API를 통한 명령행 처리기의 SQL 문을 사용하여 작성할 수 있습니다.

SQL문에 대해서는 *SQL 참조서* 매뉴얼을 참조하십시오. 명령행 처리기 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오. APIs 사용에 대해서는 *Administrative API Reference* 매뉴얼을 참조하십시오.

주: 사용자의 플랫폼은 데이터베이스 오브젝트를 작성할 수 있는 사용자 인터페이스를 지원할 수 있습니다. SQL문, 명령행 처리기 명령 또는 사용자 API 대신 이러한 인터페이스를 사용할 수 있습니다. 이 기능이 있는지 알아보려면 사용 중인 플랫폼의 빠른 시작 매뉴얼을 참조하십시오.

이 장에서 제어 센터를 사용하여 작업을 완료하는 메소드는 상자를 띄워 강조표시됩니다. 이것은 명령행을 사용하는 비교대상 메소드가 바로 다음에 오며, 때로는 예제를 동반합니다. 어떤 경우에, 오직 하나의 메소드만을 표시하는 작업이 있을 수 있습니다. 제어 센터에서 작업할 때, 여기에 있는 개요 정보보다 더 자세한 세부사항을 제공하는 도움말을 사용할 수 있음을 기억하십시오.

이 장에서는 관련된 모든 오브젝트로 데이터베이스를 작성하기 전에 알아야 하는 정보에 초점을 둡니다. 데이터베이스를 작성하기 전에 수행해야 하는 여러 작업과 마찬가지로 여러 전제조건 개념 및 주제가 있습니다.

이 장의 다음 장에는 사용자 데이터베이스 설계 구현의 일부일 수 있는 다양한 오브젝트에 대한 간단한 설명이 들어 있습니다.

이 파트의 마지막 장은 데이터베이스를 변경하기 전에 고려해야 하는 주제를 나타낸 다음 데이터베이스 오브젝트를 변경하거나 제거하는 방법을 설명합니다.

DB2 Universal Database가 운영 체제와 상호 작용한 영역의 경우, 이 장과 다음 장의 일부 주제는 운영 체제 특유의 차이점을 나타냅니다. 운영 체제의 원시 성능 또는 DB2 UDB에 의해 제공된 성능간의 차이점을 사용할 수 있습니다. 정확한 차이점에 대해서는 해당 빠른 시작 매뉴얼과 특정 운영 체제 관련 문서를 참조하십시오.

예를 들어, Windows NT**는 『서비스』라고 하는 응용프로그램 유형을 지원합니다. Windows NT용 DB2에는 서비스라고 정의된 DB2 인스턴스가 있습니다. 서비스는 서비스 제어판 애플릿을 통한 사용자 또는 Microsoft** Win32** API에 포함된 서비스 기능을 사용하는 Win32 기반 응용프로그램에 의해 시스템 부트에서 자동으로 시작될 수 있습니다. 서비스는 시스템에 로그인한 사용자 없어도 실행할 수 있습니다.

데이터베이스를 작성하기 전에

데이터베이스를 구현하기 전에 다음 전제조건에 대해 이해하고 있어야 합니다.

- 『DB2 시작』
- 63 페이지의 『Windows NT에서 DB2 UDB 시작』
- 64 페이지의 『데이터베이스 관리 프로그램의 다중 인스턴스 사용』
- 65 페이지의 『스키마별 오브젝트 구성 및 그룹화』
- 66 페이지의 『병렬 처리 작동』
- 69 페이지의 『데이터 파티션 작동』
- 72 페이지의 『DB2 중지』

DB2 시작

일반적인 비지니스를 수행하는 동안 DB2를 시작하거나 중지시켜야 하는 경우가 있습니다. 예를 들어, 다음 작업을 수행하기 전에 인스턴스를 시작해야 합니다.

- 인스턴스에서 데이터베이스에 연결
- 응용프로그램 사전 처리 컴파일
- 패키지를 데이터베이스에 바인드
- 호스트 데이터베이스에 액세스

시스템에서 DB2 인스턴스를 시작하려면, 다음을 수행하십시오.

1. 인스턴스에 대해 SYSADM, SYSCTRL 또는 SYSDMAINT 권한이 있는 사용자 ID 및 이름으로 로그인하거나 인스턴스 소유자로 로그인하십시오.
2. UNIX 운영 체제에서, 다음과 같이 시동 스크립트를 수행하십시오.

```
. INSTHOME/sql1lib/db2profile      (Bourne 또는 Korn 셸의 경우)
source INSTHOME/sql1lib/db2cshrc   (C 셸의 경우)
```

여기서, INSTHOME은 사용할 인스턴스의 홈 디렉토리입니다.

3. 다음 두 메소드를 사용하여 인스턴스를 시작하십시오.
 - a. 제어 센터를 사용하여 인스턴스를 시작

- 1) 인스턴스 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
- 2) 오른쪽 마우스 버튼으로 시작하고자 하는 인스턴스를 클릭하고 팝업 메뉴에서 시작을 선택하십시오.

- b. 명령행을 사용하여 인스턴스를 시작하십시오.

```
db2start
```

주: **db2start** 명령은 82 페이지의 『현재 인스턴스 설정』에 있는 규칙에 따라 인스턴스를 시작합니다.

Windows NT에서 DB2 UDB 시작

db2start 명령은 DB2를 NT 서비스로서 시작합니다. Windows NT용 DB2는 DB2START를 호출할 때 "/D" 스위치를 지정하여 프로세스로서 실행될 수 있습니다. 또한, DB2는 제어판 또는 "NET START" 명령을 사용하여 서비스로서 시작될 수도 있습니다.

DB2START에서 DB2를 서비스로서 시작하려면, 사용자 계정은 Windows NT 운영 체제에 정의된 대로 NT 서비스를 시작할 올바른 특권을 가지고 있어야 합니다. 사용자 계정은 관리자, 서버 조작자 또는 전원 사용자 그룹의 구성원이 될 수 있습니다.

파티션된 데이터베이스 환경에서 실행될 때, 각 데이터베이스 파티션 서버는 NT 서비스로서 시작됩니다.

데이터베이스 관리 프로그램의 다중 인스턴스 사용

데이터베이스 관리 프로그램의 다중 인스턴스는 단일 서버에서 작성될 수 있습니다. 이는 사용자가 실제 머신에 동일한 제품의 여러 인스턴스를 작성하여 이를 동시에 수행할 수 있음을 의미합니다. 이는 환경 설정에 있어서 융통성을 제공합니다.

다음 환경 작성을 위해 다중 인스턴스를 가질 수도 있습니다.

- 사용자의 실행 환경에서 개발 환경을 분리시키기 위해
- 서비스하는 특정 응용프로그램에 대해 각각 개별적으로 조정하기 위해
- 관리자로부터 중요한 정보를 보호하기 위해. 예를 들어, 급여 데이터베이스가 자신의 인스턴스에 대해 보호되어 있어, 다른 인스턴스의 소유자가 급여 데이터를 볼 수 없게끔 합니다.

DB2 프로그램 파일은 특정 머신의 하나의 위치에 실제로 저장됩니다. 작성된 각각의 인스턴스가 다시 이 위치를 지시하고 있기 때문에, 작성된 각 인스턴스에 대해 프로그램 파일이 중복되지는 않습니다. 관련된 여러 데이터베이스가 하나의 인스턴스에 놓일 수 있습니다.

인스턴스는 노드 디렉토리에 지역 또는 원격으로 카탈로그화됩니다. 기본 인스턴스는 DB2INSTANCE 환경 변수에 의해 정의됩니다. 데이터베이스 작성, 응용프로그램의 강제, 데이터베이스의 모니터 또는 데이터베이스 관리 프로그램 구성의 갱신과 같이 인스턴스 레벨에서만 수행할 수 있는 작업을 개발하고 유지보수하기 위해 다른 인스턴스에 접속할 수 있습니다. 기본 인스턴스가 아닌 다른 인스턴스에 접속하려고 시도할 때, 해당 인스턴스와 통신하는 방법을 결정하는 데 노드 디렉토리가 사용됩니다.

*Command Reference*에서는 각 명령을 실행하는 데 필요한 연결 유형 정보를 설명합니다.

다중 인스턴스에 대한 DB2 지원은 운영 체제별로 다양합니다. 하나의 머신에서 다중 DB2 인스턴스를 정의하는 것에 대해서는 사용자 플랫폼에 적합한 빠른 시작 안내서를 참조하십시오.

원격 상태인 다른 인스턴스에 접속하려면, *Command Reference* 매뉴얼에 설명된 것처럼 ATTACH 명령을 사용하십시오.

제어 센터를 사용하십시오.

1. 인스턴스 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
2. 작업하려는 인스턴스를 클릭하십시오.
3. 인스턴스 이름을 선택하려면 오른쪽을 클릭하십시오.
4. DB2 창에서 작업하려면, 사용자 ID와 암호를 입력한 후 확인을 클릭하십시오.

명령행을 사용하여 인스턴스를 접속하십시오.

```
db2 attach to <database_name>
```

이 명령을 수행하면 노드 디렉토리 내에 이미 카탈로그화된 testdb2라는 인스턴스에 접속됩니다.

```
db2 attach to testdb2
```

testdb2 인스턴스에 대한 유지보수 활동을 수행한 다음, 다음 명령을 실행하여 해당 인스턴스에서 분리할 수 있습니다.

```
db2 detach
```

스키마별 오브젝트 구성 및 그룹화

데이터베이스 오브젝트 이름은 하나의 식별자로 구성되거나, 두 개의 식별자로 구성되는 스키마 규정 오브젝트입니다. 스키마 규정 오브젝트의 스키마 또는 상위 자리 부분은 데이터베이스의 그룹 오브젝트를 분류하거나 그룹화할 수 있는 방법을 제공합니다. 테이블, 뷰, 별명, 구별 유형, 함수, 색인, 패키지 또는 트리거와 같은 오브젝트가 작성될 때, 오브젝트는 스키마에 할당됩니다. 이러한 할당은 외부적으로 또는 내부적으로 이루어집니다.

명령문에서 해당 오브젝트를 참조할 때 두 부분으로 구성된 오브젝트 이름의 상위 자리 부분을 사용하는 경우 스키마를 명시적으로 사용하게 됩니다. 예를 들어, 사용자 A는 CREATE TABLE문을 다음과 같이 스키마 C로 발행합니다.

```
CREATE TABLE C.X (COL1 INT)
```

두 부분으로 구성된 오브젝트 이름의 상위 자리 부분을 사용하지 않는 경우 스키마를 내재적으로 사용하게 됩니다. 이 경우, 오브젝트 이름의 상위 자리 부분을 완료하는 데 사용되는 스키마 이름을 식별하는 데 CURRENT SCHEMA 특수 레지스터가 사용됩니다. CURRENT SCHEMA의 초기값은 현재 세션 사용자의 권한 부여 ID입니다. 현재 세션중에 이 값을 변경하려는 경우, SET SCHEMA문을 사용하여 다른 스키마 이름으로 특수 레지스터를 설정할 수 있습니다. SQL 참조서에서 자세한 내용을 참조하십시오.

125 페이지의 『시스템 카탈로그 테이블의 정의』에서 설명된 대로, 데이터베이스가 작성될 때 특정 스키마 내에 일부 오브젝트가 작성됩니다.

동적 SQL문에서, 스키마 규정 오브젝트 이름은 CURRENT SCHEMA 특수 레지스터 값을 규정되지 않은 오브젝트 이름 참조용 규정자로서 내재적으로 사용됩니다. 정적 SQL문에서, QUALIFIER 사전 처리 컴파일/바인드 옵션은 규정되지 않은 데이터베이스 오브젝트 이름에 대한 규정자를 내재적으로 지정합니다.

사용자 오브젝트를 작성하기 전에, 사용자 스키마에 오브젝트를 작성하려는지, 아니면 논리적으로 오브젝트를 그룹화하는 다른 스키마를 사용하여 작성하려는지를 고려해야 합니다. 공유될 오브젝트를 작성하는 경우에는 다른 스키마 이름을 사용하는 것이 좋습니다. 스키마를 확실히 작성하는 방법에 대해서는 138 페이지의 『스키마 작성』에서 자세한 내용을 참조하십시오.

병렬 처리 작동

구성 매개변수를 변경하여 데이터베이스 파티션 내의 또는 파티션되지 않은 데이터베이스 내의 병렬 처리를 사용해야 합니다. 예를 들어, 파티션 내 병렬 처리를 이용하여 대칭 다중 프로세서(SMP) 머신에서 다중 프로세서를 사용할 수 있습니다.

파티션 내 병렬 처리 작동

제어 센터는 특정 데이터베이스 또는 데이터베이스 관리 프로그램 구성 파일에서 각 항목의 값을 찾아내거나, 수정하는 데 사용될 수 있습니다.

특정 데이터베이스 또는 데이터베이스 관리 프로그램 구성 파일의 개별 항목 값을 알아내려면 GET DATABASE CONFIGURATION 및 GET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오. 특정 데이터베이스 또는

데이터베이스 관리 프로그램 구성 파일의 개별 항목 값을 변경하려면 UPDATE DATABASE CONFIGURATION 및 UPDATE DATABASE MANAGER CONFIGURATION 명령을 각각 사용하십시오.

파티션 내 병렬 처리에 영향을 미치는 구성 매개변수는 *max_querydegree* 및 *intra_parallel* 데이터베이스 관리 프로그램 매개변수, *dft_degree* 데이터베이스 매개변수를 포함합니다. 구성 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

파티션 내 조회 병렬 처리 작동

파티션 내 병렬 처리가 발생하려면, 데이터베이스 구성 매개변수와 데이터베이스 관리 프로그램 구성 매개변수를 변경해야 합니다.

INTRA_PARALLEL

데이터베이스 관리 프로그램 구성 매개변수. 이 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

DFT_DEGREE

데이터베이스 구성 매개변수. DEGREE 바인드 옵션 및 CURRENT DEGREE 특수 레지스터에 대한 기본값을 제공합니다. 이 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

DEGREE

정적 SQL에 대한 사전 처리 컴파일 또는 바인드 옵션. *Command Reference*에서 자세한 내용을 참조하십시오.

CURRENT DEGREE

동적 SQL용 특수 레지스터. *SQL 참조서*에서 자세한 내용을 참조하십시오.

구성 매개변수 설정값 및 병렬로 응용프로그램이 처리되도록 작동하는 방법에 관한 자세한 정보는 *관리 안내서: 성능의 "DB2 구성"*을 참조하십시오.

파티션간 조회 병렬 처리 작동

파티션간 병렬 처리는 이 파티션의 데이터베이스 파티션 수와 데이터 분배에 기초하여 자동으로 발생합니다.

유틸리티 병렬 처리 작동

이 절에서는 다음 유틸리티에 대한 파티션 내 병렬 처리 작동 방법의 개요를 설명합니다.

- 로드
- 색인 작성
- 데이터베이스/테이블 공간 백업
- 데이터베이스/테이블 공간 복원

유틸리티에 대한 파티션간 병렬 처리는 데이터베이스 파티션 수에 기초하여 자동으로 발생합니다.

로드: 로드 유틸리티가 자동으로 병렬 처리를 사용하거나 또는 사용자가 다음 매개변수를 LOAD 명령에서 사용할 수 있습니다.

- CPU_PARALLELISM
- DISK_PARALLELISM

LOAD 명령에 대해서는 *데이터 이동 유틸리티 안내* 및 참조서에서 자세한 내용을 참조하십시오.

AutoLoader: autoloader.cfg 파일의 LOAD 스펙에 대해 MODIFIED BY ANYORDER 매개변수를 지정하여 AutoLoader에 다중 파티션 프로세스를 작동시킬 수 있습니다. *데이터 이동 유틸리티 안내* 및 참조서에서 자세한 내용을 참조하십시오.

색인 작성: 색인을 작성할 때 병렬 처리를 작동시키려면,

- The INTRA_PARALLEL 데이터베이스 관리 프로그램 구성 매개변수가 ON 상태여야 합니다.
- 테이블은 병렬 처리에서 도움이 될 수 있도록 커야 합니다.
- SMP 머신에서 다중 프로세서가 작동 가능 상태여야 합니다.

CREATE INDEX문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

데이터 파티션 작동

다중 파티션 환경에서 수행할 때, CREATE DATABASE 명령 또는 sqlcrea() API를 사용하여 db2nodes.cfg 파일에 존재하는 모든 노드로부터 데이터베이스를 작성할 수 있습니다. 자세한 정보는 *Command Reference* 및 *Administrative API Reference* 매뉴얼을 참조하십시오.

파티션된 데이터베이스를 작성하기 전에, 우선 사용자는 데이터베이스가 작성될 인스턴스에 대해 지역 클라이언트가 될 것인지 원격 클라이언트가 될 것인지를 판별해야 합니다. 두번째로는 인스턴스에 접속해야 합니다. 그리고 어떤 데이터베이스 파티션이 데이터베이스에 대한 카탈로그 노드가 될 것인지를 선택해야 합니다. 접속하고 CREATE DATABASE 명령을 실행하는 데이터베이스 파티션은 해당 특정 데이터베이스에 대한 카탈로그 노드가 됩니다.

카탈로그 노드는 모든 시스템 카탈로그 테이블이 저장되는 데이터베이스 파티션입니다. 시스템 테이블의 모든 액세스는 이 데이터베이스 파티션을 거쳐야 합니다. 모든 연합 데이터베이스 오브젝트(랩퍼, 서버, 별명 등)는 이 모드에서 시스템 카탈로그 테이블에 저장됩니다.

가능하다면, 별도의 인스턴스에서 하나의 데이터베이스를 작성해야 합니다. 그렇지 않은 경우, 즉 인스턴스마다 둘 이상의 데이터베이스를 작성해야 한다면, 사용 가능한 데이터베이스 파티션간에 카탈로그 노드를 확대해야 합니다. 이렇게 하면, 단일 데이터베이스 파티션에서 카탈로그 정보에 대한 경합이 줄어듭니다.

주: 카탈로그 노드를 정기적으로 백업하고 데이터를 여기에 넣지 않아야(가능하면 언제나) 하는데, 그 이유는 다른 데이터가 백업에 필요한 시간을 증가시키기 때문입니다.

데이터베이스를 작성할 때, db2nodes.cfg 파일에서 정의된 모든 데이터베이스 파티션에 걸쳐 자동으로 작성됩니다.

시스템에서 첫번째 데이터베이스가 작성될 때, 시스템 데이터베이스 디렉토리가 형성됩니다. 이것은 사용자가 작성하는 다른 데이터베이스에 대한 정보와 함께 첨부됩니다. 시스템 데이터베이스 디렉토리는 sqlbdir이며, 홈 디렉토리 아래의 sqllib 디렉토리에 위치합니다. 이 디렉토리는 공유 파일 시스템(예를 들어, UNIX

플랫폼의 NFS)에 상주해야 하는데, 그 이유는 파티션된 데이터베이스를 구성하는 모든 데이터베이스 파티션용으로 하나의 시스템 데이터베이스 디렉토리만 있기 때문입니다.

또한, `sqldbdir` 디렉토리에는 시스템 인텐션 파일도 상주합니다. 이것은 `sqldbins` 이라 불리며, 데이터베이스 파티션이 동기화된 상태로 있도록 합니다. 모든 데이터베이스 파티션에 걸쳐 단 하나의 디렉토리만이 존재하기 때문에 이 파일은 공유 파일 시스템(예: UNIX 플랫폼의 NFS)에도 상주해야 합니다. 이 파일은 데이터베이스를 구성하는 모든 파티션에서 공유합니다.

데이터 파티션 기능을 활용하려면, 구성 매개변수가 수정되어야 합니다. 특정 데이터베이스 또는 데이터베이스 관리 프로그램 구성 파일의 개별 항목 값을 알아내려면 `GET DATABASE CONFIGURATION` 및 `GET DATABASE MANAGER CONFIGURATION` 명령을 사용하십시오. 특정 데이터베이스 인스턴스 또는 데이터베이스 관리 프로그램 구성 파일의 개별 항목 값을 변경하려면 `UPDATE DATABASE CONFIGURATION` 및 `UPDATE DATABASE MANAGER CONFIGURATION` 명령을 사용하십시오.

파티션된 데이터베이스에 영향을 주는 데이터베이스 관리 프로그램 구성 매개변수는 `conn_elapse`, `fcm_num_anchors`, `fcm_num_buffers`, `fcm_num_connect`, `fcm_num_rqb`, `max_connretries`, `max_coordagents`, `max_time_diff`, `num_poolagents` 및 `stop_start_time`이 있습니다.

구성 매개변수에 대해서는 **관리 안내서**: 성능에서 자세한 내용을 참조하십시오.

데이터베이스 테이블 공간 백업

데이터베이스 또는 테이블 공간을 백업할 때 입출력 병렬 처리를 작동시키려면, 다음을 수행하십시오.

- 둘 이상의 대상 미디어를 사용하십시오.
- 병렬 입출력에 대해 테이블 공간을 구성하십시오.
- `BACKUP` 명령에서 `PARALLELISM` 매개변수를 사용하여 병렬 처리 정도를 지정하십시오.

- BACKUP 명령에서 WITH num-buffers BUFFERS 매개변수를 사용하여 병렬 처리 정도를 수용할 만큼 충분한 버퍼가 있는지 확인하십시오. 버퍼의 수는 소유하고 있는 대상 미디어의 수에 선택된 병렬 처리 정도를 더한 값과 같아야 합니다.

또한, 다음과 같은 백업 버퍼 크기를 사용하십시오.

- 실행할 수 있을 만큼의 크기. 4MB 또는 8MB(1024 또는 2048페이지)가 적합합니다.
- 최소한 백업중인 테이블 공간의 가장 큰(Extent 크기 * 컨테이너 수) 제품만큼 크게

BACKUP DATABASE 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

데이터베이스테이블 공간 복원

데이터베이스 또는 테이블 공간을 복원할 때 입출력 병렬 처리를 작동시키려면, 다음을 수행하십시오.

- 둘 이상의 소스 미디어를 사용하십시오.
- 병렬 입출력에 대해 테이블 공간을 구성하십시오.
- RESTORE 명령에서 PARALLELISM 매개변수를 사용하여 병렬 처리 정도를 지정합니다.
- BACKUP 명령에서 WITH num-buffers BUFFERS 매개변수를 사용하여 병렬 처리 정도를 수용할 만큼 충분한 버퍼가 있는지 확인하십시오. 버퍼의 수는 소유하고 있는 대상 미디어의 수에 선택된 병렬 처리 정도를 더한 값과 같아야 합니다.

또한, 다음과 같은 복원 버퍼 크기를 사용하십시오.

- 실행할 수 있을 만큼의 크기. 4MB 또는 8MB(1024 또는 2048페이지)가 적합합니다.
- 최소한 복원중인 테이블 공간의 가장 큰(Extent 크기 * 컨테이너 수) 제품만큼 크게
- 백업 버퍼 크기와 같거나 짝수의 배수만큼

RESTORE DATABASE 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

DB2 중지

db2stop 명령은 서버에서만 실행될 수 있습니다 이 명령이 실행중일 때에는 어떠한 데이터베이스 연결도 허용되지 않지만, 인스턴스 접속이 있을 경우에는 DB2가 중지되기 전에 강제됩니다.

주: 명령행 처리기 세션이 인스턴스에 접속된 경우, **db2stop** 명령을 수행하기 전에 **종료** 명령을 수행하여 각 세션을 종료해야 합니다. **db2stop** 명령은 DB2INSTANCE 환경 변수로 정의된 인스턴스를 중지시킵니다.

시스템에서 DB2 인스턴스를 중지시키려면, 다음을 수행해야 합니다.

1. 인스턴스에 대해 SYSADM, SYSCTRL 또는 SYSMOINT 권한이 있는 사용자 ID 및 이름으로 로그인 또는 접속하거나 인스턴스 소유자로 로그인하십시오.
2. 중지하고자 하는 특정 데이터베이스에 연결된 모든 응용프로그램과 사용자를 표시하십시오. 중요한 응용프로그램이 실행되고 있지 않은지 확인하려면, 응용프로그램을 나열해야 합니다. 이에 대해 SYSADM, SYSCTRL 또는 SYSMOINT 권한이 필요합니다.
3. 모든 응용프로그램과 사용자를 데이터베이스에서 강제하십시오. 사용자를 강제하려면 SYSADM 및 SYSCTRL 권한이 필요합니다.
4. UNIX 운영 체제에서, 다음과 같이 시동 스크립트를 수행하십시오.

```
. INSTHOME/sql1lib/db2profile      (Bourne 또는 Korn 셸의 경우)  
source INSTHOME/sql1lib/db2cshrc  (C 셸의 경우)
```

여기서, INSTHOME은 사용할 인스턴스의 홈 디렉토리입니다.

5. 다음 방식 중 하나를 사용하여 인스턴스를 중지하십시오.

- a. 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
- b. 중지를 원한다면 각 인스턴스를 클릭하십시오.
- c. 인스턴스를 선택한 후 오른쪽을 클릭하면, 자동(팝업) 메뉴로부터 중지를 선택합니다.
- d. 창에서 중지를 확실하게 하려면, 확인을 클릭하십시오.

명령행을 사용하여 인스턴스를 중지하십시오.

```
db2stop
```

데이터베이스에서 세부사항 작성

데이터베이스를 작성하기 전에, 다음 작업을 고려하거나 수행해야 합니다.

- 데이터베이스의 논리적 특성 및 물리적 특성 설계
- 인스턴스 작성
- 환경 변수 및 프로파일 레지스트리 설정
- DB2 관리 서버(DAS)
- 노드 구성 파일 작성
- 데이터베이스 구성 파일 작성
- FCM 통신 작동

데이터베이스의 논리적 특성 및 물리적 특성 설계

데이터베이스를 작성하기 전에, 데이터베이스 설계에 대한 논리적 사항 및 물리적 사항을 결정해야 합니다. *관리 안내서: 계획*에서 논리 및 물리 데이터베이스 디자인에 대한 자세한 내용을 참조하십시오.

인스턴스 작성

인스턴스는 데이터베이스를 카탈로그로 만들고 구성 매개변수를 설정하는 논리 데이터베이스 관리 프로그램 환경입니다. 사용자의 필요에 따라 둘 이상의 인스턴스를 작성할 수 있습니다. 다중 인스턴스를 사용하여 다음을 수행할 수 있습니다.

- 개발 환경에 대해 하나의 인스턴스를 사용하고 생산 환경에 대해 또다른 인스턴스를 사용합니다.
- 특정 환경에 맞게 인스턴스를 조정합니다.
- 관련 정보에 대한 액세스를 제한합니다.
- 각 인스턴스에 대한 SYSADM, SYSCTRL 및 SYSMANT 권한의 지정을 제어합니다.
- 각 인스턴스에 대해 데이터베이스 관리 프로그램 구성을 최적화합니다.

- 인스턴스 실패의 영향을 제한합니다. 인스턴스가 실패하는 이벤트에서 하나의 인스턴스만 영향을 받습니다. 기타 인스턴스는 계속 정상적으로 기능할 수 있습니다.

다중 인스턴스에는 몇 가지 사소한 단점이 있습니다.

- 각 인스턴스에 추가 시스템 자원(가상 메모리 및 디스크 공간)이 필요합니다.
- 관리할 인스턴스가 추가되므로 더 많은 관리 작업이 필요합니다.

인스턴스 디렉토리는 데이터베이스 인스턴스에 속하는 모든 정보를 저장합니다. 인스턴스 디렉토리가 일단 작성되면, 위치를 변경할 수 없습니다. 디렉토리에는 다음이 들어 있습니다.

- 데이터베이스 관리 프로그램 구성 파일
- 시스템 데이터베이스 디렉토리
- 노드 디렉토리
- DB2 진단 파일(db2diag.log)
- 노드 구성 파일(db2nodes.cfg)
- 예외/레지스터 덤프 또는 DB2 프로세스에 대한 호출 스택과 같은 디버깅 정보가 들어 있는 다른 모든 파일

UNIX 운영 체제에서 인스턴스 디렉토리는 INSTHOME/sql1lib 디렉토리에 위치해 있으며, INSTHOME은 인스턴스 소유자의 홈 디렉토리입니다.

파티션된 데이터베이스 시스템에서 인스턴스 디렉토리는 인스턴스에 속하는 모든 데이터베이스 파티션 서버 사이에서 공유됩니다. 그러므로, 인스턴스 디렉토리는 인스턴스에 있는 모든 머신이 액세스할 수 있는 네트워크 공유 드라이브에서 작성되어야 합니다.

설치 프로시저의 일부로서, 『DB2』라고 하는 DB2의 초기 인스턴스를 작성할 수 있습니다. UNIX에서, 초기 인스턴스 이름은 이름 지정 규칙 지시사항 내에서 원하는 것을 지정할 수 있습니다. 인스턴스 이름은 디렉토리 구조 설정에 사용됩니다.

이 인스턴스를 즉시 사용할 수 있도록 지원하려면, 설치 중에 다음을 설정합니다.

- 환경 변수인 DB2INSTANCE를 『DB2』로 설정합니다.

- DB2 레지스트리 변수인 DB2INSTDEF를 『DB2』로 설정합니다.

UNIX에서, 기본 인스턴스 이름은 이름 지정 규칙 지시사항 내에서 원하는 것을 지정할 수 있습니다.

이들 설정값은 『DB2』를 기본 인스턴스로 설정합니다. 기본값으로 사용되는 인스턴스를 변경할 수 있지만, 먼저 추가 인스턴스를 작성해야 합니다.

DB2를 사용하기 전에, 각 사용자에게 데이터베이스 환경은 인스턴스에 액세스하고 DB2 프로그램을 수행할 수 있도록 갱신되어야 합니다. 이는 모든 사용자에게 적용됩니다(관리 사용자 포함).

UNIX 운영 체제에서, 데이터베이스 환경을 설정할 수 있도록 샘플 스크립트 파일이 제공됩니다. 파일은 Bourne 또는 Korn 셸의 경우 db2profile이고, C 셸의 경우 db2cshrc입니다. 이들 스크립트는 인스턴스 소유자의 홈 디렉토리에 있는 sqllib 서브디렉토리에 위치해 있습니다. 인스턴스 소유자 또는 인스턴스의 SYSADM 그룹에 속하는 모든 사용자는 인스턴스의 모든 사용자를 위해 스크립트를 사용자 정의할 수 있습니다. 또는, 스크립트는 각 사용자를 위해 복사되고 사용자 정의할 수도 있습니다.

샘플 스크립트에는 다음을 수행하기 위한 명령문이 들어 있습니다.

- 인스턴스 소유자의 홈 디렉토리의 sqllib 서브디렉토리에 있는 bin, adm 및 misc 서브디렉토리를 기존의 검색 경로에 추가하여 사용자의 PATH를 갱신합니다.
- DB2INSTANCE 환경 변수를 인스턴스 이름으로 설정합니다.

DB2 환경 자동 설정

주: 다음 설명은 UNIX 운영 체제 환경에만 적용됩니다.

기본적으로, 스크립트는 현재 세션이 지속하는 동안에만 사용자 환경에 영향을 줍니다. .profile 파일을 변경하여 사용자가 Bourne 또는 Korn 셸을 통해 로그인할 때 db2profile 스크립트를 자동으로 수행하도록 할 수 있습니다. C 셸의 사용자의 경우, .login 파일을 변경하여 db2shrc 스크립트 파일을 수행하게 만들 수 있습니다.

다음 명령문 중 하나를 `.profile` 또는 `.login` 스크립트 파일에 추가하십시오.

- 스크립트의 하나의 버전을 공유하는 사용자의 경우, 다음을 추가하십시오.

```
. INSTHOME/sql1lib/db2profile    (Bourne 또는 Korn 셸의 경우)
source INSTHOME/sql1lib/db2cshrc (C 셸의 경우)
```

여기서, `INSTHOME`은 사용할 인스턴스의 홈 디렉토리입니다.

- 홈 디렉토리에 스크립트의 사용자 정의된 버전이 있는 사용자에게 대해 다음을 추가하십시오.

```
. USERHOME/db2profile    (Bourne 또는 Korn 셸의 경우)
source USERHOME/db2cshrc (C 셸에서)
```

여기서, `USERHOME`은 사용자의 홈 디렉토리입니다.

DB2 환경의 수동 설정

주: 다음 설명은 UNIX 운영 체제 환경에만 적용됩니다.

사용할 인스턴스를 선택하려면, 다음 명령문 중 하나를 명령 프롬프트에 입력하십시오. 마침표(.) 및 공간이 필요합니다.

- 스크립트의 하나의 버전을 공유하는 사용자의 경우, 다음을 추가하십시오.

```
. INSTHOME/sql1lib/db2profile    (Bourne 또는 Korn 셸의 경우)
source INSTHOME/sql1lib/db2cshrc (C 셸의 경우)
```

여기서, `INSTHOME`은 사용할 인스턴스의 홈 디렉토리입니다.

- 홈 디렉토리에 스크립트의 사용자 정의된 버전이 있는 사용자에게 대해 다음을 추가하십시오.

```
. USERHOME/db2profile    (Bourne 또는 Korn 셸의 경우)
source USERHOME/db2cshrc (C 셸에서)
```

여기서, `USERHOME`은 사용자의 홈 디렉토리입니다.

동시에 둘 이상의 인스턴스에 대해 작업하려면, 별도의 창에서 사용할 각 인스턴스에 대해 스크립트를 수행하십시오. 예를 들어, `test` 및 `prod`라고 하는 두 개의 인스턴스가 있고 홈 디렉토리가 `/u/test` 및 `/u/prod`라고 가정하십시오.

창 1에서,

- Bourne 또는 Korn 셸에서, 다음을 입력하십시오.


```
. /u/test/sqllib/db2profile
```

- C 셸에서, 다음을 입력하십시오.

```
source /u/test/sqllib/db2cshrc
```

창 2에서,

- Bourne 또는 Korn 셸에서, 다음을 입력하십시오.

```
. /u/prod/sqllib/db2profile
```

- C 셸에서, 다음을 입력하십시오.

```
source /u/prod/sqllib/db2cshrc
```

창 1을 사용하여 test 인스턴스에 대해 작업하고 창 2를 사용하여 prod 인스턴스에 대해 작업하십시오.

주: which db2 명령을 입력하여 검색 경로가 올바르게 설정되었는지 확인하십시오. 이 명령은 DB2 CLP 실행 파일의 절대 경로를 리턴합니다. 이것이 인스턴스의 sqllib 디렉토리아에 위치해 있는지 검증하십시오.

시스템의 다중 인스턴스

시스템에 둘 이상의 인스턴스가 있을 수 있습니다. 한 번에 DB2의 하나의 인스턴스 내에서만 작업할 수 있습니다.

인스턴스 소유자 및 그룹, 즉 모든 인스턴스에 관련된 시스템 관리(SYSADM) 그룹. 인스턴스 소유자 및 SYSADM 그룹은 인스턴스 작성 프로세스 중에 지정됩니다. 하나의 사용자 ID 또는 사용자 이름은 하나의 인스턴스에만 사용될 수 있습니다. 해당 사용자 ID 또는 사용자 이름을 인스턴스 소유자라고도 합니다.

각 인스턴스 소유자에게는 고유 홈 디렉토리가 있어야 합니다. 인스턴스 수행에 필요한 모든 파일은 인스턴스 소유자의 사용자 ID 또는 사용자 이름의 홈 디렉토리에 작성됩니다. 시스템에서 인스턴스 소유자의 사용자 ID 또는 사용자 이름을 제거해야 하는 경우에는 잠재적으로 인스턴스에 관련된 파일을 유실하고 이 인스턴스에 저장된 데이터에 액세스할 수 없게 됩니다. 이러한 이유로, 인스턴스 소유자의 사용자 ID 또는 사용자 이름을 DB2 수행에만 사용하도록 하는 것이 바람직합니다.

인스턴스 소유자의 1차 그룹 또한 중요합니다. 이 1차 그룹은 인스턴스에 대해 자동으로 시스템 관리 그룹이 되고 인스턴스에 대한 SYSADM 권한을 얻습니다. 인스턴스 소유자의 1차 그룹의 구성원인 기타 사용자 ID 또는 사용자 이름도 이 레벨의 권한을 얻습니다. 이러한 이유로, 인스턴스 소유자의 사용자 ID 또는 사용자 이름을 인스턴스 관리를 위해 예약해 놓은 1차 그룹에 지정할 수 있습니다(또한, 1차 그룹을 인스턴스 소유자의 사용자 ID 또는 사용자 이름에 지정하도록 하십시오. 그렇지 않으면, 시스템 기본 1차 그룹이 사용됩니다).

이미 인스턴스에 대해 시스템 관리 그룹으로 만들고 싶은 그룹이 있으면, 인스턴스 소유자의 사용자 ID 또는 사용자 이름을 작성할 때 이 그룹을 1차 그룹으로 지정하기만 하면 됩니다. 인스턴스에서 기타 사용자에게 관리 권한을 부여하려면, 시스템 관리 그룹으로 지정된 그룹에 사용자를 추가하십시오.

인스턴스간에 SYSADM 권한을 분리하려면, 각 인스턴스 소유자의 사용자 ID 또는 사용자 이름이 다른 1차 그룹을 사용해야 합니다. 그러나, 다중 인스턴스에 대해 공통 SYSADM 권한이 있도록 선택하는 경우, 다중 인스턴스에 대해 동일한 1차 그룹을 사용할 수 있습니다.

인스턴스 추가

OS/2에서 관리 권한이 있거나 Windows NT에서 관리 그룹에 속하는 경우, UNIX 플랫폼에서 루트 권한이 있는 DB2 인스턴스를 추가할 수 있습니다. 인스턴스를 추가하는 머신은 인스턴스 소유 머신이 됩니다(노드 0). 관리 서버가 상주하는 머신에서 인스턴스를 추가하십시오.

인스턴스를 추가하려면, 다음 단계를 수행하십시오.

1. 관리 권한이 있거나 관리자 그룹에 속하는 사용자 ID 또는 이름으로 로그인하십시오.
2. 인스턴스를 추가하려면, 다음 방식 중 하나를 사용하십시오.
제어 센터를 사용하십시오.

- a. 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
- b. 인스턴스 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 추가를 선택하십시오.
- c. 정보를 완료하고, 적용을 클릭하십시오.

명령행을 사용하여 인스턴스를 추가하십시오.

```
db2icrt <instance_name>
```

3. 관리 서버를 작성하십시오.

db2icrt 명령을 사용할 때에는, 인스턴스 소유자의 로그인 이름을 제공하고, 인스턴스의 인증 유형을 선택적으로 지정해야 합니다. 인증 유형은 해당 인스턴스에서 작성된 모든 데이터베이스에 적용됩니다. 인증 유형은 사용자 인증이 일어나는 곳의 명령문입니다. 인증에 대해서는 249 페이지의 『제5장 데이터베이스 액세스 제어』에서 자세한 내용을 참조하십시오.

주: **db2iupdt** 명령을 사용하여 인스턴스 구성을 갱신하도록 선택할 수 있습니다.

DB2INSTPROF 환경 변수를 사용하여 인스턴스 디렉토리의 위치를 DB2PATH 로 변경할 수 있습니다. 인스턴스 디렉토리에 대해 쓰기 액세스가 필요합니다. DB2PATH 외의 경로에서 디렉토리를 작성하려면, **db2icrt** 명령을 입력하기 전에 DB2INSTPROF를 설정해야 합니다.

인스턴스 추가 시 DB2 Enterprise - Extended Edition 세부사항: DB2 Universal Database Enterprise - Extended Edition으로 작업할 때, 파티션된 데이터베이스 시스템인 새로운 인스턴스를 추가 중임을 선언할 필요가 있습니다. 이는 명령행에서 **-s eee**를 사용하여 이루어집니다.

인스턴스를 작성할 때 UNIX 세부사항: UNIX 운영 체제에서 작업할 때, **db2icrt** 명령에는 다음과 같은 선택적 매개변수가 있습니다.

- **-h** 또는 **-?**

이 매개변수는 명령에 대한 도움말 메뉴를 표시할 때 사용됩니다.

- **-d**

이 매개변수는 문제점 판별중에 사용할 디버그 모드를 설정합니다.

- **-a AuthType**

이 매개변수는 인스턴스에 대한 인증 유형을 지정합니다. 유효한 인증 유형은 SERVER, CLIENT, DCS 또는 DCE입니다. 지정되지 않은 경우 DB2 서버가 설치되면, 기본값은 SERVER입니다. 그렇지 않으면, CLIENT로 설정됩니다.

주:

1. 인스턴스의 인증 유형은 인스턴스에 속하는 모든 데이터베이스에 적용됩니다.
2. UNIX 운영 체제에서 인증 유형 DCE는 유효하지 않은 선택입니다.

- -u FencedID

이 매개변수는 분리 사용자 정의 함수(UDF) 및 저장 프로시저가 실행되는 사용자입니다. 이는 DB2 클라이언트 또는 DB2 응용프로그램 개발 클라이언트를 치할 때는 필요하지 않습니다. 기타 DB2 제품에서는 필수 매개변수입니다.

주: FencedID는 『root』 또는 『bin』이 될 수 없습니다.

- -p PortName

이 매개변수는 사용될 TCP/IP 서비스 이름 또는 포트 번호를 지정합니다. 이 값은 인스턴스의 데이터베이스 구성 파일에 설정됩니다.

- -s InstType

서로 다른 유형의 인스턴스를 추가할 수 있게 합니다. 유효한 인스턴스 유형은 ee, eee 및 client입니다.

예:

- DB2 서버에 대해 인스턴스를 추가하려면, 다음 명령을 사용하십시오.

```
db2icrt -u db2fenc1 db2inst1
```

- DB2 Connect Enterprise Edition만 설치한 경우, 인스턴스 이름을 분리(fenced) ID로 사용할 수도 있습니다.

```
db2icrt -u db2inst1 db2inst1
```

- DB2 클라이언트에 대해 인스턴스를 추가하려면, 다음 명령을 사용하십시오.

```
db2icrt db2inst1 -s client
```

DB2 클라이언트 인스턴스는 워크스테이션을 다른 데이터베이스 서버에 연결하려고 할 때 작성되며 해당 워크스테이션에서 지역 데이터베이스가 필요하지 않습니다.

인스턴스를 작성할 때 Windows NT 세부사항: Windows NT 운영 체제에서 작업할 때, **db2icrt** 명령에는 다음과 같은 선택적 매개변수가 있습니다.

|

- -s InstType

| 서로 다른 유형의 인스턴스를 작성할 수 있게 합니다. 유효한 인스턴스 유형은 ee, eee 및 client입니다.

|

- /p:InstProf_Path

| 이것은 다른 인스턴스 프로파일 경로를 지정하기 위한 선택적 매개변수입니다. 경로를 지정하지 않으면, 인스턴스 디렉토리는 SQLLIB 디렉토리하에서 작성되고 주어진 공유 이름인 DB2는 인스턴스 이름에 결합됩니다. 읽기 및 쓰기 승인은 도메인에 있는 모든 사용자에게 자동으로 권한 부여됩니다. 승인을 변경하여 디렉토리에 대한 액세스를 제한할 수 있습니다.

| 다른 인스턴스 프로파일 경로를 지정하면, 공유 드라이브 또는 디렉토리를 작성해야 합니다.

|

- /u:username,password

| 파티션된 데이터베이스 환경을 작성할 때, DB2 서비스의 로그인 및 계정 이름과 암호를 선언해야 합니다.

|

- /r:base_port,end_port

| 이것은 FCM(Fast Communications Manager)의 TCP/IP 포트 범위를 지정하기 위한 선택적 매개변수입니다. TCP/IP 포트 범위를 지정하는 경우, 포트 범위가 파티션된 데이터베이스 시스템의 모든 머신에서 사용 가능해야 합니다.

| 예를 들어, Windows NT용 Enterprise Edition에서 다음과 같은 예가 있을 수 있습니다.

| db2icrt inst1 -s eee
| /p:\machineA\db2mpp
| /u:yourname,yourpwd /r:9010,9015

| **주:** db2icrt 명령은 인스턴스를 작성하는 데 사용되는 사용자 이름을 권한 부여합니다.

- 운영 체제의 일부로 작동시키십시오.
- 토큰 오브젝트를 작성하십시오.
- 할당량을 증가시키십시오.
- 서비스로 로그인하십시오.
- 프로세스 레벨 토큰을 대체하십시오.

공유 드라이브에 액세스하고, 사용자 계정을 인증하며, DB2를 Windows NT 서비스로 수행하기 위해 인스턴스는 이러한 사용자 권한이 필요합니다.

인스턴스 나열

시스템에서 사용할 수 있는 모든 인스턴스 목록을 확보하려면, 제어 센터를 사용하십시오.

1. 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 인스턴스 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 추가를 선택하십시오.
3. 『데이터베이스 추가』 창에서, 새로 고침을 클릭하십시오.
4. 원격 데이터베이스 인스턴스를 보려면 드롭다운 화살표를 클릭하십시오.
5. 창을 나가려면 취소를 클릭하십시오.

명령행을 사용하여 시스템에서 사용할 수 있는 모든 인스턴스 목록을 나열하려면, 다음을 입력하십시오.

```
db2ilist
```

현재 세션에 적용되는 인스턴스를 판별하려면, 다음을 입력하십시오.

```
set db2instance
```

주: UNIX 운영 체제에서,

```
db2 get instance
```

현재 인스턴스 설정

명령을 수행하여 인스턴스의 데이터베이스 관리 프로그램을 시작하거나 중지시키려면, DB2는 명령을 현재 인스턴스에 적용합니다. DB2는 다음과 같이 현재 인스턴스를 판별합니다.

- DB2INSTANCE 환경 변수가 현재 세션에 대해 설정되면, 값은 현재 인스턴스가 됩니다. DB2INSTANCE 환경 변수를 설정하려면, 다음을 입력하십시오.

```
set db2instance=<new_instance_name>
```

- DB2INSTANCE 환경 변수가 현재 세션에 대해 설정되지 않으면, DB2는 시스템 환경 변수에서 DB2INSTANCE 환경 변수에 대한 설정을 사용합니다.

Windows NT에서 시스템 환경 변수는 시스템 환경에서 설정됩니다. Windows 95에서는 autoexec.bat 파일에 설정됩니다. OS/2에서는 config.sys 파일에 설정됩니다.

- DB2INSTANCE 환경 변수가 전혀 설정되지 않으면, DB2는 DB2INSTDEF 레지스트리 변수를 사용합니다.

레지스트리의 전역 레벨에서 DB2INSTDEF 레지스트리 변수를 설정하려면, 다음을 입력하십시오.

```
db2set db2instdef=<new_instance_name> -g
```

인스턴스 자동 시작

UNIX 운영 체제에서, 시스템이 재시작할 때마다 인스턴스가 자동으로 시작되게 하려면, 다음 명령을 입력하십시오.

```
db2iauto -on InstName
```

여기서, InstName은 인스턴스의 로그인 이름입니다.

UNIX 운영 체제에서, 시스템이 재시작할 때마다 인스턴스가 자동으로 시작되지 않도록 하려면, 다음 명령을 입력하십시오.

```
db2iauto -off InstName
```

여기서, InstName은 인스턴스의 로그인 이름입니다.

다중 인스턴스 동시 수행

다중 DB2 인스턴스가 같은 코드 레벨에 속하는 한 이를 시작할 수 있습니다.

제어 센터를 사용하여 동시에 다중 인스턴스를 수행하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 오른쪽 마우스 버튼으로 인스턴스를 클릭하고 팝업 메뉴에서 시작을 선택하십시오.
3. 동시에 수행하려는 모든 인스턴스를 시작할 때까지 2 단계를 반복하십시오.

명령행을 사용하여 다중 인스턴스를 수행하십시오.

- 다음을 입력하여 DB2INSTANCE 변수를 원하는 다른 인스턴스의 이름으로 설정하십시오.

```
set db2instance=<another_instName>
```

- **db2start** 명령을 입력하여 인스턴스를 시작하십시오.

사용권 관리

DB2 제품에 대한 사용권 관리는 제품에 대한 온라인 인터페이스의 제어 센터에 있는 사용권 센터를 통해 주로 이루어집니다. 사용권 센터에서 사용권 정보, 통계, 등록된 사용자, 그리고 설치된 각 제품의 현재 사용자를 확인할 수 있습니다.

환경 변수 및 프로파일 레지스트리 설정

환경 변수 및 레지스트리 변수는 데이터베이스 환경을 제어합니다.

DB2 프로파일 레지스트리가 도입되기 전에, Windows 또는 OS/2 워크스테이션에서 환경 변수를 변경하려면 환경 변수를 변경한 후 재부팅해야 했습니다. 이제, 사용자의 환경은 DB2 프로파일 레지스트리에 저장된 레지스트리 변수에 의해 제어됩니다(몇몇 예외 제외). 주어진 인스턴스에 대한 시스템 관리(SYSADM) 권한이 있는 사용자는 해당 인스턴스에 대해 레지스트리값을 갱신할 수 있습니다. 재부팅하지 않고 레지스트리 값을 갱신하려면 **db2set** 명령을 사용하십시오. 이 정보는 즉시 프로파일 레지스트리에 저장됩니다. DB2 레지스트리는 변경이 이루어진 후 시작된 DB2 서버 인스턴스 및 DB2 응용프로그램에 갱신된 정보를 적용합니다.

레지스트리를 갱신할 때 변경사항은 현재 수행중인 DB2 응용프로그램 또는 사용자에게 영향을 주지 않습니다. 갱신 다음에 시작되는 응용프로그램은 새 값을 사용합니다.

주: DB2 환경 변수 DB2INSTANCE, DB2NODE, DB2PATH 및 DB2INSTPROF는 운영 체제에 따라 DB2 프로파일 레지스트리에 저장되지 않을 수도 있습니다. 이러한 환경 변수를 갱신하려면, **set** 명령을 사용하여 시스템을 재부팅해야 합니다. UNIX 플랫폼에서는 **set** 명령 대신 내보내기 명령이 사용되고 시스템 재부팅은 필요하지 않습니다.

프로파일 레지스트리를 사용하면 환경 변수를 중앙에서 제어할 수 있습니다. 관리 안내서: 성능의 『DB2 레지스트리 및 환경 변수』에는 많은 환경 변수 및 레지스

트리 변수가 나열되어있습니다. 다양한 환경 프로파일을 통해 다양한 레벨의 지원이 제공됩니다. DB2 관리 서버를 사용하면, 환경 변수를 원격으로 관리할 수도 있습니다.

다음은 네 개의 프로파일 레지스트리입니다.

- DB2 인스턴스 레벨 프로파일 레지스트리. 대부분의 DB2 환경 변수는 이 레지스트리 내에 배치됩니다. 이 레지스트리에는 특정 인스턴스용 환경 변수 설정값이 있습니다. 이 레벨에서 정의된 값은 전역 레벨에서 설정값을 대체합니다.
- DB2 전역 레벨 프로파일 레지스트리. 특정 인스턴스에 대해 환경 변수가 설정되어 있지 않을 경우, 이 레지스트리가 사용됩니다. 이 레지스트리에는 장치 전체에 적용되는 환경 변수 설정값이 있습니다. DB2 UDB EEE에는 각 머신마다 하나의 전역 레벨 프로파일이 있습니다.
- DB2 인스턴스 노드 레벨 프로파일 레지스트리. 다양한 데이터베이스 파티션간에 데이터베이스가 파티션된 시스템에서, 이 레지스트리는 모든 노드(머신)에 상주하며 노드에 데이터를 저장한 모든 인스턴스에 대한 환경 변수 설정값이 들어 있습니다. 이 레벨에서 정의된 값은 인스턴스 및 전역 레벨의 비교 가능한 설정값을 대체합니다.
- DB2 인스턴스 프로파일 레지스트리. 이 레지스트리에는 이 시스템이 인식하는 모든 인스턴스 이름 목록이 들어 있습니다.

사용자는 **db2set** 명령을 사용하여 세션 환경 변수 설정값을 변경하여 해당 세션에 대한 DB2 인스턴스 프로파일 레지스트리 환경 변수 설정값을 대체할 수 있습니다.

DB2는 레지스트리 값 및 환경 변수를 점검하고, 다음 순서로 해결함으로써, 운영 환경을 구성합니다.

1. set 명령으로 설정된 환경 변수(또는 UNIX 플랫폼에서 내보내기 명령).
2. 인스턴스 노드 레벨 프로파일(아래의 노드 번호를 가진 db2set -i 명령 사용)로 설정된 레지스트리 값.
3. db2set 명령으로 설정된 레지스트리 값.
4. 인스턴스 프로파일(아래의 db2set -i 명령 사용)로 설정된 레지스트리 값.
5. 전역 프로파일(아래의 db2set -g 명령 사용)로 설정된 레지스트리 값.

db2set 명령 사용

db2set 명령은 레지스트리 값(및 환경 변수)의 지역 선언을 지원합니다.

명령에 대한 도움말 정보를 표시하려면, 다음을 사용하십시오.

```
db2set ?
```

지원되는 모든 레지스트리 변수의 전체 세트를 나열하려면, 다음을 사용하십시오.

```
db2set -lr
```

이 세션에 대해 현재 정의된 레지스트리 변수를 나열하려면, 다음을 사용하십시오.

```
db2set
```

프로파일 레지스트리에서 정의된 모든 레지스트리 변수를 나열하려면, 다음을 사용하십시오.

```
db2set -all
```

레지스트리 변수의 현재 세션 값을 표시하려면, 다음을 사용하십시오.

```
db2set registry_variable_name
```

모든 레벨에서 레지스트리 변수의 값을 표시하려면, 다음을 사용하십시오.

```
db2set registry_variable_name -all
```

특정 레벨에서 변수 값을 삭제하려면, 같은 명령 구문을 사용하여 변수를 설정할 수 있지만 변수 값에 대해 아무것도 지정하지 마십시오. 예를 들어, 노드 레벨에서 변수의 설정을 삭제하려면, 다음을 입력하십시오.

```
db2set registry_variable_name= -i instance_name  
node_number
```

변수 값이 더 높은 프로파일 레벨에서 정의된 경우, 이를 삭제하고 사용을 제한하려면, 다음을 입력하십시오.

```
db2set registry_variable_name= -null instance_name
```

이 명령은 사용자가 지정한 매개변수에 대한 설정을 삭제하고 높은 레벨의 프로파일 이 이 변수 값을 변경하지 못하도록 제한할 수 있습니다(이 경우, DB2 전역 레벨 프로파일). 그러나, 사용자가 지정한 변수는 더 낮은 레벨의 프로파일(이 경우, DB2 노드 레벨 프로파일)로 설정될 수는 있습니다.

이 세션만의 레지스트리 변수를 변경하려면, 다음을 사용하십시오.

```
db2set registry_variable_name=new_value
```

인스턴스의 모든 데이터베이스에 대한 레지스트리 변수 기본값을 변경하려면, 다음을 사용하십시오.

```
db2set registry_variable_name=new_value  
-i instance_name
```

시스템의 모든 인스턴스에 대한 레지스트리 변수 기본값을 변경하려면, 다음을 사용하십시오.

```
db2set registry_variable_name=new_value -g
```

사용자 레벨에서 레지스트리 변수를 설정하려면, 다음을 사용하십시오.

```
db2set -u1
```

특정 사용자에 대해 사용자 레벨에서 레지스트리 변수를 설정하려면, 다음을 사용하십시오.

```
db2set -u1 user_name
```

주:

1. 매개변수 "-i", "-g" 및 "-u1"은 같은 명령에서 동시에 사용될 수 없습니다.
2. 일부 매개변수는 기본적으로 항상 전역 레벨 프로파일을 사용합니다. 이를 인스턴스 또는 노드 레벨 프로파일에서 설정할 수 없습니다(예: db2system 및 db2instdef).
3. UNIX에서, 인스턴스에 대한 레지스트리 값을 변경하려면, 시스템 관리 권한 (SYSADM)이 있어야 합니다. 루트 권한이 있는 사용자만이 전역 레벨 레지스트리에 있는 매개변수를 변경할 수 있습니다.

LDAP 환경에서 수행할 때, 범위가 디렉토리 파티션이나 Windows NT 도메인에 속하는 모든 머신 및 모든 사용자에게 전역인 DB2 레지스트리 변수 값을 LDAP에 설정하는 것이 가능합니다. 현재, LDAP 전역 레벨에 설정될 수 있는 유일한 DB2 레지스트리 변수는 DB2LDAP_SEARCH_SCOPE입니다.

LDAP 전역 레벨에서 이 변수를 설정하려면, db2set 명령에 대해 -g1 옵션을 사용하십시오.

| 주: 이것은 머신 전역 레벨에서 DB2 레지스트리 변수를 설정하는 데 사용되는 -g
| 옵션과는 다릅니다. -g1은 LDAP 전역 레벨에 고유합니다. 또한, 이 LDAP
| 에서 이 DB2 레지스트리 변수를 설정하는 것은 Windows 플랫폼에서만 지
| 원됩니다.

| LDAP의 전역 레벨에서 검색 범위 값을 설정하려면, 다음을 사용하십시오.

```
db2set -g1 db2ldap_search_scope = value
```

| 여기서, *value*는 『지역』, 『도메인』 또는 『전역』일 수 있습니다.

| 인스턴스의 특정 노드에 대한 레지스트리 변수 기본값을 변경하려면, 다음을 사용
| 하십시오.

```
db2set registry_variable_name=new_value  
-i instance_name node_number
```

| 인스턴스에 대한 모든 레지스트리 변수를 전역 프로파일 레지스트리의 기본값으로
| 재설정하려면, 다음을 사용하십시오.

```
db2set -r registry_variable_name
```

| 노드에 대한 모든 레지스트리 변수를 전역 프로파일 레지스트리의 기본값으로 재
| 설정하려면, 다음을 사용하십시오.

```
db2set -r registry_variable_name node_number
```

OS/2에 환경 변수 설정

모든 DB2 고유 레지스트리 변수를 DB2 프로파일 레지스트리에서 정의하는 것이
바람직합니다. DB2 변수가 레지스트리 외부에서 설정되면, 해당 변수를 원격으로
관리할 수 없게 되며 워크스테이션은 변수 값이 적용되도록 재부팅되어야 합니다.

OS/2에서, DB2PATH 및 DB2INSTPROF와 구별하여 config.sys에서 정의된 환
경 변수는 없습니다. 모든 변수는 실제 환경 변수로 남아 있는 값을 제외하고
db2set 명령을 사용하여 프로파일 레지스트리에서 정의된 상태여야 합니다.

DB2INSTANCE도 진정한 환경 변수로 남지만, DB2INSTDEF 레지스트리 변수
를 사용할 경우 반드시 필요한 것은 아닙니다. 이 레지스트리 변수는
DB2INSTANCE가 설정되지 않은 경우 사용할 기본 인스턴스 이름을 정의합니다.

DB2INSTANCE 및 DB2PATH는 DB2가 설치될 때 설정됩니다. DB2INSTPROF는 설치 후에 설정됩니다. DB2PATH 환경 변수가 반드시 설정되어야 합니다. 이 환경 변수는 설치중에 설정되고 사용자는 이를 수정하지 않아야 합니다. DB2INSTANCE 및 DB2INSTPROF 환경 변수는 선택적입니다.

환경 변수의 설정을 판별하려면, 다음을 입력하십시오.

```
set variable
```

환경 변수의 설정을 변경하려면, 다음 명령을 입력하십시오.

```
set variable=value
```

시스템 환경 변수를 설정하려면, config.sys 파일을 편집한 후, 변경사항이 효과를 발휘하도록 시스템을 재부팅하십시오.

다음에 따라 다양한 프로파일 레지스트리가 배치됩니다.

- DB2 인스턴스 레벨 프로파일 레지스트리 파일은 다음 위치에 배치됩니다.

```
%DB2INSTPROF%\instance_name\PROFILE.ENV
```

주: *instance_name*은 작업중인 데이터베이스 파티션에 대해 고유합니다.

- DB2 전역 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

```
%DB2INSTPROF%\DEFAULT.ENV
```

- DB2 인스턴스 노드 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

```
%DB2INSTPROF%\instance_name\NODES\node_number.ENV
```

주: *instance_name*과 *node_number*는 작업중인 데이터베이스 파티션에 대해 고유합니다.

정의된 모든 노드를 추적하는 추가 레지스트리 파일이 있습니다. 이 파일의 정보는 db2nodes.cfg 파일의 내용과 거의 비슷합니다.

```
%DB2INSTPROF%\instance_name\NODES.CFG
```

- DB2 인스턴스 프로파일 레지스트리는 다음 위치에 배치됩니다.

```
%DB2INSTPROF%\PROFILES.REG
```

Windows NT 및 Windows 95에서 환경 변수 설정

모든 DB2 고유 레지스트리 변수를 DB2 프로파일 레지스트리에서 정의하는 것이 바람직합니다. DB2 변수가 레지스트리 외부에서 설정되면, 해당 변수를 원격으로 관리할 수 없게 되며 워크스테이션은 변수 값이 적용되도록 재부팅되어야 합니다.

Windows 32비트 운영 체제에는 하나의 DB2INSTANCE 시스템 환경 변수가 있는데, 이는 프로파일 레지스트리 외부에서만 설정될 수 있습니다. 그러나, DB2INSTANCE를 설정할 필요는 없습니다. DB2 프로파일 레지스트리 변수인 DB2INSTDEF를 전역 레벨 프로파일에서 설정하여 DB2INSTANCE가 정의되지 않은 경우 사용할 인스턴스 이름을 지정할 수 있습니다.

Windows NT용 DB2 Enterprise - Extended Edition 서버에는 프로파일 레지스트리 외부에서만 설정될 수 있는 DB2INSTANCE 및 DB2NODE의 두 개의 시스템 환경 변수가 있습니다. DB2INSTANCE를 설정할 필요는 없습니다. DB2 프로파일 레지스트리 변수인 DB2INSTDEF를 전역 레벨 프로파일에서 설정하여 DB2INSTANCE가 정의되지 않은 경우 사용할 인스턴스 이름을 지정할 수 있습니다.

DB2NODE 환경 변수는 머신 내에서 대상 논리 노드로 라우트 요청을 하기 위해 사용됩니다. 이 환경 변수는 DB2 프로파일 레지스트리가 아닌 응용프로그램 또는 명령이 발행되는 세션에서 설정되어야 합니다. 이 변수가 설정되지 않으면, 대상 논리 노드는 기본적으로 머신의 포트 0으로 정의되는 논리 노드가 됩니다.

환경 변수의 설정값을 판별하려면, **echo** 명령을 사용하십시오. 예를 들어, DB2PATH 환경 변수의 값을 확인하려면, 다음을 입력하십시오.

```
echo %db2path%
```

시스템 환경 변수를 설정하려면, 다음을 수행하십시오.

Windows 95 및 Windows 98에서: *autoexec.bat* 파일을 편집한 후 시스템을 재부팅하여 변경사항을 적용하십시오.

Windows NT 4.x에서: DB2 환경 변수인 DB2INSTANCE, DB2PATH 및 DB2INSTPROF를 다음과 같이 설정할 수 있습니다.

- 시작, 설정, 제어판을 선택하십시오.

- 시스템 아이콘을 두 번 클릭하십시오.
- 시스템 제어판의 시스템 환경 변수 섹션에서, 다음을 수행하십시오.
 1. DB2INSTANCE 변수가 없으면,
 - a. 시스템 환경 변수를 선택하십시오.
 - b. 변수 필드에 있는 이름을 DB2INSTANCE로 변경하십시오.
 - c. 값 필드를 db2inst와 같은 인스턴스 이름으로 변경하십시오.
 2. DB2INSTANCE 변수가 이미 있으면, 새 값을 첨부하십시오.
 - a. DB2INSTANCE 환경 변수를 선택하십시오.
 - b. 값 필드를 db2inst와 같은 인스턴스 이름으로 변경하십시오.
 3. 설정을 선택하십시오.
 4. 확인을 선택하십시오.
 5. 이 변경사항을 적용하려면 시스템을 재부팅하십시오.

주: DB2INSTANCE 환경 변수는 세션(프로세스) 레벨에서도 설정될 수 있습니다.

예를 들어, TEST라고 하는 두번째 DB2 인스턴스를 시작하려면, 명령 창에서 다음 명령을 발행하십시오.

```
set db2instance=TEST
db2start
```

프로파일 레지스트리는 다음과 같이 배치됩니다.

- Windows NT 운영 체제 레지스트리에서 DB2 인스턴스 레벨 프로파일 레지스트리는 다음 경로를 가지고 있습니다.

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\PROFILES\instance_name
```

주: *instance_name*은 작업중인 데이터베이스 파티션에 대해 고유합니다.

- Windows NT 레지스트리에서 DB2 전역 인스턴스 레벨 프로파일 레지스트리는 다음 경로를 가지고 있습니다.

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\GLOBAL_PROFILE
```

- Windows NT 레지스트리에서 DB2 인스턴스 노드 레벨 프로파일 레지스트리는 다음 경로를 가지고 있습니다.

```
...\SOFTWARE\IBM\DB2\PROFILES\instance_name\NODES\node_number
```

주: *instance_name*과 *node_number*는 작업중인 데이터베이스 파티션에 대해 고유합니다.

DB2 UDB는 원격 머신의 인스턴스 레벨에서 DB2 UDB 레지스트리 변수에 액세스하는 기능을 제공합니다. DB2 UDB 레지스트리 변수는 세 가지 레벨인 머신 또는 전역 레벨, 인스턴스 레벨 및 노드 레벨에 저장됩니다. DB2REMOTEPREG를 사용하면 인스턴스 레벨(노드 레벨 포함)에 저장된 레지스트리 변수를 다른 머신으로 재지정할 수 있습니다. DB2REMOTEPREG를 설정하면, DB2 UDB는 DB2REMOTEPREG가 지정하는 머신에서 DB2 UDB 레지스트리 변수에 액세스합니다. 예를 들면, 다음과 같습니다.

```
db2set DB2REMOTEPREG=rmtwkstn
```

여기서, *rmtwkstn*은 원격 워크스테이션 이름입니다.

주: 모든 DB2 인스턴스 프로파일 및 인스턴스 목록은 고유 원격 장치 이름에 위치하므로 이 옵션을 설정할 때에는 주의해야 합니다.

이 기능을 설정값 DBINSTPROF와 함께 사용하여 레지스트리를 담고 있는 동일한 머신의 원격 LAN 드라이브를 가리키도록 할 수 있습니다.

UNIX 시스템에 환경 변수 설정

모든 DB2 고유 레지스트리 변수를 DB2 프로파일 레지스트리에서 정의하는 것이 바람직합니다. DB2 변수가 레지스트리 외부에서 설정되면, 해당 변수를 원격으로 관리할 수 없습니다.

UNIX 운영 체제에서는 시스템 환경 변수인 DB2INSTANCE를 설정해야 합니다.

스크립트 **db2profile**(Korn 셸의 경우) 및 **db2cshrc**(Bourne 셸 또는 C 셸의 경우)가 데이터베이스 환경을 설정하는 데 도움을 주기 위한 예로서 제공됩니다. *insthome/sqllib*에 이 파일이 존재합니다. 여기서, *insthome*은 인스턴스 소유자의 홈 디렉토리입니다.

이 스크립트는 다음 명령문을 포함합니다.

- 다음 디렉토리로 사용자 경로를 갱신합니다.
 - *insthome/sqllib/bin*

- insthome/sqllib/adm
- insthome/sqllib/misc

- DB2INSTANCE를 기본 지역 instance_name으로 설정하여 실행합니다.

주: PATH 및 DB2INSTANCE를 제외한 다른 모든 DB2 지원 변수는 DB2 프로파일 레지스트리에 설정되어야 합니다. DB2에서 지원되지 않는 변수를 설정하려면, 이를 스크립트 파일인 **db2profile** 및 **db2cshrc**에 정의하십시오.

인스턴스 소유자 또는 SYSADM 사용자는 인스턴스의 모든 사용자를 위해 이 스크립트를 사용자 정의할 수 있습니다. 선택적으로, 사용자는 스크립트를 복사하고 사용자 정의할 수 있으며, 스크립트를 직접 호출하거나 스크립트를 .profile 또는 .login 파일에 추가할 수 있습니다.

현재 세션에 대한 환경 변수를 변경하려면, 다음과 유사한 명령을 발행하십시오.

- Korn 셸의 경우,

```
db2instance=inst1
export db2instance
```

- Bourne 셸이나 C 셸의 경우,

```
set db2instance inst1
```

DB2 프로파일 레지스트리가 적절히 관리되도록 하려면, UNIX 운영 체제에서는 다음의 파일 소유권 규칙을 반드시 지켜야 합니다(DB2 관리 서버(DAS)에 대해서는 94 페이지의 『DB2 관리 서버(DAS)』에서 자세한 내용을 참조하십시오).

- DB2 인스턴스 레벨 프로파일 레지스트리 파일은 다음 위치에 배치됩니다.

```
$INSTHOME/sqllib/profile.env
```

이 파일의 액세스 권한 및 소유권은 다음과 같습니다.

```
-rw-r--r-- Instance_Owner DAS_Instance_Group profile.env
```

*\$INSTHOME*은 인스턴스 소유자의 홈 경로입니다.

- DB2 전역 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

- AIX, Solaris, SINIX 및 NUMA-Q(Sequent) 운영 체제용
/var/db2/<version_id>/default.env(여기서 <version_id>은 현재 버전입니다).

- HP-UX 운영 체제용 `/var/opt/db2/<version_id>/default.env` (여기서 `<version_id>`는 현재 버전입니다).

이 파일의 액세스 권한 및 소유권은 다음과 같습니다.

```
-rw-r--r-- DAS_Instance_Owner DAS_Instance_Group default.env
```

전역 레지스트리 변수를 수정하려면, 사용자가 루트나 DAS 인스턴스 소유자로서 로그인되어야 합니다. DB2 관리 서버에 대해서는 『DB2 관리 서버(DAS)』에서 자세한 내용을 참조하십시오.

- DB2 인스턴스 노드 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

```
$INSTHOME/sqllib/nodes/node_number.env
```

디렉토리와 이 파일의 액세스 권한 및 소유권은 다음과 같습니다.

```
drwxrwxr-x Instance_Owner DAS_Instance_Group nodes
```

```
-rw-r--r-- Instance_Owner DAS_Instance_Group node_number.env
```

주: `Instance_Owner`와 `DAS_Instance_Owner`는 모두 `DAS_Instance_Group`의 구성원이어야 합니다.

`$INSTHOME`은 인스턴스 소유자의 홈 경로입니다.

- DB2 인스턴스 프로파일 레지스트리는 다음 위치에 배치됩니다.

- AIX, Solaris, SINIX 및 NUMA-Q(Sequent) 운영 체제용 `/var/db2/<version_id>/profiles.reg`(여기서 `<version_id>`은 현재 버전입니다).

- HP-UX 운영 체제용 `/var/opt/db2/<version_id>/profiles.reg` (여기서 `<version_id>`는 현재 버전입니다).

이 파일의 액세스 권한 및 소유권은 다음과 같습니다.

```
-rw-r--r-- root system profiles.reg
```

DB2 관리 서버(DAS)

DAS(DB2 관리 서버)는 다른 DB2 서버의 관리 작업을 보조하는 데에만 사용되는 특수 DB2 관리 제어점입니다. 클라이언트 구성 지원프로그램 또는 제어 셸

터를 사용하려면, DAS가 수행중이어야 합니다. DAS는 다음과 같은 관리 작업을 수행할 때 제어 센터 및 클라이언트 구성 지원 프로그램을 보조합니다.

- DB2 서버의 원격 관리를 작동시킵니다.
- DB2 및 운영 체제 명령 스크립트의 실행 스케줄을 설정하는 기능을 포함하여 작업 관리 기능을 제공합니다. 이러한 명령 스크립트는 사용자가 정의합니다. 제어 센터는 작업 스케줄을 정의하고, 완성된 작업 결과를 보고, DAS로부터 원격 또는 지역에 위치한 작업에 대한 다른 관리 작업을 수행하는 데 사용됩니다.
- DB2 발견 유틸리티와 함께 DB2 인스턴스, 데이터베이스 및 기타 DB2 관리 서버의 구성에 대한 정보를 발견하는 방법을 제공합니다. 클라이언트 구성 지원 프로그램 및 제어센터는 이 정보를 사용하여 DB2 데이터베이스로의 클라이언트 연결 구성을 간소화하고 자동화합니다.

한 머신에서 하나의 DAS만 가질 수 있습니다. DAS는 운영 체제가 부트될 때 시작할 설치 동안에 구성됩니다.

DAS는 제어 센터 또는 클라이언트 구성 지원 프로그램으로부터 클라이언트 요청을 대신하여 호스트 시스템에서 원격 작업을 수행하는 데 사용됩니다. DAS에 권한 부여된 액세스는 SYSADM 권한이 있는 클라이언트를 요구합니다. 모든 클라이언트는 SYSADM_GROUP 구성 매개변수의 일부가 될 수 있습니다.

요청된 일부 작업을 수행하려면 특정 권한이 필요합니다. DAS는 특정 사용자의 식별자 아래에서 수행합니다. 관리자가 수행하는 작업이나 조작과 연관된 명령에만 사용자가 제한되도록 권한 부여되는 특권. 일반적으로, 필요한 작업 또는 조작에는 다음이 포함됩니다.

- 운영 체제(OS) 구성 정보 조회
- 사용자 및 그룹용 OS 정보 조회
- 다른 DB2 인스턴스를 시작하거나 중지하기 위한 조치
- 스케줄된 작업 실행
- 연결성 및 프로토콜 구성을 위한 정보 수집

DAS 통신 설정에 대해서는 사용자 플랫폼용 빠른 시작에서 자세한 내용을 참조하십시오.

DAS 작성

일반적으로, 설치 프로그램은 DB2 설치중에 인스턴스를 소유하는 머신에서 DAS를 작성합니다. 그러나, 설치 프로그램에서 이를 작성하지 못하면, DAS를 수동으로 작성할 수 있습니다.

DAS와 관련하여 설치 프로세스중 발생하는 사항에 대한 개요로서 다음을 고려하십시오.

- OS/2 또는 Windows NT 플랫폼에서,

지역 관리자 권한이 있는 계정을 사용하여 DAS를 작성할 머신에 로그인합니다. 특정 사용자를 식별해야 하는 경우에는 지역 관리자 권한의 사용자를 작성합니다. db2admin create를 입력하십시오(특정 사용자 계정을 원하는 경우, db2admin create를 발행할 때 『/USER:』 및 『/PASSWORD:』를 사용해야 합니다).

DAS를 작성하는 경우, 사용자 계정 이름과 사용자 암호를 선택적으로 제공할 수 있습니다. 유효하면, 사용자 계정 이름과 암호는 DAS의 소유자를 식별합니다. DAS에 대해 작성된 사용자 ID 또는 계정 이름을 사용자 계정으로 사용하지 마십시오. 계정 이름에 대한 암호를 『절대로 만기되지 않는 암호』로 설정합니다. DAS를 작성한 후에, **db2admin setid** 명령을 사용한 사용자 계정 이름과 사용자 암호를 제공하여 해당 소유권을 설정하거나 변경할 수 있습니다. 이들 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

Windows NT용 DB2 UDB Enterprise - Extended Edition에서, 클라이언트 구성 지원 프로그램 또는 제어 센터를 사용하여 DB2 서버에 대한 연결 구성을 자동화하는 경우, DAS와 같은 머신에 있는 데이터베이스 파티션 서버는 조정자 노드가 됩니다. 즉, 클라이언트에서 데이터베이스로의 모든 물리적 연결은 다른 데이터베이스 파티션 서버로 라우트되기 전에 인스턴스를 소유하는 머신의 데이터베이스 파티션 서버로 라우트됩니다.

Windows NT Enterprise - Extended Edition용 DB2 UDB에서, 추가 관리 서버를 다른 머신에 작성하면, 클라이언트 구성 지원 프로그램 또는 제어 센터에서 DB2 발견을 사용하여 기타 시스템을 조정자 노드로 구성할 수 있습니다. 이렇게 하려면, 다음을 수행하십시오.

1. 지역 관리자 권한이 있는 계정을 사용하여 머신에 로그인하십시오.

2. DAS에서 사용될 지역 관리자 권한이 있는 Windows NT 계정을 작성하십시오. 계정의 사용자 이름은 DB2 이름 지정 규칙을 준수해야 합니다. DAS에 대해 계정을 작성할 때, 다음 사항에 주의하십시오.
 - DAS에 대한 계정을 사용자 계정으로 사용하지 마십시오.
 - 계정에 대한 암호를 암호 사용 기간 제한 없음으로 설정하십시오.
3. 다음 명령을 수행하십시오.

```
db2admin create /user:username
              /password:passwd
```

여기서, *username* 및 *passwd*는 DAS에 대한 사용자 이름과 암호입니다.

- UNIX 플랫폼에서,

1. 사용자에게 루트 권한이 있는지 확인하십시오.
2. 명령 프롬프트에서, DB2 Universal Database 인스턴스 경로의 *instance* 서브디렉토리로부터 다음 명령을 발행하십시오.

```
dasicrt ASName
```

- AIX에서,

```
/usr/lpp/db2_nn_00&/instance/
dasicrt ASName
```

- HP-UX, NUMA-Q(Sequent) 또는 Solaris에서:

```
/opt/IBMDB2/<version_id>/instance/
dasicrt ASName
```

- Linux에서:

```
/usr/IBMDB2/<version_id>/instance/
dasicrt ASName
```

여기서 *ASName*은 관리 서버의 인스턴스 이름이며 *db2_nn_00&* 또는 *<version_id>*는 현재 버전 식별자입니다.

주: NIS 및 NIS+를 수행하는 경우, 다음과 같이 사용자 이름과 그룹 이름을 설정해야 합니다.

- DAS의 1차 그룹은 모든 인스턴스의 2차 그룹에 있어야 합니다.
- DAS의 2차 그룹은 모든 인스턴스의 1차 그룹을 포함해야 합니다.

2차 그룹 목록은 NIS 및 NIS+가 시스템에서 수행되지 않을 경우에만 자동으로 수정됩니다.

사용자 ID는 인스턴스를 하나만 소유할 수 있으므로, 작성하는 각 DB2 관리 서버(DAS)를 소유할 수 있도록 별도의 사용자 ID가 있어야 합니다.

관리 서버를 작성하면, 관리 서버를 사용하여 디렉토리 구조 및 액세스 사용권한을 설정합니다.

DAS 시작 및 중지

DAS를 수동으로 시작하거나 중지시키려면, 먼저 지역 관리자 권한이 있는 계정 또는 사용자 ID를 사용하여 머신에 로그인해야 합니다.

OS/2용 DB2 또는 Windows NT용 DB2에서 작업할 때, 다음을 수행해야 합니다.

- DAS를 시작하려면, db2admin start를 입력하십시오.
- DAS를 중지하려면, db2admin stop을 입력하십시오.

주: Windows NT에서 두 경우 모두, 이러한 명령을 사용하려면 사용자는 SYSADM, SYSCTRL 또는 SYSMOINT 권한이 있어야 합니다.

모든 UNIX 운영 체제의 DB2에서 작업할 때, 다음을 수행해야 합니다.

- DAS를 시작하려면 다음을 수행하십시오.
 1. DAS 소유자로 로그인하십시오.
 2. 다음 중 하나를 사용하여 시동 스크립트를 수행하십시오.

```
. INSTHOME/sql1ib/db2profile    (Bourne 또는 Korn 셸의 경우)
source INSTHOME/sql1ib/db2cshrc (C 셸의 경우)
```

여기서, INSTHOME은 인스턴스의 홈 디렉토리입니다.

3. DAS를 시작하려면 **db2admin** 명령을 사용하십시오.

```
db2admin start
```

주: 각 시스템이 재부팅된 후 DAS는 자동으로 시작됩니다.

- DAS를 중지하려면 다음을 수행하십시오.
 1. DAS 소유자로 로그인하십시오.

2. 다음 중 하나를 사용하여 시동 스크립트를 수행하십시오.

```
. INSTHOME/sql1lib/db2profile ( Bourne 또는 Korn 셸의 경우 )  
source INSTHOME/sql1lib/db2cshrc ( C 셸의 경우 )
```

여기서, INSTHOME은 인스턴스의 홈 디렉토리입니다.

3. 다음과 같이 **db2admin** 명령을 사용하여 DAS를 중지하십시오.

```
db2admin stop
```

주: UNIX에서 두 경우 모두, 이 명령을 사용하는 사용자는 DAS 소유자의 권한 부여 ID로 로그인해야 합니다.

DAS 나열

머신에서 DAS 인스턴스 이름을 확보하려면, 다음을 사용하십시오.

```
db2admin
```

이 명령은 DAS를 시작하거나 중지하고, 새로운 사용자 및 암호를 작성하고, DAS 인스턴스를 삭제하고, DAS 인스턴스와 연관된 사용자 계정을 설정하거나 수정하는 데 사용됩니다.

DAS 구성

DAS와 관련된 관리 구성 매개변수용 현재 값을 보려면, 다음을 입력하십시오.

```
db2 get admin cfg
```

여기에서는 제품 설치중에 기본값으로 주어진 현재 값이나 이전에 구성 매개변수를 갱신할 때 주어진 값을 보여줍니다.

DAS와 관련된 데이터베이스 관리 프로그램 구성 파일의 개별 항목을 갱신하려면, 다음을 입력하십시오.

```
db2 update admin cfg using ...
```

수정될 수 있는 데이터베이스 관리 프로그램 구성 매개변수에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

권장된 데이터베이스 관리 프로그램 기본값에 대한 구성 매개변수를 재설정하려면, 다음을 입력하십시오.

```
db2 reset admin cfg
```

데이터베이스 관리 프로그램 구성 파일에 대한 변경사항은 변경사항이 메모리에 로드된 후에만 적용됩니다(즉, db2admin stop 다음에 db2admin start가 오거나 또는 Windows NT 플랫폼의 경우 서비스를 중지시키고 시작할 때).

DAS에 대한 통신 프로토콜을 설정하려면, 사용자의 플랫폼에 대한 빠른 시작에서 자세한 내용을 참조하십시오.

DAS의 보안 고려사항

먼저 지역 관리자 권한이 있는 계정 또는 사용자 ID를 사용하여 머신에 로그인해야 합니다.

주: 일부 필요한 액세스 권한이 로그인 계정에 대해 설정되지 않으므로, Windows NT에서 제어판의 서비스 유틸리티를 사용하여 DAS용 로그인 계정을 변경하지 말아야 합니다. 항상 **db2admin** 명령을 사용하여 DB2 관리 서버(DAS)에 대한 로그인 계정을 설정하거나 변경하십시오.

DAS를 작성한 후 다음과 같이 **db2admin** 명령을 사용하여 로그인 계정을 설정하거나 변경하십시오.

```
db2admin setid username password
```

여기서, *username* 및 *password*는 지역 관리자 권한이 있는 계정의 사용자 이름과 암호입니다.

사용자 ID 또는 사용자 이름은 환경에서 필요에 따라 기타 인스턴스를 시작하거나 중지시킬 수 있도록 각 서버에 대해 SYSADM 권한을 갖는것이 바람직합니다.

DAS 갱신

UNIX 운영 체제에서 프로그램 임시 수정(PTF) 또는 코드 패치를 설치하여 DB2가 갱신되면, 모든 DB2 관리 서버(DAS)뿐만 아니라 기존의 모든 인스턴스가 갱신되어야 합니다. DAS를 갱신하려면, 설치된 DB2 버전 및 릴리스에 대해 고유한 서브디렉토리에 있는 instance 서브디렉토리에서 사용 가능한 **dasiupdt** 명령을 사용하십시오.

우선 계정을 사용하여 『루트』(UNIX에서)로서 머신에 로그인하거나 지역 관리자 권한을 가진 사용자 ID로서 머신에 로그인해야 합니다.

명령은 다음과 같습니다.

```
dasiupdt InstName
```

InstName은 인스턴스 소유자의 로그인 이름입니다. 이 명령에 대해 InstName 앞에 오며 공백으로 구분될 수 있는 선택적 매개변수도 있습니다.

- -h 또는 -?

이 명령에 대한 도움말 메뉴를 표시합니다.

- -d

문제점 분석에 사용되는 디버그 모드를 설정합니다.

DAS 제거

우선 『root』(UNIX에서)로서 또는 지역 관리자 권한을 가진 사용자 ID로 머신에 로그인해야 합니다.

DAS를 제거하려면, 다음 단계를 수행하십시오.

- OS/2 또는 Windows NT 운영 체제에서,
 1. db2admin stop을 사용하여 DAS를 중지시키십시오.
 2. sqllib 서브디렉토리에 있는 db2das00 서브디렉토리에서 모든 파일을 백업(필요하면)하십시오. 인스턴스 디렉토리는 *DB2INSTPROF* 레지스트리 변수로 표시됩니다.

주: 이 예에서는 db2das00을 제거될 DAS의 이름으로 간주합니다.

3. db2admin drop을 사용하여 DAS를 삭제하십시오.

주: Windows NT에서 이 명령을 사용하는 사용자는 SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다.

- UNIX 운영 체제에서,

1. DAS 소유자로 로그인하십시오.

2. 다음 중 하나를 사용하여 시동 스크립트를 수행하십시오.

```
. INSTHOME/sqllib/db2profile (Bourne 또는 Korn 셸의 경우)  
source INSTHOME/sqllib/db2cshrc (C 셸의 경우)
```

여기서, INSTHOME은 인스턴스의 홈 디렉토리입니다.

3. 다음과 같이 **db2admin** 명령을 사용하여 DAS를 중지하십시오.

```
db2admin stop
```

4. DAS의 홈 디렉토리에 있는 **sqllib** 서브디렉토리에 있는 모든 파일을 백업 (필요하면)하십시오. 인스턴스 디렉토리는 **DB2INSTPROF** 레지스트리 변수로 표시됩니다.

5. 로그오프하십시오.

6. **root**로 로그인한 후 다음과 같이 **dasidrop** 명령을 사용하여 DAS를 제거하십시오.

```
dasidrop ASName
```

여기서, **ASName**은 관리 서버의 인스턴스 이름입니다. 이 명령은 설치된 DB2 버전 및 릴리스에 대해 고유한 서브디렉토리에 있는 **instance** 서브디렉토리에 있습니다.

주: **dasidrop** 명령은 DB2 관리 서버(DAS)의 홈 디렉토리에 있는 **sqllib** 디렉토리를 제거합니다.

EEE 시스템으로 DAS 설정

다음 정보는 제어 센터를 사용하여 원격관리용으로 DB2 EEE 서버(Solaris, NT, Sequent, HP-UX, 및 AIX)를 구성하는 데 필요한 단계입니다.

설치중에 설치 프로그램은 인스턴스를 소유하는 머신에서 단일 DAS를 작성합니다. 제어 센터 또는 클라이언트 구성 지원 프로그램이 기타 조정자 노드에 액세스할 수 있도록 다른 머신에서 추가 DAS를 작성할 수 있습니다. 조정자 노드로 작업할 때의 오버헤드는 인스턴스에서 둘 이상의 노드에 파티션될 수 있습니다.

조정자 함수를 분배하려면, 다음을 수행하십시오.

1. 파티션된 데이터베이스 시스템에서 선택된 추가 머신에 새 DAS를 작성하십시오.
2. 각 DAS를 제어 센터 또는 클라이언트 구성 지원에서 별도의 시스템으로 카탈로그화하십시오.
3. 각 새 시스템에서 동일한 인스턴스를 카탈로그화한 후, DAS를 카탈로그화할 때 사용했던 동일한 머신 이름을 매번 지정하십시오.

구성에 대해 두 가지 양상이 있습니다. 즉, 하나는 DB2 관리 서버(DAS)에 필요하고 또 하나는 DB2 인스턴스로 관리되는 대상에 권장되는 것입니다. 다음 세 절에서는 각 절마다 두 가지 구성 주제 중 하나씩을 다룹니다. 각 구성 주제 앞에는 가정된 환경을 설명하는 절이 옵니다.

환경 예:

제품/버전:

DB2 UDB EEE V7.1

설치 경로:

install_path

TCP 서비스 파일:

tcp_services_file

DB2 인스턴스:

이름: db2inst

소유자 ID:

db2inst

인스턴스 경로:

instance_path

노드: 3 노드, db2nodes.cfg:

- 0 hostA 0 hostA0switch
- 1 hostA 1 hostA1switch
- 2 hostB 0 hostBswitch

DB 이름:

db2instDB

DAS:

이름: db2as

소유자/사용자 ID:

db2as

인스턴스 경로:

das_path

호스트 설치/수행:

hostA

인터노드 통신 포트:

16000(hostA 및 hostB용으로 사용되지 않는 포트)

주: 사이트 고유 값을 위의 필드로 대체하십시오. 예를 들어, 다음 표에는 지원되는 각 EEE 플랫폼에 대한 경로 이름 예가 들어 있습니다.

표 1. 지원되는 EEE 플랫폼의 예 경로 이름

경로	AIX용 DB2 UDB EEE	Solaris용 DB2 UDB EEE	Windows NT용 DB2 UDB EEE
<i>install_path</i>	/usr/lpp/<v_r_ID>	/opt/IBMDB2/<v_r_ID>	C:\sqllib
<i>instance_path</i>	/home/db2inst/sqllib	/home/db2inst/sqllib	C:\profiles\db2inst
<i>das_path</i>	/home/db2as/sqllib	/home/db2as/sqllib	C:\profiles\db2as
<i>tcp_services_file</i>	/etc/services	/etc/services	C:\winnt\system32 \drivers\etc\services

테이블에서, <v_r_ID>는 플랫폼 특유의 버전이며 릴리스 식별자입니다. 예를 들어 AIX 버전 5.2용 DB2 UDB EEE의 경우, <v_r_ID>는 db2_05_00입니다.

DB2 UDB EEE를 설치할 때, 설치 프로그램은 인스턴스를 소유하는 머신에서 DAS를 작성합니다. 데이터베이스 파티션 서버는 DAS와 같은 머신에 있으며 인스턴스에 대한 연결 지점이 됩니다. 즉, 이 데이터베이스 파티션 서버는 제어 센터 또는 클라이언트 구성지원 프로그램에서 인스턴스에 발행된 요청에 대한 조정자 노드입니다.

DAS 구성: DAS는 제어 센터 대신 특정 작업을 수행하는 관리 제어점입니다. 물리 머신당 최대 하나의 DAS가 있을 수 있습니다. 몇몇 머신으로 구성되는 EEE 인스턴스의 경우, 적어도 하나의 머신이 DAS를 실행하여 제어 센터가 EEE 인스턴스를 관리할 수 있도록 해야 합니다. DAS(db2as)는 대상 DB2 인스턴스(db2inst)의 상위로서 제어 센터 네비게이터에 있는 시스템을 『나타냅니다』.

예를 들어, db2inst는 두 개의 물리 머신 또는 호스트에 분배되는 세개의 노드로 구성됩니다. db2das를 hostA 또는 hostB에서 실행하여 최소한의 요구사항을 충족시킬 수 있습니다.

주:

1. 호스트 A에 있는 파티션의 수는 해당 호스트에서 수행될 수 있는 DAS의 수와는 상관이 없습니다. 해당 호스트의 다중 논리 노드(MLN) 구성과 상관없이 hostA에서 하나의 db2as 사본만을 실행할 수 있습니다.
2. 모든 호스트에서 DAS ID db2as를 작성해야 하는 것은 아닙니다. 이것이 실행되는 호스트에만 있으면 됩니다. 또한, DAS ID의 홈 디렉토리를 모든 호스트에 마운트할 필요는 없습니다. 특히 이 예의 경우, ID db2as는 hostA에는 존재해야 하지만, hostB에는 필요하지 않으며, db2as의 홈 디렉토리를 hostB에 마운트할 필요가 없습니다.

DAS와의 제어 센터 통신: 서비스 포트: 제어 센터는 523 TCP 서비스 포트를 사용하여 DAS와 통신합니다. 이 포트는 DB2 UDB 전용으로 예약되어 있기 때문에, *tcp_services_file*에 새 항목을 삽입할 필요는 없습니다.

인터노드 관리 통신: 서비스 포트: 일부 관리 TASK의 경우, DAS는 모든 노드와의 통신을 설정해야 합니다. 이를 위해, 인스턴스에 참여하는 각 호스트의 *tcp_services_file*에 이름 지정된 TCP 포트를 정의해야 합니다.

주: Windows NT EEE는 *tcp_services_file*에 TCP 포트 항목을 추가하려고 시도합니다.

예를 들어, db2inst가 두 호스트인 hostA와 hostB에 정의되어 있습니다. 103 페이지의 『환경 예』에 지정된 대로 16000 포트는 두 호스트에서 사용되지 않습니다. 그러므로, hostA와 hostB 모두의 *tcp_services_file*에 다음 행을 삽입해야 합니다.

```
db2ccmsrv 16000/tcp
```

db2ccmsrv 포트 이름의 철자는 위와 정확히 일치해야 하며, 선택된 동일한 포트 번호가 모든 호스트에 사용되어야 합니다.

인터노드 관리 통신: UNIX DB2 EEE 서버: 일단 TCP 포트 행이 hostA 및 hostB의 *tcp_services_file*에 삽입되면, 인스턴스에 참여하는 모든 호스트에서 관리 대기자 프로세스 또는 디먼, db2cclst를 시작해야 합니다. 명령행에서 수동으로 하거나 또는 시스템이 부트할 때마다 자동으로 db2cclst를 호출하도록 시스템을 구성할 수 있습니다.

수동: 관리할 인스턴스의 ID db2inst에서 다음 hostA 또는 hostB의 명령을 호출하십시오.

```
rah 'install_path/bin/db2cclst'
```

예를 들어, AIX에서 이 명령 호출은 다음과 같습니다.

```
rah '/usr/lpp/<v_r_ID>/bin/db2cclst'
```

rah 명령은 버전 및 릴리스 서브디렉토리의 instance 서브디렉토리에서 찾을 수 있습니다. 버전 및 릴리스 서브디렉토리의 정확한 이름은 운영 체제마다 다릅니다. instance는 사용하려는 인스턴스의 홈 디렉토리입니다.

이 경우에, <v_r_ID>는 플랫폼 특유의 버전이며 릴리스 식별자입니다. 예를 들어 AIX 버전 5.2용 DB2 UDB EEE의 경우, <v_r_ID>는 db2_05_00입니다.

자동: 수퍼유저 특권(루트 등)을 갖는 ID로 다음 명령을 hostA와 hostB에서 실행하십시오.

```
mkitab "db2cclst::once:su - db2inst -c install_path /bin/db2cclst"
```

예를 들어, AIX에서 이 명령 호출은 다음과 같습니다.

```
mkitab "db2cclst::once:su - db2inst -c install_path  
/usr/lpp/<v_r_ID>/bin/db2cclst"
```

머신이 부팅될 때마다 사용자의 조작 없이도 db2cclst가 호출됩니다.

테이블에서, <v_r_ID>는 플랫폼 특유의 버전이며 릴리스 식별자입니다. 예를 들어 AIX 버전 5.2용 DB2 UDB EEE의 경우, <v_r_ID>는 db2_05_00입니다.

대기자 디먼이 각 호스트에서 사용중인지를 확인하기 위해 인스턴스 ID db2inst에서 다음 명령을 호출할 수 있습니다.

```
rah 'ps -ef | grep db2cc1st'
```

db2cc1st 프로세스가 각 호스트에서 수행되고 있지 않을 경우 각 호스트에서 다음 행을 /etc/syslog.conf에 추가하여 추가 진단 정보를 얻을 수 있습니다.

```
*.info /tmp/db2/user.info
```

여기서, /tmp/db2/user.info 파일은 보다 적절한 파일로 대체할 수 있습니다.

주: 이 파일은 반드시 있어야 하며 변경이 이루어진 후 해당 구성 파일을 다시 읽도록 SYSLOG 디먼에 요청해야 합니다.

```
kill -1 <syslogd PID>
```

여기서, 다음을 실행하여 syslogd PID를 확보할 수 있습니다.

```
ps -ef | grep syslogd
```

그런 다음, 위에서 설명한 대로 대기자를 수동으로 호출한 후, 실패 호스트의 syslogd 파일 /tmp/db2/user.info에서 db2cc1st가 생성한 오류 메시지를 볼 수 있습니다.

인터노드 관리 통신: Windows NT DB2 EEE 서버: DB2 원격 명령 서비스 (db2rcmd.exe)는 인터노드 관리 통신을 자동으로 조절합니다. 장애가 발생하는 경우, Windows NT 레지스트리는 진단 정보를 수록합니다.

보안: DAS가 인스턴스에 대해 몇몇 관리 타스크를 수행하려면, 충분한 권한을 가지고 있어야 합니다. 특히, DAS는 대상인 관리 인스턴스의 시스템 관리자 (SYSADM)여야 합니다.

DAS가 관리할 모든 DB2 인스턴스에 대한 권한을 DAS에 권한 부여해야 합니다. DAS와 동일한 머신에 설치된 인스턴스가 후보가 됩니다. DB2 EEE 인스턴스의 경우, 위에서 설명한 사항이 적용되려면 DAS와 동일한 머신에 적어도 하나의 데이터베이스 파티션 서버가 존재해야 합니다.

예를 들어 UNIX에서, db2inst를 관리하는 데 필요한 권한을 db2as에게 권한 부여하는 한 가지 방법은 db2inst 및 db2as의 1차 그룹이 동일한지 확인하는 것입니다. 아니면, db2inst의 1차 그룹을 db2as의 2차 그룹으로 만들고, db2as의 1차

그룹을 db2inst의 2차 그룹으로 만드는 것으로 충분합니다. 마지막으로, 또다른 옵션은 db2inst의 SYSADM_GROUP 데이터베이스 관리 구성 매개변수를 db2as의 1차 그룹으로 설정하는 것입니다.

Windows NT에서, db2as는 hostA와 hostB의 지역 관리자 그룹의 구성원이어야 합니다. db2as ID를 작성하고 이를 두 호스트의 지역 관리자 그룹에 추가하는 옵션 이외의 다른 옵션은 db2as의 도메인 ID를 작성하고 이 도메인 ID를 각 호스트의 지역 관리자 그룹에 추가하는 것입니다.

환경: DAS 설치시 올바른 조작에 필요한 특정 레지스트리 변수를 구성해야 합니다. 이들 변수에 대한 현재 값을 확인하려면, 다음 명령을 DB2 인스턴스 ID db2inst나 또는 DAS ID db2das에서 수행하십시오.

```
db2set -g
```

적어도 다음 값을 가진 다음 매개변수를 정의해야 합니다.

```
DB2SYSTEM=hostA  
DB2ADMINSERVER=db2as
```

또한, 제어 센터로부터 DAS와 통신하려면, DB2COMM 레지스트리 변수가 『TCPIP』로 설정되어 있는지 확인하십시오. 이 설정을 확인하려면, 다음 명령을 DAS ID db2as에서 실행한 후 전역(-g) 및 인스턴스(-i) 레벨에서 점검하십시오 (하나만 설정하면 됨).

```
db2set -all
```

동일한 행에서, 제어 센터와 db2inst간에 통신을 가능하게 하려면 db2inst ID에서 다음 명령을 실행하여 DB2COMM 매개변수가 DB2 인스턴스용 『TCPIP』로 설정되어 있는지 확인하십시오.

```
db2set -all
```

DAS에 대한 이 매개변수를 수정한 경우, 변경사항이 유효해지려면 DAS를 재시작해야 합니다. 또한 DB2 인스턴스에 대해 이 매개변수가 변경된 경우, DB2 인스턴스를 재시작해야 합니다. db2inst의 경우, db2stop을 db2start 다음에 실행하고 DAS에 대해서는 db2admin stop 및 db2admin start를 실행하십시오.

관리 서버, 인스턴스 및 데이터베이스의 발견: 알려진 발견(Known Discovery)을 통해 사용자는 클라이언트에 알려진 시스템상의 인스턴스와 데이터베이스를 발견할 수 있으며, 인스턴스와 데이터베이스를 발견할 수 있도록 새 시스템을 추가할 수 있습니다. 검색 발견(Search Discovery)은 알려진 발견의 모든 기능을 제공하고 옵션을 추가하여 지역 네트워크에서 기타 DB2 서버를 검색할 수 있도록 합니다.

서버가 알려진 발견을 지원하도록 하려면, DAS 구성 파일에 있는 *discover* 매개변수를 KNOWN으로 설정하십시오. 검색 발견을 지원하게 하려면, 이 매개변수를 SEARCH로 설정하십시오. 서버, 모든 인스턴스 및 데이터베이스를 발견하지 못하도록 막으려면, 이 매개변수를 DISABLE로 설정하십시오.

주: 검색 발견으로 클라이언트에 리턴된 TCP/IP 호스트 이름은 **hostname** 명령을 입력할 때 DB2 서버 시스템으로 리턴된 것과 같은 호스트 이름입니다. 클라이언트에서 이 호스트 이름이 맵되는 IP 주소는 클라이언트 머신에서 구성된 TCP/IP 도메인 이름 서버(DNS) 또는 DNS가 구성되지 않은 경우, 클라이언트의 hosts 파일에 있는 맵핑 항목으로 판별됩니다. DB2 서버 시스템에 다중 어댑터 카드가 구성된 경우, 서버에 TCP/IP가 구성되어 있어야만 올바른 호스트 이름을 리턴할 수 있으며, DNS 또는 지역 클라이언트의 hosts 파일이 호스트 이름을 원하는 IP 주소에 맵할 수 있습니다.

클라이언트에서 발견은 *discover* 매개변수를 사용하여 작동되기도 합니다. 그러나 이 경우, *discover* 매개변수는 다음과 같이 클라이언트 인스턴스(또는 클라이언트로서 동작하는 서버)에 설정됩니다.

- **KNOWN**

클라이언트 구성 지원이 알려진 목록에서 시스템을 새로 고치고 시스템 추가 누름 버튼을 사용하여 새 시스템을 목록에 추가할 수 있게 합니다. *discover* 매개변수가 KNOWN으로 설정되면, 클라이언트 구성 지원은 네트워크를 검색할 수 없게 됩니다.

- **SEARCH**

알려진 발견의 모든 기능을 작동시키고 네트워크 검색도 작동시킵니다.

이것이 선택되면 『기타 시스템(네트워크 검색)』 아이콘만이 표시됩니다. 이는 기본입니다.

• **DISABLE**

발견을 작동금지시킵니다. 이 경우, 네트워크 검색 옵션을 『데이터베이스 Wizard 추가』에서 사용할 수 없습니다.

주: *discover* 매개변수는 모든 클라이언트와 서버 인스턴스에서 기본값이 SEARCH가 됩니다. *discover* 매개변수는 모든 DB2 관리 서버(DAS)에서 기본값이 SEARCH가 되지만, 단 DAS가 UNIX Enterprise - Extended Edition 환경에 설치된 경우에는 *discover*의 기본값은 KNOWN이 됩니다.

검색 발견에 대한 추가 설정: 검색 발견에서는 *discover_comm* 매개변수를 서버(DB2 관리 서버의 구성 파일에서) 및 클라이언트(데이터베이스 관리 프로그램 구성 파일에서) 모두에 설정해야 합니다.

discover_comm 매개변수는 서버가 클라이언트에서 검색 요청을 청취하게 될 통신 프로토콜을 제어하는 데 사용되고, 해당 클라이언트가 검색 요청을 전송할 때에도 사용됩니다. *discover_comm* 매개변수는 TCP/IP 또는 NetBIOS로 설정될 수 있습니다. 현재는 이러한 프로토콜만 지원됩니다.

DAS에서 *discover_comm*에 지정된 값은 *db2comm*에 설정된 값과 같거나 부속 집합이어야 합니다.

주: 제어 센터 및 클라이언트 구성 지원과의 문제점을 피하려면, DB2COMM 레지스트리 변수가 **db2set** 명령을 사용하여 DB2 레지스트리에 설정되어야 합니다. DB2COMM 레지스트리를 설정하기 위해 다른 방식을 사용하는 것은 바람직하지 않습니다.

서버에서 *discover_comm* 매개변수는 DAS 구성 파일에 설정됩니다. 클라이언트(또는 클라이언트로 동작하는 서버)에서 *discover_comm*은 데이터베이스 관리 프로그램 구성 파일에 설정됩니다.

주: 검색 발견을 사용할 때 클라이언트에서 *discover_comm* 매개변수로 지정된 최소한 하나의 프로토콜은 DAS에서 *discover_comm* 매개변수로 지정된 것과 일치해야 합니다. 일치하는 것이 없으면, 서버는 클라이언트의 요청에 응답하지 않습니다.

DB2COMM 레지스트리 변수의 설정을 점검하려면, 다음을 입력하십시오.

db2set db2comm

그리고, 두 개의 DB2 프로파일 레지스트리 변수 DB2DISCOVERYTIME 및 DB2NBDISCOVERYRECVBUFS는 클라이언트에서 NetBIOS를 통해 검색 발견을 조정하기 위해 사용될 수 있습니다. 기본값은 대부분의 경우에 레지스트리 변수에 적합해야 합니다.

DB2DISCOVERYTIME 및 DB2NBDISCOVERRCVBUFS 프로파일 레지스트리 변수는 클라이언트 인스턴스(또는 클라이언트로 동작하는 서버)에서 설정됩니다. 레지스트리 변수를 다음과 같이 설정하십시오.

- DB2DISCOVERYTIME 레지스트리 값을 60초로 설정하려면, 다음 명령을 입력하십시오.

```
db2set db2discoverytime=60
```

이는 검색 발견이 서버에서 응답을 받을 때까지 60초를 기다려야 한다고 지정합니다.

- DB2NBDISCOVERRCVBUFS 레지스트리 값을 20으로 설정하려면, 다음 명령을 입력하십시오.

```
db2set db2nbdiscoverrcvbufs=20
```

이는 발견된 서버에서 동시 응답 메시지에 대해 할당될 NetBIOS 버퍼의 수를 지정합니다.

발견에서 서버 인스턴스 및 데이터베이스 숨기기: 서버에 다중 인스턴스가 있을 수 있으며, 이들 인스턴스에 다중 데이터베이스가 있을 수 있습니다. 이 중 일부를 발견 프로세스가 찾지 못하도록 숨길 수 있습니다.

클라이언트가 시스템에서 서버 인스턴스를 발견하도록 하려면, 시스템의 각 서버 인스턴스에 있는 *discover_inst* 데이터베이스 관리 프로그램 구성 매개변수를 ENABLE(기본값)로 설정하십시오. 이 매개변수를 DISABLE로 설정하면 이 인스턴스와 데이터베이스를 발견에서 숨길 수 있습니다.

클라이언트가 데이터베이스를 발견하도록 하려면, *discover_db* 데이터베이스 구성 매개변수를 ENABLE(기본값)로 설정하십시오. 이 매개변수를 DISABLE로 설정하면 데이터베이스를 발견에서 숨길 수 있습니다.

발견 매개변수 설정: *discover* 및 *discover_comm* 매개변수는 서버 시스템의 DAS 구성 파일과 클라이언트의 데이터베이스 관리 프로그램 구성 파일에서 설정됩니다. 매개변수를 다음과 같이 설정하십시오.

- DAS에서,

다음 명령 프로세스를 사용하여 DAS 구성 파일을 갱신하십시오.

```
update admin cfg using discover [ DISABLE | KNOWN |  
SEARCH ]  
update admin cfg using discover_comm [ NETBIOS | TCPIP ]
```

다음 명령을 입력하여 DAS를 중지시킨 후 재시작하십시오.

```
db2admin stop  
db2admin start
```

주: 검색 발견은 NetBIOS 및 TCP/IP에서만 작동합니다.

- 제어 센터의 사용

1. 클라이언트 구성 지원 시작
2. 클라이언트 설정 누름 버튼을 클릭하십시오.
3. 통신 탭을 선택하십시오.
4. 수정할 매개변수를 매개변수 창에서 선택하십시오.
5. 수정할 매개변수의 값을 값 상자에서 선택합니다.
6. 확인 누름 버튼을 클릭하여 클라이언트 설정 창을 닫으십시오. DB2 메시지 창이 열립니다.
7. 확인 누름 버튼을 클릭한 후 응용프로그램을 재시작하여 변경사항을 적용하십시오.

주: *discover_comm*에 NETBIOS가 포함되면, 워크스테이션 이름(*nname*) 매개변수가 클라이언트와 DAS 모두에 대해 설정되어야 합니다. 또한, DB2NBADAPTERS 레지스트리 변수는 사용자가 사용할 어댑터 번호로 설정되어야 합니다.

제어 센터를 사용하여 *discover_inst* 및 *discover_db* 매개변수를 설정하십시오.

1. 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 인스턴스를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 구성을 선택하십시오.
3. 『환경』 페이지에서, *discover_inst* 매개변수를 선택하십시오.
4. 클라이언트에서 서버 인스턴스를 발견하려면, 사용을 선택하고 확인을 클릭하십시오.
5. 오브젝트 트리에서 데이터베이스를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 구성을 선택하십시오.
6. 『환경』 페이지에서, *discover_db* 매개변수를 선택하십시오.
7. 데이터베이스가 클라이언트에서 발견되지 하려면, 사용을 선택하고 확인을 클릭하십시오.

명령행을 사용하여 다중 인스턴스를 수행하십시오.

- 다음을 입력하여 DB2INSTANCE 변수를 원하는 다른 인스턴스의 이름으로 설정하십시오.

```
set db2instance=<another_instName>
```

- **db2start** 명령을 입력하여 인스턴스를 시작하십시오.

DAS를 설정하여 클라이언트 구성 지원 및 제어 센터 사용

네트워크에서 시스템에 대한 정보를 검색하려면 DB2 발견을 구성해야 합니다. DB2 발견은 클라이언트 구성 지원 프로그램 및 제어 센터에서 사용되는 기능입니다. 이 기능에 대해 구성할 때에는 인스턴스 목록 및 DB2 관리 서버(DAS) 구성을 갱신하여 DB2 발견이 올바른 정보를 검색하도록 해야 합니다.

인스턴스 목록 갱신: DB2 관리 서버(DAS)는 인스턴스가 초기에 작성될 때 인스턴스를 소유하는 머신상의 DAS만 인스턴스에 대해 알고 있으므로, 파티션된 데이터베이스 시스템의 모든 인스턴스에 대해 알지 못할 수도 있습니다.

DAS가 없는 머신에서 인스턴스를 작성하는 경우, 이 머신에 DAS를 작성하여 인스턴스를 알릴 수 있습니다.

둘 이상의 DAS를 작성하고 각 DAS가 파티션된 데이터베이스 시스템에 있는 모든 인스턴스에 대해 알고 있게 하려면, 다음 단계를 수행하십시오.

1. 각 DAS의 경우

db2ilist 명령을 관리 서버 머신에서 수행하여 이 DAS에 알려진 인스턴스의 목록을 표시하십시오.

주: 인스턴스 목록이 완료되면, 나머지 단계를 수행하지 않고 다음 섹션으로 넘어갈 수 있습니다.

2. 이전 단계에서 인스턴스 목록에 없는 각 인스턴스의 경우

인스턴스를 소유하는 머신에서 **db2nlist** 명령을 수행하여 DAS가 있는 머신에 대한 항목이 있는지 보십시오. 없을 경우, **db2ncrt** 명령을 수행하려면 이 머신을 인스턴스에 추가해야 합니다.

주: 인스턴스에 대한 네트워크 공유 드라이브를 DAS 머신에서 사용할 수 있어야 합니다.

DAS 구성 갱신

기본적으로, 설치 프로그램은 DB2SYSTEM 레지스트리 변수를 Windows NT 컴퓨터 이름으로 설정합니다. 발견으로 검색되는 시스템 이름은 DB2 관리 서버(DAS)가 상주하는 시스템입니다. 발견은 연결이 이루어질 때 이들 시스템을 조정자 노드로 사용합니다.

DAS 구성을 갱신하는 데에는 두 가지 방법이 있습니다.

- DB2 시스템 목록에서 조정자 노드를 선택하려면, DISCOVER=SEARCH(기본값)를 각 DB2 관리 서버의 구성 파일에 설정하십시오.

다중 DAS가 있을 때에는 동일한 인스턴스가 클라이언트 구성 지원 또는 제어 센터 인터페이스의 여러 시스템에 나타날 수 있습니다. 그러나, 시스템마다 인스턴스에 대한 다른 통신 액세스 경로가 있습니다. 사용자는 통신에 대해 다른 DB2 시스템을 조정자 노드로 선택할 수 있으므로 작업 로드를 재분배할 수 있습니다.

- 사용자가 조정자 노드를 선택할 수 없게 하려면, DISCOVER=SEARCH를 DAS 구성에 있는 하나의 DAS에만 설정하고 DISCOVER=KNOWN을 모든 DAS에 설정하십시오. 발견은 연결이 이루어질 때 DAS가 조정자 노드로 상주하는 데이터베이스 파티션 서버를 사용합니다.

노드 구성 파일 작성

사용자의 데이터베이스가 파티션된 데이터베이스 환경에서 작동하려면, db2nodes.cfg라는 노드 구성 파일을 작성해야 합니다. 이 파일은 인스턴스용 홈 디렉토리의 sqllib 서브디렉토리에 먼저 배치되어야 다중 파티션에 병렬 성능을 사용하는 데이터베이스 관리 프로그램을 시작할 수 있습니다. 파일에는 인스턴스의 모든 데이터베이스 파티션에 대한 구성 정보가 수록되어 있으며, 해당 인스턴스의 모든 데이터베이스 파티션이 이 파일을 공유합니다.

Windows NT 고려사항: Windows NT에서 DB2 Enterprise - Extended Edition 을 사용하는 경우, 인스턴스를 작성할 때 노드 구성 파일이 작성됩니다.

주: 인스턴스가 삭제될 때 데이터의 손실이 발생하지 않게 하려면, DB2가 작성한 것이 아닌 다른 sqllib 서브디렉토리에 파일 또는 디렉토리를 작성하지 않아야 합니다. 여기에는 두 가지 예외사항이 있습니다. 시스템이 저장 프로시저어를 지원할 경우, 저장 프로시저어 응용프로그램을 sqllib 서브디렉토리하의 function 서브디렉토리에 배치하십시오(저장 프로시저어에 대해서는 관리 안내서: 성능의 『저장 프로시저어』를 참조하십시오). 또다른 예외사항은 사용자 정의 구별 함수(UDF)가 작성된 경우입니다. UDF 실행 파일은 동일한 디렉토리에 와야 합니다.

파일은 인스턴스에 속하는 각 데이터베이스 파티션당 하나의 행을 수록합니다. 각 행의 형식은 다음과 같습니다.

```
nodenum hostname [logical-port [netname]]
```

토큰은 공백에 의해 구분됩니다. 변수는 다음과 같습니다.

nodenum

노드 번호는 0부터 99까지의 수이며 노드를 고유하게 정의합니다. 노드 번호는 오름차순으로 되어 있어야 합니다. 순서에는 간격이 있을 수 있습니다.

일단 노드 번호가 지정되면, 이는 변경될 수 없습니다(그렇지 않으면, 데이터의 파티션 방식을 지정하는 파티션 맵의 정보가 절충됩니다).

노드를 삭제할 경우, 추가한 새로운 노드에 대해 해당 노드 번호를 다시 사용할 수 있습니다.

노드 번호는 데이터베이스 디렉토리의 노드 이름을 작성하는 데 사용됩니다. 형식은 다음과 같습니다.

`NODEnnnn`

`nnnn`은 노드 번호입니다. 이 노드 번호는 `CREATE DATABASE` 및 `DROP DATABASE` 명령에 의해서도 사용됩니다.

hostname

파티션간 통신용 IP 주소의 호스트 이름(`netname`이 지정될 경우 예외가 발생합니다. 이 경우, 대부분의 통신에 `netname`이 사용되고, `hostname`은 `DB2START`, `DB2STOP` 및 `db2_a11`에만 사용됩니다.)

logical-port

이 매개변수는 선택적이며, 노드용 논리 포트 번호를 지정합니다. 이 번호는 데이터베이스 관리 프로그램 인스턴스 이름과 함께 사용되어, `etc/services` 파일의 TCP/IP 서비스 이름 항목을 식별합니다.

IP 주소와 논리 포트를 결합하면 잘 알려진 주소로 사용할 수 있고, 노드 간 통신 연결을 지원하는 모든 응용프로그램에서 고유합니다.

각 `hostname`에 대해 하나의 `logical-port`는 0 또는 공백이어야 합니다(기본값은 0). 이 `logical-port`와 연관된 노드는 클라이언트가 연결될 호스트의 기본 노드입니다. `db2profile` 스크립트에서 또는 `sqlsetc()` API를 사용하여 `DB2NODE` 환경 변수로 대체할 수 있습니다.

다중 노드를 같은 호스트에 가지고 있을 경우(즉, 하나의 호스트에 하나 이상의 `nodenum`이 있을 경우), `logical-port` 번호를 논리 노드에 0에서부터 공백 없이 오름차순으로 할당해야 합니다.

netname

이 매개변수는 선택적으로서, 각각 자체의 호스트 이름을 가지고, 작동 중인 TCP/IP 인터페이스를 두 개 이상 가진 호스트를 지원하는 데 사용됩니다.

다음 예는, SP2EN1이 다중 TCP/IP 인터페이스, 즉 두 개의 논리 노드를 가지고 있으며 SP2SW1을 DB2 Universal Database 인터페이스로 사용하는 RS/6000 SP 시스템용으로 가능한 노드 구성 파일을 나타냅니다. 또한 (0보다는) 1에서 시작하는 노드 번호와 *nodenum* 순서의 공백을 나타냅니다.

<i>nodenum</i>	<i>hostname</i>	<i>logical-port</i>	<i>netname</i>
1	SP2EN1	0	SP2SW1
2	SP2EN1	1	SP2SW1
4	SP2EN2	0	
5	SP2EN3		

임의의 편집기를 사용하여 *db2nodes.cfg*를 갱신할 수 있습니다. 예외 조항은 편집기를 Windows NT에서 사용하지 말아야 한다는 것입니다. 그러나 데이터 파티션시에는 노드 번호를 변경하면 안 되므로, 파일 정보의 무결성이 보호되도록 주의해야 합니다. DB2START를 실행하면 노드 구성 파일이 잠기고, DB2STOP가 데이터베이스 관리 프로그램을 종료하면 잠금이 해제됩니다. 파일이 잠길 경우, DB2START 명령은 필요에 따라 파일을 갱신할 수 있습니다. 예를 들어, DB2START를 RESTART 옵션 또는 ADDNODE 옵션과 함께 발행할 수 있습니다.

주: DB2STOP 명령이 성공하지 못하여 노드 구성 파일의 잠금을 해제하지 못할 경우, DB2STOP FORCE를 발행하여 잠금 해제하십시오.

데이터베이스 구성 파일 작성

또한, 데이터베이스 구성 파일도 각 데이터베이스에 대해 작성됩니다. 이 파일은 자동으로 작성됩니다. 이 파일에는 다음과 같이 데이터베이스의 사용에 영향을 주는 여러 가지 구성 매개변수에 대한 값이 들어 있습니다.

- 데이터베이스 작성시 지정 및 사용된 매개변수(예: 데이터베이스 코드 페이지, 조합 순서, DB2 릴리스 레벨)
- 데이터베이스의 현재 상태를 나타내는 매개변수(예: 백업 보류 플래그, 데이터베이스 일관성 플래그, 롤 포워드 보류 플래그)
- 데이터베이스 조작시 사용할 수 있는 시스템 자원의 양을 정의하는 매개변수(예: 버퍼 풀 크기, 데이터베이스 로그, 정렬 메모리 크기)

이들 매개변수는 관리 안내서: 성능에 있는 『DB2 구성』에서 자세히 다룹니다.

성능 추가 정보: 구성 매개변수의 대부분은 기본값으로 주어지지만, 데이터베이스의 최적의 성능을 이루려면 이 값을 갱신해야 합니다.

다중 파티션의 경우: 둘 이상의 파티션에 걸쳐 파티션된 데이터베이스가 있으면, 구성 파일은 모든 데이터베이스 파티션에서 동일해야 합니다. SQL 컴파일러는 지역 노드 구성 파일의 정보에 근거하여 분산 SQL문을 컴파일하고, SQL문의 요구를 충족시킬 액세스 플랜을 작성하기 때문에 일관성이 요구됩니다. 데이터베이스 파티션에서 다른 구성 파일을 유지보수하는 작업은 명령문이 준비된 데이터베이스 파티션에 따라 다른 액세스 플랜이 될 수 있습니다. 구성 파일이 모든 데이터베이스 파티션에서 동기화되게 유지하려면 **db2_all**을 사용하십시오.

응답 파일을 사용한 구성 정보 복제

*db2rspgn*이라고 하는 응답 파일 생성 프로그램 유틸리티를 사용하여 시스템을 다시 설치할 때 사용될 수 있는 응답 파일을 작성하거나 현재 시스템의 레지스트리 변수, 데이터베이스 관리 프로그램, 구성 매개변수 및 데이터베이스 관리 구성 매개변수를 동일한 시스템에 복제할 수 있습니다.

하나 이상의 DB2 제품으로 시스템을 설치하고 환경에 맞게 매개변수를 조정 한 후, *db2rspgn*을 사용하여 필요한 값을 응답 파일에서 생성할 수 있습니다. 응답 파일은 동일한 시스템을 다시 작성하는 데 사용될 수 있습니다.

명령행 구문은 응답 파일 및 모든 지원 파일에 대한 대상 디렉토리를 선언합니다. 이 외에도, 복사할 인스턴스를 선택적으로 지정하고 관리 인스턴스 및 DataLinks 서버 인스턴스를 선택적으로 작동 불가능하게 할 수 있습니다.

이 유틸리티의 구문에 관한 세부사항 및 생성된 응답 파일 사용법에 관한 설명은 해당 빠른 시작에서 자세한 내용을 참조하십시오.

FCM 통신 작동

파티션된 데이터베이스 환경에서는 고속 통신 관리 프로그램(FCM)으로 데이터베이스 파티션간의 대부분의 통신을 다룹니다. 데이터베이스 파티션에서 FCM을 사용하고 다른 데이터베이스 파티션과의 통신을 허용하려면, 아래 표시된 것처럼 etc

디렉토리의 파티션 `services` 파일에서 서비스 항목을 작성해야 합니다. FCM은 지정된 포트를 사용하여 통신합니다. 같은 호스트에 다중 파티션이 있으면, 다음과 같이 포트 범위를 정의해야 합니다.

Windows NT 고려사항

Windows NT 환경에서 DB2 Enterprise - Extended Edition을 사용하는 경우, TCP/IP 포트 범위는 다음에 의해 자동으로 서비스 파일에 추가됩니다.

- 인스턴스를 작성하거나 새로운 노드를 추가할 때의 설치 프로그램
- 새로운 인스턴스를 작성할 때의 `db2icrt` 유틸리티
- 머신에 첫번째 노드를 추가할 때의 `db2ncrt` 유틸리티

추가 정보는 *Windows용 DB2 Enterprise - Extended Edition 빠른 시작*에서 자세한 내용을 참조하십시오.

서비스 항목의 구문은 다음과 같습니다.

```
DB2_instance port/tcp #comment
```

DB2_instance

인스턴스 값은 데이터베이스 관리 프로그램 인스턴스 이름과 같습니다. 이름의 모든 문자는 소문자로 되어 있어야 합니다. 인스턴스 이름이 `db2puser`일 경우, `DB2_db2puser`를 지정하십시오.

port/tcp

데이터베이스 파티션에 대해 예약하고자 하는 TCP/IP 포트

#comment

항목과 연관시키고자 하는 모든 주석. 주석 앞에는 우물정자(#)가 와야 합니다.

`/etc/services` 파일을 공유하는 경우, 파일에 할당된 포트 수가 인스턴스의 다중 데이터베이스 파티션의 가장 큰 수와 같거나 더 큰 수인지 확인해야 합니다. 또한 포트를 할당할 때에는, 백업으로 사용할 수 있는 모든 프로세서를 설명하도록 합니다.

/etc/services 파일을 공유할 수 없으면, 위의 경우와 더불어 한 가지 추가 고려사항이 있습니다. DB2 인스턴스에 정의된 항목이 모든 /etc/services 파일에서 동일해야 합니다(파티션된 데이터베이스에 적용되지 않는 기타 항목은 같을 필요가 없다고 해도).

하나의 인스턴스의 동일한 호스트에 다중 데이터베이스 파티션이 있을 경우, 사용할 FCM에 대해 둘 이상의 포트를 정의해야 합니다. 이 내용을 수행하려면, 두 행을 etc/services 파일에 입력하여 할당중인 포트 범위를 지정하십시오. 첫번째 행은 첫번째 포트를 지정하고, 두 번째 행은 포트 블록의 끝을 나타냅니다. 다음 예에서는 인스턴스 sales에 대해 5개의 포트가 할당됩니다. 즉, 인스턴스의 어떠한 프로세서도 6개 이상의 데이터베이스 파티션을 갖지 않습니다.

```
DB2_sales      9000/tcp
DB2_sales_END  9004/tcp
```

주: END는 반드시 대문자로 지정해야 합니다. 또한 두 개의 밑줄(_) 문자도 넣어야 합니다.

제3장 데이터베이스 작성

이 장에서는 사용자 데이터베이스 디자인의 구현의 일부일 수 있는 다양한 각 오브젝트에 대한 간단한 설명을 제공합니다.

이전 장은 데이터베이스를 작성하기 전에 알아야 하는 정보에 초점을 두었습니다. 또한 데이터베이스를 작성하기 전에 수행해야 하는 여러 주제와 타스크를 다루었습니다.

이 부분의 최종 장에서는 데이터베이스를 변경하기 전에 고려해야 하는 사항을 나 타냅니다. 그리고, 데이터베이스 오브젝트를 변경하거나 삭제하는 방법을 설명합니 다.

데이터베이스를 작성할 때, 다음과 같은 각 타스크가 수행됩니다.

- 데이터베이스가 필요로하는 모든 시스템 카탈로그 테이블 설정
- 데이터베이스 복구 로그의 할당
- 데이터베이스 구성 파일 및 기본값이 설정됩니다.
- 데이터베이스 유틸리티를 데이터베이스에 바인딩

데이터베이스 특권(시스템 카탈로그 뷰에 있는 CREATETAB, BINDADD, CONNECT, IMPLICIT_SCHEMA 및 SELECT 특권)은 자동으로 PUBLIC에 권한 부여됩니다.

제어 센터를 사용하여 데이터베이스를 작성하십시오.

- | |
|---|
| <ol style="list-style-type: none">1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.2. 데이터베이스 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 -> 마법사를 사 용한 데이터베이스를 선택하십시오.3. 이 타스크를 완료하려면 단계에 따르십시오. |
|---|

연관된 주석 "Personnel DB for BSchiefer Co"와 함께 다음의 명령행 처리기 명 령은 기본 위치에 person1이라는 데이터베이스를 작성합니다.

```
create database person1
with "Personnel DB for BSchiefer Co"
```

다른 데이터베이스 관리 프로그램 인스턴스(가능하다면 원격)에 데이터베이스를 작성하려면, 64 페이지의 『데이터베이스 관리 프로그램의 다중 인스턴스 사용』에서 자세한 내용을 참조하십시오. 이 주제는 또한 원격 인스턴스를 포함하여 기본 인스턴스 이외의 인스턴스에 대해 인스턴스 레벨 관리를 수행하는 데 필요한 명령을 소개합니다.

주: 기본 데이터베이스 위치 및 CREATE DATABASE 명령으로 다른 위치를 지정하는 것에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

데이터베이스 관리 프로그램을 작성할 때 사용자나 데이터베이스 관리 프로그램에 의해 수행되는 타스크는 다음 절에서 설명됩니다.

- 123 페이지의 『초기 노드 그룹의 정의』
- 123 페이지의 『초기 테이블 공간의 정의』
- 125 페이지의 『시스템 카탈로그 테이블의 정의』
- 126 페이지의 『데이터베이스 디렉토리의 정의』
- 128 페이지의 『DCE 디렉토리 서비스』
- 129 페이지의 『LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스』
- 130 페이지의 『데이터베이스 복구 로그의 정의』
- 131 페이지의 『유틸리티를 데이터베이스에 바인딩』
- 131 페이지의 『데이터베이스 카탈로그화』
- 129 페이지의 『노드 그룹 작성』
- 132 페이지의 『테이블 공간 작성』
- 138 페이지의 『스키마 작성』
- 140 페이지의 『테이블 작성 및 데이터 수록』
- 158 페이지의 『트리거 작성』
- 160 페이지의 『사용자 정의 함수(UDF) 및 메소드 작성』
- 164 페이지의 『사용자 정의 유형(UDT) 작성』

- 167 페이지의 『뷰 작성』
- 170 페이지의 『요약 테이블 작성』
- 173 페이지의 『별명 작성』
- 175 페이지의 『랩퍼 작성』
- 176 페이지의 『서버 작성』
- 184 페이지의 『별명 작성』
- 187 페이지의 『색인, 색인 확장 또는 색인 스펙 작성』

데이터베이스의 실제 구현에 관련된 추가 정보는 *관리 안내서: 계획에서 자세한 내용을 참조하십시오.*

초기 노드 그룹의 정의

데이터베이스가 처음 작성될 때, 데이터베이스 파티션은 db2nodes.cfg 파일에 지정된 모든 파티션용으로 작성됩니다. 기타 파티션은 ADD NODE 명령 및 DROP NODE 명령을 사용하여 추가하거나 제거할 수 있습니다.

세 개의 노드 그룹은 다음과 같이 정의됩니다.

- 시스템 카탈로그 테이블이 들어 있는 SYSCATSPACE 테이블 공간의 경우, IBMCATGROUP
- 데이터베이스 처리중에 작성된 임시 테이블이 들어 있는 TEMPSPACE1의 경우, IBMTEMPGROUP
- 기본적으로 사용자 테이블과 색인이 들어 있는 USERSPACE1의 경우, IBMDEFAULTGROUP

초기 테이블 공간의 정의

데이터베이스가 작성되면, 세 개의 테이블 공간이 정의됩니다.

- 시스템 카탈로그 테이블의 경우, SYSCATSPACE(125 페이지의 『시스템 카탈로그 테이블의 정의』 참조)
- 데이터베이스 처리 시스템에 작성된 임시 테이블의 경우, TEMPSPACE1
- 사용자 정의 테이블 및 색인의 경우, USERSPACE1

주: 처음 데이터베이스를 작성할 때 사용자 임시 테이블 공간이 작성되지 않습니다.

CREATE DATABASE 명령을 사용하여 다른 테이블 공간 매개변수를 지정하지 않으면, 데이터베이스 관리 프로그램은 시스템 관리 저장영역(SMS) 디렉토리 컨테이너를 사용하여 이 테이블 공간을 작성합니다. 이 디렉토리 컨테이너는 데이터베이스용으로 작성된 서브디렉토리에 작성됩니다(관리 안내서: 계획에서 데이터베이스 실제 디렉토리에 대한 자세한 내용을 참조하십시오). 이러한 테이블 공간에 extent 크기는 기본값으로 설정됩니다.

제어 센터를 사용하여 초기 테이블 공간을 정의하려면 다음을 수행하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 데이터베이스 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 → 마법사를 사용한 데이터베이스를 선택하십시오.
3. 이 작업을 완료하려면 단계를 따르십시오.

명령행을 사용하여 초기 테이블 공간을 정의하려면 다음을 수행하십시오.

```
CREATE DATABASE <name>
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('<path>')
    EXTENTSIZE <value> PREFETCHSIZE <value>
  USER TABLESPACE
    MANAGED BY DATABASE USING (FILE'<path>' 5000,
                                FILE'<path>' 5000)
    EXTENTSIZE <value> PREFETCHSIZE <value>
  TEMPORARY TABLESPACE
    MANAGED BY SYSTEM USING ('<path>')
  WITH "<comment>"
```

이러한 테이블 공간에 대해 기본 정의를 사용하지 않으려면, CREATE DATABASE 명령에 특성을 지정해야 합니다. 예를 들어, 다음 명령은 OS/2에서 데이터베이스를 작성할 때 사용될 수 있습니다.

```
CREATE DATABASE PERSONL
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('d:\pcatalog','e:\pcatalog')
    EXTENTSIZE 16 PREFETCHSIZE 32
  USER TABLESPACE
```



```

MANAGED BY DATABASE USING (FILE'd:\db2data\person1' 5000,
                             FILE'd:\db2data\person1' 5000)
EXTENTSIZE 32 PREFETCHSIZE 64
TEMPORARY TABLESPACE
MANAGED BY SYSTEM USING ('f:\db2temp\person1')
WITH "Personnel DB for BSchiefer Co"

```

이 예에서, 각각의 초기 테이블 공간에 대한 정의는 명시적으로 제공됩니다. 기본 정의를 사용하지 않으려는 경우에는 테이블 공간에 대해 사용자가 테이블 공간 정의를 지정하기만 하면 됩니다.

CREATE DATABASE 명령의 MANAGED BY절을 코드화하는 형식은 CREATE TABLESPACE 명령의 MANAGED BY절의 형식과 동일합니다. 추가 예는 132 페이지의 『테이블 공간 작성』에서 자세한 내용을 참조하십시오.

데이터베이스를 작성하기 전에 테이블 공간을 디자인하고 선택하는 정보와 관리 안내서: 계획 매뉴얼에서 자세한 내용을 참조하십시오.

시스템 카탈로그 테이블의 정의

시스템 카탈로그 테이블 세트는 각 데이터베이스에 대해 작성되고 유지보수됩니다. 이 테이블에는 데이터베이스 오브젝트(예: 테이블, 뷰, 색인, 패키지) 정의에 대한 정보와 이 오브젝트에 대해 사용자가 가진 액세스 유형에 대한 보안 정보가 들어 있습니다. 이 테이블은 SYSCATSPACE 테이블 공간에 저장됩니다.

이 테이블은 데이터베이스를 조작하는 동안에 갱신됩니다(예를 들어, 테이블이 작성될 때). 사용자가 이들 테이블을 명시적으로 작성하거나 삭제할 수는 없지만, 해당 내용을 조회하거나 볼 수는 있습니다. 데이터베이스가 작성되면, 시스템 카탈로그 테이블 오브젝트 이외에 다음 데이터베이스 오브젝트가 시스템 카탈로그에 정의됩니다.

- 사용자 정의 함수(UDF)의 세트가 SYSFUN 스키마에 작성됩니다. 이들 시스템 작성 함수에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.
- 시스템 카탈로그 테이블에 대한 읽기 전용 뷰의 세트가 SYSCAT 스키마에 작성됩니다. 이들 뷰에 대해서는 *SQL 참조서*의 『카탈로그 뷰』를 참조하십시오.
- 갱신 가능한 카탈로그 뷰의 세트가 SYSSTAT 스키마에 생성됩니다. 이들 갱신 가능 뷰를 사용하면, 특정 통계 정보를 갱신하여 가상 데이터베이스의 성능

을 조사하거나 RUNSTATS 유틸리티를 사용하지 않고 통계를 갱신할 수 있습니다. 이들 뷰에 대해서는 *SQL 참조서*의 『갱신 가능한 카탈로그 뷰』를 참조하십시오.

데이터베이스가 작성되고 나면, 305 페이지의 『시스템 카탈로그 뷰 보안』에 기술된 대로 시스템 카탈로그 뷰에 대한 액세스를 제한하고자 할 수 있습니다.

데이터베이스 디렉토리의 정의

새 데이터베이스를 설치하거나 설정할 때에는 세 개의 디렉토리가 사용됩니다.

- 지역 데이터베이스 디렉토리
- 시스템 데이터베이스 디렉토리
- 노드 디렉토리

지역 데이터베이스 디렉토리

지역 데이터베이스 디렉토리 파일은 데이터베이스가 정의된 각 경로(일부 운영 체제에서 『드라이브』라고 함)에 존재합니다. 이 디렉토리에는 해당 위치에서 액세스할 수 있는 각 데이터베이스에 대해 하나의 항목이 들어 있습니다. 각 항목에는 다음 항목이 들어 있습니다.

- CREATE DATABASE 명령으로 제공되는 데이터베이스 이름
- 데이터베이스 별명(별명이 지정되지 않으면 동일한 데이터베이스 이름으로 대신 함)
- CREATE DATABASE 명령으로 제공된 데이터베이스를 설명하는 주석
- 데이터베이스에 대한 루트 디렉토리의 이름
- 기타 시스템 정보

특정 데이터베이스에 대한 이 파일의 내용을 보려면 다음 명령을 발행하는데, 여기서 *location*은 데이터베이스의 위치를 지정합니다.

```
LIST DATABASE DIRECTORY ON location
```

시스템 데이터베이스 디렉토리

시스템 데이터베이스 디렉토리 파일은 데이터베이스 관리 프로그램의 인스턴스마다 존재하며, 이 인스턴스용으로 카탈로그화된 각 데이터베이스 항목이 하나 들어 있습니다. 데이터베이스는 CREATE DATABASE 명령이 발행될 때 내재적으로 카탈로그화되거나 CATALOG DATABASE 명령을 사용하여 명시적으로 카탈로그화될 수도 있습니다. 데이터베이스 카탈로그화에 대해서는 131 페이지의 『데이터베이스 카탈로그화』에서 자세한 내용을 참조하십시오.

작성된 각 데이터베이스의 경우, 항목은 다음 정보가 들어 있는 디렉토리에 추가됩니다.

- CREATE DATABASE 명령으로 제공되는 데이터베이스 이름
- 데이터베이스 별명(데이터베이스 이름과 동일함)
- CREATE DATABASE 명령으로 제공되는 데이터베이스 주석
- 지역 데이터베이스 디렉토리의 위치
- 데이터베이스가 간접적임을 나타내는 표시기. 간접적이란 데이터베이스가 시스템 데이터베이스 디렉토리 파일과 동일한 머신에 있음을 의미합니다.
- 기타 시스템 정보

이 파일의 내용을 보려면, 데이터베이스 디렉토리 파일 위치를 지정하지 않고 LIST DATABASE DIRECTORY 명령을 발행하십시오.

파티션된 데이터베이스 환경에서, 모든 데이터베이스 파티션이 같은 시스템 데이터베이스 디렉토리 파일인 sqlbdir, 즉 인스턴스의 홈 디렉토리에 있는 sqlbdir 서브디렉토리의 파일에 항상 액세스하는지 확인해야 합니다. 동일한 sqlbdir 서브디렉토리에 있는 시스템 데이터베이스 디렉토리 또는 시스템 인텐션 파일 sqlbins가 공유 파일 시스템의 다른 파일에 대한 기호 링크일 경우, 예측할 수 없는 오류가 발생할 수 있습니다. 이들 파일은 69 페이지의 『데이터 파티션 작동』에 설명되어 있습니다.

노드 디렉토리

데이터베이스 관리 프로그램은 첫번째 데이터베이스 파티션이 카탈로그화될 때 노드 디렉토리를 작성합니다. 데이터베이스 파티션을 카탈로그화하려면, CATALOG

NODE 명령을 사용하십시오. 지역 노드 디렉토리의 내용을 나열하려면, LIST NODE DIRECTORY 명령을 사용하십시오. 각 데이터베이스 클라이언트에서 노드 디렉토리가 작성되고 유지보수됩니다. 디렉토리에는 클라이언트가 액세스할 수 있는 하나 이상의 데이터베이스를 갖는 각 원격 워크스테이션에 대한 항목이 들어 있습니다. DB2 클라이언트는 데이터베이스 연결 또는 인스턴스 접속이 요청될 때 마다 노드 디렉토리의 통신 끝점 정보를 사용합니다.

또한, 디렉토리의 항목에는 클라이언트에서 원격 데이터베이스 파티션으로 통신하는 데 사용되는 통신 프로토콜 유형에 대한 정보도 들어 있습니다. 지역 데이터베이스 파티션을 카탈로그화하면 동일한 머신에 상주하는 인스턴스의 별명도 작성됩니다. 사용자의 클라이언트에서 액세스해야 할 동일한 워크스테이션에 둘 이상의 인스턴스가 있을 경우, 지역 노드가 카탈로그화됩니다.

DCE 디렉토리 서비스

DCE는 분산된 이질적 컴퓨팅 환경에서 응용프로그램의 작성, 사용 및 유지보수를 지원하는 도구와 서비스를 제공하는 OSF**(Open Systems Foundation**) 아키텍처입니다. 이는 운영 체제, 네트워크 및 분산 응용프로그램(클라이언트 응용프로그램이 원격 서버에 액세스할 수 있도록 하는)간의 계층 구조입니다.

지역 디렉토리과 함께 대상 데이터베이스의 실제 위치가 데이터베이스 디렉토리와 노드 디렉토리의 각 클라이언트 워크스테이션에 개별적으로 저장됩니다. 그러므로, 데이터베이스 관리자가 이 디렉토리를 갱신하고 변경하는 데 많은 시간을 소비하게 됩니다. DCE 디렉토리 서비스는 지역 디렉토리에 대한 대안으로 중앙 디렉토리를 제공합니다. 이로 인해 데이터베이스 또는 데이터베이스 관리 프로그램 인스턴스에 관한 정보가 중앙 위치에 일단 기록될 수 있으며, 어떠한 변경사항 또는 갱신사항도 하나의 지점에서 작성될 수 있습니다.

DCE가 DB2를 수행하기 위한 전제조건은 아니지만, 사용자가 DCE 환경에서 작업중이면, 453 페이지의 『부록B. DCE 디렉토리 서비스 사용』에서 자세한 내용을 참조하십시오.

LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스

LDAP(Lightweight Directory Access Protocol)은 디렉토리 서비스에 대한 산업 표준 액세스 방식입니다. 디렉토리 서비스는 분산 환경 내의 여러 시스템과 서비스에 대한 자원 정보의 저장소이며, 이들 자원에 대한 클라이언트 및 서버 액세스를 제공합니다. 각 데이터베이스 서버 인스턴스는 LDAP 서버로 그 존재를 발행하며 데이터베이스가 작성될 때 LDAP 등록부로 데이터베이스 정보를 제공합니다. 클라이언트가 데이터베이스에 연결할 때, 카탈로그에 대한 정보는 LDAP 등록부에서 검색될 수 있습니다. 각 클라이언트는 각 머신에 지역적으로 카탈로그 정보를 저장하는 것이 더 이상 필요하지 않습니다. 클라이언트 응용프로그램은 데이터베이스에 연결하는 데 필요한 정보 요구를 찾기 위해 LDAP 등록부를 검색합니다.

LDAP가 DB2를 수행하기 위한 전제조건은 아니지만, 사용자가 LDAP 환경에서 작업중일 경우, 559 페이지의 『부록J. LDAP(Lightweight Directory Access Protocol) 등록부 서비스』에서 자세한 내용을 참조하십시오.

노드 그룹 작성

CREATE NODEGROUP문을 사용하여 노드 그룹을 작성하십시오. 이 명령문은 테이블 공간 컨테이너 및 테이블 데이터가 상주하는 노드 세트를 지정합니다. 이 명령문은 또한 다음 기능을 수행합니다.

- 노드 그룹에 대해 파티션 맵을 작성합니다. *관리 안내서: 계획에서 파티션 맵에 대한 자세한 내용을 참조하십시오.*
- 파티션 맵 ID를 생성합니다.
- 레코드를 다음 카탈로그 테이블에 삽입합니다.
 - SYSCAT.NODEGROUPS
 - SYSCAT.PARTITIONMAPS
 - SYSCAT.NODEGROUPDEF

제어 센터를 사용하여 노드 그룹을 작성하십시오.

1. 노드 그룹 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 노드 그룹 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 작성을 선택하십시오.
3. 노드 그룹 작성 창에서, 정보를 완료하고, 화살표를 사용하여 사용 가능한 노드 상자에서 선택된 노드 상자로 노드를 이동시키고, 확인을 클릭하십시오.

명령행을 사용하여 노드 그룹을 작성하려면 다음을 입력하십시오.

```
CREATE NODEGROUP <name> ON NODES (<value>,<value>)
```

데이터베이스에 있는 데이터베이스 파티션의 부속 집합에 일부 테이블을 로드하고자 한다고 가정해 보십시오. 다음 명령을 사용하여 최소한 세(0에서 2) 노드를 구성하는 데이터베이스에서 두 개의 노드(1과 2)의 노드 그룹을 작성합니다.

```
CREATE NODEGROUP mixng12 ON NODES (1,2)
```

노드 그룹 작성에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

또한, CREATE DATABASE 명령 또는 sqlcrea() API는 기본 시스템 노드 그룹, IBMDEFAULTGROUP, IBMCATGROUP 및 IBMTEMPGROUP을 작성합니다 (관리 안내서: 계획에서 자세한 내용을 참조하십시오).

데이터베이스 복구 로그의 정의

데이터베이스 복구 로그는 새로운 테이블의 추가 또는 기존 테이블의 갱신을 포함한 데이터베이스에서 이루어진 모든 변경사항을 기록합니다. 이 로그는 다수의 로그 extent로 구성되며, 로그 파일인 분리 파일에 각각 들어 있습니다.

데이터베이스 복구 로그를 사용하여 장애(예를 들어, 시스템 전원 중단 또는 응용 프로그램 오류)가 상반된 상태에서 데이터베이스를 떠나지 않은 것을 확인할 수 있습니다. 장애가 발생한 경우, 이미 내용이 변경되었으나 확약되지 않은 변경사항은 구간 복원되고, 물리적으로 디스크에 기록되지 않은 확약된 모든 트랜잭션은 다시 수행됩니다. 이러한 조치로 데이터베이스의 무결성을 확인할 수 있습니다.

347 페이지의 『제8장 데이터베이스 복구』에서 자세한 내용을 참조하십시오.

유틸리티를 데이터베이스에 바인딩

데이터베이스가 작성되면, 데이터베이스 관리 프로그램은 db2ubind.lst에 있는 유틸리티를 데이터베이스에 바인드하려고 합니다. 이 파일은 sqllib 디렉토리의 bnd 서브디렉토리에 저장됩니다.

유틸리티를 바인딩하면 패키지가 작성됩니다. 이 패키지는 단일 소스 파일로부터 특정 SQL문을 처리하는 데 필요한 모든 정보를 가지고 있는 오브젝트입니다.

주: 클라이언트로부터 이들 유틸리티를 사용하려면, 유틸리티를 명시적으로 바인드해야 합니다. 자세한 정보는 플랫폼에 맞는 빠른 시작 매뉴얼을 참조하십시오.

유틸리티를 데이터베이스에 바인드하거나 재바인드해야 하는 경우, 명령행 처리기를 사용하여 다음 명령을 발행하십시오.

```
connect to sample
bind @db2ubind.lst
```

주: sample 데이터베이스에 패키지를 작성하려면 사용자는 이들 파일이 상주하는 디렉토리 내에 있어야 합니다. 바인드 파일은 SQLLIB 디렉토리의 BND 서브디렉토리에 있습니다. 이 예에서, sample은 데이터베이스 이름입니다.

데이터베이스 카탈로그화

새로운 데이터베이스를 작성하면, 시스템 데이터베이스 디렉토리 파일에 자동으로 카탈로그화됩니다. 사용자는 데이터베이스를 시스템 데이터베이스 디렉토리 파일에 명시적으로 카탈로그화하기 위해 CATALOG DATABASE 명령을 사용할 수도 있습니다. CATALOG DATABASE 명령을 사용함으로써, 다른 별명으로 데이터베이스를 카탈로그화하거나, 이전에 UNCATALOG DATABASE 명령을 사용하여 삭제된 데이터베이스 항목을 카탈로그화할 수 있습니다.

다음의 명령행 처리기 명령은 person1 데이터베이스를 humanres로 카탈로그화합니다.

```
catalog database person1 as humanres
with "Human Resources Database"
```

여기서, 시스템 데이터베이스 디렉토리 항목은 데이터베이스 별명으로서 humanres 를 가지는데, 이는 데이터베이스 이름(person1)과는 다릅니다.

기본 인스턴스가 아닌 다른 인스턴스에서 데이터베이스를 카탈로그화할 수도 있습니다. 다음 예에서는 데이터베이스 B를 INSTANCE_C에 연결합니다.

```
catalog database b as b at node instance_c
```

주: CATALOG DATABASE 명령은 데이터베이스 서버 머신상에 상주하는 데이터베이스를 카탈로그화하기 위해 클라이언트 노드에서도 사용됩니다. 자세한 정보는 플랫폼에 맞는 빠른 시작에서 자세한 내용을 참조하십시오.

DCE 셸 디렉토리에 대해서는 128 페이지의 『DCE 디렉토리 서비스』와 453 페이지의 『부록B. DCE 디렉토리 서비스 사용』에서 자세한 내용을 참조하십시오.

주: 성능을 개선하기 위해, 데이터베이스 디렉토리를 포함하여 디렉토리 파일을 메모리에 캐쉬할 수 있습니다. (디렉토리 캐쉬 작동에 대해서는 관리 안내서: 성능의 『디렉토리 캐쉬 지원(dir_cache)』에서 자세한 내용을 참조하십시오.) 디렉토리 캐쉬가 작동되면, 다른 응용프로그램이 디렉토리에 대해 수행한 변경사항(예를 들어, CATALOG DATABASE 또는 UNCATALOG DATABASE 명령을 사용하여)은 사용자의 응용프로그램이 다시 시작될 때까지 적용되지 않습니다. 명령행 처리기 세션에 의해 사용된 디렉토리 캐쉬를 새로 고치려면, db2 terminate 명령을 실행하십시오.

응용프로그램 레벨 캐쉬뿐만 아니라, 데이터베이스 관리 프로그램 레벨 캐쉬도 내재적으로 데이터베이스 관리 프로그램을 탐색하는 데 사용됩니다. 이러한 『공유』 캐쉬를 새로 고치려면, db2stop 및 db2start 명령을 발행하십시오.

디렉토리 캐쉬에 관한 자세한 정보는 관리 안내서: 성능의 『디렉토리 캐쉬 지원 (dir_cache)』을 참조하십시오.

테이블 공간 작성

데이터베이스 내에서 테이블 공간을 작성하면 컨테이너가 테이블 공간에 할당되고 정의 및 속성이 데이터베이스 시스템 카탈로그에 기록됩니다. 그런 다음, 이 테이블 공간에 테이블을 작성할 수 있습니다.

테이블 공간에 관한 설계 정보는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.

CREATE TABLESPACE문 구문은 *SQL 참조서 매뉴얼*에 자세히 설명되어 있습니다. SMS 및 DMS 테이블 공간에 대해서는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 테이블 공간을 작성하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 공간 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
3. 마법사의 단계에 따라 작업을 완료하십시오.

명령행을 사용하여 SMS 테이블 공간을 작성하십시오.

```
CREATE TABLESPACE <NAME>  
    MANAGED BY SYSTEM  
    USING ('<path>')
```

명령행을 사용하여 SMS 테이블 공간을 작성하십시오.

```
CREATE TABLESPACE <NAME>  
    MANAGED BY DATABASE  
    USING (FILE'<path>' <size>)
```

다음 SQL문은 세 개의 개별 드라이브의 세 개의 디렉토리를 사용하여 OS/2 또는 Windows NT에 SMS 테이블 공간을 작성합니다.

```
CREATE TABLESPACE RESOURCE  
    MANAGED BY SYSTEM  
    USING ('d:\acc_tbsp', 'e:\acc_tbsp', 'f:\acc_tbsp')
```

다음 SQL문은 각각 5,000 페이지인 두 개의 파일 컨테이너를 사용하여 OS/2에서 DMS 테이블 공간을 작성합니다.

```
CREATE TABLESPACE RESOURCE  
    MANAGED BY DATABASE  
    USING (FILE'd:\db2data\acc_tbsp' 5000,  
          FILE'e:\db2data\acc_tbsp' 5000)
```

위의 두 예에서 컨테이너에 대해 명시적인 이름이 제공됩니다. 그러나, 상대 컨테이너 이름을 지정하는 경우, 컨테이너가 데이터베이스용으로 작성된 서브디렉토리에 작성됩니다(관리 안내서: 계획에서 데이터베이스 실제 디렉토리에 대한 자세한 내용을 참조하십시오).

또한 지정된 경로 이름의 일부가 존재하지 않으면, 데이터베이스 관리 프로그램이 작성합니다. 서브디렉토리가 데이터베이스 관리 프로그램에 의해 작성된 경우, 서브디렉토리는 테이블 공간이 삭제될 때 데이터베이스 관리 프로그램에 의해 삭제될 수도 있습니다.

위 예에서는 테이블 공간이 특정 노드 그룹과 연관되지 않은 것으로 가정합니다. 기본 노드 그룹 IBMDEFAULTGROUP은 다음 매개변수가 명령문에 지정되지 않은 경우에 사용됩니다.

```
IN nodegroup
```

다음 SQL문은 각각 10,000페이지의 세 개의 논리 볼륨을 사용하여 UNIX 기반 시스템에 DMS 테이블 공간을 작성하고, 입출력 특성을 지정합니다.

```
CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rdblv6' 10000,
        DEVICE '/dev/rdblv7' 10000,
        DEVICE '/dev/rdblv8' 10000)
  OVERHEAD 24.1
  TRANSFERRATE 0.9
```

이 SQL문에서 언급된 UNIX 장치가 이미 존재하고 있어야 하며, 인스턴스 소유자 및 SYSADM 그룹이 이를 작성할 수 있어야 합니다.

다음 예에서는 UNIX 파티션 데이터베이스의 ODDNODEGROUP이라는 노드 그룹에 DMS 테이블 공간을 작성합니다. 이전에 ODDNODEGROUP이 CREATE NODEGROUP문으로 작성되어 있어야 합니다. 이 경우, ODDNODEGROUP 노드 그룹은 1, 3, 5로 번호가 매겨진 데이터베이스 파티션으로 구성된 것으로 가정됩니다. 모든 데이터베이스 파티션에서 10 000 4KB 페이지에 /dev/hdisk0 장치를 사용하십시오. 또한, 40 000 4KB 페이지의 각 데이터베이스 파티션에 대해서도 장치를 선언하십시오.

```

CREATE TABLESPACE PLANS
MANAGED BY DATABASE
USING (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n1hd01' 40000) ON NODE 1
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n3hd03' 40000) ON NODE 3
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n5hd05' 40000) ON NODE 5

```

UNIX 장치는 두 개의 범주(문자 직렬 장치 및 블록 구조 장치)로 분류됩니다. 모든 파일 시스템 장치의 경우, 각 블록 장치(또는 *cooked* 장치)에 대해 해당하는 문자 직렬 장치(또는 원시 장치)를 갖는 것이 일반적입니다. 블록 구조 장치는 일반적으로 『hd0』 또는 『fd0』와 유사한 이름으로 이름 지정됩니다. 문자 직렬 장치는 일반적으로 『rhd0』, 『rfd0』 또는 『rmt0』와 유사한 이름으로 이름 지정됩니다. 이 문자 직렬 장치는 블록 장치보다 액세스 속도가 빠릅니다. 문자 직렬 장치 이름은 CREATE TABLESPACE 명령에 사용되며, 블록 장치 이름은 그렇지 않습니다.

오버헤드 및 전송률은 SQL문이 컴파일될 때 사용할 최상의 액세스 경로를 결정하는 데 도움이 됩니다. OVERHEAD 및 TRANSFERRATE 매개변수에 대해서는 *관리 안내서: 성능의 『응용프로그램 성능 조정』*을 참조하십시오.

DB2는 병렬 입출력을 사용하는 순차 프리페치 기능을 사용하여 순차적인 입출력의 성능을 크게 개선할 수 있습니다. 이 기능의 세부사항은 *관리 안내서: 성능의 『순차적 프리페칭 이해』*를 참조하십시오.

또한 기본값인 4KB보다 큰 페이지 크기를 사용하는 테이블 공간을 작성할 수 있습니다. 다음 SQL문은 UNIX 기반 시스템에 8KB 페이지 크기의 SMS 테이블 공간을 작성합니다.

```

CREATE TABLESPACE SMS8K
PAGE SIZE 8192
MANAGED BY SYSTEM
USING ('FSMS_8K_1')
BUFFERPOOL BUFPPOOL8K

```

연관된 버퍼 풀 또한 동일한 8KB 페이지 크기여야 한다는 점을 유의하십시오.

참조하는 버퍼 풀이 활성화될 때까지는 작성된 테이블 공간을 사용할 수 없습니다.

ALTER TABLESPACE SQL문은 DMS 테이블 공간에 컨테이너를 추가하고 테이블 공간에 대해 PREFETCHSIZE, OVERHEAD 및 TRANSFERRATE 설정

값을 수정하는 데 사용될 수 있습니다. 테이블 공간 명령문을 발행하는 트랜잭션은 시스템 카탈로그 경합을 막기 위해 가능한 빨리 확장되어야 합니다.

주: PREFETCHSIZE 값은 EXTENTSIZE 값의 배가 되어야 합니다. 예를 들어, EXTENTSIZE가 10이면 PREFETCHSIZE는 20 또는 30이 되어야 합니다. 자세한 정보는 *관리 안내서*: 성능의 『순차적 프리페칭 이해』를 참조하십시오.

시스템 임시 테이블 공간 작성

시스템 임시 테이블 공간은 시스템 임시 테이블을 저장하는 데 사용됩니다. 데이터베이스가 작성되면, 정의된 세가지 기본 테이블 공간 중 하나는 『TEMPSPACE1』이라고 하는 시스템 임시 테이블 공간입니다.

주: 시스템 임시 테이블이 그러한 테이블 공간에 저장될 수 있으므로 데이터베이스는 항상 최소한 하나의 시스템 임시 테이블 공간을 가져야 합니다.

CREATE TABLESPACE문을 사용하여 또다른 시스템 임시 테이블 공간을 작성할 수 있습니다. 예를 들면, 다음과 같습니다.

```
CREATE SYSTEM TEMPORARY TABLESPACE tmp_tbsp
MANAGED BY SYSTEM
USING ('d:\tmp_tbsp','e:\tmp_tbsp')
```

시스템 임시 테이블 공간을 작성할 때 지정될 수 있는 유일한 노드 그룹은 IBMTEMPGROUP입니다.

사용자 임시 테이블 공간 작성

사용자 임시 테이블 공간은 선언된 임시 테이블을 저장하는 데 사용됩니다.

CREATE TABLESPACE문을 사용하여 사용자 임시 테이블 공간을 작성할 수 있습니다.

```
CREATE USER TEMPORARY TABLESPACE usr_tbsp
MANAGED BY DATABASE
USING (FILE 'd:\db2data\user_tbsp' 5000,
FILE 'e:\db2data\user_tbsp' 5000)
```

일반 테이블 공간과 같이, 사용자 임시 테이블 공간은 IBMTEMPGROUP 이외의 임의 노드 그룹에 작성될 수 있습니다. 사용자 임시 테이블 공간을 작성할 때 사용되는 기본 노드 그룹은 IBMDEFAULTGROUP입니다.

DECLARE GLOBAL TEMPORARY TABLE문은 사용자 임시 테이블 공간 내에서 사용할 선언된 임시 테이블을 정의합니다.

노드 그룹에 테이블 공간 작성

다중 데이터베이스 파티션 노드 그룹에 테이블 공간을 배치하면, 테이블 공간 내의 모든 테이블이 노드 그룹의 각 데이터베이스 파티션에 파티션됩니다. 테이블 공간은 노드 그룹 안에 작성됩니다. 일단 테이블 공간이 노드 그룹 내에 작성되면, 계속 그 안에 남아 있어야 합니다. 다른 노드 그룹으로 변경될 수 없습니다. 테이블 공간과 노드 그룹을 연관시키는 데에는 CREATE TABLESPACE문이 사용됩니다.

원시 입출력

DB2 Universal Database는 직접 디스크 액세스(원시 입출력)를 지원합니다. 이는 DB2 Universal Database 시스템에 직접 디스크 액세스(원시) 장치를 접속할 수 있게 합니다. 유일한 예외는 Linux, Windows 95 및 Windows 98 운영 체제입니다. 다음 목록은 이 장치 유형을 식별하는 물리적 및 논리적 방법을 설명합니다.

- Windows에서, 실제 하드 드라이브를 지정하려면, 다음과 같은 구문을 사용하십시오.

```
\\.\PhysicalDriveN
```

여기서, N은 시스템의 물리 드라이브 중 하나를 나타냅니다. 예를들어, N은 0, 1, 2 또는 다른 양의 정수로 대체될 수 있습니다.

```
\\.\PhysicalDisk5
```

- Windows에서, 논리 원시 파티션(즉, 형식화되지 않은 파티션)을 지정하려면 다음과 같은 구문을 사용하십시오.

```
\\.\N:
```

여기서, N:은 시스템의 논리 드라이브 문자를 나타냅니다. 예를 들어, N:은 E: 또는 임의의 다른 드라이브 이름으로 바꿀 수 있습니다.

- **참고:** 로그를 장치에 기록하기 위해서는 서비스 팩 3이 설치된 Windows NT 버전 4.0이 있어야 합니다.
- UNIX 기반 플랫폼에서, /dev/rhd0와 같이 문자 연속 장치 이름을 사용하십시오.

스키마 작성

데이터를 테이블에 구성하는 동안, 테이블(및 기타 관련 오브젝트)을 서로 그룹화하는 것이 좋습니다. 이 작업은 CREATE SCHEMA문을 사용하여 스키마를 정의함으로써 이루어집니다. 스키마 정보는 사용자가 연결한 데이터베이스의 시스템 카탈로그 테이블에 저장됩니다. 이 정보는 다른 오브젝트가 작성될 때, 이 스키마 내에 배치될 수 있습니다.

CREATE SCHEMA문 구문은 *SQL 참조서 매뉴얼*에 자세히 설명되어 있습니다. 새로운 스키마 이름은 시스템 카탈로그에 아직 존재하지 않으며, "SYS"로 시작되지 않습니다.

사용자가 SYSADM이나 DBADM 권한을 가지고 있을 경우, 유효한 이름을 사용하여 스키마를 작성할 수 있습니다. 데이터베이스가 작성될 때, IMPLICIT_SCHEMA 권한에는 PUBLIC(즉, 모든 사용자가 사용 가능한) 권한이 부여됩니다.

CREATE SCHEMA문의 일부로서 작성된 모든 오브젝트의 작성자는 스키마 소유자입니다. 이 소유자는 다른 사용자에게 스키마 특권을 권한 부여(GRANT)하거나 권한 취소(REVOKE)할 수 있습니다.

이 명령문은 DBADM 권한을 가진 사용자가 발행해야 합니다.

스키마는 사용자가 IMPLICIT_SCHEMA 권한을 가진 경우에 내재적으로 작성될 수도 있습니다. 이 권한으로, 사용자는 아직 존재하지 않는 스키마 이름을 가진 오브젝트를 작성할 때마다 스키마를 내재적으로 작성합니다.

사용자에게 `IMPLICIT_SCHEMA` 권한이 없을 경우, 사용자의 권한 부여 ID와 동일한 이름을 갖는 스키마만을 작성할 수 있습니다.

제어 센터를 사용하여 스키마를 작성하십시오.

1. 스키마 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 스키마 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 작성을 선택하십시오.
3. 새로운 스키마에 대한 정보를 완료하고, 확인을 클릭하십시오.

명령행을 사용하여 스키마를 작성하십시오.

```
CREATE SCHEMA <name> AUTHORIZATION <name>
```

다음은 권한 부여 ID "joe"를 가진 개별 사용자용 스키마를 작성하는 `CREATE SCHEMA`문의 예입니다.

```
CREATE SCHEMA joeschma AUTHORIZATION joe
```

스키마 설정

특정 DB2 연결 내에서 발행한 동적 SQL문에 규정화되지 않은 오브젝트 참조로 사용할 기본 스키마를 설정하려고 할 수 있습니다. 이것은 기본적으로 사용하려는 스키마에 특수 레지스터 `CURRENT_SCHEMA`를 설정하여 수행됩니다. 모든 사용자가 이 특수 레지스터를 설정할 수 있으며, 권한 부여는 필요하지 않습니다.

`SET SCHEMA`문의 구문은 *SQL 참조서 매뉴얼*에 자세히 설명되어 있습니다.

다음은 `CURRENT_SCHEMA` 특수 레지스터 설정 방법에 대한 예입니다.

```
SET CURRENT_SCHEMA = 'SCHEMA01'
```

이 명령문은 응용프로그램 내에서 사용될 수 있거나 대화식으로 실행될 수 있습니다. 일단 `CURRENT_SCHEMA` 특수 레지스터 값이 설정되면, 이는 동적 SQL문의 규정화되지 않은 오브젝트 참조용 규정자(스키마)로서 사용됩니다(데이터베이스 오브젝트로의 규정화되지 않은 참조가 있는 `CREATE SCHEMA`문의 경우 예외).

`CURRENT_SCHEMA` 특수 레지스터의 초기 값은 현재 세션 사용자의 권한 부여 ID와 같습니다.

테이블 작성 및 데이터 수록

테이블 안에서 데이터를 어떻게 구성할 것인지를 결정하고 나면 그 다음 단계는 CREATE TABLE문을 사용하여 이 테이블을 작성하는 것입니다. 테이블 설명은 사용자가 연결한 데이터베이스의 시스템 카탈로그에 저장됩니다.

CREATE TABLE문 구문은 *SQL 참조서 매뉴얼*에 자세히 설명되어 있습니다. 170 페이지의 『요약 테이블 작성』에서 요약 테이블 작성에 대한 자세한 정보를 참조하십시오. 테이블, 컬럼 및 기타 데이터베이스 오브젝트의 이름 지정 규칙에 대해서는 443 페이지의 『부록A. 이름 지정 규칙』에서 자세한 내용을 참조하십시오.

CREATE TABLE문은 테이블에 이름을 부여합니다. 이 이름은 규정화되거나 규정화되지 않은 식별자이며, 각 컬럼에 대한 정의입니다. 테이블 공간에 단 하나의 테이블만 포함되도록, 각 테이블을 개별 테이블 공간에 저장할 수 있습니다. 테이블이 자주 삭제되고 작성될 경우, 개별 테이블 공간에 저장한 다음, 테이블 대신 테이블 공간을 삭제하는 것이 훨씬 더 효과적입니다. 사용자는 단일 테이블 공간에 여러 개의 테이블을 저장할 수도 있습니다. 파티션된 데이터베이스 환경에서, 선택된 테이블 공간은 테이블이 저장되어 있는 노드 그룹 및 데이터베이스 파티션도 정의합니다.

처음에는 테이블에 데이터가 들어 있지 않습니다. 데이터 행을 추가하려면, 다음 중 하나를 사용하십시오.

- *SQL 참조서*에 설명된 INSERT문
- *Command Reference*에 설명된 LOAD 또는 IMPORT 명령

테이블 안팎으로 이루어지는 데이터 이동에 대한 세부사항은 *데이터 이동 유틸리티 안내 및 참조서*에 있습니다.

데이터를 테이블에 추가하는 것은 변경사항을 로그하지 않고 수행될 수 있습니다. CREATE TABLE문의 NOT LOGGED INITIALLY 절은 테이블에 변경사항을 로그하지 못하게 합니다. 테이블이 작성되는 작업 단위(UOW)에서 INSERT, DELETE, UPDATE, CREATE INDEX, DROP INDEX 또는 ALTER TABLE 조작에 의해 테이블에 가해지는 모든 변경사항은 로그되지 않습니다. 기록은 다음 작업 단위(UOW)에서 시작됩니다.

테이블은 하나 이상의 컬럼 정의로 구성됩니다. 한 테이블에 대해 최대 500 컬럼을 정의할 수 있습니다. 컬럼은 엔티티의 속성을 나타냅니다. 컬럼에 있는 값들은 모두 동일한 유형의 정보입니다. *SQL* 참조서에서 자세한 내용을 참조하십시오.

주: 4KB 페이지 크기를 사용할 때에는 최대 500 컬럼입니다. 8KB, 16KB 또는 32KB 페이지 크기를 사용할 때에는 최대 1012 컬럼을 사용할 수 있습니다.

컬럼 정의 포함에는 컬럼 이름, 데이터 유형, 필요한 모든 널(NULL) 속성 또는 기본값(사용자가 선택한)이 있습니다.

컬럼 이름은 컬럼에 포함된 정보를 나타내면서 쉽게 기억할 수 있는 것이어야 합니다. 이 이름은 테이블 내에서 고유한 것이어야 하지만, 같은 이름을 다른 테이블에서 사용할 수도 있습니다. 이름 지정 규칙에 대해서는 447 페이지의 『오브젝트 이름』에서 자세한 내용을 참조하십시오.

컬럼의 데이터 유형은 컬럼 안에 있는 값의 길이와 컬럼에 유효한 데이터의 종류를 나타냅니다. 데이터베이스 관리 프로그램은 문자열, 수치, 날짜, 시간 및 대형 오브젝트(LOB) 데이터 유형을 사용합니다. 그래픽 문자열 데이터 유형은 다중 바이트 문자 세트를 사용하는 데이터베이스 환경에서만 사용 가능합니다. 또한, 컬럼은 사용자 정의 구별 유형으로 정의될 수 있으며, 이 유형에 대해서는 164 페이지의 『사용자 정의 유형(UDT) 작성』에서 설명됩니다.

기본 속성에 대한 스펙은 값이 제공되지 않을 경우에 사용해야 할 값을 지시합니다. 기본값을 지정하거나 시스템 정의 기본값을 사용할 수 있습니다. 널(NULL) 속성 스펙이 있는(또한 없는) 컬럼에 대해 기본값을 지정할 수 있습니다.

널(NULL) 속성 스펙은 컬럼이 널(NULL) 값을 포함하고 있는지의 여부를 나타냅니다.

제어 센터를 사용하여 테이블을 작성하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 → 마법사를 사용한 테이블을 선택하십시오.
3. 마법사의 단계에 따라 작업을 완료하십시오.

명령행을 사용하여 테이블을 작성하십시오.

```
CREATE TABLE <NAME>
  (<column_name> <data_type> <null_attribute>)
  IN <TABLE_SPACE_NAME>
```

다음은 RESOURCE 테이블 공간에 EMPLOYEE 테이블을 작성하는 CREATE TABLE문의 예입니다. 이 테이블은 샘플 데이터베이스에서 정의됩니다.

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(6)      NOT NULL PRIMARY KEY,
   FIRSTNAME  VARCHAR(12)  NOT NULL,
   MIDINIT    CHAR(1)     NOT NULL WITH DEFAULT,
   LASTNAME   VARCHAR(15)  NOT NULL,
   WORKDEPT   CHAR(3),
   PHONENO    CHAR(4),
   PHOTO      BLOB(10M)   NOT NULL)
  IN RESOURCE
```

테이블을 작성할 때에는 구조화 유형의 속성에 기초하여 테이블 컬럼을 보유하도록 선택할 수 있습니다. 이러한 테이블을 『입력된 테이블』이라고 합니다.

입력된 테이블은 다른 입력된 테이블에서 컬럼을 상속하도록 정의될 수 있습니다. 이러한 테이블을 『하위 테이블』이라고 하며 이 테이블이 상속하는 테이블을 『상위 테이블』이라고 합니다. 입력된 테이블과 하위 테이블의 조합을 『테이블 계층구조』라고 합니다. 테이블 계층구조에서 가장 위에 있는 테이블(상위 테이블이 없는 것)은 계층구조의 『루트 테이블』이라고 합니다.

다음 절에서는 앞의 예를 토대로 하여 사용자가 고려해야 할 기타 옵션을 다룹니다.

- 143 페이지의 『대형 오브젝트(LOB) 컬럼 고려사항』
- 145 페이지의 『고유 제한조건 정의』
- 146 페이지의 『참조 제한조건 정의』
- 150 페이지의 『테이블 점검 제한조건 정의』
- 154 페이지의 『입력된 테이블 작성』
- 155 페이지의 『입력된 테이블에 데이터 수록』
- 155 페이지의 『다중 테이블 공간에 테이블 작성』
- 156 페이지의 『파티션된 데이터베이스에서 테이블 작성』

- 166 페이지의 『사용자 정의 구조화 유형 작성』

또한 조회 결과에 기초하여 정의된 테이블을 작성할 수도 있습니다. 이러한 유형의 테이블을 요약 테이블이라고 합니다. 170 페이지의 『요약 테이블 작성』에서 자세한 내용을 참조하십시오.

대형 오브젝트(LOB) 컬럼 고려사항

대형 오브젝트(LOB) 컬럼이 들어 있는 테이블을 작성하기 전에, 다음과 같은 사항을 결정해야 합니다.

1. 변경사항을 LOB 컬럼에 기록하시겠습니까?

이 변경사항을 기록하지 않으려면, 테이블을 작성할 때 NOT LOGGED절을 지정하여 로그 기능을 해제해야 합니다.

```
CREATE TABLE EMPLOYEE
(EMPNO CHAR(6) NOT NULL PRIMARY KEY,
FIRSTNAME VARCHAR(12) NOT NULL,
MIDINIT CHAR(1) NOT NULL WITH DEFAULT,
LASTNAME VARCHAR(15) NOT NULL,
WORKDEPT CHAR(3),
PHONENO CHAR(4),
PHOTO BLOB(10M) NOT NULL NOT LOGGED)
IN RESOURCE
```

LOB 컬럼이 1GB보다 크면, 기록하지 마십시오(보통 사용자는 10MB보다 큰 LOB 컬럼을 기록하는 일은 없을 것입니다). 컬럼 정의에 지정된 다른 옵션처럼, 기록 옵션을 변경하는 방법은 테이블을 다시 작성하는 것입니다.

변경사항을 기록하지 않도록 선택했다라도, 구간 복원이 시스템 생성 오류의 결과로 인하여 응용프로그램 요구조건 상관없이, 변경사항이 구간 복원되도록 하기 위해 LOB 컬럼이 음영처리됩니다. 음영처리는 현재의 저장영역 페이지 내용이 겹쳐서 기록되지 않도록 하는 복구 기법입니다. 즉, 수정되지 않은 기존의 페이지는 『음영』 사본으로 보관됩니다. 트랜잭션 구간 복원을 지원하는 데 이들 사본이 더 이상 필요하지 않으면 폐기됩니다.

주: RESTORE 및 ROLLFORWARD 명령을 사용하여 데이터베이스를 복원할 때, 『로그되지 않고』 마지막 백업 이후에 기록된 LOB 데이터는 2진 0으로 대체됩니다.

2. LOB 컬럼에 필요한 공간을 최소화하시겠습니까?

사용자는 CREATE TABLE문의 COMPACT절을 사용하여 LOB 컬럼을 가능한 작게 작성할 수 있습니다. 예를 들면, 다음과 같습니다.

```
CREATE TABLE EMPLOYEE
(EMPNO CHAR(6) NOT NULL PRIMARY KEY,
FIRSTNME VARCHAR(12) NOT NULL,
MIDINIT CHAR(1) NOT NULL WITH DEFAULT,
LASTNAME VARCHAR(15) NOT NULL,
WORKDEPT CHAR(3),
PHONENO CHAR(4),
PHOTO BLOB(10M) NOT NULL NOT LOGGED COMPACT)
IN RESOURCE
```

최소 LOB 컬럼을 가진 테이블에 추가할 경우, 특히 LOB 값의 크기가 증가 되면(이미 만들어진 저장영역 조절 때문에) 성능비용이 있습니다.

LOB가 SMS 테이블 공간에 위치하고 회소 파일 할당을 지원하지 않는 OS/2 와 같은 플랫폼에서 COMPACT절 사용을 고려하십시오. 회소 파일 할당은 운영 체제가 실제 디스크 공간을 사용하는 방법과 연관이 있습니다. 회소 파일 할당을 지원하는 운영 체제는 회소 파일 할당을 지원하지 않는 운영체제와 비교하면 LOB 저장에 많은 물리 디스크 공간을 사용하지 않습니다. COMPACT 옵션은 회소 파일 할당 지원과는 별도로 훨씬 큰 물리 디스크 공간 『절약』을 가능하게 합니다. 어느 정도의 물리 디스크 공간 절약을 확보할 수 있기 때문에, COMPACT를 사용하려면 운영 체제가 회소 파일 할당을 지원하지 않는 경우의 COMPACT 사용을 고려해야 합니다.

주: DB2 시스템 카탈로그는 LOB 컬럼을 사용하고 이전 버전보다 더 많은 공간을 차지합니다.

3. DB2 시스템 카탈로그에 있는 LOB 컬럼을 포함하여 LOB 컬럼의 성능을 향상시키시겠습니까?

카탈로그 테이블에는 대형 오브젝트(LOB) 컬럼이 있습니다. LOB 데이터는 다른 데이터와 함께 버퍼 풀에 보관되지 않고, 필요할 때마다 디스크에서 읽습니다. 디스크로부터 읽을 경우, 카탈로그의 LOB 컬럼이 사용되는 DB2의 성능이 느려질 수 있습니다. 보통 파일 시스템은 자체적으로 데이터를 저장(또는 캐

형)할 장소를 가지고 있기 때문에, 파일 컨테이너에 구축된 SMS 테이블 공간 또는 DMS 테이블 공간은 앞에서 LOB가 참조될 경우 입출력을 피할 수 있게 합니다.

제한조건 정의

이 절에서는 제한조건의 정의 방법을 설명합니다.

- 『고유 제한조건 정의』
- 146 페이지의 『참조 제한조건 정의』
- 150 페이지의 『테이블 점검 제한조건 정의』.

제한조건에 대한 자세한 내용은 *관리 안내서: 계획에 있는 제한조건 강제 실시의 계획에 대한 절차 SQL 참조서를 참조하십시오.*

고유 제한조건 정의

고유 제한조건은 지정된 키의 모든 값이 고유한지 확인합니다. 테이블에는 기본 키로 정의된 많아야 한 가지의 고유 제한조건을 가진 많은 고유 제한조건이 있습니다.

CREATE TABLE 또는 ALTER TABLE문의 UNIQUE절을 가진 고유 제한조건을 정의하십시오. 고유 키는 둘 이상의 컬럼으로 구성될 수 있습니다. 한 테이블에 둘 이상의 고유 제한조건이 있을 수 있습니다. 그러나, 고유 제한조건은 하위 테이블에서 정의되지 않을 수도 있습니다.

일단 설정되면, 고유 제한조건은 INSERT 또는 UPDATE문이 테이블 데이터를 변경할 때 데이터베이스 관리 프로그램에 의해 자동으로 강제됩니다. 고유 제한조건은 고유 색인을 통해 강제됩니다.

ALTER TABLE문에 고유 제한조건이 정의되며, 해당 고유 키의 동일한 컬럼 세트에 대해 색인이 존재할 경우, 해당 색인은 고유 색인이 되어 제한조건에 의해 사용됩니다.

임의의 고유 제한조건을 선택하여 해당 제한조건을 기본 키로 사용할 수 있습니다. 기본 키는 참조 제한조건(다른 고유 제한조건과 함께)의 상위 키로 사용될 수 있

습니다. 테이블마다 한 개의 기본 키만 있습니다. CREATE TABLE 또는 ALTER TABLE문의 PRIMARY KEY절로 고유 키를 정의하십시오. 고유 키는 둘 이상의 컬럼으로 구성될 수 있습니다.

1차 색인은 고유 키의 값을 고유한 상태로 유지시킵니다. 테이블이 고유 키와 함께 작성되면, 데이터베이스 관리 프로그램은 해당 키에 대해 1차 색인을 작성합니다.

고유 제한조건으로 사용된 색인의 여러 성능 추가 정보는 다음 내용을 포함합니다.

- 색인이 있는 빈 테이블을 초기에 로드할 때, LOAD를 사용하는 것이 IMPORT를 사용하는 경우보다 나은 성능을 제공합니다. LOAD의 INSERT 또는 LOAD REPLACE 모드 중 어느 것을 사용하는지 여부는 문제가 되지 않습니다.
- 색인이 있는 기존 테이블에 상당한 양의 데이터를 추가할 경우(IMPORT INSERT 또는 LOAD INSERT 사용), LOAD는 IMPORT보다 약간 나은 성능을 제공합니다.
- 대량의 데이터를 초기에 로드하기 위해 IMPORT 명령을 사용하는 경우, 데이터가 가져오기 되거나 로드된 후에 고유 키를 작성하십시오. 이로 인해 테이블이 로드되는 동안 색인을 유지보수하는 데 생기는 오버헤드를 막을 수 있습니다. 그 결과, 색인은 최소량의 저장영역을 사용하게 됩니다.
- REPLACE 모드에서 LOAD 유틸리티를 사용하려면, 데이터를 로드하기 전에 고유 키를 작성하십시오. 이 경우, 로드중에 색인을 작성하는 것이 로드 후에 CREATE INDEX문을 사용하는 것보다 효율적입니다.

참조 제한조건 정의

참조 무결성은 테이블과 컬럼 정의에 참조 제한조건을 추가함으로써 이루어집니다. 참조 제한조건은 FOREIGN KEY 절과, CREATE TABLE 또는 ALTER TABLE문의 REFERENCES 절로 설정됩니다. 입력된 테이블에 대한 참조 제한조건의 효과에 대한 자세한 내용은 SQL 참조서와 입력된 테이블인 상위 테이블을 참조하십시오.

외부 키 식별은 한 테이블 행 내의 값 또는 두 테이블 행 사이의 값에 대한 제한조건을 시행합니다. 데이터베이스 관리 프로그램은 테이블 정의에 지정된 제한조건

을 점검하여 그에 따라 관계를 유지보수합니다. 목적은 하나의 데이터베이스 오브젝트가 다른 데이터베이스 오브젝트를 참조할 때마다 무결성을 유지보수하는 것입니다.

예를 들어, 기본 키와 외부 키는 각각 부서 번호 컬럼을 가지고 있습니다. EMPLOYEE 테이블의 경우에 컬럼 이름은 WORKDEPT이고, DEPARTMENT 테이블의 경우에는 DEPTNO입니다. 두 테이블간의 관계는 다음 제한조건으로 정의됩니다.

- EMPLOYEE 테이블에 있는 각 사원에 대해 하나의 부서 번호만 있고, 해당 번호가 DEPARTMENT 테이블에 존재합니다.
- EMPLOYEE 테이블의 각 행은 DEPARTMENT 테이블에 있는 하나의 행에만 관련됩니다. 테이블간에는 고유한 관계가 있습니다.
- WORKDEPT에 대해 널(NULL)이 아닌 값을 가지는 EMPLOYEE 테이블 내의 각 행은 DEPARTMENT 테이블의 DEPTNO 컬럼에 있는 행과 관련이 있습니다.
- DEPARTMENT 테이블이 상위 테이블이고 EMPLOYEE 테이블이 종속 테이블입니다.

상위 테이블인 DEPARTMENT를 정의하는 SQL문은 다음과 같습니다.

```
CREATE TABLE DEPARTMENT
  (DEPTNO   CHAR(3)      NOT NULL,
   DEPTNAME VARCHAR(29) NOT NULL,
   MGRNO    CHAR(6),
   ADMRDEPT CHAR(3)     NOT NULL,
   LOCATION CHAR(16),
   PRIMARY KEY (DEPTNO))
IN RESOURCE
```

종속 테이블인 EMPLOYEE를 정의하는 SQL문은 다음과 같습니다.

```
CREATE TABLE EMPLOYEE
  (EMPNO     CHAR(6)      NOT NULL PRIMARY KEY,
   FIRSTNAME VARCHAR(12) NOT NULL,
   LASTNAME  VARCHAR(15) NOT NULL,
   WORKDEPT  CHAR(3),
   PHONENO   CHAR(4),
```

```
PHOTO BLOB(10m) NOT NULL,  
FOREIGN KEY DEPT (WORKDEPT)  
REFERENCES DEPARTMENT ON DELETE NO ACTION)  
IN RESOURCE
```

DEPARTMENT 테이블의 고유 키로 DEPTNO 컬럼을 지정하고 EMPLOYEE 테이블의 외부 키로 WORKDEPT를 지정함으로써, 사용자는 WORKDEPT 값에 대해 참조 제한조건을 정의하게 됩니다. 이 제한조건은 두 테이블의 값 사이에 참조 무결성을 강제합니다. 이 경우, EMPLOYEE 테이블에 추가되는 사원은 DEPARTMENT 테이블에서 발견할 수 있는 부서 번호를 가지고 있어야 합니다.

사원 테이블의 참조 제한조건에 대한 삭제 규칙은 NO ACTION이며, 이는 해당 부서에 사원이 있는 경우에 DEPARTMENT 테이블에서 부서를 삭제시킬 수 없음을 의미합니다.

앞의 예에서는 참조 제한조건을 추가하기 위해 CREATE TABLE문을 사용했지만, ALTER TABLE문을 사용할 수도 있습니다. 216 페이지의 『구조 및 내용에서 테이블 수정』에서 자세한 내용을 참조하십시오.

기타 예: 앞의 예에 사용된 것과 같은 테이블 정의가 사용됩니다. 또한 DEPARTMENT 테이블이 EMPLOYEE 테이블보다 먼저 작성됩니다. 각 부서에 관리자가 있고, 해당 관리자는 EMPLOYEE 테이블에 나열되어 있습니다. DEPARTMENT 테이블의 MGRNO는 실제로 EMPLOYEE 테이블의 외부 키입니다. 이 참조 순환으로 이 제한조건은 심각한 문제가 되지 않습니다. 나중에 외부 키를 추가할 수 있습니다(219 페이지의 『기본 키 및 외부 키의 추가』에서 자세한 내용을 참조하십시오). 또한 CREATE SCHEMA문을 사용하여 동시에 EMPLOYEE 및 DEPARTMENT 테이블을 작성할 수 있습니다(SQL 참조서의 예를 참조하십시오).

FOREIGN KEY절: 외부 키는 같은 테이블 또는 다른 테이블의 기본 키나 고유 키를 참조합니다. 외부 키 할당은 지정된 참조 제한조건에 따라 참조 무결성이 유지보수됨을 나타냅니다. 사용자는 CREATE TABLE 또는 ALTER TABLE문의 FOREIGN KEY절로 외부 키를 정의합니다.

외부 키의 컬럼 번호는 해당하는 상위 테이블의 기본 또는 고유 제한조건(상위 키라고 함)의 컬럼 번호와 일치해야 합니다. 또한, 키 컬럼 정의의 해당 부분은 동일

한 데이터 유형과 길이를 가지고 있어야 합니다. 외부 키는 **제한조건 이름**을 할당 받을 수 있습니다. 사용자가 이름을 할당하지 않으면, 자동으로 할당됩니다. 쉽게 사용하려면, **제한조건 이름**을 지정하고 시스템 생성 이름을 사용하지 않는 것이 좋습니다.

복합 외부 키의 값은 외부 키의 각 컬럼 값이 상위 키의 해당 컬럼 값과 같을 경우, 상위 키의 값과 일치합니다. 널(NULL) 값이 들어 있는 외부 키는 상위 키의 값과 일치할 수 없는데, 이는 상위 키는 널(NULL) 값을 가질 수 없는 것으로 정의되어 있기 때문입니다. 그러나, 널(NULL) 외부 키 값은 외부 키의 널(NULL) 이외 부분의 값에 관계없이 항상 유효합니다.

다음과 같은 규칙이 외부 키 정의에 적용됩니다.

- 한 테이블이 여러 개의 외부 키를 가질 수 있습니다.
- 외부 키는 키의 모든 부분이 널(NULL) 입력 가능한 상태일 때 널(NULL) 입력 가능합니다.
- 어떤 부분이 널(NULL) 값일 경우, 외부 키 값은 널(NULL)이 됩니다.

REFERENCES절: REFERENCES절은 테이블간의 관계에서 상위 테이블을 식별하고, 필요한 제한조건을 정의합니다. 사용자는 컬럼 정의에 이를 포함시키거나, CREATE TABLE 또는 ALTER TABLE문에서 FOREIGN KEY절을 동반한 개별 절로서 이를 포함시킬 수 있습니다.

REFERENCES절을 컬럼 제한조건으로서 지정하면, 내재적인 컬럼 목록이 컬럼 이름으로 구성됩니다. 여러 컬럼에 개별 REFERENCES절이 들어 갈 수 있고, 단일 컬럼에 둘 이상의 REFERENCES절이 들어 갈 수 있음에 유의하십시오.

REFERENCES절에는 삭제 규칙이 포함되어 있습니다. 예에서는 ON DELETE NO ACTION 규칙이 사용되며, 이는 부서에 사원이 할당되어 있는 경우에는 어느 부서도 삭제될 수 없음을 의미합니다. 다른 삭제 규칙으로는 ON DELETE CASCADE, ON DELETE SET NULL 및 ON DELETE RESTRICT가 있습니다. **관리 안내서: 계획**에서 참조 무결성을 구현할 때 DELETE 규칙에 대한 자세한 내용을 참조하십시오.

유틸리티 조작에 대한 포함: LOAD 유틸리티는 자체 참조 및 종속 테이블을 점검 보류 상태로 둬으로써 테이블에 대한 제한조건 점검을 오프시킵니다. LOAD 유틸

탈리기가 완료되면, 오픈된 모든 테이블에 대해 제한조건 점검을 다시 작동시켜야 합니다. 예를 들어, DEPARTMENT 및 EMPLOYEE 테이블만 점검 보류 상태에 있을 경우, 다음 명령을 실행할 수 있습니다.

```
SET INTEGRITY FOR DEPARTMENT, EMPLOYEE IMMEDIATE CHECKED
```

IMPORT 유틸리티는 다음과 같은 방법으로 참조 제한조건의 영향을 받습니다.

- REPLACE 및 REPLACE CREATE 함수는 오브젝트 테이블에 자신을 제외한 다른 종속 테이블이 있는 경우에는 허용되지 않습니다.

이러한 함수를 사용하려면, 먼저 상위 테이블에 있는 모든 외부 키를 삭제하십시오. 반입이 완료되면, ALTER TABLE문으로 외부 키를 다시 작성하십시오.

- 자체 참조 제한조건을 가진 테이블로의 가져오기 성공 여부는 행이 가져오는 순서에 따라 달라집니다.

테이블 점검 제한조건 정의

테이블 점검 제한조건은 테이블 점검 제한조건이 정의된 테이블의 각 행에 대해 시행되는 검색 조건을 지정합니다. 테이블이 작성되거나 변경될 때 테이블과 점검 제한조건 정의를 연관시킴으로써 테이블에 대해 테이블 점검 제한조건을 작성할 수 있습니다. 이 제한조건은 INSERT 또는 UPDATE문이 테이블에 있는 데이터를 수정할 때 자동으로 활성화됩니다. 테이블 점검 제한조건은 DELETE문 또는 SELECT문에 아무런 영향을 주지 않습니다. 점검 제한조건은 입력된 테이블에 연관될 수 없습니다.

제한조건 이름은 동일한 CREATE TABLE문에서 지정된 다른 제한조건과는 달라야 합니다. 사용자가 제한조건 이름을 지정하지 않으면, 시스템이 제한조건에 대해 18자의 고유 식별자를 생성합니다.

테이블 점검 제한조건은 키 고유성 또는 참조 무결성 제한조건에서 다루지 않는 데이터 무결성 규칙을 시행하는 데 사용됩니다. 어떤 경우에는 테이블 점검 제한조건이 도메인 점검을 구현하는 데 사용될 수도 있습니다. CREATE TABLE문에서 발행된 다음 제한조건은 모든 활동의 시작 날짜가 동일한 활동의 종료 날짜보다 늦지 않도록 합니다.

```
CREATE TABLE EMP_ACT
  (EMPNO      CHAR(6)      NOT NULL,
   PROJNO     CHAR(6)      NOT NULL,
```

```

ACTNO      SMALLINT      NOT NULL,
EMPTIME    DECIMAL(5,2),
EMSTDATE   DATE,
EMENDATE   DATE,
CONSTRAINT ACTDATES CHECK(EMSTDATE <= EMENDATE) )
IN RESOURCE

```

앞의 예에서는 테이블 점검 제한조건을 추가하기 위해 CREATE TABLE문을 사용했지만, ALTER TABLE문도 사용될 수 있습니다. 216 페이지의 『구조 및 내용에서 테이블 수정』에서 자세한 내용을 참조하십시오.

생성된 컬럼을 새 테이블에 정의

생성된 컬럼은 저장된 값이 삽입 또는 갱신 조작을 통해 지정되기 보다는 표현식을 사용하여 계산되는 기본 테이블에 정의됩니다. 특정 표현식이나 술어가 항상 사용될 것으로 알려진 테이블을 작성할 때, 하나 이상의 생성된 컬럼을 해당 테이블에 추가할 수 있습니다. 생성된 컬럼을 사용하여 테이블 데이터를 조회할 때 성능을 향상시킬 수 있습니다.

예를 들어, 성능이 중요할 때 표현식의 평가가 비용이 많이 들 수 있는 두가지 경우가 있습니다.

1. 표현식의 평가가 조회 중 여러 번 수행된 경우.
2. 계산이 복잡한 경우.

조회 성능을 개선시키기 위해, 표현식의 결과를 포함하는 추가적인 컬럼을 정의할 수 있습니다. 그런 다음 동일한 표현식을 포함하는 조회를 발행할 때, 생성되는 컬럼이 직접 사용될 수 있거나, 최적화 알고리즘의 조회 재작성 구성요소가 표현식을 생성된 컬럼으로 대체할 수 있습니다.

또한 생성된 컬럼에서 고유하지 않은 색인을 작성하는 것이 가능합니다.

조회가 둘 이상의 테이블의 데이터의 조인에 관련된 곳에서 생성된 컬럼을 추가하면, 최적화 알고리즘이 가능한 더 나은 조인 전략을 선택할 수 있습니다.

다음은 CREATE TABLE문에서 생성된 컬럼을 정의하는 예입니다.

```

CREATE TABLE t1 (c1 INT,
                  c2 DOUBLE,
                  c3 DOUBLE GENERATED ALWAYS AS (c1 + c2)
                  c4 GENERATED ALWAYS AS
                  (CASE WHEN c1 > c2 THEN 1 ELSE NULL END))

```

이 테이블을 작성한 후에, 색인은 생성된 컬럼을 사용하여 작성될 수 있습니다. 예를 들면, 다음과 같습니다.

```
CREATE INDEX i1 ON t1(c4)
```

조치는 생성된 컬럼의 이점을 이용할 수 있습니다. 예를 들면, 다음과 같습니다.

```
SELECT COUNT(*) FROM t1 WHERE c1 > c2
```

다음과 같이 쓸 수 있습니다.

```
SELECT COUNT(*) FROM t1 WHERE c4 IS NOT NULL
```

또다른 예는 다음과 같습니다.

```
SELECT c1 + c2 FROM t1 WHERE (c1 + c2) * c1 > 100
```

다음과 같이 쓸 수 있습니다.

```
SELECT c3 FROM t1 WHERE c3 * c1 > 100
```

생성 컬럼을 사용하여 조회 성능을 높일 수 있습니다. 결과적으로, 생성된 컬럼은 테이블이 작성되어 데이터 상주된 후에 추가될 수 있습니다. 자세한 정보는 140 페이지의 『테이블 작성 및 데이터 수록』를 참조하십시오.

사용자 정의 임시 테이블 작성

DECLARE GLOBAL TEMPORARY TABLE문을 사용하여 임시 테이블을 정의할 수 있습니다. 명령문은 응용프로그램 내에서 사용됩니다. 사용자 정의 임시 테이블만이 응용프로그램이 데이터베이스에서 연결해제될 때까지 지속됩니다.

이 테이블의 설명은 지속적이지 않게 하는 시스템 카탈로그에 표시되지 않으며 다른 응용프로그램과 공유될 수 없습니다.

이 테이블을 사용하는 응용프로그램이 데이터베이스에서 종료하거나 연결해제될 때 테이블에 있는 임의의 데이터가 삭제되며 테이블이 내재적으로 제거됩니다.

임시 테이블을 정의할 수 있는 방법의 예제는 다음과 같습니다.

```
DECLARE GLOBAL TEMPORARY TABLE gbl_temp
  LIKE emp1tab1
  ON COMMIT DELETE ROWS
  NOT LOGGED
  IN usr_tbsp
```

이 명령문은 gbl_temp라고 하는 사용자 임시 테이블을 작성합니다. 사용자 임시 테이블은 emp1tab1의 컬럼과 동일한 이름과 설명을 가진 컬럼으로 정의됩니다. 내재된 정의만이 컬럼 이름, 데이터 유형, 널 기능 특성 및 컬럼 기본 값 속성을 포함합니다. 고유 제한조건, 외부 키 제한조건, 트리거 및 색인을 포함하는 다른 모든 컬럼 속성은 정의되지 않습니다. COMMIT 조작이 수행될 때, WITH HOLD 커서가 테이블에서 열려있지 않은 경우 테이블에 있는 모든 데이터는 삭제됩니다. 사용자 임시 테이블에 작성한 변경사항은 로그되지 않습니다. 사용자 임시 테이블은 지정된 사용자 임시 테이블 공간에 위치합니다. 이 테이블 공간이 존재해야 합니다. 그렇지 않으면 이 테이블의 선언은 실패합니다.

DECLARE GLOBAL TEMPORARY TABLE문에 관한 추가 정보는 *SQL 참조*에서 자세한 내용을 참조하십시오.

주: 사용자 정의 임시 테이블을 지원하지 않습니다.

- LOB 유형 컬럼(또는 LOB에 근거한 구별 유형 컬럼)
- 사용자 정의 유형 컬럼
- LONG VARCHAR 컬럼
- DATALINK 컬럼

식별 컬럼을 새 테이블에 정의

식별 컬럼은 테이블에 추가된 각 행의 보증된 고유한 숫자값을 자동으로 생성하는 방법을 DB2에 제공합니다. 테이블에 추가될 각 행을 고유하게 식별하는 데 필요한 테이블을 사용자가 아는 곳에 작성할 때, 식별 컬럼을 테이블에 추가할 수 있습니다.

테이블을 작성하면, 식별 컬럼을 포함하기 위해 테이블 설명을 변경할 수 없습니다.

식별 컬럼의 스펙을 참작하는 CREATE TABLE문에 AS IDENTITY 절이 있습니다.

다음은 CREATE TABLE문에서 식별 컬럼을 정의하는 예입니다.

```
CREATE TABLE table (col1 INT,  
                    col2 DOUBLE,  
                    col3 INT NOT NULL GENERATED ALWAYS AS IDENTITY  
                    (START WITH 100, INCREMENT BY 5))
```

이 예에서 세번째 컬럼은 식별 컬럼입니다. 또한, 컬럼을 추가할 때 각 행을 고유하게 식별하기 위해 컬럼에서 사용되는 값을 지정할 수 있습니다. 입력된 첫번째 행은 컬럼에 위치한 값 『100』을 가지며, 테이블에 추가되는 차후의 모든 행은 5씩 증가되는 연관된 값을 갖습니다.

식별 컬럼의 일부 추가적인 예 사용은 주문 번호, 사원 번호, 재고 번호 또는 변환 기록 번호입니다. DB2가 생성하는 식별 컬럼 값은 ALWAYS 또는 BY DEFAULT 입니다.

GENERATED ALWAYS로서 정의되는 식별 컬럼은 고유한 것으로 보장됩니다. 사용되는 값은 항상 DB2가 생성합니다. 응용프로그램은 명시적 값을 제공하도록 허용되지 않습니다. GENERATED BY DEFAULT로서 정의된 식별 컬럼은 식별 컬럼의 값을 명시적으로 제공하는 방법을 응용프로그램에 제공합니다. 응용프로그램이 값을 제공하지 않는 경우, DB2는 값을 생성합니다. 응용프로그램이 값을 제어하므로, DB2가 값의 고유성을 보장할 수 없습니다. GENERATED BY DEFAULT 절은 기존 테이블의 내용을 복사하거나 테이블을 로드 해제 및 다시 로드할 의도가 있는 곳에서 데이터 보급을 위해 사용됩니다.

주: 식별 컬럼은 파티션된 데이터베이스 환경에서 현재 지원되지 않습니다.

SQL 참조서에서 새로운 테이블의 식별 컬럼을 정의하는 자세한 내용을 참조하십시오.

입력된 테이블 작성

CREATE TABLE문의 변형을 사용하여 입력된 테이블을 작성할 수 있습니다. 응용프로그램 개발 안내서에서 입력된 테이블에 필요한 자세한 내용을 참조하십시오.

입력된 테이블에 데이터 수록

구조화 유형을 작성하고 해당 테이블 및 하위테이블을 작성한 후에 입력된 테이블에 데이터 수록을 할 수 있습니다. **응용프로그램 개발 안내서**에서 입력된 테이블에 필요한 자세한 내용을 참조하십시오.

계층구조 테이블

계층구조 테이블은 입력된 테이블 계층구조의 구현에 관련된 테이블입니다. 이는 계층구조의 루트 테이블과 동시에 작성됩니다. **응용프로그램 개발 안내서**에서 계층구조 테이블에 필요한 자세한 내용을 참조하십시오.

다중 테이블 공간에 테이블 작성

테이블 데이터는 테이블 색인과 테이블과 연관된 긴 컬럼 데이터와 동일한 테이블 공간에 저장될 수 있습니다. 또한 나머지 테이블 데이터의 테이블 공간과는 분리하여, 개별 테이블 공간에 색인을 위치시키고, 개별 테이블 공간에 긴 컬럼 데이터를 위치시킬 수 있습니다. CREATE TABLE문을 수행하기 전에 모든 테이블 공간이 있어야 합니다. 테이블을 일부 분리하는 것은 DMS 테이블 공간을 사용하여서만 가능합니다.

제어 센터를 사용하여 여러 테이블 공간에 테이블을 작성하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 → 마법사를 사용한 테이블을 선택하십시오.
3. 테이블 이름을 입력하고 다음을 클릭하십시오.
4. 사용자 테이블의 컬럼을 선택하십시오.
5. 테이블 공간 페이지에서, 별도의 색인 공간 사용 및 별도의 long 공간 사용을 클릭하고, 정보를 지정하고 완료를 클릭하십시오.

명령행을 사용하여 여러 테이블 공간에 테이블을 작성하려면 다음을 입력하십시오.

```
CREATE TABLE <name>  
  (<column_name> <data_type> <null_attribute>)  
  IN <table_space_name>  
  INDEX IN <index_space_name>  
  LONG IN <long_space_name>
```

다음 예에서는 테이블의 여러부분을 서로 다른 테이블 공간에 저장하기 위해 EMP_PHOTO 테이블이 어떻게 작성되는지를 보여줍니다.

```
CREATE TABLE EMP_PHOTO
  (EMPNO          CHAR(6)          NOT NULL,
   PHOTO_FORMAT  VARCHAR(10)     NOT NULL,
   PICTURE       BLOB(100K) )
IN RESOURCE
INDEX IN RESOURCE_INDEXES
LONG IN RESOURCE_PHOTO
```

이 예는 EMP_PHOTO 데이터가 다음과 같이 저장되도록 만듭니다.

- EMP_PHOTO 테이블용으로 작성된 색인은 RESOURCES_INDEXES 테이블 공간에 저장됩니다.
- PICTURE 컬럼에 대한 데이터는 RESOURCE_PHOTO 테이블 공간에 저장됩니다.
- EMPNO 및 PHOTO_FORMAT 컬럼에 대한 데이터는 RESOURCE 테이블 공간에 저장됩니다.

관리 안내서: 계획에서 단일 테이블의 다중 DMS 테이블 공간의 사용에 대한 자세한 내용을 참조하십시오.

SQL 참조서에서 자세한 내용을 참조하십시오.

파티션된 데이터베이스에서 테이블 작성

물리적으로 나누어지거나 파티션된 테이블을 작성하기 전에, 다음과 같은 사항을 결정해야 합니다.

- 테이블 공간은 둘 이상의 데이터베이스 파티션을 확장시킬 수 있습니다. 스캔하는 파티션 수는 노드 그룹의 파티션 수에 따라 결정됩니다.
- 테이블은 동일한 테이블 공간에 배치하거나, 첫번째 테이블 공간과 함께 동일한 노드 그룹과 연관된 다른 테이블 공간에 배치함으로써 할당될 수 있습니다. *관리 안내서:* 계획에서 자세한 내용을 참조하십시오.

파티션된 데이터베이스 환경에서 테이블을 작성할 경우에는 *파티션* 키라는 추가 옵션이 있습니다. 파티션 키는 테이블의 정의 부분인 키입니다. 이 키는 데이터의 각 행이 저장된 파티션을 결정합니다.

이는 나중에 변경될 수 없기 때문에 적합한 파티션 키를 선택하는 것이 중요합니다. 다른 고유 색인(고유 키 또는 기본 키)은 파티션 키의 상위집합으로 정의되어야 합니다. 다시 말해서, 파티션 키가 정의되면, 고유 키와 기본 키는 파티션 키(더 많은 컬럼을 가지고 있음)처럼 동일한 모든 컬럼을 포함해야 합니다.

파티션 키를 확실히 지정하지 않은 경우, 다음의 기본값이 사용됩니다. 기본 파티션 키가 적합한지 확인하십시오

- 기본 키가 CREATE TABLE문에서 지정되지 않은 경우, 기본 키의 첫번째 컬럼은 파티션 키로 사용됩니다.
- 기본 키가 없을 경우, 긴 필드가 아닌 첫번째 컬럼이 사용됩니다.
- 어떠한 컬럼도 기본 파티션 키의 요구사항을 충족시킬 수 없는 경우, 이 옵션 없이 테이블이 작성됩니다(단일 파티션 노드 그룹의 경우에만 허용됨).

다음은 예입니다.

```
CREATE TABLE MIXREC (MIX_CNTL INTEGER NOT NULL,
                     MIX_DESC CHAR(20) NOT NULL,
                     MIX_CHR CHAR(9) NOT NULL,
                     MIX_INT INTEGER NOT NULL,
                     MIX_INTS SMALLINT NOT NULL,
                     MIX_DEC DECIMAL NOT NULL,
                     MIX_FLT FLOAT NOT NULL,
                     MIX_DATE DATE NOT NULL,
                     MIX_TIME TIME NOT NULL,
                     MIX_TMSTMP TIMESTAMP NOT NULL)
IN MIXTS12
PARTITIONING KEY (MIX_INT) USING HASHING
```

앞의 예에서, 테이블 공간은 MIXTS12이며 파티션 키는 MIX_INT입니다. 파티션 키가 확실히 지정되어 있지 않으면, 테이블 공간은 MIX_CNTL입니다(기본 키가 지정되어 있지 않고 파티션 키가 정의되어 있지 않을 경우, 파티션 키는 목록에서 첫 번째의 길지 않은 컬럼이 됩니다).

테이블 행 및 그 행에 대한 모든 정보는 항상 동일한 데이터베이스 파티션에 상주합니다.

테이블의 한 파티션의 최대 크기는 64GB 또는 사용 가능한 디스크 공간 중 더 적은 쪽입니다(이 경우, 테이블 공간 크기를 4KB 페이지 크기로 가정합니다). 테이블의 크기는 데이터베이스 파티션의 64GB(또는 사용 가능한 디스크 공간)에 데

이터베이스 파티션 수를 곱한 만큼 클 수 있습니다. 테이블 공간의 페이지 크기가 18KB인 경우, 테이블의 크기는 데이터베이스 파티션의 128GB(또는 사용 가능한 디스크 공간)에 데이터베이스 파티션 수를 곱한 만큼 클 수 있습니다. 테이블 공간에 대한 페이지 크기가 16KB이면, 테이블의 크기는 256GB(또는 사용 가능한 디스크 공간)에 데이터베이스 파티션 수를 곱한 만큼 클 수 있습니다. 테이블 공간에 대한 페이지 크기가 32KB이면, 테이블의 크기는 512GB(또는 사용 가능한 디스크 공간)에 데이터베이스 파티션 수를 곱한 만큼 클 수 있습니다.

트리거 작성

트리거는 지정된 기본 테이블 및 유형 테이블에서 INSERT, UPDATE 또는 DELETE절과함께 실행되는 조치 또는 이러한 절에 의해 트리거되는 일련의 조치를 정의합니다. 트리거는 다음 경우에 사용됩니다.

- 입력 데이터의 유효성 검사를 위해
- 새로 삽입된 행의 값을 생성하기 위해
- 상호 참조 목적으로 다른 테이블을 읽기 위해
- 감사 추적 목적으로 다른 테이블에 기록하기 위해

별명이 있는 트리거를 사용할 수 없습니다.

사용자는 무결성 또는 비즈니스 규칙의 일반 양식을 지원하기 위해 트리거를 사용할 수 있습니다. 예를 들어, 트리거는 주문을 받아들이기 전에 고객의 신용 한도를 점검하거나 요약 데이터 테이블을 갱신할 수 있습니다.

트리거 사용의 이점은 다음과 같습니다.

- 보다 빠른 응용프로그램 개발: 트리거는 데이터베이스에 저장되므로 모든 응용 프로그램에서 이루어지는 조치를 코드화할 필요가 없습니다.
- 보다 용이한 유지보수: 일단 트리거가 정의되면, 이를 작성한 테이블이 액세스 될 때 자동으로 호출됩니다.
- 비즈니스 규칙의 전역 시행: 비즈니스 규정이 변경될 경우, 트리거만 변경하고 각각의 응용프로그램은 변경하지 않아도 됩니다.

제어 센터를 사용하여 트리거를 작성하십시오.

1. 트리거 폴더를 찾을 때까지 오브젝트 트리러 확장하십시오.
2. 트리거 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 작성을 선택하십시오.
3. 트리거용 정보를 지정하십시오.
4. 트리거가 호출하려는 조치를 지정하고 확인을 클릭하십시오.

명령행을 사용하여 트리거를 작성하십시오.

```
CREATE TRIGGER <name>  
    <action> ON <table_name>  
    <operation>  
    <triggered_action>
```

다음의 SQL문은 신입 사원이 입사할 때마다 사원의 수를 증가시키는 트리거를 작성하는 것으로, EMPLOYEE 테이블에 행이 추가될 때마다 COMPANY_STATS 테이블의 사원 번호(NBEMP) 컬럼에 1을 추가합니다.

```
CREATE TRIGGER NEW_HIRED  
    AFTER INSERT ON EMPLOYEE  
    FOR EACH ROW MODE DB2SQL  
    UPDATE COMPANY_STATS SET NBEMP = NBEMP+1;
```

트리거 본문에는 하나 이상의 SQL문, 즉 INSERT, 검색된 UPDATE, 검색된 DELETE, 전체 선택, SET 전이 변수 및 SIGNAL SQLSTATE가 포함될 수 있습니다. 트리거는 참조하는 INSERT, UPDATE 또는 DELETE문 이전 또는 이후에 활성화될 수 있습니다. CREATE TRIGGER문에 대한 자세한 구문 정보는 *SQL 참조서*에서 자세한 내용을 참조하십시오. 트리거의 작성 및 사용에 대해서는 *응용프로그램 개발 안내서*에서 자세한 내용을 참조하십시오.

주: 트리거가 사전 트리거이면, 트리거 조치에 의해 지정되는 컬럼 이름은 식별 컬럼 이외에 생성된 컬럼일 수 없습니다. 즉, 생성된 식별 값은 사전 트리거에 가시적입니다.

트리거 종속성

일부 다른 오브젝트에 대한 트리거의 모든 종속성이 SYSCAT.TRIGDEP 카탈로그에 기록됩니다. 트리거는 여러 오브젝트에 따라 결정될 수 있습니다. 이들 오브젝트 및 종속 트리거는 SQL 참조서의 DROP문 설명 부분에 자세히 기술되어 있습니다.

이들 오브젝트 중 하나가 삭제되면, 트리거는 조작 불가능하게 되지만 정의는 카탈로그에 남아 있습니다. 이 트리거를 다시 확인하려면, 카탈로그에서 정의를 검색하여 새로운 CREATE TRIGGER문을 제출해야 합니다.

트리거가 삭제되면, SYSCAT.TRIGGERS 카탈로그 뷰에서 트리거에 관한 설명이 삭제되고 SYSCAT.TRIGDEP 카탈로그 뷰에서 모든 종속성이 삭제됩니다. 트리거에 대해 UPDATE, INSERT 또는 DELETE 종속성을 가지고 있는 모든 패키지도 무효가 됩니다.

종속 오브젝트가 뷰이고 뷰가 조작 불가능하게 되면, 트리거도 조작 불가능으로 표시됩니다. 조작 불가능이라고 표시된 트리거에 종속된 임의의 패키지는 유효하지 않습니다(244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오).

사용자 정의 함수(UDF) 및 메소드 작성

사용자 정의 함수(UDF)는 SQL의 내장 함수에서 제공하는 지원에 확장되고 추가되며, 내장 함수가 사용되는 곳이면 어디든지 사용될 수 있습니다. 다음과 같이 UDF를 작성할 수 있습니다.

- 외부 함수 : 프로그래밍 언어로 작성됨
- 전래 함수. 일부 기존의 다른 함수로부터 물려받아 구현되는 함수

UDF에는 다음과 같은 세 가지 유형이 있습니다.

스칼라 호출 때마다 한 가지 값의 응답을 리턴합니다. 예를 들어, 내장 함수 SUBSTR()는 스칼라 함수입니다. 스칼라 UDF는 외부에 있거나 전래될 수 있습니다.

컬럼 비슷한 값(컬럼)의 세트에서 한 가지 값의 응답을 리턴합니다. 또한, 종종

DB2에서는 총계 함수라고 합니다. 컬럼 함수의 예로는 내장 함수 AVG()가 있습니다. 외부 컬럼 UDF는 DB2에 정의될 수 없지만, 내장된 컬럼 함수 중 하나에서 전래된 컬럼 UDF는 정의될 수 있습니다. 이 사항은 구별 유형에 유용합니다.

예를 들어, 기본 유형 INTEGER로 정의된 구별 유형 SHOESIZE가 있는 경우, 내장 함수 AVG(INTEGER)에서 전래된 UDF AVG(SHOESIZE)는 정의될 수 있으며 컬럼 함수가 됩니다.

테이블 테이블을 참조하는 SQL문으로 리턴합니다. 테이블 함수는 SELECT문의 FROM절에서만 참조됩니다. 이 함수를 사용하여 SQL 언어 처리 기능을 DB2 데이터가 아닌 데이터에 적용하거나, 데이터를 DB2 테이블로 변환할 수 있습니다.

예를 들어, 테이블 함수는 파일을 선택하여 테이블로 변환하고 World Wide Web의 샘플 데이터를 표로 만들거나 Lotus Notes 데이터베이스에 액세스하고 날짜, 송신자, 메일 메시지 텍스트와 같은 정보를 리턴할 수 있습니다. 이 정보는 데이터베이스의 다른 테이블과 조인될 수 있습니다.

테이블 함수는 외부 함수만 될 수 있습니다. 전래 함수는 될 수 없습니다.

기존의 UDF 정보는 SYSCAT.FUNCTIONS 및 SYSCAT.FUNCPARMS 카탈로그 뷰에 기록됩니다. 시스템 카탈로그에는 UDF에 대한 실행 가능한 코드가 들어 있지 않습니다(그러므로, 백업 플랜 및 복구 플랜을 작성하려면, 실행가능한 UDF 관리 방법을 고려해야 합니다).

UDF의 성능에 대한 통계는 SQL문을 컴파일할 때 중요합니다. 시스템 카탈로그에서 UDF 통계를 갱신하는 방법에 대해서는 **관리 안내서: 성능의 『사용자 정의 함수에 대한 통계 갱신』**을 참조하십시오.

CREATE FUNCTION문을 사용하여 특정한 응용프로그램에 맞는 UDF를 작성하는 방법에 대해서는 **응용프로그램 개발 안내서**에서 자세한 내용을 참조하십시오. UDF 구문에 대해서는 **SQL 참조서**에서 자세한 내용을 참조하십시오.

함수 맵핑 작성

연합 데이터베이스에서 지역 함수 또는 지역 함수 템플릿(163 페이지의 『함수 템플릿 작성』에서 설명)를 하나 이상의 데이터 소스에서 함수와 함께 맵할 때 함수 맵핑을 작성하십시오. 기본 함수 맵핑은 많은 데이터 소스 함수에 제공됩니다.

함수 맵핑은 다음과 같은 경우에 유용합니다.

- 내장된 새 함수가 데이터 소스에서 사용 가능해집니다.
- 데이터 소스에서 사용자 정의 함수를 지역 함수에 맵핑해야 합니다.
- 응용프로그램은 기본 맵핑으로 제공되는 다른 기본 동작이 필요합니다.

CREATE FUNCTION MAPPING문으로 정의된 함수 맵핑은 연합 데이터베이스에 저장됩니다.

함수(또는 함수 템플릿)에는 데이터 소스 함수와 동일한 수의 입력 매개변수가 있어야 합니다. 추가로, 연합 측의 입력 매개변수의 데이터 유형은 데이터 소스 측의 입력 매개변수의 데이터 유형과 호환 가능해야 합니다.

CREATE FUNCTION MAPPING문을 사용하여 함수 맵핑을 작성하십시오. 예를 들어, Oracle1 서버에서 Oracle AVGNEW 함수와 DB2 동등 함수간에 함수 맵핑을 작성하려면, 다음을 입력하십시오.

```
CREATE FUNCTION MAPPING ORAVGNEW FOR SYSIBM.AVG(INT) SERVER ORACLE1
OPTIONS (REMOTE_NAME 'AVGNEW')
```

연합 데이터베이스에서 SYSADM 또는 DBADM 권한 중 하나를 보유하여 이 명령문을 사용해야 합니다. 함수 맵핑 속성은 SYSCAT.FUNCMAPPINGS에 저장됩니다.

연합 서버는 입력 호스트 변수를 바인드하거나 LOB, LONG VARCHAR/VARGRAPHIC, DATALINK, 구분 유형 및 구조화된 유형의 결과를 검색하지 않습니다. 입력 매개변수 또는 리턴된 값에 이들 유형 중 하나가 포함되면, 함수 맵핑이 작성될 수 없습니다.

함수 맵핑 사용 및 작성에 대해서는 *응용프로그램 개발 안내서*에서 자세한 내용을 참조하십시오. CREATE FUNCTION MAPPING 구문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

함수 템플릿 작성

연합 시스템에서 함수 템플릿은 함수 맵핑에 대한 『앵커』를 제공합니다. 이를 사용하여 해당 DB2 함수가 연합 서버에 없을 때 데이터 소스 함수의 맵핑을 작동할 수 있게 합니다. 함수 맵핑에서는 DB2에 함수 템플릿 또는 기존의 비슷한 함수가 있어야 합니다.

템플릿은 단지 함수 셸일 뿐입니다. 이는 이름, 입력 매개변수 및 리턴 값입니다. 함수에 대한 지역 실행 파일이 없습니다.

함수에 대한 지역 실행 파일이 없으므로, 함수를 데이터 소스에서 사용할 수는 있어도 함수 템플릿을 호출할 수 없게 될 수도 있습니다. 예를 들어, 다음 조회를 고려해 보십시오.

```
SELECT myfunc(C1)
FROM nick1
WHERE C2 < 'A'
```

DB2 및 nick1로 참조되는 오브젝트가 들어 있는 데이터 소스에 동일한 조합 순서가 없으면, 함수가 데이터 소스에 있는 동안 DB2에서 비교가 이루어져야 하므로 조회는 실패합니다. 조합 순서가 같으면, 비교 조작은 myfunc로 참조되는 기초 함수가 있는 데이터 소스에서 이루어질 수 있습니다.

함수(또는 함수 템플릿)에는 데이터 소스 함수와 동일한 수의 입력 매개변수가 있어야 합니다. 연합측의 입력 매개변수의 데이터 유형은 데이터 소스측의 입력 매개변수의 데이터 유형과 호환 가능해야 합니다. 이들 요구사항은 리턴 값에도 적용됩니다.

CREATE FUNCTION문을 AS TEMPLATE 키워드와 함께 사용하여 함수 템플릿을 작성하십시오. 템플릿이 작성된 후 CREATE FUNCTION MAPPING 문을 사용하여 템플릿을 데이터 소스에 맵합니다.

예를 들어, 서버 S1에서 함수 MTS1FUNC에 대한 함수 템플릿 및 함수 맵핑을 작성하려면, 다음을 입력하십시오.

```
CREATE FUNCTION MYFUNC(INT) RETURNS INT AS TEMPLATE

CREATE FUNCTION MAPPING S1_MYFUNC FOR MYFUNC(INT) SERVER S1 OPTIONS
(REMOTE_NAME 'MYS1FUNC')
```

함수 템플릿 사용 및 작성에 대해서는 *응용프로그램 개발 안내서*에서 자세한 내용을 참조하십시오. CREATE FUNCTION 구문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

사용자 정의 유형(UDT) 작성

사용자 정의 유형(UDT)은 사용자가 데이터베이스에 작성한 이름 지정된 데이터 유형입니다. UDT는 내장 데이터 유형과 공통된 표현을 공유하는 구별 유형일 수도 있고 각각 유형을 가지는 이름 지정된 속성 순서를 갖는 구조화 유형일 수도 있습니다. 구조화 유형은 유형 계층을 정의하는 다른 구조화 유형(상위 유형이라 함)의 부속 유형일 수 있습니다.

UDT는 강력한 유형 분류를 지원합니다. 즉, UDT가 다른 유형과 동일한 표현을 공유하더라도 해당 UDT 값은 동일한 UDT 또는 동일한 유형 계층의 UDT 값과 고만 호환되는 것으로 간주됩니다.

SYSCAT.DATATYPES 카탈로그 뷰를 사용하면 데이터베이스에 대해 정의된 UDT를 볼 수 있습니다. 이 카탈로그 뷰에서는 또한 데이터베이스가 작성될 때 데이터베이스 관리 프로그램에 의해 정의된 데이터 유형을 보여줍니다. 모든 데이터 유형의 완전한 목록은 *SQL 참조서*에서 자세한 내용을 참조하십시오.

UDT는 대부분의 시스템 제공 함수 또는 내장 함수의 인수로는 사용될 수 없습니다. 이러한 조작 및 기타 다른 조작을 작동시키려면 사용자 정의 함수가 제공되어야 합니다.

다음 경우에만 UDT를 삭제할 수 있습니다.

- 기존 테이블에 대한 컬럼 정의에서 사용되지 않습니다.
- 기존 입력된 테이블 또는 입력된 뷰의 유형으로 사용되지 않습니다.
- 삭제될 수 없는 UDF 함수에서 사용되지 않습니다. 뷰, 트리거, 테이블 점검 제한조건 또는 다른 UDF가 이에 종속적인 경우에는, UDF가 삭제될 수 없습니다.

UDT가 삭제되면, 종속된 다른 함수도 삭제됩니다.

사용자 정의 구별 유형 작성

사용자 정의 구별 유형은 정수, 십진수 또는 문자 유형과 같은 기존 유형에서 파생된 데이터 유형입니다. CREATE DISTINCT TYPE문을 사용하여 구별 유형을 작성할 수 있습니다.

다음 SQL문은 구별 유형 t_educ를 smallint로 작성합니다.

```
CREATE DISTINCT TYPE T_EDUC AS SMALLINT WITH COMPARISONS
```

WITH COMPARISONS절이 CREATE DISTINCT TYPE문에 지정된 경우(예에서처럼), 동일한 구별 유형의 인스턴스가 서로 비교될 수 있습니다. 소스 데이터 유형이 대형 오브젝트(LOB), DATALINK, LONG VARCHAR 또는 LONG VARGRAPHIC 유형인 경우 WITH COMPARISONS절을 지정할 수 없습니다.

구별 유형의 인스턴스는 소스 유형에 정의된 조건의 함수의 인수 또는 피연산자로서 사용할 수 없습니다. 마찬가지로, 소스 유형은 구별 유형을 사용하도록 정의된 인수 또는 피연산자에는 사용할 수 없습니다.

일단 구별 유형을 작성하면, 이를 사용하여 CREATE TABLE문에서 컬럼을 정의할 수 있습니다.

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(6)      NOT NULL,
   FIRSTNAME  VARCHAR(12)  NOT NULL,
   LASTNAME   VARCHAR(15)  NOT NULL,
   WORKDEPT   CHAR(3),
   PHONENO    CHAR(4),
   PHOTO      BLOB(10M)    NOT NULL,
   EDLEVEL    T_EDUC)
IN RESOURCE
```

구별 유형을 작성하면, 구별 유형과 소스 유형 사이의 변환 지원 또한 생성됩니다. T_EDUC 유형 값을 SMALLINT 값으로 변환하고 SMALLINT 값을 T_EDUC 값으로 변환할 수 있습니다.

CREATE DISTINCT TYPE문에 대한 완전한 구문 정보는 SQL 참조서에서 자세한 내용을 참조하십시오. 구별 유형 작성 및 사용에 대해서는 응용프로그램 개발 안내서에서 자세한 내용을 참조하십시오.

변환을 사용하여 UDT를 기본 데이터 유형으로 변환하고 기본 데이터 유형을 UDT로 변환할 수 있습니다. 변환 함수의 작성은 CREATE TRANSFORM문을 통해 이루어집니다.

변환 지원은 CREATE METHOD문 및 CREATE FUNCTION문의 확장을 통해 얻을 수도 있습니다. 이 지원에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

사용자 정의 구조화 유형 작성

구조화 유형은 각각 이름과 자체 데이터 유형이 있는 하나 이상의 속성을 포함하는 사용자 정의 유형입니다. 구조화 유형은 테이블 유형으로 사용될 수 있는데, 테이블의 각 컬럼은 이름과 데이터 유형을 구조화 유형의 속성 중 하나에서 가져옵니다. *응용프로그램 개발 안내서*에서 구조화된 유형에 필요한 자세한 내용을 참조하십시오.

유형 매핑 작성

연합 시스템에서 유형 매핑을 통해 데이터 소스 테이블에 있는 특정 데이터 유형 및 뷰를 DB2 구별 데이터 유형에 매핑할 수 있습니다. 유형 매핑은 하나의 데이터 소스 또는 데이터 소스의 범위(유형, 버전)에 적용될 수 있습니다.

기본 데이터 유형 매핑은 내장된 데이터 소스 유형과 내장된 DB2 유형에 제공됩니다. 새 데이터 유형 매핑(사용자가 작성하는)은 SYSCAT.TYPEMAPPINGS 뷰에 나열됩니다.

CREATE TYPE MAPPING문으로 유형 매핑을 작성하십시오. 연합 데이터베이스에서 SYSADM 또는 DBADM 권한 중 하나를 보유하여 이 명령문을 사용해야 합니다.

유형 매핑 명령문의 한 예는 다음과 같습니다.

```
CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(10,2)
TO SERVER ORACLE1 TYPE NUMBER([10..38],2)
```

LOB, LONG VARCHAR/VARGRAPHIC, DATALINK, 구조화 유형 또는 구별 유형에 대해 유형 매핑을 작성할 수 없습니다.

유형 매핑 사용 및 작성에 대해서는 응용프로그램 개발 안내서에서 자세한 내용을 참조하십시오. CREATE TYPE MAPPING 구문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

뷰 작성

뷰는 하나 이상의 기본 테이블, 별명 또는 뷰에서 파생되고 데이터 검색시 기본 테이블과 함께 교환되어 사용될 수 있습니다. 뷰에 표시된 데이터를 변경하면, 테이블 내의 데이터도 변경됩니다.

뷰는 감지 가능한 데이터에 대한 액세스를 제한하면서, 반면에 기타 데이터에 대한 일반적인 액세스는 좀더 많이 허용하도록 작성될 수 있습니다.

뷰 정의의 SELECT 목록이 직접 또는 간접으로 기본 테이블의 식별 컬럼 이름을 포함하는 뷰에 삽입할 때, INSERT문이 기본 테이블의 식별 컬럼을 직접 참조한 것처럼 동일한 규칙을 적용합니다. INSERT문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

위에서 설명된 대로 뷰를 사용하는 것 외에도, 뷰는 다음의 경우에도 사용될 수 있습니다.

- 응용프로그램에 영향을 주지 않고 테이블을 변경합니다. 기본 테이블에 기초하여 뷰를 작성하면 이 작업을 수행할 수 있습니다. 기본 테이블을 사용하는 응용프로그램은 새 뷰의 작성에 영향받지 않습니다. 새 응용프로그램은 기본 테이블을 사용하는 응용프로그램과는 다른 목적으로 작성된 뷰를 사용할 수 있습니다.
- 컬럼의 값을 합산하고, 최대값 또는 평균값을 선택합니다.
- 하나 이상의 데이터 소스에 정보에 대한 액세스를 제공합니다. CREATE VIEW 문에서 별명을 참조하고 다중 위치/전역 뷰(뷰는 서로 다른 시스템에 위치한 복수 데이터 소스에 있는 정보를 결합할 수 있음)를 작성할 수 있습니다.

표준 CREATE VIEW 구문을 사용하는 별명을 참조하는 뷰를 작성할 때, 뷰 작성자 인증 ID 대신 기초를 이루는 오브젝트 또는 데이터 소스에 있는 오브

젝트에 액세스하는 데 뷰 사용자의 인증 ID가 사용된다는 사실을 알려주는 경고가 표시됩니다. FEDERATED 키워드를 사용하여 이 경고가 표시되지 않도록 하십시오.

뷰를 작성하는 또하나의 이유는 중첩 또는 공통 테이블 표현식을 사용하여 카탈로그 조사를 줄이고 성능을 개선하기 위해서입니다. 공통 테이블 표현식에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 보기를 작성하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 뷰 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 작성을 선택하십시오.
3. 정보를 완료하고, 확인을 클릭하십시오.

명령행을 사용하여 보기를 작성하십시오.

```
CREATE VIEW <name> (<column>, <column>, <column>)  
  SELECT <column_names> FROM <table_name>  
  WITH CHECK OPTION
```

예를 들어, EMPLOYEE 테이블에는 아무나 사용할 수 없는 급여 정보가 들어 있습니다. 그러나, 사원의 전화번호는 일반적으로 액세스할 수 있어야 합니다. 이 경우, LASTNAME 및 PHONENO 컬럼에서만 뷰가 작성될 수 있습니다. 뷰에 대한 액세스는 PUBLIC으로 권한 부여되는 반면에, 전체 EMPLOYEE 테이블에 대한 액세스는 급여 정보를 볼 수 있는 권한을 가진 사용자로 제한될 수 있습니다. 읽기 전용 뷰에 대해서는 *SQL 참조서 매뉴얼*을 참조하십시오.

뷰를 사용하여, 응용프로그램에 대해 사용 가능한 테이블 데이터 부속 집합을 만들고, 삽입되거나 갱신될 데이터의 유효성 검사를 수행할 수 있습니다. 뷰에서는 원래 테이블의 해당 컬럼 이름과는 다른 컬럼 이름을 사용할 수 있습니다.

뷰를 사용하면, 사용자 프로그램에 융통성을 더할 수 있고 일반 사용자 조회에서 테이블 데이터를 볼 수 있습니다.

다음 SQL문은 EMPLOYEE 테이블에 A00 부서의 모든 사원과 전화번호를 나열하는 뷰를 작성합니다.

```

CREATE VIEW EMP_VIEW (DA00NAME, DA00NUM, PHONENO)
  AS SELECT LASTNAME, EMPNO, PHONENO FROM EMPLOYEE
  WHERE WORKDEPT = 'A00'
  WITH CHECK OPTION

```

이 명령문의 첫번째 행에서는 뷰를 이름 지정하고 컬럼을 정의합니다. 이름 EMP_VIEW는 SYSCAT.TABLES의 스키마에서 고유해야 합니다. 뷰 이름은 안에 데이터가 들어 있지 않더라도 테이블 이름으로 나타납니다. 뷰는 EMPLOYEE 테이블의 컬럼 LASTNAME, EMPNO, PHONENO에 해당하는 DA00NAME, DA00NUM, PHONENO라는 세 개의 컬럼을 가지게 됩니다. 나열된 컬럼 이름은 SELECT문의 선택 목록에 일대일로 적용됩니다. 컬럼 이름이 지정되지 않으면, 뷰에서는 SELECT문의 결과 테이블의 컬럼과 동일한 이름을 사용합니다.

두 번째 행은 데이터베이스에서 어느 값이 선택될 것인지를 설명하는 SELECT문입니다. 여기에 ALL, DISTINCT, FROM, WHERE, GROUP BY 및 HAVING 절을 포함할 수도 있습니다. 뷰에 대한 컬럼을 선택할 데이터 오브젝트의 이름(들)이 FROM절 뒤에 와야 합니다.

WITH CHECK OPTION절은 갱신되거나 뷰에 삽입된 행을 뷰 정의에 대해 점검해야 하며, 맞지 않을 경우 거부됨을 나타냅니다. 이로 인해, 데이터 무결성은 강화되지만 추가 처리가 요구됩니다. 이 절을 생략하면, 삽입 및 갱신 내용이 뷰 정의에 맞는지 점검되지 않습니다.

다음 SQL문은 SELECT AS절을 사용하여 EMPLOYEE 테이블에 대해 동일한 뷰를 작성합니다.

```

CREATE VIEW EMP_VIEW
  SELECT LASTNAME AS DA00NAME,
         EMPNO AS DA00NUM,
         PHONENO
  FROM EMPLOYEE
  WHERE WORKDEPT = 'A00'
  WITH CHECK OPTION

```

사용자는 정의 안에 UDF를 사용하는 뷰를 작성할 수 있습니다. 그러나, 이 뷰를 갱신하여 최근 함수를 포함하려면, 뷰를 삭제한 후 다시 작성해야 합니다. 뷰가 UDF에 종속적이면, 해당 함수는 삭제될 수 없습니다.

다음 SQL문은 정의에 함수가 있는 뷰를 작성합니다.

```
CREATE VIEW EMPLOYEE_PENSION (NAME, PENSION)
AS SELECT NAME, PENSION(HIREDATE,BIRTHDATE,SALARY,BONUS)
FROM EMPLOYEE
```

UDF 함수 PENSION은 공식에 따라 HIREDATE, BIRTHDATE, SALARY 및 BONUS를 비롯하여 사원이 받아야 할 현재의 수당을 계산합니다.

입력된 뷰 작성

CREATE VIEW문을 사용하여 입력된 뷰를 작성할 수 있습니다. 응용프로그램 개발 안내서에서 입력된 뷰에 필요한 자세한 내용을 참조하십시오.

요약 테이블 작성

요약 테이블은 조회 결과에 기초하여 정의되는 테이블입니다. 일반적으로, 요약 테이블은 해당 정의가 기초하고 있는 테이블에 존재하는 데이터에 의거하여 사전 계산된 결과를 수록합니다. SQL 컴파일러가 조회는 기본 테이블보다 요약 테이블에 대해 보다 효과적으로 실행될 것이라고 판단한 경우, 해당 조회는 요약 테이블에 대해 실행되며 사용자는 다른 경우보다 빨리 결과를 취득하게 됩니다.

복제 옵션을 사용하여 요약 테이블을 작성하면 파티션된 데이터베이스 환경의 모든 노드에 걸쳐 테이블을 복제할 수 있습니다. 이러한 테이블을 『복제된 요약 테이블』이라 합니다. 관리 안내서: 계획에서 그러한 테이블에 대한 개요 정보를 참조하십시오.

주: 요약 테이블은 정적 SQL 또는 별명과 함께 사용되지 않습니다.

일반적으로, 요약 테이블 또는 복제된 요약 테이블은 요약 테이블 또는 복제된 요약 테이블의 분리 레벨이 조회의 분리 레벨과 같거나 더 높으면, 조회 최적화에 사용됩니다. 예를 들어, 조회가 커서 안정성(CS) 분리 레벨 아래서 수행중일 때 CS 또는 더 높은 분리 레벨하에서 정의된 요약 테이블과 복제된 요약 테이블만 최적화에 사용됩니다.

요약 테이블을 작성하려면, CREATE SUMMARY TABLE문을 AS *fullselect*절 및 REFRESH DEFERRED 옵션과 함께 사용하십시오.

요약 테이블의 컬럼 이름을 고유하게 식별하는 옵션을 갖습니다. 컬럼 이름의 목록은 전체 선택의 결과 테이블에 컬럼이 있을 때만큼 많은 이름을 포함해야 합니다. 전체 선택의 결과 테이블이 중복된 컬럼 이름이나 이름없는 컬럼을 갖는다면, 컬럼 이름 목록이 제공되어야 합니다. 이름없는 컬럼은 상수, 함수, 표현식 또는 선택 목록의 AS 절을 사용하는 이름지정되지 않은 세트 조작에서 파생됩니다. 컬럼 이름의 목록이 지정되지 않은 경우, 테이블의 컬럼은 전체 선택의 결과 세트 컬럼의 이름을 계승합니다.

요약 테이블을 작성하는 경우, 기본 테이블이 변경될 때 요약 테이블이 자동으로 정리되는지 또는 REFRESH TABLE문을 사용하여 정리되는지를 지정하는 옵션이 제공됩니다. 기본 테이블을 변경할 때 요약 테이블을 자동으로 정리하려면, REFRESH IMMEDIATE 키워드를 지정하십시오. 정리는 다음과 같은 경우에 즉시 유용합니다.

- 기본 테이블에 대하여 수행할 때 오래 걸리는 조치가 있습니다
- 기본 테이블은 자주 변경되지 않습니다
- 새로 고침은 비용이 적게 듭니다.

이러한 상황에서 요약 테이블은 사전 처리된 결과를 제공할 수 있습니다. 요약 테이블 정리를 지연시키려면, REFRESH DEFERRED 키워드를 지정하십시오. REFRESH DEFERRED를 사용하여 지정된 요약 테이블은 기본 테이블 변경사항을 반영하지 않습니다. 변경이 꼭 필요하지 않은 곳에서 요약 테이블을 사용해야 합니다. 예를 들어, DSS 조회를 실행할 경우, 요약 테이블을 사용하여 기존 데이터를 취득할 수 있습니다.

다음과 같은 경우 REFRESH DEFERRED를 사용하여 정의한 요약 테이블을 조회 대신에 사용할 수 있습니다.

- 다음과 같은 경우를 제외하고 지금 새로 고침 테이블의 fullselect 제한사항을 따릅니다.
 - SELECT 목록이 COUNT(*) 또는 COUNT_BIG(*)을 포함할 필요가 없는 경우
 - SELECT 목록이 MAX 및 MIN 컬럼 함수를 포함할 수 있는 경우
 - HAVING절이 허용되는 경우

SQL 특수 레지스터 CURRENT REFRESH AGE SQL이 ANY로 설정되거나 9999999999999999 값을 가지는 경우, 9의 콜렉션은 DECIMAL(20,6) 데이터 유형을 가진 시간소인 지속시간 값인 이 특수 레지스터에 허용되는 최대값입니다.

주: REFRESH DEFERRED로 정의된 요약 테이블은 정적 SQL 최적화에 사용되지 않습니다.

CURRENT REFRESH AGE 특수 레지스터를 사용하면 정리가 연기된 요약 테이블을 새로 고치기 전에 동적 조회에 사용할 수 있는 시간을 지정할 수 있습니다. CURRENT REFRESH AGE 특수 레지스터 값을 설정하려면 SET CURRENT REFRESH AGE문을 사용하면 됩니다. CURRENT REFRESH AGE 특수 레지스터 및 SET CURRENT REFRESH AGE문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

REFRESH IMMEDIATE로 정의된 요약 테이블은 정적 및 동적 조회에 모두 적용 가능하고 CURRENT REFRESH AGE 특수 레지스터를 사용하지 않아도 됩니다.

주: CURRENT REFRESH AGE 특수 레지스터 값을 0이 아닌 값에 설정할 때에는 주의해야 합니다. 기본 테이블의 값을 표시하지 않는 요약 테이블을 조회 처리 최적화에 사용할 수 있도록 하면, 조회 결과는 기본 테이블의 데이터를 정확히 표시하지 않습니다. 이는 기본 데이터가 변경되지 않거나 데이터에 대한 지식에 기초하여 결과의 오류 정도를 수용하려는 경우에 해당합니다.

활동이 소스 데이터에 영향을 주기 때문에, 시간이 지난 요약 테이블에는 더 이상 정확한 데이터가 수록되지 않습니다. REFRESH TABLE문을 사용해야 합니다. *SQL 참조서*에서 자세한 내용을 참조하십시오.

유효한 *fullselect*에 기초하여 새 기본 테이블을 작성하려는 경우, 테이블을 작성할 때 DEFINITION ONLY 키워드를 지정하십시오. 테이블 작성 조작이 완료되면, 새 테이블은 요약 테이블이 아닌 기본 테이블로 간주됩니다. 예를 들어, 다음과 같이 LOAD 및 SET INTEGRITY에 사용되는 예외 테이블을 작성할 수 있습니다.

```
CREATE TABLE XT AS
(SELECT T.*, CURRENT TIMESTAMP AS TIMESTAMP,CLOB(",32K)
AS MSG FROM T) DEFINITION ONLY
```


다음은 요약 테이블과 관련된 몇 가지 주요 제한사항입니다.

1. 요약 테이블을 변경할 수 있습니다.
2. 해당 테이블에 요약 테이블이 있는 경우 기본 테이블의 컬럼 길이는 변경할 수 없습니다.
3. 요약 테이블에는 데이터를 가져올 수 없습니다.
4. 요약 테이블에 고유 색인을 작성할 수 없습니다.
5. 하나 이상의 별명을 참조하는 조회 결과를 기초로 한 요약 테이블을 작성할 수 없습니다.

요약 테이블 제한사항에 관한 완전한 명령문은 *SQL 참조서*에서 자세한 내용을 참조하십시오.

별명 작성

별명은 테이블, 별명 또는 뷰를 참조하는 간접 방식입니다. 그러므로, SQL문은 해당 테이블이나 뷰의 완전한 이름과는 무관합니다. 테이블이나 뷰 이름이 변경될 경우에는 별명 정의만 변경하면 됩니다. 또다른 별명에 대한 별명이 작성될 수 있습니다. 별명은, 기존의 테이블 또는 뷰 이름을 참조할 수 있는 테이블 제한조건 정의를 제외하고는, 뷰 또는 트리거 정의와 임의의 SQL문에서 사용할 수 있습니다.

별명은 테이블 이름이 사용되는 곳이면 어디든지 사용될 수 있으며, 별명의 체인을 따라 순환 또는 반복적으로 참조되지 않을 경우, 다른 별명을 참조할 수 있습니다.

별명은 기존의 테이블, 뷰 또는 별명과 동일할 수 없으며, 동일한 데이터베이스 내의 한 테이블만을 언급할 수 있습니다. CREATE TABLE 또는 CREATE VIEW 문에 사용된 테이블 또는 뷰의 이름은 동일한 스키마에 있는 다른 별명과 달라야 합니다.

사용자의 현재 권한 부여 ID가 소유하고 있는 스키마가 아닌 다른 스키마에 별명이 존재하지 않는 한, 별명을 작성하는 데에는 특수 권한이 필요하지 않습니다. 이 예의 경우에는 DBADM 권한이 필요합니다.

별명은 정의시에 존재하지 않는 테이블, 뷰 또는 별명에 대해 정의될 수 있습니다. 그러나, 별명이 들어 있는 SQL문이 컴파일될 때에는 반드시 있어야 합니다.

별명 또는 별명이 가리키는 오브젝트가 삭제될 경우, 그 별명과 관련된 모든 패키지에 유효하지 않다는 표시가 주어지고 별명에 의존한 모든 뷰와 트리거에 작동 불가능이라는 표시가 주어집니다.

제어 센터를 사용하여 별명을 작성하십시오.

1. 별명 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 별명 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 작성을 선택하십시오.
3. 정보를 완료하고, 확인을 클릭하십시오.

명령행을 사용하여 별명을 작성하십시오.

```
CREATE ALIAS <alias_name> FOR <table_name>
```

별명은 명령문 컴파일시에 테이블 또는 뷰 이름에 의해 대체됩니다. 별명 또는 별명 체인을 테이블 또는 뷰 이름으로 분석할 수 없는 경우, 오류가 발생합니다. 예를 들어, WORKERS가 EMPLOYEE의 별명이면, 컴파일할 때 다음 명령문

```
SELECT * FROM WORKERS
```

는 실제로 다음과 같이 됩니다.

```
SELECT * FROM EMPLOYEE
```

다음 SQL문은 EMPLOYEE 테이블에 대한 별명 WORKERS를 작성합니다.

```
CREATE ALIAS WORKERS FOR EMPLOYEE
```

주: MVS/ESA용 DB2는 ALIAS 및 SYNONYM 별명의 두 가지 구별 개념을 사용합니다. 이 두 개념은 다음과 같이 DB2 Universal Database와 다릅니다.

- MVS/ESA용 DB2에 있는 ALIAS
 - 작성자에게 특수 권한 또는 특권이 있어야 합니다.
 - 다른 별명을 언급할 수 없습니다.
- MVS/ESA용 DB2에 있는 SYNONYM:

- 작성자만 사용할 수 있습니다.
- 항상 규정화되지 않습니다.
- 참조 테이블이 삭제될 때 삭제됩니다.
- 테이블 또는 뷰와 함께 이름공간을 공유하지 않습니다.

랩퍼 작성

연합 데이터베이스에서 CREATE WRAPPER문은 랩퍼를 등록합니다. 명령문은 연합 서버가 데이터 소스의 특정 범주와 상호작용할 수 있는 메커니즘을 정의합니다.

특정 라이브러리는 특정 데이터 소스 유형, 버전, 통신 프로토콜 및 운영 체제에 사용되어야 합니다. 예를 들어, OS/390용 AS/400 및 DB2 데이터 소스는 Windows NT 운영 체제에서 APPC 통신을 통해 작동하는 연합 데이터베이스에 대해 libdrda.dll 라이브러리를 사용하여 액세스됩니다.

CREATE WRAPPER문을 사용하려면, 연합 데이터베이스에서 SYSADM 또는 DBADM 권한이 있어야 합니다.

제어 센터나 명령행에서 랩퍼를 작성하면 연합 데이터베이스에 등록합니다.

제어 센터를 사용하여 랩퍼를 작성하십시오.

1. 연합 데이터베이스 오브젝트 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 연합 데이터베이스 오브젝트 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 랩퍼 작성을 선택하십시오.
3. 정보를 완료하고, 확인을 클릭하십시오.

명령행을 사용하여 랩퍼를 작성하십시오.

```
CREATE WRAPPER <wrapper_name> LIBRARY '<library_name>'
```

다음 SQL문은 랩퍼 ORACLE8을 Windows NT 운영 체제에 등록합니다.

```
CREATE WRAPPER ORACLE8 LIBRARY 'libnet8.dll'
```

CREATE WRAPPER문 사용 방법에 대한 세부사항은 *SQL 참조서*에서 자세한 내용을 참조하십시오.

서버 작성

연합 데이터베이스에서 서버를 작성하여 DB2에 데이터 소스를 정의하고 해당 특성인 이름, 랩퍼, 유형, 버전, 위치 및 옵션을 설명하십시오. 이 정보는 별명을 특정 데이터 관리 시스템에 맵핑하고 DB2 최적화 알고리즘에 정보를 제공하는 데 사용됩니다. 서버 정보는 SYSCAT.SERVERS 및 SYSCAT.SERVEROPTIONS 카탈로그 뷰에 위치해 있습니다.

주: 이 절에서 서버는 DRDA 서버 또는 DB2 서버가 아닌 데이터 소스를 나타냅니다. 다른 데이터 소스(예를 들어, Oracle)에 액세스하려면, DB2 Connect가 필요합니다.

랩퍼가 작성된 경우에만 서버 오브젝트를 작성할 수 있습니다.

이 명령문을 사용하려면, 연합 데이터베이스에서 SYSADM 또는 DBADM 권한이 있어야 합니다.

DB2와 데이터 소스 서버간의 인증 처리 차이점을 관리하는 사용자 맵핑을 작성할 수 있습니다. 사용자 맵핑은 269 페이지의 『사용자 맵핑』에 자세히 설명되어 있습니다.

서버가 삭제되면, 해당 서버에 종속되어 있는 모든 오브젝트(사용자 맵핑, 별명, 함수 맵핑, 유형 맵핑, 플랜 등)는 삭제됩니다.

서버를 작성할 때 서버 옵션을 제공하십시오. 이들 옵션은 서버에 필요한 세부사항을 포함합니다(예: 노드 이름). 서버 옵션은 또한 특정 성능 또는 보안 값을 설정합니다.

제어 센터 또는 명령행 처리기에서 서버를 작성할 수 있습니다.

제어 센터를 사용하여 랩퍼를 작성하십시오.

1. 연합 데이터베이스 오브젝트 폴더 아래에서 서버 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 서버 폴더를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 랩퍼 작성을 선택하십시오.
3. 정보를 완료하고, 확인을 클릭하십시오.

명령행을 사용하여 래퍼를 작성하십시오.

```
CREATE SERVER <server_name> TYPE <server_type>
  VERSION <server_version> WRAPPER <wrapper_name>
  OPTIONS (<server_option_name> <string_constant>)
```

다음 샘플 SQL문은 Oracle 서버 ORA8을 작성합니다.

```
CREATE SERVER ORA8 TYPE ORACLE VERSION 8 WRAPPER ORACLE8 OPTIONS
(NODE 'ONODE')
```

다음 샘플 SQL문은 DB2 서버 DB2TEST를 작성합니다.

```
CREATE SERVER DB2TEST TYPE DB2 VERSION 6.1 WRAPPER DB2UDB OPTIONS
(NODE 'DB2TEST', DBNAME 'TEST1')
```

SERVER SQL문에 있는 NODE의 정의는 데이터 소스에 따라 달라집니다. 데이터 소스가 DB2 DBMS이면, 값은 하나 이상의 데이터베이스가 DB2 인스턴스를 참조합니다. 앞의 예에서, DBNAME 옵션은 데이터베이스 이름을 지정한다는 것에 유의하십시오. 데이터 소스가 OS/390용 DB2 DBMS이면, 값은 OS/390용 DB2의 LOCATION 이름을 참조합니다. 데이터 소스가 Oracle DBMS이면, Oracle 인스턴스에는 오직 하나의 데이터베이스만이 들어 있으므로 DBNAME 옵션은 필요하지 않습니다.

SQL 참조서에서 CREATE SERVER문 구문에 대한 자세한 내용을 참조하십시오. 설치 및 구성 보충 설명서에서 CREATE SERVER문 사용에 대한 자세한 내용을 참조하십시오.

서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화

서버 옵션이라고 하는 변수를 연합 서버가 데이터 소스에 액세스하는 방법에 영향을 주는 값으로 설정할 수 있습니다. 이 절에서는 다음을 설명합니다.

- 서버 옵션의 목적을 설명합니다
- 서버 옵션을 지정할 때 사용하는 SQL문을 설명합니다
- 서버 옵션 및 설정값을 보여줍니다

서버 옵션의 목적

일반적으로, 서버 옵션을 사용하여 다음을 수행할 수 있습니다.

- 데이터 소스에 대한 정보를 제공하고 갱신합니다. 서버 참조에는 이름과 같이 데이터 소스에 대한 기본 정보와 시간에 따라 변경될 수 있는 정보가 있습니다. 변경 가능한 일부 정보는 서버 옵션에 지정된 값으로 전달됩니다. 예를 들어, `cpu_ratio` 옵션에 지정된 값은 데이터 소스의 CPU가 DB2 시스템 CPU보다 빠르거나 느리다는 것을 나타냅니다. DB2 시스템에서 하나 이상의 프로세서가 업그레이드되면, 이 값은 변경되어야 합니다.
- 인증 작업을 쉽게 만듭니다. 일부 서버 옵션을 설정하여 사용자 ID와 암호가 적합한 형태(대문자 또는 소문자)로 데이터 소스에 전송되는지 확인할 수 있습니다. 예를 들어, `fold_id` 옵션을 설정하여 연합 서버가 사용자 ID를 데이터 소스에 전송하기 전에, 연합 서버는 이름을 데이터 소스에서 필요한 형태(대문자 또는 소문자)로 변환합니다. 또는 연합 서버에서 사용자 ID를 필요한 형태로 정의하는 경우, `fold_id` 옵션을 설정하여 서버가 문자 형태를 변경하고 프로세스에서 오버헤드를 차지하지 못하도록 막을 수 있습니다.
- 조화를 최적화합니다. 일부 서버 옵션과 해당 값은 최적화를 쉽게 수행할 수 있게 합니다. 예를 들어, `CREATE SERVER`문에서 옵션 값에 따라 특정 성능 통계를 지정할 수 있습니다. 특별히, `cpu_ratio` 옵션을 데이터 소스의 상대 속도 및 연합 서버의 CPU를 나타내는 값으로 설정할 수 있습니다. 그런 다음, `io_ratio` 옵션의 값을 데이터 소스 및 연합 서버의 입출력 장치의 상대적 비율로 설정할 수 있습니다. `CREATE SERVER`를 수행하면, 이 통계는 `SYSCAT.SERVEROPTIONS` 카탈로그 뷰에 추가되고, 최적화 알고리즘은 이를 사용하여 데이터 소스에 대한 액세스 플랜을 개발합니다. 통계가 변경되면 (예를 들어, 데이터 소스 CPU가 업그레이드될 경우에 발생할 수 있음), `ALTER SERVER`문을 사용하여 변경된 내용으로 `SYSCAT.SERVEROPTIONS`를 갱신할 수 있습니다. 그러면, 최적화 알고리즘은 갱신된 내용을 사용하여 데이터 소스에 대한 다음 액세스 플랜을 개발합니다.

서버 옵션에 대한 SQL

`CREATE SERVER`, `ALTER SERVER` 및 `SET SERVER OPTION`과 같이 값을 서버 옵션에 지정할 수 있는 세 개의 SQL문이 있습니다.

`CREATE SERVER`문을 사용하여 데이터 소스에 대한 다중 연결에 대해 무한히 지속되는 값으로 옵션을 설정하십시오. 이 명령문으로 옵션을 기본값 이외의 값으로 설정하거나 옵션에 기본값이 없으면, 이를 초기 값으로 설정할 수 있습니다.

서버 옵션을 CREATE SERVER문으로 설정한 후 이를 다중 연결에서 지속되는 다른 값으로 설정하려면, ALTER SERVER문을 사용하십시오.

SET SERVER OPTION문을 사용하여 데이터베이스에 대한 단일 연결중에 서버 옵션 값을 임시로 변경하십시오. SET SERVER OPTION문은 데이터 소스 연결 다음에 첫번째 작업 단위 내에서 먼저 실행되어야 합니다.

예를 들어, Oracle 서버 ORASEB1에 대한 플랜 추가정보를 임시로 사용하려면, 다음 명령문을 발행하십시오.

```
SET SERVER OPTION plan_hints TO 'Y' FOR SERVER ORASEB1
```

서버 옵션 및 설정값

다음 표에서는 서버 옵션과 설정할 수 있는 값에 대해 설명합니다. 달리 표시하지 않는 한, 모든 서버 옵션 값을 작은따옴표로 묶어야 합니다.

표 2. 서버 옵션 및 설정값

옵션	유효 설정값	기본 설정값
collating_sequence	<p>코드 세트 및 국가 정보에 따라 데이터 소스가 연합 데이터베이스와 동일한 기본 조합 순서를 사용할지 여부를 지정합니다. 데이터 소스가 DB2의 조합 순서와 다른 조합 순서를 사용할 경우, DB2의 조합 순서에 의존하는 대부분의 조작은 데이터 소스에서 원격으로 평가될 수 없습니다. 예에서는 다른 조합 순서를 사용하여 데이터 소스에서 별명 문자 컬럼에 대해 MAX 컬럼 함수를 실행합니다. MAX 함수는 원격 데이터 소스에서 평가될 경우 결과가 다를 수 있기 때문에, DB2는 집계 조작과 MAX 함수를 지역적으로 수행합니다.</p> <p>조회에 등호가 들어 있으면, 조합 순서가 다르더라도 조회의 해당 부분을 데이터 소스로 넘길 수 있습니다('N'으로 설정). 예를 들어, 술어 C1 = 'A'는 데이터 소스로 넘겨질 수 있습니다. 물론, 데이터 소서의 조합 순서가 대소문자를 구별하지 않을 경우에는 조회를 넘길 수 없습니다. 데이터 소스가 대소문자를 구별하지 않을 경우, C1= 'A' 및 C1 = 'a'의 결과는 동일하며, 이는 대소문자 구별 환경(DB2)에서는 받아들여지지 않습니다.</p> <p>관리자는 데이터 소스 조합 순서와 일치하는 특정 조합 순서를 사용하여 연합 데이터베이스를 작성할 수 있습니다. 이 접근 방법은 모든 데이터 소스가 동일한 조합 순서를 사용하거나 대부분의 또는 모든 컬럼 함수가 동일한 조합 순서를 사용하는 데이터 소스로 지정될 경우, 성능이 향상됩니다.</p> <p>'Y' 데이터 소스의 조합 순서가 연합 데이터베이스의 조합 순서와 동일합니다.</p> <p>'N' 데이터 소스의 조합 순서가 연합 데이터베이스의 조합 순서와 동일하지 않습니다.</p> <p>'I' 데이터 소스의 조합 순서가 연합 데이터베이스의 조합 순서와 다르며, 대소문자를 구별하지 않습니다(예를 들면, 'TOLLESON'과 'TolLESon'이 동일하게 간주됩니다).</p>	'N'
comm_rate	연합 서버와 연관된 데이터 소스간의 통신 비율을 지정합니다. 초당 메가바이트로 표현됩니다.	'2.0'
connectstring	OLE DB 제공자에 연결하기 위해 필요한 초기 등록 정보를 지정합니다. 연결 문자열의 완전한 구문 및 의미는 <i>Microsoft OLE DB 2.0 Programmer's Reference</i> 및 <i>Data Access SDK, Microsoft Press, 1998</i> 에 있는 "Data Link API of the OLE DB Core Components"를 참조하십시오.	없음

표 2. 서버 옵션 및 설정값 (계속)

옵션	유효 설정값	기본 설정값
cpu_ratio	연합 서버의 CPU 속도보다 데이터 소스의 CPU 속도가 얼마나 빠르 지 또는 느린지를 나타냅니다.	'1.0'
dbname	연합 서버가 액세스하도록 할 데이터 소스 데이터베이스의 이름. DB2 계열 데이터 소스에 필요하며, Oracle** 데이터 소스에는 적용되지 않 습니다.	없음
fold_id(이 표의 끝부분에 있는 주 1 및 4 참조)	연합 서버가 인증을 위해 데이터 소스로 전송한 사용자 ID에 적용됩니 다. 유효한 값은 다음과 같습니다. 'U' 연합 서버는 사용자 ID를 데이터 소스로 전송하기 전에 대문 자로 변경합니다. 이는 DB2 계열 및 Oracle** 데이터 소스 에 대한 논리적 선택입니다(이 표의 끝부분 있는 주 2 참조). 'N' 연합 서버는 사용자 ID를 데이터 소스로 전송하기 전에 아무 것도 수행하지 않습니다(이 표의 끝부분에 있는 주 2 참조). 'L' 연합 서버는 사용자 ID를 데이터 소스로 전송하기 전에 소문 자로 변경합니다. 이들 설정값 중 아무것도 사용되지 않으면, 연합 서버는 대문자로 사용 자 ID를 데이터 소스로 전송하려 합니다. 사용자 ID가 실패하면, 서버 는 소문자로 전송을 시도합니다.	없음
fold_pw(이 표의 끝부분에 있 는 주 1, 3 및 4 참조)	연합 서버가 인증을 위해 데이터 소스로 전송하는 암호에 적용됩니다. 유효한 값은 다음과 같습니다. 'U' 연합 서버는 암호를 데이터 소스로 전송하기 전에 대문자로 변 경합니다. 이는 DB2 계열 및 Oracle** 데이터 소스에 대한 논리적 선택입니다. 'N' 연합 서버는 암호를 데이터 소스로 전송하기 전에 아무 것도 수행하지 않습니다. 'L' 연합 서버는 암호를 데이터 소스로 전송하기 전에 소문자로 변 경합니다. 이들 설정값 중 아무것도 사용되지 않으면, 연합 서버는 대문자로 암호 를 데이터 소스로 전송하려 합니다. 암호가 실패하면, 서버는 소문자로 전송을 시도합니다.	없음
io_ratio	연합 서버의 입출력 속도보다 데이터 소스의 입출력 속도가 얼마나 빠 른지 또는 느린지를 나타냅니다.	'1.0'

표 2. 서버 옵션 및 설정값 (계속)

옵션	유효 설정값	기본 설정값
노드	<p>데이터 소스가 해당 RDBMS에서 인스턴스로 정의되는 이름입니다. 모든 데이터 소스에 필요합니다.</p> <p>DB2 계열 데이터 소스의 경우, 이 이름은 연합 데이터베이스의 DB2 노드 디렉토리에 지정되어 있습니다. 이 디렉토리를 보려면, db2 list node directory 명령을 발행하십시오.</p> <p>Oracle** 데이터 소스의 경우, 이 이름은 Oracle** tnsnames.ora 파일에 지정된 서버 이름입니다. Windows NT 플랫폼에서 이 이름에 액세스하려면, Oracle** SQL Net Easy Configuration 도구의 구성 정보 보기 옵션을 지정하십시오.</p>	없음
암호	<p>데이터 소스로 암호를 전송할지 여부를 지정합니다.</p> <p>‘Y’ 암호가 항상 데이터 소스로 전송되며 검증됩니다. 이는 기본값입니다.</p> <p>‘N’ 암호가 데이터 소스로 전송되지 않으며(사용자 매핑에 관계없이) 검증되지 않습니다.</p> <p>‘ENCRYPTION’ 암호가 항상 암호화 형식을 사용하여 데이터 소스로 전송되며 검증됩니다. 암호화된 암호를 지원하는 DB2 계열 데이터 소스에 대해서만 유효합니다.</p>	‘Y’
plan_hints	<p>플랜 추가정보가 사용 가능한지 여부를 지정합니다. 플랜 추가정보는 데이터 소스 최적화 알고리즘에 대한 추가 정보를 제공하는 명령문입니다. 특정조회 유형의 경우, 이 정보는 조회 성능을 향상시킬 수 있습니다. 플랜 추가정보는 색인 사용 여부, 사용할 색인 또는 사용할 테이블 조인 순서를 데이터 소스 최적화 알고리즘이 결정하는 데 도움이 됩니다.</p> <p>‘Y’ 데이터 소스가 플랜 추가정보를 지원할 경우, 데이터 소스에서 플랜 추가정보를 사용할 수 있습니다.</p> <p>‘N’ 데이터 소스에서 플랜 추가정보를 사용할 수 없습니다.</p>	‘N’
푸시다운	<p>‘Y’ DB2는 데이터 소스가 조작을 평가하도록 합니다.</p> <p>‘N’ DB2는 원격 데이터 소스에 있는 컬럼만을 검색하며 데이터 소스가 조인과 같은 다른 조작은 평가하지 못하게 합니다.</p>	‘Y’

표 2. 서버 옵션 및 설정값 (계속)

옵션	유효 설정값	기본 설정값
varchar_no_trailing_blanks	이 데이터 소스가 공백이 아닌 문자로 채워진 VARCHAR 비교 의미론을 사용할지 여부를 지정합니다. 뒤 공백이 없는 가변 길이 문자열의 경우, 일부 DBMS의 공백이 아닌 문자로 채워진 비교 의미론은 DB2의 비교 의미론과 동일한 결과를 리턴합니다. 데이터 소스에 있는 모든 VARCHAR 테이블/뷰 컬럼에 뒤 공백이 포함되어 있지 않으면, 이 서버 옵션을 데이터 소스의 경우에는 'Y'로 설정하십시오. 이 옵션은 종종 Oracle** 데이터 소스에도 사용됩니다. 별명(뷰를 포함하여)을 사용할 가능성이 있는 모든 오브젝트도 고려하도록 하십시오. 'Y' 이 데이터 소스는 DB2와 유사한 공백이 아닌 문자로 채워진 비교 의미론을 가집니다. 'N' 이 데이터 소스는 DB2와 동일한 공백이 아닌 문자로 채워진 비교 의미론을 갖지 않습니다.	'N'

이 테이블에서의 주의사항.

1. 이 필드는 인증에 지정된 값에 관계없이 적용됩니다.
2. DB2는 사용자 ID를 대문자로 저장하므로, 'N' 및 'U' 값은 논리적으로 서로 동일합니다.
3. fold_pw의 설정값은 암호에 대한 설정값이 'N'이면 영향을 미치지 않습니다. 암호가 전송되지 않기 때문에, 대소문자는 인수가 될 수 없습니다.
4. 이 옵션 모드에 널(NULL)을 설정하는 것은 피하도록 하십시오. DB2가 사용자 ID 및 암호를 해석하기 위해 여러 번의 시도를 하기 때문에 널(NULL)로 설정할 수 있으나 성능이 저하됩니다.(DB2 데이터 소스 인증을 성공적으로 전달하기 전에 DB2가 네 번까지 사용자 ID 및 암호를 전송하는 것은 가능합니다.)

서버로 통과 세션 사용

통과 세션을 통해 응용프로그램은 서버의 원시 클라이언트 액세스 방식 및 원시 SQL 통용어를 사용하여 서버와 직접 통신할 수 있습니다.

통과 세션은 다음과 같은 경우에 유용합니다.

- 응용프로그램이 데이터 소스에서 오브젝트를 작성하거나 INSERT, UPDATE 또는 DELETE 조작을 수행해야 할 경우
- DB2가 고유한 데이터 소스 조작을 지원하지 않을 경우

통과 세션에서 오브젝트를 참조하는 경우 오브젝트의 참 이름(별명 아님)을 사용하십시오.

SET PASSTHRU문을 사용하여 통과 세션을 시작하고 서버에 직접 액세스하십시오. 이 명령문은 동적으로 발행되어야 합니다. 이 명령문의 예는 다음과 같습니다.

```
SET PASSTHRU BACKEND
```

이는 데이터 소스 BACKEND에 대한 통과 세션을 엽니다.

통과 세션에서의 SET PASSTHRU 및 SQL 처리에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

별명 작성

연합 데이터베이스에서 별명은 데이터 소스, 테이블, 별명 및 뷰에 대한 식별자입니다. 분산 요청은 일반적으로 데이터 소스 테이블이나 뷰가 아닌 별명을 참조합니다.

별명은 DB2가 위치 투명성을 제공하는 수단의 일부입니다. 별명은 데이터 소스 위치 정보에 대한 서버 정의를 사용하여 데이터 소스를 찾고 효과적으로 액세스할 수 있습니다. 예를 들어, ALTER SERVER문은 새 별명 없이 또는 응용프로그램 코드를 변경하지 않고 서버 성능 데이터와 모든 사용자 및 응용프로그램에 대한 버전 정보를 투명하게 갱신할 수 있습니다.

별명은 제어 센터 또는 명령행 처리기에서 작성됩니다. 같은 데이터 소스 테이블 또는 뷰에 대해 둘 이상의 별명을 정의할 수 있습니다.

별명은 정적 SQL문에서 사용될 수 없습니다.

별명을 작성하기 전에 데이터 소스에서 RUNSTATS 명령과 동등한 명령을 수행하고 데이터 소스 오브젝트에 대한 통계를 갱신하십시오. 통계적 정보는 별명이 작

성되고 연합 데이터베이스 카탈로그에 저장될 때 데이터 소스에서 수집됩니다. 이 카탈로그 데이터에는 테이블, 컬럼 정의, 그리고 사용 가능한 경우 색인 정의와 통계가 들어있습니다.

다음 SQL문은 별명 CUSTOMER를 작성합니다.

```
CREATE NICKNAME CUSTOMER for OS390A.SHAWNB.CUSTLIST
```

SYSADM 또는 DBADM 권한 중 하나를 보유하거나 연합 데이터베이스에서 IMPLICIT_SCHEMA 데이터베이스 특권 또는 CREATEIN 스키마 특권(현재 스키마에 대해)이 있어야 이 명령문을 사용할 수 있습니다.

CREATE NICKNAME문 사용 방법에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

별명 및 데이터 소스 오브젝트 참조

데이터 소스 오브젝트에 대한 참조는 일반적으로 정의된 별명을 사용합니다. 단 pass-through 세션에서의 참조는 예외입니다(자세한 정보는 183 페이지의 『서버로 통과 세션 사용』 참조). 예를 들어, 데이터 소스 테이블 DB2MVS1.PERSON.DEPT에 대해 별명 DEPT를 정의하면, SELECT * FROM DEPT문을 사용할 수 있습니다. SELECT * FROM DB2MVS1.PERSON.DEPT문은 사용할 수 없습니다.

별명 및 데이터 소스 오브젝트와의 작업

대부분의 유틸리티 명령(LOAD, IMPORT, EXPORT, REORGCHK, REORGANIZE TABLE)은 별명을 지원하지 않습니다.

COMMENT ON은 지원됩니다. 이는 연합 데이터베이스에서 시스템 카탈로그를 갱신합니다.

INSERT, UPDATE 및 DELETE 조작용 별명에 대해 지원되지 않습니다.

기존의 별명 및 데이터 소스 식별

여러 개의 별명을 작성한 후, 다음 정보를 사용하여 주어진 별명이 해당되는 데이터 소스를 식별하거나 주어진 데이터 소스에서 모든 별명을 식별할 수 있습니다.

별명 및 데이터 소스 식별

이 예에서는 사용자가 별명(PAYROLL)과 작성자(ACCTG)를 알고 있지만, 데이터 소스에 대해서 추가 정보가 필요하다고 간주합니다. 다음 SQL문을 사용하여 먼저 PAYROLL이 데이터 소스(SERVER)에서 어떻게 알려져 있는지에 대한 정보를 확보하십시오.

```
select option, setting
       from syscat.taboptions
  where tablename = 'PAYROLL'
        and tabschema = 'ACCTG'
        and option in ('SERVER', 'REMOTE_SCHEMA', 'REMOTE_TABLE');
```

이 명령문에서 응답 세트는 DB2_MVS, FINANCE, DEPTJ35_PAYROLL입니다. 이제 PAYROLL이 DB2_MVS라고 하는 서버에서 FINANCE가 소유한 DEPTJ35_PAYROLL이라는 테이블의 별명임을 알 수 있습니다. 후속 SELECT 문에서 이 정보를 사용할 수 있습니다.

```
select option,setting
       from syscat.serveroptions
  where servername = 'DB2_MVS'
        and option in ('NODE','DBNAME');
```

이 명령문에서 응답 세트는 REGIONW 및 DB2MVSDB3입니다. 이제 테이블 DEPTJ35_PAYROLL이 REGIONW라고 하는 노드에서 DB2MVSDB3이라고 하는 데이터베이스에 있다는 것을 알 수 있습니다.

이 정보를 가지고 LIST NODE DIRECTORY 명령을 사용하여 사용된 통신 프로토콜 및 보안 유형과 같은 REGIONW 노드에 대한 정보를 얻을 수 있습니다. 노드가 DB2 계열이 아닌 데이터 소스를 위한 것이 있으면, 해당 데이터 소스의 구성 파일을 점검하여 비슷한 정보를 찾아야 합니다. 예를 들어, 노드가 Oracle 데이터 소스를 위한 것이면, 사용자는 Oracle tnsnames.ora 파일에서 비슷한 정보를 얻을 수 있습니다.

시스템 카탈로그 뷰의 세부사항에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

DB2에 알려진 모든 별명 식별

다음 SQL문은 각 별명에 대한 스키마 이름과 원격 서버를 비롯한 연합 데이터베이스에 알려진 모든 별명의 목록을 제공합니다.

```
select tabname, tabschema, setting as remote_server
       from syscat.taboptions
       where option = 'SERVER';
```

색인, 색인 확장 또는 색인 스펙 작성

색인은 행 위치의 목록으로서 하나 이상 지정된 컬럼의 내용에 의해 분류됩니다. 색인은 일반적으로 테이블에 대한 액세스 속도를 높이기 위해 사용됩니다. 그러나, 논리 데이터 설계 목적으로도 사용될 수 있습니다. 예를 들어, 고유 색인은 컬럼에 값이 중복된 항목이 오도록 허용하지 않기 때문에, 테이블의 어떠한 행도 같지 않다고 보증합니다. 색인은 컬럼 내의 값을 오름차순 또는 내림차순으로 지정하기 위해 작성될 수도 있습니다.

색인 확장은 구조화된 유형이나 구별 유형 컬럼을 갖는 색인과 함께 사용하는 색인 오브젝트입니다.

색인 스펙은 메타데이터 구성입니다. 이는 최적화 알고리즘에 데이터 소스 오브젝트(테이블 또는 뷰)에 대해 별명으로 참조되는 색인이 있다는 것을 알려줍니다. 색인 스펙에는 행 위치 목록이 없습니다. 이는 색인의 설명일 뿐입니다. 최적화 알고리즘은 색인 스펙을 사용하여 별명으로 참조되는 오브젝트에 대한 액세스를 향상시킵니다. 별명이 처음으로 작성되면, 데이터 소스에서 기초가 되는 테이블에 대해 DB2에서 인식할 수 있는 형식으로 이루어진 색인이 있을 때 색인 스펙이 생성됩니다.

주: 필요할 경우, 테이블 별명 또는 하나의 테이블상에 뷰가 있는 뷰 별명에서 색인 스펙을 작성하십시오.

다음과 같은 경우에 색인 또는 색인 스펙을 수동으로 작성하십시오.

- 성능을 향상시키려는 경우. 예를 들어, 최적화 알고리즘이 특정 테이블 또는 별명을 중첩된 루프 조인의 내부 테이블로 사용하도록 하려면, 색인이 없을 때 조인하는 컬럼에서 색인 스펙을 작성하십시오. 색인 또는 색인 스펙을 원하는 경우 *관리 안내서*: 성능에서 자세한 내용을 참조하십시오.
- 기본 테이블에 대한 색인이 해당 테이블의 별명이 작성된 후에 추가된 경우.

색인 스펙은 기본 테이블에 색인이 없을 때 작성될 수 있습니다.(DB2는 CREATE INDEX문을 발행할 때 원격 색인을 점검하지 않습니다.) 색인 스펙은 UNIQUE 키워드가 지정되어도 행의 고유성을 강요하지는 않습니다.

DB2 색인 보조 프로그램은 최적의 색인 세트 선택을 지원하는 마법사입니다. 제어 센터를 통해 이 마법사에 액세스할 수 있습니다. 비교 가능한 유틸리티를 *db2advis*라고 합니다.

색인은 기본 테이블의 컬럼에 의해 정의됩니다. 이는 테이블 작성자 또는 특정 컬럼에 직접 액세스해야 한다는 점을 알고 있는 사용자가 정의할 수 있습니다. 사용자 정의 색인이 미리 존재하지 않는 한, 1차 색인 키가 자동으로 기본 키에 작성됩니다.

특정 기본 테이블에서 임의의 수의 색인을 정의할 수 있으며, 조회 성능에도 도움이 될 수 있습니다. 그러나 색인이 많을수록 데이터베이스 관리 프로그램은 갱신, 삭제 및 삽입 조작 동안에 더 많이 수정해야 합니다. 많은 내용이 갱신되는 테이블에 대해 대규모의 색인을 작성하면 요구 처리 시간이 더 길어질 수 있습니다. 그러므로, 잦은 액세스로 인해 분명한 이점이 있는 경우에만 색인을 사용하십시오.

색인 키의 일부인 컬럼은 255바이트로 제한됩니다. 색인에서 컬럼의 최대 수는 16입니다. 입력된 테이블을 색인하는 중이라면, 최대 컬럼 수는 15입니다. 색인 키의 최대 길이는 1024바이트입니다. 이전에 언급한 것처럼, 테이블에 있는 많은 색인 키는 요청 처리 속도를 느리게 할 수 있습니다. 유사하게, 역시 큰 색인 키는 요청 처리 속도를 느리게 할 수 있습니다.

색인 키는 색인이 정의된 컬럼의 콜렉션 또는 컬럼이며, 색인의 사용성을 결정합니다. 색인 키를 구성하는 컬럼 순서는 색인 키 작성과 다른 점이 없지만, 색인의 사용 여부를 결정하는 경우 최적화 알고리즘과는 다를 수 있습니다.

색인화된 테이블이 비어 있을 경우에도 색인이 계속 작성되지만, 테이블이 로드되거나 행이 삽입될 때까지 어떠한 색인 항목도 작성되지 않습니다. 테이블이 비어 있지 않으면, 데이터베이스 관리 프로그램은 CREATE INDEX문을 처리하는 동안에 색인 항목을 만듭니다.

클러스터링 색인의 경우, 비슷한 키 값을 가진 기존 행에 근접하여 새 행이 실제로 삽입됩니다. 이는 데이터 페이지에 대해 좀더 선형적인 액세스 패턴을 만들어 내고 좀더 효과적인 프리페치 결과를 가져오므로 조회시 성능상의 이점을 가져옵니다.

기본 키 색인이 클러스터링 색인이 되게 하려면, 기본 키가 CREATE TABLE에서 지정되지 않아야 합니다. 일단 기본 키가 작성되면, 관련 색인은 수정될 수 없습니다. 대신, 기본 키 절 없이 CREATE TABLE을 수행하십시오. 그런 다음, CREATE INDEX문을 발행하여 클러스터링 속성을 지정하십시오. 마지막으로, ALTER TABLE문을 사용하여 방금 작성한 색인에 해당하는 기본 키를 추가하십시오. 이 색인은 기본 키 색인으로서 사용됩니다.

일반적으로, 클러스터링 색인이 고유하므로 클러스터링이 더 효과적으로 유지보수됩니다.

고유 색인 키의 일부는 아니지만, 색인에 저장/유지보수되는 컬럼 데이터를 포함 컬럼이라고 합니다. 포함 컬럼은 고유 색인 전용으로 지정될 수 있습니다. 포함 컬럼으로 색인 작성시, 고유 키 컬럼만이 고유성을 위해 저장 및 고려됩니다. 포함 컬럼의 사용은 색인 액세스가 포함될 때 데이터 검색의 성능을 개선시킵니다.

데이터베이스 관리 프로그램은 맨 아래 레벨이 리프 노드로 구성된 B+ 트리 구조를 사용하여 색인을 저장합니다. 리프 노드 또는 페이지는 실제 색인 키 값이 저장되는 위치입니다. 색인을 작성할 때, 해당 색인 리프 페이지가 병합되거나 온라인으로 재구성되도록 할 수 있습니다. 온라인 색인 재구성은 여러 번의 삭제 및 갱신 활동 후 많은 색인 리프에 적은 수의 색인 키만 남는 상황을 방지하는 데 사용됩니다. 이러한 상황에서 온라인 재구성 없이 공간은 데이터 및 색인의 오프라인 재구성으로만 사용될 수 있습니다. 온라인으로 색인 페이지를 재구성하는 기능으로 색인을 작성할 지 여부를 결정할 때, 다음 질문을 고려해야 합니다. 키 삭제가 발생할 때마다 병합할 공간을 점검하는 성능 비용이 추가됩니까? 그리고 공간이 충분한 경우 병합을 완료할 실제 비용이 색인의 더 나은 공간 활용의 이점보다 크고, 공간을 재요구하는 오프라인 재구성의 감축된 비용보다 더 적습니까?

주: 온라인 재구성 병합 이후에 해제되는 페이지는 같은 테이블에서 기타 색인에 대해서만 다시 사용될 수 있습니다. 전체 재구성에서 해제되는 해당 페이지는 다른 오브젝트(데이터베이스 관리 저장영역으로 작업할 때) 또는 디스크 공간

(시스템 관리 저장영역으로 작업할 때)에서 사용될 수 있습니다. 이외에도, 온라인 재구성은 색인의 리프가 아닌 페이지를 해제하지 않는 반면, 전체 재구성은 색인을 가능한 작게 만들고, 리프가 아닌 및 리프 페이지뿐만 아니라 색인의 레벨 수도 줄입니다.

온라인으로 재구성하는 색인을 구현하는 방법에 대해서는 192 페이지의 『CREATE INDEX문 사용』에서 자세한 내용을 참조하십시오.

파티션된 데이터베이스의 테이블에 대한 색인은 동일한 CREATE INDEX문을 사용하여 구축됩니다. 이 색인은 테이블의 파티션 키에 의거하여 파티션됩니다. 테이블의 색인은 노드 그룹 내 각 노드의 해당 테이블의 지역 색인으로 구성됩니다. 다중 파티션 환경에서 정의된 고유 색인은 파티션 키의 상위집합이어야 한다는 사실을 주의하십시오.

성능 추가 정보: 다음 작업을 수행하려면, LOAD 유틸리티를 사용하기 전에 색인을 작성하십시오.

- 테이블 작성
- 테이블 로드
- 색인 작성
- RUNSTATS 수행

타스크 실행은 다음 방법순으로 한다는 사실을 고려하십시오.

1. 테이블 작성
2. 색인 작성
3. 요청된 statistics yes 옵션을 사용하여 테이블 로드

LOAD 성능 향상에 대해서는 데이터 이동 유틸리티 안내 및 참조서에서 자세한 내용을 참조하십시오.

색인은 작성된 후 유지보수됩니다. 그 이후, 응용프로그램이 키 값을 사용하여 테이블 내의 행을 임의로 액세스하고 처리할 때, 해당 키 값에 따른 색인은 행을 직접 액세스하는 데 사용될 수 있습니다. 이는 기본 테이블에서 행의 물리적 저장영역이 순서화되지 않았기 때문에 매우 중요합니다. 클러스터링 색인이 정의되지 않은 경우 행이 삽입되면, 행이 들어갈 수 있는 가장 편리한 저장영역 위치에 이 행

이 위치합니다. 특정 선택 조건에 부합하는 테이블 행을 검색하고 테이블에 색인이 없으면, 전체 테이블은 스캔됩니다. 색인은 순차 검색을 길게 수행하지 않고도 데이터 검색을 최적화할 수 있습니다.

색인에 대한 데이터는 테이블 데이터와 동일한 테이블 공간에 저장되거나, 색인 데이터를 담고 있는 별도의 테이블 공간에 저장될 수 있습니다. 색인 데이터를 저장하는 데 사용되는 테이블 공간은 테이블이 작성될 때 결정됩니다(155 페이지의 『다중 테이블 공간에 테이블 작성』 참조).

제어 센터를 사용하여 색인을 작성하십시오.

- | |
|--|
| <ol style="list-style-type: none">1. 색인 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.2. 색인 폴더를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 작성 → 마법사를 사용한 색인을 선택하십시오.3. 마법사의 단계에 따라 작업을 완료하십시오. |
|--|

명령행을 사용하여 색인을 작성하십시오.

```
CREATE INDEX <name> ON <table_name> (<column_name>)
```

다음 두 절 『색인 사용』 및 192 페이지의 『CREATE INDEX문 사용』에서는 색인 작성에 대해 좀더 자세히 설명합니다.

색인 사용

색인은 응용프로그램이 직접 사용하지 않습니다. 색인의 사용 여부와 잠재적으로 사용 가능한 색인을 결정하는 것은 최적화 알고리즘의 책임입니다.

한 테이블에서 최적의 색인은 다음과 같은 색인입니다.

- 고속 디스크를 사용하는 색인
- 고도로 클러스터링된 색인
- 몇 개의 좁은 컬럼으로만 구성되는 색인

색인을 효과적으로 사용하는 방법에 관한 자세한 정보는 [관리 안내서: 성능의 『색인 스캔 개념』](#)을 참조하십시오.

CREATE INDEX문 사용

사용자는 고유 키가 아닌 컬럼으로 좀더 효과적으로 검색할 수 있도록 중복(비고 유 색인)을 허용하고 색인화된 컬럼에 중복 값이 존재할 수 있도록 색인을 작성할 수 있습니다.

다음의 SQL문은 EMPLOYEE 테이블의 LASTNAME 컬럼으로부터 LNAME이라는 고유하지 않은 오름차순으로 정렬된 색인을 작성합니다.

```
CREATE INDEX LNAME ON EMPLOYEE (LASTNAME ASC)
```

다음 SQL문은 전화번호 컬럼에 고유한 색인을 작성합니다.

```
CREATE UNIQUE INDEX PH ON EMPLOYEE (PHONENO DESC)
```

고유 색인은 색인화된 컬럼에 중복값이 존재하지 않도록 합니다. 제한조건은 행을 갱신하거나 새 행을 삽입하는 SQL문의 끝에 강제됩니다. 이러한 유형의 색인은 이미 하나 이상의 중복 값을 가지고 있는 컬럼에서는 작성될 수 없습니다.

키워드 ASC는 컬럼에 의해 오름차순으로 색인 항목을 입력하고, DESC는 컬럼에 의해 내림차순으로 항목을 입력합니다. 기본값은 오름차순입니다.

구조화 유형으로 작업할 때 사용자 정의 색인 유형을 작성해야 합니다. 여기서는 색인 유지보수, 색인 검색 및 색인 사용 기능 정의 수단이 필요합니다. 색인 유형 작성에 대한 요구사항에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

다음 SQL문은 EMPLOYEE 테이블의 LASTNAME 컬럼에 INDEX1이라고 하는 클러스터링 색인을 작성합니다.

```
CREATE INDEX INDEX1 ON EMPLOYEE (LASTNAME) CLUSTER
```

효과적으로 수행하려면, ALTER TABLE문과 연관된 PCTFREE 매개변수와 함께 클러스터링 색인을 사용하여 클러스터링 순서를 유지보수하는 올바른 페이지에 새 데이터가 삽입될 수 있도록 하십시오. 일반적으로 테이블에서의 INSERT 활동이 더 클수록 클러스터링을 유지보수하는 데 필요한 (테이블의) PCTFREE 값이 더 커집니다. 이 색인은 데이터가 실제 페이지에 놓이는 순서를 결정하므로, 하나의 클러스터링 색인만이 특정 테이블에 정의될 수 있습니다.

반면, 예를 들어 이들 새로운 행의 색인 키 값은 항상 새로운 높은 키 값이므로, 테이블의 클러스터링 속성은 이들 색인 키 값을 테이블의 끝에 위치시키려고 합니다. 다른 페이지에 사용 가능한 공간이 있으면 클러스터링을 거의 보존하지 않습니다. 이 경우, 추가 모드에 테이블을 위치시키는 것이 색인을 클러스터링하고 테이블이 큰 PCTFREE 값을 갖도록 변경하는 것보다 나은 선택일 수 있습니다. ALTER TABLE APPEND ON을 발행하여 추가 모드에 테이블을 위치시킬 수 있습니다. ALTER TABLE에 관한 추가 개요 정보는 229 페이지의 『테이블 속성 변경』에서 자세한 내용을 참조하십시오. ALTER TABLE에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

또한, 위의 설명은 행의 크기를 증가시키는 UPDATE의 결과인 새로운 "오버플로우" 행에 적용합니다.

CREATE INDEX문의 MINPCTUSED절은 색인 리프 페이지에서 사용되는 최소 공간량에 대한 임계값을 지정합니다. 이 절이 사용되면, 온라인 색인 재구성이 이 색인에 대해 작동됩니다. 작동되면, 다음 고려사항을 사용하여 온라인 재구성이 이루어지는 지를 판별하게 됩니다. 키가 이 색인의 리프 페이지에서 삭제되고 페이지에서 사용된 공간의 비율이 지정된 임계값 보다 작으면, 이웃하는 색인 리프 페이지를 점검하여 두리프 페이지에 있는 키가 단일 색인 리프 페이지로 병합될 수 있는지를 판별합니다.

예를 들어, 다음 SQL문은 온라인 재구성이 작동된 상태에서 색인을 작성합니다.

```
CREATE INDEX LASTN ON EMPLOYEE (LASTNAME) MINPCTUSED=20
```

키가 이 색인에서 삭제될 때 색인 페이지의 나머지 키가 색인 페이지에서 20% 이하의 공간을 차지하면, 이 색인 페이지의 키와 이웃하는 색인 페이지의 키를 병합하여 색인 페이지를 삭제하려는 시도가 이루어집니다. 결합된 키가 모든 단일 페이지에 들어갈 수 있으면, 이 병합이 수행되고 색인 페이지 중 하나가 삭제됩니다.

CREATE INDEX문의 PCTFREE절은 색인이 작성될 때 사용 가능한 공간으로 남겨둘 각 색인 페이지의 비율을 지정합니다. 색인 페이지에 사용 가능한 공간을 더 많이 두면 페이지 파티션이 줄어듭니다. 그러면, 프리페치를 증가시키는 순차 색인 페이지를 다시 얻기 위해 테이블을 재구성할 필요성이 줄어듭니다. 그리고, 프리페치는 성능을 향상시키는 주요 구성요소입니다. 또한, 항상 높은 키 값이 있는

경우, CREATE INDEX문의 PCTFREE의 값을 낮추는 방안을 고려해 볼 수 있습니다. 이렇게 하면, 각 색인 페이지별 소모 공간을 제한할 수 있습니다.

다중 파티션 데이터베이스에서, 고유 색인은 파티션 키의 상위 집합으로 정의된 상태여야 합니다.

복제된 요약 테이블을 가지고 있는 경우, 해당 기본 테이블은 고유 색인을 가져야 하며 복제된 요약 테이블을 정의하는 조회에 색인 키 컬럼을 사용해야 합니다. 관리 안내서: 계획에서 복제된 요약 테이블에 대한 자세한 내용을 참조하십시오.

파티션 내 병렬 처리의 경우, 색인 작성중에 수행된 데이터 스캐닝 및 정렬에 다중 프로세서를 사용하여 색인 작성 성능이 향상됩니다. 다중 프로세서는 `intra_parallel`을 YES(1) 또는 ANY(-1)로 설정하여 사용할 수 있습니다. 색인 작성 중에 사용되는 프로세서 수는 시스템에 의해 결정되며, 구성 매개변수 `dft_degree`, `max_querydegree`, 응용프로그램 런타임 정도 또는 SQL문 컴파일 정도에 의해 영향을 받지 않습니다. 데이터베이스 구성 매개변수 `index sort`가 NO로 설정될 경우, 색인 작성은 다중 프로세서를 사용하지 않습니다.

사용자 정의 확장 색인 유형 작성

사용자 정의 색인 유형을 지원하기 위해, DB2 Universal Database는 색인이 작업하는 방식을 구성하는 기본 구성요소에 대해 사용자 자신의 논리를 작성하여 적용하게 합니다. 대체될 수 있는 구성요소는 다음과 같습니다.

- 색인 유지보수. 이것은 색인 컬럼 내용을 색인 키에 매핑하는 기능을 허용합니다. 그러한 매핑은 사용자 정의 매핑 기능을 통해 수행됩니다. 정확하게 하나의 구성 유형 컬럼이 확장 색인에 참여할 수 있습니다. 일반 색인과 달리, 확장 색인을 행당 둘 이상의 색인 항목을 가질 수 있습니다. 행당 여러 색인 항목은 문서에 있는 각 키워드의 각 색인 항목이 있는 오브젝트로서 텍스트 문서를 저장하게 할 수 있습니다.
- 색인 개발. 이것은 응용프로그램 디자이너가, 임의의 사용자 정의 함수(UDF)와 필터링 조건(범위 술어)을 연관시키게 하며, UDF는 필터링 조건과 연관되지 않으면 최적화 알고리즘에 불투명하게 됩니다. 이것은 DB2 각 행에 대한 개별 UDF 호출을 작성하지 못하게 하고 클라이언트 및 서버 간의 문맥 전환을 하지 못하게 하여, 성능을 크게 개선시킵니다.

주: 사용자 정의 함수(UDF) 정의는 결정적이어야 하며 최적화 알고리즘으로 검색될 수 있는 외부 조치를 허용하지 말아야 합니다.

선택적 데이터 필터 기능도 지정될 수 있습니다. 최적화 알고리즘은 사용자 정의 함수(UDF)가 평가되기 전에 폐치된 튜플에 대해 필터를 사용합니다.

구조화된 유형이나 구별 컬럼 유형만이 색인 확장을 사용하여 이 오브젝트에서 사용자 정의 확장 색인 유형을 작성할 수 있습니다. 사용자 정의 확장 색인 유형은 다음이 아니어야 합니다.

- 클러스터된 색인과 정의된 상태
- INCLUDE 컬럼

색인 유지보수의 세부사항

CREATE INDEX EXTENSION문을 통해 색인의 조사를 구성하는 두개의 구성 요소를 정의합니다.

색인 유지보수는 색인 컬럼 내용(또는 소스 키)을 목표 색인 키로 변환하는 프로세스입니다. 변환 프로세스는 이전에 데이터베이스에서 정의된 테이블 함수를 사용하여 정의됩니다.

FROM SOURCE KEY 절은 이 색인 확장으로 지원되는 소스 키 컬럼에 대한 구조화된 데이터 유형이나 구별 유형을 지정합니다. 단일 매개변수 이름 및 데이터 유형이 제공되며 소스 키 컬럼과 연관됩니다.

GENERATE KEY USING 절은 색인 키를 생성하기 위해 사용되는 사용자 정의 테이블 함수를 지정합니다. 이 함수의 출력은 TARGET KEY 절 스펙(아래 표 시됨)에 지정되어야 합니다. 이 함수의 출력은 FILTER USING 절에서 지정한 색인 필터링 함수의 입력으로서 사용될 수 있습니다.

색인 검색의 세부사항

색인 검색은 검색 인수를 검색 범위에 맵핑합니다.

WITH TARGET KEY 절은 GENERATE KEY USING 절에 지정된 키 생성 함수의 출력인 목표 키 매개변수를 지정합니다. 단일 매개변수 이름 및 데이터 유

형이 제공되며 목표 키 컬럼과 연관됩니다. 이 매개변수는 GENERATE KEY USING 절의 사용자 정의 테이블 함수의 RETURNS 테이블 컬럼에 대응합니다.

SEARCH METHODS 절은 색인용으로 정의된 하나 이상의 검색 메소드를 소개합니다. 각 검색 메소드는 메소드 이름, 검색 인수, 함수 생성 범위 및 선택적 색인 필터 함수로 구성됩니다. 각 검색 메소드는 주요 사용자 정의 색인의 색인 검색 범위가 사용자 정의 테이블 함수에 의해 생성되는 방식을 정의합니다. 더우기, 각 검색 메소드는 특정 검색 범위에 있는 색인 항목이 사용자 정의 스칼라 함수에 의해 자세히 규정될 수 있는 방식을 정의합니다.

- WHEN 절은 레이블을 검색 메소드와 연관시킵니다. 레이블은 색인 검색 규칙(사용자 정의 함수(UDF)의 PREDICATES 절에 있음)에 지정된 메소드 이름과 관련된 SQL 식별자입니다. 하나 이상의 매개변수 이름과 데이터 유형은 범위 함수 및/또는 색인 필터링 함수에서 인수로서 사용하도록 제공됩니다. WHEN 절은 CREATE FUNCTION문의 PREDICATES 절이 입력 조회와 일치할 때 최적화 알고리즘이 취할 수 있는 조치를 지정합니다.
- RANGE THROUGH 절은 색인 키 범위를 생성하는 사용자 정의 외부 테이블 함수를 지정합니다. 이것은 색인 키가 키 범위를 벗어날 때 최적화 알고리즘이 연관된 UDF를 호출하지 못하게 할 수 있습니다.
- FILTER USING 절은 범위 생성 함수에서 리턴된 색인 항목을 필터하기 위해 사용되는 CASE 표현식 또는 사용자 정의 외부 테이블 함수를 지정하는 선택적 방법입니다. 색인 필터 함수나 CASE 표현식에 의해 리턴된 값이 1이면, 색인 항목에 대응하는 행이 테이블에서 검색됩니다. 리턴되는 값이 1이 아닌 경우, 색인 항목이 삭제됩니다. 이 기능은 두번째 필터의 비용이 원래 메소드를 평가하는 비용과 비교할 때 더 낮고 보조 필터의 선택성이 상대적으로 낮을 때 유용합니다.

색인 검색 세부사항

색인 검색은 검색 메소드의 평가에서 발생합니다.

CREATE FUNCTION(외부 스칼라)문은 색인 확장용으로 정의된 검색 메소드에서 사용되는 사용자 정의 술어를 작성합니다.

PREDICATES 절은 색인 확장을 검색할 수 있는(그리고 술어의 검색 조건에 대해 선택적 SELECTIVITY 절 사용) 이 함수를 사용하여 술어를 식별합니다. PREDICATES 절이 지정되면, 함수는 NO EXTERNAL ACTION을 사용한 DETERMINISTIC으로서 정의되어야 합니다.

- WHEN 절은 비교 연산자(=, >, < 및 기타) 및 상수나 표현식(EXPRESSION AS 절 사용)이 있는 술어에서 정의되는 함수의 특수한 사용을 소개합니다. 술어가 동일한 비교 연산자와 제공된 상수나 표현식과 함께 이 함수를 사용하는 경우, 필터링 및 색인 검색이 사용될 수 있습니다. 상수의 사용은 결과 유형이 1 또는 0 중 하나인 부울 표현식을 주로 다루기 위해 제공됩니다. 다른 모든 경우에는, EXPRESSION AS 절이 더 낫습니다.
- FILTER USING 절은 결과 테이블의 추가 필터링을 수행하기 위해 사용할 수 있는 필터 함수를 식별합니다. 행이 규정하는지 여부를 판별하기 위해 사용자 정의 술어를 실행해야 하는 행의 수를 감축하는 정의된 함수(술어에서 사용)를 대체하는 더 빠른 버전입니다. 사용자 정의 술어에 의해 예상되는 결과에 근접하는 색인으로 결과를 생성한 경우, 이 필터 함수의 응용프로그램은 중복될 수 있습니다.
- 색인 확장의 각 검색 메소드가 색인을 검색하는 규칙 세트를 선택적으로 정의할 수 있습니다. 또한 검색 목표, 검색 인수, 그리고 색인 검색을 수행하기 위해 이들을 사용하는 방법을 기술하기 위해 색인 확장에서 검색 메소드를 정의할 수 있습니다.
 - SEARCH BY INDEX EXTENSION 절은 색인 확장을 식별합니다.
 - 선택적 EXACT 절은 색인 찾아보기가 술어 평가에서 일치함을 나타냅니다. 이 절은 색인 찾아보기 이후에 데이터베이스가 필터 함수나 원래의 사용자 제공 술어 함수를 적용하지 않도록 알립니다. 색인 찾아보기가 사용되지 않는 경우, 원래의 술어 및 필터 함수가 적용되어야 합니다. EXACT 절이 사용되지 않는 경우, 원래의 사용자 제공 술어는 색인 찾아보기 이후에 적용됩니다. EXACT 술어는 색인 찾아보기가 술어와 동일한 결과를 리턴할 때 유용합니다. 이것은 조회 실행 시 색인 찾아보기에서 획득한 결과에 사용자 정의 술어를 적용하지 못하게 합니다. 색인이 술어와 비슷한 것만을 제공하리라고 예상되는 경우, EXACT 절을 지정하지 마십시오.

- WHEN KEY 절은 검색 목표를 정의합니다. 키에 대해 오직 하나의 검색 목표가 지정됩니다. WHEN KEY 절 다음에 제공된 값은 정의되는 함수의 매개변수 이름을 식별합니다. 이름 지정된 매개변수의 값이 지정된 색인 확장에 근거한 색인으로 처리될 때 이 절은 참으로 평가됩니다.
- USE 절은 검색 인수를 정의합니다. 검색 인수는 색인 확장에 정의된 어떤 메소드가 사용되는가를 식별합니다. 여기에서 제공한 메소드 이름은 색인 확장에 정의한 메소드와 일치해야 합니다. 하나 이상의 매개변수 값은 정의되는 함수의 매개변수 이름을 식별하며 어떤 것이 검색 목표에 지정한 매개변수 이름과 달라야 하는지 나타냅니다. 매개변수 값의 수와 각 데이터 유형은 색인 확장에서 메소드용으로 정의한 매개변수와 일치해야 합니다. 내장 및 구별 데이터 유형의 경우 완전히 일치해야 하며 동일한 구조 유형 내에 있어야 합니다.

색인 확장을 정의하는 시나리오

색인 확장을 정의하는 시나리오는 다음과 같습니다.

1. 구조화된 유형(형태의 경우)을 정의하십시오. CREATE TYPE문을 사용하여 형태가 supertype 및 nullshape, 포인트, 라인이며, 다각형이 부속유형인 유형 계층구조를 정의하십시오. 이것은 유형 모델 공간 엔터티를 구조화했습니다. 예를 들어, 상점의 위치는 포인트이며, 강의 경로는 라인이며, 비즈니스 존의 경계는 다각형입니다. 최소 경계의 사각형(mbr)은 속성입니다. gtype 속성은 연관된 엔터티가 포인트, 라인 또는 다각형인지 여부를 지정합니다. 지리적인 경계는 numpart, numpoint 및 geometry 속성으로 모델화됩니다. 다른 모든 속성은 이 시나리오와 관련이 없으므로 무시됩니다.
2. 색인 확장을 작성하십시오.
 - CREATE FUNCTION문을 사용하여 키 변환(gridentry), 범위 생성(gridrange) 및 색인 필터(checkduplicate 및 mbroverlap)에 사용되는 함수를 작성하십시오.
 - CREATE INDEX EXTENSION 문을 사용하여 색인의 구성요소에 필요한 나머지를 작성하십시오.
3. 색인의 색인 유지보수 구성요소에 대응하는 키 변환을 작성하십시오.

```

CREATE INDEX EXTENSION iename (parm_name datatype, ...)
FROM SOURCE KEY (parm_name datatype)
GENERATE KEY USING table_function_invocation
...

```

FROM SOURCE KEY 절은 키 변환의 매개변수 및 데이터 유형을 식별합니다. GENERATE KEY USING 절은 함수에서 생성된 값으로 소스 키를 맵핑하는 데 사용되는 함수를 식별합니다.

4. 색인의 색인 검색 구성요소에 대응하는 범위 생성 및 색인 필터 함수를 정의하십시오.

```

CREATE INDEX EXTENSION iename (parm_name datatype, ...)
...
WITH TARGET KEY
WHEN method_name (parm_name datatype, ...)
RANGE THROUGH range_producing_function_invocation
FILTER USING index_filtering_function_invocation

```

WITH TARGET KEY 절은 검색 메소드 정의를 식별합니다. WHEN 절은 메소드 이름을 식별합니다. RANGE THROUGH 절은 사용되는 색인의 범위를 제한하는 데 사용되는 함수를 식별합니다. FILTER USING 절은 결과 색인 값에서 불필요한 항목을 제거하는 데 사용되는 함수를 식별합니다.

주: FILTER USING 절은 색인 필터링 함수 대신 CASE 표현식을 식별할 수 있습니다.

5. 색인 확장을 검색하려면 술어를 정의하십시오.

```

CREATE FUNCTION within (x shape, y shape)
RETURNS INTEGER
...
PREDICATES
WHEN = 1
FILTER USING mbrWithin (x..mbr..xmin, ...)
SEARCH BY INDEX EXTENSION grid_extension
WHEN KEY (parm_name) USE method_name(parm_name)

```

PREDICATES 절은 각 WHEN 절로 시작되는 하나 이상의 술어를 소개합니다. WHEN 절은 상수 또는 EXPRESSION AS 절 중 하나가 다음에 오는 비교 연산자를 사용하여 술어에 대한 스펙을 시작합니다. FILTER USING 절은 결과 테이블의 추가 필터링을 수행하기 위해 사용할 수 있는 필터 함수를 식별합니다. 이것은 규정하는 행을 판별하기 위해 사용자 정의 술어가 실행되

어야 하는 행의 수를 감축하는 정의된 함수(술어에서 사용)의 비용이 저렴한 버전입니다. SEARCH BY INDEX EXTENSION 절은 색인 검색이 발생하는 곳을 지정합니다. 색인 검색은 색인을 검색하기 위해 사용될 수 있는 색인 확장의 검색 메소드를 사용하여 규칙 세트를 정의합니다. WHEN KEY 절은 검색 규칙을 지정합니다. 검색 규칙은 검색 메소드를 통해 색인 검색을 수행하는데 사용될 수 있는 방법은 물론 검색 목표 및 검색 인수를 설명합니다.

6. 필터 함수를 정의하십시오.

```
CREATE FUNCTION mbrWithin (...)
```

여기에 정의된 함수는 색인 확장의 술어에서 사용하도록 작성됩니다.

조회 최적화 알고리즘이 작성된 색인을 정상적으로 검색하여 조회 성능을 개선시키기 위해, SELECTIVITY 옵션이 함수 호출에서 사용 가능합니다. 술어가 리턴할 수 있는 행의 백분율에 대한 아이디어가 있는 경우에, 함수 호출에서 SELECTIVITY 옵션을 사용하여 DB2 최적화 알고리즘이 더 효과적인 액세스 경로를 선택할 수 있도록 도울 수 있습니다.

다음 예에서, within 사용자 정의 함수는 중심 및 반경(각각 첫번째와 두번째 매개변수에 근거하여)을 계산하고 적절한 선택으로 명령문 문자열을 빌드합니다.

```
SELECT * FROM customer
WHERE within(loc, circle(100, 100, 10) = 1 SELECTIVITY .05
```

이 예에서, 표시된 술어(SELECTIVITY .05)는 customer 테이블에서 95퍼센트 행을 필터링합니다.

제4장 데이터베이스 변경

이 장은 데이터베이스를 변경하기 전에 고려해야 하는 사항과 데이터베이스 오브젝트를 변경하거나 삭제하는 방식에 초점을 둡니다.

데이터베이스를 변경하기 전에

데이터베이스 설계가 구현된 후 데이터베이스 설계를 변경하고자 하는 경우가 있습니다. 이전 설계와 관련된 주요 설계 문제를 재검토해야 합니다. 다음 사항에 특히 주의를 기울여야 합니다.

- 『논리 및 물리 설계 특성 변경』
- 『사용권 정보 변경』
- 『인스턴스 변경』
- 206 페이지의 『환경 변수 및 프로파일 레지스트리 변수 변경』
- 206 페이지의 『노드 구성 파일 변경』
- 206 페이지의 『데이터베이스 구성 변경』

논리 및 물리 설계 특성 변경

전체 데이터베이스에 영향을 주는 변경을 수행하기 전에, 모든 논리 및 물리 설계 결정을 검토해야 합니다. 예를 들어, 테이블 공간을 변경할 때에는 SMS 또는 DMS 저장영역 유형과 관련하여 설계 결정을 검토해야 합니다(자세한 내용은 [관리 안내서: 계획](#)을 참조하십시오).

사용권 정보 변경

DB2 제품 사용권 관리의 일부로서, 사용자는 사용권 수를 늘려야 하는 필요성을 느끼게 됩니다. 제어 센터에 있는 사용권 센터를 사용하여 설치된 제품의 사용량을 점검하고 해당 사용량을 근거로 사용권의 수를 늘릴 수 있습니다.

인스턴스 변경

인스턴스는 제품의 후속 설치 및 제거의 영향에서 가능한 벗어나도록 설계됩니다.

대부분의 경우, 기존 인스턴스는 설치 또는 제거되는 제품의 기능을 상속하거나 이에 대한 액세스를 유실합니다. 그러나, 특정 실행 파일 또는 구성요소가 설치되거나 제거되면, 기존의 인스턴스는 새 시스템 구성 매개변수를 자동으로 상속하거나 모든 추가 기능에 대한 액세스를 확보하지는 않습니다. 인스턴스는 갱신되어야 합니다.

DB2가 프로그램 임시 수정(PTF) 또는 패치를 설치하여 갱신되는 경우, 기존의 모든 DB2 인스턴스는 **db2iupdt** 명령을 통해 갱신되어야 합니다. **dasiupdt** 명령을 사용하여 관리 서버(DAS)도 갱신해야 합니다.

인스턴스를 변경하거나 삭제하기 전에 인스턴스와 인스턴스 내에 있는 데이터베이스 파티션 서버에 대해 이해하고 있어야 합니다.

인스턴스 나열

시스템에서 사용할 수 있는 모든 인스턴스 목록을 확보하려면, 제어 센터를 사용하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 목록 인스턴스를 원하는 데이터베이스를 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 추가를 선택하십시오.
3. 새로 고침을 클릭하고, 데이터베이스 이름 필드의 끝에서 화살표를 클릭하여 인스턴스 목록을 참조하십시오.
4. 취소를 누르십시오.

시스템에서 사용할 수 있는 모든 인스턴스 목록을 확보하려면, 명령행을 사용하십시오.

```
db2ilist
```

현재 세션에 적용되는 인스턴스를 판별하려면(OS/2 또는 지원되는 Windows 플랫폼에서), 다음을 사용하십시오.

```
set db2instance
```

인스턴스 구성 갱신

db2iupdt 명령을 수행하면, 다음을 통해 지정된 인스턴스를 갱신합니다.

- 인스턴스 소유자의 홈 디렉토리에 있는 sqllib 서브디렉토리의 파일을 대체합니다.
- 노드 유형이 변경되면, 새 데이터베이스 관리 프로그램 구성 파일이 작성됩니다. 이는 기존의 데이터베이스 관리 프로그램 구성 파일에서 관련 값을 새 노드 유형에 대한 데이터베이스 관리 프로그램 구성 파일과 병합하여 이루어집니다. 새 데이터베이스 관리 프로그램 구성 파일이 작성되면, 이전 파일은 인스턴스 소유자의 홈 디렉토리에 있는 sqllib 서브디렉토리의 backup 서브디렉토리로 백업됩니다.

db2iupdt 명령은 버전 및 릴리스 서브디렉토리의 instance 서브디렉토리에서 찾을 수 있습니다(정확한 이름은 운영 체제마다 다름).

명령은 다음과 같이 사용됩니다.

```
db2iupdt InstName
```

InstName은 인스턴스 소유자의 로그인 이름입니다.

이 명령에 관련된 기타 선택적 매개변수가 있습니다.

- -h 또는 -?

이 명령에 대한 도움말 메뉴를 표시합니다.
- -d

문제점 판별 중에 사용할 디버그 모드를 설정합니다.
- -a AuthType

인스턴스에 대한 인증 유형을 지정합니다. 유효한 인증 유형은 SERVER, CLIENT, DCS 또는 DCE입니다. 지정되지 않은 경우 DB2 서버가 설치되면, 기본값은 SERVER입니다. 그렇지 않으면, CLIENT로 설정됩니다. 인스턴스의 인증 유형은 인스턴스에 속하는 모든 데이터베이스에 적용됩니다.

UNIX 운영 체제에서, DCE는 유효한 인증 유형이 아닙니다.
- -e

존재하는 각 인스턴스를 갱신하도록 허용하며 **db2ilist**를 사용하여 표시될 수 있습니다.
- -u FencedID

분리(fenced) 사용자 정의 함수(UDF) 및 저장 프로시저가 실행되는 사용자를 지정합니다. 이는 DB2 클라이언트 또는 DB2 Software's Developer Kit을 설치할 때는 필요하지 않습니다. 기타 DB2 제품에서는 필수 매개변수입니다.

주: FencedID는 『root』 또는 『bin』이 될 수 없습니다.

- -k

이 매개변수는 현재 인스턴스 유형을 보존합니다. 이 매개변수를 지정하지 않으면, 현재 인스턴스는 다음 순서 중에서 가장 높은 인스턴스 유형으로 업그레이드됩니다.

- 지역 및 원격 클라이언트가 있는 파티션된 데이터베이스 서버(DB2 Enterprise - Extended Edition 기본 인스턴스 유형)
- 지역 및 원격 클라이언트가 있는 데이터베이스 서버(DB2 Universal Database Enterprise Edition 기본 인스턴스 유형)
- 클라이언트(DB2 클라이언트 기본 인스턴스 유형)

예:

- 인스턴스가 작성된 후 DB2 Universal Database Workgroup Edition 또는 DB2 Universal Database Enterprise Edition을 설치하면, 다음 명령을 입력하여 해당 인스턴스를 갱신하십시오.

```
db2iupdt -u db2fenc1 db2inst1
```

- 인스턴스를 작성한 후 DB2 Connect Enterprise Edition을 설치하면, 인스턴스 이름을 분리 ID로 사용할 수도 있습니다.

```
db2iupdt -u db2inst1 db2inst1
```

- 클라이언트 인스턴스를 갱신하려면, 다음 명령을 사용하십시오.

```
db2iupdt db2inst1
```

인스턴스 제거

제어 센터를 사용하여 인스턴스를 제거하십시오.

1. 제거하려는 인스턴스를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 인스턴스 이름을 오른쪽 마우스 버튼으로 클릭하고, 팝업 메뉴에서 제거를 선택하십시오.
3. **확정** 상자를 클릭하고 **확인**을 클릭하십시오.

명령행을 사용하여 인스턴스를 제거하십시오.

```
db2idrop <instance_name>
```

명령행을 사용하여 인스턴스를 제거하는 준비 및 세부사항은 다음과 같습니다.

1. 현재 인스턴스를 사용중인 모든 응용프로그램을 중지하십시오.
2. 각 DB2 명령 창에서 **db2 terminate** 명령을 수행하여 명령행 처리기를 중지시키십시오.
3. **db2stop** 명령을 수행하여 인스턴스를 중지시키십시오.
4. DB2INSTPROF 레지스트리 변수로 표시된 인스턴스 디렉토리를 백업하십시오. UNIX 운영 체제에서는 INSTHOME/sql1lib 디렉토리(여기서, INSTHOME은 인스턴스 소유자의 홈 디렉토리임)에 있는 파일의 백업을 고려하십시오. 예를 들어, 데이터베이스 관리 프로그램 구성 파일, db2system, db2nodes.cfg 파일, 사용자 정의 함수(UDF) 또는 강제 저장 프로시저 응용프로그램을 저장할 수 있습니다.
5. (UNIX 운영 체제에서만) 인스턴스 소유자로 로그오프하십시오.
6. (UNIX 운영 체제에서만) 루트 권한을 가진 사용자로 로그인하십시오.
7. **db2idrop** 명령을 발행하십시오.

```
db2idrop InstName
```

여기서, InstName은 삭제중인 인스턴스 이름입니다.

이 명령은 인스턴스 목록에서 인스턴스 항목을 제거한 후 인스턴스 디렉토리를 제거합니다.

8. (UNIX 운영 체제에서만) 선택적으로, 루트 권한을 가진 사용자로서 인스턴스 소유자의 사용자 ID 및 그룹(해당 인스턴스에 대해서만 사용된 경우)을 제거하십시오. 인스턴스를 다시 작성할 계획이면, 제거하지 마십시오.

이 단계는 인스턴스 소유자 및 인스턴스 소유자 그룹이 다른 목적으로 사용될 수 있으므로 선택적입니다.

db2idrop 명령은 인스턴스 목록에서 인스턴스 항목을 제거한 후, 인스턴스 소유자의 홈 디렉토리에 있는 sql1lib 서브디렉토리를 제거합니다.

환경 변수 및 프로파일 레지스트리 변수 변경

특정 운영 체제에서 변경되어야 하는 환경 변수를 고려해야 합니다(이러한 변수가 있을 경우). 환경 변수가 변경되고 현재 UNIX 플랫폼을 사용하지 않으면, 새 환경 변수를 적용하기 위해 시스템을 다시 시작해야 합니다. 데이터베이스를 변경하기 전에 전역 프로파일 레지스트리의 프로파일 레지스트리 변수를 재설정해야 하는지 검토하십시오. 그런 다음, 해당 프로파일 레지스트리 변수를 새 데이터베이스 환경에 맞는 최적 값으로 재설정할 수 있습니다. 프로파일 레지스트리 변수만이 변경될 경우, 시스템을 재시작할 필요가 없습니다.

노드 구성 파일 변경

노드 그룹을 변경하려는 경우(노드를 추가 및 삭제하거나 기존 노드를 이동) 수행할 작업에 대한 자세한 내용은 **관리 안내서: 성능의 『프로세서 추가를 통해 사용자 구성 확장』**에서 참조하십시오.

데이터베이스 구성 변경

데이터베이스를 변경하기로 계획중인 경우, 구성 매개변수 값을 검토해야 합니다. 데이터베이스 사용에 기초하여 수행중인 데이터베이스 변경의 일환으로서 때때로 일부 값을 조정할 수 있습니다.

데이터베이스 구성을 변경하려면, 제어 센터에 있는 성능 구성 마법사를 사용하십시오. 이 마법사는 어떤 구성 매개변수를 수정할 것인지 제안하고, 제안된 값을 제공함으로써 성능 조정을 하고, 인스턴스별 단일 데이터베이스의 메모리 요구사항의 균형을 맞추도록 도와줍니다.

주: 매개변수를 수정하려는 경우, 값은 다음과 같을 때까지 갱신되지 않습니다.

- 데이터베이스 매개변수의 경우, 모든 응용프로그램이 연결해제된 후에 데이터베이스에 대한 새로운 첫번째 연결
- 데이터베이스 관리 프로그램 매개변수의 경우, 인스턴스를 중지시킨 후 다시 시작할 때

대부분의 경우, 성능 구성 마법사에 의해 권장된 값은 기본값 보다 더 나은 성능을 제공하는데, 그 이유는 작업 로드 및 사용자의 특정 서버에 대한 정보에 근거한 값이기 때문입니다. 그러나, 값이 데이터베이스 시스템의 성능을 반드시 최적화

한 것은 아니지만, 향상할 목적으로 설계되었음을 주의하십시오. 최적화된 성능을 얻기 위해 조절이 가능한 시작 포인트로 생각하십시오.

제어 센터를 사용하여 데이터베이스 구성을 변경하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 변경하려는 인스턴스 또는 데이터베이스를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 마법사를 사용한 성능 구성을 선택하십시오.
3. 각 페이지를 클릭하고 필요하면 정보를 변경하십시오.
4. 결과 페이지를 클릭하여 작업을 검토하고 제시된 구성 매개변수를 적용하십시오.
5. 적용하는 갱신사항을 완료하면, 완료를 클릭하십시오.

명령행을 사용하여 데이터베이스 구성을 변경하십시오.

```
UPDATE DBM CFG FOR <database_alias>  
USING <config_keyword>=<value>
```

하나 이상의 <config_keyword>=<value> 조합을 단일 명령에서 갱신할 수 있습니다. 데이터베이스 구성 파일에 대한 대부분의 변경사항은 메모리로 로드된 후에만 유효합니다. 서버 구성 매개변수의 경우, 이것은 START DATABASE MANAGER 명령의 수행 중 발생합니다. 클라이언트 구성 매개변수의 경우, 이것은 응용프로그램이 재시작될 때 발생합니다.

현재 데이터베이스 관리 프로그램 구성 매개변수를 보거나 인쇄하려면, GET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오.

벤치마킹에 의한 시스템 조정 및 시스템 구성에 관한 세부사항은 관리 안내서: 성능의 『벤치마크 테스트』 및 『DB2 구성』을 참조하십시오.

다중 파티션의 경우: 둘 이상의 파티션에 걸쳐 파티션된 데이터베이스가 있으면, 구성 파일은 모든 데이터베이스 파티션에서 동일해야 합니다. SQL 컴파일러는 지역 노드 구성 파일의 정보에 근거하여 분산 SQL문을 컴파일하고, SQL문의 요구를 충족시킬 액세스 플랜을 작성하기 때문에 일관성이 요구됩니다. 데이터베이스 파티션에서 다른 구성 파일을 유지보수하는 작업은 명령문이 준비된 데이터베이스 파티션에 따라 다른 액세스 플랜이 될 수 있습니다. 구성 파일이 모든 데이터베이스 파티션에서 유지보수되게 하려면 **db2_all**을 사용하십시오.

데이터베이스 변경

데이터베이스를 변경할 때에는 데이터베이스를 작성할 때 만큼의 작업을 수행해야 합니다. 이 작업은 이전에 작성된 데이터베이스의 양상을 갱신하거나 삭제합니다. 수행할 작업은 다음과 같은 것이 있습니다.

- 『데이터베이스 삭제』
- 209 페이지의 『노드 그룹 변경』
- 209 페이지의 『테이블 공간 변경』
- 215 페이지의 『스키마 삭제』
- 216 페이지의 『구조 및 내용에서 테이블 수정』
- 231 페이지의 『사용자 정의 구조화 유형 변경』
- 231 페이지의 『입력된 테이블의 행 삭제 및 갱신』
- 231 페이지의 『기존 테이블 이름 바꾸기』
- 233 페이지의 『테이블 삭제』
- 235 페이지의 『트리거 삭제』
- 236 페이지의 『사용자 정의 함수(UDF), 유형 맵핑 또는 메소드 삭제』
- 236 페이지의 『사용자 정의 유형(UDT) 또는 유형 맵핑 삭제』
- 237 페이지의 『뷰 변경 또는 삭제』
- 239 페이지의 『요약 테이블 삭제』
- 241 페이지의 『서버 변경 또는 삭제』
- 241 페이지의 『별명 변경 또는 삭제』
- 243 페이지의 『색인, 색인 확장 또는 색인 스펙 삭제』
- 244 페이지의 『오브젝트 변경시의 명령문 종속성』

데이터베이스 삭제

비록 데이터베이스에 있는 일부 오브젝트를 변경할 수는 있어도, 데이터베이스 자체를 변경할 수는 없습니다. 삭제하거나 다시 작성해야 합니다. 데이터베이스를 삭제하게 되면 해당되는 모든 오브젝트, 컨테이너 및 관련 파일이 삭제되므로 파급 효과가 매우 커집니다. 삭제된 데이터베이스는 데이터베이스 디렉토리에서 제거(카탈로그 해제)됩니다.

제어 센터를 사용하여 데이터베이스를 삭제하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 데이터베이스를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. 확인 상자를 클릭하고 확인을 클릭하십시오.

명령행을 사용하여 데이터베이스를 삭제하십시오.

```
DROP DATABASE <name>
```

다음 명령은 데이터베이스 SAMPLE을 삭제합니다.

```
DROP DATABASE SAMPLE
```

주: SAMPLE 데이터베이스의 시험을 계속하려면, 이를 삭제하면 안됩니다. SAMPLE 데이터베이스를 삭제한 후 이를 다시 사용하려면, 다시 작성해야 합니다.

노드 그룹 변경

관리 안내서: 성능의 『프로세서 추가를 통해 사용자 구성 조정』장에서 노드 그룹 변경에 대한 자세한 내용을 참조하십시오.

노드를 추가하거나 삭제하면, 노드 그룹의 새로운 노드 세트에 현재 데이터를 재분배해야 합니다. 이를 수행하려면, REDISTRIBUTE NODEGROUP 명령을 사용하십시오. 관리 안내서: 성능의 『데이터베이스 파티션에서의 데이터 재분배』와 *Command Reference*에서 자세한 내용을 참조하십시오.

테이블 공간 변경

데이터베이스를 작성하면, 최소한 세 개의 테이블 공간을 작성하십시오. 카탈로그 테이블 공간(SYSCATSPACE), 사용자 테이블 공간(기본 이름은 USERSPACE1) 및 하나의 시스템 임시 테이블 공간(기본 이름은 TEMPSPACE1)입니다. 적어도 이 테이블 공간 중 하나가 있어야 합니다. 원하는 경우, 추가 사용자와 임시 테이블 공간을 추가할 수 있습니다.

주: 카탈로그 테이블 공간 SYSCATSPACE를 삭제해야 하며 항상 적어도 하나의 시스템 임시 테이블 공간이 있어야 합니다. 또한 테이블 공간을 작성한 후에 페이지 크기나 테이블 공간의 확장 크기를 변경할 수 없습니다.

이 절에서는 다음과 같이 테이블 공간을 변경하는 방법에 대해 설명합니다.

- 『DMS 테이블 공간에 컨테이너 추가』
- 211 페이지의 『DMS 테이블 공간에서 컨테이너 수정』
- 213 페이지의 『사용자 테이블 공간 삭제』
- 214 페이지의 『시스템 임시 테이블 공간 삭제』.

테이블 공간에 관한 설계 정보는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.

DMS 테이블 공간에 컨테이너 추가

테이블 공간에 하나 이상의 컨테이너를 추가하여 DMS 테이블 공간(MANAGED BY DATABASE)으로 작성된 테이블 공간의 크기를 증가시킬 수 있습니다.

테이블 공간의 내용이 모든 컨테이너를 통해 다시 조정됩니다. 재조정하는 동안에도 테이블 공간에 대한 액세스는 제한되지 않습니다. 둘 이상의 컨테이너를 추가해야 할 경우에는, 컨테이너를 동시에 추가해야 합니다.

제어 센터를 사용하여 DMS 테이블 공간에 컨테이너를 추가하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 컨테이너를 추가하려는 테이블 공간을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 추가를 클릭하고, 정보를 완료하고 확인을 클릭하십시오.
4. 테이블 공간이 파티션된 데이터베이스 환경인 경우, 테이블 공간의 성능 매개변수를 변경하려는 경우 고급을 클릭하십시오.
5. 확인을 클릭하십시오.

명령행을 사용하여 DMS 테이블 공간에 컨테이너를 추가하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
  ADD (DEVICE '<path>' <size>)
```

다음 예에서는 UNIX 기반 시스템에서 테이블 공간에 두 개의 새 장치 컨테이너 (각각 10 000 페이지씩)를 추가하는 방법을 설명합니다.

```
ALTER TABLESPACE RESOURCE
  ADD (DEVICE '/dev/rhd9' 10000,
       DEVICE '/dev/rhd10' 10000)
```

ALTER TABLESPACE문을 사용하여, 성능에 영향을 줄 수 있는 테이블 공간의 기타 등록 정보를 변경할 수 있습니다. 자세한 정보는 *관리 안내서: 성능의 『조회 최적화에서의 테이블 공간 영향』*을 참조하십시오.

DMS 테이블 공간에서 컨테이너 수정

테이블 공간과 연관된 하나 이상의 컨테이너를 확장하거나 하나 이상의 컨테이너의 크기를 재조정하여 DMS 테이블 공간(MANAGED BY DATABASE절로 작성된 테이블 공간)에서 컨테이너의 크기를 증가시킬 수 있습니다.

명령행을 사용하여 DMS 테이블 공간에서 하나 이상의 컨테이너의 크기를 재조정하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
  RESIZE (DEVICE '<path>' <size>)
```

다음 예는 UNIX 기반 시스템의 테이블 공간에서 두 개의 장치 컨테이너(1,000 페이지가 이미 있는)를 증가시킬 수 있는 방법을 설명합니다.

```
ALTER TABLESPACE HISTORY
  RESIZE (DEVICE '/dev/rhd7' 2000,
         DEVICE '/dev/rhd8' 2000)
```

이 조치 다음에, 두 개의 장치는 1,000 페이지에서 2,000 페이지로 증가합니다. 새로운 컨테이너를 추가하는 것과 유사하게, 테이블 공간의 내용이 모든 컨테이너를 통해 다시 조정됩니다. 재조정하는 동안에도 테이블 공간에 대한 액세스는 제한되지 않습니다.

명령행을 사용하여 DMS 테이블 공간에서 하나 이상의 컨테이너를 확장하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
    EXTEND (DEVICE '<path>' <size>)
```

다음 예는 UNIX 기반 시스템의 테이블 공간에서 두 개의 장치 컨테이너(1,000 페이지가 이미 있는)를 증가시킬 수 있는 방법을 설명합니다.

```
ALTER TABLESPACE HISTORY
    EXTEND (DEVICE '/dev/rhd11' 1000,
           DEVICE '/dev/rhd12' 1000)
```

이 조치 다음에, 두 개의 장치는 1,000 페이지에서 2,000 페이지로 증가합니다. 새로운 컨테이너를 추가하는 것과 유사하게, 테이블 공간의 내용이 모든 컨테이너를 통해 다시 조정됩니다. 재조정하는 동안에도 테이블 공간에 대한 액세스는 제한되지 않습니다.

주: 컨테이너의 크기를 줄일 수 없습니다.

ALTER TABLESPACE문을 사용하여, 성능에 영향을 줄 수 있는 테이블 공간의 기타 등록 정보를 변경할 수 있습니다. 자세한 정보는 *관리 안내서: 성능의 『조치 최적화에서의 테이블 공간 영향』*을 참조하십시오.

테이블 공간 이름 바꾸기

테이블 공간 내에서 각 오브젝트와 관련되지 않고 새로운 이름을 기존 테이블 공간에 부여할 수 있습니다. 테이블 공간의 이름을 바꿀 때, 해당 테이블 공간을 참조하는 모든 카탈로그 레코드가 변경됩니다.

SYSCATSPACE 테이블 공간의 이름을 바꿀 수 없습니다.

『롤 포워드 보류』 또는 『롤 포워드 진행』 상태인 테이블 공간의 이름을 바꿀 수 없습니다.

백업된 이후에 이름이 바뀐 테이블 공간을 복원할 때, RESTORE DATABASE 명령에서 새로운 테이블 공간 이름을 사용해야 합니다. 이전 테이블 공간 이름을 사용하는 경우, 찾지 못합니다. 유사하게 ROLLFORWARD DATABASE 명령으로 테이블 공간을 롤 포워드하는 경우, 새로운 이름을 사용해야 합니다. 이전 테이블 공간 이름을 사용하는 경우, 찾지 못합니다.

사용자 테이블 공간 삭제

사용자 테이블 공간을 삭제하는 경우, 해당 테이블 공간의 모든 데이터를 삭제하고, 컨테이너를 해제하고, 카탈로그 항목을 제거하십시오. 그러면, 테이블 공간의 모든 정의된 오브젝트는 삭제되거나 유효하지 않은 것으로 표시됩니다.

테이블 공간을 삭제하여 빈 테이블 공간에서 컨테이너를 다시 사용할 수 있지만, 컨테이너를 다시 사용하기 전에 **DROP TABLESPACE** 명령을 쿼리게 표시해야 합니다.

사용자는 단일 사용자 테이블 공간에서 색인 및 LOB 데이터를 비롯한 모든 테이블 데이터가 들어 있는 사용자 테이블 공간을 삭제할 수 있습니다. 여러 개의 테이블 공간에 분산되어 있는 사용자 테이블 공간을 삭제할 수도 있습니다. 즉, 테이블 데이터를 테이블 공간 하나에 보관하고 색인은 또다른 테이블 공간에 보관하여 모든 LOB를 세 번째 테이블 공간에 보관할 수 있습니다. 테이블 데이터가 있는 테이블 공간이 먼저 삭제되는 한 각 테이블 공간을 개별적으로 삭제할 수 있습니다. 또는, 단일 명령문에서 세 개의 테이블 공간을 전부 동시에 삭제할 수 있습니다. 테이블이 들어 있는 모든 분산된 테이블 공간은 이 단일 명령문의 일부가 되지 않으면, 삭제 요청이 실패합니다. 분산된 테이블 데이터가 들어 있는 테이블 공간 삭제 방법에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 사용자 테이블 공간을 삭제하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 테이블 공간을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. 확정 상자를 클릭하고 확인을 클릭하십시오.

명령행을 사용하여 사용자 테이블을 삭제하십시오.

```
DROP TABLESPACE <name>
```

다음 SQL문은 ACCOUNTING 테이블 공간을 삭제합니다.

```
DROP TABLESPACE ACCOUNTING
```

시스템 임시 테이블 공간 삭제

또다른 시스템 임시 테이블 공간을 우선 작성하지 않고 시스템 임시 테이블 공간을 삭제할 수 없습니다. 왜냐하면, 데이터베이스에는 적어도 하나의 시스템 임시 테이블 공간이 항상 있어야 하기 때문입니다. 예를 들어, SMS 임시 테이블 공간에 컨테이너를 추가하려면, 먼저 새 시스템 임시 테이블 공간을 추가한 후 이전 시스템 임시 테이블 공간을 삭제해야 합니다.

제어 센터를 사용하여 시스템 테이블 공간을 삭제하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 기타 시스템 임시 테이블 공간에서, 테이블 공간 아이콘을 마우스의 오른쪽 버튼으로 선택하고 작성 → 마법사를 사용한 테이블 공간을 선택하십시오. 그렇지 않으면 단계 4로 건너뛰십시오.
3. 필요한 경우 새로운 시스템 임시 테이블 공간을 작성하려면 마법사에 있는 단계에 따르십시오.
4. 창의 오른쪽에서(목차 패널) 테이블 공간의 목록을 표시하려면 테이블 공간 폴더를 다시 클릭하십시오.
5. 삭제하려는 시스템 임시 테이블 공간을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 클릭하십시오.
6. 확정 상자를 클릭하고 확인을 클릭하십시오.

오직 하나의 시스템 임시 테이블 공간을 가진 경우, 삭제하기 전에 또다른 것을 작성해야 합니다. 이것은 다음을 입력하여 명령행을 사용하여 수행될 수 있습니다.

```
CREATE SYSTEM TEMPORARY TABLESPACE <name>  
MANAGED BY SYSTEM USING ('<device>')
```

명령행을 사용하여 시스템 테이블 공간을 삭제하십시오.

```
DROP TABLESPACE <name>
```

다음 SQL문은 TEMPSPACE2라고 하는 새 시스템 임시 테이블 공간을 작성합니다.

```
CREATE SYSTEM TEMPORARY TABLESPACE TEMPSPACE2  
MANAGED BY SYSTEM USING ('d')
```

일단 TEMPSPACE2가 작성되면, 원래의 시스템 임시 테이블 공간인 TEMPSPACE1을 다음 명령으로 삭제할 수 있습니다.

DROP TABLESPACE TEMPSPACE1

테이블 공간을 삭제하여 빈 테이블 공간에서 컨테이너를 다시 사용할 수 있지만, 컨테이너를 다시 사용하기 전에 **DROP TABLESPACE** 명령을 쿼게해야 합니다.

임시 테이블 공간 삭제

해당 테이블 공간에 정의되어 있고, 현재 선언된 테이블 공간이 없는 경우 단지 사용자 임시 테이블 공간을 삭제할 수 있습니다. 테이블 공간을 삭제할 때, 테이블 공간에 있는 모든 선언된 임시 테이블을 삭제하려는 시도는 수행되지 않습니다.

주: 선언된 임시 테이블은 그것을 선언한 응용프로그램이 데이터베이스에서 연결해 제되면 내재적으로 삭제됩니다.

스키마 삭제

스키마를 삭제하기 전에, 해당 스키마에 있는 모든 오브젝트가 삭제되거나 다른 스키마로 이동되어야 합니다. **DROP**문을 시도할 때에는 스키마 이름이 카탈로그에 있어야 하고, 그렇지 않으면 오류가 리턴됩니다.

제어 센터를 사용하여 스키마를 삭제하십시오.

1. 스키마 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 스키마를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. **확정** 상자를 클릭하고 **확인**을 클릭하십시오.

명령행을 사용하여 스키마를 삭제하십시오.

```
DROP SCHEMA <name>
```

다음 예에서는 스키마 "joeschma"가 삭제됩니다.

```
DROP SCHEMA joeschma RESTRICT
```

RESTRICT 키워드는 데이터베이스에서 삭제되는 스키마의 경우 어떠한 오브젝트도 지정된 스키마에 정의될 수 없다는 규칙을 실시합니다.

구조 및 내용에서 테이블 수정

테이블 구조 및 내용 수정에 필요한 TASK에는 다음이 포함됩니다.

- 『기존 테이블에 컬럼 추가』
- 217 페이지의 『컬럼 정의 수정』
- 218 페이지의 『제한조건 변경』
- 218 페이지의 『제한조건 추가』
- 221 페이지의 『제한조건 삭제』
- 228 페이지의 『휘발성 테이블 선언』
- 228 페이지의 『파티션 키 변경』
- 229 페이지의 『테이블 속성 변경』
- 231 페이지의 『요약 테이블의 데이터 새로 고침』

테이블용 트리거를 변경할 수 없습니다. 더 이상 적합하지 않은 모든 트리거를 삭제해야 하고(235 페이지의 『트리거 삭제』 참조), 대체 트리거를 추가해야 합니다(158 페이지의 『트리거 작성』 참조).

기존 테이블에 컬럼 추가

컬럼 정의에는 컬럼 이름, 데이터 유형 및 필요한 제한조건이 포함됩니다.

새 컬럼이 기존 테이블에 추가될 때에는, 시스템 카탈로그의 테이블 설명만이 수정되므로, 테이블로의 액세스 시간에 즉각 영향을 주지 않습니다. 기존의 레코드는 UPDATE문을 사용하여 수정될 때까지 물리적으로 변경되지 않았습니다. 테이블에서 기존의 행을 검색할 때, 새로운 컬럼이 정의된 방법에 따라 널(NULL) 또는 기본값이 새로운 컬럼에 제공됩니다. 테이블이 작성된 후 추가되는 컬럼은 NOT NULL로 정의될 수 없습니다. 이는 NOT NULL WITH DEFAULT 또는 널(NULL) 입력 가능으로 정의되어야 합니다.

제어 센터를 사용하여 기존 테이블에 컬럼을 추가하려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 컬럼을 추가하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 컬럼 페이지를 점검하고, 컬럼에 대한 정보를 완료하고 **확인**을 클릭하십시오.

명령행을 사용하여 기존 테이블에 컬럼을 추가하려면 다음을 수행하십시오.

```
ALTER TABLE <table_name>  
ADD <column_name> <data_type> <null_attribute>
```

컬럼은 SQL문으로 추가될 수 있습니다. 다음 명령문은 ALTER TABLE문을 사용하여 세 개의 컬럼을 EMPLOYEE 테이블에 추가합니다.

```
ALTER TABLE EMPLOYEE  
ADD MIDINIT CHAR(1) NOT NULL WITH DEFAULT  
ADD HIREDATE DATE  
ADD WORKDEPT CHAR(3)
```

컬럼 정의 수정

기존 VARCHAR 컬럼의 길이를 증가시켜 컬럼의 특성을 수정할 수 있습니다. 문자 수는 사용되는 페이지 크기에 맞는 값까지 증가될 수 있습니다.

제어 센터를 사용하여 기존 테이블의 컬럼을 수정하려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 오른쪽 분할 창에 있는 테이블 목록에서, 컬럼을 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 **변경**을 선택하십시오.
3. 컬럼 페이지를 점검하고, 컬럼을 선택하고, **변경**을 클릭하십시오.
4. 컬럼에 새 바이트 수를 길이에 입력하고 **확인**을 클릭하십시오.

명령행을 사용하여 기존 테이블의 컬럼을 수정하려면 다음을 입력하십시오.

```
ALTER TABLE ALTER COLUMN  
<column_name> <modification_type>
```

예를 들어, 컬럼을 최대 4000자까지 증가시키려면, 다음과 유사한 것을 사용하십시오.

```
ALTER TABLE ALTER COLUMN
COLNAM1 SET DATA TYPE VARCHAR(4000)
```

입력된 테이블의 컬럼은 변경할 수 없습니다. 그러나, 아직 범위가 정의되지 않은 기존 참조 유형 컬럼에 범위를 추가할 수는 있습니다. 예를 들면, 다음과 같습니다.

```
ALTER TABLE ALTER COLUMN
COLNAMT1 ADD SCOPE TYPTAB1
```

ALTER TABLE문에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

제한조건 변경

삭제한 다음 제한조건을 변경할 수 있으며 그 자리에 새로운 제한조건을 추가할 수 있습니다. 자세한 정보는 다음을 참조하십시오.

- 『제한조건 추가』
- 221 페이지의 『제한조건 삭제』

제한조건에 대해서는 145 페이지의 『제한조건 정의』에서 자세한 내용을 참조하십시오.

제한조건 추가

ALTER TABLE문을 사용하여 제한조건을 추가하십시오. 이 명령문에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

제한조건에 대해서는 145 페이지의 『제한조건 정의』에서 자세한 내용을 참조하십시오.

고유 제한조건 추가: 고유 제한조건은 기존의 테이블에 추가될 수 있습니다. 제한조건 이름은 ALTER TABLE문에 지정된 다른 제한조건과 동일할 수 없으며, 테이블 안에서 고유해야 합니다(여기에는 정의된 참조 무결성 제한조건도 포함합니다). 명령문이 수행되기 전에 기존의 데이터가 새로운 조건에 맞는지 점검합니다.

다음 SQL문은 테이블에서 사원을 식별하는 새로운 방법을 나타내는 고유 제한조건을 EMPLOYEE 테이블에 추가합니다.

```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT NEWID UNIQUE(EMPNO,HIREDATE)
```

기본 키 및 외부 키의 추가: 대형 테이블에 제한조건을 추가하려면, 테이블을 점검 보류 상태로 놓고, 제한조건을 추가한 후, 테이블에서 위반 행의 통합 목록을 점검하는 것이 훨씬 더 효율적입니다. SET INTEGRITY문을 사용하여 점검 보류 상태를 명시적으로 설정하십시오. 테이블이 상위 테이블인 경우에는 모든 종속 테이블 및 하위 테이블에 대해 점검 보류가 내재적으로 설정됩니다.

제어 센터를 사용하여 기본 키를 추가하려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 기본 키 페이지에서, 기본 키로서 하나 이상의 컬럼을 선택하고, 화살표를 클릭하여 이동하십시오.
4. 선택적: 기본 키의 제한조건 이름을 입력하십시오.
5. 확인을 클릭하십시오.

명령행을 사용하여 기본 키를 추가하십시오.

```
ALTER TABLE <name>  
  ADD CONSTRAINT <column_name>  
  PRIMARY KEY <column_name>
```

외부 키가 테이블에 추가되면, 다음과 같은 명령문이 들어 있는 패키지와 캐쉬된 동적 SQL은 유효하지 않음으로 표시됩니다.

- 외부 키가 들어 있는 테이블을 삽입하고 갱신하는 명령문
- 상위 테이블을 갱신하거나 삭제하는 명령문

244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 외부 키를 추가하려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 외부 키 페이지에서, 추가를 클릭하십시오.
4. 외부 키 추가 창에서, 상위 테이블 정보를 지정하십시오.
5. 하나 이상의 컬럼이 외부 키가 되도록 선택하고, 화살표를 클릭하여 이동하십시오.
6. 상위 테이블의 행이 삭제되거나 갱신될 때 종속 테이블에서 수행할 조치를 지정하십시오. 또한 외부 키에 대한 제한조건 이름을 추가할 수 있습니다.
7. 확인을 클릭하십시오.

명령행을 사용하여 외부 키를 추가하십시오.

```
ALTER TABLE <name>
  ADD CONSTRAINT <column_name>
  FOREIGN KEY <column_name>
  ON DELETE <action_type>
  ON UPDATE <action_type>
```

다음 예에서는 기본 키와 외부 키를 테이블에 추가하는 ALTER TABLE문을 보여줍니다.

```
ALTER TABLE PROJECT
  ADD CONSTRAINT PROJECT_KEY
  PRIMARY KEY (PROJNO)
ALTER TABLE EMP_ACT
  ADD CONSTRAINT ACTIVITY_KEY
  PRIMARY KEY (EMPNO, PROJNO, ACTNO)
  ADD CONSTRAINT ACT_EMP_REF
  FOREIGN KEY (EMPNO)
  REFERENCES EMPLOYEE
  ON DELETE RESTRICT
  ADD CONSTRAINT ACT_PROJ_REF
  FOREIGN KEY (PROJNO)
  REFERENCES PROJECT
  ON DELETE CASCADE
```

테이블 점검 제한조건의 추가: 점검 제한조건은 ALTER TABLE문을 사용하여 기존의 테이블에 추가될 수 있습니다. 제한조건 이름은 ALTER TABLE 명령문에 지정된 다른 제한조건과 동일할 수 없으며, 테이블 안에서 고유해야 합니다.(여

기에는 정의된 참조 무결성 제한조건의 이름을 포함합니다.) 명령문이 수행되기 전에 기존의 데이터가 새로운 조건에 맞는지 점검합니다.

대형 테이블에 제한조건을 추가하려면, 테이블을 점검 보류 상태로 놓고, 제한조건을 추가한 후, 테이블에서 위반 행의 통합 목록을 점검하는 것이 훨씬 더 효율적입니다. SET INTEGRITY문을 사용하여 점검 보류 상태를 명시적으로 설정하십시오. 테이블이 상위 테이블인 경우에는 모든 종속 테이블 및 하위 테이블에 대해 점검 보류가 내재적으로 설정됩니다.

테이블 점검 제한조건이 추가되면, 테이블을 삽입하거나 갱신하는 패키지 및 캐쉬화된 동적 SQL이 유효하지 않음으로 표시됩니다. 244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 테이블 점검 제한조건을 추가하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 점검 제한조건 페이지에서, 추가를 클릭하십시오.
4. 점검 제한조건 추가에서, 정보를 완료하고 확인을 클릭하십시오.
5. 점검 제한조건 페이지에서, 확인을 클릭하십시오.

명령행을 사용하여 테이블 점검 제한조건을 추가하십시오.

```
ALTER TABLE <name>  
ADD CONSTRAINT <name> (<constraint>)
```

다음 SQL문은 급료와 커미션을 합해 \$25,000를 넘어야 한다는 제한조건을 EMPLOYEE 테이블에 추가합니다.

```
ALTER TABLE EMPLOYEE  
ADD CONSTRAINT REVENUE CHECK (SALARY + COMM > 25000)
```

제한조건 삭제

ALTER TABLE문을 사용하여 제한조건을 삭제합니다. 이 명령문에 관한 자세한 정보는 SQL 참조서 매뉴얼을 참조하십시오.

제한조건에 대해서는 145 페이지의 『제한조건 정의』에서 자세한 내용을 참조하십시오.

고유 제한조건 삭제: ALTER TABLE문을 사용하여 고유 제한조건을 명시적으로 삭제할 수 있습니다. 테이블의 모든 고유 제한조건의 이름은 SYSCAT.INDEXES 시스템 카탈로그 뷰에 있습니다.

다음 SQL문은 EMPLOYEE 테이블에서 고유 제한조건 NEWID를 삭제합니다.

```
ALTER TABLE EMPLOYEE
  DROP UNIQUE NEWID
```

이 고유 제한조건을 삭제하면, 제한조건에 사용된 모든 패키지 또는 캐쉬된 동적 SQL이 무효로 됩니다.

기본 키 및 외부 키 삭제: 제어 센터를 사용하여 기본 키를 삭제하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 기본 키 페이지에서, 오른쪽에서 삭제할 기본 키를 선택하고, 화살표를 클릭하여 왼쪽에 있는 사용가능한 컬럼 상자로 이동하십시오.
4. 확인을 클릭하십시오.

명령행을 사용하여 기본 키를 삭제하십시오.

```
ALTER TABLE <name>
  DROP PRIMARY KEY
```

외부 키 제한조건이 삭제되면, 다음과 같은 사항이 들어 있는 패키지와 캐쉬된 동적 SQL문이 유효하지 않으므로 표시됩니다.

- 외부 키가 들어 있는 테이블을 삽입하고 갱신하는 명령문
- 상위 테이블을 갱신하거나 삭제하는 명령문

244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 외부 키를 삭제하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. **외부 키** 페이지에서, **추가**를 클릭하십시오.
4. 오른쪽에서 삭제할 외부 키를 선택하고, 화살표를 클릭하여 왼쪽에 있는 **사용 가능한 컬럼** 상자로 이동하십시오.
5. **외부 키** 페이지에서, **확인**을 클릭하십시오.

명령행을 사용하여 외부 키를 삭제하십시오.

```
ALTER TABLE <name>  
DROP FOREIGN KEY <foreign_key_name>
```

다음 예에서는 ALTER TABLE문의 DROP PRIMARY KEY 및 DROP FOREIGN KEY절을 사용하여 테이블에서 기본 키와 외부 키를 삭제합니다.

```
ALTER TABLE EMP_ACT  
DROP PRIMARY KEY  
DROP FOREIGN KEY ACT_EMP_REF  
DROP FOREIGN KEY ACT_PROJ_REF  
ALTER TABLE PROJECT  
DROP PRIMARY KEY
```

ALTER TABLE문에 대해서는 *SQL 참조서 매뉴얼*을 참조하십시오.

테이블 점검 제한조건 삭제: ALTER TABLE문을 사용하여 테이블 점검 제한조건을 명시적으로 삭제 또는 변경하거나, DROP TABLE문의 결과로서 이를 내재적으로 삭제할 수 있습니다.

테이블 점검 제한조건을 삭제하면, 테이블에 INSERT 또는 UPDATE 종속이 있는 모든 패키지와 캐쉬된 SQL문이 무효화됩니다.(244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.) 테이블에 대한 모든 점검 제한조건은 SYSCAT.CHECKS 카탈로그 뷰에 있습니다.시스템이 생성한 이름을 갖는 테이블 점검 제한조건을 삭제하기 전에, SYSCAT.CHECKS 카탈로그 뷰에서 이름을 찾으십시오.

제어 센터를 사용하여 테이블 점검 제한조건을 삭제하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 점검 제한조건 페이지에서, 삭제할 점검 제한조건을 선택하고, **제거**를 클릭하고, **확인**을 클릭하십시오.

명령행을 사용하여 테이블 점검 제한조건을 삭제하십시오.

```
ALTER TABLE <table_name>
  DROP CHECK <check_constraint_name>
```

다음 SQL문은 EMPLOYEE 테이블에서 테이블 점검 제한조건 REVENUE를 삭제합니다.

```
ALTER TABLE EMPLOYEE
  DROP CHECK REVENUE
```

생성된 컬럼을 기존 테이블에 정의

생성된 컬럼은 저장된 값이 삽입 또는 갱신 조작을 통해 지정되기 보다는 표현식을 사용하여 계산되는 기본 테이블에 정의됩니다. 생성된 컬럼은 테이블이 작성될 때 작성되거나 기존 테이블에 대한 수정으로서 작성됩니다.

다음 단계를 수행하여 생성된 컬럼을 정의하십시오.

1. 테이블을 점검 보류 상태로 지정하십시오.

```
SET INTEGRITY FOR t1 OFF
```

2. 하나 이상의 생성된 컬럼을 추가하려면 테이블을 변경하십시오.

```
ALTER TABLE t1 ADD COLUMN c3 DOUBLE GENERATED ALWAYS AS (c1 + c2),
  ADD COLUMN c4 GENERATED ALWAYS AS
  (CASE WHEN c1 > c3 THEN 1 ELSE NULL END)
```

3. 여기에서 테이블에 대해 수행되는 작업에 근거하여 이 작업을 완료하는 여러가지 방법이 있습니다.

- 테이블은 매우 크며 작업을 완료할 로그 공간이 충분한지 확인하지 못합니다. 데이터를 로드한 후에 그렇지만 무결성 점검을 다시 on으로 설정하기 전에, 다음을 수행해야 합니다.

```
COMMIT
```

그런 다음 db2gncol 유틸리티를 사용하여 생성된 컬럼을 설정할 필요가 있습니다. 이 유틸리티는 bin 하위 디렉토리의 sqllib 디렉토리 아래에 위치합니다. 다음과 같이 유틸리티를 사용합니다.

```
db2gncol -d <dbname> -s <schema> -t <table_name>
-c <commitcount>
```

dbname은 테이블이 위치한 데이터베이스의 별명 이름을 지정합니다. schema는 테이블의 스키마 이름을 지정하며 대소문자가 구별됩니다. table_name은 표현식이 생성하는 컬럼의 새 값이 계산되는 테이블을 지정합니다. schema 및 table_name 둘다 대소문자 구별됩니다. commitcount는 로그를 제거하기 위해 각 내부 확장 간에 처리하는 행 수입니다. 이 매개변수는 컬럼 값의 생성을 수행하는 데 필요한 로그 공간의 크기에 영향을 미칩니다.

위의 예에 표시되지 않는 두 개의 선택적 매개변수가 있습니다. 이것은 사용자와 암호를 식별하는 -u <username> 및 -p <password>입니다. 사용자는 SYSADM 또는 DBADM 권한을 가져야 합니다. 식별되는 사용자와 암호가 없는 경우, 현재 사용자 ID가 사용됩니다.

이 유틸리티에서 도움말 정보를 원하는 경우, 다음을 입력하십시오.

```
db2gncol -h
```

도움말 매개변수가 사용되면, 다른 모든 매개변수는 무시됩니다.

테이블이 잠금 보류 상태에 있을지라도 테이블은 전체 프로세스에 대해 잠깁니다. 잠금의 이유는 잠금 보류 상태에 있는 테이블에 액세스할 수 있는 다른 유틸리티가 있기 때문입니다. 잠금은 다른 유틸리티와 충돌하는 것을 방지합니다.

- 생성된 컬럼을 갱신할 로그 공간이 SET INTEGRITY에 대해 충분할 것입니다. 정상적인 경우에 그렇습니다. 데이터를 로드한 후에 다음을 사용하여 생성된 컬럼에 대한 값을 재계산하고, 재지정하십시오.

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
FORCE GENERATED
```

주: 예외 테이블을 여기에서 사용할 수 있습니다.

- 테이블은 매우 크며 작업을 완료할 로그 공간이 충분한지 확인하지 못하면, 위에 표시된 첫번째 메소드를 선택하지 않습니다. 데이터를 로드한 후에 그렇지만 무결성 점검을 다시 on으로 설정하기 전에, 다음을 수행해야 합니다.

- a. 테이블에서 독점 잠금을 얻으십시오. 이것은 미확약 읽기(UR) 트랜잭션을 제외하고 모두 테이블을 액세스하지 못하게 합니다.

```
LOCK TABLE t1
```

- b. 점검되지 않은 데이터로 테이블을 온라인 상태로 이동시키십시오.

```
SET INTEGRITY FOR t1 ALL IMMEDIATE UNCHECKED
```

- c. 생성된 컬럼을 간헐적인 확약 및 술어를 사용해서 갱신하여 로그가 가득 차는 것을 피하십시오.

```
UPDATE t1 SET (c3, c4) = (DEFAULT, DEFAULT) WHERE <predicate>
```

- d. 테이블을 온라인 상태로 가져와 무결성을 점검하십시오.

```
SET INTEGRITY FOR t1 OFF
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

- e. 확약 명령문을 사용하여 트랜잭션을 완료하여 테이블을 잠금해제하십시오.

```
COMMIT
```

- 테이블이 초기에 로그되지 않은 옵션으로 작성되었습니다. 이러한 방식으로, 테이블의 로깅은 생성된 컬럼 값으로 작업하는 중에 일반 포함 및 위험으로 설정 해제됩니다.

- a. 초기에 로그되지 않은 옵션을 활성화하십시오.

```
ALTER TABLE t1 ACTIVATE NOT LOGGED INITIALLY
```

- b. 값을 생성하십시오.

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED FORCE GENERATION
```

- c. 트랜잭션을 확약하여 처음에 로그되지 않는 옵션을 설정해제하십시오.

```
COMMIT
```

생성된 컬럼의 값은 표현식을 등식 점검 제한조건인 것처럼 적용하여 간단히 점검할 수 있습니다.

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

예를 들어 LOAD를 사용하여 생성된 컬럼에 값이 위치하며 값이, 생성된 표현식과 일치함을 아는 경우, 값을 점검하거나 지정하지 않고 점검 보류 상태로부터 테이블을 가져올 수 있습니다.

```
SET INTEGRITY FOR t1 GENERATED COLUMN IMMEDIATE UNCHECKED
```

생성된 컬럼은 동일한 비교가 정의된 데이터 유형에서만 정의될 수 있습니다. 생성된 컬럼에서 제외된 데이터 유형은 다음과 같습니다. 구조화된 유형, LOB, CLOB, DBCLOB, LONG VARCHAR, LONG VARGRAPHIC 및 제외된 데이터 유형과 동일한 것을 사용하여 정의된 사용자 정의 유형.

생성된 컬럼은 제한조건, 고유 색인, 참조 제한조건, 기본 키 및 전역 임시 테이블에서 사용될 수 없습니다. LIKE 및 구체화된 뷰로 작성된 테이블은 생성된 컬럼 등록 정보를 계승하지 않습니다.

생성된 컬럼은 키워드 DEFAULT 없이 삽입되거나 갱신될 수 없습니다. 삽입할 때, DEFAULT를 사용하면 컬럼 목록에 컬럼을 열거할 필요가 없습니다. 대신, 생성된 컬럼은 값 목록에 있는 DEFAULT로 설정될 수 있습니다. 갱신할 때, DEFAULT는 점검하지 않고도 SET INTEGRITY에 의해 온라인으로 위치한 생성된 컬럼의 재계산을 가능하게 합니다.

트리거의 처리 순서는 BEFORE 트리거가 머리글(갱신 이전) 또는 본문에서 생성된 컬럼을 참조할 수 없도록 요구합니다. 처리 순서에서, 생성된 컬럼은 BEFORE 트리거 이후에 처리됩니다.

db2look 유틸리티는 생성된 컬럼에 의해 생성된 점검 제한조건을 모릅니다.

복제 사용 시, 목표 테이블은 맵핑에서 생성된 컬럼을 사용하지 말아야 합니다. 복제할 때 두가지 선택사항이 있습니다.

- 목표 테이블은 일반 컬럼으로서 생성된 컬럼을 포함해야 합니다.
- 목표 테이블은 맵핑에서 생성된 컬럼을 생략해야 합니다.

생성된 컬럼으로 작업할 때 몇가지 제한사항이 있습니다.

- 생성된 컬럼은 서로 종속성을 갖지 말아야 합니다.
- 생성된 컬럼을 작성하기 위해 사용되는 표현식은 부속 조희를 포함하지 말아야 합니다. 이것은 SQL 데이터를 읽는 함수와 함께 표현식을 포함합니다.

- 생성된 컬럼에 어떠한 점검 제한조건도 허용되지 않습니다.
- 생성된 컬럼에 어떠한 고유 색인도 허용되지 않습니다. 이것은 고유 제한조건 및 기본 키를 포함합니다.

휘발성 테이블 선언

휘발성 테이블은 런타임 크기가 공백에서 대형까지 변환 수 있는 내용을 가진 테이블로 정의됩니다. 이 테이블의 휘발성 또는 극도의 변경 가능성은 RUNSTATS 로 수집된 통계 사용을 부정확하게 만듭니다. 통계는 특정 시점에서 수집되고 특정 시점만을 반영합니다. 휘발성 테이블을 사용하는 액세스 플랜을 생성하면, 이는 부정확하거나 성능이 떨어지는 결과를 냅니다. 예를 들어, 휘발성 테이블이 비어 있을 때 통계가 모아지면, 최적화 알고리즘은 색인 스캔보다는 테이블 스캔을 사용하여 휘발성 테이블 액세스를 선호하는 경향이 있습니다.

이를 방지하려면, ALTER TABLE문을 사용하여 테이블을 휘발성(volatile)으로 선언해야 합니다. 테이블을 휘발성으로 선언하면, 최적화 알고리즘은 테이블 스캔보다 색인 스캔 사용을 고려합니다. 선언된 휘발성 테이블을 사용하는 액세스 플랜은 해당 테이블에 대해 기존의 통계에 의존하지 않습니다.

제어 센터를 사용하여 테이블을 휘발성으로 선언하려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 테이블 페이지에서, 런타임 시에 기본 행 수 변동 선택란을 선택하고 확인을 클릭하십시오.

명령행을 사용하여 테이블을 휘발성으로 선언하려면, 다음을 수행하십시오.

```
ALTER TABLE <table_name>
VOLATILE CARDINALITY
```

테이블을 『휘발성』으로 선언하려면, 다음을 수행하십시오.

```
ALTER TABLE TABLENAME
VOLATILE CARDINALITY
```

파티션 키 변경

단일 파티션 노드 그룹에 있는 테이블의 파티션 키만 변경할 수 있습니다. 우선 기존의 파티션 키를 삭제한 후 또다른 것을 작성하십시오.

다음 SQL문은 MIXREC 테이블에서 파티션 키 MIX_INT를 삭제합니다.

```
ALTER TABLE MIXREC
  DROP PARTITIONING KEY
```

자세한 정보는 *SQL 참조서 매뉴얼*의 ALTER TABLE문을 참조하십시오.

다중 데이터베이스 파티션 노드 그룹의 테이블에 있는 파티션 키는 변경할 수 없습니다. 이를 삭제하려고 할 경우, 오류가 표시됩니다.

여러 데이터베이스 파티션 노드 그룹의 파티션 키를 변경하려면 다음 중 하나를 수행하십시오.

- 모든 데이터를 단일 파티션 노드 그룹으로 내보낸 후 위의 지시를 따릅니다.
- 모든 데이터를 내보내고, 테이블을 삭제하며, 파티션 키를 다시 정의한 다음, 모든 데이터를 가져옵니다.

이 방법은 대형 데이터베이스에는 사용 가능하지 않습니다. 그러므로, 대형 데이터베이스 설계를 구현하기 전에 적합한 파티션 키를 정의하는 것이 중요합니다.

테이블 속성 변경

데이터 캡처 옵션, 각 페이지의 사용 가능한 공백 백분율(PCTFREE), 잠금 크기 또는 추가 모드와 같은 테이블 속성을 변경할 경우가 있습니다.

테이블의 각 페이지에 남아 있는 사용 가능한 공간의 양은 PCTFREE를 통해 지정되며, 클러스터링 색인의 효과적인 사용을 위한 주요 고려사항입니다. 지정할 양은 기존 데이터 및 예상 장래 데이터의 특성에 좌우됩니다. PCTFREE는 LOAD 및 REORG로 고려되지만, 삽입, 갱신 및 가져오기 활동에 의해 무시됩니다.

PCTFREE를 더 큰 값에 설정하면 더 긴 기간 동안 클러스터를 유지보수하지만, 더 많은 디스크 공간이 필요합니다.

LOCKSIZE 매개변수를 사용하여 테이블이 액세스될 때 사용되는 잠금의 크기(세분성)를 지정할 수 있습니다. 기본적으로, 테이블이 작성될 때 하위 레벨 잠금이 정의됩니다. 테이블 레벨 잠금의 사용은 취득하거나 해제하는 데 필요한 잠금 수를 제한하여 조회의 성능을 향상시킬 수 있습니다.

APPEND ON을 지정하여, 전체 성능을 향상시킬 수 있습니다. 사용 가능한 공간에 대한 정보의 유지보수를 줄이고, 더 빠른 삽입을 허용합니다.

클러스터링 색인이 있는 테이블은 추가 모드를 설정하도록 변경할 수 없습니다. 마찬가지로, 클러스터링 색인은 추가 모드가 있는 테이블에서 작성될 수 없습니다.

요약 테이블 등록 정보 변경

일부 제한사항이 있는 경우, 요약 테이블을 일반 테이블로 변경하거나 일반 테이블을 요약 테이블로 변경할 수 있습니다. 다른 테이블 유형을 변경할 수 없으며, 오직 일반 및 요약 테이블만을 변경할 수 있습니다. 예를 들어, 복제된 요약 테이블을 일반 테이블로 변경할 수 없으며, 역으로도 변경할 수 없습니다.

일반 테이블이 요약 테이블로 변경되면, 테이블은 점검 보류 상태가 됩니다. 이러한 방식으로 변경할 때, 요약 테이블 정의의 fullselect는 원래 테이블 정의와 일치해야 합니다.

- 컬럼 수가 동일해야 합니다.
- 컬럼 이름 및 위치가 일치해야 합니다.
- 데이터 유형이 동일해야 합니다.

요약 테이블이 원래의 테이블에 정의된 경우, 요약 테이블로 변경될 수 없습니다. 원래 테이블에 트리거, 점검 제한조건, 참조 제한조건 또는 정의된 고유 색인이 있는 경우, 요약 테이블로 변경할 수 없습니다. 요약 테이블을 정의하기 위해 테이블 등록 정보를 변경하는 경우, 동일한 ALTER TABLE문에서 다른 방식으로 테이블을 변경하도록 허용되지 않습니다.

일반 테이블을 요약 테이블로 변경할 때, 요약 테이블 정의의 fullselect은 직접 또는 간접으로 뷰, 별명 또는 요약 테이블을 통해 원래의 테이블을 참조할 수 없습니다.

요약 테이블을 일반 테이블로 변경하려면, 다음을 사용하십시오.

```
ALTER TABLE sumtable
SET SUMMARY AS DEFINITION ONLY
```

일반 테이블을 요약 테이블로 변경하려면, 다음을 사용하십시오.

```
ALTER TABLE regtable
SET SUMMARY AS <fullselect>
```

일반 테이블을 요약 테이블로 변경할 때 fullselect에 대한 제한사항은 CREATE SUMMARY TABLE문을 사용하여 요약 테이블을 작성할 때의 제한사항과 매우 흡사합니다.

SQL 참조서에서 CREATE SUMMARY TABLE문에 대한 자세한 내용을 참조하십시오.

요약 테이블의 데이터 새로 고침

REFRESH TABLE문을 사용하여 하나 이상의 요약 테이블의 데이터를 새로 고칠 수 있습니다. 응용프로그램에 해당 명령문을 삽입하거나 동적으로 발행할 수 있습니다. 이 명령문을 사용하려면 새로 고칠 테이블에 대해 SYSADM 또는 DBADM 권한 또는 CONTROL 특권을 가지고 있어야 합니다.

다음 예에서는 요약 테이블에서 데이터를 새로 고치는 방법을 보여줍니다.

```
REFRESH TABLE SUMTAB1
```

REFRESH TABLE문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

사용자 정의 구조화 유형 변경

구조화 유형을 작성한 후, 해당 구조화 유형과 연관된 속성을 추가하거나 삭제해야 하는 경우가 있습니다. ALTER TYPE(구조화)문을 사용하여 이 작업을 수행할 수 있습니다. 응용프로그램 개발 안내서에서 구조화된 유형에 필요한 자세한 내용을 참조하십시오.

입력된 테이블의 행 삭제 및 갱신

검색되거나 배치된 DELETE문을 사용하여 입력된 테이블에서 행을 삭제할 수 있습니다. 검색되거나 배치된 UPDATE문을 사용하여 입력된 테이블에서 행을 갱신할 수 있습니다. 응용프로그램 개발 안내서에서 입력된 테이블에 필요한 자세한 내용을 참조하십시오.

기존 테이블 이름 바꾸기

사용자는 스키마 내에서 기존의 테이블에 새 이름을 부여할 수 있으며, 원래의 테이블에서 작성된 권한 부여와 색인을 유지보수할 수 있습니다.

테이블을 식별하는 별명으로 기존 테이블의 이름을 변경할 수 있습니다. 이름 바꾸기하려는 기존의 테이블은 카탈로그 테이블, 요약 테이블, 입력된 테이블 또는 테이블 또는 별명이 아닌 오브젝트의 이름을 사용할 수 없습니다.

다음에서는 기존의 테이블이 참조되지 않습니다.

- 뷰
- 트리거
- 참조 제한조건
- 요약 테이블
- 기존 참조 컬럼의 범위

또한, 테이블 내에 점검 제한조건이 있어서는 안되며, 식별 컬럼 이외에 생성된 컬럼이 있어서는 안됩니다. 원래의 테이블에 종속된 모든 패키지 또는 캐쉬된 동적 SQL문은 무효로 됩니다. 마지막으로, 원래의 테이블을 언급하는 모든 별명은 수정되지 않습니다.

해당 시스템 카탈로그 테이블을 점검하여 이름 바꾸기중인 테이블이 이들 제한사항에 의해 영향받지 않도록 해야 합니다.

패키지가 방금 이름 바뀌기된 테이블을 언급할 경우에는 다시 바인드되어야 합니다. 다음과 같은 경우 패키지는 내재적으로 다시 바인드될 수 있습니다.

- 다른 테이블이 원래의 테이블 이름을 사용하여 이름 바뀌기될 경우
- 별명이나 뷰가 원래의 테이블 이름을 사용하여 작성될 경우

내재적 또는 명시적 리바인드가 수행되기 전에, 이 두 선택항목 중 하나가 완료되어야 합니다. 선택이 이루어지지 않을 경우, 리바인드는 실패하게 됩니다.

제어 센터를 사용하여 기존 테이블의 이름을 바꾸려면 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 이름 바꾸기하려는 테이블을 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 이름 바꾸기를 선택하십시오.
3. 테이블의 새 이름을 입력하고 확인을 클릭하십시오.

명령행을 사용하여 기존 테이블의 이름을 바꾸려면 다음을 입력하십시오.

```
RENAME TABLE <schema_name.>table_name> TO <new_name>
```

다음 SQL문은 COMPANY 스키마 내의 EMPLOYEE 테이블을 EMPL로 이름 바꾸기합니다.

```
RENAME TABLE COMPANY.EMPLOYEE TO EMPL
```

RENAME TABLE문에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

테이블 삭제

테이블은 DROP TABLE SQL문으로 삭제될 수 있습니다.

테이블이 삭제되면, 삭제된 테이블에 대한 정보가 들어 있는 SYSCAT.TABLES 카탈로그의 행과, 테이블에 의존하는 다른 오브젝트는 영향을 받습니다. 예를 들면, 다음과 같습니다.

- 모든 컬럼 이름이 삭제됩니다.
- 테이블의 컬럼에 대해 작성된 색인이 삭제됩니다.
- 테이블에 근거한 모든 뷰가 사용 불가능 상태로 표시됩니다(238 페이지의 『사용 불가능 상태 뷰 복구』에서 자세한 내용을 참조하십시오).
- 삭제된 테이블과 종속 뷰에 대한 모든 특권도 내재적으로 권한 취소됩니다.
- 상위 테이블 또는 종속 테이블에 있는 모든 참조 제한조건이 삭제됩니다.
- 삭제된 테이블에 종속된 모든 패키지 및 캐쉬된 동적 SQL문은 유효하지 않음으로 표시되며 종속 오브젝트가 다시 작성될 때까지 이 상태로 남아 있습니다. 계층구조에서 삭제중인 하위테이블 위에 있는 모든 상위테이블에 종속된 패키지도 여기에 포함됩니다(244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오).
- 삭제된 테이블이 참조 범위로서 정의된 모든 참조 컬럼은 『범위 없는 상태』가 됩니다.
- 별명이 정의될 수 없기 때문에 테이블에 대한 별명 정의는 효력이 없습니다.
- 삭제된 테이블에 종속적인 모든 트리거는 작동 불능으로 표시됩니다.

- DATALINK 컬럼과 링크된 모든 파일은 링크해제됩니다. 링크해제 조작은 비동기식으로 수행되는데, 해당 파일을 다른 조작용으로 즉시 사용할 수 없다는 의미입니다.

제어 센터를 사용하여 테이블을 삭제하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 테이블을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. 확인 상자를 클릭하고 확인을 클릭하십시오.

명령행을 사용하여 테이블을 삭제하십시오.

```
DROP TABLE <table_name>
```

다음 명령문은 DEPARTMENT라고 하는 테이블을 삭제합니다.

```
DROP TABLE DEPARTMENT
```

각각의 테이블은 하위테이블이 있을 때는 삭제될 수 없습니다. 그러나, 테이블 계층구조에 있는 모든 테이블은 다음 예에서처럼 단일 DROP TABLE HIERARCHY 문으로 삭제될 수 있습니다.

```
DROP TABLE HIERARCHY person
```

DROP TABLE HIERARCHY문은 삭제될 계층구조의 루트 테이블을 이름 지정해야 합니다.

특정 테이블을 삭제하는 것과 테이블 계층구조를 삭제하는 것에는 차이가 있습니다.

- DROP TABLE HIERARCHY는 개별 DROP 테이블문으로 활성화되는 삭제 트리거를 활성화시키지 않습니다. 예를 들어, 개별 하위테이블을 삭제하면, 상위 테이블에 대한 삭제 트리거가 활성화됩니다.
- DROP TABLE HIERARCHY는 삭제된 테이블의 개별 행에 대해 로그 항목을 만들지 않습니다. 대신 계층구조 삭제는 단일 이벤트로 기록됩니다.

DROP문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

사용자 정의 임시 테이블 삭제

DECLARE GLOBAL TEMPORARY TABLE문을 사용하여 작성된 사용자 정의 임시 테이블을 삭제할 때 알아야 할 몇 가지 고려사항이 있습니다.

그러한 테이블을 삭제할 때, 테이블 이름은 스키마 이름 SESSION으로 규정되어야 하며 테이블을 작성한 응용프로그램에 있어야 합니다.

패키지는 이 테이블 유형에 종속될 수 없으므로 그러한 테이블을 삭제할 때 유효합니다.

사용자 정의 임시 테이블이 삭제되고, 임시 테이블 작성이 사용중인 작업 단위(UOW)나 savepoint를 선행한 경우, 테이블은 기능적으로 삭제되어 응용프로그램이 테이블에 액세스할 수 없습니다. 그러나, 테이블은 여전히 테이블 공간에 예약된 일부 공간을 가지며 이것은 작업 단위(UOW)가 확약되거나 savepoint가 종료될 때까지 사용자 임시 테이블 공간을 삭제하지 못하게 합니다.

DROP문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

트리거 삭제

트리거 오브젝트는 DROP문을 사용하여 삭제될 수 있지만, 이 프로시저어로 인해 종속 패키지는 다음과 같이 유효하지 않음으로 표시됩니다.

- 명시적 컬럼 목록 없이 갱신 트리거가 삭제될 경우, 대상 테이블에서 갱신 사용이 있는 패키지가 무효화됩니다.
- 컬럼 목록과 함께 갱신 트리거가 삭제될 경우, 패키지가 CREATE TRIGGER 문의 column-name 목록 내의 적어도 하나의 컬럼에서 갱신 사용이 있을 경우에만 대상 테이블에 갱신 사용이 있는 패키지가 무효화됩니다.
- 삽입 트리거가 삭제될 경우, 대상 테이블에서 삽입 사용이 있는 패키지가 무효화됩니다.
- 삭제 트리거가 삭제될 경우, 대상 테이블에서 삭제 사용이 있는 패키지가 무효화됩니다.

패키지는 응용프로그램이 명시적으로 바인드 또는 리바인드될 때까지 무효로 있거나, 패키지가 실행되면 데이터베이스 관리 프로그램이 자동으로 이를 리바인드합니다.

사용자 정의 함수(UDF), 유형 맵핑 또는 메소드 삭제

사용자 정의 함수(UDF), 함수 템플릿 또는 함수 맵핑은 DROP문으로 삭제될 수 있습니다.

DISABLE 맵핑 옵션으로 함수 맵핑을 작동 불가능하게 할 수 있습니다. 수행 방법에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

뷰, 트리거, 테이블 점검 제한조건 또는 다른 UDF가 이에 종속적인 경우에는, UDF가 삭제될 수 없습니다. CREATE DISTINCT TYPE문에 의해 내재적으로 생성된 함수는 삭제될 수 없습니다. SYSIBM 스키마 또는 SYSFUN 스키마에 있는 함수는 삭제할 수 없습니다.

기타 오브젝트는 함수 또는 함수 템플릿에 종속될 수 있습니다. 함수 맵핑을 비롯한 이러한 모든 종속성은 함수가 삭제되기 전에 삭제되어야 합니다. 단, 작동하지 않는 것으로 표시된 패키지는 예외입니다. 이러한 패키지는 내재적으로 리바인드되지 않습니다. 이것은 BIND 또는 REBIND 명령을 사용하여 리바인드되거나, PREP 명령을 사용하여 준비되어야 합니다. 이들 명령에 관한 자세한 정보는 *Command Reference* 매뉴얼을 참조하십시오. UDF를 삭제하면, 이것을 사용한 다른 패키지 또는 캐쉬된 동적 SQL문이 무효화됩니다.

함수 맵핑을 삭제하면, 패키지는 유효하지 않은 것으로 표시됩니다. 자동 리바인드가 일어나며 최적화 알고리즘은 지역 함수를 사용하려고 시도합니다. 지역 함수가 템플릿인 경우, 내재적 리바인드는 실패합니다.

(244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.)

사용자 정의 유형(UDT) 또는 유형 맵핑 삭제

사용자는 DROP문을 사용하여 사용자 정의 유형(UDT) 또는 유형 맵핑을 삭제할 수 있습니다. UDT가 다음에 사용될 때는 삭제할 수 없습니다.

- 기존 테이블 또는 뷰(구별 유형)에 대한 컬럼 정의에 사용
- 기존의 입력된 테이블 또는 입력된 뷰(구조화 유형)의 유형으로 사용
- 또 다른 구조화 유형의 상위 유형으로 사용

기본 유형 매핑을 삭제할 수 없습니다. 또다른 매핑을 작성하여 이를 대체하기만 할 수 있습니다.

데이터베이스 관리 프로그램은 이러한 구별 유형에 종속적인 모든 함수 삭제를 시도할것입니다. UDF가 삭제될 수 없으면, UDT도 삭제될 수 없습니다. 뷰, 트리거, 테이블 점검 제한조건 또는 다른 UDF가 이에 종속적인 경우에는, UDF가 삭제될 수 없습니다. UDT를 삭제하면, 이것을 사용한 다른 패키지 또는 캐쉬된 동적 SQL문이 무효화됩니다.

UDT에 대한 변환을 작성하고 UDT를 삭제할 계획이라면, 변환을 삭제해야 하는지 고려해야 합니다. 이는 DROP TRANSFORM문을 통해 이루어집니다. 이 명령문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오. 사용자 또는 기타 응용프로그램 개발자가 정의한 변환만 삭제될 수 있다는 점에 유의하십시오. 내장된 변환과 관련 그룹 정의는 삭제될 수 없습니다.

사용자 정의 유형에 관한 자세한 정보는 SQL 참조서 및 응용프로그램 개발 안내서 매뉴얼을 참조하십시오.

뷰 변경 또는 삭제

ALTER VIEW문은 참조 유형 컬럼이 범위를 추가하도록 변경하여 기존 뷰를 수정합니다. 뷰에 대한 다른 모든 변경을 수행할 때에는 해당 뷰를 삭제한 다음, 다시 작성해야 합니다.

뷰를 변경할 때에는 아직 범위가 정의되지 않은 기존 참조 유형 컬럼에 범위를 추가해야 합니다. 그리고, 상위뷰로부터 컬럼을 계승할 수 없습니다.

ALTER VIEW문의 컬럼 이름 데이터 유형은 REF(입력된 테이블 이름 또는 입력된 뷰 이름의 유형)여야 합니다.

패키지 및 캐쉬된 동적 명령문에 무효 표시가 되어 있더라도, 테이블 및 색인과 같은 기타 데이터베이스 오브젝트에는 영향을 주지 않습니다. 244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.

ALTER VIEW문에 관한 추가 정보는 SQL 참조서에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 뷰를 변경하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 뷰를 오른쪽 마우스 버튼으로 클릭한 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 뷰 변경 창에서, 주석을 입력하거나 수정하고 **확인**을 클릭하십시오.

제어 센터를 사용하여 보기를 삭제하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 뷰를 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 **삭제**를 선택하십시오.
3. **확인** 상자를 클릭하고 **확인**을 클릭하십시오.

명령행을 사용하여 보기를 삭제하십시오.

```
DROP VIEW <view_name>
```

다음 예는 EMP_VIEW를 삭제하는 방법을 보여줍니다.

```
DROP VIEW EMP_VIEW
```

삭제될 뷰에 종속적인 모든 뷰가 작동 불가능 상태로 됩니다(『사용 불가능 상태 뷰 복구』에서 자세한 내용을 참조하십시오).

테이블 계층구조의 경우, 다음 예에서처럼 루트 뷰를 이름 지정하여 하나의 명령문에서 전체 뷰 계층구조를 삭제할 수 있습니다.

```
DROP VIEW HIERARCHY VPerson
```

뷰 삭제 및 작성에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

사용 불가능 상태 뷰 복구

기본 테이블의 SELECT 특권을 권한 취소하면 뷰가 **사용 불가능** 상태가 될 수 있습니다.

다음 단계는 사용 불가능 상태 뷰를 복구하는 데 도움이 됩니다.

- 뷰를 작성하기 위해 초기에 사용된 SQL문을 알아내십시오. SYSCAT.VIEW 카탈로그 뷰의 TEXT 컬럼으로부터 이 정보를 얻을 수 있습니다.

- 동일한 뷰 이름과 동일한 정의로 CREATE VIEW문을 사용하여 뷰를 다시 작성하십시오.
- GRANT문을 사용하여, 뷰에 대해 이전에 권한 부여된 모든 특권을 다시 권한 부여하십시오(사용 불가능 상태 뷰에 권한 부여된 특권이 모두 권한 취소됨을 유의하십시오).

작동 불능 뷰를 복구하지 않을 경우, DROP VIEW문으로 작동 불능 뷰를 명시적으로 삭제하거나 또는 정의는 다르지만 동일한 이름을 가진 새 뷰를 작성할 수 있습니다.

작동 불능 뷰에는 SYSCAT.TABLES 및 SYSCAT.VIEWS 카탈로그 뷰의 항목만 있습니다. SYSCAT.VIEWDEP, SYSCAT.TABAUTH, SYSCAT.COLUMNS 및 SYSCAT.COLAUTH 카탈로그 뷰에 있는 모든 항목은 제거됩니다.

요약 테이블 삭제

요약 테이블을 변경할 수는 없지만, 삭제할 수는 있습니다.

테이블을 참조하는 모든 색인, 기본 키, 외부 키 및 점검 제한조건이 삭제됩니다. 테이블을 참조하는 모든 뷰 및 트리거는 사용 불가능 상태가 됩니다. 삭제되거나 사용 불가능 상태로 표시된 오브젝트에 종속되는 모든 패키지는 무효화됩니다. 패키지 종속성에 대해서는 244 페이지의 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오.

제어 센터를 사용하여 요약 테이블을 삭제하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 요약 테이블을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. 확정 상자를 클릭하고 확인을 클릭하십시오.

명령행을 사용하여 요약 테이블을 삭제하십시오.

```
DROP TABLE <table_name>
```

다음 SQL문은 요약 테이블 XT를 삭제합니다.

DROP TABLE XT

작동 불능 요약 테이블 복구

기본 테이블의 SELECT 특권을 권한 취소하면 요약 테이블이 작동 불능 상태가 될 수 있습니다.

다음 단계는 작동 불능 요약 테이블을 복구하는 데 도움이 됩니다.

- 요약 테이블을 작성하기 위해 초기에 사용된 SQL문을 알아내십시오. SYSCAT.VIEW 카탈로그 뷰의 TEXT 컬럼으로부터 이 정보를 얻을 수 있습니다.
- 동일한 요약 테이블 이름 및 동일한 정의를 통해 CREATE SUMMARY TABLE문을 사용하여 요약 테이블을 재작성하십시오.
- GRANT문을 사용하여, 요약 테이블에 이전에 권한 부여된 모든 특권을 다시 권한 부여하십시오(사용 불가능 상태 요약 테이블에 권한 부여된 특권이 모두 권한 취소됨을 유의하십시오.)

사용 불가능 상태 요약 테이블을 복구하지 않으려는 경우, DROP TABLE문을 사용하여 사용 불가능 상태 요약 테이블을 명시적으로 삭제하거나 정의는 다르지만 동일한 이름을 사용하여 새 요약 테이블을 작성할 수 있습니다.

사용 불가능 상태 요약 테이블에는 SYSCAT.TABLES 및 SYSCAT.VIEWS 카탈로그 뷰에 있는 항목만 있습니다. SYSCAT.VIEWDEP, SYSCAT.TABAUTH, SYSCAT.COLUMNS 및 SYSCAT.COLAUTH 카탈로그 뷰의 모든 항목은 제거됩니다.

래퍼 삭제

DROP문은 데이터베이스에서 래퍼를 제거할 수 있습니다. 다음 예에서는 DRDA 래퍼를 삭제하는 방법을 보여줍니다.

```
DROP WRAPPER DRDA
```

삭제되는 래퍼에 종속된 모든 서버, 유형 맵핑, 함수 맵핑, 사용자 맵핑 및 별명이 삭제됩니다. 래퍼를 삭제할 때에는 신중해야 합니다.

DROP 랩퍼에 대해 SYSADM 또는 DBADM 권한 중 하나를 가지고 있어야 합니다.

랩퍼 삭제에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

서버 변경 또는 삭제

ALTER SERVER문은 연합 데이터베이스 카탈로그에서 기존의 서버 정의를 수정합니다. 이 명령문을 사용하여 다음을 수행할 수 있습니다.

- 특정 데이터 소스의 정의를 수정합니다.
- 특정 유형 또는 버전의 다중 데이터 소스의 정의를 수정합니다.
- 특정 데이터 소스의 구성을 변경합니다. 예를 들어, 특정 서버로 식별된 DBMS가 더 빠른 프로세서를 갖춘 새 워크스테이션으로 이주되면, `cpu_ratio` 서버 옵션을 갱신해야 합니다.

이 명령문을 사용하여 `dbname` 또는 `node` 서버 옵션을 수정할 수 없습니다.

다음 예에서는 ORA1 서버를 변경하는 방법을 보여줍니다.

```
ALTER SERVER ORA1 OPTIONS (SET CPU_RATIO '5.0')
```

서버는 연합 데이터베이스에서 삭제될 수 있습니다. 다음 예에서는 ORALOC01 서버를 삭제하는 방법을 보여줍니다.

```
DROP SERVER ORALOC01
```

삭제되는 서버에 종속된 모든 유형 맵핑, 함수 맵핑, 사용자 맵핑 및 별명은 삭제됩니다. 서버를 삭제할 때에는 신중해야 합니다.

ALTER 또는 DROP 서버에 대해 SYSADM 또는 DBADM 권한 중 하나를 가지고 있어야 합니다.

서버 삭제 및 변경에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

별명 변경 또는 삭제

ALTER NICKNAME문을 사용하여 데이터 소스 테이블 또는 뷰에 대해 지역적으로 저장된 정보를 갱신합니다. 예를 들어, 이 명령문을 사용하여 컬럼에 대한 지역 이름을 변경하거나 또는 컬럼 데이터 유형을 다른 데이터 유형으로 맵할 수 있

습니다. 이 명령문을 사용하여 컬럼 옵션을 추가할 수도 있습니다. ALTER NICKNAME 구문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

별명이 삭제되면, 해당 별명에 작성된 뷰가 작동 불능으로 표시됩니다. 별명이 뷰에서 참조되면, 별명 컬럼 이름 또는 데이터 유형을 변경할 수 없습니다.

이 명령문을 사용하려면, SYSADM 또는 DBADM 권한 중 하나를 보유하거나, 별명에 대해 CONTROL 또는 ALL 데이터베이스 특권이 있거나, ALTERIN(현재 스키마에 대해) 스키마 특권이 있거나, 연합 데이터베이스에서 별명 정의자여야 합니다.

별명 컬럼 변경 및 별명 삭제

다음 예에서는 컬럼의 지역 이름을 COL1에서 NEWCOL로 변경하여 별명 TESTNN을 변경하는 방법을 보여줍니다.

```
ALTER NICKNAME TESTNN ALTER COLUMN COL1 LOCAL NAME NEWCOL
```

다음 예에서는 별명 TESTNN을 삭제하는 방법을 보여줍니다.

```
DROP NICKNAME TESTNN
```

별명 컬럼 옵션 변경

컬럼 옵션이라고 하는 매개변수에 지정한 값의 양식으로 컬럼 정보를 지정하십시오. 이들 값을 대문자 또는 소문자로 지정할 수 있습니다. 다음 표에서는 값을 설명하고 추가 정보를 제공합니다.

표 3. 컬럼 옵션 및 설정값

옵션	유효 설정값	기본 설정값
numeric_string	<p>‘Y’ 예, 이 컬럼은 숫자 데이터로만 이루어진 문자열을 포함합니다. 중요: 이 컬럼이 뒤 공백을 가진 숫자 문자열을 포함할 경우에는 ‘Y’를 지정하지 않는 것이 좋습니다.</p> <p>‘N’ 아니오, 이 컬럼은 숫자 데이터 문자열로만 제한되지 않습니다.</p> <p>컬럼의 numeric_string을 ‘Y’로 설정하면, 최적화 알고리즘에 이 컬럼이 컬럼 데이터의 정렬과 인터페이스될 수 있는 공백을 포함하지 않음을 알려 주십시오. 이 옵션은 데이터 소스의 조합 순서가 DB2와 다를 경우에 도움이 됩니다. 이 옵션과 함께 선택된 컬럼은 다른 조합 순서로 인해 지역(데이터 소스) 평가에서 제외되지 않습니다.</p>	‘N’

표 3. 컬럼 옵션 및 설정값 (계속)

옵션	유효 설정값	기본 설정값
varchar_no_trailing_blanks	특정 VARCHAR 컬럼에서 뒤 공백을 제거할지 여부를 나타냅니다. ‘Y’ 예, 이 VARCHAR 컬럼에서 뒤 공백이 제거됩니다. ‘N’ 아니오, 이 VARCHAR 컬럼에서 뒤 공백을 제거하지 않습니다. 데이터 소스 VARCHAR 컬럼에 채움 공백이 없으면, 여기에 액세스하기 위한 최적화 알고리즘의 전략은 뒤 공백을 포함하는지에 따라 부분적으로 달라집니다. 기본적으로, 최적화 알고리즘은 실제로 뒤 공백을 포함한다고 『가정합니다』. 이 가정에 따라, 최적화 알고리즘은 이 컬럼에서 리턴되는 값이 사용자가 예상하는 값이 되도록 조회 수정을 포함하는 액세스 전략을 개발합니다. 그러나, VARCHAR 컬럼에 뒤 공백이 없으면, 최적화 알고리즘에게 이를 알려주어야 좀더 효율적인 액세스 전략을 개발할 수 있습니다. 최적화 알고리즘에 특정 컬럼에 뒤 공백이 없음을 알려려면, ALTER NICKNAME문에 해당 컬럼을 지정하십시오(구문은 SQL 참조서 참조).	‘N’

색인, 색인 확장 또는 색인 스펙 삭제

색인 정의, 색인 확장 또는 색인 스펙의 절을 변경할 수 없습니다. 색인 또는 색인 확장을 삭제한 후 다시 작성해야 합니다(색인 또는 색인 스펙을 삭제해도 다른 오브젝트가 삭제되지는 않지만, 일부 패키지는 무효화될 수 있습니다).

제어 센터를 사용하여 색인, 색인 확장 또는 색인 스펙을 삭제하려면, 다음을 수행하십시오.

1. 색인 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 삭제하려는 색인을 오른쪽 마우스 버튼으로 클릭하고 팝업 메뉴에서 삭제를 선택하십시오.
3. 확인 상자를 클릭하고 확인을 클릭하십시오.

명령행을 사용하여 색인, 색인 확장 또는 색인 스펙을 삭제하려면, 다음을 입력하십시오.

```
DROP INDEX <index_name>
```

다음 SQL문은 PH라고 하는 색인을 삭제합니다.

다음 SQL문은 IX_MAP이라고 하는 색인 확장을 삭제합니다.

```
DROP INDEX EXTENSION ix_map RESTRICT
```

색인 확장의 이름은 카탈로그에 기술된 색인 확장을 식별해야 합니다. RESTRICT 절은 색인 확장 정의에 종속되는 색인을 정의할 수 없다는 규칙을 실시합니다. 기본 색인이 이 색인 확장에 종속되는 경우, 삭제에 실패합니다.

기본 키 또는 고유 색인 키(색인 스펙이 아닌 한)는 명시적으로 삭제될 수 없습니다. 다음 방식 중 하나를 사용하여 이를 삭제해야 합니다.

- 1차 색인 또는 고유 제한조건이 기본 키 또는 고유 키를 사용하여 자동으로 작성됐을 경우, 기본 키 또는 고유 키를 삭제하면 색인이 삭제됩니다. 삭제는 ALTER TABLE문을 사용하여 이루어집니다.
- 1차 색인 또는 고유 제한조건이 사용자 정의됐을 경우, 기본 키 또는 고유 키는 ALTER TABLE문을 사용하여 먼저 삭제되어야 합니다. 기본 키 또는 고유 키가 삭제되고 나면, 색인은 더 이상 1차 색인 또는 고유 색인으로 간주되지 않고, 명시적으로 삭제될 수 있습니다.

삭제된 색인에 의존하는 모든 패키지 및 캐쉬된 동적 SQL문은 유효하지 않음으로 표시됩니다. 『오브젝트 변경시의 명령문 종속성』에서 자세한 내용을 참조하십시오. 응용프로그램은 색인의 추가 또는 삭제로 인한 변경사항의 영향을 받지 않습니다.

오브젝트 변경시의 명령문 종속성

명령문 종속성에는 패키지 및 캐쉬된 동적 SQL문이 포함됩니다. 패키지란 특정 응용프로그램에 대해 가장 효과적인 방법으로 데이터에 액세스하기 위해 데이터베이스 관리 프로그램이 필요로 하는 정보가 들어 있는 데이터베이스 오브젝트입니다. 바인딩이란 응용프로그램이 실행될 때 데이터베이스에 액세스하기 위해 데이터베이스 관리 프로그램이 필요로 하는 패키지를 작성하는 프로세스입니다. 응용프로그램 개발 안내서에서 패키지 작성 방법을 자세하게 설명합니다.

패키지 및 캐쉬된 동적 SQL문은 여러 유형의 오브젝트에 종속될 수 있습니다. 해당 오브젝트의 전체 목록은 SQL 참조서에서 자세한 내용을 참조하십시오.

이들 오브젝트는 SQL SELECT문에 포함된 테이블 또는 사용자 정의 함수와 같이 명시적으로 참조될 수 있습니다. 오브젝트는 또한 내재적으로 참조될 수도 있습니다. 예를 들어, 상위 테이블의 행이 삭제될 때 참조 제한조건에 위배되지 않도록 하기 위해 점검되어야 하는 종속 테이블이 여기에 해당됩니다. 패키지는 패키지 작성자에게 권한 부여된 특권에도 종속됩니다.

패키지 또는 캐쉬된 동적 SQL문이 오브젝트에 의존하고 해당 오브젝트가 삭제되면, 패키지 또는 캐쉬된 동적 SQL문은 "유효하지 않은" 상태가 됩니다. 패키지가 사용자 정의 함수에 의존하고 해당 함수가 삭제되면, 패키지는 "사용 불가" 상태에 놓입니다.

유효하지 않은 상태에 있는 캐쉬된 동적 SQL문은 다음에 사용될 때 자동으로 다시 최적화됩니다. 명령문에서 필요한 오브젝트가 삭제되면, 동적 SQL문의 실행은 오류 메시지와 함께 실패하게 됩니다.

유효하지 않은 상태에 있는 패키지는 다음에 사용될 때 내재적으로 리바인드됩니다. 이러한 패키지 또한 명시적으로 리바인드될 수 있습니다. 트리거가 삭제되었기 때문에 패키지가 유효하지 않은 것으로 표시된 경우, 리바인드 패키지는 더 이상 트리거를 호출하지 않습니다.

작동 불가능 상태에 있는 패키지는 사용되기 전에 명시적으로 리바인드되어야 합니다. 패키지 바인드 및 리바인드에 대해서는 응용프로그램 개발 안내서에서 자세한 내용을 참조하십시오.

연합 데이터베이스 오브젝트에는 비슷한 종속성이 있습니다. 예를 들어, 서버를 삭제하면, 해당 서버에 관련된 별명을 참조하는 패키지 또는 캐쉬된 동적 SQL은 유효하지 않게 됩니다.

어떤 경우에는, 패키지를 리바인드하는 것이 불가능할 때도 있습니다. 예를 들어, 테이블이 삭제된 후 다시 작성되지 않으면, 패키지는 리바인드될 수 없습니다. 이 경우에는, 오브젝트를 다시 작성하거나 삭제된 오브젝트를 사용하지 않도록 응용 프로그램을 변경해야 합니다.

대부분의 경우, 제한조건 중 하나가 삭제될 경우에는 패키지를 리바인드할 수 있습니다.

다음 시스템 카탈로그 뷰는 패키지 및 패키지 종속성의 상태를 판별하는 데 도움을 줍니다.

- SYSCAT.PACKAGEAUTH
- SYSCAT.PACKAGEDEP
- SYSCAT.PACKAGES

오브젝트 종속성에 관한 자세한 정보는 *SQL 참조서 매뉴얼*에 있는 DROP문을 참조하십시오.

제3부 데이터베이스 보안

제5장 데이터베이스 액세스 제어

데이터베이스 관리자와 시스템 관리자의 가장 중요한 책임 중의 하나는 데이터베이스 보안입니다. 데이터베이스의 보안에는 여러 가지 활동이 포함됩니다.

- 장비 또는 시스템 고장으로 인한 데이터나 데이터 무결성의 갑작스런 손실을 방지
- 권한 부여되지 않은 액세스로부터 가치 있는 데이터를 보호. 중요한 정보가 『권한』 없는 사용자에게 의해 액세스되지 않는지 확인해야 합니다.
- 권한 부여되지 않은 사용자가 악의로 데이터를 삭제하거나 고치는 등의 위해를 가하지 못하도록 방지
- 307 페이지의 『제6장 DB2 활동 감사』에서 언급한 사용자의 데이터 액세스 모니터링

이 절에는 다음과 같은 내용이 들어 있습니다.

- 250 페이지의 『설치를 위한 사용자 ID 및 그룹 선택』
- 253 페이지의 『서버에 대한 인증 방식 선택』
- 259 페이지의 『원격 클라이언트에 대한 인증 고려사항』
- 260 페이지의 『파티션된 데이터베이스 고려사항』
- 260 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』
- 273 페이지의 『특권, 권한 및 권한 부여』
- 290 페이지의 『데이터베이스 오브젝트에 대한 액세스 제어』
- 301 페이지의 『타스크 및 필수 권한 부여』
- 302 페이지의 『시스템 카탈로그 사용』.

보안을 위한 계획: 데이터베이스 액세스 제어 플랜의 목표를 정의하는 것으로 시작하여, 누가 어떤 환경에서 무엇에 액세스할 것인지 지정하십시오. 사용자의 계획에는 데이터베이스 기능, 기타 다른 프로그램의 기능 및 관리 프로시ду어를 사용하여 이러한 목표를 충족시킬 수 있는 방법이 다루어져야 합니다.

설치를 위한 사용자 ID 및 그룹 선택

보안 문제는 제품이 설치되는 순간부터 DB2 관리자에게 중요합니다. 각 플랫폼 고유의 빠른 시작 책에서는 DB2 계획, 설치 및 구성하는 데 필요한 모든 정보를 제공합니다.

DB2의 설치를 완료하는 단계는 사용자 이름, 그룹 이름 및 암호를 필요로 합니다. 설치 동안에, 관리자는 각 이들 요구사항에 대한 기본값을 가집니다. DB2의 설치 동안에 기본값이 사용되면, 관리자는 데이터베이스가 인스턴스를 작성하기 전에 새로운 사용자 이름, 그룹 이름 및 암호를 작성하도록 권장합니다. 새로운 사용자 이름, 그룹 이름 및 암호를 사용하면 관리자가 인스턴스 및 데이터베이스 내에서 부적절한 유형으로 기본값을 사용하는 것보다 사용자의 위험을 최소화합니다.

DB2를 설치할 때 따르는 또다른 보안 권장사항은 사용자에게 권한 부여된 기본 특권의 변경입니다. 설치 프로세스 동안, 시스템 관리(SYSADM) 특권은 각 운영 체제에서 다음 사용자에게 권한 부여됩니다.

OS/2 사용자 프로파일 관리(UPM) 관리자 또는 지역 관리자 그룹에 속하는 유효한 DB2 사용자 ID

Windows 95 또는 Windows 98

Windows 95 또는or Windows 98 사용자

Windows NT 또는 Windows 2000

관리자 그룹에 속하는 유효한 DB2 사용자 이름

UNIX

인스턴스 소유자의 사용자 ID의 기본 그룹에 속하는 유효한 DB2 사용자 이름

SYSADM 특권은 DB2 내에서 사용 가능한 특권의 가장 강력한 세트입니다(특권은 이 장에서 나중에 다룹니다). 그 결과, 사용자는 다른 모든 사용자가 기본값으로 SYSADM 특권을 가지도록 원하지 않습니다. DB2는 관리자에게 그룹 및 각 사용자 ID에게 특권을 권한 부여하고 권한 취소할 수 있는 능력을 제공합니다.

그룹 및 사용자 ID를 작성하고 지정하는 플랫폼별 정보는 여러 빠른 시작 책에 있습니다. 관리자는 데이터베이스 관리 프로그램 구성 매개변수 SYSADM_GROUP을 갱신하여, 시스템 관리자 특권을 가진 시스템 관리 그룹으로서 정의되는 그룹

을 제어할 수 있습니다. DB2 설치 및 차후 인스턴스 및 데이터베이스 작성에 대한 보안 요구사항을 완료하려면 다음 지시사항에 따라야 합니다.

시스템 관리자 그룹으로 정의된 임의의 그룹이 있어야 합니다(SYSADM_GROUP 을 갱신하여). 이 그룹의 이름은 인스턴스 소유자에 작성된 그룹으로서 용이한 식별을 허용해야 합니다. 이 그룹에 속하는 사용자 ID 및 그룹은 각 인스턴스에 대한 시스템 관리자 권한을 가집니다.

특정 인스턴스와 연관된 것으로 쉽게 인식되는 인스턴스 사용자 ID를 작성하는 방안을 고려해 보십시오. 이 사용자 ID는 위에서 작성한 SYSADM 그룹 중 하나로 이 그룹의 이름입니다. 또다른 권장사항은 이 인스턴스 사용자 ID를 인스턴스 소유자 그룹의 구성원으로서만 사용하고 다른 그룹에는 사용하지 않는 것입니다. 그러면 인스턴스 환경을 수정할 수 있는 사용자 ID와 그룹의 증가를 제어할 수 있습니다.

작성한 사용자 ID는 인스턴스 내의 데이터와 데이터베이스에 입력하기 전에 인증 되도록 암호와 연관되어야 합니다. 암호 작성시 권장사항은 조직의 암호 이름 지정 지시사항에 따르는 것입니다.

Windows NT 플랫폼 고려사항

Windows NT용 Enterprise – Extended Edition에서 작업할 때, SYSADM(System Administration) 권한은 계정이 정의된 머신에 있는 지역 관리자 그룹에 속하는 유효한 DB2 사용자 계정에 부여됩니다.

예를 들어, 사용자가 도메인 계정에 로그인하고 DB2 데이터베이스에 액세스하려고 시도하는 경우, DB2는 도메인 제어기로 가서 그룹을 열거합니다(관리자 그룹 포함). 다음 두가지 방법 중 하나로 이 활동을 변경할 수 있습니다.

1. 레지스트리 변수 DB2_GRP_LOOKUP = local로 설정하고 도메인 계정(또는 전역 그룹)을 지역 관리자 그룹에 추가하십시오.
2. 새 그룹을 지정하려면 데이터베이스 관리 프로그램 구성 파일을 갱신하십시오. 그룹이 지역 머신에 열거되기를 원하는 경우, DB2_GRP_LOOKUP 레지스트리 변수를 설정해야 합니다.

Windows NT 도메인 환경에서 기본값으로, PDC(기본 도메인 제어기)에서 관리자 그룹에 속하는 도메인 사용자만이 인스턴스에 대한 SYSADM 권한을 갖습니

다. 계정이 정의된 머신에서 항상 권한 부여를 수행하므로, 서버에서 지역 관리자 그룹에 도메인 사용자를 추가하면 도메인 사용자 SYSADM 권한을 그룹에 부여하지 않습니다.

PDC에서 관리자 그룹에 도메인 사용자를 추가하지 않으려면, 전역 그룹을 작성하여 SYSADM 권한을 부여하려는 사용자(도메인 및 지역 둘 다)를 추가해야 합니다. 이렇게 하려면, 다음 명령을 입력하십시오.

```
DB2STOP
DB2 UPDATE DBM CFG USING SYSADM_GROUP global_group
DB2START
```

UNIX 플랫폼 고려사항

UNIX 기반 플랫폼의 경우, 분리 사용자 정의 함수(UDF) 및 저장 프로시저에 대한 그룹이 작성되어야 하며, 분리 UDF 또는 저장 프로시저를 사용하는 모든 사용자 ID는 이 그룹의 구성원이어야 합니다. SYSADM 그룹을 사용하여, 분리 UDF 또는 저장 프로시저 그룹의 이름이 손쉬운 식별을 보장해야 합니다. 분리 UDF 또는 저장 프로시저에 속하는 사용자 ID는 기본적으로 그룹과 연관된 권한 및 특권을 가집니다.

보안상의 이유로, 분리(fenced) ID로서 인스턴스 이름을 사용하지 않도록 권장합니다. 그러나, 분리(fenced) UDF나 저장 프로시저를 사용하려고 계획하지 않는 경우, 다른 사용자 ID를 작성하는 대신 인스턴스 이름에 분리(fenced) ID를 설정할 수 있습니다.

권장사항은 이 그룹과 관련하여 인식되는 사용자 ID를 작성하는 것입니다. 분리(fenced) UDF 및 저장 프로시저에 대한 사용자는 인스턴스 작성 스크립트의 매개변수로서(db2icrt ... -u <FencedID>) 지정됩니다. DB2 클라이언트 또는 DB2 SDK(Software Developer's Kit)를 설치하는 경우 이것은 필요하지 않습니다.

일반 규칙

모든 오브젝트 및 사용자의 이름을 지정할 때 적용되는 규칙이 있습니다. 일부 규칙은 작업중인 플랫폼에 고유한 것입니다. 예를 들어, 이름에 대소문자를 사용하는 것과 관련된 규칙이 있습니다.

- UNIX 플랫폼에서, 이름은 소문자여야 합니다.

- OS/2에서, 이름은 대문자여야 합니다.
- Windows 플랫폼에서, 이름은 대문자, 소문자, 그리고 혼합 문자일 수 있습니다.

기타 이름 지정 규칙은 443 페이지의 『부록A. 이름 지정 규칙』에서 자세한 내용을 참조하십시오.

db2icrt 명령은 인스턴스 소유자의 홈 디렉토리 아래에 기본 SQL 라이브러리(sqllib) 디렉토리를 작성합니다.

서버에 대한 인증 방식 선택

인스턴스 또는 데이터베이스로의 액세스에는 사용자가 인증이 된 상태여야 합니다. 각 인스턴스에 대한 인증 유형은 사용자가 확인되는 방법과 확인될 장소를 결정합니다. 인증 유형은 서버의 데이터베이스 관리 프로그램 구성 파일에 저장됩니다. 인스턴스가 작성될 때에 처음 설정됩니다. 이 데이터베이스 관리 프로그램 구성 매개 변수에 대해서는 관리 안내서: 성능에 있는 『DB2 구성』을 참조하십시오. 인스턴스마다 하나의 인증 유형이 있으며, 해당 데이터베이스 서버와 제어하에 있는 모든 데이터베이스로의 액세스를 포함합니다.

연합 데이터베이스에서 데이터 소스에 액세스하려는 경우, 연합 인증 유형에 대한 데이터 소스 데이터베이스 처리 및 정의를 고려해야 합니다. 267 페이지의 『연합 데이터베이스 인증 처리』에서 자세한 내용을 참조하십시오.

다음 인증 유형이 제공됩니다.

SERVER

인증이 지역 운영 체제 보안을 사용하여 서버에서 일어남을 지정합니다. 사용자 ID와 암호가 연결 또는 접속 시도중에 지정되는 경우, 서버의 유효한 사용자 ID 및 암호의 결합과 비교되어 사용자가 인스턴스에 액세스하는 데 허가를 받았는지 결정합니다. 이것이 기본 보안 메커니즘입니다.

주: 서버 코드는 연결이 지역인지 원격인지를 감지합니다. 지역 연결의 경우, 인증이 SERVER라면, 사용자 ID와 암호가 인증에 필요하지 않습니다.

원격 인스턴스가 SERVER 인증을 가지고 있으면, 사용자 ID와 암호는 사용자에게 의해 제공되거나 DB2에 의해 검색되어, 비록 사용자가 지역 머신 또는 도메인 머신에 이미 로그인했다더라도 유효성 확인을 위해 서버에 제공되어야 합니다.

SERVER_ENCRYPT

서버가 암호화된 SERVER 인증 체계를 수용하도록 지정합니다. 클라이언트 인증이 지정되지 않으면, 클라이언트는 서버에서 선택된 방법을 사용하여 인증됩니다.

클라이언트 인증이 DCS 또는 SERVER인 경우, 클라이언트는 사용자 ID와 암호를 서버에 전달하여 인증됩니다. 클라이언트 인증이 DCS_ENCRYPT 또는 SERVER_ENCRYPT인 경우, 클라이언트는 사용자 ID와 암호화된 암호를 전달하여 인증됩니다.

SERVER_ENCRYPT가 클라이언트에서 지정되고 SERVER가 서버에서 지정되면, 인증 레벨 불일치로 인해 오류가 리턴됩니다.

CLIENT

운영 체제 보안을 사용하여 응용프로그램이 호출된 데이터베이스 파티션에서 인증이 일어남을 지정합니다. 연결 또는 접속 시도중에 지정된 사용자 ID와 암호는 클라이언트 노드에서 유효한 사용자 ID 및 암호의 결합과 비교되어 사용자 ID가 인스턴스로의 액세스를 허가받았는지 결정합니다. 데이터베이스 서버에서 인증은 더 이상 일어나지 않습니다.

사용자가 지역 또는 클라이언트 로그인을 실행하면, 사용자는 해당 지역 클라이언트 워크스테이션에만 알려집니다.

원격 인스턴스가 CLIENT 인증을 가지고 있으면, 다른 두 매개변수 (*trust_allclnts* 및 *trust_clntauth*)가 최종 인증 유형을 결정합니다.

트러스트된 클라이언트용 CLIENT 레벨 보안

트러스트된 클라이언트는 신뢰 받는 지역 보안 시스템을 갖는 클라이언트입니다. 특히, 모든 클라이언트는 Windows 95 및 Windows 98 운영 체제를 제외하고 트러스트된 클라이언트입니다.

CLIENT의 인증 유형이 선택되면, 추가 옵션이 선택되어 운영 환경에 고유의 보안이 없는 클라이언트를 보호합니다.

보안이 없는 클라이언트를 보호하려면, 관리자는 `trust_allclnts` 매개변수를 NO로 설정하여 트러스트된 클라이언트 인증을 선택할 수 있습니다. 이는 모든 트러스트된 플랫폼이 서버를 대신하여 사용자를 인증할 수 있음을 나타냅니다. 신뢰 받지 않은 클라이언트는 서버에서 인증을 받고 사용자 ID와 암호를 제공해야 합니다. `trust_allclnts` 구성 매개변수를 사용하면 사용자가 신뢰 받고 있는 클라이언트인지 나타냅니다. 매개변수의 기본값은 YES입니다.

주: 모든 클라이언트(`trust_allclnts`가 YES임)가 인증에 대한 원시 안전 보안 시스템이 없는 클라이언트로 신뢰 받을 수 있습니다.

트러스트된 클라이언트라 하더라도 서버에서 완전한 인증을 받아야 할 경우가 있습니다. 트러스트된 클라이언트를 유효하게 하는 곳을 나타내기 위해, `trust_clntauth` 구성 매개변수를 사용합니다. 이 매개변수의 기본값은 CLIENT입니다. 이 매개변수에 관한 자세한 정보는 *관리 안내서: 성능에 있는 『DB2 구성』*을 참조하십시오.

주: 트러스트된 클라이언트에서만, CONNECT 또는 ATTACH 시도중에 사용자 ID 또는 암호가 명시적으로 제공되면, 사용자의 유효성이 클라이언트에서 일어납니다. `trust_clntauth` 매개변수만 사용하여 USER/USING절에서 제공된 정보를 유효화하는 곳을 결정합니다.

DRDA 클라이언트를 제외한 모든 클라이언트를 MVS 및 OS/390용 DB2, VM 및 VSE용 DB2, 그리고 OS/400용 DB2에서 보호하려면, `trust_allclnts` 매개변수를 DRDAONLY로 설정하십시오. 이 클라이언트만이 클라이언트측 인증을 수행하도록 신뢰 받을 수 있습니다. 다른 모든 클라이언트는 서버에 의해 인증되려면 사용자 ID와 암호를 제공해야 합니다.

`trust_clntauth` 매개변수는 위 클라이언트가 인증되는 위치를 결정하는데 사용됩니다. `trust_clntauth`가 "클라이언트"이면, 인증은 클라이언트에서 발생합니다. `trust_clntauth`가 "서버"이면, 인증은 암호가 제공되지 않을 경우는 클라이언트에서, 암호가 제공되면 서버에서 발생합니다.

표 4. TRUST_ALLCLNTS 및 TRUST_CLNTAUTH 매개변수 조합을 사용한 인증 모드

TRUST_ ALLCLNTS	TRUST_ CLNTAUTH	트러스트되지 않은 DRDA 클라이언트 인증. 암호 없음	트러스트되지 않은 DRDA 클라이언트 인증. 암호 있음	트러스트된 비DRDA 클라이언트 인증. 암호 없음	트러스트된 비DRDA 클라이언트 인증. 암호 있음	DRDA 클라이언트 인증. 암호 없음	DRDA 클라이언트 인증. 암호 있음
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

DCS DB2 Connect를 사용하여 액세스된 데이터베이스를 카탈로그화하는 데 주로 사용됩니다(이 주제에 대해서는 보안의 *DB2 Connect 사용자 안내서* 섹션을 참조하십시오). 이를 사용하여 데이터베이스 관리 프로그램 구성 과일의 인스턴스용으로 인증 유형을 지정하는 경우, APPC 프로토콜을 사용한 DRDA 응용프로그램 서버(AS)를 통해 서버가 액세스되지 않으면, 인증 **SERVER**와 같아집니다. 이 경우, **DCS**를 사용하면 인증이 서버에서 발생하지만, 이것은 APPC 계층에만 해당됩니다. DB2 코드에서 인증이 더 이상 일어나지 않습니다. 이 값은 연결용 APPC SECURITY 매개변수가 SAME 또는 PROGRAM으로 지정된 경우에만 지원됩니다.

DCS_ENCRYPT

DB2 Connect가 암호화된 SERVER 인증 체계를 수용하도록 지정합니다. 클라이언트 인증이 지정되지 않으면, 클라이언트는 서버에서 선택된 방법을 사용하여 인증됩니다.

클라이언트 인증이 DCS 또는 SERVER인 경우, 클라이언트는 사용자 ID와 암호를 DB2 Connect에 전달하여 인증됩니다. 클라이언트 인증이 DCS_ENCRYPT 또는 SERVER_ENCRYPT인 경우, 클라이언트는 사용자 ID와 암호화된 암호를 전달하여 인증됩니다.

DCS_ENCRYPT가 클라이언트에서 지정되고 DCS가 서버에서 지정되면, 인증 레벨 불일치로 인해 오류가 리턴됩니다.

DCE 사용자가 DCE 보안 서비스를 사용하여 인증 받은 것을 지정합니다. DCE 보안에 대해서는 260 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』에서 자세한 내용을 참조하십시오.

DCE_SERVER_ENCRYPT

서버가 DCE 인증 또는 암호화된 SERVER 인증 체계를 수용하도록 지정합니다. 클라이언트 인증이 DCE이거나 지정되지 않으면, 클라이언트는 DCE 보안 서버를 사용하여 인증됩니다. DCE 보안에 대해서는 260 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』에서 자세한 내용을 참조하십시오.

클라이언트 인증이 DCE 또는 SERVER이면, 클라이언트는 사용자 ID와 암호를 서버로 전달하여 인증됩니다. 클라이언트 인증이 SERVER_ENCRYPT 또는 DCS_ENCRYPT인 경우, 클라이언트는 사용자 ID와 암호화된 암호를 전달하여 인증됩니다. 클라이언트의 인증 유형은 DCE_SERVER_ENCRYPT로서 지정될 수 없습니다. 인스턴스의 인증 유형이 DCE_SERVER_ENCRYPT로서 지정되면, 모든 지역 응용프로그램은 DCE를 인증 체계로서 사용합니다. 이것은 데이터베이스 연결 또는 인스턴스 접속이 필요없는 유틸리티 명령에도 적용됩니다.

DCE 및 SERVER_ENCRYPT 인증 유형의 혼합을 허용할 뿐만 아니라, DCE_SERVER_ENCRYPT 인증 유형은 DCE 내의 그룹을 사용할 때 제한사항 중 하나를 완화시키기도 합니다. 인증 유형이 DCE_SERVER_ENCRYPT로 설정되면, 인증시가 아닌 다른 때 요청되는 그룹 목록은 DCE가 아닌 기본 운영 체제에서 생성된 것으로 가정합니다. 그런 다음, 관리자로서 인증시 지원되는 것 외에 그룹 목록 지원을 제공하기 위해 축약형 DCE 이름과 일치하도록 서버에서 사용자를 설정할 수도 있습니다.

KERBEROS

Kerberos 보안 프로토콜을 지원하는 운영 체제에 DB2 클라이언트 및 서버 둘다가 있을 때 사용됩니다. Kerberos 보안 프로토콜은 공유 비밀 키를 작성하기 위해 일반 암호를 사용하여 씨드 파티 인증 서비스로서 인증을 수행합니다. 이 키는 사용자 증명 되며 지역 또는 네트워크 서비스가 요청될 때 모든 경우에서 사용자 식별을 검증하는 데 사용됩니다. 키는 명

확한 텍스트로서 네트워크에서 사용자 이름 및 암호를 전달하는 필요성을 줄입니다. Kerberos를 사용하여 보안 프로토콜은 원격 DB2 서버로의 단일 사인온의 사용을 가능하게 합니다.

KRB_SERVER_ENCRYPT

서버가 KERBEROS 인증 또는 암호화된 SERVER 인증 체계를 수용하도록 지정합니다. 클라이언트 인증이 KERBEROS이면, 클라이언트는 kerberos 보안 시스템을 사용하여 인증됩니다. 클라이언트 인증이 KERBEROS가 아니라면, 시스템 인증 유형은 SERVER_ENCRYPT와 같습니다.

주: Kerberos 인증 유형은 Windows 2000을 수행하는 클라이언트 및 서버에서만 지원됩니다.

주:

1. 선택한 인증 유형은 원격 데이터베이스 클라이언트가 데이터베이스에 액세스할 때 또는 연합된 데이터베이스 기능을 사용하는 경우에만 중요합니다. 지역 클라이언트를 통해 데이터베이스에 액세스하는 대부분의 사용자는 항상 데이터베이스와 동일한 머신에서 인증되어야 합니다. DCE 보안 서비스가 사용될 때에는 예외가 있을 수도 있습니다. 원격 클라이언트 지원 및 사용에 대해서는 빠른 시작 매뉴얼을 참조하십시오.
2. 구성 파일 자체로의 액세스는 구성 파일의 정보에 의해 보호되므로, 인증 정보를 변경하려면 인스턴스로부터 부주의하게 사용자를 잠그지 마십시오. 다음의 데이터베이스 관리 프로그램 구성 파일 매개변수는 다음 인스턴스로의 액세스를 제어합니다.

- AUTHENTICATION *
- SYSADM_GROUP *
- TRUST_ALLCLNTS
- TRUST_CLNTAUTH
- SYSCTRL_GROUP
- SYSMANT_GROUP

* 가장 중요한 두 매개변수를 나타내며 이 매개변수가 문제를 일으킬 수 있습니다.

이러한 상황이 발생하지 않도록 할 수 있는 방법이 있습니다. 우연히 DB2 시스템으로부터 자신을 잠그게 되면, 확실한 특권을 받은 지역 운영체제 보안 사용자를 사용하여 데이터베이스 관리 프로그램 구성 파일을 갱신하는 보통 DB2 보안 점검을 대체할 수 있는 모든 플랫폼에서 사용 가능한 장애 안전 옵션이 있습니다. 이 사용자는 데이터베이스 관리 프로그램 구성 파일을 갱신하여 문제를 수정하는 특권을 항상 가지고 있습니다. 그러나, 이 보안 바이패스는 데이터베이스 관리 프로그램 구성 파일의 지역 갱신으로 제한됩니다. 원격으로 또는 다른 DB2 명령에 대해 장애 안전 사용자를 사용할 수 없습니다. 이 특별한 사용자는 다음과 같이 식별됩니다.

- UNIX 플랫폼: 인스턴스 소유자
- NT 플랫폼: 지역 『관리자』 그룹에 속한 사용자
- OS/2 플랫폼: UPM 관리자
- 기타 플랫폼: 기타 플랫폼에는 지역 보안이 없으므로, 모든 사용자는 지역 보안 점검을 전달합니다.

3. Windows NT 보안은 519 페이지의 『부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법』에서 자세한 내용을 참조하십시오.

원격 클라이언트에 대한 인증 고려사항

원격 액세스용 데이터베이스를 카탈로그화할 경우, 인증 유형은 데이터베이스 디렉토리 항목에 지정됩니다.

DB2 Connect를 사용하여 액세스된 데이터베이스: 값이 지정되지 않으면, SERVER 인증으로 간주됩니다.

DB2 Connect를 사용하지 않고 원격으로 액세스된 데이터베이스: 인증 유형은 필요하지 않습니다. 그러나, 지정되지 않으면 클라이언트는 먼저 서버에 접속하여 인증 흐름을 시작하기 전에 값을 얻어야 합니다. 지정되면, 인증은 서버에서 지정된 값과 일치하는 값을 제공하여 즉시 시작할 수 있습니다. 불일치가 감지되면, DB2는 복구를 시도하여 차이를 조정하기 위해 더 많은 흐름이 발생하거나, DB2가 복구할 수 없으면 오류를 나타냅니다. 불일치의 경우, 서버의 값은 옳은 것으로 간주합니다.

파티션된 데이터베이스 고려사항

파티션된 데이터베이스에서, 데이터베이스의 파티션마다 정의된 같은 사용자 및 그룹 세트가 있어야 합니다. 정의가 동일하지 않으면, 사용자는 다른 파티션에 다른 사항을 실행하도록 권한이 부여됩니다. 모든 파티션에 걸쳐서 일관성이 권장됩니다.

사용자 인증을 위한 DCE 보안 서비스 사용

분산 데이터베이스 환경에서 보안을 고려하는 경우, DCE 보안 서비스는 DCE가 다음을 제공하므로 좋은 선택입니다.

- 사용자와 암호의 집중 관리
- 뚜렷한 텍스트 암호와 사용자 ID를 전송하지 않음
- 사용자에 대한 단일 사인온

DB2는 DCE 기본 로그인 문맥, 연결 로그인 문맥, 대표 문맥을 지원합니다. 기본 로그인 문맥은 사용자가 클라이언트에서 `dce_login`을 수행할 때 설정됩니다. 이후의 DB2 명령은 이 문맥에 대한 액세스를 가지고 있으며, 더 이상의 사용자 개입 없이(즉, 사용자 ID 또는 암호가 필요없음) 사용자 인증을 실행합니다. 연결 로그인 문맥은 USER/USING 절을 사용하여 CONNECT나 ATTACH에 제공된 사용자 ID와 암호를 사용하여 DB2 세션용으로 구축됩니다. 마지막으로, 대표 로그인 문맥은 DB2 클라이언트가 DB2 서버 응용프로그램의 일부로 사용된 경우 발생됩니다. DCE 서버 응용프로그램(또한 DB2 클라이언트)은 사용자의 원래 ID가 발원한 시점에서 DCE 클라이언트 응용프로그램의 요청을 수신합니다. DCE 클라이언트와 DCE 서버가 올바르게 구성되어 DCE 서버가 DCE 클라이언트에 대해 대표하는 경우, DB2는 대표 토큰을 얻어 이 토큰을 DB2 서버로 가져갑니다. DB2 서버가 DCE 서버 ID를 사용하기 보다는 DCE 클라이언트의 원래 ID를 사용하여 요청을 처리할 수 있습니다. 대표 로그인 문맥을 설정하는 방법에 대해서는 사용자 플랫폼용 DCE 문서에서 얻을 수 있습니다.

주: DCE를 지원하는 벤더 제품으로는 여러 개가 있습니다. Windows NT용 DB2 UDB가 보안 서비스 영역에서 IBM의 DCE 제품에 대해 작업할 수 있도록 새로운 두 개의 DLL(`db2dces.ibm` 및 `db2dcec.ibm`)이 제공됩니다(이 DLL 파일은 Windows NT용으로만 적합합니다). 보안 서비스용 IBM DCE 제품

을 구입하여 사용하는 경우, 이 두 파일을 각각 db2dces.dll 및 db2dcec.dll에 복사해야 합니다. 다른 벤더의 DCE 제품 사용을 고려하고 있는 경우, 벤더의 보안 서비스용 DCE 제품이 DB2 UDB와 함께 작동하는지 알아보려면 벤더 서비스 기관 및 DB2 UDB 서비스 기관에 문의해야 합니다.

DCE용 DB2 사용자 설정 방법

사용자는 DCE 레지스트리에 등록되어 있어야 하고 DB2와 사용하기 전에 올바른 속성을 가지고 있어야 합니다. DCE 핵심부를 작성하는 방법에 관한 자세한 정보는 플랫폼에 맞는 DCE 문서를 참조하십시오.

DCE 인증 서버를 사용하고자 하는 각 DB2 사용자는 활성화된 클라이언트 플러그와 함께 DCE 레지스트리에 정의된 계정 및 DCE 핵심부가 있어야 합니다. 이 핵심부에는 특정한 DCE 인증된 서버로 연결할 때 권한 부여 이름이 이 핵심부에 대해 사용되는 것을 나타내는 확장 레지스트리 속성(ERA) 섹션 항목이 있어야 합니다.

또한 데이터베이스에서 그룹 특권을 사용하기 위해, 핵심부가 그룹의 구성원이 되도록 할 수 있습니다. 그룹 ERA의 비슷한 정보는 그룹 이름을 DB2 권한 부여 이름에 맵합니다. 권한 부여 이름은 2차적 권한 부여 이름이지만, 동일한 제한사항이 적용됩니다. 그룹 작성 및 구성원 추가에 관한 자세한 정보는 DCE 데이터를 참조하십시오.

ERA의 정보는 사용자의 DCE 핵심부 이름 또는 그룹 이름을 특정 서버 DCE 핵심부 이름의 DB2 권한 부여 이름에 맵합니다. ERA를 사용하려면, 이 속성 형식을 나타내는 ERA 스키마가 정의된 상태여야 합니다. DCE 셸마다 한 번씩 이루어져야 하고 다음 단계를 수행하여 완료됩니다.

1. 유효한 DCE 관리자로서 DCE에 로그인하십시오.
2. dcecp를 호출하고 프롬프트에서 다음을 입력하십시오.

```
> xattrschema create ./:/sec/xattrschema/db2map \  
> -aclmgr {{principal r m r m } {group r m r m }} \  
> -annotation {Schema entry for DB2 database access} \  
> -encoding stringarray \  
> -multivalued no \  
> -uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b
```

이 단계는 확장 레지스트리 속성 db2map을 작성합니다.

이 맵핑을 보려면, dcecp 프롬프트에서 다음 명령을 발행하십시오.

```
> xattrschema show /./sec/xattrschema/db2map
```

다음은 볼 수 있습니다.

```
{axlmgr
  {{principal {{query r} {update m} {test r} {delete m}}}
   {group      {{query r} {update m} {test r} {delete m}}}}
{annotation {Schema entry for DB2 database access}}
{applydefs no}
{intercell rejects}
{multivalued no}
{reserved no}
{scope {}}
{trigbind {}}
{trigtype none}
{unique no}
{uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b}
```

주: ERA에 기록된 권한 부여 이름의 내용에 대한 제한사항은 DCE에 의해 지켜지지 않습니다. DCE 핵심부 또는 그룹에 유효하지 않은 권한 부여 이름이 주어진 경우, DB2가 해당 사용자를 인증하려는 시도를 할 때 오류가 발생합니다(인증이 CONNECT, ATTACH, DB2START 또는 인증을 요구하는 다른 조작에서 일어남을 기억하십시오). 또한, 권한 부여 이름과 DCE 핵심부의 지정이 1대1의 고유한 형식인지 확인해야 합니다. DCE는 이러한 조건을 점검하지 않습니다.

DB2 클라이언트가 DB2 UDB 서버에 액세스하면, DCE 핵심부로 등록된 상태인 경우, ERA 정보를 추가하여 핵심부 이름에서 권한 부여 이름까지 맵핑을 제공해야 합니다. 사용자나 그룹마다 한 번씩 이루어져야 하고, 다음 단계를 수행하여 완료됩니다.

- 유효한 DCE 관리자로서 DCE에 로그인하십시오.
- dcecp를 호출하고 프롬프트에서 다음을 입력하십시오.

```
> principal modify principal_name \  
> -add {db2map map_1 map_2...map_n}
```

여기서, map_n은 다음 형식을 사용합니다.

DCE_server_principal,DB2_authid

여기서, DCE_server_principal은 DB2 UDB 서버에 대한 유효한 DCE 핵심부 이름이며(또는 이 맵핑이 다른 map_n 항목에서 지정하지 않은 임의의 DB2 서버에 대해 유효함을 나타내는 와일드카드 *임), DB2_authid는 유효한 DB2 권한 부여 이름입니다.

DCE 핵심부에 대해 DCE 그룹을 사용하려는 경우, 해당 그룹은 SYSADM 또는 SYSCTRL 권한과 같은 적절한 권한을 가지고 있는 DB2 authid에 맵핑되어야 합니다.

DB2 authid에 DCE 핵심부 이름을 맵하기 위해 사용되는 DCE 스키마에 지정한 권한 부여 식별자(authid)는 반드시 대문자로 지정해야 함을 기억하십시오. 소문자 또는 혼합 문자 authid의 사용은 오류를 야기합니다.

DCE를 사용하기 위한 DB2 서버 설정 방법

서버는 DCE 레지스트리에 등록된 핵심부여야 하며, DB2와 함께 사용하려면 올바른 속성을 가지고 있어야 합니다. DCE 서버 핵심부를 작성하는 방법에 대해서는 플랫폼에 맞는 지정된 DCE 문서를 참조하십시오.

DCE 보안 클라이언트 런타임 코드는 서버 인스턴스에 의해 설치되고 액세스가 가능해야 합니다.

인증 메커니즘으로 DCE를 사용하려는 각 DB2 서버는 DB2START를 발행할 때 DCE로 등록해야 합니다. 이를 수동으로 수행하지 않기 위해, DCE는 서버가 자체 사용자 ID와 암호(키) 정보를 키탭(keytab) 파일이라는 특수 파일에 유지보수하는 방법을 제공합니다. DB2START에서, DB2는 데이터베이스 관리 프로그램 구성 파일을 읽고 인스턴스에 대한 인증 유형을 확보합니다. 인증 유형이 DCE이면, DB2 서버는 DCE를 호출하여 키탭 파일로부터 정보를 얻게 됩니다. 이 정보는 DCE를 갖는 서버를 등록하는 데 사용된 정보입니다. 이 등록으로 서버는 DCE 클라이언트로부터 DCE 토큰을 승인하여 이 사용자를 인증하는 데 사용할 수 있습니다.

인스턴스 관리자는 DCE 명령을 사용하여 인스턴스용 키탭 파일을 작성해야 합니다. 키탭 파일을 작성하는 방법에 대한 세부사항은 사용자 플랫폼에 맞는 DCE 문

서에 있습니다. 해당 문서에서, 키탭 파일과 `dcecp keytab` 명령 또는 `rgy_edit` 명령과 연관된 세부사항을 참조하십시오. DB2 키탭 파일의 이름은 `keytab.db2`여야 하고, 인스턴스용 `sqllib` 디렉토리의 `security` 서브디렉토리에 상주해야 합니다 (Intel 운영 체제에서, 파일은 `sqllib` 디렉토리에 있는 `INSTANCENAME` 서브디렉토리의 `security` 서브디렉토리에 상주해야 합니다. 여기서, `INSTANCENAME`은 작업하는 데이터베이스의 인스턴스 이름입니다). 지정된 인스턴스용 서버 핵심부에 대해 한 항목만이 들어 있습니다. `DB2START`시에 다른 항목은 오류로 나타납니다. UNIX 운영 체제 플랫폼에서, 이 파일은 인스턴스 소유자의 읽기/쓰기만을 허용하도록 파일 사용권한이 설정되어 있어야 합니다.

다음은 키탭 파일 작성에 대한 예입니다.

- 유효한 DCE 사용자로서 DCE에 로그인하십시오.
- `rgy_edit`를 호출하고 프롬프트에서 다음을 입력하십시오.


```
> ktadd -p principal_name -pw principal_password \
> -f keytab.db2
```

DCE 구성이 완료된 후에 DCE 인증을 사용하여 DB2를 시작하려면, 인증 유형 『DCE』를 갖는 데이터베이스 관리 프로그램을 갱신하여 DCE 인증을 사용하는 것을 DB2에 알려야 합니다. 다음 CLP 명령을 발행하여 수행됩니다.

```
db2 update database manager configuration using authentication DCE
sysadm_group DCE_group_name
```

그런 다음, 유효한 DB2 DCE 사용자에게 대해 `dce_login`을 수행하고 `DB2START`를 실행하십시오.

주: DCE 인증을 사용하여 DB2를 시작하기 전에, 인스턴스에 대한 `SYSADM`으로 사용되는 DCE 사용자 핵심부를 정의하여 인스턴스를 시작, 중지하고 관리할 수 있는 유효한 DCE 사용자 ID를 가져야 합니다. 수행 방법에 관한 지시 사항은 261 페이지의 『DCE용 DB2 사용자 설정 방법』에서 자세한 내용을 참조하십시오.

이 지시사항에 추가하여, 작성된 핵심부는 인스턴스에 대한 `SYSADM_GROUP`의 구성원이어야 합니다. 기본적으로, 이 그룹 이름은 명시적으로 지정된 그룹이 없는(즉, `SYSADM_GROUP`이 널(NULL)인 경우) DCE 인증용 `DB2ADMIN`이지만, 인스턴스용 인증 유형을 사용자가 선택하

는 그룹 이름(인증 이름)으로 변경하기 전에 갱신될 수 있습니다. 선택하는 DCE 그룹에는, 해당 이름을 지정된 SYSADM_GROUP 권한 부여 이름에 맵하는 정의된 ERA가 있어야 합니다.

DB2 관리 서버의 기능 중 하나는 DB2 인스턴스를 시작하는 것입니다. AUTHENTICATION = DCE이면, 인스턴스용 DB2 키탭 파일에 사용되는 DCE 핵심부는 DB2 authid에 맵핑될 유효한 DCE 핵심부를 가져야 합니다. 이 맵핑은 DB2 인스턴스를 시작하기 위한 DB2 관리 서버에 필요합니다. 유효한 맵핑은 이 ID가 서버뿐 아니라 클라이언트로 작동할 수 있도록 합니다.

DCE를 사용하기 위한 DB2 클라이언트 인스턴스 설정 방법

클라이언트 전용 인스턴스는 데이터베이스 관리 프로그램 구성 파일을 갱신하고 인증 유형을 DCE로 설정하여 지역 조작용 DCE 인증을 사용하기 위해 설정될 수 있습니다. DCE에 등록해야 하는 서버가 없으므로 클라이언트 전용 인스턴스에 대한 키탭 파일이 있어야 할 필요는 없습니다. 일반적으로, 클라이언트 전용 DB2 인스턴스가 DCE 인증을 사용하는 것은 권장되지는 않지만(또는 필요없음), 지원됩니다.

DCE 보안을 사용하여 원격 데이터베이스에 액세스하려는 클라이언트는 적용 가능한 DCE 보안 제품에 액세스해야 합니다. 선택적으로, 사용자는 데이터베이스 디렉토리에서 목표 데이터베이스용 인증 유형 카탈로그화를 선택할 수 있습니다. 클라이언트가 DCE 인증 지정을 선택하는 경우, 완전 규정화된 DCE 서버 핵심부 이름도 지정됩니다. DCE 인증이 디렉토리에서 지정되지 않은 경우, 인증 정보와 핵심부 정보는 CONNECT 시간에 서버로부터 얻어집니다.

DCE 보안을 사용한 DB2 제한사항

DCE 인증 사용에는, DB2에 의해 제공되고 그룹 지원과 관련된 특정 SQL 함수에 대한 제한사항이 있습니다. 다음의 제한사항은 DCE 인증을 사용할 경우에 존재합니다.

1. GRANT 또는 REVOKE문을 사용하는 경우, USER 및 GROUP 키워드는 반드시 지정된 권한 부여 이름에 자격을 주도록 지정되어야 하며, 그렇지 않으면 오류가 나타납니다.

2. CREATE SCHEMA문의 AUTHORIZATION절을 사용하는 경우, 지정된 권한 부여 이름의 그룹 멤버십은 이 절 다음에 나오는 명령문을 실행하는 데 필요한 권한 부여 평가시 고려되지 않을 것입니다. CREATE SCHEMA문을 실행하는 동안 권한 부여 실패가 발생합니다.
3. 패키지가 패키지의 원래 바인더를 제외하고 사용자에게 의해 리바인드되면, 원래 바인더의 특권은 재평가됩니다. 이 경우, 원래 바인더의 그룹 멤버십은 특권을 재평가할 때 고려되지 않습니다. 리바인딩 동안 권한 부여 실패가 발생합니다.

DB2에 의해 수행되는 DCE 인증은 OSF DCE GSSAPI(Generic Security Services Application Programming Interface)를 사용하여 얻는 DCE Ticket에서 수행됩니다. DCE 보안에 대한 모든 인증은 데이터베이스 프로토콜 계층에서 발생합니다. 특정 통신 메커니즘은 DCE와 통합하는 데 필수적이지 않은 추가 통신 계층 보안을 제공할 수 있습니다. 통신 계층 인증이 완전히 데이터베이스 프로토콜 계층 인증과 무관하게 유지될 수 있는 경우에, 어떠한 제한사항도 실시되지 않습니다. 그러나, 성공적으로 연결이 설정되기 전에 데이터베이스 프로토콜 계층 및 통신 계층 인증 양쪽에 대한 기준이 만족되어야 합니다. 데이터베이스 프로토콜 계층 및 통신 프로토콜 계층 인증 메커니즘이 상호작용하는 경우, 일부 조합이 보안 노출되면 사용이 제한될 수 있습니다.

DCE 인증은 TCPIP SOCKS 지원과 결합하여 사용될 수 있으나, 두 개의 보안 메커니즘은 서로 별개로 동작합니다. 이것은 유효한 DCE 로그인 문맥을 제공해야 하며, SOCKS 서버의 기준을 충족시키는 지역 운영 체제 사용자 ID에 로그인해야 함을 의미할 수 있습니다.

DCE 인증은 NT Named Pipe와 결합하여 사용될 수 있으나, 두 개의 보안 메커니즘은 서로 별개로 동작합니다. 사용자는 유효한 DCE 로그인 컨텍스트를 제공해야 하며, NT Named Pipe 지원의 기준을 충족시키는 사용자 ID에 대한 NT 도메인에 로그인해야 합니다.

위의 두 가지 예에서와 같이 DCE 핵심부 및 지역 운영 체제 사용자 ID가 모두 인증에 사용되는 가능한 혼동을 처리하기 위해, 통합된 DCE 로그인이 사용될 수 있습니다. 이 경우 시스템에 로그인할 때, 사용자는 자동으로 해당 DCE 핵심부로 로그인됩니다. 이 기능 사용 방법에 대한 세부사항이 지원되는 경우, 사용자 플랫폼에 대한 DCE 문서를 참조하십시오. 이 접근 방법 사용시, 동일한 이름이 DCE

핵심부 및 지역 운영 체제 ID에 사용됨을 기억하십시오. 이것은 DCE 암호화 티켓에 포함된 동일한 값이 통신 계층의 암호화되지 않은 전선에서 흐름을 의미할 수 있습니다.

SECURITY 매개변수가 NONE으로 설정될 때 DCE 인증은 단지 APPC 통신과 함께 사용될 수 있습니다. 이것은 데이터베이스 프로토콜 계층에서 동일한 핵심부에 대해 암호화된 DCE 토큰을 사용하는 동안, 통신 계층에서 암호화되지 않은 핵심부 및 암호를 송신하는 가능성을 피하기 위한 것입니다. 이 때 APPC 계층에서의 DCE 보안은 DB2에 의해 지원되지 않습니다.

연합 데이터베이스 인증 처리

분산 조인 설치 기능을 설치하고 DB2 구성 변수 *federated*를 'YES'로 설정하면, DB2 시스템은 연합 시스템으로서 동작합니다. 연합 시스템에 있는 데이터베이스 인증 설정값은 표준 DB2 정의와 약간 다릅니다. 연합 시스템에서 중요한 것은 데이터 소스의 인증 요구사항을 고려하는 것입니다. 일반적으로, 데이터 소스(DB2, Oracle, OS/390용 DB2 등)는 인증이 필요하도록 설정됩니다. 이는 해당 ID와 암호(필요하므로)가 데이터 소스로 전달될 수 있도록 해야 하는 것을 의미합니다. DB2는 데이터 소스에서 인증 지원을 위해 여러 방법을 제공하며, 이에 대해서는 이 절에서 설명합니다.

인증 설정값

SERVER

DB2에 연결된 클라이언트가 사용자 ID와 암호를 제공하여 DB2에 액세스하도록 지정합니다. 이 경우, 사용자 ID와 암호는 데이터 소스로의 전송에 사용할 수 있습니다. 사용자는 서버 옵션과 사용자 맵핑을 통해 데이터 소스로 실제 전달된 내용을 제어하지만, 인증 정보는 데이터 소스로의 전송에 사용할 수 있습니다.

CLIENT

운영 체제 보안을 사용하여 응용프로그램이 호출되는 데이터베이스 파티션에서 인증이 발생하도록 지정합니다. 데이터 소스로의 직접 전송에는 암호를 사용할 수 없습니다. 이 경우, 데이터 소스에 인증이 필요하다면, 하나

이상의 사용자 맵핑을 작성해야 합니다. 또한 정확한 사용자 ID와 암호 정보를 데이터 소스에 전달하도록 서버 옵션이 제대로 설정되어 있는지 확인해야 합니다.

CLIENT 인증을 사용할 경우는 주의하십시오. 보안 네트워크에 대해서만 이러한 양식의 인증을 고려하십시오. 사용자는 다음 조건을 충족할 때 연합 데이터베이스에 대해 SYSADM 권한을 갖습니다.

- 인증이 CLIENT로 설정됩니다.
- 사용자는 클라이언트에서 루트 상태에 있습니다.
- 사용자는 SYSADM의 권한 부여 이름을 알고 있습니다.
- 사용자는 DB2상의 SYSADM과 같은 이름을 가진 권한 부여 이름을 클라이언트에서 정의합니다.

DCS DB2가 아닌 데이터 소스에서 인증이 발생되도록 지정합니다. 이 경우, 표준 DB2 인증 처리는 생략됩니다. 사용자 ID와 암호는 서버 옵션 설정값에 따라 직접 데이터 소스로 전달됩니다. 인증은 Oracle 또는 DB2 계열 데이터 소스에서만 발생합니다.

인증이 DCS로 설정되어 있으면 주의하십시오. 인증은 클라이언트에서도 DB2에서도 수행되지 않습니다. SYSADM 인증 이름을 알고 있는 사용자는 연합 서버에 대해 SYSADM 권한을 가지고 있는 것으로 가정할 수 있습니다.

DCE 인증이 DCE로 설정되면, 사용자 ID만이 데이터 소스로의 전송에 사용할 수 있습니다. 암호는 사용할 수 없습니다. 데이터 소스에 인증 처리(사용자 ID와 암호)가 필요하다면, 암호(가능한 경우, 사용자 ID)를 데이터 소스로 전송할 사용자 맵핑을 정의해야 합니다. 데이터 소스가 DB2 연결을 신뢰하면, 외부 보안 시스템에서 받은 ID가 데이터 소스로 전달될 수 있으므로 사용자 맵핑은 필요하지 않습니다.

기타 DB2 인증 설정값도 가능하며, 하나 이상을 설정하면 데이터 소스로의 전송을 위해 DB2에서 암호를 사용할 수도 있습니다. DB2와 클라이언트 인증 설정값으로 DB2로 암호가 전송되면, 해당 암호는 데이터 소스에서 추가 인증 처리에 사용할 수 있습니다. 256 페이지의 표4에서 자세한 내용을 참조하십시오.

ID 및 암호를 데이터 소스로 전달

데이터 소스로의 인증 데이터 전송을 제어하는 방법에는 DB2 인증, 사용자 맵핑, 서버 옵션 및 APPC 보안 설정값이 있습니다.

인증 설정값

이 절의 목적은 인증 설정값이 연합 시스템에서의 전역 인증 처리에 미치는 영향 정도를 명확히 하는 것입니다(인증 설정값 정의는 267 페이지의 『인증 설정값』에 있습니다). 예를 들어 DB2 인증이 SERVER 또는 DCS로 설정되면, 연결에는 사용자 ID와 암호가 필요합니다. 그러므로, 사용자 ID와 암호를 데이터 소스로의 전송에 사용할 수 있습니다. 인증이 DCE 또는 CLIENT로 설정되고, 연합 데이터 베이스가 들어 있는 DB2 시스템에서 인증이 발생되지 않으면, 사용자 ID만을 사용할 수 있습니다. 데이터 소스 인증 처리에 암호(또는 다른 사용자 ID와 암호)가 필요하다면, 사용자 맵핑을 작성해야 합니다. 인증이 CLIENT로 설정되고 *trust_clntauth* 매개변수 설정값이 SERVER이면, 암호가 DB2로 전송되어 데이터 소스로의 전송에 사용될 수도 있습니다.

사용자 맵핑

DB2는 DB2로의 연결에 사용된 권한 부여 이름 또는 DB2에 정의된 권한 부여 이름을 전송할 수 있습니다. 사용자 맵핑은 DB2에 정의된 권한 부여 이름을 저장합니다. 맵핑은 CREATE USER MAPPING문으로 작성됩니다.

사용자 맵핑은 유연성이 있으므로, ID를 암호만이 아닌 새로운 ID와 암호로 맵핑할 수 있습니다. 그런 다음, 이를 사용하여 누락된 정보를 제공하거나, ID와 암호를 데이터 소스에서 수용된 값으로 변경할 수 있습니다.

사용자 맵핑을 작성하거나 변경하려면, SYSADM 또는 DBADM 권한 중 하나를 가지고 있거나, 인증 ID가 명령문에 지정된 권한 부여 이름과 일치해야 합니다.

사용자 맵핑문 예는 다음과 같습니다.

```
CREATE USER MAPPING FOR "SHAWN" SERVER DB21 OPTIONS (REMOTE_AUTHID "SHAWNBCA",  
REMOTE_PASSWORD "MAPLELEAF")
```

여기서, DB2 인증 ID(SHAWN)는 DB21 서버에 대한 원격 ID SHAWNBCA로, 원격 암호 MAPLELEAF로 맵핑됩니다.

DB2에서의 권한 부여 이름(또는 암호)과 데이터 소스에서의 권한 부여 이름(또는 암호)의 유일한 차이점이 전달된 문자열의 대소문자 여부라면, 새로운 ID와 암호를 작성하는 대신 서버 옵션을 사용하여 원하는 설정값으로 문자를 변환시키도록 하십시오. 『서버 옵션』에서 자세한 내용을 참조하십시오.

인증 설정값이 DCE이고 데이터 소스에 인증 처리(암호를 예상함)가 필요하다면 사용자 맵핑을 작성해야 합니다. DB2는 DB2 사용자 ID만을 데이터 소스로 전달합니다. 암호는 해당 사용자 ID에 맵핑된 다음 데이터 소스로 전송되어야 합니다.

서버 옵션

서버 옵션은 전체적인 인증 지원을 제공하는 데 사용될 수 있습니다. 이를 사용하여 암호가 데이터 소스로 전달되었는지(보통 예)와, 사용자 ID와 암호를 대문자 또는 소문자로 변환해야 하는지 여부를 나타내십시오. 서버 옵션은 CREATE SERVER, ALTER SERVER 및 SET SERVER OPTION문을 사용하여 설정됩니다.

인증 처리 고유의 서버 옵션에 대해서는 이 절의 나머지 부분에서 설명합니다. 좀 더 완벽한 서버 옵션 목록은 177 페이지의 『서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화』에 있습니다.

암호 서버 옵션: 암호의 기본 설정값은 ‘Y’입니다(암호는 데이터 소스로 전달됨). 데이터 소스가 인증을 수행하지만, 암호화된 암호를 예상하지 않는 모든 경우에 이 옵션을 그대로 두거나 ‘Y’로 설정하십시오.

DB2는 암호화된 암호를 전송할 수 있습니다. 암호를 암호화된 양식으로 DB2 계열 데이터 소스에 전송해야 하는 경우, 서버 옵션 암호를 ‘ENCRYPTION’으로 설정하십시오. DB2에서 인증 설정값이 CS_ENCRYPT 또는 SERVER_ENCRYPT이면, 암호를 ‘ENCRYPTION’으로 설정하는 것이 좋습니다.

사용자 ID는 언제나 데이터 소스로 전달됩니다.

ID 및 암호 변환 옵션: 권한 부여 이름과 암호를 변경해야 하는 경우도 있습니다. 다른 데이터 소스가 ID 및 암호에 대해 다른 권한 부여 이름과 암호 요구사항(대문자를 사용하는지 또는 소문자를 사용하는지에 따라)을 가질 수 있습니다.

DB2는 이름 지정 차이점을 해결하는 데 도움이 될 수 있는 두 개의 서버 옵션을 제공합니다. 옵션 이름은 **fold_id** 및 **fold_pw**이고, 설정값은 다음과 같습니다.

‘U’ DB2가 권한 부여 이름 또는 암호를 데이터 소스로 전달하기 전에 대문자로 변환합니다.

‘N’ DB2는 권한 부여 이름 또는 암호를 변환하지 않습니다.

‘L’ DB2가 권한 부여 이름 또는 암호를 데이터 소스로 전달하기 전에 소문자로 변환합니다.

널(NULL)

DB2는 먼저 권한 부여 이름과 암호를 대문자로서 전달합니다. 실패하면, 이를 소문자로 변환하여 다시 전송합니다.

널(NULL) 설정값은 많은 경우를 처리할 수 있으므로 매력적으로 보일 수 있습니다. 그러나 성능면에서 보면, 연결에 대해 하나의 시도만을 수행하도록 이러한 옵션을 설정하는 것이 가장 좋습니다. **fold_id** 및 **fold_pw** 옵션 모두가 널(NULL)로 설정되면, DB2가 권한 부여 이름과 암호를 전송하기 위해 네 가지의 시도를 하게 됩니다.

1. 권한 부여 이름과 암호를 모두 대문자로
2. 권한 부여 이름은 대문자, 암호는 소문자로
3. 권한 부여 이름은 소문자, 암호는 대문자로
4. 권한 부여 이름과 암호 모두를 소문자로

APPC 보안 설정값

사용자 ID와 암호가 필요한 APPC를 통해 DRDA 데이터 소스로 연결하는 경우 또는 인증 설정값이 DCS이고 DRDA 데이터 소스에서 인증하는 경우, APPC 보안 설정값이 DB2와 해당 데이터 소스간의 연결을 위해 PROGRAM으로 되어 있는지 확인하십시오.

연합 데이터베이스 인증 예

이 절에서는 연합 시스템 인증 및 권한 부여 과정에 대한 개요를 설명합니다. 연합 데이터베이스 인증 및 권한 부여 처리에 대한 개요는 272 페이지의 그림3에서 자세한 내용을 참조하십시오.

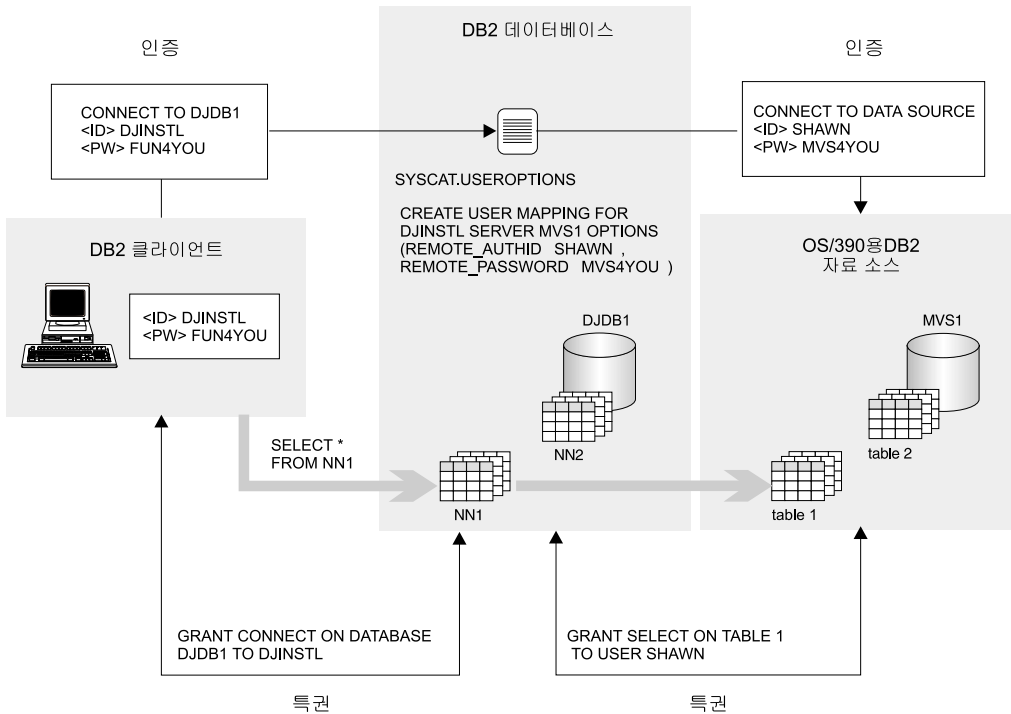


그림 3. 연합 데이터베이스 인증 및 권한 부여 처리

이 시나리오에 있는 타스크는 사용자 DJINSTL을 사용 가능하게 하여 두 개의 별명(NN1 및 NN2)에 대해 UNION 조작을 수행하는 것입니다. 별명은 두 개의 테이블을 나타냅니다. 하나의 데이터 소스는 DJINSTL에 다른 사용자 ID와 MVS1 이름의 암호가 있는 OS/390용 DB2 시스템입니다(그림3 참조). 사용자 맵핑은 MVS1에서 정보에 액세스하는데 필요합니다. 다른 데이터 소스는 DJINSTL의 ID와 암호가 같은 DB2 시스템입니다. 이 데이터 소스 DB21에서는 사용자 ID와 암호가 대문자로 전송되기만 하면 됩니다.

DB2 인증은 SERVER로 설정됩니다. DJINSTL은 TCP/IP 연결을 통해 NT 클라이언트에서 DB2에 액세스합니다. DB2에서 OS/390용 DB2로의 연결을 TCP/IP라고도 합니다. 연합 데이터베이스 이름은 DJDB1입니다.

먼저 DB2가 암호를 예상하고 있고 암호가 전송되고 있는지 확인하십시오. 또한, 클라이언트와 서버 인증 유형이 같은지도 확인하십시오. DB2 서버에서 다음 명령을 입력하여 DB2 서버 인증 유형을 점검하십시오.

GET DATABASE MANAGER CONFIGURATION

클라이언트에서 다음 명령을 입력하여 클라이언트 인증 유형을 점검하십시오.

```
LIST DATABASE DIRECTORY
```

둘다의 경우, 인증이 SERVER로 설정되어 있는지 확인하십시오. 클라이언트에 대한 설정 값이 DCE 또는 CLIENT이면, UNCATALOG DATABASE 및 CATALOG DATABASES 명령을 발행하여 이를 변경할 수 있습니다.

그런 다음, 암호가 데이터 소스로 전송되는지 확인하십시오. 연합 데이터베이스 DJDB1으로 연결한 후, 다음 명령을 발행하십시오.

```
ALTER SERVER MVS1 OPTIONS (SET password 'Y')
ALTER SERVER DB21 OPTIONS (SET password 'Y')
```

암호가 적합한 문자로 DB21 데이터 소스에 전송되는지 확인하십시오.

```
ALTER SERVER DB21 OPTIONS (ADD fold_id 'U')
ALTER SERVER DB21 OPTIONS (ADD fold_pw 'U')
```

다음 단계는 사용자 DJINSTL이 연합 데이터베이스 DJDB1로 연결되도록 특권을 권한 부여하고 별명을 선택하는 것입니다.

```
GRANT CONNECT ON DATABASE DJDB1 TO DJINSTL;
```

DJINSTL의 DB2 ID와 암호를 MVS1 서버에 대한 적합한 사용자 ID 및 암호로 다음과 같이 맵핑하십시오.

```
CREATE USER MAPPING FOR "DJINSTL" SERVER MVS1 OPTIONS (REMOTE_AUTHID "SHAWN",
REMOTE_PASSWORD "MVS4YOU")
```

이때, DB2 사용자 ID DJINSTL은 데이터 소스로 요청을 전송할 수 있습니다. 별명이 참조하는 데이터 소스 오브젝트에 액세스하는 데 추가 과정이 필요할 수도 있습니다(특권은 주로 별명이 참조하는 테이블과 뷰에 필요합니다).

특권, 권한 및 권한 부여

특권은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있도록 합니다. 권한 레벨은 상위 레벨 데이터베이스 관리 프로그램 유지보수 및 유틸리티 조작과 특권을 그룹화하는 방식을 제공합니다. 이들이 함께 활동하여 데이터베이스 관리 프로그램과 데이터베이스 오브젝트에 대한 액세스를 제어합니다. 사용자는 적절한 권한 부여, 즉 필요한 특권 또는 권한을 가지고 있는 오브젝트만을 액세스할 수 있습니다.

다음 권한이 있습니다.

- 276 페이지의 『시스템 관리 권한(SYSADM)』
- 279 페이지의 『데이터베이스 관리 권한(DBADM)』
- 277 페이지의 『시스템 제어 권한(SYSCTRL)』
- 278 페이지의 『시스템 유지보수 권한(SYSMAINT)』.

다음 유형의 특권이 있습니다.

- 281 페이지의 『데이터베이스 특권』
- 283 페이지의 『스키마 특권』
- 284 페이지의 『테이블 공간 특권』
- 284 페이지의 『테이블 및 뷰 특권』
- 287 페이지의 『별명 특권』
- 288 페이지의 『서버 특권』
- 289 페이지의 『패키지 특권』
- 290 페이지의 『색인 특권』.

275 페이지의 그림4에서는 권한과 이들의 제어 스펙간의 관계(데이터베이스, 데이터베이스 관리 프로그램)에 대해 설명합니다.

권한

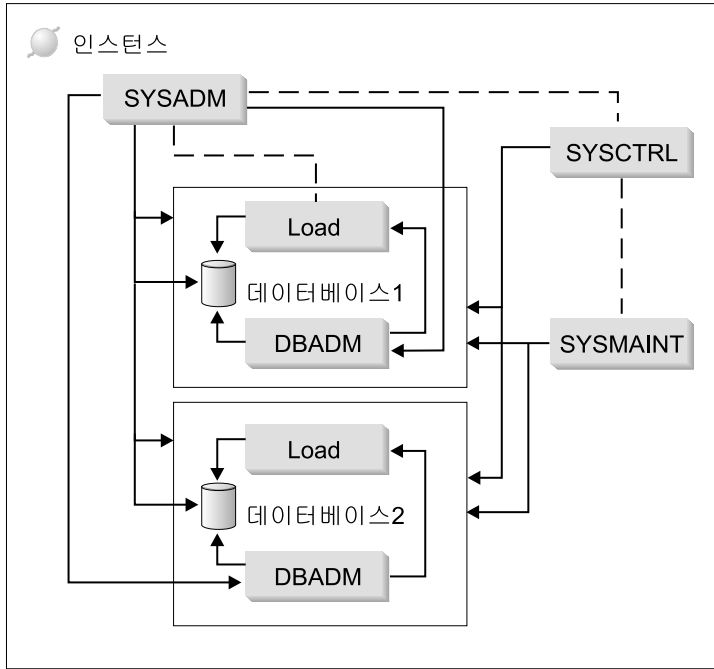


그림 4. 권한 계층

사용자 또는 그룹이 다음과 같은 권한 부여 레벨 중 하나 이상에 있을 수 있습니다.

- 관리 권한(SYSADM 또는 DBADM)은 오브젝트 집합에 대한 전반적인 특권을 부여합니다.
- 시스템 권한(SYSCTRL 또는 SYSMAINT)은 시스템 관리에 대한 전반적인 특권을 부여하지만, 데이터에 액세스하지는 못합니다.
- LOAD 권한(LOAD)은 LOAD 유틸리티 및 AutoLoader 유틸리티 특권을 제공하여 데이터를 테이블로 로드합니다.
- 소유권 특권(또한 어떤 경우에는 CONTROL 특권)은 특정 오브젝트에 대한 완전한 특권을 제공합니다.
- 개인 특권은 사용자가 특정 오브젝트에 대해 특정 기능을 수행할 수 있도록 권한 부여할 수 있습니다.
- 내재적 특권은 패키지를 실행하는 특권을 갖는 사용자에게 권한 부여할 수 있습니다. 사용자가 응용프로그램을 수행하는 동안에는, 패키지 내에서 사용되는 데이터 오브젝트에

대한 명시적인 특권이 반드시 필요하지는 않습니다. 295 페이지의 『패키지를 통한 간접 특권 허용』에서 자세한 내용을 참조하십시오.

관리 권한(SYSADM 또는 DBADM) 또는 소유권 특권(CONTROL)이 있는 사용자는 GRANT 및 REVOKE문을 사용하여 다른 사용자에게 권한 부여하거나 권한 취소합니다 (290 페이지의 『데이터베이스 오브젝트에 대한 액세스 제어』 참조). 해당 특권이 WITH GRANT OPTION이면, 테이블, 뷰 또는 스키마 특권을 다른 사용자에게 권한 부여할 수도 있습니다. 그러나, WITH GRANT OPTION은 권한을 부여한 사용자가 일단 권한 부여한 권한을 권한 취소할 수는 없습니다. 권한을 권한 취소하려면 SYSADM 권한, DBADM 권한 또는 CONTROL 권한이 있어야 합니다.

사용자 또는 그룹은 특권 또는 권한을 조합할 수 있는 권한이 부여될 수 있습니다. 자원과 관련이 있는 특권일 경우에는, 해당 자원이 존재해야 합니다. 예를 들어, 사용자에게는 테이블이 이전에 작성되지 않는 한, 테이블에 대한 SELECT 특권이 주어질 수 없습니다.

주: 권한 부여 이름에 권한과 특권이 부여되고 해당 권한 부여 이름을 통해 사용자가 작성되어 있지 않을 경우 주의해야 합니다. 얼마 후, 해당 권한 부여 이름을 통해 사용자를 작성할 수 있으며, 작성된 사용자는 해당 권한 부여 이름과 관련된 모든 권한과 특권을 자동으로 부여받게 됩니다.

특정 명령, API 또는 SQL문에 필요한 권한에 대해서는 *Command Reference*, *Administrative API Reference* 또는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

시스템 관리 권한(SYSADM)

SYSADM 권한은 최상위 레벨의 관리 권한입니다. SYSADM 권한을 갖는 사용자는 유틸리티를 수행하고, 데이터베이스 및 데이터베이스 관리 프로그램 명령을 발행하며, 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스에 있는 테이블의 데이터에 액세스할 수 있습니다. 인스턴스에 있는 모든 데이터베이스 오브젝트, 즉 데이터베이스, 테이블, 뷰, 색인, 패키지, 스키마, 서버, 별명, 데이터 유형, 함수, 프로시저, 트리거, 테이블 공간, 노드 그룹, 버퍼 풀 및 이벤트 모니터 등을 제어할 수 있는 기능을 제공합니다.

SYSADM 권한은 *sysadm_group* 구성 매개변수에서 지정한 그룹으로 지정됩니다 (관리 안내서: 성능에 있는 『DB2 구성』 참조). 해당 그룹의 멤버십은 사용자 플랫폼에 사용된 보안 기능을 통해 데이터베이스 관리 프로그램 외부에서 제어됩니

다. 시스템 보안 기능을 사용하여 SYSADM 권한을 작성, 변경 또는 삭제하는 방법에 대해서는 빠른 시작에서 자세한 내용을 참조하십시오.

SYSADM 권한을 가진 사용자만이 다음 기능을 수행할 수 있습니다.

- 데이터베이스 이주
- 데이터베이스 관리 프로그램 구성 파일 변경(SYSCTRL 또는 SYSMMAINT 권한을 갖는 그룹 지정 포함)
- DBADM 권한 부여

뿐만 아니라, SYSADM 권한을 가진 사용자는 다음 권한을 가진 사용자의 기능도 수행할 수 있습니다.

- 『시스템 제어 권한(SYSCTRL)』
- 278 페이지의 『시스템 유지보수 권한(SYSMAINT)』
- 279 페이지의 『데이터베이스 관리 권한(DBADM)』

주: SYSADM 권한을 가지고 있는 사용자가 데이터베이스를 작성할 때에는, 데이터베이스에 대한 명시적인 DBADM 권한이 사용자에게 자동으로 권한 부여됩니다. SYSADM 그룹에서 데이터베이스 작성자가 제거되고 그룹이 해당 데이터베이스에 DBADM으로 액세스하지 못하도록 하고자 할 경우, 명시적으로 이 DBADM 권한을 권한 취소해야 합니다.

시스템 제어 권한(SYSCTRL)

SYSCTRL 권한은 최상위 레벨의 시스템 제어 권한입니다. 이 권한은 데이터베이스 관리 프로그램 인스턴스와 데이터베이스에 대해 유지보수 및 유틸리티 조작을 수행할 수 있는 기능을 제공합니다. 이러한 조작은 시스템 자원에 영향을 미칠 수는 있으나, 데이터베이스 내의 데이터에 직접 액세스하지는 못합니다. 시스템 제어 권한은 사용자가 중요한 데이터가 들어 있는 데이터베이스 관리 프로그램 인스턴스를 관리하기 위한 목적으로 설계되었습니다.

SYSCTRL 권한은 *sysctrl_group* 구성 매개변수에서 지정한 그룹으로 지정됩니다 (관리 안내서: 성능에 있는 『DB2 구성』 참조). 그룹이 지정되면, 해당 그룹 내의 멤버십은 사용자의 플랫폼에서 사용되는 보안 기능을 통해 데이터베이스 관리 프로그램 외부에서 제어됩니다.

SYSCtrl 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 데이터베이스, 노드 또는 분산 연결 서비스(DCS) 디렉토리 갱신
- 사용자가 시스템을 강제로 끄도록 함
- 데이터베이스의 작성 또는 삭제
- 테이블 공간의 삭제, 작성 또는 변경
- 새로운 데이터베이스로 복원

또한, SYSCtrl 권한을 갖는 사용자는 『시스템 유지보수 권한(SYSMAINT)』 권한을 가진 사용자의 기능도 실행할 수 있습니다.

SYSCtrl 권한을 갖는 사용자는 데이터베이스에 연결할 수 있는 내재적인 특권도 갖게 됩니다.

주: SYSCtrl 권한을 가지고 있는 사용자가 데이터베이스를 작성할 때에는, 데이터베이스에 대한 명시적인 DBADM 권한이 사용자에게 자동으로 권한 부여됩니다. SYSCtrl 그룹에서 데이터베이스 작성자가 제거되고 그룹이 해당 데이터베이스에 DBADM으로 액세스하지 못하도록 하고자 할 경우, 명시적으로 이 DBADM 권한을 권한 취소해야 합니다.

시스템 유지보수 권한(SYSMAINT)

SYSMAINT 권한은 시스템 제어 권한 중 두 번째로 높은 레벨입니다. 이 권한은 데이터베이스 관리 프로그램 인스턴스와 데이터베이스에 대해 유지보수 및 유틸리티 조작을 수행할 수 있는 기능을 제공합니다. 이러한 조작은 시스템 자원에 영향을 미칠 수는 있으나, 데이터베이스 내의 데이터에 직접 액세스하지는 못합니다. 시스템 유지보수 권한은 사용자가 중요한 데이터가 들어 있는 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스를 유지보수하기 위한 목적으로 설계되었습니다.

SYSMAINT 권한은 *sysmaint_group* 구성 매개변수에서 지정한 그룹으로 지정됩니다(관리 안내서: 성능에있는 『DB2 구성』 참조). 그룹이 지정되면, 해당 그룹 내의 멤버십은 사용자의 플랫폼에서 사용되는 보안 기능을 통해 데이터베이스 관리 프로그램 외부에서 제어됩니다.

SYSMAINT 이상의 시스템 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 데이터베이스 구성 파일 갱신
- 데이터베이스 또는 테이블 공간 백업
- 기존의 데이터베이스로 복원
- 롤 포워드 복구의 수행
- 데이터베이스 인스턴스의 시작 또는 중지
- 테이블 공간 복원
- 추적 수행
- 데이터베이스 관리 프로그램 인스턴스 또는 해당 데이터베이스의 데이터베이스 시스템 모니터 스냅샷

SYSMANT, DBADM 또는 그 이상의 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 테이블 공간의 상태 조회
- 로그 실행기록 파일 갱신
- 테이블 공간 Quiesce
- 테이블 재구성
- RUNSTATS 유틸리티를 사용하여 카탈로그 통계 수집

SYSMANT 권한을 갖는 사용자는 데이터베이스에 연결할 수 있는 내재적인 특권도 갖게 됩니다.

데이터베이스 관리 권한(DBADM)

DBADM 권한은 관리 권한 중 두번째로 높은 레벨입니다. 이것은 특정 데이터베이스에만 적용되며, 사용자가 특정 유틸리티를 실행하고, 데이터베이스 명령을 발행하고, 데이터베이스 테이블의 데이터에 액세스할 수 있도록 합니다. DBADM 권한이 권한 부여되면, BINDADD, CONNECT, CREATETAB, CREATE_NOT_FENCED 및 IMPLICIT_SCHEMA 특권도 권한 부여됩니다. SYSADM 권한을 가지고 있는 사용자만이 DBADM 권한을 권한 부여하거나 권한 취소할 수 있습니다. DBADM 권한을 가지고 있는 사용자는 다른 사용자에게 데이터베이스에 대한 특권을 권한 부여할 수 있고, 누가 특권을 권한 부여했는지에 관계없이 사용자에게서 임의의 특권을 권한 취소할 수 있습니다.

DBADM 이상의 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 로그 파일 읽기
- 이벤트 모니터 작성, 활성화 및 삭제

DBADM, SYSMANT 이상의 권한을 가진 사용자는 다음을 수행할 수 있습니다.

- 테이블 공간의 상태 조회
- 로그 실행기록 파일 갱신
- 테이블 공간 Quiesce
- 테이블 재구성
- RUNSTATS 유틸리티를 사용하여 카탈로그 통계 수집

주: DBADM은 DBADM 권한이 보유된 데이터베이스에 대해서만 위의 기능을 수행할 수 있습니다.

LOAD 권한

테이블에서 INSERT 특권은 물론, 데이터베이스 레벨에서 LOAD 권한을 가지는 사용자는 LOAD 명령이나 AutoLoader 유틸리티를 사용하여 데이터를 테이블로 로드시킬 수 있습니다.

테이블에서 INSERT 특권은 물론, 데이터베이스 레벨에서 LOAD 권한을 가지는 사용자는 이전 로드 조작이 데이터를 삽입하기 위해 로드된 경우 LOAD RESTART 또는 LOAD TERMINATE할 수 있습니다.

이전 로드 조작이 로드 바꾸기인 경우, 사용자가 LOAD RESTART 또는 LOAD TERMINATE하기 전에 DELETE 특권이 부여되어야 합니다.

예외 테이블이 LOAD의 일부로서 사용되는 경우, 사용자는 예외 테이블에 INSERT 특권을 가져야 합니다.

이 권한이 있는 사용자는 QUIESCE TABLESPACES FOR TABLE, RUNSTATS 및 LIST TABLESPACES 명령을 수행할 수 있습니다.

데이터베이스 특권

그림5에서는 데이터베이스 특권을 보여줍니다.

데이터베이스 특권

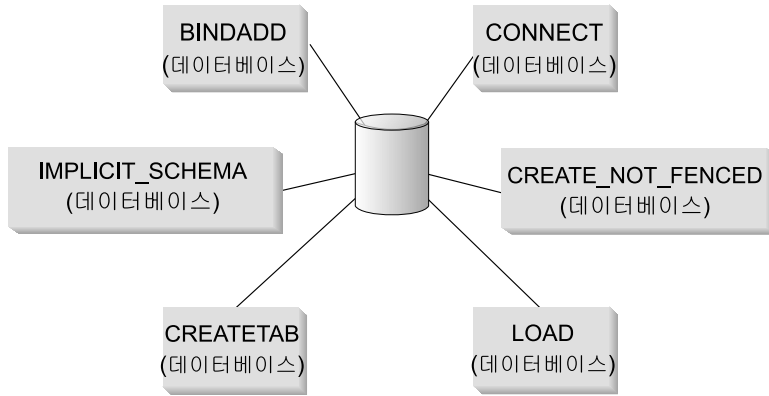


그림 5. 데이터베이스 특권

데이터베이스 특권에는 데이터베이스에 대한 조치가 전체적으로 포함됩니다.

- CONNECT는 사용자가 데이터베이스에 액세스할 수 있도록 합니다.
- BINDADD는 사용자가 데이터베이스에 새로운 패키지를 작성할 수 있도록 합니다.
- CREATETAB는 사용자가 데이터베이스에 새로운 테이블을 작성할 수 있도록 합니다.
- CREATE_NOT_FENCED에서 사용자는 사용자 정의 함수(UDF) 또는 『비분리(non-fenced)』 프로시저를 작성할 수 있습니다. 『비분리(non-fenced)』 UDF 또는 프로시저는 데이터베이스 관리 프로그램이 이러한 UDF 또는 프로시저로부터의 저장 영역 또는 제어 블록을 보호하지 않기 때문에 세심하게 테스트되어야 합니다(결과적으로, 『비분리(non-fenced)』를 수행하도록 허용된 UDF 또는 프로시저가 완벽하게 작성되어 있지 않거나 테스트되지 않으면, 시스템에 심각한 문제가 발생할 수 있습니다). (응용프로그램 개발 안내서 또는 SQL 참조서에서 자세한 내용을 참조하십시오)
- IMPLICIT_SCHEMA는 모든 사용자가 아직 존재하지 않는 스키마 이름과 함께 CREATE문을 사용하여 오브젝트를 작성함으로써 스키마를 내재적으로 작성할 수 있도록 합니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.
- LOAD는 사용자가 데이터를 테이블로 로드하게 합니다.

SYSADM 또는 DBADM 권한을 가지고 있는 사용자만이 다른 사용자에게 이러한 특권을 권한 부여하고 권한 취소할 수 있습니다.

주: 데이터베이스가 작성되면, 다음 특권이 자동으로 PUBLIC에 권한 부여됩니다.

- CREATETAB
- BINDADD
- CONNECT
- IMPLICIT_SCHEMA
- USERSPACE1 테이블 공간에서의 USE 특권
- 시스템 카탈로그 뷰에 대한 SELECT 특권

어떤 특권을 제거하려면, DBADM 또는 SYSADM은 PUBLIC으로부터 명시적으로 특권을 권한 취소해야 합니다.

내재된 스키마 권한(IMPLICIT_SCHEMA)에 대한 고려사항

새로운 데이터베이스가 작성되거나 데이터베이스가 이전 릴리스로부터 이주될 경우, PUBLIC에는 IMPLICIT_SCHEMA 데이터베이스 권한이 부여됩니다. 이 권한이 있는 모든 사용자는 오브젝트를 작성하고 아직 존재하지 않는 스키마 이름을 지정함으로써 스키마를 작성할 수 있습니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.

데이터베이스에서 스키마 오브젝트를 내재적으로 작성할 수 있는 사용자를 제어할 필요가 있을 경우, IMPLICIT_SCHEMA 데이터베이스 권한을 PUBLIC에서 권한 취소해야 합니다. 일단 이렇게 되면, 스키마 오브젝트를 작성할 수 있는 방법은 다음 세가지 밖에 없습니다.

- 모든 사용자는 CREATE SCHEMA문에 자신의 권한 부여 이름을 사용하여 스키마를 작성할 수 있습니다.
- DBADM 권한을 갖는 모든 사용자는 아직 존재하지 않는 모든 스키마를 명시적으로 작성할 수 있으며, 다른 사용자를 스키마 소유자로 선택적으로 지정할 수 있습니다.
- DBADM 권한이 있는 모든 사용자는 IMPLICIT_SCHEMA 데이터베이스 권한이 있어(PUBLIC과 별도로), 다른 데이터베이스 오브젝트를 작성할 때 어떠한

한 이름으로도 스키마를 내재적으로 작성할 수 있습니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 스키마에서 오브젝트를 작성할 수 있는 특권이 부여됩니다.

사용자는 항상 자신의 권한 부여 이름을 사용하여 자신의 스키마를 명시적으로 작성할 수 있는 기능을 가집니다.

스키마 특권

스키마 특권은 오브젝트 특권 범주 내에 있습니다. 오브젝트 특권은 그림6에 나와 있습니다.

오브젝트 특권

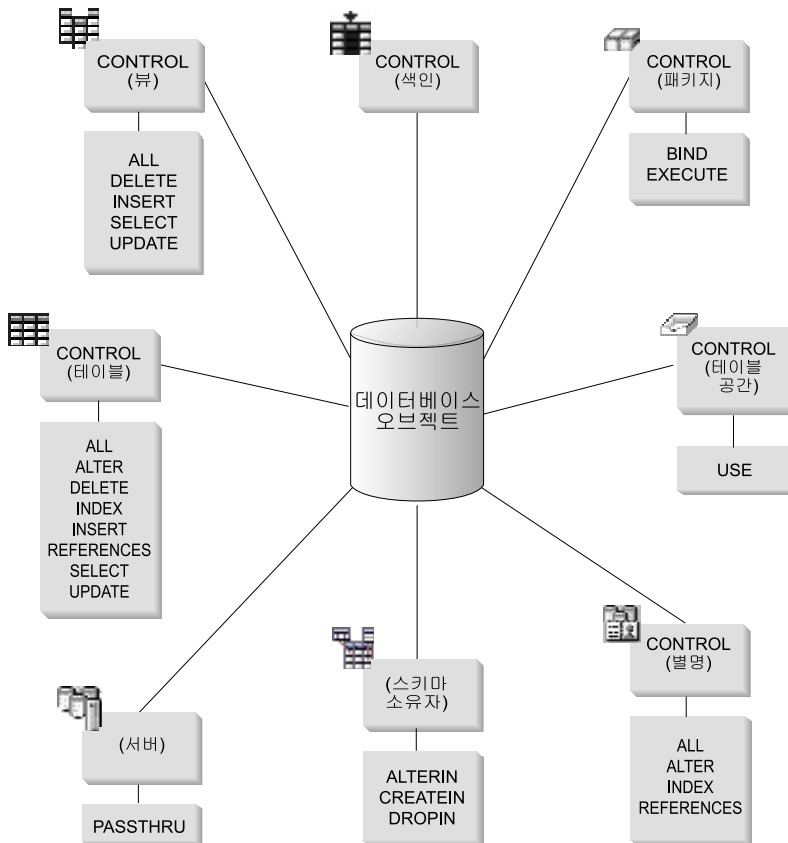


그림 6. 오브젝트 특권

스키마 권한은 데이터베이스의 스키마에 조치를 취할 수 있는 권한입니다. 사용자에게는 다음 권한이 권한 부여됩니다.

- CREATEIN은 사용자가 스키마 내에서 오브젝트를 작성하도록 합니다.
- ALTERIN은 사용자가 스키마 내에서 오브젝트를 변경하도록 합니다.
- DROPIN은 사용자가 스키마 내에서 오브젝트를 삭제하도록 합니다.

스키마 소유자는 모두 이러한 특권과 기능을 다른 사용자에게 권한 부여할 권한을 가집니다. 스키마 오브젝트 안에서 조정된 오브젝트에는 테이블, 뷰, 색인, 패키지, 데이터 유형, 함수, 트리거, 프로시저, 별명이 있습니다.

테이블 공간 특권

테이블 공간 특권은 데이터베이스에서 테이블 공간에 대한 조치를 포함합니다. 사용자는 테이블 공간 내에서 테이블을 작성하도록 허용하는 테이블 공간에 대한 USE 특권을 부여할 수 있습니다.

테이블 공간의 소유자는 보통 SYSADM 또는 SYSCTRL 권한을 가진 작성자로 USE 특권과 이 특권을 다른 사람에게 부여하는 능력을 갖습니다. 기본적으로, 데이터베이스 작성 시간에 테이블 공간 USERSPACE1에 대한 USE 특권이 PUBLIC으로 부여되지만, 이 특권은 권한 취소될 수 있습니다.

USE 특권은 SYSCATSPACE나 임의의 시스템 임시 테이블 공간에서 사용될 수 없습니다.

테이블 및 뷰 특권

테이블 및 뷰 특권에는 데이터베이스 내의 테이블 또는 뷰에 대한 조치가 포함됩니다. 사용자는 데이터베이스에 대해 CONNECT 특권을 가져야 다음과 같은 특권을 사용합니다.

- CONTROL은 테이블 또는 뷰를 삭제할 권한을 포함하여 테이블 또는 뷰에 대한 모든 특권을 제공하고, 개별 테이블 특권을 권한 부여하거나 권한 취소합니다. CONTROL을 권한 부여하려면 SYSADM 또는 DBADM 권한이 있어야 합니다. 테이블의 작성자는 테이블에 대해 자동으로 CONTROL 특권을 가지게 됩니다. 뷰의 작성자는 뷰 정의에서 언급된 모든 테이블 및 뷰에 대해 작성자가 CONTROL 특권을 가지고 있거나 SYSADM 또는 DBADM 권한을 가지고 있는 경우에만, 자동으로 CONTROL 특권을 부여받습니다.

- ALTER를 사용하면 사용자는 컬럼을 테이블에 추가하거나 테이블의 주석과 해당 컬럼을 추가하며, 기본 키 또는 고유 제한조건을 추가하고, 테이블 점검 제한조건을 작성하거나 삭제할 수 있습니다. 사용자는 또한 트리거에서 참조된 모든 오브젝트에 대해(트리거가 테이블의 컬럼을 참조할 경우에는 테이블에 대한 SELECT 포함) 추가 권한이 필요하더라도, 테이블에 대해 트리거를 작성할 수 있습니다. 모든 하위 테이블에 대해 ALTER 특권을 가진 사용자는 기본 키를 삭제할 수 있고, 테이블에 대해 ALTER 특권을 가지고 상위 테이블에 대해 REFERENCES 특권을 가지고 있거나 해당 컬럼에 대해 REFERENCES 특권을 가진 사용자는 외부 키를 작성하거나 삭제할 수 있습니다. ALTER 특권을 갖는 사용자는 테이블에서 COMMENT ON을 수행할 수 있습니다.
- DELETE는 사용자가 테이블 또는 뷰에서 행을 삭제할 수 있도록 합니다.
- INDEX는 사용자가 데이터베이스에 색인을 작성할 수 있도록 합니다. 색인의 작성자는 색인에 대해 자동으로 CONTROL 특권을 가지게 됩니다. 290 페이지의 『색인 특권』에서 자세한 내용을 참조하십시오.
- INSERT는 사용자가 테이블 또는 뷰에 항목을 삽입하고 IMPORT 유틸리티를 수행할 수 있도록 합니다.
- REFERENCES는 사용자가 테이블을 관계의 상위로 지정하여, 외부 키를 작성하거나 삭제할 수 있도록 합니다. 사용자는 특정 컬럼에 대해서만 이 특권을 가질 수도 있습니다.
- SELECT는 사용자가 테이블 또는 뷰에서 행을 검색하고, 테이블에 뷰를 작성하고, EXPORT 유틸리티를 수행할 수 있도록 합니다.
- UPDATE는 사용자가 테이블의 한 항목, 뷰 또는 테이블이나 뷰의 하나 이상의 특정 컬럼을 변경할 수 있도록 합니다. 사용자는 특정 컬럼에서만 이 특권을 가질 수 있습니다.

이러한 특권을 다른 사용자에게 권한 부여하는 특권은 GRANT문에서 WITH GRANT OPTION을 사용하여 권한 부여될 수도 있습니다.

주: 사용자 또는 그룹에 테이블에 대한 CONTROL 특권이 권한 부여될 경우, 해당 테이블의 다른 모든 특권에는 자동으로 WITH GRANT OPTION이 권한 부여됩니다. 나중에 사용자에게서 테이블에 대한 CONTROL 특권을 권한 취소해도, 사용자는 자동으로 권한 부여된 다른 특권을 계속 보유하게 됩니다.

CONTRL 특권으로 권한 부여된 모든 특권을 권한 취소하려면, 각 특권을 명시적으로 권한 취소하거나, REVOKE문에서 ALL 키워드를 지정해야 합니다. 예를 들면, 다음과 같습니다.

```
REVOKE ALL
ON EMPLOYEE FROM USER HERON
```

입력된 테이블에 대한 작업을 하는 경우, 테이블 및 뷰 특권과 관련하여 고려할 사항이 있습니다.

주: 특권은 테이블 계층의 모든 레벨에서 별도로 권한 부여되어야 합니다. 그 결과, 입력된 테이블 내의 상위 테이블에서 특권을 권한 부여받은 사용자는 하위 테이블에도 간접적으로 영향을 줄 수 있습니다. 그러나, 해당 하위 테이블에 필수 특권이 설정되어 있는 경우, 사용자는 하위 테이블에 직접적으로 조작을 수행해야 합니다.

테이블 계층 내 테이블간의 상위 테이블/하위 테이블 관계는 SELECT, UPDATE 및 DELETE와 같은 조장이 해당 조장의 목표 테이블 및 모든 해당 하위 테이블(있는 경우)에 영향을 줄 것임을 의미합니다. 이러한 작용을 『대체 가능성』이라 부를 수 있습니다. 예를 들어, 유형이 Manager_t인 하위 테이블 Manager로 유형이 Employee_t인 Employee 테이블을 작성했다고 가정합니다. 관리자도 사원의 일부이므로, 구조화 유형 Employee_t 및 Manager_t간의 유형/부속유형 관계와, 테이블 Employee 및 Manager간의 해당 테이블/하위 테이블 관계로 표시됩니다. 이 관계의 결과 다음 SQL 조회

```
SELECT * FROM Employee
```

는 사원과 관리자 모두에 대한 오브젝트 식별자 및 Employee_t 속성을 리턴합니다. 이와 마찬가지로, 다음 갱신 조작

```
UPDATE Employee SET Salary = Salary + 1000
```

은 정규 사원뿐 아니라 관리자의 급여를 1000달러 인상합니다.

Employee의 SELECT 특권을 가진 사용자는 Manager에 대한 명시적인 SELECT 특권을 가지고 있지 않더라도 이 SELECT 조작을 수행할 수 있습니다. 그러나, 그러한 사용자는 Manage 하위 테이블에 직접 SELECT 조작을 수행할 수 없기 때문에, Manager 테이블의 계승되지 않은 컬럼에는 액세스할 수 없습니다.

마찬가지로, Employee에 대한 UPDATE 특권을 가진 사용자는 Employee에 대해 UPDATE 조작을 수행할 수 있기 때문에, Manager 테이블에 대한 명시적 UPDATE 특권을 가지고 있지 않더라도 정규 사원과 관리자 모두에게 영향을 줍니다. 그러나, 그러한 사용자는 Manage 하위 테이블에 직접 UPDATE 조작을 수행할 수 없기 때문에, Manager 테이블의 계승되지 않은 컬럼은 갱신할 수 없습니다.

다음 매뉴얼에서는 특정 명령, API 또는 SQL문을 실행하는 데 필요한 권한 부여에 대한 정보를 제공합니다.

- *SQL 참조서*
- *Command Reference*
- *Administrative API Reference.*

카탈로그 통계 갱신에 필요한 권한 부여에 대해서는 *관리 안내서: 성능에서 자세한 내용을 참조하십시오.*

뷰 특권이 결정되는 방법에 대해서는 *SQL 참조서 매뉴얼의 CREATE VIEW문을 참조하십시오.*

별명 특권

별명 특권은 데이터베이스에 있는 별명에 대해 조치를 취할 수 있는 특권입니다. 이러한 특권은 별명이 참조하는 데이터 소스 오브젝트에 대한 특권에 영향을 주지 않습니다. 사용자는 데이터베이스에 대해 CONNECT 특권을 가져야 다음과 같은 특권을 사용합니다.

- CONTROL은 사용자에게 별명 특권을 삭제하고, 각 별명 특권을 권한 부여 및 권한 취소할 수 있는 별명에 대한 모든 특권을 제공합니다. CONTROL을 권한 부여하려면 SYSADM 또는 DBADM 권한이 있어야 합니다. 별명 작성자는 자동으로 별명에 대해 CONTROL 특권을 가지게 됩니다.
- ALTER를 사용하면 별명에서 컬럼 이름을 변경하고, 컬럼의 데이터 유형이 맵핑되는 DB2 유형을 추가 또는 변경하며, 별명 컬럼에 대한 컬럼 옵션을 설정할 수 있습니다.
- INDEX를 사용하면 별명에서 색인 스펙을 작성할 수 있습니다. 색인 스펙 작성자는 자동으로 색인에 대해 CONTROL 특권을 갖습니다.

- REFERENCES를 사용하면 관계에서 상위로 별명을 지정하여 외부 키를 작성 및 삭제할 수 있습니다. 사용자는 특정 컬럼에서만 이 특권을 가질 수 있습니다.

이러한 특권을 다른 사용자에게 권한 부여하는 특권은 GRANT문에서 WITH GRANT OPTION을 사용하여 권한 부여될 수도 있습니다.

주: 사용자 또는 그룹에 별명에 대한 CONTROL 특권이 권한 부여되면, 해당 별명에 대한 다른 모든 특권은 WITH GRANT OPTION으로 자동 권한 부여됩니다. 나중에 별명에 대한 CONTROL 특권을 권한 취소해도, 사용자는 자동으로 권한 부여된 다른 특권을 계속 보유합니다.

데이터 소스에 액세스하려면, 별명이 참조하는 데이터 소스의 오브젝트에 대해 적합한 권한을 가지고 있어야 합니다.

사용자가 하나 이상의 별명을 참조하는 뷰에 액세스하는 경우, 사용자는 데이터 소스에서 별명이 참조하는 뷰와 오브젝트에 액세스할 수 있는 권한을 가지고 있어야 합니다.

서버 특권

하나의 서버 특권 PASSTHRU가 있습니다. 이 특권은 DDL 및 DML문을 데이터 소스로 직접 발행할 수 있는(통과 조작) 권한 부여 ID를 제어합니다.

DB2는 두 개의 SQL문을 제공하여 통과 조작을 제어합니다.

- GRANT PASSTHRU, 데이터 소스에 대해 SET PASSTHRU문을 발행하고 해당 데이터 소스로 DML 및 DDL문을 통과하도록 권한을 권한 부여합니다.
- REVOKE PASSTHRU, 데이터 소스에 대해 SET PASSTHRU문을 발행하고 해당 데이터 소스로 DML 및 DDL문을 통과하도록 하는 권한을 권한 취소합니다.

ORACLE1 서버에 대한 사용자 SHAWN으로 통과 권한을 권한 부여하는 샘플 문은 다음과 같습니다.

```
GRANT PASSTHRU ON SERVER ORACLE1 TO USER SHAWN
```

PASSTHRU문 구문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

패키지 특권

패키지란 특정 응용프로그램에 대해 가장 효과적인 방법으로 데이터에 액세스하기 위해 데이터베이스 관리 프로그램이 필요로 하는 정보가 들어 있는 데이터베이스 오브젝트입니다. 패키지 특권은 사용자가 패키지를 작성하고 조작할 수 있도록 합니다. 사용자는 다음 특권을 사용하기 위해 데이터베이스에 대해 CONNECT 특권을 가지고 있어야 합니다.

- CONTROL은 패키지를 리바인드, 삭제 또는 실행할 능력과 이러한 특권을 다른 사람에게 확대할 수 있는 능력을 제공합니다. 패키지의 작성자는 자동으로 이 특권을 받게 됩니다. CONTROL 특권을 가지고 있는 사용자는 BIND, EXECUTE 특권을 권한 부여받고, 다른 사용자에게도 BIND 및 EXECUTE 특권을 권한 부여할 수 있습니다. CONTROL 특권을 권한 부여하려면, 사용자에게 SYSADM 또는 DBADM 권한이 있어야 합니다.
- BIND는 사용자가 기존 패키지를 리바인드할 수 있도록 합니다.
- EXECUTE는 사용자가 패키지를 실행할 수 있도록 합니다.

이러한 패키지 특권 외에도, 사용자는 BINDADD 데이터베이스 특권을 사용하여 데이터베이스에 새로운 패키지를 작성하거나 기존의 패키지를 리바인드할 수 있습니다.

별명이 들어 있는 패키지를 실행할 권한을 가진 사용자들은 패키지 내의 별명에 대해 추가 특권 또는 권한 레벨이 필요하지 않습니다. 그러나, 별명이 참조하는 오브젝트가 들어 있는 데이터 소스에서 인증 점검을 통과해야 합니다. 뿐만 아니라, 패키지 사용자는 데이터 소스에 있는 데이터 소스 오브젝트에 대해 적합한 특권 또는 권한 레벨을 가지고 있어야 합니다.

DB2가 DB2 계열 데이터 소스와 통신할 때 동적 SQL을 사용하므로 별명이 들어 있는 패키지에서 추가 권한 부여 단계가 필요할 수도 있습니다. 데이터 소스에서 패키지를 수행하는 권한 부여 ID는 해당데이터 소스에서 동적으로 패키지를 실행할 권한을 가지고 있어야 합니다. DB2가 정적 및 동적 SQL을 처리하는 방법에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

색인 특권

색인 또는 색인 스펙 작성자는 색인에 대해 자동으로 CONTROL 특권을 갖게 됩니다. 색인에 대한 CONTROL 특권은 색인을 삭제할 권한이라 할 수 있습니다. 색인에 대한 CONTROL 특권을 권한 부여하려면, 사용자에게 SYSADM 또는 DBADM 권한이 있어야 합니다.

사용자는 테이블 레벨 INDEX 특권으로 해당 테이블에 대한 색인을 작성할 수 있습니다(284 페이지의 『테이블 및 뷰 특권』 참조).

데이터베이스 오브젝트에 대한 액세스 제어

데이터 액세스를 제어하려면 직접 및 간접 특권, 관리자 권한, 패키지에 대해 이해해야 합니다. 이 절에서는 이러한 주제에 대해 설명하고 몇 가지 예도 제공합니다.

직접적으로 권한 부여된 특권은 시스템 카탈로그에 저장됩니다. 데이터베이스 액세스 제어 계획의 구현을 감사하는 방법에 대해서는 302 페이지의 『시스템 카탈로그 사용』에서 설명됩니다.

권한 부여는 다음 세 가지 방식으로 제어됩니다.

- 명시적 권한 부여는 GRANT 및 REVOKE문으로 제어되는 특권을 통해 제어됩니다.
- 내재적 권한 부여는 오브젝트를 작성 및 삭제함으로써 제어됩니다.
- 간접 특권은 패키지와 연관됩니다.

이 절에는 다음과 같은 내용이 들어 있습니다.

- 291 페이지의 『특권 권한 부여』
- 292 페이지의 『특권 취소』
- 294 페이지의 『오브젝트 작성 및 삭제에 의한 내재적 권한 부여 관리』
- 295 페이지의 『패키지를 통한 간접 특권 허용』
- 296 페이지의 『뷰를 사용하여 데이터에 대한 액세스 제어』
- 300 페이지의 『감사 기능을 사용하여 데이터에 대한 액세스 모니터링』.

특권 권한 부여

GRANT문은 권한 부여된 사용자가 특권을 권한 부여할 수 있도록 합니다. 특권을 하나의 명령문으로 하나 이상의 권한 부여 이름에 권한 부여하거나 PUBLIC에 권한 부여하여, 모든 사용자가 특권을 사용하게 할 수 있습니다. 권한 부여 이름은 개별 사용자 또는 그룹이 될 수 있습니다.

사용자와 그룹이 동일한 이름으로 존재하는 운영 체제에서는, 사용자 또는 그룹에 특권을 권한 부여할 것인지 여부를 지정해야 합니다. GRANT 및 REVOKE문은 둘다 키워드 USER 및 GROUP을 지원합니다. 이들 선택적 키워드가 사용되지 않는 경우에는, 데이터베이스 관리 프로그램이 운영 체제 보안 기능을 검사하여 권한 부여 이름이 사용자 또는 그룹 식별 여부를 판별합니다. 권한 부여 이름이 사용자 또는 그룹 모두인 경우에는, 오류가 리턴됩니다.

다음 예에서는 HERON 사용자에게 EMPLOYEE 테이블에 대한 SELECT 특권을 권한 부여합니다.

```
GRANT SELECT
ON EMPLOYEE TO USER HERON
```

다음 예에서는 HERON 그룹에게 EMPLOYEE 테이블에 대한 SELECT 특권을 권한 부여합니다.

```
GRANT SELECT
ON EMPLOYEE TO GROUP HERON
```

대부분의 데이터베이스 오브젝트에 특권을 권한 부여하려면, 사용자는 해당 오브젝트에 대해 SYSADM 권한, DBADM 권한 또는 CONTROL 특권을 가지고 있거나, WITH GRANT OPTION 특권을 가지고 있어야 합니다. 특권은 기존의 오브젝트에 대해서만 권한 부여될 수 있습니다. 그 밖의 다른 사용자에게 CONTROL 특권을 권한 부여하려면, 사용자에게 SYSADM 또는 DBADM 권한이 있어야 합니다. DBADM 권한을 권한 부여하려면, 사용자에게 SYSADM 권한이 있어야 합니다.

GRANT문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

특권 취소

REVOKE문은 권한 부여된 사용자가 다른 사용자에게 이미 권한 부여된 특권을 권한 취소할 수 있도록 합니다. 데이터베이스 오브젝트에 대한 특권을 권한 취소하려면, 해당 오브젝트에 대한 DBADM 권한, SYSADM 권한 또는 CONTROL 특권을 가지고 있어야 합니다. WITH GRANT OPTION 특권만으로는 특권을 권한 취소할 수 없다는 점을 유의하십시오. 다른 사용자로부터 CONTROL 특권을 권한 취소하려면, SYSADM 또는 DBADM 권한을 가지고 있어야 합니다. DBADM 권한을 권한 취소하려면, SYSADM 권한을 가지고 있어야 합니다. 특권은 기존 오브젝트에 대해서만 권한 취소될 수 있습니다.

주: 테이블이나 뷰에 대한 DBADM 권한 또는 CONTROL 특권을 가지고 있지 않은 사용자는 WITH GRANT OPTION을 사용하여 권한 부여한 특권을 권한 취소할 수 없습니다. 또한, 권한 취소된 사용자에게 특권을 권한 부여 받은 사용자의 권한을 연쇄적으로 권한 취소할 수 없습니다. 특권을 권한 취소하는데 필요한 권한에 관한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

특권이 동일한 이름을 갖는 사용자와 그룹에 모두 권한 부여되면, 특권을 권한 취소할 때 GROUP 또는 USER 키워드를 지정해야 합니다. 다음 예에서는 사용자 HERON에게서 EMPLOYEE 테이블의 SELECT 특권을 취소합니다.

```
REVOKE SELECT
ON EMPLOYEE FROM USER HERON
```

다음 예에서는 그룹 HERON에게서 EMPLOYEE 테이블의 SELECT 특권을 권한 취소합니다.

```
REVOKE SELECT
ON EMPLOYEE FROM GROUP HERON
```

그룹에서의 특권 권한 취소가 해당 그룹의 모든 구성원에서 특권을 권한 취소하는 것이 아님에 유의하십시오. 특권을 개별 이름에 직접 권한 부여한 경우에는, 해당 특권이 직접 권한 취소될 때까지 이름을 유지합니다.

테이블 특권이 사용자에게서 권한 취소되면, 해당 사용자가 작성한 뷰에 대한 특권도 권한 취소된 테이블 특권에 종속하는 것이므로 권한 취소됩니다. 그러나, 시스템에 의해 권한 부여된 특권만이 권한 취소됩니다. 다른 사용자에 의해 뷰에 대한 특권을 직접 권한 부여 받은 경우에는, 특권이 계속 보유됩니다.

명시적으로 권한 부여된 테이블(또는 뷰) 특권이 DBADM 권한을 갖는 사용자로부터 권한 취소된 경우에는, 해당 테이블에 대해 정의된 다른 뷰에서 특권을 권한 취소할 수 없습니다. 이는 뷰 특권이 DBDAM 권한을 통해 사용 가능하고, 기초가 되는 테이블에 대한 명시적인 특권에 종속되지 않기 때문입니다.

하나 이상의 기초 테이블 또는 뷰에 근거하여 뷰를 정의한 경우와, 하나 이상의 테이블 또는 뷰에 대한 SELECT 특권이 없으면, 이 뷰는 사용될 수 없습니다.

주: 테이블 또는 뷰의 사용자가 CONTROL 특권을 권한 취소하더라도, 사용자는 여전히 특권을 다른 사용자에게 권한 부여할 수 있는 권한을 지닙니다. 사용자에게 CONTROL 특권이 주어질 경우, 다른 모든 특권 WITH GRANT OPTION도 권한 부여됩니다. 일단 CONTROL이 권한 취소되면, 다른 모든 특권이 명시적으로 권한 취소될 때까지 WITH GRANT OPTION이 남아 있습니다.

권한 취소된 특권에 종속적인 모든 패키지는 무효로 표시되지만, 해당 권한을 갖는 사용자에 의해 리바인드되면 다시 유효화될 수 있습니다. 특권이 연속적으로 응용프로그램 바인더에 다시 권한 부여될 경우 패키지도 다시 구축될 수 있으며, 응용프로그램을 수행하면 내재적 리바인드가 성공적으로 트리거됩니다. 특권이 PUBLIC으로부터 권한 취소되면, PUBLIC 특권에 근거해야만 바인드 가능한 사용자가 바인드한 모든 패키지는 무효화됩니다. DBADM 권한이 사용자로부터 권한 취소되면, 데이터베이스 유틸리티와 연관된 패키지를 포함하여 해당 사용자가 바인드한 모든 패키지가 무효가 됩니다. 무효로 표시된 패키지를 사용하려고 시도하면 시스템이 패키지를 리바인드하려고 합니다. 이러한 리바인드 시도가 실패할 경우, 오류가 발생합니다(SQLCODE-727). 이 경우에는, 다음과 같은 권한을 갖는 사용자가 패키지를 명시적으로 리바인드해야 합니다.

- 패키지를 리바인드할 수 있는 권한
- 패키지에서 사용한 오브젝트에 대한 해당 권한

이들 패키지는 특권이 권한 취소될 때마다 리바인드되어야 합니다. REVOKE 및 REBIND PACKAGE문에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

하나 이상의 특권에 근거하는 트리거를 정의한 경우와, 하나 이상의 해당 특권이 없으면, 트리거는 사용될 수 없습니다.

오브젝트 작성 및 삭제에 의한 내재적 권한 부여 관리

데이터베이스 관리 프로그램은 CREATE SCHEMA, CREATE TABLESPACE, CREATE VIEW 또는 CREATE INDEX문을 나타내는 사용자 또는 PREP 명령이나 BIND 명령을 사용하여 새로운 패키지를 작성하는 사용자에게 임의의 특권을 내재적으로 권한 부여합니다. 또한, SYSADM 또는 DBADM 권한을 갖는 사용자가 오브젝트를 작성할 때에도 특권이 권한 부여됩니다. 마찬가지로, 특권은 오브젝트가 삭제될 때 삭제됩니다.

작성된 오브젝트가 테이블 공간, 테이블, 색인 또는 패키지인 경우, 사용자는 오브젝트에 대한 CONTROL 특권을 받게 됩니다. 오브젝트가 뷰이면, 뷰 정의에서 언급된 모든 테이블과 뷰에 대해 사용자가 CONTROL 특권을 가지고 있는 경우에만, 뷰에 대한 CONTROL 특권이 내재적으로 권한 부여됩니다.

명시적으로 작성된 오브젝트가 스키마일 경우, 스키마 소유자에게는 ALTERIN, CREATEIN 및 DROPIN 특권 WITH GRANT OPTION이 부여됩니다. 내재적으로 작성된 스키마는 PUBLIC에 권한 부여된 CREATEIN을 갖습니다.

뷰 특권이 결정되는 방법에 대해서는 SQL 참조서 매뉴얼의 CREATE VIEW문을 참조하십시오.

플랜 또는 패키지의 소유권 설정

BIND 및 PRECOMPILE 명령은 응용프로그램 패키지를 작성하고 변경합니다. 어느 경우에서든, OWNER 옵션을 사용하여 발생한 패키지의 소유자 이름을 지정하십시오. 패키지 소유권의 이름 지정을 위한 간단한 규칙이 있습니다.

- 사용자는 자신을 소유자로서 지정할 수 있습니다. OWNER 옵션이 지정되지 않은 경우 이것이 기본값입니다.
- SYSADM 또는 DBADM 권한을 가진 ID는 OWNER 옵션을 사용하여 권한 부여 ID를 소유자로서 지정할 수 있습니다.

DB2 데이터베이스 제품을 사용하여 패키지를 바인드할 수 있는 모든 운영 체제가 OWNER 옵션을 지원하는 것은 아닙니다.

BIND 및 PRECOMPILE 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

패키지를 통한 간접 특권 허용

데이터베이스 내의 데이터에 대한 액세스는 대화식 워크스테이션 세션에서 작업하고 있는 사용자뿐만 아니라, 응용프로그램에서도 요구할 수 있습니다. 패키지에는 사용자가 수 많은 데이터베이스 오브젝트에 대해 다양한 조치를 수행할 수 있도록 하는 명령문이 들어 있습니다. 이 각각의 조치에는 하나 이상의 특권이 필요합니다.

패키지를 바인드하는 개인 및 PUBLIC에 권한 부여된 권한은 정적 SQL이 바인드될 때 권한 점검용으로 사용됩니다. 그룹을 통해 권한 부여된 특권은 정적 SQL이 바인드될 때 권한 부여 검사용으로 사용되지 않습니다. 패키지를 바인드하는 유효한 *authID*는 패키지에서 정적 SQL문을 실행하도록 요청된 모든 특권이 명시적으로 부여되거나 패키지를 바인딩할 때 VALIDATE RUN이 지정되는 경우를 제외하고는 PUBLIC을 통해 필요한 특권이 내재적으로 부여되어야 합니다. VALIDATE RUN이 BIND 때 지정된 경우, 이 패키지 내에서 정적 SQL문에 대한 모든 권한 부여 실패는 BIND를 실패하도록 야기하지 않으며 그러한 SQL문은 런타임에서 재검증됩니다. PUBLIC, 그룹, 사용자 특권은 사용자에게 패키지를 바인드하는 적합한 권한 부여(BIND 또는 BINDADD 특권)가 있는지를 검사할 때 모두 사용됩니다.

패키지에는 정적 및 동적 SQL이 모두 포함됩니다. 정적 SQL로 패키지를 처리하려면, 사용자가 패키지에 대해 EXECUTE 특권만 가지고 있으면 됩니다. 그런 다음, 사용자는 패키지가 가지고 있는 제한 범위 내에서, 패키지에 있는 정적 SQL에 대해 패키지 바인더의 특권을 간접적으로 얻을 수 있습니다.

동적 SQL문으로 패키지를 처리하려면, 사용자가 패키지에 대해 EXECUTE 특권을 가지고 있어야 합니다. 사용자는 패키지에 대한 EXECUTE 특권과, 패키지에서 동적 SQL문을 실행하는 데 필요한 특권을 가지고 있어야 합니다. 바인더의 권한과 특권은 패키지의 모든 정적 SQL에 사용될 수 있습니다.

별명이 들어 있는 패키지를 통한 간접 특권 허용

패키지에 별명에 대한 참조가 들어 있으면, 패키지 작성자와 패키지 사용자에게 대한 권한 부여 처리는 다소 복잡합니다. 패키지 작성자가 별명이 들어 있는 패키지를 바인드하는 경우, 패키지 작성자는 데이터 소스에서 별명이 참조하는 테이블과

뷰에 대해 인증 점검 또는 특권 점검을 통과할 필요는 없습니다. 그러나, 패키지 실행자는 데이터 소스에서 인증 및 권한 부여 점검을 통과해야 합니다.

예를 들어, 패키지 작성자의 .SQL 파일에 여러 SQL문이 들어 있는 것으로 가정하십시오. 하나의 정적 명령문은 지역 테이블을 참조합니다. 또다른 동적 명령문은 별명을 참조합니다. 패키지가 바인드되면, 패키지 작성자의 인증 ID(authid)는 지역 테이블에 대한 특권을 확인하는 데 사용됩니다. 그러나, 별명이 식별하는 데이터 소스 오브젝트에 대해서는 어떠한 점검도 수행되지 않습니다. 다른 사용자가 패키지를 수행하면, 해당 패키지에 대해 EXECUTE 특권을 가지고 있고 사용자가 테이블을 참조하는 명령문에 대한 추가 특권 점검을 통과할 필요가 없는 것으로 가정합니다. 그러나 별명을 참조하는 명령문의 경우, 패키지를 실행하는 사용자는 데이터 소스에서 인증 점검 및 특권 점검을 통과해야 합니다.

.SQL 파일에 모든 동적 SQL문과, 테이블과 별명 참조의 혼합이 들어 있으면, 지역 오브젝트 및 별명에 대한 DB2 권한 부여 점검은 비슷합니다. 패키지 사용자는 명령문 내의 지역 오브젝트(테이블, 뷰)에 대한 특권 점검을 통과해야 하며, 또한 별명 오브젝트에 대한 특권 점검도 통과해야 합니다(패키지 사용자는 별명이 식별하는 오브젝트가 들어 있는 데이터 소스에서 인증 및 특권 점검을 통과해야 합니다). 어느 경우이든, 패키지 사용자는 EXECUTE 특권을 가지고 있어야 합니다.

패키지 실행자의 ID와 암호는 모든 데이터 소스 인증 및 특권 처리에 사용됩니다. 이 정보는 사용자 매핑을 작성하여 변경될 수 있습니다.

주: 별명은 정적 SQL에서 지정될 수 없습니다. 별명이 들어 있는 패키지와 함께 DYNAMICRULES 옵션(BIND로 설정된)을 사용하지 마십시오.

DB2가 DB2 계열 데이터 소스와 통신할 때 동적 SQL을 사용하므로 별명이 들어 있는 패키지에서 추가 권한 부여 단계가 필요할 수도 있습니다. 데이터 소스에서 패키지를 수행하는 권한 부여 ID는 해당 데이터 소스에서 동적으로 패키지를 실행할 권한을 가지고 있어야 합니다. DB2가 정적 및 동적 SQL을 처리하는 방법에 대해서는 SQL 참조서에서 자세한 내용을 참조하십시오.

뷰를 사용하여 데이터에 대한 액세스 제어

뷰는 다음 사항을 허용함으로써 테이블에 대한 액세스 제어 또는 특권 확장의 수단을 제공합니다.

- 테이블의 지정된 컬럼에 대해서만 액세스
테이블의 특정 컬럼에 대한 액세스만이 필요한 사용자와 응용프로그램의 경우에, 권한 부여된 사용자가 뷰를 작성하여 필요한 컬럼에만 주소지정되도록 컬럼을 제한할 수 있습니다.
- 테이블 행의 부속 집합만을 액세스
뷰 정의의 부속 조화에 WHERE절을 지정함으로써, 권한 부여된 사용자가 뷰를 통해 주소지정되는 행을 제한할 수 있습니다.
- 데이터 소스 테이블이나 뷰에 있는 행 또는 컬럼의 부속 집합에만 액세스. 별명을 통해 데이터 소스에 액세스하는 경우, 별명을 참조하는 지역 DB2 뷰를 작성할 수 있습니다. 이들 뷰는 하나 이상의 데이터 소스에서 별명을 참조할 수 있습니다.

주: 둘 이상의 데이터 소스에 대한 별명 참조사항이 들어 있는 뷰를 작성할 수 있으므로, 사용자는 하나의 뷰에서 여러 데이터 소스에 있는 데이터에 액세스할 수 있습니다. 이들 뷰를 다중 위치 뷰라고도 합니다. 이러한 뷰는 분산 환경을 통해 중요 테이블의 컬럼에서 정보를 조인할 때 또는 각 사용자에게 특정 오브젝트에 대한 데이터 소스에 필요한 특권이 부족할 때 유용합니다.

뷰를 작성하려면, 사용자는 뷰 정의에서 참조되는 각각의 테이블 또는 뷰에 대해 SYSADM 권한, DBADM 권한 또는 CONTROL 또는 SELECT 특권을 가지고 있어야 합니다. 또한, 사용자는 뷰에 대해 지정된 스키마에서 오브젝트를 작성할 권한도 가지고 있어야 합니다. 즉, 스키마가 아직 존재하지 않을 경우 기존 스키마에 대한 CREATEIN 특권 또는 데이터베이스에 대한 IMPLICIT_SCHEMA 권한이 있어야 합니다. 167 페이지의 『뷰 작성』에서 자세한 내용을 참조하십시오.

별명을 참조하는 뷰를 작성하는 경우, 뷰에서 별명이 참조하는 데이터 소스 오브젝트(테이블 및 뷰)에 대해 권한을 추가할 필요는 없습니다. 그러나, 사용자는 뷰에 액세스할 때 기초 데이터 소스 오브젝트에 대해 동등한 권한 부여 레벨 또는 SELECT 권한을 가지고 있어야 합니다.

사용자가 기초 오브젝트(테이블 및 뷰)에 대한 데이터 소스에서 적합한 권한을 가지고 있지 않으면, 다음을 수행할 수 있습니다.

- 데이터 소스 테이블에서 사용자가 액세스할 수 있는 컬럼에 대해 데이터 소스 뷰를 작성합니다.
- 이 뷰에 대한 SELECT 특권을 사용자에게 권한 부여합니다.
- 뷰를 참조하는 별명을 작성합니다.

그런 다음, 새로운 별명을 참조하는 SELECT문을 발행하여 컬럼에 액세스할 수 있습니다.

다음 시나리오에서는 뷰가 정보에 대한 액세스를 제한하기 위해 사용되는 방법의 상세한 예를 제공합니다.

많은 사용자가 서로 다른 이유로 STAFF 테이블 내의 정보를 액세스해야 할 경우가 있습니다. 예를 들면, 다음과 같습니다.

- 인사부에서는 전체 테이블을 갱신하고 볼 수 있어야 합니다.

이러한 요구는 STAFF 테이블에 대한 SELECT 및 UPDATE 특권을 그룹 PERSONNL에 권한 부여함으로써 쉽게 충족될 수 있습니다.

```
GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
```

- 부서 관리자는 각각 부하 직원에 대한 급여 정보를 보아야 합니다.

이러한 요구는 각 부서 관리자에 대한 뷰를 작성함으로써 충족될 수 있습니다. 예를 들어, 다음과 같은 뷰가 부서 번호 51의 관리자에 대해 작성될 수 있습니다.

```
CREATE VIEW EMP051 AS
  SELECT NAME,SALARY,JOB FROM STAFF
  WHERE DEPT=51
GRANT SELECT ON TABLE EMP051 TO JANE
```

권한 부여 이름 JANE을 갖는 관리자는 STAFF 테이블처럼 EMP051 뷰를 조회합니다. STAFF 테이블의 EMP051 뷰에 액세스하면, 이 관리자는 다음과 같은 정보를 볼 수 있습니다.

NAME	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- 모든 사용자는 다른 사원을 찾을 수 있어야 합니다. 이러한 요구는 STAFF 테이블의 NAME 컬럼과 ORG 테이블의 LOCATION 컬럼에 대한 뷰를 작성하여, DEPT 및 DEPTNUMB 컬럼 각각에 대해 두 개의 테이블을 조인함으로써 충족될 수 있습니다.

```
CREATE VIEW EMPLOCS AS
  SELECT NAME, LOCATION FROM STAFF, ORG
  WHERE STAFF.DEPT=ORG.DEPTNUMB
GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
```

사원 위치 뷰에 액세스하는 사용자는 다음과 같은 정보를 볼 수 있습니다.

NAME	LOCATION
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta

NAME	LOCATION
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

감사 기능을 사용하여 데이터에 대한 액세스 모니터링

DB2 감사 기능은 일련의 사전 정의된 데이터베이스 이벤트에 대한 감사 추적을 생성하고 유지보수할 수 있게 합니다. 감사 기능은 데이터에 대한 액세스를 금지하지는 않지만, 데이터 오브젝트를 액세스하거나 수정하려고 시도하는 레코드를 모니터링하고 유지할 수 있습니다.

SYSADM 권한은 감사 기능 관리자 톨 db2audit을 사용하는 것이 필요합니다.

DB2 감사 기능에 대해서는 307 페이지의 『제6장 DB2 활동 감사』에서 자세한 내용을 참조하십시오.

타스크 및 필수 권한 부여

모든 조직이 동일한 방식으로 작업량을 분담하지는 못합니다. 표5에는 기타 몇몇 공통된 직책, 보통 작업에 수반되는 타스크, 이러한 타스크를 수행하는 데 필요한 권한 또는 특권이 나열되어 있습니다.

표 5. 공통 직책, 타스크 및 필수 권한 부여

직책	타스크	필수 권한 부여
부서 관리자	부서 시스템 감독, 데이터베이스 작성	SYSCTRL 권한. 부서에 자신의 인스턴스가 있으면 SYSADM 권한
보안 관리자	일부 또는 모든 권한 및 특권에 대해 다른 사용자에게 권한 부여	SYSADM 또는 DBADM 권한
데이터베이스 관리자	하나 이상의 데이터베이스를 설계, 개발, 조작, 보호 및 유지보수	하나 이상의 데이터베이스에 대한 DBADM 및 SYSMANT 권한. 어떤 경우에는 SYSCTRL 권한
시스템 조직원	데이터베이스를 모니터링하고 백업 기능 수행함	SYSMANT 권한
응용프로그램 프로그래머	데이터베이스 관리 프로그램 응용프로그램 개발 및 테스트, 테스트 데이터 테이블 작성	기존 패키지에 BINDADD 및 BIND, 하나 이상의 데이터베이스에 CONNECT 및 CREATETAB, 일부 특정 스키마 권한, 일부 테이블에 대한 특권 목록
사용자 분석가	시스템 카탈로그 뷰를 검사하여 응용 프로그램에 대한 데이터 요구량 정의	카탈로그 뷰에 대해서는 SELECT, 하나 이상의 데이터베이스에 대해서는 CONNECT
프로그램 일반 사용자	응용프로그램 실행	패키지에 대해서는 EXECUTE, 하나 이상의 데이터베이스에 대해서는 CONNECT. 이 테이블 다음의 주 참조
정보 센터 상담원	조회 사용자의 데이터 요구사항 정의, 테이블 및 뷰를 작성하고 데이터베이스 오브젝트로 액세스 권한을 부여하여 데이터 제공	하나 이상의 데이터베이스에 대한 DBADM 권한
조회 사용자	SQL문을 발행하여 데이터 검색, 추가, 삭제 또는 변경, 결과를 테이블로 저장	하나 이상의 데이터베이스에 대해서는 CONNECT, 작성중인 테이블 및 뷰의 스키마에 대해서는 CREATEIN, 그리고 일부 테이블 및 뷰에 대해서는 SELECT, INSERT, UPDATE, DELETE

응용프로그램에 동적 SQL문이 들어 있으면, 프로그램 일반 사용자가 EXECUTE 및 CONNECT 이외의 다른 특권(예: SELECT, INSERT, DELETE 및 UPDATE)이 필요할 수도 있습니다.

시스템 카탈로그 사용

각 데이터베이스에 대한 정보는 자동으로 시스템 카탈로그라고 하는 뷰 세트에 보존됩니다. 이 시스템 카탈로그는 데이터베이스가 작성될 때 작성됩니다. 이러한 시스템 카탈로그에서는 테이블, 컬럼, 색인, 프로그램, 특권 및 기타 오브젝트에 대해 설명합니다.

이 중 6개의 뷰에는 사용자의 특권과 각 특권을 권한 부여하는 사용자의 ID가 나열되어 있습니다.

SYSCAT.DBAUTH	데이터베이스 특권 나열
SYSCAT.TABAUTH	테이블과 뷰 특권 나열
SYSCAT.COLAUTH	컬럼 특권 나열
SYSCAT.PACKAGEAUTH	패키지 특권 나열
SYSCAT.INDEXAUTH	색인 특권 나열
SYSCAT.SCHEMAAUTH	스키마 특권 나열
SYSCAT.PASSTHROUGHAUTH	서버 특권 나열

시스템이 사용자에게 권한 부여한 특권은 SYSIBM을 권한 준 사용자로 갖게 됩니다. SYSADM, SYSMANT 및 SYSCTRL은 시스템 카탈로그에 나열되지 않습니다.

CREATE 및 GRANT문은 시스템 카탈로그에 특권을 위치시킵니다. SYSADM 및 DBADM 권한을 가지고 있는 사용자는 시스템 카탈로그 뷰에 대해 SELECT 특권을 권한 부여하고 권한 취소할 수 있습니다. 다음 예에서는 이러한 SQL 조회를 사용하여 특권 정보를 발췌하는 방법을 보여줍니다.

- 303 페이지의 『권한 부여된 특권이 있는 권한 부여 이름 검색』
- 303 페이지의 『DBADM 권한이 있는 모든 이름 검색』
- 303 페이지의 『테이블에 액세스하기 위해 권한 부여된 이름 검색』

- 304 페이지의 『사용자에게 권한 부여된 모든 특권 검색』
- 305 페이지의 『시스템 카탈로그 뷰 보안』.

권한 부여된 특권이 있는 권한 부여 이름 검색

모든 특권에 관한 정보가 하나의 시스템 카탈로그 뷰에 들어 있지는 않습니다. 다음 명령문에서는 특권을 가진 모든 권한 부여 이름을 검색합니다.

```
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE ' FROM SYSCAT.TABAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN ' FROM SYSCAT.COLAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER ' FROM SYSCAT.PASSTHRUAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

정기적으로 이 명령문에 의해 검색되는 목록은 시스템 보안 기능에 정의된 사용자 및 그룹 이름의 목록과 비교되어야 합니다. 그러면 사용자는 더 이상 유효하지 않은 해당 권한 부여 이름을 식별할 수 있습니다.

주: 사용자가 원격 데이터베이스 클라이언트를 지원하고 있는 경우, 원격 클라이언트에만 권한 부여 이름이 정의되고 사용자의 데이터베이스 서버 머신에는 정의되지 않도록 할 수 있습니다.

DBADM 권한이 있는 모든 이름 검색

다음 명령문은 직접적으로 DBADM 권한이 권한 부여된 모든 권한 부여 이름을 검색합니다.

```
SELECT DISTINCT GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
```

테이블에 액세스하기 위해 권한 부여된 이름 검색

다음 명령문에서는 JAMES 규정자로 테이블 EMPLOYEE에 액세스하도록 직접 권한이 부여된 모든 권한 부여 이름을 검색합니다.

```

SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
    WHERE TABNAME = 'EMPLOYEE'
        AND TABSCHEMA = 'JAMES'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
    WHERE TABNAME = 'EMPLOYEE'
        AND TABSCHEMA = 'JAMES'

```

규정자 JAMES로 테이블 EMPLOYEE를 갱신할 수 있는 사용자를 알아내려면, 다음 명령문을 발행하십시오.

```

SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
    WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
        (CONTROLAUTH = 'Y' OR
         UPDATEAUTH = 'Y' OR UPDATEAUTH = 'G')
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
    WHERE DBADMAUTH = 'Y'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
    WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
        PRIVTYPE = 'U'

```

이는 CONTROL 또는 UPDATE 특권이 직접 권한 부여된 이름뿐만 아니라, DBADM 권한을 갖는 모든 권한 부여 이름도 검색합니다. 그러나, SYSADM 권한만을 보유하고 있는 사용자의 권한 부여 이름은 리턴되지 않습니다.

권한 부여 이름 중 일부는 개별 사용자가 아닌 그룹 ID일 수도 있음을 기억하십시오.

사용자에게 권한 부여된 모든 특권 검색

시스템 카탈로그 뷰에 대해 조회함으로써, 사용자가 보유하고 있는 특권의 목록과 다른 사용자에게 권한 부여한 특권의 목록을 검색할 수 있습니다. 예를 들어, 다음 명령문에서는 개별 권한 부여 이름에 직접 권한 부여된 데이터베이스 특권의 목록을 검색합니다.

```

SELECT * FROM SYSCAT.DBAUTH
    WHERE GRANTEE = USER AND GRANTEETYPE = 'U'

```

다음 명령문에서는 특정 사용자에게 의해 직접 권한 부여된 테이블 특권의 목록을 검색합니다.

```
SELECT * FROM SYSCAT.TABAUTH
WHERE GRANTOR = USER
```

다음 명령문에서는 특정 사용자에게 의해 직접 권한 부여된 각 컬럼 특권의 목록을 검색합니다.

```
SELECT * FROM SYSCAT.COLAUTH
WHERE GRANTOR = USER
```

이 명령문에서 키워드 **USER**는 사용자의 권한 부여 이름 값과 항상 같습니다. **USER**는 읽기 전용 특수 레지스터입니다. 특수 레지스터에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

시스템 카탈로그 뷰 보안

데이터베이스가 작성되는 동안, 시스템 카탈로그 뷰에 대한 **SELECT** 특권이 **PUBLIC**에 권한 부여됩니다.(**PUBLIC**에 자동으로 권한 부여되는 다른 특권은 281 페이지의 『데이터베이스 특권』에서 자세한 내용을 참조하십시오.) 대부분의 경우, 보안상 문제점이 나타나지 않습니다. 그러나 매우 중요한 데이터의 경우에 이 테이블들이 데이터베이스에 있는 모든 오브젝트를 서술하기 때문에, 적절하지 않을 수도 있습니다. 이 경우, **PUBLIC**으로부터 **SELECT** 특권을 취소하고, 특정 사용자에게 필요한 **SELECT** 특권을 권한 부여하는 방법을 고려해보십시오. 시스템 카탈로그 뷰에 대해 **SELECT**를 권한 부여하고 취소하는 방식은 모든 뷰에서 동일하지만, 사용자가 이를 수행할 수 있는 **SYSADM** 또는 **DBADM** 권한 가지고 있어야 합니다.

최소한 다음 카탈로그 뷰로의 액세스를 제한하는 방법을 고려해야 합니다.

- SYSCAT.DBAUTH
- SYSCAT.TABAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.COLAUTH
- SYSCAT.PASSTHROUGHAUTH
- SYSCAT.SCHEMAAUTH

이렇게 하면, 브레이크인에 대한 권한 부여 이름을 대상으로 지정하는 데 사용될 수 있는 사용자 특권에 대한 정보를 데이터베이스에 액세스하는 모든 사용자가 사용할 수 없게 됩니다.

또한, 수집된 통계에 대한 컬럼도 조사해야 합니다(*관리 안내서: 성능의 『카탈로그 통계』* 참조). 시스템 카탈로그에 기록된 일부 통계에는 사용자의 환경에서 중요한 데이터가 될 수 있는 데이터값이 들어 있습니다. 이러한 통계에 중요한 데이터가 들어 있으면, SYSCAT.COLUMNS 및 SYSCAT.COLDIST 카탈로그 뷰에 대해 PUBLIC으로부터 SELECT 특권을 권한 취소할 수도 있습니다.

시스템 카탈로그 뷰로의 액세스를 제한하려는 경우, 뷰를 정의하여 각 권한 부여 이름이 자체 특권 정보를 검색하도록 할 수 있습니다.

예를 들어, 다음의 뷰 MYSELECTS에는 사용자의 권한 부여 이름이 SELECT 특권에 직접 권한 부여된 모든 테이블의 이름 및 소유자가 있습니다.

```
CREATE VIEW MYSELECTS AS
  SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
  WHERE GRANTEETYPE = 'U'
  AND GRANTEE = USER
  AND SELECTAUTH = 'Y'
```

이 명령문에서 키워드 USER는 권한 부여 이름 값과 항상 같습니다.

다음 명령문에서는 모든 권한 부여 이름에 사용 가능한 뷰를 만듭니다.

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

그리고 마지막으로, 기본 테이블에 대한 SELECT 특권을 권한 취소해야 합니다.

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC
```

제6장 DB2 활동 감사

알려지거나 예측된 데이터 액세스를 제어하는 데에는 인증, 권한 및 특권을 사용할 수 있으나, 이들 방법은 알려지지 않거나 예측되지 않은 데이터 액세스를 방지하는 데에는 충분하지 않습니다. 데이터 액세스의 이 나중 유형의 검출을 돕기 위해, DB2는 감사 기능을 제공합니다. 원하지 않는 데이터 액세스 및 차후 분석의 성공적인 모니터링은 데이터 액세스의 제어와, 데이터에 대해 악의적이거나 부주의한 권한이 없는 액세스의 방지를 개선시킬 수 있습니다. 시스템 관리 조치를 포함하여 응용프로그램 및 각 사용자 액세스의 모니터링은 사용자 데이터베이스 시스템에서의 활동 실행기록 레코드를 제공할 수 있습니다.

DB2 감사 기능은 일련의 사전 정의된 데이터베이스 이벤트에 대한 감사 추적을 생성하고 유지보수할 수 있게 합니다. 이 기능에서 생성된 레코드는 감사 로그 파일에 유지됩니다. 이들 레코드의 분석은 시스템 오용을 식별하는 사용 패턴을 나타낼 수 있습니다. 일단 시스템 오용이 식별되면, 그러한 시스템 오용을 감소시키거나 줄이기 위한 조치가 취해질 수 있습니다.

감사 기능은 모든 인스턴스 레벨 활동 및 데이터베이스 레벨 활동을 기록하며 인스턴스 레벨에서 동작합니다.

파티션된 데이터베이스 환경에서 작업할 때, 사용자가 연결된 파티션(조정자 노드) 또는 카탈로그 노드(이들 이벤트가 동일한 파티션이 아닌 경우)에서 감사할 수 있는 여러 이벤트가 발생합니다. 이것이 함축하는 의미는 감사 레코드가 둘 이상의 파티션에 의해 생성될 수 있다는 것입니다. 각 감사 레코드의 일부에는 조정자 노드 및 원래 노드 식별자에 대한 정보가 들어 있습니다.

감사 로그(db2audit.log) 및 감사 구성 파일(db2audit.cfg)은 인스턴스의 security 서브디렉토리에 있습니다. 인스턴스 작성시, 가능하다면 운영 체제에 의해 이들 파일에 읽기/쓰기 허용이 설정됩니다. 기본적으로, 허용은 인스턴스 소유자 전용의 읽기/쓰기입니다. 이들 허용을 변경하지 않도록 권장합니다.

감사 기능 관리자 도구(db2audit)의 사용자는 SYSADM 권한/특권을 가져야 합니다.

감사 기능은 명시적으로 중단 및 시작되어야 합니다. 감사 기능이 시작되면, 기존 감사 구성 정보를 사용합니다. 감사 기능은 DB2 서버와 무관하므로, 인스턴스가 중단되더라도 사용중인 상태로 남습니다. 사실, 인스턴스가 중단되면, 감사 레코드는 감사 로그에서 생성될 수 있습니다.

감사 기능의 권한 부여된 사용자는 감사 기능 내에서 다음 조치를 제어할 수 있습니다.

- DB2 인스턴스 내에서 감사 가능한 이벤트 기록을 시작합니다.
- DB2 인스턴스 내에서 감사 가능한 이벤트 기록을 중지합니다.
- 기록될 감사할 수 있는 이벤트 범주 선택을 비롯한 감사 기능의 작동을 구성합니다.
- 현재 감사 구성의 설명을 요청합니다.
- 인스턴스에서 보류중인 감사 레코드를 비우고 감사 로그에 씁니다.
- 감사 로그에서 플랫폼 파일 또는 ASCII 분리 파일로 감사 레코드를 형식화하고 복사하여 감사 레코드를 발취합니다. 로그 레코드의 분석을 위한 준비 또는 로그 레코드의 정리를 위한 준비에서 발취가 수행됩니다.
- 현재 감사 로그에서 감사 레코드를 제거합니다.

생성될 수 있는 감사 레코드의 다른 범주가 있습니다. 감사에 사용 가능한 이벤트 범주의 설명에서(아래), 각 범주의 이름 다음에 오는 것은 범주 유형을 식별하기 위해 사용하는 하나의 단어 키워드임을 기억해야 합니다. 감사에 사용 가능한 이벤트의 범주는 다음과 같습니다.

- 감사(AUDIT). 감사 설정값이 변경되거나 감사 로그가 액세스될 때 레코드를 생성합니다.
- 권한 부여 점검(CHECKING). DB2 오브젝트 또는 기능에 액세스하거나 조작하려는 시도에 대한 권한 점검 동안에 레코드를 생성합니다.
- 오브젝트 유지보수(OBJMAINT). 데이터 오브젝트를 작성하거나 삭제할 때 레코드를 생성합니다.

- 보안 유지보수(SECMANT). 오브젝트 또는 데이터베이스 특권 또는 DBADM 권한을 권한 부여하거나 권한 취소할 때 레코드를 생성합니다. 데이터베이스 관리 프로그램 보안 구성 매개변수 SYSADM_GROUP, SYSCTRL_GROUP 또는 SYSMANT_GROUP이 수정될 때 레코드가 생성됩니다.
- 시스템 관리(SYSADMIN). SYSADM, SYSMANT 또는 SYSCTRL 권한이 필요한 조작이 수행될 때 레코드를 생성합니다.
- 사용자 확인(VALIDATE). 사용자를 인증하거나 시스템 보안 정보를 검색할 때 레코드를 생성합니다.
- 조작 문맥(CONTEXT). 데이터베이스 조작이 수행될 때 조작 문맥을 표시하기 위해 레코드를 생성합니다. 이 범주는 감사 로그 파일을 더 잘 이해하게 합니다. 로그의 이벤트 상관자 필드를 사용할 때, 이벤트 그룹은 다시 하나의 데이터베이스 조작에 연관될 수 있습니다. 예를 들어, 동적 SQL에 대한 SQL문, 정적 SQL에 대한 패키지 식별자 또는 CONNECT와 같이 수행되는 조작 유형의 표시기는 감사 결과를 분석할 때 필요한 문맥을 제공할 수 있습니다.

주: 조작 문맥을 제공하는 SQL문은 매우 길며 CONTEXT 레코드 내에서 완전히 표시됩니다. 이것은 CONTEXT 레코드를 매우 크게 만들 수 있습니다.

- 실패 또는 성공 중 하나를 감사하거나 두 가지 모두를 감사할 수 있습니다.

데이터베이스에서의 조작은 여러 레코드를 생성할 수 있습니다. 생성되어 감사 로그로 이동되는 실제 레코드 수는 감사 기능 구성에서 지정한 대로 기록되는 이벤트의 범주 수에 따라 다릅니다. 또한 성공, 실패 또는 두 가지 모두 중 어느 것을 감사하는지 여부에 따라서도 다릅니다. 이러한 이유로 감사할 이벤트를 선택하는 것은 중요합니다.

감사 기능 활동

감사 기능은 영향을 미치는 데이터베이스 인스턴스를 포함하는 감사 가능한 이벤트를 기록합니다. 이러한 이유로, 감사 기능은 DB2 인스턴스가 중단되는 경우에도 조작할 수 있는 DB2의 독립적 일부입니다. 감사 기능이 사용중이면, 중단된 인스턴스가 시작될 때 인스턴스의 데이터베이스 이벤트 감사가 재개됩니다.

감사 로그에 감사 레코드를 쓰는 시기는 인스턴스에서의 데이터베이스 성능에 상당한 영향을 줍니다. 감사 레코드의 쓰기는 이들 레코드의 생성을 일으키는 이벤트의 발생과 함께 동기적 또는 비동기적으로 발생할 수 있습니다. AUDIT_BUF_SZ 데이터베이스 관리 프로그램 구성 매개변수의 값은 감사 레코드가 수행되는 시기를 판별합니다.

이 매개변수의 값이 0이면, 쓰기는 동기적으로 수행됩니다. 감사 레코드를 생성하는 이벤트가 레코드를 디스크에 쓸 때까지 대기합니다. 각 레코드와 연관된 대기는 DB2 성능이 저하되게 합니다.

AUDIT_BUF_SZ의 값이 0보다 크면, 레코드 쓰기는 비동기적으로 수행됩니다. AUDIT_BUF_SZ의 값이 0보다 커지면 내부 버퍼를 작성하기 위해 몇 개의 4KB 페이지가 사용됩니다. 감사 레코드의 그룹을 디스크에 쓰기 전에 몇 개의 감사 레코드를 유지하기 위해 내부 버퍼가 사용됩니다. 감사 레코드를 감사 이벤트의 결과로서 생성하는 명령문은 레코드가 디스크에 기록될 때까지 기다리지 않고 작업을 계속할 수 있습니다.

비동기 경우에, 감사 레코드가 얼마 동안 채워지지 않은 버퍼에 남아 있을 수 있습니다. 이것이 확장 기간 동안 발생하지 않게 하기 위해 데이터베이스 관리 프로그램은 정기적으로 감사 레코드 쓰기를 강제합니다. 감사 기능의 권한이 부여된 사용자는 명시적 요청으로 감사 버퍼를 비울 수 있습니다.

동기 레코드 쓰기가 있는지 비동기 레코드 쓰기가 있는지 여부에 따라 오류 발생 시기가 다릅니다. 비동기 모드 경우, 감사 레코드가 디스크에 기록되기 전에 버퍼 처리되기 때문에 몇몇 레코드가 유실될 수 있습니다. 동기 모드에서는 오류가 발생해도 많아야 하나의 감사 레코드만 쓰기 금지되기 때문에 하나의 레코드가 유실될 수 있습니다.

ERRORTYPE 감사 기능 매개변수의 설정값은 오류가 DB2와 감사 기능간에 어떻게 관리되는지를 제어합니다. 감사 기능이 사용중이고, ERRORTYPE 감사 기능 매개변수가 AUDIT이면, 감사 기능은 DB2의 다른 부분과 동일한 방식으로 처리됩니다. 성공적이라고 간주되는 명령문과 연관된 감사 이벤트에 대한 감사 레코드가 쓰여져야 합니다(동기 모드에서 디스크로 또는 비동기 모드에서 감사 버퍼로). 이 모드를 수행할 때 오류가 발생할 때마다, 감사 레코드를 생성하는 명령문에 대해 음의 SQLCODE가 응용프로그램에 리턴됩니다. 오류 유형이 NORMAL로 설정

되면, db2audit로부터의 모든 오류는 무시되며 조작의 SQLCODE가 리턴됩니다. ERRORTYPE 감사 기능 매개변수(및 기타 관련 매개변수)에 대해서는 312 페이지의 『감사 기능 사용 시나리오』에서 자세한 내용을 참조하십시오.

API 또는 SQL문과 DB2 인스턴스에 대한 감사 설정값에 따라, 없음, 하나 또는 여러 감사 레코드가 특정 이벤트에 대해 생성될 수 있습니다. 예를 들어, SELECT 부속 조회가 있는 SQL UPDATE문은 테이블에서 UPDATE 특권에 대한 권한 부여 점검의 결과가 들어 있는 하나의 감사 레코드와 테이블에서 SELECT 특권에 대한 권한 부여 점검의 결과가 들어 있는 다른 레코드의 결과를 가져올 수 있습니다.

동적 DML문의 경우,문이 준비되면 모든 권한 부여 점검에 대해 감사 레코드가 생성됩니다. 이 때는 어떠한 감사 점검도 발생하지 않으므로 동일한 사용자에게 의한 이들 명령문의 재사용은 다시 감사되지 않습니다. 그러나, 특권 정보가 들어 있는 카탈로그 테이블 중 하나가 변경되면, 다음 작업 단위(UOW)에서 캐쉬된 동적 SQL 문에 대한 명령문 특권이 점검되며 하나 이상의 새로운 감사 레코드가 작성됩니다.

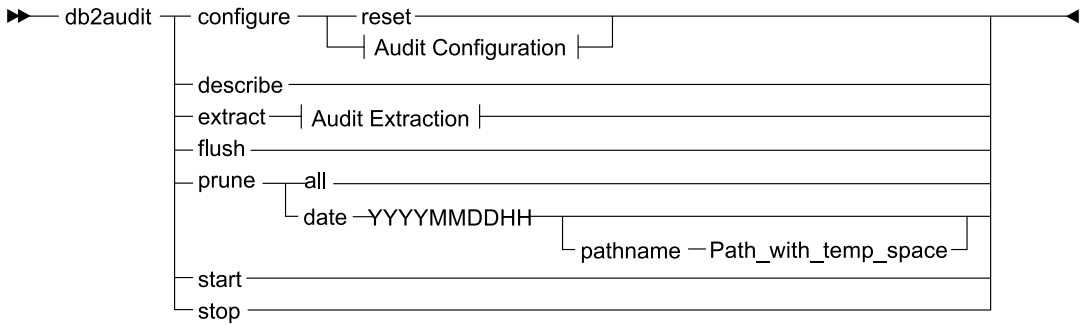
정적 DML문만이 들어 있는 패키지의 경우, 감사 레코드를 생성할 수 있는 유일한 감사 가능한 이벤트는 사용자가 해당 패키지를 실행할 특권을 가지고 있는지 여부를 보기 위한 권한 부여 점검입니다. 패키지에서 정적 SQL문에 필요한 가능한 감사 레코드 작성과 권한 부여 점검은 패키지가 사전 처리 컴파일되거나 바인드될 때 수행됩니다. 패키지 내의 정적 SQL문의 실행은 감사 가능하지 않습니다. 패키지가 사용자에게 의해 명시적으로 또는 시스템에 의해 내재적으로 다시 바인드될 때, 정적 SQL문에 의해 필요한 권한 부여 점검에 대한 감사 레코드가 생성됩니다.

권한 부여 점검이 명령문 실행 시간에서 수행되는 명령문의 경우(예: DDL, GRANT 및 REVOKE문), 이들 명령문이 사용될 때마다 감사 레코드가 생성됩니다.

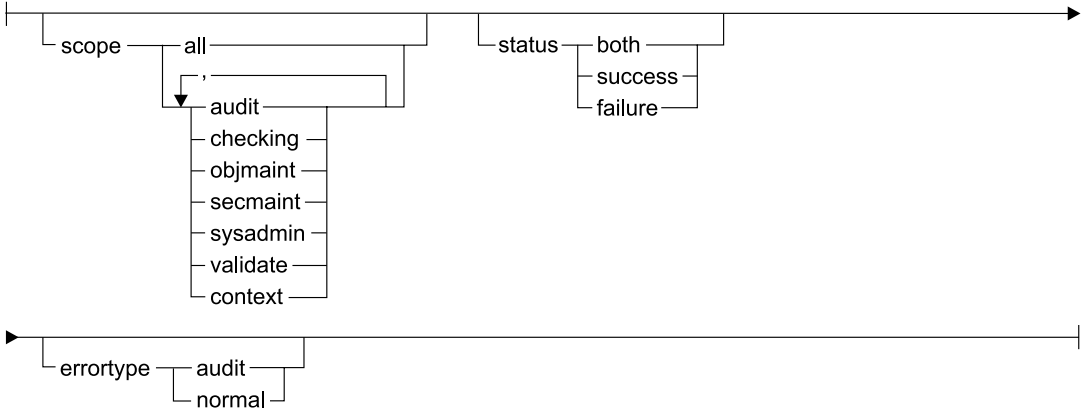
주: DDL을 실행할 때, 감사 레코드의 모든 이벤트(문맥 이벤트 제외)에 기록되는 색션 번호는 명령문의 실제 색션 번호와 상관없이 0이 됩니다.

감사 기능 사용 시나리오

감사 기능의 구문을 고려함으로써, 기능이 사용될 수 있는 방식을 검토할 수 있습니다.



감사 구성:



감사 추출:

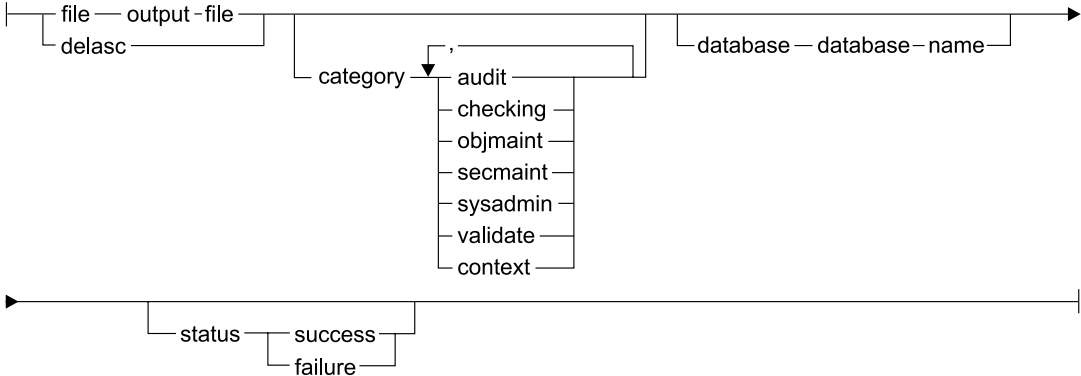


그림 7. DB2AUDIT 구문

다음은 각 매개변수의 내포된 사용 및 설명입니다.

configure

이 매개변수를 사용하면 인스턴스의 *security* 서브디렉토리에 있는 *db2audit.cfg* 구성 파일을 수정할 수 있습니다. 이 파일에 대한 갱신은 인스턴스가 종료될 때에도 발생할 수 있습니다. 인스턴스가 동적으로 사용중일 때 발생하는 갱신은 모든 파티션에 걸쳐 DB2에 의해 수행중인 감사에 영향을 미칩니다. 감사 기능이 시작됐을 때 구성 파일을 구성하면 감사 레코드가 작성되고 감사 가능한 이벤트의 감사 범주가 감사됩니다.

다음은 구성 파일에서 가능한 조치입니다.

- **RESET.** 이 조치는 구성 파일이 초기 구성으로 되돌아가게 합니다(여기서 SCOPE는 CONTEXT를 제외한 모든 범주이며, STATUS는 FAILURE이며, ERRORTYPE은 NORMAL이며, AUDIT는 OFF입니다). 이 조치는 원래의 감사 구성 파일이 유실되거나 손상된 경우 새로운 감사 구성 파일을 작성합니다.
- **SCOPE.** 이 조치는 이벤트의 어떤 범주가 감사되는지를 지정합니다. 또한, 이 조치는 감사의 특정 초점을 허용하며 로그의 성장을 감소시킵니다. 가능한 한 로그되는 이벤트의 수와 유형을 제한하도록 권장합니다. 그렇지 않으면, 감사 로그가 급격히 증가합니다.

주: 기본값 SCOPE는 CONTEXT를 제외한 모든 범주이며 레코드가 빠르게 생성되는 결과를 가져올 수 있음을 기억하십시오. 모드(동기 또는 비동기)와 결합된, 범주의 선택은 상당한 성능 저하를 가져오며 디스크 요구사항을 상당히 증가시키는 결과를 가져올 수 있습니다.

- **STATUS.** 이 조치는 성공하거나 실패한 이벤트만 또는 성공 및 실패한 이벤트 둘다를 기록하는지 여부를 지정합니다.

주: 조작 상태가 알려지기 전에 문맥 이벤트가 발생합니다. 그러므로, 이러한 이벤트는 이 매개변수와 연관된 값과 무관하게 기록됩니다.

- **ERRORTYPE.** 이 조치는 감사 오류가 사용자에게 리턴되는지, 아니면 무시되는지 여부를 지정합니다. 이 매개변수 값은 다음과 같습니다.
 - **AUDIT.** 감사 기능 내에서 발생하는 오류를 포함하여 모든 오류는 DB2가 관리하며 모든 음의 SQLCODE는 호출자에게 다시 보고됩니다.
 - **NORMAL.** db2audit가 생성한 모든 오류는 무시되며 수행중인 조작과 연관된 오류에 대한 SQLCODE만이 응용프로그램에 리턴됩니다.

describe

이 매개변수는 현재 감사 구성 정보 및 상태를 표준 출력에 표시합니다.

extract 이 매개변수는 감사 로그로부터 표시된 대상으로 감사 레코드의 이동을 허용합니다. 선택적 절이 지정되지 않으면, 모든 감사 레코드가 발췌되어 플랫폼 보고서 파일에 위치합니다. 『extract』 매개변수가 지정되지 않으면, 감사 레코드는 security 디렉토리의 db2audit.out라는 파일에 위치합니다. output_file이 있는 경우, 오류 메시지가 리턴됩니다.

다음은 발췌할 때 사용할 수 있는 가능한 옵션입니다.

- FILE. 발췌된 감사 레코드는 파일(output_file)에 위치합니다.
- DELASC. 발췌된 감사 레코드는 DB2 관계 테이블로 로드하는 데 적합한 컬럼 식별자가 있는 ASCII 형식으로 위치합니다. 출력은 각 범주에 하나씩 개별 파일에 위치합니다. 파일 이름은 다음과 같습니다.
 - audit.del
 - checking.del
 - objmaint.del
 - secmaint.del
 - sysadmin.del
 - validate.del
 - context.del

DELASC 선택사항은 감사 로그에서 발췌할 때 기본 감사 문자열 분리 문자(『0xff』)를 대체할 수 있게 합니다. 감사 레코드를 보유할 테이블로 로드하는 준비에서 사용하려는 새로운 분리 문자가 뒤에 DELASC DELIMITER를 사용하십시오. 새로운 로드 분리 문자는 단일 문자(예: !) 또는 16진수(예: 0xff)를 표현하는 4바이트 문자열 중 하나입니다. 335 페이지의 『감사 기능 추가 정보 및 기술』에서 자세한 내용을 참조하십시오.

- CATEGORY. 감사 이벤트의 지정 범주에 대한 감사 레코드가 발췌됩니다. 지정되지 않으면, 모든 범주가 발췌 대상이 됩니다.
- DATABASE. 지정 데이터베이스의 감사 레코드가 발췌됩니다. 지정되지 않으면, 모든 데이터베이스가 발췌 대상이 됩니다.
- STATUS. 지정 상태의 감사 레코드가 발췌됩니다. 지정되지 않으면, 모든 레코드가 발췌 대상이 됩니다.

비우기 이 매개변수는 보류중인 모든 감사 레코드가 감사 로그에 쓰여지도록 강제합니다.

또한, 감사 상태는 감사 기능이 오류 상태에 있는 경우 『기록할 수 없음』에서 『기록 준비』 상태로 엔진에서 재설정됩니다.

제거 이 매개변수는 감사 로그에서 감사 레코드의 삭제를 허용합니다. 감사 기능이 사용중이며 이벤트의 『감사』 범주가 감사용으로 지정된 경우, 감사 레코드는 감사 로그가 제거된 후에 기록됩니다.

다음은 정리할 때 사용할 수 있는 가능한 옵션입니다.

- ALL. 감사 로그의 모든 감사 레코드가 삭제됩니다.
- DATE `yyyymmddhh`. 지정한 날짜/시간에 발생하거나 지정 날짜/시간 이전에 발생한 모든 감사 레코드를 감사 로그에서 삭제하도록 지정할 수 있습니다. 사용자는 감사 로그를 정리할 때 감사 기능이 임시 공간으로서 사용할

`pathname`

을 선택적으로 제공할 수 있습니다. 이 임시 공간은 감사 로그가 있는 디스크가 가득 차서 정리 조작할 공간이 충분하지 않을 때 감사 로그를 정리할 수 있게 합니다.

start 이 매개변수는 `db2audit.cfg` 파일의 내용에 근거하여 감사 기능이 감사 이벤트를 시작할 수 있게 합니다. 파티션된 DB2 인스턴스에서, 이 절이 지정되면 모든 파티션에서 감사가 시작됩니다. 이벤트의 『감사』 범주가 감사에 지정된 경우, 감사 기능이 시작될 때 감사 레코드가 기록됩니다.

stop 이 매개변수는 감사 기능이 감사 이벤트를 중단하게 합니다. 파티션된 DB2 인스턴스에서, 이 절이 지정되면 모든 파티션에서 감사가 중단됩니다. 이벤트의 『감사』 범주가 감사에 지정된 경우, 감사 기능이 중단될 때 감사 레코드가 기록됩니다.

감사 기능 메시지

SQL1322N 감사 로그 파일에 쓰는 중에 오류가 발생했습니다.

설명: 감사 이벤트를 감사 로그 파일에 기록하도록 DB2 감사 기능이 호출될 때 오류가 발생했습니다. 감사 로그가 있는 파일 시스템에 공간이 없습니다.

사용자 응답: 시스템 관리자는 이 파일 시스템에서 공간을 해제하거나 감사 로그의 크기를 줄이도록 제거해야 합니다.

더 많은 공간이 사용 가능하면, `db2audit`를 사용하여 메모리의 데이터를 지우고 감사 프로그램을 준비 상태로 재설정하십시오. 삭제한 레코드를 복구할 수 없으면, 해당 발취가 발생했는지 또는 로그를 정리하기 전에 로그 사본을 작성했는지 확인하십시오.

sqlcode: 1322

sqlstate: 50030

SQL1323N 감사 구성 파일 액세스 중에 오류가 발생했습니다.

설명: 감사 구성 파일(db2audit.cfg)을 열 수 없거나 유효하지 않습니다. 이 오류의 가능한 이유는 db2audit.cfg 파일이 없거나 손상된 것입니다.

사용자 응답: 다음 조치 중 하나를 취하십시오.

- 파일의 저장 버전에서 복원하십시오.

- 다음을 발행하여 감사 기능 구성 파일을 재설정하십시오.

db2audit reset

sqlcode: 1323

sqlstate: 57019

감사 기능 레코드 레이아웃

DELASC 발취 옵션을 사용하여 감사 로그에서 감사 레코드가 발취되면, 각 레코드는 다음 테이블에 표시된 형식 중 하나를 가집니다. 각 테이블은 샘플 레코드의 내용을 표시하여 시작합니다. 레코드의 각 항목 설명이 해당 테이블에서 한 번에 하나의 행에 표시됩니다. 항목이 중요하면, 해당 항목의 이름이 강조표시(굵은체로 표시)됩니다. 이들 항목에는 사용자에게 가장 관심 있는 정보가 들어 있습니다.

주:

1. 샘플 레코드의 모든 필드가 값을 갖지는 않습니다.
2. 『액세스가 시도됨』과 같은 일부 필드는 비트맵으로서 컬럼 식별자가 있는 ASCII 형식으로 저장됩니다. 그러나, 이 플랫폼 보고서 파일에서 이들 필드는 비트맵 값을 표현하는 문자열 세트로서 나타납니다.

표 6. AUDIT 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-11.54.05.151232;category=AUDIT;audit event=START; event correlator=0;event status=0; userid=boss;authid=BOSS;		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. AUDIT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 CONFIGURE, DB2AUD, EXTRACT, FLUSH, PRUNE, START, STOP 및 UPDATE_ADMIN_CFG입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.

표 6. AUDIT 이벤트에 대한 감사 레코드 레이아웃 (계속)

```
timestamp=1998-06-24-11.54.05.151232;category=AUDIT;audit event=START;
event correlator=0;event status=0;
userid=boss;authid=BOSS;
```

NAME	FORMAT	DESCRIPTION
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID

표 7. CHECKING 이벤트에 대한 감사 레코드 레이아웃

```
timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT;
event correlator=2;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
object name=F00;object type=DATABASE;
access approval reason=DATABASE; access attempted=CONNECT;
```

NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. CHECKING
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 CHECKING_OBJECT 및 CHECKING_FUNCTION 입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레 코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID

표 7. CHECKING 이벤트에 대한 감사 레코드 레이아웃 (계속)

timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE; access attempted=CONNECT;		
NAME	FORMAT	DESCRIPTION
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER MAPPING, SERVER OPTION, TRANSFORM, TYPE MAPPING, FUNCTION MAPPING, SUMMARY TABLES 및 NONE입니다.
액세스 승인 이유	CHAR(18)	액세스가 이 감사 이벤트에 승인된 이유를 나타냅니다. 가능한 값은 이 표 다음에 오는 첫번째 목록에 표시됩니다.
액세스가 시도됨	CHAR(18)	시도된 액세스 유형을 나타냅니다. 가능한 값은 이 표 다음에 오는 두번째 목록에 표시됩니다.

다음은 가능한 CHECKING 액세스 승인 이유의 목록입니다.

0x0000000000000001 ACCESS DENIED

액세스가 승인되지 않고 거부되었습니다.

0x0000000000000002 SYSADM

액세스가 승인되었으며, 응용프로그램/사용자가 SYSADM 권한을 가집니다.

0x0000000000000004 SYSCTRL

액세스가 승인되었으며, 응용프로그램/사용자가 SYSCTRL 권한을 가집니다.

0x0000000000000008 SYSMANT

액세스가 승인되었으며, 응용프로그램/사용자가 SYSMANT 권한을 가집니다.

0x0000000000000010 DBADM

액세스가 승인되었으며, 응용프로그램/사용자가 DBADM 권한을 가집니다.

0x0000000000000020 DATABASE PRIVILEGE

액세스가 승인되었으며, 응용프로그램/사용자가 데이터베이스에서 명시적 특권을 가집니다.

0x0000000000000040 OBJECT PRIVILEGE

액세스가 승인되었으며, 응용프로그램/사용자가 오브젝트 또는 기능에서 명시적 특권을 가집니다.

0x0000000000000080 DEFINER

액세스가 승인되었으며, 응용프로그램/사용자가 오브젝트 또는 기능의 정의자입니다.

0x0000000000000100 OWNER

액세스가 승인되었으며, 응용프로그램/사용자가 오브젝트 또는 기능의 소유자입니다.

0x0000000000000200 CONTROL

액세스가 승인되었으며, 응용프로그램/사용자가 오브젝트 또는 기능에서 CONTROL 특권을 가집니다.

0x0000000000000400 BIND

액세스가 승인되었으며, 응용프로그램/사용자가 패키지에서 바인드 특권을 가집니다.

다음은 가능한 CHECKING 액세스가 시도된 유형의 목록입니다.

0x0000000000000002 ALTER

오브젝트 변경 시도

0x0000000000000004 DELETE

오브젝트 삭제 시도

0x0000000000000008 INDEX

색인 사용 시도

0x0000000000000010 INSERT

오브젝트로 삽입 시도

0x0000000000000020 SELECT

테이블 또는 뷰의 조회 시도

0x0000000000000040 UPDATE

오브젝트에서의 데이터 갱신 시도

0x0000000000000080 REFERENCE

오브젝트간의 참조 제한조건 설정 시도

0x0000000000000100 CREATE

오브젝트 작성 시도

0x0000000000000200 DROP

오브젝트 삭제 시도

0x0000000000000400 CREATEIN

다른 스키마 내에서 오브젝트 작성 시도

0x0000000000000800 DROPIN

다른 스키마 내에서 발견된 오브젝트 삭제 시도

0x0000000000001000 ALTERIN

다른 스키마 내에서 발견된 오브젝트 변경 또는 수정 시도

0x0000000000002000 EXECUTE

응용프로그램 실행 또는 수행 시도

0x0000000000004000 BIND

응용프로그램 바인드 또는 준비 시도

0x0000000000008000 SET EVENT MONITOR

이벤트 모니터 스위치 설정 시도

0x0000000000010000 SET CONSTRAINTS

오브젝트에서 제한조건 설정 시도

0x0000000000020000 COMMENT ON

오브젝트에서 주석 작성 시도

0x0000000000040000 GRANT

오브젝트의 특권을 또다른 사용자 ID로 권한 부여 시도

0x0000000000080000 REVOKE

사용자 ID에서 오브젝트의 특권 권한 취소 시도

0x0000000000100000 LOCK

오브젝트 잠금 시도

0x0000000000200000 RENAME

오브젝트 이름 바꾸기 시도

0x0000000000400000 CONNECT

오브젝트 연결 시도

0x0000000000800000 Member of SYS Group

SYS 그룹의 구성원에 액세스하거나 사용하려고 시도

0x0000000001000000 Access All

보유된 오브젝트에서 모든 필수 특권으로 명령문 실행 시도 (DBADM/
SYSADM에만 사용).

0x0000000002000000 Drop All

다중 오브젝트 삭제 시도

0x0000000004000000 LOAD

테이블 공간에서 테이블 로드 시도

0x0000000008000000 USE

테이블 공간에서 테이블 작성 시도

표 8. OBJMAINT 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. OBJMAINT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 CREATE_OBJECT, RENAME_OBJECT 및 DROP_OBJECT입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름

표 8. OBJMAINT 이벤트에 대한 감사 레코드 레이아웃 (계속)

timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;		
NAME	FORMAT	DESCRIPTION
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER MAPPING, SERVER OPTION, TRANSFORM, TYPE MAPPING, FUNCTION MAPPING, SUMMARY TABLES 및 NONE입니다.

표 9. SECMAINT 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS; object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. SECMAINT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 GRANT, REVOKE, IMPLICIT_GRANT, IMPLICIT_REVOKE 및 UPDATE_DBM_CFG입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.

표 9. SECMAINT 이벤트에 대한 감사 레코드 레이아웃 (계속)

timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS; object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;		
NAME	FORMAT	DESCRIPTION
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER MAPPING, SERVER OPTION, TRANSFORM, TYPE MAPPING, FUNCTION MAPPING, SUMMARY TABLES 및 NONE입니다.
권한 준 사용자	VARCHAR(128)	권한을 부여한 사용자 ID

표 9. SECMAINT 이벤트에 대한 감사 레코드 레이아웃 (계속)

```
timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT;
event correlator=4;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.boss.980624155728;application name=db2bp;
package schema=NULLID;package name=SQLC28A1;
package section=0;object schema=BOSS; object name=T1;object type=TABLE;
grantor=BOSS;grantee=WORKER;grantee
type=USER;privilege=SELECT;
```

NAME	FORMAT	DESCRIPTION
권한 받은 사용자	VARCHAR(128)	특권 또는 권한이 권한 부여되거나 권한 취소된 권한 받은 사용자 ID
권한 받은 사용자 유형	VARCHAR(32)	권한 부여되거나 권한 취소된 권한 받은 사용자의 유형. 가능한 값은 USER, GROUP 또는 BOTH입니다.
특권 또는 권한	CHAR(18)	특권 또는 권한 유형이 권한 부여되거나 권한 취소되었음을 나타냅니다. 가능한 값은 이 표 다음에 오는 목록에 표시됩니다.

다음은 가능한 SECMAINT 특권 또는 권한의 목록입니다.

0x0000000000000001 Control Table

테이블에서 권한 부여되거나 권한 취소된 특권 제어

0x0000000000000002 ALTER TABLE

테이블을 변경하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000004 GRANT를 가진 ALTER TABLE

허용된 특권 권한 부여로 테이블을 변경하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000008 DELETE TABLE

테이블을 삭제하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000010 DELETE TABLE with GRANT

허용된 특권 권한 부여로 테이블을 삭제하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000020 Table Index

색인에서 권한 부여하거나 권한 취소한 특권

0x0000000000000040 Table Index with GRANT

허용된 특권 권한 부여로 색인에서 권한 부여되거나 권한 취소된 특권

0x0000000000000080 Table INSERT

테이블의 삽입에서 권한 부여하거나 권한 취소된 특권

0x0000000000000100 Table INSERT with GRANT

허용된 특권 권한 부여로 테이블의 삽입에서 권한 부여되거나 권한 취소된 특권

0x0000000000000200 Table SELECT

테이블의 선택에서 권한 부여하거나 권한 취소한 특권

0x0000000000000400 Table SELECT with GRANT

허용된 특권 권한 부여로 테이블의 선택에서 권한 부여되거나 권한 취소된 특권

0x0000000000000800 Table UPDATE

테이블의 갱신에서 권한 부여하거나 권한 취소된 특권

0x0000000000001000 Table UPDATE with GRANT

허용된 특권 권한 부여로 테이블의 갱신에서 권한 부여되거나 권한 취소된 특권

0x0000000000002000 Table REFERENCE

테이블의 참조에서 권한 부여하거나 권한 취소된 특권

0x0000000000004000 Table REFERENCE with GRANT

허용된 특권 권한 부여로 테이블의 참조에서 권한 부여되거나 권한 취소된 특권

0x0000000000008000 Package BIND

패키지에서 권한 부여되거나 권한 취소된 바인드 특권

0x0000000000010000 Package EXECUTE

패키지에서 권한 부여되거나 권한 취소된 실행 특권

0x0000000000020000 CREATEIN Schema

스키마에서 권한 부여되거나 권한 취소된 Createin 특권

0x000000000040000 CREATEIN Schema with GRANT

허용된 특권 권한 부여로 스키마에서 권한 부여되거나 권한 취소된 Createin 특권

0x000000000080000 DROPIN Schema

스키마에서 권한 부여되거나 권한 취소된 Dropin 특권

0x000000000100000 DROPIN Schema with GRANT

허용된 특권 권한 부여로 스키마에서 권한 부여되거나 권한 취소된 Dropin 특권

0x000000000200000 ALTERIN Schema

스키마에서 권한 부여되거나 권한 취소된 Alterin 특권

0x000000000400000 ALTERIN Schema with GRANT

허용된 특권 권한 부여로 스키마에서 권한 부여되거나 권한 취소된 Alterin 특권

0x000000000800000 DBADM Authority

권한 부여되거나 권한 취소된 DBADM 권한

0x000000001000000 CREATETAB Authority

권한 부여되거나 권한 취소된 Createtab 권한

0x000000002000000 BINDADD Authority

권한 부여되거나 권한 취소된 Bindadd 권한

0x000000004000000 CONNECT Authority

권한 부여되거나 권한 취소된 Connect 권한

0x000000008000000 Create not fenced Authority

권한 부여되거나 권한 취소된 분리(fenced) 권한을 작성하지 않음

0x000000010000000 Implicit Schema Authority

권한 부여되거나 권한 취소된 내재된 스키마

0x000000020000000 Server PASSTHRU

이 서버에 대해 pass-through 기능을 사용하기 위해 권한 부여되거나 권한 취소된 특권(연합 데이터베이스 데이터 소스)

0x0000000100000000 Table Space USE

테이블 공간에서 테이블을 작성하기 위해 권한 부여되거나 권한 취소된 특
권

0x0000000200000000 Table Space USE with GRANT

특권이 허용된 테이블 공간에서 테이블을 작성하기 위해 권한 부여되거나
권한 취소된 특권

0x0000000400000000 Column UPDATE

테이블에 있는 하나 이상의 특정 컬럼에 대한 갱신에서 권한 부여되거나
권한 취소된 특권

0x0000000800000000 Column UPDATE with GRANT

특권이 허용된 테이블의 하나 이상의 특정 컬럼에 대한 갱신에서 권한 부
여되거나 권한 취소된 특권

0x0000001000000000 Column REFERENCE

테이블에서 하나 이상의 특정 컬럼에 대한 참조에서 권한 부여되거나 권
한 취소된 특권

0x0000002000000000 Column REFERENCE with GRANT

특권이 허용된 테이블의 하나 이상의 특정 컬럼에 대한 참조에서 권한 부
여되거나 권한 취소된 특권

0x0000004000000000 LOAD Authority

권한 부여되거나 권한 취소된 LOAD 권한

표 10. SYSADMIN 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. SYSADMIN
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 이 표 다음에 오는 목록에 표시됩니다.

표 10. SYSADMIN 이벤트에 대한 감사 레코드 레이아웃 (계속)

timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;		
NAME	FORMAT	DESCRIPTION
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호

다음은 가능한 SYSADMIN 감사 이벤트의 목록입니다.

표 11. SYSADMIN 감사 이벤트

START_DB2	ROLLFORWARD_DB
STOP_DB2	SET_RUNTIME_DEGREE
CREATE_DATABASE	SET_TABLESPACE_CONTAINERS
DROP_DATABASE	UNCATALOG_DB
UPDATE_DBM_CFG	UNCATALOG_DCS_DB
UPDATE_DB_CFG	UNCATALOG_NODE
CREATE_TABLESPACE	UPDATE_ADMIN_CFG
DROP_TABLESPACE	UPDATE_MON_SWITCHES
ALTER_TABLESPACE	LOAD_TABLE
RENAME_TABLESPACE	DB2AUDIT
CREATE_NODEGROUP	SET_APPL_PRIORITY
DROP_NODEGROUP	CREATE_DB_AT_NODE
ALTER_NODEGROUP	KILLDBM
CREATE_BUFFERPOOL	MIGRATE_SYSTEM_DIRECTORY
DROP_BUFFERPOOL	DB2REMOT
ALTER_BUFFERPOOL	DB2AUD
CREATE_EVENT_MONITOR	MERGE_DBM_CONFIG_FILE
DROP_EVENT_MONITOR	UPDATE_CLI_CONFIGURATION
ENABLE_MULTIPAGE	OPEN_TABLESPACE_QUERY
MIGRATE_DB_DIR	SINGLE_TABLESPACE_QUERY
DB2TRC	CLOSE_TABLESPACE_QUERY
DB2SET	FETCH_TABLESPACE
ACTIVATE_DB	OPEN_CONTAINER_QUERY
ADD_NODE	FETCH_CONTAINER_QUERY
BACKUP_DB	CLOSE_CONTAINER_QUERY
CATALOG_NODE	GET_TABLESPACE_STATISTICS
CATALOG_DB	DESCRIBE_DATABASE
CATALOG_DCS_DB	ESTIMATE_SNAPSHOT_SIZE
CHANGE_DB_COMMENT	READ_ASYNC_LOG_RECORD
DEACTIVATE_DB	PRUNE_RECOVERY_HISTORY
DROP_NODE_VERIFY	UPDATE_RECOVERY_HISTORY
FORCE_APPLICATION	QUIESCE_TABLESPACE
GET_SNAPSHOT	UNLOAD_TABLE
LIST_DRDA_INDOUBT_TRANSACTIONS	UPDATE_DATABASE_VERSION
MIGRATE_DB	CREATE_INSTANCE
RESET_ADMIN_CFG	DELETE_INSTANCE
RESET_DB_CFG	SET_EVENT_MONITOR
RESET_DBM_CFG	GRANT_DBADM
RESET_MONITOR	REVOKE_DBADM
RESTORE_DB	GRANT_DB_AUTHORITIES
	REVOKE_DB_AUTHORITIES
	REDIST_NODEGROUP

표 12. VALIDATE 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;audit event=CHECK_GROUP_MEMBERSHIP; event correlator=2;event status=-1092; database=F00;userid=boss;authid=BOSS;execution id=newton; application id=*LOCAL.newton.980624124210;application name=testapp; auth type=SERVER;		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. VALIDATE
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 GET_GROUPS, GET_USERID, AUTHENTICATE_PASSWORD 및 VALIDATE_USER입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
실행 ID	VARCHAR(1024)	감사 이벤트시 사용중인 실행 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
인증 유형	VARCHAR(32)	감사 이벤트시 인증 유형
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호

표 13. CONTEXT 이벤트에 대한 감사 레코드 레이아웃

timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;audit event=EXECUTE_IMMEDIATE; event correlator=3; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=203;text=create table audit(c1 char(10), c2 integer);		
NAME	FORMAT	DESCRIPTION
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. CONTEXT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 이 표 다음에 오는 목록에 표시됩니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 식별자. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트가 발생할 때 사용중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트가 발생할 때 사용중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트가 발생할 때 사용중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트가 발생할 때 사용중인 패키지의 섹션 번호
명령문 텍스트(명령문)	CLOB(32K)	적용 가능한 경우 SQL문의 텍스트. SQL문이 사용 가능하지 않은 경우 널(NULL)입니다.

다음은 가능한 CONTEXT 감사 이벤트의 목록입니다.

표 14. CONTEXT 감사 이벤트

CONNECT	SET_APPL_PRIORITY
CONNECT_RESET	RESET_DB_CFG
ATTACH	GET_DB_CFG
DETACH	GET_DFLT_CFG
DARI_START	UPDATE_DBM_CFG
DARI_STOP	SET_MONITOR
BACKUP_DB	GET_SNAPSHOT
RESTORE_DB	ESTIMATE_SNAPSHOT_SIZE
ROLLFORWARD_DB	RESET_MONITOR
OPEN_TABLESPACE_QUERY	OPEN_HISTORY_FILE
FETCH_TABLESPACE	CLOSE_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	FETCH_HISTORY_FILE
OPEN_CONTAINER_QUERY	SET_RUNTIME_DEGREE
CLOSE_CONTAINER_QUERY	UPDATE_AUDIT
FETCH_CONTAINER_QUERY	DBM_CFG_OPERATION
SET_TABLESPACE_CONTAINERS	DISCOVER
GET_TABLESPACE_STATISTIC	OPEN_CURSOR
READ_ASYNC_LOG_RECORD	CLOSE_CURSOR
QUIESCE_TABLESPACE	FETCH_CURSOR
LOAD_TABLE	EXECUTE
UNLOAD_TABLE	EXECUTE_IMMEDIATE
UPDATE_RECOVERY_HISTORY	PREPARE
PRUNE_RECOVERY_HISTORY	DESCRIBE
SINGLE_TABLESPACE_QUERY	BIND
LOAD_MSG_FILE	REBIND
UNQUIESCE_TABLESPACE	RUNSTATS
ENABLE_MULTIPAGE	REORG
DESCRIBE_DATABASE	REDISTRIBUTE
DROP_DATABASE	COMMIT
CREATE_DATABASE	ROLLBACK
ADD_NODE	REQUEST_ROLLBACK
FORCE_APPLICATION	IMPLICIT_REBIND

감사 기능 추가 정보 및 기술

CHECKING 이벤트로 작업할 때 대부분의 경우, 감사 레코드의 오브젝트 유형 필드는 필수 특권 또는 권한이 오브젝트에 액세스하려는 사용자 ID에 의해 보유되는지 여부를 알아 보기 위해 점검되는 오브젝트입니다. 예를 들어, 사용자가 컬럼을 추가하여 테이블을 변경하려고 하는 경우, CHECKING 이벤트 감사 레코드는 시도된 액세스가 『ALTER』이고 점검중인 오브젝트 유형이 『TABLE』이었음을 나타냅니다(주: 점검되어야 하는 것은 테이블 특권이므로 컬럼이 아닙니다).

그러나, 사용자 ID가 오브젝트를 작성 또는 바인드하거나, 오브젝트를 삭제할 수 있게 하는 데이터베이스 권한이 있는지 여부를 확인하는 것이 점검에 포함되면, 데이터베이스에 대한 점검이 있다고 할지라도, 오브젝트 유형 필드는(데이터베이스 자체보다는) 작성, 바인드 또는 삭제되는 오브젝트를 지정합니다.

테이블에서 색인을 작성할 때, 색인을 작성하는 특권이 필요하므로, CHECKING 이벤트 감사 레코드는 『작성』보다는 『색인』의 액세스 시도 유형을 갖습니다.

기존의 패키지를 바인딩할 때, OBJMAINT 이벤트 감사 레코드가 패키지의 DROP 용으로 작성되며, 다른 OBJMAINT 이벤트 감사 레코드는 패키지의 새로운 사본의 CREATE용으로 작성됩니다.

SQL DDL는 성공적인 것으로 기록되는 OBJMAINT 또는 SECMAINT 이벤트를 생성할 수 있습니다. 그러나 이벤트 로깅 이후에 뒤따르는 오류가 ROLLBACK을 유발할 수 있습니다. 이것은 오브젝트를 작성하지 않은 채로 두거나, GRANT 또는 REVOKE 조치를 완료하지 않은 채로 둡니다. 이 경우, CONTEXT 이벤트의 사용이 중요합니다. 이러한 CONTEXT 이벤트 감사 레코드, 특히 이벤트를 종료하는 명령문은 시도된 조작의 완료 특성을 나타냅니다.

DB2 관계 테이블로 로드하는 데 적합한 컬럼 식별자가 있는 ASCII 형식에서 감사 레코드를 받체할 때, 명령문 텍스트 필드 내에서 사용하는 분리 문자를 지워야 합니다. 이것은 컬럼 식별자가 있는 ASCII 파일을 받체할 때 수행될 수 있으며, 다음을 사용하여 수행됩니다.

```
db2audit extract delasc delimiter <load delimiter>
```

로드 분리 문자는 단일 문자(예: ")이거나 16진수(예: 『0xff』)를 표현하는 4바이트 문자열입니다. 유효한 명령의 예는 다음과 같습니다.

```
db2audit extract delasc
db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

발췌할 때 분리 문자로서 기본 로드 분리 문자(『0xff』)가 아닌 다른 문자를 사용한 경우, LOAD 명령에 MODIFIED BY 옵션을 사용해야 합니다. 분리 문자로서 『0xff』가 사용된 LOAD 명령의 부분 예는 다음과 같습니다.

```
db2 load from context.del of del modified by char del 0xff replace into ...
```

이것은 기본 로드 문자열 분리 문자인 『0xff』를 겹쳐씹니다.

DB2 감사 기능 활동 제어

감사 기능 활동 제어에 대한 설명의 일부로서, 다음과 같은 간단한 시나리오를 사용합니다. *newton* 사용자는 테이블을 연결하고 작성하는 *testapp* 응용프로그램을 실행합니다. 이와 동일한 응용프로그램이 아래 설명하는 각 예에서 사용됩니다.

극단적인 예를 표시하며 시작합니다. 모든 성공 및 실패 감사 이벤트를 감사하도록 결정했으므로, 다음과 같은 방법으로 감사 기능을 구성합니다.

```
db2audit configure scope all status both
```

주: 이것은 가능한 모든 감사 기능 이벤트에 대한 감사 레코드를 작성합니다. 결과적으로, 많은 레코드가 감사 로그에 쓰여지며 이것은 데이터베이스 관리 프로그램의 성능을 저하시킵니다. 이러한 극단적인 경우는 시범 용도로만 표시되며, 위에서 표시한 명령이 있는 감사 기능을 구성하도록 권장하지 않습니다.

이 구성(『db2audit start』 사용)을 사용하여 감사 기능을 시작한 다음 *testapp* 응용프로그램을 실행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다. 로그에서 감사 레코드를 발췌하여, 응용프로그램이 수행한 두 가지 조치에 대해 생성된 다음과 같은 레코드를 볼 수 있습니다.

조치 작성된 레코드의 유형

CONNECT

```

timestamp=1998-06-24-08.42.10.555345;category=CONTEXT;
audit event=CONNECT;event correlator=2;database=FOO;
application id=*LOCAL.newton.980624124210;
application name=testapp;

timestamp=1998-06-24-08.42.10.944374;category=VALIDATE;
audit event=AUTHENTICATION;event correlator=2;event status=0;
database=FOO;userid=boss;authid=BOSS;execution id=newton;
application id=*LOCAL.newton.980624124210;application name=testapp;
auth type=SERVER;

timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=FOO;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
application name=testapp;auth type=SERVER;

timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=FOO;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
application name=testapp;auth type=SERVER;

timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=FOO;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
application name=testapp;auth type=SERVER;

timestamp=1998-06-24-08.42.11.622984;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=2;event status=0;
database=FOO;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
object name=FOO;object type=DATABASE;access approval reason=DATABASE;
access attempted=CONNECT;

timestamp=1998-06-24-08.42.11.801554;category=CONTEXT;
audit event=COMMIT;event correlator=2;database=FOO;userid=boss;
authid=BOSS;application id=*LOCAL.newton.980624124210;
application name=testapp;

timestamp=1998-06-24-08.42.41.450975;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=2;event status=0;
database=FOO;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;
object name=SQLC28A1;object type=PACKAGE;
access approval reason=OBJECT;access attempted=EXECUTE;

```

CREATE TABLE

```

timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;
audit event=EXECUTE_IMMEDIATE;event correlator=3;database=FOO;
userid=boss;authid=BOSS;application id=*LOCAL.newton.980624124210;
application name=testapp;package schema=NULLID;package name=SQLC28A1;
package section=203;text=create table audit(c1 char(10), c2 integer);

timestamp=1998-06-24-08.42.41.539692;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=FOO;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;

```

```
package schema=NULLID;package name=SQLC28A1;package section=0;
object schema=BOSS;object name=AUDIT;object type=TABLE;
access approval reason=DATABASE;access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.570876;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;
access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;
audit event=CREATE_OBJECT;event correlator=3;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object schema=BOSS;object name=AUDIT;object type=TABLE;
```

```
timestamp=1998-06-24-08.42.42.018900;category=CONTEXT;audit event=COMMIT;
event correlator=3;database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;
```

이와 같이, 가능한 모든 감사 이벤트 및 유형의 감사를 요청하는 감사 구성에서 생성된 중요한 몇 개의 감사 레코드가 있습니다.

대부분의 경우, 감사하려는 이벤트에 대해 더 제한되거나 초점이 있는 뷰의 감사 기능을 구성합니다. 예를 들어, 실패한 이벤트만을 감사하려면, 감사 기능은 다음과 같이 구성될 수 있습니다.

```
db2audit configure scope audit,checking,objmaint,secmaint,sysadmin,
validate status failure
```

주: 이 구성은 초기 감사 구성이거나 감사 구성이 재설정될 때 발생하는 구성입니다.

이 구성을 사용하여 감사 기능을 시작한 다음 *testapp* 응용프로그램을 실행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다(그리고 이전에 *testapp*를 실행하지 않은 것으로 가정합니다). 로그에서 감사 레코드를 발췌하여, 응용프로그램이 수행한 두 가지 조치에 대해 생성된 다음과 같은 레코드를 볼 수 있습니다.

조치 작성된 레코드의 유형

CONNECT

```
timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=F00;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
```

```

application name=testapp;auth type=SERVER;

timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=F00;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
application name=testapp;auth type=SERVER;

timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;
event status=-1092;database=F00;userid=boss;authid=BOSS;
execution id=newton;application id=*LOCAL.newton.980624124210;
application name=testapp;auth type=SERVER;

```

CREATE TABLE

(none)

이벤트 시도가 실패한 경우에만 가능한 모든 감사 이벤트(CONTEXT 제외)의 감사를 요청하는 감사 구성에서 생성된 훨씬 더 적은 감사 레코드가 있습니다. 사용자는 감사 구성을 변경하여 생성된 감사 레코드의 유형 및 특성을 제어할 수 있습니다.

감사 기능은 감사하려는 감사 레코드가 오브젝트에 대해 성공적으로 특권이 권한 부여됐을 때 감사 레코드를 작성할 수 있게 합니다. 이 경우, 다음과 같이 감사 기능을 구성할 수 있습니다.

```
db2audit configure scope checking status success
```

이 구성을 사용하여 감사 기능을 시작한 다음 *testapp* 응용프로그램을 실행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다(그리고 이전에 *testapp*를 실행하지 않은 것으로 가정합니다). 로그에서 감사 레코드를 발췌하여, 응용프로그램이 수행한 두 가지 조치에 대해 생성된 다음과 같은 레코드를 볼 수 있습니다.

조치 작성된 레코드의 유형

CONNECT

```

timestamp=1998-06-24-08.42.11.622984;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=2;event status=0;
database=F00;userid=boss;authid=BOSS;

timestamp=1998-06-24-08.42.41.450975;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=2;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;

```

```
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;
object name=SQLC28A1;object type=PACKAGE;
access approval reason=OBJECT;access attempted=EXECUTE;

timestamp=1998-06-24-08.42.41.539692;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=FOO;userid=BOSS;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object schema=BOSS;object name=AUDIT;object type=TABLE;
access approval reason=DATABASE;access attempted=CREATE;

timestamp=1998-06-24-08.42.41.570876;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=FOO;userid=BOSS;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;
access attempted=CREATE;
```

CREATE TABLE

(none)

제4부 데이터 이동

제7장 데이터 이동을 위한 유틸리티

LOAD 유틸리티는 데이터를 테이블로 이동시키고, 기존의 색인을 확장하며 통계를 생성합니다. LOAD는 대량의 데이터를 다룰 때 IMPORT 유틸리티보다 훨씬 빨리 데이터를 이동시킵니다. EXPORT 유틸리티를 사용하여 로드 해제된 데이터를 LOAD 유틸리티를 사용하여 로드할 수 있습니다.

AutoLoader 유틸리티는 대량의 데이터를 나누어서, 나뉜 데이터를 파티션된 데이터베이스의 다른 파티션으로 로드합니다.

IMPORT 및 EXPORT 유틸리티는 테이블이나 뷰 및 다른 데이터베이스나 스프레드시트 프로그램간에, DB2 데이터베이스간에, DB2 데이터베이스와 호스트 데이터베이스간에 DB2 연결을 사용하여 데이터를 이동시킵니다.

DPROPR(DataPropagator Relational)은 다른 DB2 관계형 데이터베이스의 다른 테이블에 대한 테이블 갱신사항을 자동으로 복사할 수 있는 DB2 Universal Database의 구성요소입니다.

주: 이들 주제에 관한 모든 정보와 *Command Reference* 및 *Administrative API Reference*의 비교 가능한 주제는 *데이터 이동 유틸리티 안내 및 참조서*에 포함되어 있습니다.

*데이터 이동 유틸리티 안내 및 참조서*는 이들 주제에 대한 1차 단일 정보 소스입니다.

제5부 복구

제8장 데이터베이스 복구

데이터베이스가 하드웨어 또는 소프트웨어 오류(아니면 둘 다)로 인해 사용할 수 없게 되고 여러 장애 상황은 여러 가지 복구 조치를 필요로 하는 경우가 있습니다. 이런 오류가 일어날 가능성에 대비하여 데이터베이스를 보호하는 적합한 전략이 필요합니다. 전략을 설계할 때에는 리허설이 또한 필요합니다. 리허설을 하면, 플랜의 단점을 감지하여 데이터베이스를 복구해야 할 경우 문제점이 발생하는 것을 피할 수 있습니다.

이 장에서는 데이터베이스와 관련된 문제가 있을 경우에 사용할 수 있는 여러 복구 방법에 관해 설명합니다. 또한, 사용자의 비즈니스 환경에 가장 적합한 복구 방법을 결정하는 데 도움이 될 고려사항과 의사결정에 대해서도 설명합니다. 각 복구 방법은 연관된 개념 및 이 방법을 지원하는 제품에 제공된 명령과 함께 설명되어 있습니다.

이 장에는 다음과 같은 내용이 들어 있습니다.

- 시스템 손상 복구
- 복구 방법: 버전 복구
- 복구 방법: 롤 포워드 복구
- Tivoli 저장영역 관리 프로그램
- 호스트에서 2단계 확약중 이상 실패 트랜잭션 복구.

DATALINK 컬럼이 들어 있는 테이블이 있다면 413 페이지의 『DB2 Data Links Manager 고려사항』에서 자세한 내용을 참조하십시오.

특정 시점까지의 롤 포워드 복구가 필요한 문제 유형 중 한 가지는 응용프로그램에서의 잘못된 논리 또는 틀린 입력으로 인한 데이터의 손상입니다. 응용프로그램이 데이터베이스 작업을 시작하는 시점에 가까운 시점까지 데이터베이스를 복구하기 위해 롤 포워드 복구 방법을 사용할 수 있습니다. 또는, 반대로 트랜잭션을 실행하여 데이터베이스에서 응용프로그램 영향을 받지 않으려고 시도할 수 있습니다.

두 번째 접근 방법을 따르기로 했을 경우, 주의를 해야 합니다. 이 장에서 응용프로그램 오류에 대한 설명은 자세히 제공하지 않습니다.

시스템 손상 복구

RESTART DATABASE 명령 또는 자동 재시작 활성화 구성 매개변수(*autorestart*)를 사용한 손상 복구는 데이터베이스가 일관되지 않고 사용 불가능한 상태가 되지 않도록 해줍니다.

다음 주제 항목에서는 추가 정보를 제공합니다.

- 일관성 있는 데이터베이스 확보
- 파티션된 데이터베이스 환경의 트랜잭션 오류 복구
- 실패한 데이터베이스 파티션 서버 식별.

일관성 있는 데이터베이스 확보

데이터베이스 명령과 응용프로그램은 여러 가지 이유로 인해 실패할 수 있습니다. 트랜잭션 실패는 잘못된 매개변수, 제한 초과, 교착상태로 인한 구간 복원에 의한 것일 경우 데이터베이스 조치 실패가 아닙니다. 그렇다기 보다는 데이터베이스 또는 데이터베이스 관리 프로그램이 비정상적으로 종료하도록 하는 심각한 오류이거나 조건으로서, 이 경우 데이터베이스를 복구해야 합니다. 예로는 머신의 전원 오류(데이터베이스 관리 프로그램과 데이터베이스 파티션이 중단되도록 하는) 또는 데이터베이스 로그를 담고 있는 디스크가 가득 찼기 때문에 데이터베이스가 중단되도록 하는 COMMIT/ROLLBACK 오류가 있으며, 어떠한 추가 로그 파일도 COMMIT/ROLLBACK 레코드 작성용으로 할당할 수 없습니다.

응용프로그램 또는 명령이 데이터베이스에 대해 수행중인 동안, 전원의 중단 또는 응용프로그램의 실패로 인해 데이터베이스에 대한 모든 활동이 즉각 중단될 수 있습니다. 하나 이상의 응용프로그램 또는 명령이 데이터베이스의 데이터에 대한 작업을 시작했지만 완료되지 않을 수 있습니다. 또한, 일부 확약된 작업 단위(UOW)가 디스크로 기록되지 않을 수 있습니다. 부분적으로 완료된(기록되지 않은) 작업 단위로 인해 데이터베이스가 불일치 상태 또는 사용 불가능 상태가 됩니다.

자세한 정보는 다음 주제 항목을 참조하십시오.

- 자동 재시작 사용 계획
- 자동 재시작 작동.

자동 재시작 사용 계획

실패시에 완료되지 않은 작업 단위(UOW)의 구간 복원을 데이터베이스 관리 프로그램이 자동으로 수행할 것인지 여부만 고려하면 됩니다. 그럴 경우, 자동 재시작 (autorestart) 구성 매개변수를 『ON』으로 설정하여 사용 가능하게 하십시오. (이 컨테이너는 기본값입니다.) 자동 재시작이 사용 가능하지 않은 경우에는 데이터베이스 장애가 발생할 때 RESTART DATABASE 명령을 발행할 준비가 되어 있어야 합니다.

주: 자동 재시작을 사용 가능하게 한 경우, 데이터베이스 장애 다음에 재시작이 시작됩니다. 재시작이 시간이 걸릴수 있으므로, 데이터베이스가 정지중이거나 응답없음 상태라고 생각하지 마십시오. 데이터베이스가 재시작할 때 db2diag.log 파일 레코드가 시작합니다. 항상 무엇이 일어나고 있는지 알수 있고, 데이터베이스가 응답하게 하려면, 자동 재시작을 사용하지 않는 것을 고려해야 합니다.

자동 재시작 작동

자동 재시작은 *autorestart* 데이터베이스 매개변수를 사용하여 작동될 수 있습니다. 이 매개변수의 기본값은 자동 재시작 『활성화(on)』 상태입니다. 이 매개변수에 대해서는 *관리 안내서*: 성능에서 자세한 내용을 참조하십시오.

파티션된 데이터베이스 환경의 트랜잭션 오류 복구

일반적으로, 데이터베이스 복구는 같은 트랜잭션 또는 응용프로그램에서 서로 관련된 오류가 발생한 데이터베이스 파티션 서버 및 다른 데이터베이스 파티션 서버 모두에서 필요합니다. 실패한 데이터베이스 파티션 서버의 데이터베이스 복구를 종종 손상 복구라고 합니다. 손상 복구는 오류를 유발한 조건이 정정된 후(예를 들어, 전원 공급이 다시 활성화된 경우) 실패한 데이터베이스 파티션 서버에서 수행됩니다. 다른(아직 사용중인) 데이터베이스 파티션 서버에서 데이터베이스 복구는 오류가 감지된 후에 즉시 발생합니다. 이 복구 프로세스에서, 데이터베이스 파티션 오류 복구라는 자원은 종종 실패한 트랜잭션 또는 응용프로그램에 대해 투명하게 정리됩니다.

『사용중인 데이터베이스 파티션 서버의 오류 복구』 및 351 페이지의 『오류 데이터베이스 파티션 서버에서의 트랜잭션 오류 복구』에서 자세한 내용을 참조하십시오.

2단계 확약 프로토콜

여기서 2단계 확약 프로토콜에 대해 논의하는 것은 파티션된 데이터베이스에서의 손상 복구에 대해 소개하기 위한 것입니다. 2단계 확약에 대해서는 **관리 안내서: 계획**에서 자세한 내용을 참조하십시오.

파티션된 데이터베이스 환경에서, 응용프로그램이 제출된 데이터베이스 파티션 서버는 조정자 노드이며, 응용프로그램에 대해 작동하는 첫번째 에이전트는 조정자 에이전트입니다. 조정자 에이전트는 다른 데이터베이스 파티션 서버로 작업을 분배해야 하며, 트랜잭션과 관련된 것이 어떤 것인지 추적합니다. 응용프로그램이 트랜잭션에 대해 COMMIT을 나타내는 경우, 조정자 에이전트는 2단계 확약 프로토콜을 사용하여 트랜잭션을 확약합니다. 1단계에서, 조정자 노드는 트랜잭션에 참여하고 있는 다른 모든 데이터베이스 파티션 서버에 PREPARE 요청을 분배합니다. 그런 다음, 이들 서버는 다음 중 하나로 응답합니다.

READ-ONLY	이 서버에서 데이터 변경이 발생하지 않았습니다.
YES	이 서버에서 데이터 변경이 발생했습니다.
NO	오류 때문에, 서버는 확약할 준비가 되어 있지 않습니다.

서버 중 하나가 『NO』로 응답할 경우, 트랜잭션은 구간 복원됩니다. 그렇지 않으면, 조정자 노드가 2단계를 시작합니다.

2단계에서 조정자 에이전트는 확약 로그 레코드를 기록한 다음, COMMIT 요청을 『YES』로 응답하는 모든 서버에 분배합니다. 다른 모든 데이터베이스 파티션 서버가 확약된 후, 이들은 조정자 노드로 COMMIT 확인을 전송합니다. 조정자 에이전트가 참여한 모든 서버로부터 COMMIT 확인을 수신하면 트랜잭션이 완료됩니다. 이때, 조정자 에이전트는 FORGET 로그 레코드를 기록합니다.

사용중인 데이터베이스 파티션 서버의 오류 복구

어떤 데이터베이스 파티션 서버가 다른 서버가 다운되었음을 발견하면, 고장난 데이터베이스 파티션 서버와 관련된 모든 작업이 중단됩니다.

- 계속 사용중인 데이터베이스 파티션 서버가 응용프로그램에 대한 조정자 노드이고 응용프로그램이 실패한 데이터베이스 파티션 서버(COMMIT 준비가 되지 않은)에서 실행중일 경우, 조정자 에이전트는 오류 복구로 인터럽트됩니다. 조정자 에이전트가 2단계 COMMIT 처리 상태일 경우, 응용프로그램은 SQL 오류 메시지 SQL0279N을 수신하고, 데이터베이스 연결이 해제됩니다. 그렇지 않으면, 조정자 에이전트는 트랜잭션에 참여하고 있는 다른 서버에 ROLLBACK 요청을 분배하고, 응용프로그램에는 SQL1229N이 표시됩니다.
- 실패한 데이터베이스 파티션 서버가 응용프로그램에 대한 조정자 노드라면, 사용중인 서버의 응용프로그램에 대해 작업중인 에이전트는 오류 복구를 하기 위해 인터럽트됩니다. 현재 트랜잭션이 준비되어 있으며 트랜잭션 결과를 기다리고 있지 않은 경우, 각 서버에서 지역으로 현재 트랜잭션이 구간 복원됩니다. 이 경우, 트랜잭션은 사용중인 데이터베이스 파티션 서버에서 2단계 확약중 이상 실패 상태로 남아 있으며, 조정자 노드는 이에 대해 알지 못합니다(사용할 수 없기 때문).

관리 안내서: 계획에서 2단계 확약중 이상 실패 트랜잭션의 해결 방법을 참조하십시오.

- 실패한 데이터베이스 파티션 서버에 응용프로그램이 연결되어 있지만(실패하기 전에), 지역 데이터베이스 서버나 실패한 데이터베이스 파티션 서버는 모두 조정자 노드가 아닐 경우, 이 응용프로그램에 대해 작업중인 에이전트가 인터럽트됩니다. 조정자 노드는 다른 데이터베이스 파티션 서버에 구간 복원 또는 연결해제 메시지를 송신합니다. 조정자 노드가 SQL0279를 리턴할 경우, 트랜잭션은 데이터베이스 파티션 서버에서만 2단계 확약중 이상 실패 상태입니다.

실패한 서버로 요청을 전송하고자 하는 모든 프로세스(예: 에이전트 또는 교착상태 검출기)에는 요청을 전송할 수 없다는 정보가 전달됩니다.

오류 데이터베이스 파티션 서버에서의 트랜잭션 오류 복구

프로세서가 재시작될 때 오류로 인해 데이터베이스 관리 프로그램이 비정상적으로 종료될 경우, DB2START를 RESTART 옵션과 함께 발행하여 데이터베이스 관리 프로그램을 재시작할 수 있습니다. 프로세서를 시작할 수 없으면 또한 DB2START를 사용하여 다른 프로세서에서 데이터베이스 관리 프로그램을 재시

작할 수 있습니다. 자세한 정보는 *Command Reference* 및 *Administrative API Reference*에 있는 START DATABASE MANAGER 명령 및 API를 참조하십시오.

비정상 종료로 인해 서버의 데이터베이스 파티션이 일관되지 않는 상태(즉, 사용할 수 없는 상태)가 될 수 있습니다. 이를 일관성을 유지하는 상태에서 사용 가능하게 하려면, 손상 복구가 필요합니다. 시스템 손상 복구는 데이터베이스 파티션 서버에서 다음과 같이 트리거될 수 있습니다.

- RESTART DATABASE 명령을 통해 명시적으로
- *autorestart* 데이터베이스 구성 매개변수가 활성화될 때 CONNECT 요청을 통해 내재적으로

데이터베이스의 모든 완전한 트랜잭션 효과를 보증하기 위해, 사용중인 로그 파일의 로그 레코드에 손상 복구가 다시 적용됩니다. 모든 변경사항이 다시 적용된 후, 2단계 확약중 이상 실패 트랜잭션을 제외한 확약되지 않은 모든 트랜잭션이 지역으로 구간 복원됩니다. 파티션된 데이터베이스 환경에서, 다음 두 가지 유형의 2단계 확약중 이상 실패 트랜잭션이 있습니다.

- 조정자 노드가 아닌 데이터베이스 파티션 서버에서, 트랜잭션이 준비는 되어 있지만 확약되어 있지는 않을 경우 트랜잭션은 2단계 확약중 이상 실패 상태입니다.
- 조정자 노드에서, 트랜잭션이 확약되어 있지만 완료된 것으로 기록되어 있지 않을 경우(즉, FORGET 레코드가 기록되어 있지 않을 경우), 트랜잭션은 2단계 확약중 이상 실패 상태입니다. 조정자 에이전트가 응용프로그램에 대해 작업하는 모든 서버로부터 모든 COMMIT 확인을 받지 않은 경우 이러한 상황이 발생합니다.

손상 복구는 다음 중 한 가지를 수행함으로써 모든 2단계 확약중 이상 실패 트랜잭션을 해결하고자 합니다. 수행된 조치는 데이터베이스 파티션 서버가 응용프로그램에 대한 조정자 노드였는지에 따라 달라집니다.

- 재시작된 서버가 응용프로그램에 대한 조정자 노드가 아닐 경우, 서버는 조정자 에이전트로 조회 메시지를 전송하여 트랜잭션의 결과를 알아냅니다.

- 재시작된 서버가 응용프로그램용 조정자 노드인 경우, 메시지를 조정자 에이전트가 여전히 COMMIT 승인을 기다리는 모든 다른 에이전트(종속 에이전트)로 송신합니다.

손상 복구를 통해 모든 2단계 확약중 이상 실패 트랜잭션을 해결할 수 없을 경우가 있습니다(예를 들면, 일부 데이터베이스 파티션 서버가 사용 가능하지 않을 경우). 이 경우, SQL 경고 메시지 SQL1061W가 리턴됩니다. 잠금 및 사용중인 로그 공간처럼 2단계 확약중 이상 실패 트랜잭션이 자원을 보유하는 것을 주의해야 합니다. 사용중인 로그 공간을 2단계 확약중 이상 실패 트랜잭션이 차지하고 있기 때문에, 데이터베이스에 아무런 변경도 가할 수 없는 시점에 이를 수 있습니다. 그렇기 때문에, 손상 복구 후 2단계 확약중 이상 실패 트랜잭션이 남아 있는지 조사해야 하며, 2단계 확약중 이상 실패 트랜잭션을 가능한 한 신속하게 해결하는 데 필요한 모든 데이터베이스 파티션 서버를 복구해야 합니다.

2단계 확약중 이상 실패 트랜잭션을 해결하는 데 필요한 하나 이상의 서버를 적시에 복구할 수 없으며 다른 서버의 데이터베이스 파티션에 액세스가 필요할 경우, 경험적 결정을 내려 수동으로 2단계 확약중 이상 실패 트랜잭션을 해결할 수 있습니다. LIST INDOUBT TRANSACTIONS 명령을 사용하여, 서버에서 2단계 확약중 이상 실패 트랜잭션을 조회, 확약, 구간 복원할 수 있습니다. 자세한 정보는 *Command Reference* 및 *Administrative API Reference* 매뉴얼에 있는 LIST INDOUBT TRANSACTIONS 명령 및 API를 참조하십시오.

주: LIST INDOUBT TRANSACTIONS 명령은 분산 트랜잭션 환경의 트랜잭션에도 사용됩니다. 두 가지 유형의 2단계 확약중 이상 실패 트랜잭션을 구분하기 위해 LIST INDOUBT TRANSACTIONS이 리턴한 출력에 있는 『originator』 필드에서 다음 중 하나를 표시합니다.

- 파티션된 데이터베이스 환경에서 발원된 트랜잭션을 나타내는 DB2 Universal Database Enterprise - Extended Edition
- 트랜잭션이 분산 환경에서 발생했음을 나타내는 XA

분산 환경에 대해서는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.

실패한 데이터베이스 파티션 서버 식별

데이터베이스 파티션 서버에 오류가 발생하면, 응용프로그램은 보통 다음의 SQLCODE 중 하나를 수신합니다. 실패한 데이터베이스 관리 프로그램을 발견하는 방법은 수신된 SQLCODE에 따라 다릅니다.

SQL0279N

이 SQLCODE는 트랜잭션에 속한 데이터베이스 파티션 서버가 COMMIT 처리중에 종료되면 수신됩니다.

SQL1224N

이 SQLCODE는 실패한 데이터베이스 파티션 서버가 트랜잭션에 대한 조정자 노드인 경우 수신됩니다.

SQL1229N

이 SQLCODE는 실패한 데이터베이스 파티션 서버가 트랜잭션에 대한 조정자 노드가 아닌 경우 수신됩니다.

데이터베이스 파티션 서버가 실패한 것을 결정하는 것은 두 단계의 프로세스입니다. SQLCODE SQL1229N과 연관된 SQLCA에는 *sqlerrd* 필드의 6번째 배열 위치에서 오류를 검출한 서버의 노드 번호가 들어 있습니다(서버용으로 기록된 노드 번호는 *db2nodes.cfg* 파일의 노드 번호와 일치합니다). 오류를 검출한 데이터베이스 파티션 서버에서는, 실패한 서버의 노드 번호를 나타내는 메시지가 *db2diag.log* 파일에 기록됩니다.

주: 다중 논리 노드가 프로세서에서 사용중이면, 하나의 논리 노드의 오류는 동일한 프로세서에서 다른 논리 노드가 실패하도록 합니다.

데이터베이스 파티션 서버의 오류를 복구하려면, 다음을 수행하십시오.

1. 오류를 일으킨 문제점을 정정하십시오.
2. 모든 데이터베이스 파티션 서버에서 DB2START 명령으로 데이터베이스 관리 프로그램을 재시작하십시오.
3. 실패한 데이터베이스 파티션 서버에서 RESTART DATABASE 명령으로 데이터베이스를 재시작하십시오.

복구 방법: 버전 복구

RESTORE 명령과 함께 BACKUP 명령을 사용하는 버전 복구는 데이터베이스 또는 테이블 공간을 이전 저장 상태로 복구합니다. 복구 불가능한 데이터베이스(즉, 보존 로그가 없는 데이터베이스)의 경우 이 복구 방법을 사용할 수 있습니다. WITHOUT ROLLING FORWARD 옵션을 사용하여 복구 가능한 데이터베이스에 이 방법을 사용할 수도 있습니다.

이 절에서는 고려사항의 플랜과 이 방법을 수행하는 특정 유틸리티 또는 명령을 호출하는 방법을 검토합니다. 그런 다음, 이 방법을 효율적으로 사용할 수 있는 개념 또는 관련된 문제를 제시합니다.

다음 주제 항목에서는 추가 정보를 제공합니다.

- 데이터베이스 백업
- 데이터베이스 복원
- 복구 실행기록 파일 정보

데이터베이스 백업

데이터베이스의 백업 사본을 작성하려면, BACKUP 명령 또는 제어 센터를 사용합니다. 제어 센터내에서, 백업될 데이터베이스를 선택하고 백업 조치를 선택합니다.

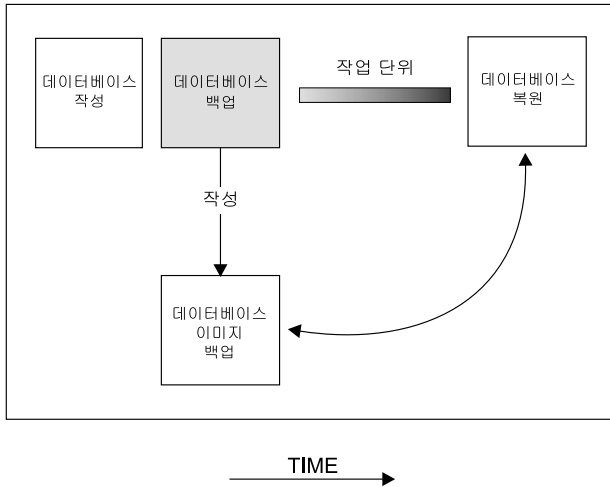


그림 8. 데이터베이스 이미지 작성

파티션된 데이터베이스 시스템에서는, `BACKUP DATABASE` 명령을 사용하여 데이터베이스 파티션을 개별적으로 백업합니다. 명령을 나타내는 데이터베이스 파티션 서버에 대한 지역 조작입니다. 그러나, 인스턴스의 데이터베이스 파티션 서버 중 하나에서 `db2_all`을 발행하여 서버 목록에서 해당 노드 번호로 식별하는 `BACKUP` 명령을 제출할 수 있습니다. 이렇게 하려면, 카탈로그 노드를 먼저 백업하고 다른 데이터베이스 파티션을 백업해야 합니다. 또한, 제어 센터를 사용하여 데이터베이스 파티션을 백업할 수도 있습니다.

파티션된 데이터베이스에서, `LIST NODES` 명령을 사용하여 사용자 테이블이 있는 노드 목록(데이터베이스 파티션 서버)을 결정할 수 있습니다. 이 복구 방법은 롤 포워드 복구 기능을 지원하지 않기 때문에, 이 노드 목록에서 정기적으로 데이터베이스를 백업하십시오.

분산 요청 시스템에서, `BACKUP` 및 `RESTORE` 명령은 데이터베이스 카탈로그에 저장된 미디어(랩퍼, 서버, 별명 등) 및 분산 요청 데이터베이스에 적용됩니다. 데이터 소스 오브젝트(테이블 및 뷰)는 이러한 오브젝트가 분산 요청 데이터베이스에 저장되지 않으면 백업되거나 복원되지 않습니다.

사용할 복구 방법을 기억하십시오. 다음 절에서는 이 태스크에 적용되는 요구사항 및 기타 고려사항을 제공합니다.

- `BACKUP` 명령 사용 계획
- `BACKUP` 명령 호출
- `BACKUP`으로 작성된 백업 이미지

주: 다음 절에서 『페이지』를 설명할 때 백업 및 복원 유틸리티에 특정한 페이지를 참조합니다. 이 페이지는 항상 4KB 크기이어야 하며 데이터베이스 데이터용으로 허용된 여러 페이지 크기와 혼동하지 말아야 합니다.

BACKUP 명령 사용 계획

계획시 고려해야 할 사항은 다음과 같습니다.

- BACKUP 명령을 사용하기 위해서는 SYSADM, SYSCTRL 또는 SYSMAINT 권한을 가져야 합니다.
 - 데이터베이스는 지역 또는 원격이 될 수 있습니다. Tivoli* 저장영역 관리 프로그램(TSM)과 같은 저장영역 관리 프로그램 제품이 사용되지 않은 경우 백업은 데이터베이스 서버에 남아 있습니다.
 - 고정 디스크, 테이프 또는 TSM 유틸리티나 다른 벤더 저장영역 관리프로그램 제품이 관리하는 위치에 데이터베이스 또는 테이블 공간을 백업할 수 있습니다. TSM에 대해서는 425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오.
- OS/2에서, 디스켓 또는 User Exit 프로그램으로 백업할 수도 있습니다.

주: OS/2에서, User Exit 프로그램으로 데이터베이스를 온라인 백업할 때에는 백업이 시작되기 전에 해당 데이터베이스가 quiesce 상태로 된다는 점을 주의하십시오. 이와 같이, 백업은 시작 전에 모든 트랜잭션이 확약 또는 구간 복원되기를 기다립니다. 백업이 수행되는 동안 모든 새로운 트랜잭션은 백업이 완료될 때까지 기다리고, 백업이 일단 완료되면 모든 트랜잭션은 정상 시처럼 처리를 계속합니다.

- 백업 이미지를 작성할 때(또는 백업 이미지를 복원할 때) 버퍼 크기는 1024페이지(4KB 크기)입니다. 백업 장치로서 테이프를 사용하는 경우 이것은 중요합니다. 변수 블록 크기를 사용할 때 반드시 테이프 드라이브가 사용하는 범위로 버퍼 크기를 더 줄어야 합니다.
- 지원되는 Windows 운영 체제에서, 디스켓으로 백업할 수 있습니다.
- SCSI 테이프 장치로의 백업 및 복원을 위해 DB2의 기본 버퍼 크기를 사용하는 대부분의 Linux 버전 사용자는 SQL2025N의 오류 메시지와 이유 코드 『75』의 결과를 갖습니다. Linux 내부 SCSI 버퍼의 오버플로우를 방지하려면, 다음 형식을 사용하십시오.

`bufferpages <= ST_MAX_BUFFERS * ST_BUFFER_BLOCKS / 4`

여기서 `bufferpages`는 `backbufsz` 또는 `restbufsz` 중 하나의 값입니다. `ST_MAX_BUFFERS` 및 `ST_BUFFER_BLOCKS`는 `drivers/scsi` 디렉토리에서 Linux 커널에 정의됩니다.

- OS/2에서, 운영 체제에 원시 테이프 지원이 없어 테이프 백업 중인 경우 User Exit 프로그램이 사용됩니다.

UNIX 운영 체제 및 Windows NT에서, 원시 테이프 지원은 사용 가능합니다.

주: 테이프 장치에서 가변 블록 크기를 사용하는 경우, DB2 버퍼 크기는 장치가 구성되는 최대 가변 블록 크기와 같거나 작습니다. 그렇지 않으면, 백업은 계속되지만 결과 이미지는 복구 가능한 것으로 인정되지 않습니다.

- 데이터베이스의 백업 데이터가 들어가도록 여러 파일을 작성할 수 있습니다.
- 온라인 백업의 완료 시, 사용 중인 로그가 닫히며 디스크에 쓰여집니다.
- 파티션된 데이터베이스 환경에서, 오프라인 백업은 해당 데이터베이스 파티션 서버에서 데이터베이스로의 독점적 연결을 사용하므로(즉, 조작식 데이터베이스 파티션으로 독점적 연결이 필요함) 다른 응용프로그램은 데이터베이스 파티션으로 연결될 수 없습니다. 카탈로그 노드의 오프라인 백업을 수행할 경우, 비카탈로그 데이터베이스 파티션 서버의 데이터베이스 백업을 포함하여 전체 데이터베이스에 어떠한 활동도 없습니다. `db2_a11`을 사용하여 데이터베이스를 백업할 수 있습니다. 그러나, 카탈로그 노드가 먼저 백업되었는지 확인해야 합니다. 카탈로그 노드가 백업된 후에, 다른 데이터베이스 파티션이 동시에 백업될 수 있습니다.
- 파티션된 데이터베이스 시스템에서 또한 사용자는 백업 사본으로 `db2nodes.cfg` 파일의 사본을 보존하여, 이 파일이 손상되는 경우에 대비해야 합니다.

ATALINK 컬럼이 들어 있는 테이블이 있다면, 414 페이지의 『백업 유틸리티 고려사항』에서 자세한 내용을 참조하십시오.

테이프 장치를 사용하려면, SCO UnixWare 7상의 DB2 사용자는 BUFFER를 16으로 지정해야 합니다. BUFFER의 기본값은 1024페이지입니다. BUFFER가 0으로 설정되면, 데이터베이스 관리 프로그램 구성 매개변수 `backbufsz`는 16으로 설정되어야 합니다.

백업할 때 사용자 테이프 계획: 테이블 공간이나 데이터베이스를 백업할 때, 블록 크기와 사용자 버퍼 크기를 제대로 설정해야 합니다. 변수 블록 크기를 사용 중 일 때에는 특히 그렇습니다(예를 들어, 블록 크기가 0으로 설정되어 있을 때 AIX에서).

백업할 때 사용될 수 있는 FIXED 블록 크기의 수에 제한사항이 있습니다. DB2는 4KB 블록으로서 백업 이미지 헤더를 작성하므로 이 제한사항이 있습니다. DB2가 지원하는 유일한 고정 블록 크기는 512, 1024, 2048 및 4096바이트입니다. 고정된 블록 크기를 사용 중이라면, 백업용 버퍼 크기를 지정할 수 있습니다. 그러나, 고정된 블록 크기가 여기에서 언급한 것 중 하나가 아니라면 백업이 정상적으로 완료되지 않습니다.

데이터베이스 데이터가 클때, 위에서 언급한 고정된 블록 크기를 사용하면 백업이 장시간 걸립니다. 변수 블록 크기의 사용을 고려할 수 있습니다.

변수 블록 크기를 사용할 때, 사용 중인 테이프 장치의 최대 한계보다 적거나 같은 버퍼 크기를 BACKUP 명령에 지정해야 합니다. 버퍼 크기는 최적 성능을 원하는 경우 사용 중인 장치의 최대 블록 크기 한계와 같아야 합니다.

블록 크기가 변수인 백업 이미지에서 복원하면 오류를 리턴할 수 있음에 주의하십시오. 오류가 발생하면, 적절한 블록 크기를 사용하여 이미지를 재작성할 필요가 있습니다. AIX로 이를 수행하는 예는 다음과 같습니다.

```
tcl -b 0 -Bn -f /dev/rmt0 read > backup_filename.file
dd if=backup_filename.file of=/dev/rmt0/ obs=4096 conv=sync
```

이것은 『backup_filename.file』이라고 하는 파일에 백업 이미지를 덤프합니다. 그러면 『dd』 명령은 4096의 블록 크기를 사용하여 테이프에 다시 이미지를 덤프합니다.

이미지가 파일로 덤프하기에 너무 큰 경우 정정 방법이 복잡합니다. 그렇게 큰 이미지에서의 가능한 한가지 솔루션은 하나의 테이프 장치에서 또다른 테이프 장치로 이미지를 덤프하기 위해 『dd』 명령을 사용하는 것입니다. 이것은 이미지가 둘 이상의 테이프에 걸쳐 있지 않으면 작동합니다. 두 개의 테이프 장치를 사용할 때, 『dd』 명령은 다음과 같습니다.

```
dd if=/dev/rmt1 of=/dev/rmt0 obs=4096
```

두 개의 테이프 장치 사용이 가능하지 않은 경우, 『dd』 명령을 사용하여 원시 장치로 이미지를 덤프한 후 원시 장치에서 테이프로 이미지를 덤프할 수 있습니다. 이 방법을 사용하기 어려운 점은 『dd』 명령이 반드시 원시 장치로 덤프되는 블록 수의 트랙을 유지해야 한다는 것입니다. 덤프되는 블록의 수는 이미지가 다시 테이프에 이동될 때 지정될 필요가 있습니다. 『dd』 명령이 원시 장치에서 테이프에 이미지를 덤프하기 위해 사용될 때, 명령은 원시 장치의 전체 크기를 테이프에 덤프합니다. 『dd』 명령은 얼마나 많은 원시 장치가 이미지를 보유하기 위해 사용되는가를 알려줄 수 없습니다.

BACKUP 명령을 사용할 때 사용 중인 테이프 장치의 최대 블록 크기 한계를 알 필요가 있습니다. 다음과 같은 몇 가지 예가 있습니다.

장치	접속	블록 크기 한계	DB2 버퍼 크기 한계(4KB 페이지)
8 mm	scsi	131 072	32
3420	s370	65 536	16
3480	s370	65 536	16
3490	s370	65 536	16
3490E	s370	65 536	16
7332 (4 mm)*	scsi	262 144	64
3490e	scsi	262 144	64
3590**	scsi	2 097 152	512
3570 (magstar MP)		262 144	64

주:

- * 7332는 블록 크기 한계를 구현하지 않습니다. 256KB는 단순히 제시된 값입니다. 블록 크기 한계는 상위 어댑터에 의해 부과됩니다.
- ** 3590이 2MB 블록 크기를 지원하지만, 더 낮은 값을 가질 때(256K와 같이) 성능이 사용자 요구에 적합함을 경험할 수 있습니다.
- 장치 문서 및/또는 장치 벤더로 장치 한계를 점검하십시오.

BACKUP 명령 호출

다음 고려사항은 BACKUP 명령을 수행할 때 유용합니다.

- BACKUP 명령 또는 API를 수행하기 전에 데이터베이스 관리 프로그램 (DB2START)을 먼저 시작해야 합니다. 제어 센터를 사용할 때에는, 데이터베이스 관리 프로그램을 명시적으로 시작할 필요가 없습니다.
- 제어 센터에서 명령, API 또는 타스크를 사용할 경우, 데이터베이스 이름 자체 가 아니라 데이터베이스 별명을 지정해야 합니다.
- 백업하는 데 필요한 시간을 줄이려면, 다음을 수행하십시오.

- PARALLELISM 매개변수의 값을 증가시키십시오.

이 매개변수를 사용하면 백업을 완료하는 데 필요한 시간을 현저히 줄일 수 있습니다. PARALLELISM 매개변수는 데이터베이스로부터 데이터를 읽기 위해 시작된 프로세스 또는 스레드의 수를 정의합니다. 특정 테이블 공간을 백업하기 위해 각 프로세스 또는 스레드가 할당됩니다. 테이블 공간 백업이 완료되면, 이 매개변수는 다른 것을 요청합니다. 그러나, 각 프로세스 또는 스레드에는 메모리 및 CPU 오버헤드가 모두 필요하다는 점을 유의하십시오. 로드가 많은 시스템의 경우, PARALLELISM 매개변수를 기본값인 1로 두어야 합니다.

- 백업 버퍼 크기를 증가시키십시오.
- 버퍼의 수를 증가시키십시오.

다중 버퍼 및 입출력 채널을 사용하려면, 적어도 채널 수의 두 배의 버퍼를 사용하여 채널이 데이터를 대기할 필요가 없음을 확인해야 합니다. 사용되는 버퍼의 크기 또한 백업 조작의 성능에 도움이 됩니다. 이상적인 백업 버퍼 크기는 테이블 공간 크기의 배 수가 되어야 합니다.

다른 확장 크기를 가진 다중 테이블 공간이 있으면, 최대 확장 크기의 배수인 값을 지정하십시오.

BACKUP 명령을 시작할 때 각 백업 버퍼에 사용할 페이지 수를 지정할 수 있습니다. 최소 페이지 수는 16입니다. 페이지 수를 지정하지 않으면 각 버퍼는 데이터베이스 관리 프로그램 구성 매개변수 *backbufsz*에 기초하여 할당됩니다. 버퍼를 할당하기 위해 사용할 수 있는 메모리가 충분하지 않으면 오류가 리턴됩니다.

이 구성 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

- 다중 목표 장치를 사용하십시오.

- OS/2에서, 테이프와 같이 이동 가능한 미디어로 데이터베이스를 백업할 때, 데이터베이스 관리 프로그램은 미디어 볼륨 1에 정보를 기록합니다. 일단 첫번째 미디어가 드라이브에 위치하면, 운영 체제 백업 기능이 사용자에게 미디어 2를 요구할 때까지, 미디어를 제거하지 마십시오.
- 백업 보류 상태에 있는 데이터베이스를 제외한, 사용 가능 상태에 있지 않는 데이터베이스는 백업할 수 없습니다.
 - 데이터베이스가 데이터베이스 복원 단계에서의 시스템 이상으로 인해 부분적으로만 복원된 상태인 경우, 해당 데이터베이스를 백업하려면 성공적으로 복원해야 합니다.
 - 데이터베이스가 이전 릴리스의 데이터베이스 관리 프로그램으로 작성되고 데이터베이스가 이주되지 않은 경우, 데이터베이스를 이주시켜야 백업할 수 있습니다.
데이터베이스 이주에 대해서는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.
 - 데이터베이스 내의 테이블 공간이 『비정상』 상태이면, 백업 보류 상태가 아닌 해당 데이터베이스를 백업할 수 없습니다.
- 데이터베이스 백업의 중요한 단계에서 시스템 고장이 발생하면, BACKUP 명령을 다시 발행하여 백업을 성공적으로 완료할 때까지 데이터베이스에 성공적으로 연결할 수 없습니다.
- BACKUP 명령은 서로 다른 데이터베이스의 백업 사본을 작성하는 여러 프로세스에 동시처리 제어를 제공합니다. 이러한 제어를 통해 전체 백업 프로세스가 종료될 때까지 백업 목표 장치가 보존됩니다.
백업 프로세스중에 오류가 발생하고 열기 컨테이너가 닫히지 않는 경우, 같은 대상 드라이브에 대한 기타 백업 프로세스는 액세스 오류를 수신합니다. 액세스 오류를 정정하려면, 오류를 일으킨 백업 프로세스에서 완전히 나간 후 목표 장치로부터 연결해제해야 합니다.
- 테이프의 동시 백업 프로세스에 대해 BACKUP 명령을 사용하고 있는 경우, 프로세스가 동일한 테이프를 목표로 하지 않도록 해야 합니다.

BACKUP으로 작성된 백업 이미지

백업 이미지는 사용자가 BACKUP 명령을 다음에서 호출할 때 지정된 목표 장치 또는 머신에서 작성됩니다.

- 디스크 또는 디스켓 백업을 위한 디렉토리
- 테이프 백업을 위해 지정된 장치
- Tivoli 저장영역 관리 프로그램(TSM) 서버
- 다른 벤더의 서버
- OS/2의 경우, User Exit 프로그램 사용

복구 실행기록 파일은 전체 데이터베이스의 백업 또는 복원을 수행할 때마다 요약 정보를 통해 자동으로 갱신됩니다. 이 파일은 데이터베이스 내의 활동을 복원할 유용한 추적 매커니즘이 될 수 있습니다. 이 파일은 데이터베이스 구성 파일과 같은 디렉토리에 작성됩니다. 복구 실행기록 파일에 대해서는 406 페이지의 『복구 실행기록 파일 정보』에서 자세한 내용을 참조하십시오.

UNIX 기반 환경에서, 디스크에서 작성된 파일 이름은 마침표로 분리된 다음 정보를 결합시켜 구성됩니다. 기타 플랫폼에서는 4 레벨 서브디렉토리 트리가 사용됩니다.

데이터베이스 별명	백업 명령이 시작될 때 제공된 1-8자의 데이터베이스 별명
유형	수행된 백업 유형. 여기서, 『0』은 전체 데이터베이스를 나타냅니다. 『3』은 테이블 공간 백업용입니다. 『4』는 LOAD...COPY TO 명령에 의해 생성된 백업용입니다.
인스턴스 이름	DB2INSTANCE 환경 변수에서 취한 데이터베이스 관리 프로그램의 현재 인스턴스의 1-8자의 문자 이름
노드 번호	노드 번호
카탈로그 노드 번호	데이터베이스 카탈로그 노드의 노드 번호
시간소인	백업이 수행된 날짜와 시간의 14자 표현. 시간소인은 <code>yyyymmddhhnnss</code> 형식입니다. 여기서,

yyyy는 연도(1995 - 9999)

mm은 월(01 - 12)

dd는 월의 날짜(01 - 31)

hh는 시간(00 - 23)

mn은 분(00 - 59)

ss는 초(00 - 59)

순차 번호

파일 확장자로 사용되는 3자릿수의 순차 번호

UNIX 기반 운영 체제에서, 형식은 다음과 같이 나타납니다.

```
Database alias.Type.Instance name.NODEEnnnn  
.CATNnnnn.timestamp.number
```

기타 운영 체제에서, 형식은 다음과 같이 나타납니다.

```
Database alias.Type\Instance name.NODEEnnn  
\CATNnnn\yyyymmdd\hhmmss.number
```

예를 들어, UNIX 환경의 경우 DB201 인스턴스에서 STAFF로 이름 지정된 데이터베이스를 디스크에서 다음과 같은 이름의 파일로 백업할 수 있습니다.

```
STAFF.0.DB201.NODE0000.CATN0000.19950922120112.001
```

테이프 직접 출력에서, 파일 이름이 작성되지 않았습니다. 그러나, 위의 정보는 나중에 검증 목적으로 백업 헤더에 저장됩니다.

주:

1. 데이터베이스 백업 및 복원 작업을 위해 테이프 미디어를 사용하려면, 표준 운영 체제 인터페이스를 통해 테이프 장치를 사용할 수 있어야 합니다. 그러나, 파티션된 대형 데이터베이스에서는 각 데이터베이스 파티션 서버 전용의 테이프 장치를 보유하는 것은 실용적이지 않습니다. 테이프 장치를 하나 이상의 TSM 서버에 연결하여 각 데이터베이스 파티션 서버가 이 테이프 장치에 액세스하도록 할 수 있습니다.
2. 파티션된 데이터베이스 시스템에서, REELlibrarian 4.2 또는 CLIO/S와 같이 가상 테이프 장치를 제공하는 제품을 사용할 수도 있습니다. 이 제품을 사용하여 의사 테이프 장치를 통한 다른 노드(데이터베이스 파티션 서버)에 연결된

테이프 장치를 액세스합니다. 원격 테이프 장치로의 액세스는 투명하게 제공되며, 표준 운영 체제 인터페이스를 통해 의사 테이프 장치에 액세스할 수 있습니다.

백업 정보 표시

존재하는 백업 이미지에 대한 정보를 표시하는 백업 유틸리티가 있습니다. 이 유틸리티의 이름은 `db2ckbkp`이며 다음을 수행하도록 허용합니다.

- 백업 이미지의 무결성을 테스트하며 복원할 수 있을 지 여부를 판별합니다.
- 백업 헤더에 저장된 백업에 대한 정보를 표시합니다.

이 유틸리티를 사용할 때 지정하는 백업 이미지에 읽기 사용권한을 가져야 합니다.

백업 이미지의 존재를 간단히 검증하기 위해, 다음과 같이 유틸리티를 사용할 수 있습니다.

```
db2ckbkp STAFF.0.DB201.NODE0000.CATN0000.19950922120112.001
```

이 유틸리티의 출력은 다음과 유사합니다.

```
[1] Buffers processed: ##  
  
Image Verification Complete - successful.
```

*Command Reference*에서 이 유틸리티에 대한 자세한 내용을 참조하십시오.

데이터베이스 복원

다음 절에서는 RESTORE 명령에 적용되는 요구사항 및 기타 고려사항을 제공합니다.

- RESTORE 명령 사용 계획
- RESTORE 명령 호출
- RESTORE중 테이블 공간 컨테이너 재정의
- 기존 데이터베이스로 복원
- 새로운 데이터베이스로 복원.

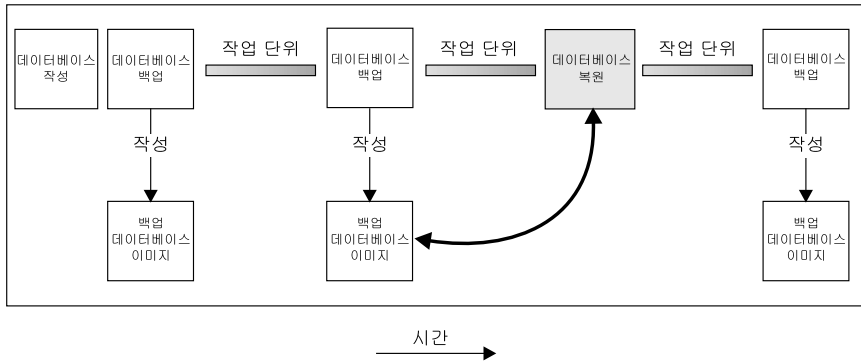


그림 9. 백업 이미지를 사용한 데이터베이스 복원

RESTORE 명령 사용 계획

다음 사항을 고려하십시오.

- 전체 데이터베이스 백업을 통해 기존 데이터베이스로 복원하려면 SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다. 새로운 데이터베이스로 복원하려면, SYSADM 또는 SYSCTRL 권한이 있어야 합니다.
- 데이터베이스가 이전에 BACKUP 명령을 통해 백업된 경우에만 이 명령을 사용할 수 있습니다.
- 제어 센터를 사용하는 경우, 현재 DB2 버전 이전에 실행된 백업을 복원할 수 없습니다.
- OS/2에서, RESTORE 명령은 User Exit 프로그램이 데이터베이스를 백업하는 데 사용된 경우에만, User Exit 프로그램을 호출할 수 있습니다.
- 수행될 복원 유형은 복원시에 선택할 수 있습니다. 다음 유형 중에서 선택할 수 있습니다.
 - 백업에서 모든 내용의 완전 복원
 - 복구 실행기록 파일만의 복원
 - 백업에서 테이블 공간의 부속 집합
- RESTORE 명령은 TSM(Tivoli Storage Manager) 유틸리티를 사용할 수 있으며, 이 유틸리티의 제한사항도 고려해야 합니다(425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오).

- 다른 벤더 저장영역 관리 프로그램 제품은 해당 제품을 사용하여 원래 백업을 저장하는 경우에 사용될 수도 있습니다.
- 데이터베이스 복원에서는 독점적인 연결이 필요합니다. 즉, 타스크가 시작될 때, 어떠한 응용프로그램도 데이터베이스에 대해 실행될 수 없습니다. 일단 타스크가 시작되면, 복원이 완료될 때까지 다른 응용프로그램이 데이터베이스에 액세스할 수 없습니다. 테이블 공간 복원은 온라인으로 수행될 수 있습니다.
- 데이터베이스는 지역 또는 원격이 될 수 있습니다.
- WITHOUT DATALINK 옵션이 지정되지 않고 DATALINK 데이터가 들어 있는 DB2 Data Links Manager를 사용할 수 없으면, 복원 조작은 실패합니다.
 옵션이 지정되고 DATALINK 데이터가 들어 있는 DB2 Data Links Manager를 사용할 수 있으면, 사용할 수 없는 서버에 대해 DATALINK 값을 가진 테이블이 들어 있는 모든 테이블 공간은 RESTORE PENDING 상태가 됩니다. 이러한 테이블 공간은 Data Links 서버가 사용 가능하게 되면 다시 복원되어야 합니다.
- 테이프 장치를 사용하려면, SCO UnixWare 7상의 DB2 사용자는 BUFFER를 16으로 지정해야 합니다. BUFFER의 기본값은 1024페이지입니다. BUFFER가 0으로 설정되면, 데이터베이스 관리 프로그램 구성 매개변수 *backbufsz*는 16으로 설정되어야 합니다.

ATALINK 컬럼이 들어 있는 테이블이 있을 경우, 415 페이지의 『복원 및 롤 포워드 유틸리티 고려사항』 및 417 페이지의 『롤 포워드 없이 오프라인 백업에서 데이터베이스 복원』에서 자세한 내용을 참조하십시오.

RESTORE 명령 호출

다음 고려사항은 RESTORE 명령을 수행할 때 유용합니다.

- 데이터베이스 관리 프로그램은 데이터베이스를 복원하기 전에 시작되어야 합니다.
- 데이터를 복원하는 데이터베이스는 데이터를 원래 백업한 데이터베이스와 같은 데이터베이스일 수도 있고 다른 데이터베이스일 수도 있습니다. 데이터를 새로운 데이터베이스 또는 기존 데이터베이스로 복원할 수 있습니다.

- 복원 절차시, 사용자는 선택적으로 복원 절차의 성능을 개선하기 위해 다중 버퍼를 사용하도록 선택하는 기능을 가집니다. 다중 내부 버퍼는 백업 미디어의 데이터로 채워질 수 있습니다.

RESTORE 명령을 시작할 때 각 복원 버퍼에 사용할 페이지 수를 지정할 수 있습니다. 지정하는 값은 백업 버퍼용으로 지정한 다중 페이지 수여야 합니다. 최소 페이지 수는 16입니다. 페이지 수를 지정하지 않으면, 데이터베이스 관리 프로그램 구성 매개변수 *restbufsz*에 기초하여 각 버퍼가 할당됩니다. 버퍼를 할당하기 위해 사용할 수 있는 메모리가 충분하지 않으면 오류가 리턴됩니다.

이 구성 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

- RESTORE DATABASE 명령의 TAKEN AT 매개변수에는 백업에 대한 시간소인이 필요합니다. 시간소인은 BACKUP 명령이 성공적으로 완료된 후에, 즉 *yyyymmddhhmmss* 형식으로 표시된 것과 정확하게 같습니다.

또한, 부분적인 시간소인을 지정할 수 있습니다. 예를 들어, 시간소인 19971001010101 및 19971002010101을 가진 두 개의 다른 백업이 있다고 가정하십시오. TAKEN AT에 대해 19971002를 지정하는 경우, 19971002010101 백업이 사용됩니다.

TAKEN AT이 지정되지 않을 경우, 소스 미디어에 한개의 백업만이 존재해야 합니다.

- RESTORE 명령에 의해 사용된 데이터베이스의 백업 사본은 하드 디스크, 테이프 또는 TSM(Tivoli* Storage Manager) 유틸리티나 다른 벤더 저장영역 관리 프로그램 제품에 의해 관리되는 위치로 지정될 수 있습니다. TSM에 대해서는 425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오.

TSM을 사용하며 TAKEN AT 매개변수를 지정하지 않은 경우, TSM은 최신 백업 사본을 검색합니다.

OS/2에서, 데이터베이스의 백업 사본은 디스켓 또는 User Exit 프로그램을 통해 배치될 수도 있습니다.

Windows 운영 체제 시스템에서, 데이터베이스의 백업 사본은 디스켓 위치를 지원합니다.

- Windows 운영 체제 및 OS/2에서, 『TO 대상 디렉토리』 절을 사용할 때 장치 이름만을 지정해야 합니다. 더 긴 경로가 지정되면, 오류가 리턴됩니다.
- 일단 RESTORE 명령이 시작되면, RESTORE 명령이 정상적으로 완료될 때까지 데이터베이스를 사용할 수 없습니다.
- 데이터베이스 복원중 어떤 단계에서 시스템 고장이 발생하면 RESTORE 명령을 다시 사용하여 복원을 성공적으로 완료할 때까지 데이터베이스에 연결할 수 없습니다.
- 복원중인 데이터베이스의 코드 페이지가 응용프로그램에 대해 사용할 수 있는 코드 페이지와 일치하지 않거나, 데이터베이스 관리 프로그램이 데이터베이스 코드 페이지에서 응용프로그램에 대해 사용할 수 있는 코드 페이지로의 코드 페이지 변환을 지원하지 않으면 복원되는 데이터베이스는 사용할 수 없게 됩니다.

RESTORE중 테이블 공간 컨테이너 재정의

데이터베이스의 백업중에, 레코드는 백업된 테이블 공간이 사용하는 모든 테이블 공간 컨테이너를 보유하고 있습니다. RESTORE중, 백업에 나열되어 있는 모든 컨테이너는 현재 존재하는지 액세스 가능한지 알기 위해 점검됩니다. 미디어 오류(또는 다른 이유로) 때문에 하나 이상의 컨테이너에 액세스할 수 없을 경우, RESTORE는 실패합니다. 이러한 경우에, 복원을 허용하기 위해 테이블 공간 컨테이너 경로 재지정이 RESTORE 동안에 지원됩니다. 이 지원에는 테이블 공간 컨테이너의 추가, 변경 또는 제거가 포함됩니다.

백업에 나열된 컨테이너가 시스템에 존재하지 않더라도, 복원하려는 경우가 있습니다. 이러한 경우의 예는 백업이 수행된 시스템이 아닌 다른 시스템상의 손상으로부터 복구하려는 경우입니다. 새로운 시스템에는 필수 컨테이너가 정의되어 있지 않을 수도 있습니다. 이 경우, RESTORE를 허용하기 위해 다른 컨테이너로의 RESTORE시 테이블 공간 컨테이너의 경로 재지정이 지원됩니다.

두 가지 상황 모두, 이 RESTORE 유형은 보통 경로 재지정 복원에서 참조됩니다.

테이블 공간 컨테이너를 제어 센터 내에서 복원 타스크를 사용하여 다시 정의할 수 있습니다. 또한, RESTORE 명령의 REDIRECT 매개변수를 사용하여 경로 재지정을 지정할 수 있습니다. 제어 센터를 사용하는 경우, 경로 재지정된 복원을 수행하는 한 가지 방법은 데이터베이스 복원 노트북의 컨테이너 페이지를 사용하는

것입니다. 이 페이지는 새 컨테이너를 추가하고, 기존 컨테이너의 경로를 변경하거나, 컨테이너를 제거할 때 사용할 수 있는 기능을 제공합니다. 복원 데이터베이스 조작중에, 유효하지 않은 컨테이너 경로가 감지되면, 제어 센터는 컨테이너 경로를 변경하거나 컨테이너를 제거하려고 합니다.

주:

1. 디렉토리 및 파일 컨테이너는 존재하지 않는 경우 자동으로 작성됩니다. 몇 가지 다른 이유로 컨테이너에 액세스할 수 없는 경우, 경로 재지정은 필요하지 않습니다. 데이터베이스 관리 프로그램은 장치 컨테이너를 자동으로 작성하지 않습니다.
2. 모든 RESTORE에서 컨테이너 경로 재지정을 수행하는 기능은 테이블 공간 컨테이너를 관리하는 데 있어 상당한 융통성을 제공합니다. 예를 들어, SMS 테이블 공간에 대한 컨테이너 추가를 직접 지원하지 않더라도, 경로 재지정된 복원에서 간단히 추가 컨테이너를 지정함으로써 이를 달성할 수 있습니다. 마찬가지로, 파일 컨테이너에서 장치 컨테이너로 DMS 테이블 공간을 이동시킬 수 있습니다.
3. 경로 재지정된 복원은 다수의 API를 통해 지원되기도 합니다. 특정한 경우, 경로 재지정된 복원을 수행하기 위해 프로그램을 작성할 수 있지만, 이들 API는 주로 범용 유틸리티를 작성하려는 개발자를 위한 것입니다.

기존 데이터베이스로 복원

전체 데이터베이스 백업의 백업 사본을 기존의 데이터베이스로 복원할 수 있습니다. 기존 데이터베이스로 복원하려면, SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다. 백업 이미지는 별명, 데이터베이스 이름 또는 데이터베이스 시드(seed)에 있어서 기존의 데이터베이스와 다를 수 있습니다.

데이터베이스 시드는 데이터베이스의 주기 동안 항구적으로 남아 있는 데이터베이스에 대한 고유 식별자입니다. 이 시드는 데이터베이스가 처음 작성될 때 데이터베이스 관리 프로그램에 의해 할당됩니다. 시드는 백업이 다른 데이터베이스 시드를 가진 경우에도, 백업 복원에 따라 변경되지 않습니다. DB2는 복원의 시드를 항상 사용합니다.

기존 데이터베이스로 복원할 때, 복원 작업은 다음과 같은 기능을 수행합니다.

- 테이블, 색인, 기존의 데이터베이스에 대한 긴 필드를 삭제하고 백업의 내용으로 대체하십시오.
- 복원중인 테이블 공간 각각에 대해 테이블 공간 테이블 항목을 대체하십시오.
- 디스크상의 복구 실행기록 파일이 손상되지 않는 한, 이를 보유하십시오. 디스크의 파일이 손상된 경우, 데이터베이스 관리 프로그램은 백업으로부터 파일을 복사합니다.
- 기존 데이터베이스에 대해 인증을 보유하십시오.
- 데이터베이스가 있는 곳과 데이터베이스가 카탈로그화되는 방법을 정의하는 기존 데이터베이스에 대한 데이터베이스 디렉토리를 보유하십시오.
- 데이터베이스 시드가 다를 경우,
 - 기존 데이터베이스와 연관된 로그를 삭제하십시오.
 - 백업으로부터 데이터베이스 구성 파일을 복사하십시오.
 - *newlogpath*가 RESTORE 명령에 지정되면, 데이터베이스 구성 파일에서 로그 경로(*logpath* 매개변수에 의해 지정됨)에 설정하십시오.
- 데이터베이스 시드가 같을 경우,
 - 파일이 손상되지 않은 경우 백업으로부터 복사된 현재의 데이터베이스 구성 파일을 유지하십시오.
 - 이미지가 복구 불가능한 데이터베이스 이미지일 경우, 로그를 삭제하십시오.
 - *newlogpath*가 RESTORE 명령에 지정되면, *logpath* 데이터베이스 구성 매개변수의 값으로 설정하십시오. 그렇지 않으면, 현재 로그 경로를 데이터베이스 구성 파일로 복사하십시오. 로그 경로를 검증하십시오. 데이터베이스에서 로그 경로를 사용할 수 없는 경우, 기본 로그 경로를 사용하도록 데이터베이스 구성을 변경하십시오.

새로운 데이터베이스로 복원

이미 존재하는 데이터베이스로 데이터베이스를 복원하려면, 새로운 데이터베이스를 먼저 작성하고 데이터의 백업을 복원합니다. 새로운 데이터베이스로 복원하려면, SYSADM 또는 SYSCTRL 권한이 있어야 합니다.

주: 백업의 코드 페이지와 목표 데이터베이스의 코드 페이지는 일치해야 합니다. 그렇지 않은 경우, 올바른 코드 페이지를 지정하여 새로운 데이터베이스를 먼저 작성하고 복원하십시오.

새로운 데이터베이스를 복원할 때, RESTORE 명령은 다음과 같은 기능을 수행합니다.

- 목표 데이터베이스 별명 매개변수에 의해 지정된 데이터베이스 이름 및 데이터베이스 별명을 사용하여 새 데이터베이스를 작성하십시오(이 목표 데이터베이스의 별명이 지정되지 않으면, RESTORE 명령은 소스 데이터베이스 별명 매개변수와 동일한 이름과 별명을 가진 데이터베이스를 작성합니다).
- 백업으로부터 데이터베이스 구성 파일을 복원하십시오.
- *newlogpath*가 RESTORE 명령에 지정되면, 데이터베이스 구성 파일에서 *logpath* 매개변수의 값으로 설정하십시오. 로그 경로를 검증하십시오. 데이터베이스에서 로그 경로를 사용할 수 없는 경우, 기본 로그 경로를 사용하도록 데이터베이스 구성을 변경하십시오.
- 백업으로부터 인증 유형을 복원하십시오.
- 데이터베이스 디렉토리에 대한 백업으로부터 데이터베이스 주석을 복원하십시오.
- 데이터베이스에 대한 복구 실행기록 파일을 복원하십시오.

복구 방법: 롤 포워드 복구

RESTORE 및 ROLLFORWARD 명령과 함께 BACKUP 명령을 사용하는 롤 포워드 복구에서는 데이터베이스 또는 테이블 공간이 지정된 시점의 상태로 복구됩니다.

처음으로 데이터베이스를 작성할 때는, 순환 로그만이 활성화됩니다. 즉, 로그가 다시 사용되고(순환식으로), 저장되거나, 보존되지 않습니다. 순환 로그가 활성화되면 롤 포워드 복구는 불가능합니다. 시스템 손상 복구 또는 버전 복구만이 가능합니다. 로그 아카이브가 실행되는 경우에도 롤 포워드 복구가 가능한데, 그 이유는 로그가 백업이 이루어진 후에 데이터베이스에 대한 변경사항을 기록하기 때문입니다. *logretain* 데이터베이스 구성 매개변수를 "RECOVERY"로 설정하거나 *userexit* 데

데이터베이스 구성 매개변수가 사용 가능한 경우 또는 모두의 경우, 로그 아카이브를 수행합니다. 이 매개변수가 이전에서 설명한 것처럼 구성되면, 데이터베이스를 롤 포워드 복구에 사용할 수 있습니다.

데이터베이스가 복구 가능한 경우, 데이터베이스 레벨 및 테이블 공간 레벨 모두에서 백업, 복원 및 롤 포워드 복구를 수행할 수 있습니다. 데이터베이스 및 테이블 공간의 백업을 온라인으로 완료할 수 있습니다. 또한, 온라인 복원 및 롤 포워드도 테이블 공간 레벨에서 수행할 수 있습니다.

롤 포워드 복구는 로그의 기록된 작업 단위(UOW)를 복원된 데이터베이스, 테이블 공간에 다시 적용합니다. 로그 끝까지 또는 특정 시점까지의 롤 포워드 복구를 지정할 수 있습니다.

롤 포워드 복구는 365 페이지의 『데이터베이스 복원』에 설명된 것처럼 전체 데이터베이스 복원이 완료된 후에 수행될 수 있습니다. 또한, 이 방법은 롤 포워드 보류 상태에 있는 테이블 공간으로 수행될 수도 있습니다. 테이블 공간을 롤 포워드하는 데 필요한 고려사항은 386 페이지의 『테이블 공간의 롤 포워드 변경사항』에서 자세한 내용을 참조하십시오.

로깅과 연관된 데이터베이스 구성 매개변수에 대해서는 382 페이지의 『데이터베이스 로깅을 위한 구성 매개변수』에서 자세한 내용을 참조하십시오.

백업 고려사항

다음은 데이터베이스에 포워드 복구가 작동 가능한 경우 적용되는 백업 고려사항입니다. 백업을 수행하는 데 적용되는 일반 정보는 다음을 참조하십시오.

- 355 페이지의 『데이터베이스 백업』
- 357 페이지의 『BACKUP 명령 사용 계획』
- 360 페이지의 『BACKUP 명령 호출』
- 363 페이지의 『BACKUP으로 작성된 백업 이미지』.
- 롤 포워드 복구는 *logretain* 및 *userexit* 구성 매개변수를 기본 설정값(『No』)으로 설정하면 작동하지 않습니다. 초기에 데이터베이스를 복구하기 위해 사용할 수 있는 백업이 없어 두 매개변수의 기본값이 『No』로 설정되고 데이터베이스는 복원될 수 없으므로, 이에 대해 포워드 복구를 수행할 수 없습니다.

새로운 데이터베이스의 롤 포워드 복구를 작동시키려면, 데이터베이스의 첫번째 백업을 취하기 전에 이 구성 매개변수 중 적어도 하나를 활성화해야 합니다. 두 매개변수 값 중 하나 또는 둘다 변경할 경우, 데이터베이스는 백업 보류 상태가 되어 데이터베이스의 오프라인 백업을 수행해야 합니다. 백업 조작이 성공적으로 완료된 다음에는 데이터베이스를 사용할 수 있습니다.

- 백업 보류 상태에 있는 데이터베이스를 제외한, 사용 가능 상태에 있지 않는 데이터베이스는 백업할 수 없습니다.
 - 데이터베이스 또는 테이블 공간이 데이터베이스 복원 단계에서의 시스템 고장으로 인해 부분적으로만 복원된 상태에 있을 경우, 데이터베이스 또는 테이블 공간을 성공적으로 복원해야 백업할 수 있습니다.
 - 데이터베이스 내의 테이블 공간이 『비정상』 상태이면, 백업 보류 상태가 아닌 데이터베이스 또는 해당 테이블 공간을 백업할 수 없습니다.
- 하드 디스크, 테이프 또는 TSM 유틸리티 또는 다른 벤더 저장영역 관리 프로그램 제품에 의해 관리되는 위치로 데이터베이스 또는 테이블 공간을 백업할 수 있습니다. TSM에 대해서는 425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오.

OS/2에서, 디스켓 또는 User Exit 프로그램으로 백업할 수도 있습니다.

- 데이터베이스가 롤 포워드 복구로 작동 가능하고 백업을 특별히 참조하는 기능을 지원하지 않는 테이프 시스템을 사용하고 있는 경우, 같은 테이프에 같은 데이터베이스의 복수 백업 사본을 유지하지 않는 것이 권장됩니다.
- 데이터베이스 또는 테이블 공간의 백업 데이터가 들어가도록 여러 파일이 작성될 수 있습니다.

OS/2에서, User Exit 프로그램으로부터 복원하고 데이터베이스를 롤 포워드할 경우, 데이터베이스로의 경로는 컨테이너를 위치지정하는 데 사용된 단 하나의 참조사항입니다. 따라서, 백업 테이프에 있는 해당 데이터베이스에 대한 모든 컨테이너가 복원됩니다.

- 백업하는 데 필요한 시간을 줄이려면, 다음을 수행하십시오.
 - 테이블 공간 백업을 사용하십시오.

BACKUP 명령의 TABLESPACE 옵션을 사용하여 데이터베이스의 부분 백업(및 후속 복구)만 수행하도록 선택할 수 있습니다. 이로 인해, 별도의 테이블 공간에 들어 있는 데이터, 색인 및 긴 필드/대형 오브젝트(LOB)의 관리가 더 쉬워집니다.

- PARALLELISM 매개변수의 값이 백업중인 테이블 공간의 수를 반영하도록 증가시키십시오.
- 테이블 공간 백업에 대해 고려할 점은 다음과 같습니다.
 - 테이블 공간 백업 및 테이블 공간 복구는 비록 다른 테이블 공간에서 수행하더라도 동시에 실행할 수는 없습니다.
 - 테이블이 둘 이상의 테이블 공간에 걸쳐 있으면, 테이블 공간 세트를 함께 백업(및 복원)해야 합니다.
 - 각 테이블 공간이 다른 디스크에 있는 경우, 미디어 오류로 인해 특정한 테이블 공간만 영향을 받고 전체 데이터베이스는 영향을 받지 않습니다. 오류를 가진 테이블 공간은 롤 포워드 보류 상태에 있게 됩니다. 이 상태의 테이블 공간에 시스템 카탈로그 테이블이 들어 있지 않으면 데이터베이스 내의 다른 테이블 공간을 사용할 수 있습니다. 이 경우, 데이터베이스에 연결할 수는 없습니다.
 - 시스템 카탈로그 테이블 공간을 포함하는 테이블 공간 레벨 백업이 사용 가능할 경우, 시스템 카탈로그 테이블 공간을 나머지 데이터베이스와는 별도로 복원할 수 있습니다.
 - 백업된 테이블 공간 목록에 임시 테이블 공간이 들어 있으면, 백업은 실패합니다.
- 파티션된 데이터베이스 환경에 대해 고려해야 할 점은 다음과 같습니다.

포워드 복구 기능이 필요할 경우, 노드 목록에서 데이터베이스를 정기적으로 백업하고, 시스템의 나머지 노드에 대해 적어도 하나의 백업을 보유해야 합니다 (노드가 해당 데이터베이스에 대한 사용자 데이터를 담고 있지 않은 경우에도). 데이터베이스에 대한 사용자 데이터가 없는 데이터베이스 파티션 서버에서 데이터베이스 파티션의 백업 이미지가 필요한 두 가지 상황이 있습니다.

 - 최종 백업을 수행한 후 데이터베이스 파티션 서버를 데이터베이스 시스템에 추가하고, 이 데이터베이스 파티션 서버에서 포워드 복구를 수행해야 합니다.
 - 적절한 복구가 사용되어 시스템의 모든 데이터베이스 파티션이 롤 포워드 보류 상태에 있어야 합니다.

복구 실행기록 파일은 전체 데이터베이스 또는 테이블 공간의 백업이나 복원을 수행할 때마다 요약 정보로 자동으로 갱신됩니다. 이 파일은 데이터베이스 내의 활동을 복원할 유용한 추적 매커니즘이 될 수 있습니다. 이 파일은 데이터베이스 구

성 파일과 같은 디렉토리에 작성됩니다. 복구 실행기록 파일에 대해서는 406 페이지의 『복구 실행기록 파일 정보』에서 자세한 내용을 참조하십시오.

UNIX 기반 환경에서, 디스크에서 작성된 파일 이름은 마침표로 분리된 다음 정보를 결합시켜 구성됩니다. 기타 플랫폼에서는 4 레벨 서브디렉토리 트리가 사용됩니다.

데이터베이스 별명	백업 명령이 시작될 때 제공된 1-8자의 데이터베이스 별명
유형	수행된 백업 유형. 여기서, 『0』은 전체 데이터베이스를 나타내고, 『3』은 테이블 공간을 나타내며, 『4』는 테이블 로드 사본을 나타내는 것입니다.
인스턴스 이름	DB2INSTANCE 환경 변수에서 취한 데이터베이스 관리 프로그램의 현재 인스턴스의 1-8자의 문자 이름
노드 번호	노드 번호
카탈로그 노드 번호	데이터베이스 카탈로그 노드의 노드 번호
시간소인	백업이 수행된 날짜와 시간의 14자 표현. 시간소인은 <i>yyyymmddhhnnss</i> 형식입니다. 여기서, <i>yyyy</i> 는 연도(1995 - 9999) <i>mm</i> 은 월(01 - 12) <i>dd</i> 는 월의 날짜(01 - 31) <i>hh</i> 는 시간(00 - 23) <i>nn</i> 은 분(00 - 59) <i>ss</i> 는 초(00 - 59)
순차 번호	파일 확장자로 사용되는 3자릿수의 순차 번호

복원 고려사항

다음은 데이터베이스에 포워드 복구가 작동 가능한 경우 적용되는 복원 고려사항입니다. 복원을 수행하는 데 적용되는 일반 정보는 다음을 참조하십시오.

- 365 페이지의 『데이터베이스 복원』

- 366 페이지의 『RESTORE 명령 사용 계획』
- 367 페이지의 『RESTORE 명령 호출』
- 369 페이지의 『RESTORE중 테이블 공간 컨테이너 재정의』
- 370 페이지의 『기존 데이터베이스로 복원』
- 371 페이지의 『새로운 데이터베이스로 복원』.

고려해야 할 항목은 다음과 같습니다.

- 전체 데이터베이스 백업 또는 테이블 공간 백업의 백업 사본을 기존의 데이터베이스로 복원할 수 있습니다. 기존 데이터베이스로 복원하려면, SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다. 백업 이미지는 별명, 데이터베이스 이름 또는 데이터베이스 시드(seed)에 있어서 기존의 데이터베이스와 다를 수 있습니다.
- 기존 데이터베이스를 복원하며 데이터베이스 시드가 동일한 경우, 로그는 유지됩니다.
- 데이터베이스 또는 테이블 공간이 이전에 BACKUP 명령을 통해 백업된 경우에만 RESTORE 명령을 사용할 수 있습니다.
- 롤 포워드 복구에 사용 가능한 데이터베이스가 복원되면, 롤 포워드 보류 상태가 됩니다. 데이터베이스는 롤 포워드되지 않으면 사용할 수 없습니다. 복원 WITHOUT ROLLING FORWARD가 지정되면 복원이 발생합니다. 온라인 데이터베이스 백업이 복원되거나 선택된 테이블 공간 백업만이 복원되면, 롤 포워드를 중지할 수 없습니다.
- RESTORE 명령에 의해 사용된 데이터베이스 또는 테이블 공간의 백업 사본은 하드 디스크, 테이프 또는 다른 벤더 저장영역 관리 프로그램 제품의 Tivoli* 저장영역 관리 프로그램(TSM) 유틸리티에 의해 관리되는 위치로 위치지정될 수 있습니다. TSM에 대해서는 425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오.
TSM을 사용하며 TAKEN AT 매개변수를 지정하지 않은 경우, TSM은 최신 백업 사본을 검색합니다.
OS/2에서, 데이터베이스 또는 테이블 공간의 백업 사본은 디스켓에 또는 User Exit 프로그램을 통해 위치지정될 수도 있습니다.
Windows 95 및 Windows NT에서, 데이터베이스 또는 테이블 공간의 백업 사본은 디스켓에 위치지정될 수도 있습니다.

- 복원 및 롤 포워드가 독립 조작인 동안, 복구 전략에는 데이터베이스의 완전한 롤 포워드 복구의 첫단계로서 복원이 포함될 수 있습니다. 성공적인 복원 후, 백업이 수행될 때 롤 포워드 복구를 위해 구성된 데이터베이스는 롤 포워드 보류 상태로 되며, ROLLFORWARD 명령이 성공적으로 수행될 때까지 사용할 수 없게 됩니다.

ROLLFORWARD 명령이 발행되면, 다음이 수행됩니다.

- 데이터베이스가 롤 포워드 보류 상태에 있을 경우, 데이터베이스는 롤 포워드됩니다.
- 데이터베이스가 롤 포워드 보류 상태에 있지 않지만 데이터베이스의 테이블 공간은 그렇지 않을 경우, ROLLFORWARD 명령을 발행하고 테이블 공간 목록을 지정하면 해당 테이블 공간만이 롤 포워드됩니다. 목록을 지정하지 않은 경우, 롤 포워드 보류 상태에 있는 모든 테이블 공간은 롤 포워드됩니다.

주: 가장 최근의 백업 다음의 테이블 공간을 이름 바꾸기한 경우, 테이블 공간을 롤 포워드할 때 새 이름을 사용하는지 확인하십시오. 이전 테이블 공간 이름은 인지되지 않습니다.

- 파티션된 데이터베이스 환경에서, 어떤 데이터베이스 파티션은 롤 포워드 보류 상태이고, 다른 데이터베이스의 어떤 테이블 공간은 롤 포워드 보류 상태에 있는 경우(데이터베이스 파티션은 그렇지 않음), 먼저 데이터베이스 파티션을 롤 포워드하고 테이블 공간을 롤 포워드해야 합니다.

롤 포워드 프로세스가 수행중이면 또다른 데이터베이스의 RESTORE가 허용되지 않습니다.

주:

1. BACKUP 명령의 오프라인 옵션을 사용하여 작성된 전체 데이터베이스 백업으로부터 복원하는 경우, 복원 프로세스중에 이 롤 포워드 보류 상태를 바이패스할 수 있습니다. WITHOUT ROLLING FORWARD 옵션을 사용하면, 데이터베이스를 롤 포워드하지 않고도 즉시 복원된 데이터베이스를 사용할 수 있습니다.
 2. BACKUP 명령의 온라인 옵션을 사용하여 작성된 백업으로부터 복원하는 경우, 이 롤 포워드 보류 상태를 바이패스할 수 없습니다.
- 테이블 공간 복원에 대해 고려해야 할 사항은 다음과 같습니다.

- 테이블 공간이 현재 존재하고 동일한 테이블 공간일 경우, 테이블 공간만 복원할 수 있습니다(『동일한 테이블 공간』은 테이블 공간이 백업 이미지 확보와 테이블 공간을 복원하려는 시도간에 삭제 및 재작성되지 않는다는 것을 의미합니다). 가장 최근의 백업 다음의 테이블 공간을 이름 바꾸기한 경우, 테이블 공간을 롤 포워드할 때 새 이름을 사용하는지 확인하십시오. 이전 테이블 공간 이름은 인지되지 않습니다.
 - 테이블 공간 백업을 새로운 데이터베이스에 복원할 수 없습니다.
 - 둘 이상의 테이블 공간에 걸쳐 있는 테이블을 백업한 경우, 모든 테이블 공간을 복원해야 합니다.
 - 일단 RESTORE 명령이 테이블 공간 백업에 대해 시작되면, 롤 포워드 복구에 의해 RESTORE 명령이 성공적으로 완료될 때까지 테이블 공간을 사용할 수 없습니다.
 - 테이블 공간 복원은 온라인(공유 모드) 또는 오프라인(독점 모드)이 될 수 있습니다.
 - 테이블 공간 백업을 복원하는 중에 시스템 오류가 발생하면, 복원중인 테이블 공간만 사용할 수 없습니다. 데이터베이스의 다른 테이블 공간은 사용할 수 있습니다.
 - 시스템 카탈로그 테이블의 온라인 테이블 공간 복원을 수행할 수 있습니다.
 - 부분 RESTORE 또는 부속 집합 RESTORE를 수행할 경우, 테이블 공간 백업 또는 전체 데이터베이스 백업을 사용하고, 해당 이미지로부터 하나 이상의 테이블 공간을 선택할 수 있습니다. 테이블 공간과 연관된 모든 로그 파일은 백업이 작성된 시점부터 존재합니다. 가장 최근의 백업 다음의 테이블 공간을 이름 바꾸기한 경우, 테이블 공간을 복원 또는 롤 포워드할 때 새 이름을 사용하는지 확인하십시오. 이전 테이블 공간 이름은 인지되지 않습니다.
- 파티션된 데이터베이스 시스템에서, 테이블 공간을 로그 끝으로 롤 포워드하려면, 해당 테이블 공간을 각 데이터베이스 파티션(노드)에서 복원할 필요는 없습니다. 복구가 필요한 데이터베이스 파티션에서 테이블 공간을 복원하면 됩니다. 특정 시점으로 테이블 공간을 롤 포워드하려면, 롤 포워드 전에 각 데이터베이스 파티션에서 테이블 공간을 복원해야 합니다.
- OS/2에서, User Exit 프로그램으로부터 복구할 때에는 부분 복원 또는 부속 집합 복원이 가능하지 않습니다.

- 경로 재지정된 복원과 관련하여 고려해야 할 사항은 다음과 같습니다.
 - 데이터베이스 또는 하나 이상의 테이블 공간의 백업중에, 레코드에는 백업된 테이블 공간에 의해 사용중인 모든 테이블 공간 컨테이너를 유지합니다. RESTORE중, 백업에 나열되어 있는 모든 컨테이너는 현재 존재하는지 액세스 가능한지 알기 위해 점검됩니다. 미디어 오류(또는 다른 이유로) 때문에 하나 이상의 컨테이너에 액세스할 수 없을 경우, RESTORE는 실패합니다. 이러한 경우에 복원을 할 수 있도록 하기 위해 RESTORE 동안에 테이블 공간 컨테이너의 경로 재지정이 지원됩니다. 이 지원에는 테이블 공간 컨테이너의 추가, 변경 또는 제거가 포함됩니다.
 - RESTORE는 종종 ROLLFORWARD 다음에 수행되어 백업이 수행된 시점 이후에 데이터베이스 로그에 기록된 변경사항을 다시 적용합니다. 롤 포워드 조작중, ADD 옵션으로 ALTER TABLESPACE를 수행하는 트랜잭션을 다시 실행하거나 수행할 수 있습니다. ROLLFORWARD를 성공적으로 수행하려면, 추가될 컨테이너에 액세스할 수 있어야 합니다. 컨테이너에 액세스할 수 없으면, 테이블 공간에 대한 롤 포워드가 일시중단되며 테이블 공간이 롤 포워드 보류 상태로 남아 있게 됩니다.
 - 데이터베이스 로그에서 컨테이너 추가 조작을 다시 수행하거나 수행하지 않으려 할 수도 있습니다. 사실상, 백업이 수행된 이후 어떤 컨테이너가 추가되었는지 알지 못할 수도 있습니다. 따라서, 필요한 컨테이너를 예상할 수 없습니다. 그렇지 않으면, 경로 재지정된 복원을 수행하는 이유에 따라, 단순히 복원시 지정된 컨테이너의 목록을 더 선호할 수도 있으며, 다른 컨테이너 추가를 원하지 않을 수 있습니다. 이러한 활동을 제어하기 위해서, ROLLFORWARD가 롤 포워드 복구중 컨테이너를 다시 작성하려는지를 복원시에 나타낼 수 있습니다(제어 센터의 데이터베이스 복원 또는 테이블 공간 복원 노트북의 CONTAINERS - CHANGE 창에서 테이블 공간 컨테이너 목록을 편집할 수 있습니다).

데이터베이스의 롤 포워드 변경사항

롤 포워드 복구는 복원된 데이터베이스에 빌드되며, 데이터베이스 백업이 수행된 이후의 특정 시점까지 데이터베이스를 복원할 수 있도록 합니다. 이 시점은 로그 끝 또는 데이터베이스 백업과 로그 끝 시간 사이의 시점일 수 있습니다.

주: 복원 및 롤 포워드가 모든 로그의 끝으로 수행될 때, LIST HISTORY 명령 다음에 표시된 백업 ID는 끝 시간을 표시합니다. 즉, 백업 ID 값은 99991231235959입니다. 백업 ID는 롤 포워드가 수행될 때 이 방식으로 변환됩니다.

사용중인 로그 또는 보존된 로그를 사용할 수 없을 경우, 특정 시점 복구 방법을 사용할 수 있습니다. 이 경우, 로그가 소실된 시점으로 롤 포워드할 수 있습니다. 또한 데이터베이스에 대해 불량 트랜잭션이 수행된 경우, 특정 시점으로 롤 포워드할 수 있습니다. 이 경우, 데이터베이스를 복원하고, 불량 트랜잭션이 실행되기 바로 전 시점으로 롤 포워드할 수 있습니다.

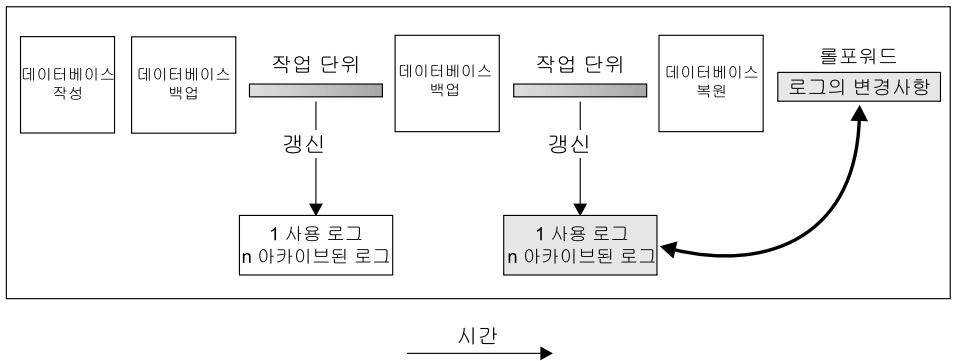


그림 10. 롤 포워드 복구

또한, 테이블 공간에서 특정 시점 롤 포워드 복구를 수행할 수도 있습니다. 추가 정보는 386 페이지의 『테이블 공간의 롤 포워드 변경사항』에서 자세한 내용을 참조하십시오.

이 방법을 사용하려면, 데이터베이스가 롤 포워드 복구를 작동할 수 있도록 구성되어야 합니다. 데이터베이스 구성 파일 및 데이터베이스 로그에 대한 고려사항은 다음 주제에 나와 있습니다.

- 데이터베이스 로그를 위한 구성 매개변수
- 테이블 공간의 롤 포워드 변경사항
- ROLLFORWARD 명령 사용 계획
- ROLLFORWARD 명령 호출
- 로드 사본 위치 파일 사용
- 로그 파일 관리를 위한 고려사항

- 로그 유실.

DATALINK 컬럼이 들어 있는 테이블이 있다면, 418 페이지의 『데이터베이스 및 테이블 공간 복원 및 로그 끝까지 롤 포워드』 및 418 페이지의 『데이터베이스 및 테이블 공간 복원 및 특정 시점까지 롤 포워드』에서 자세한 내용을 참조하십시오.

데이터베이스 로깅을 위한 구성 매개변수

데이터베이스 구성 파일은 롤 포워드 복구와 관련된 매개변수를 가지고 있습니다. 기본 매개변수는 이 복구를 지원하지 않으므로 이 복구를 사용할 계획이면, 이들 기본값 중 일부를 변경해야 합니다. DB2 UDB 구성에 대해서는 **관리 안내서: 성능**에서 자세한 내용을 참조하십시오.

1차 로그(logprimary)

이 매개변수는 작성될 1차 로그의 수를 지정합니다.

비어 있는지 가득 차 있는지에 관계없이 1차 로그는 같은 크기의 디스크 공간이 필요합니다. 그러므로, 필요한 수 이상의 로그를 구성하는 경우에는 디스크 공간을 불필요하게 사용하게 됩니다. 너무 적은 수의 로그를 구성하면 로그가 가득 찬 상태가 발생할 수 있습니다. 구성할 로그 수를 선택할 때 각 로그의 크기 및 응용프로그램이 로그 가득 참 상태를 처리할 수 있는지 여부를 고려해야 합니다.

기존 데이터베이스에 롤 포워드 복구를 작동 가능하게 하고자 할 경우, 1차 로그의 수를 1차 로그와 2차 로그의 수의 합계+1로 변경하십시오. 추가 정보는 롤 포워드 복구가 작동 가능한 데이터베이스의 긴 변수문자 및 **LOB** 필드에 기록됩니다.

총 로그 파일 크기 한계는 32GB입니다. 즉, 로그 파일 수(LOGPRIMARY + LOGSECOND)에 바이트 단위의 로그 파일크기(LOGFILSIZ * 4096)를 곱한 값은 32GB보다 작아야 합니다.

이 구성 매개변수에 대해서는 **관리 안내서: 성능**에서 자세한 내용을 참조하십시오.

2차 로그(logsecond)

이 매개변수는 복구 로그 파일에 대해 작성되고 사용되는 2차 로그 파일의 수를 지정합니다(필요한 경우에만).

1차 로그 파일이 가득 차게 되면, 2차 로그 파일(*logfilesiz* 크기)이 필요할 때 한 번에 하나씩, 이 매개변수로 조절할 수 있는 최대수까지 할당됩니다. 이 매개변수에 의해 허용된 것보다 더 많은 2차 로그가 필요한 경우, 오류 코드가 응용프로그램으로 리턴되고 데이터베이스에 대한 활동이 중단됩니다.

이 구성 매개변수에 대해서는 *관리 안내서*: 성능에서 자세한 내용을 참조하십시오.

로그 크기(*logfilesiz*)

이 매개변수는 구성된 각 로그의 페이지 수를 결정합니다. 페이지는 4KB 크기입니다.

주: 총 로그 파일 크기 한계는 32GB입니다(즉, $(logfilesiz + logprimary) \times logfilesiz < 4GB/4096$).

1차 로그의 크기는 성능과 직접적인 관계가 있습니다. 데이터베이스가 로그를 보유하도록 구성될 때 로그가 찰 때마다 새 로그의 할당과 초기화를 위한 요청이 발행됩니다. 로그의 크기를 늘리면, 새 로그 할당 및 초기화에 필요한 요청의 수가 줄어듭니다(그러나 로그 크기가 커질수록 각각의 새로운 로그를 형식화하는 데 더 많은 시간이 걸린다는 점에 유의하십시오). 새로운 로그의 형식화는 형식화가 데이터베이스 성능에 영향을 주지 않도록, 데이터베이스에 연결된 응용프로그램에 대해 투명합니다.

데이터베이스를 여는 처리 시간을 최소화하기 위해 데이터베이스를 열린 상태로 유지하는 응용프로그램을 가지고 있다고 가정할 경우, 로그 크기의 값은 오프라인 아카이브 로그 사본을 만드는 데 걸리는 시간 크기에 의해 결정되어야 합니다.

오프라인 아카이브 로그를 저장하기 위해 사용하는 장치의 데이터 전송 속도와 사본을 만드는 데 사용되는 소프트웨어는 최소한 데이터베이스 관리 프로그램이 로그에 데이터를 기록하는 평균 속도와 일치해야 합니다. 전송 속도가 새 로그 데이터의 생성 속도를 따라잡지 못하고, 오랜 시간 동안 로깅 활동이 계속되는 경우, 빈 디스크 공간 크기에 의해 결정된 디스크 공간이 모두 소모될 수 있습니다. 이 상태가 발생하면, 데이터베이스 처리가 중단됩니다.

데이터 전송 속도는 테이프 또는 광학 미디어를 사용할 때 가장 중요합니다(로그를 저장하기 위해 다른 미디어를 사용하는 것에 대해서는 485 페이지의 『부록C. 데이터베이스 복구를 위한 User Exit』에서 자세한 내용을 참조하십시오). 일부 테이프 장치는 파일의 크기에 상관없이 파일을 복사하는 데 똑같은 시간이 걸립니다. 장치를 아카이브하는 기능을 결정해야 합니다.

이 외에도, 테이프 장치는 몇 가지 고유한 고려사항을 가지고 있습니다. 아카이브 요청의 빈도가 중요합니다. 어떤 복사 조작 시간이 5분이면 로그 크기는 작업량이 최대치일 때 5분 동안의 로그 데이터를 수용할 만큼 충분히 커야 합니다. 또한, 테이프 장치는 하루당 조작의 수를 제한하는 설계상의 한계를 가질 수 있습니다. 이상의 요소가 로그 크기를 결정할 때 고려해야 할 사항입니다.

로그 파일 유실을 최소화하는 것도 로그 크기 설정에 있어서 중요한 고려 사항입니다. 아카이브는 전체 로그를 취합니다. 단일의 대형 로그를 사용할 경우, 아카이브 시간 간격을 증가시킵니다. 로그를 가진 미디어가 고장 나면 일부 트랜잭션 정보가 유실될 수도 있습니다. 로그 크기를 줄이면 아카이브의 빈도는 늘어나지만, 미디어 고장인 경우, 유실된 로그 앞의 더 작은 로그를 사용할 수 있으므로 정보 유실량을 줄일 수 있습니다.

로그 버퍼(logbufsz)

이 매개변수로 로그 레코드를 디스크에 쓰기 전에 로그 레코드용 버퍼로서 사용할 데이터베이스 공유 메모리의 크기를 지정할 수 있습니다. 로그 레코드는 다음 중 하나가 발생하면 디스크로 기록됩니다.

- 트랜잭션 요약
- 로그 버퍼가 가득 참
- 기타 내부 데이터베이스 관리 프로그램 이벤트의 결과로 인해

로그 레코드의 버퍼링은 파일 입출력을 기록하는 데 훨씬 더 효과적일 수 있습니다. 이는 로그 레코드가 디스크에 기록되는 빈도 수가 줄어들고 한 번에 더 많은 로그 레코드가 기록되기 때문입니다.

그룹 요약 수(mincommit)

이 매개변수로 최소수 만큼의 요약이 수행될 때까지 로그 레코드가 디스

크에 기록되는 것을 지연시킬 수 있습니다. 이러한 지연으로 로그 레코드 기록과 관련된 데이터베이스 관리 프로그램의 오버헤드를 줄일 수 있고, 결과적으로 데이터베이스에 대해 수행중인 응용프로그램이 여러 개 있어 매우 짧은 시간 프레임 내에 응용프로그램이 요구하는 많은 확약을 처리해야 할 때 성능이 향상됩니다.

이러한 확약 그룹화는 이 매개변수의 값이 1보다 크고, 데이터베이스에 연결된 응용프로그램의 수가 이 매개변수 값보다 클 때에만 발생합니다. 확약 그룹화가 수행되면, 응용프로그램 확약 요구가 1초 동안 또는 확약 요구의 수가 이 매개변수 값과 같아질 때까지 보류됩니다.

새로운 로그 경로(newlogpath)

데이터베이스 로그는 처음에는 데이터베이스 디렉토리의 서브디렉토리인 `SQLLOGDIR`에 작성됩니다. 이 구성 매개변수를 변경함으로써 사용 중인 로그와 장래 보존 로그가 보관되는 위치를 다른 디렉토리 또는 장치로 변경할 수 있습니다. 데이터베이스가 롤 포워드 복구용으로 구성되어 있는 경우, 데이터베이스 로그 경로 디렉토리에 저장되는 보존 로그는 새 로그 경로 디렉토리로 이동되지 않습니다.

사용자가 로그 경로 디렉토리를 변경할 수 있기 때문에, 롤 포워드 복구에 필요한 로그는 다른 디렉토리 또는 다른 장치에 있을 수 있습니다. 롤 포워드 프로세스시 이 구성 매개변수를 변경하여 여러 위치의 로그에 액세스할 수 있습니다.

`newlogpath` 값을 변경해도 데이터베이스가 일치 상태에 있지 않은 한 적용되지 않습니다. 정보 데이터베이스 구성 매개변수 `database_consistent`는 데이터베이스의 상태를 나타냅니다. `database_consistent` 상태 표시기에 대해서는 **관리 안내서: 성능**에서 자세한 내용을 참조하십시오. 데이터베이스가 불일치 상태에 있는 경우, 데이터베이스 로그가 수행하는 역할에 대해서는 400 페이지의 『로그 파일 관리를 위한 고려사항』에서 자세한 내용을 참조하십시오.

로그 보유(logretain)

이 매개변수는 보존 로그가 데이터베이스 로그 경로 디렉토리에 보관되도록 합니다. 이를 『RECOVERY』로 설정하면 데이터베이스 관리 프로그램에서 롤 포워드 복구 방법을 사용할 수 있습니다. `logretain` 구성 매개변

수가 작동 가능한 경우에는 *userexit*가 작동 가능해질 필요가 없습니다. 롤 포워드 복구 방법을 가능하게 하는 데에는 두 매개변수 중 하나만으로도 충분합니다.

이 매개변수의 사용은 기본값인 순환 기록이 대체될 것임을 의미합니다.

UserExit

이 매개변수로 인해 데이터베이스 관리 프로그램은 로그 아카이브 및 검색을 위해 User Exit 프로그램을 호출합니다. User Exit 프로그램이 작동 가능화되면 롤 포워드 복구가 허용됩니다. *userexit* 구성 매개변수가 작동 가능화되어 있을 때에는 *logretain*을 작동 가능화하지 않아도 됩니다. 롤 포워드 복구 방법을 가능하게 하는 데에는 두 매개변수 중 하나만으로도 충분합니다.

이 매개변수의 사용은 기본값인 순환 기록이 대체될 것임을 의미합니다. *userexit*는 *logretain*을 포함하지만, *logretain*은 *userexit*를 포함하지 않습니다.

User Exit 프로그램에 대해서는 485 페이지의 『부록C. 데이터베이스 복구를 위한 User Exit』에서 자세한 내용을 참조하십시오.

사용중인 로그는 롤 포워드 복구를 가능하게 하기 위해 *userexit* 구성 매개변수 또는 *logretain* 구성 매개변수 중 하나를 사용할 때 중요합니다. *userexit* 구성 매개변수를 사용할 수 있는 경우, User Exit 프로그램은 사용중인 로그 경로에서 보존 로그 파일로 호출됩니다. *logretain* 구성 매개변수가 『RECOVERY』로 설정되면, 이것으로 로그 파일이 사용중인 로그 경로에 보존됩니다. 사용중인 로그 경로는 로그 파일 경로 또는 로그 파일의 변경된 경로(*newlogpath*)에 의해 결정됩니다.

테이블 공간의 롤 포워드 변경사항

데이터베이스에서 포워드 복구가 작동 가능한 경우, 전체 데이터베이스를 사용하지 않고 테이블 공간을 백업하고 복원하며 롤 포워드하는 옵션이 있습니다. 이것으로 시간이 절약되므로 사용자는 각 테이블 공간에 대해 복구 방침을 구현하려고 합니다. 전체 데이터베이스를 복구하는 것보다 데이터베이스 일부를 복구하는 데 시간이 덜 소요됩니다. 예를 들어, 디스크가 손상되고 하나의 테이블 공간만 들어 있는 경우, 테이블 공간은 전체 데이터베이스를 복구하지 않고(그리고 나머지 데이터베이스로의 사용자 액세스에 영향을 주지 않으면서) 복원되고 롤 포워드될 수 있

습니다. 또한 테이블 공간 레벨 백업을 사용하면 데이터베이스의 중요 부분을 다른 부분보다 자주 백업할 수 있으므로, 데이터베이스 전체를 백업하는 것보다 시간이 적게 소요됩니다.

가장 최근의 백업 다음의 테이블 공간을 이름 바꾸기한 경우, 테이블 공간을 롤 포워드할 때 새 이름을 사용하는지 확인하십시오. 이전 테이블 공간 이름은 인지되지 않습니다. 또한 최소한 테이블 공간의 이름이 바뀐 특정 시점으로 롤 포워드해야 합니다.

파티션된 데이터베이스 환경에서, 어떤 데이터베이스 파티션은 롤 포워드 보류 상태이고, 다른 데이터베이스의 어떤 테이블 공간은 롤 포워드 보류 상태에 있는 경우(데이터베이스 파티션은 그렇지 않음), 먼저 데이터베이스 파티션을 롤 포워드하고 테이블 공간을 롤 포워드해야 합니다.

테이블의 데이터 및 긴 오브젝트는 다른 테이블 공간에 있고 테이블이 재인식되면, 데이터 및 긴 오브젝트 모두에 대한 테이블 공간은 복원과 롤 포워드가 함께 이루어져야 합니다. 테이블이 재인식된 후 손상된 테이블 공간을 백업해야 합니다.

다른 상태가 테이블 공간과 연관되어 현재 상태를 나타냅니다.

- 복원된 후에 테이블 공간은 롤 포워드 보류 상태이거나 입출력 오류가 나타나는 상태입니다. 입출력 오류가 정정되면, 테이블 공간은 롤 포워드 보류 상태를 제거하기 위해 롤 포워드되어야 합니다. 테이블 공간이 복원되면, 복원된 테이블 공간은 롤 포워드되어야 합니다.
- 테이블 공간은 해당 테이블 공간에서 롤 포워드 조작이 진행중일 때 진행중인 롤 포워드 상태입니다. ROLLFORWARD가 성공적으로 완료하면, 테이블 공간은 진행중인 롤 포워드 상태에서 제거됩니다.
- 백업 이미지를 온라인으로 얻고 롤 포워드 조작이 완료되지 않은 경우 테이블 공간은 롤 포워드 진행 상태가 될 수 있습니다.
- 복구 불가능한 오류가 해당 테이블 공간에서 발생하는 ROLLFORWARD CANCEL 또는 ROLLFORWARD 후에 테이블 공간은 복원 보류 상태입니다. 테이블 공간은 복원된 후 다시 롤 포워드되어야 합니다.

- 특정 시점으로 ROLLFORWARD 후에 또는 LOAD NO COPY 조작 후에 테이블 공간은 백업 보류 상태입니다. 테이블 공간을 사용하기 전에 백업해야 합니다.

테이블 공간이 복원된 후, 항상 롤 포워드 보류 상태에 있습니다(즉, 테이블 공간을 지정하고 WITHOUT ROLLING FORWARD 매개변수를 지정하면, WITHOUT ROLLING FORWARD는 무시됩니다). 테이블 공간을 사용할 수 있으려면, 테이블 공간에서 롤 포워드 복구를 실행해야 합니다. 로그 끝까지 롤 포워드하거나 특정 시점까지의 롤 포워드를 하는 옵션이 있습니다. 테이블 공간을 특정 시점까지 롤 포워드하려는 경우, 다음을 알아야 합니다.

- 시스템 카탈로그 테이블을 특정 시점으로 롤 포워드할 수 없습니다. 로그 끝까지 롤 포워드되어 데이터베이스의 모든 테이블 공간이 일관성 있는 상태로 남아 있는 것을 확인해야 합니다.
- 특정 시점으로 롤 포워드된 테이블 공간은 롤 포워드에 대해 지정된 특정 시점보다 먼저 백업으로부터 복원되어야 합니다.
- 테이블 공간을 롤 포워드하지 않으려는 경우, 복원된 백업 시점까지 테이블 공간을 롤 포워드하는 것과 동일한 ROLLFORWARD STOP을 지정하면 됩니다.

주: 백업 이미지가 온라인으로 취해진 것이라면 이 작업을 수행할 수 없습니다. 이 경우, 적어도 백업 종료 시점까지 롤 포워드해야 합니다.

- 특정 시점으로 롤 포워드하며 다중 테이블 공간에 테이블이 들어 있으면, 테이블이 들어 있는 모든 테이블 공간은 동시에 롤 포워드되어야 합니다. 예를 들어, 테이블 데이터를 이 테이블 공간에, 테이블에 대한 색인을 다른 테이블 공간에서 유지하려면, 특정 시점으로 두 테이블 공간을 동시에 롤 포워드해야 합니다.
- 테이블 공간을 롤 포워드하기 전에, LIST TABLESPACES SHOW DETAIL 명령을 사용하십시오. 이 명령을 사용하면 『최소 복구 시간』의 정보를 리턴하고, 가장 빠른 특정 시점으로 테이블 공간이 롤 포워드될 수 있습니다. DDL문이 이 테이블 공간에 대해 또는 테이블 공간의 테이블에 대해 실행되는 경우 최소 복구 시간이 갱신됩니다. 테이블 공간은 적어도 최소 복구 시간까지 롤 포워드되어 시스템 카탈로그 테이블의 정보와 동기화되어야 합니다.

- 테이블 공간의 이름을 바꾼 경우, 최소한 테이블 공간의 이름이 바뀐 특정 시점으로 롤 포워드해야 합니다.
- 로그의 끝 이전의 특정 시점으로 식별 컬럼이 있는 테이블이 들어 있는 테이블 공간을 복구하는 경우, 생성된 값 순서 사이에 갭이 있을 수 있습니다. 이 상황은 DB2가 일부 식별 컬럼에 대해 중복된 값을 생성하게 할 수 있습니다. 이 값은 사실 중복되지만, 데이터베이스의 내용을 보기만 하면 중복되게 표시되지 않습니다.
- 일정 조건하에서 갑자기 삭제된 테이블에서 데이터를 복구할 수 있습니다. 자세한 정보는 397 페이지의 『삭제된 테이블 복구』를 참조하십시오.
- QUIESCE TABLESPACES FOR TABLE을 발행하여 테이블 공간을 롤 포워드하는 데 사용할 수 있는 특정 시점 transaction-consistent를 작성할 수 있습니다. 테이블(공유된 상태, 갱신 가능 또는 독점적인)에 대한 테이블 공간을 quiesce하는 경우, 테이블 공간에 대한 새로운 요청을 블로킹하는 동안에 완료하기 위해 테이블 공간의 오브젝트에 액세스하는 수행중인 모든 트랜잭션에 대한 요청이(잠금을 통해) 대기합니다. Quiesce 요청이 권한 부여되는 경우, 모든 2단계 확약중 이상 실패의 트랜잭션은 이미 완료되었으며(확약 또는 구간 복원 되었으며) 테이블 공간은 일관성있는 상태입니다. 복구 실행기록 파일을 보면 quiesce 지점을 찾을 수 있고 롤 포워드의 예상 중지 시간을 결정하기 위해 최소한의 복구 시간이 지났는지를 점검할 수 있습니다.
- 특정 시점으로 테이블 공간을 롤 포워드하고 테이블 공간의 테이블이 참조 무결성 관계를 다른 테이블 공간에 포함된 테이블에서 간섭하는 경우, 특정 시점까지 두 테이블 공간을 동시에 롤 포워드해야 합니다. 그렇지 않은 경우, 두 테이블 공간은 특정 시점 롤 포워드 조작의 끝에서 점점 보류 상태에 있습니다. 두 테이블 공간을 동시에 롤 포워드하는 경우, 제한조건은 특정 시점 롤 포워드 조작의 끝에서 사용중입니다.
- 테이블 공간을 특정 시점으로 롤 포워드하고 테이블 공간에 있는 테이블이 다음 중 하나인 경우,
 - 다른 테이블 공간에 있는 요약 테이블에 대한 기본 테이블
 - 다른 테이블 공간의 테이블에 대한 요약 테이블
 테이블 공간을 동일한 특정 시점으로 롤 포워드해야 합니다. 그렇게 하지 않으면, 요약 테이블은 롤 포워드 조작 끝에서 점점 보류 상태가 됩니다.

- 특정 시점 테이블 공간 롤 포워드로 트랜잭션이 다른 테이블 공간에서 구간 복원되지 않고 확약되지 않는다는 것을 주의해야 합니다. 다음 경우에 발생할 수 있습니다.
 - 특정 시점 롤 포워드는 트랜잭션에 의해 갱신된 테이블 공간의 부속 집합에서 실행되고, 특정 시점은 트랜잭션이 확약된 시간 전입니다.
 - 특정 시점으로 롤 포워드되는 테이블 공간에 포함된 테이블은 연관된 트리거를 가지고 있거나, 롤 포워드되는 테이블 공간을 제외한 테이블 공간에 영향을 주는 트리거에 의해 갱신됩니다.
 이러한 상황이 일어나는 것을 막으려면 롤 포워드 중지 시간을 알아야 합니다.
- 테이블 공간 특정 시점 롤 포워드 조작이 완료된 후, 테이블 공간은 백업 보류 상태에 있게 됩니다. 사용자가 롤 포워드한 시간과 현재 시간 사이의 모든 갱신사항이 제거되지 않았기 때문에 테이블 공간을 백업해야 합니다. 이전 데이터베이스 또는 테이블 공간 백업에서 현재 시간으로 테이블 공간을 롤 포워드할 수 없습니다. 다음 예에서는 테이블 공간 백업이 필요한 이유와 사용 방법을 나타냅니다(테이블 공간을 사용 가능하게 하려면, 전체 데이터베이스, 백업 보류 상태의 테이블 공간 또는 백업 보류 상태의 테이블 공간을 포함하는 테이블 공간 세트를 백업할 수 있습니다).

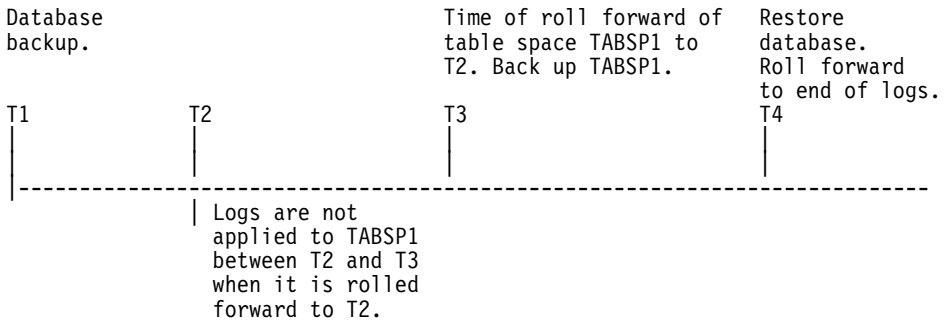


그림 11. 테이블 공간 백업 요구

위의 예에서, T1 시간에서 데이터베이스를 백업합니다. T3 시간에서, T2 시간으로 TABSP1을 롤 포워드하고 T3 후에 테이블 공간을 백업합니다(테이블 공간이 백업 보류 상태이므로, 테이블 공간을 백업해야 합니다. 테이블 공간 백업의 시간소인은 T3 다음이지만, 테이블 공간은 T2 시간입니다. 로그 레코드는 T2와 T3 사이로부터 TABSP1에

적용되지 않습니다). T4 시간에, T3 백업을 가진 데이터베이스를 복원하여 로그 끝까지 롤 포워드합니다. 테이블 공간 TABSP1은 T3 시간에 도달하면 복원 보류 상태에 있게 됩니다.

데이터베이스 관리 프로그램은 테이블 공간에 적용된 T2와 T3 사이의 로그 변경사항 없이 T3와 T4 사이에서 TABSP1에서 조작이 실행되었다고 가정하기 때문에 테이블 공간은 T3 시간에 복원 보류 상태가 됩니다. T2와 T3 사이의 로그 변경사항이 데이터베이스에서 ROLLFORWARD의 일부로 다시 적용된 경우, 이 가정은 위반됩니다. 테이블 공간이 특정 시점으로 롤 포워드된 후에 이루어져야 하는 필요한 테이블 공간 백업을 통해 이전 롤 포워드 특정 시점(T3 예에서)을 지나 테이블 공간을 롤 포워드할 수 있습니다.

테이블 공간 TABSP1을 T4로 복구하려고 가정하면, T3 후에 이루어진 백업(필요한 백업 또는 다음 백업)으로부터 테이블 공간을 복원한 다음 로그 끝까지 TABSP1을 롤 포워드합니다.

위의 예에서, 데이터베이스를 T4 시간으로 복원하는 가장 효과적인 방법은 다음 순서에 따라 필요한 단계를 실행하는 것입니다.

1. 데이터베이스를 복원하십시오.
2. 테이블 공간을 복원하십시오.
3. 데이터베이스를 롤 포워드하십시오.
4. 테이블 공간을 롤 포워드하십시오.

데이터베이스를 롤 포워드하기 전에 테이블 공간을 복원하므로, 데이터베이스가 롤 포워드된 경우에 로그 레코드를 테이블 공간에 적용하기 위해 자원이 사용되지는 않으며, 테이블 공간을 복원하기 전에 데이터베이스를 롤 포워드하는 경우에는 적용할 수 있습니다.

T3 시간 후인 TABSP1 백업 이미지를 찾을 수 없거나 TABSP1을 T3 또는 그 이전으로 복원하려면, 다음을 수행할 수 있습니다.

- 테이블 공간을 T3 시점으로 롤 포워드하십시오. 테이블 공간이 데이터베이스 백업을 통해 복원되었기 때문에 테이블 공간을 복원할 필요는 없습니다.
- T1 시간의 데이터베이스의 백업으로부터 테이블 공간을 다시 복원하고 T3 시간 이전으로 테이블 공간을 롤 포워드하십시오.
- 테이블 공간을 삭제하십시오.

파티션된 데이터베이스 환경에서 동시에 같은 특정 시점으로 테이블 공간의 모든 부분을 롤 포워드해야 합니다. 이렇게 하면, 테이블 공간은 각 데이터베이스 파티션에서 일관된 상태가 됩니다.

ROLLFORWARD 명령 사용 계획

ROLLFORWARD 명령을 사용하기 전에 다음과 같은 항목을 고려해야 합니다.

- SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다.
- 데이터베이스는 지역 또는 원격이 될 수 있습니다.
- 파티션된 데이터베이스 환경에서, 롤 포워드는 데이터베이스의 카탈로그 노드에서 발행되어야 합니다.
- 데이터베이스는 롤 포워드 복구용으로 구성되어야 합니다(즉, *logretain*, *userexit* 중 하나 혹은 둘다 사용 가능해야 합니다). 먼저 롤 포워드 기능에 맞도록 데이터베이스가 구성되면, 이에 대한 백업 사본을 작성해야 합니다.
- 데이터베이스는 롤 포워드되기 전에 복원되어야 하지만(RESTORE 명령을 사용하여), 테이블 공간은 그렇지 않습니다. 테이블 공간은 임시로 롤 포워드 보류 상태에 있게 되지만, 이 상태로 설정하기 위해 복원이 필요하지 않습니다(예를 들어, 전원 인터럽트가 발생하는 경우).
- 데이터베이스 롤 포워드는 오프라인을 수행합니다. 롤 포워드가 완료될 때까지 데이터베이스를 사용할 수 없습니다(테이블 공간 롤 포워드중 로그의 끝에 도달하거나 ROLLFORWARD 명령에 STOP을 지정하여). 그러나, SYSCATSPACE를 제외한 모든 테이블 공간의 온라인 롤 포워드를 수행할 수 있습니다. 테이블 공간에서 온라인 롤 포워드를 실행하는 경우, 사용할 수는 없습니다. 그러나, 데이터베이스의 다른 테이블 공간은 그렇지 않습니다.
- 롤 포워드시에는 다음을 수행해야 합니다.
 1. ROLLFORWARD를 발행하십시오(STOP 옵션을 사용하지 않음).
 2. ROLLFORWARD QUERY STATUS를 발행하십시오.

로그의 끝 포워드 복구를 실행하는 경우, QUERY STATUS는 QUERY STATUS에 의해 리턴된 특정 시점이 생각보다 빠른 경우 로그 파일이 유실된다는 것을 나타낼 수 있습니다.

로그의 끝 포워드 복구를 실행하는 경우, QUERY STATUS로 사용자는 롤 포워드가 올바른 지점에 있음을 확인하도록 도와줍니다.

3. ROLLFORWARD STOP을 발행하십시오. ROLLFORWARD STOP 후, 추가 변경사항을 롤 포워드할 수 없습니다.

- DB2의 현재 버전을 사용하여 작성된 백업을 부분적으로 또는 부속 집합으로 나누어 복원할 수 있습니다. 이전 버전의 DB2에서는 불가능합니다.
- 테이블 공간은 롤 포워드 보류 상태에 있는 경우에 롤 포워드 복구가 필요합니다. 테이블 공간은 미디어 오류로 인해 테이블 공간 레벨 복원 또는 오프라인을 취한 다음에 이 상태로 됩니다.

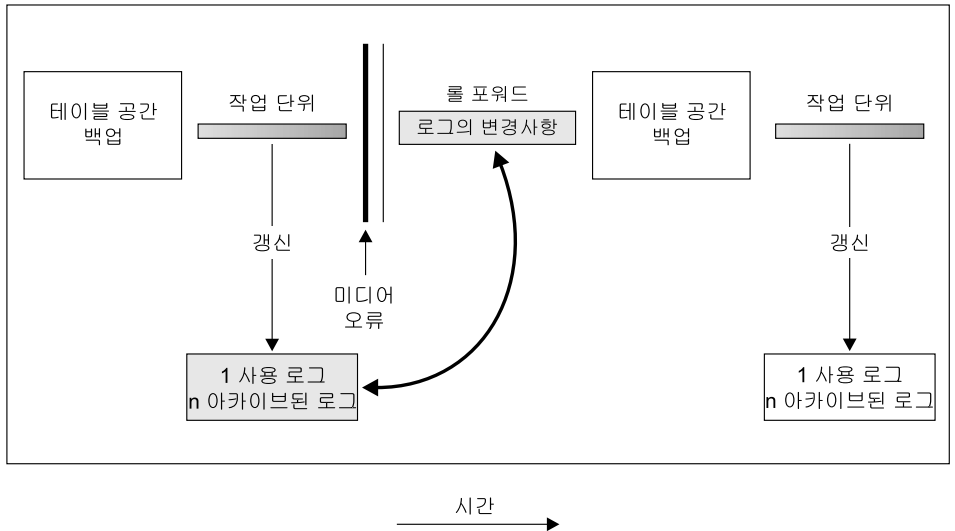


그림 12. 테이블 공간 롤 포워드 복구

- 가장 최근의 데이터베이스 백업 사본으로 데이터베이스를 복구할 필요는 없습니다. 해당 백업과 연관되며 그 다음에 오는 로그가 있으면, 어떠한 백업으로도 시작할 수 있습니다.
- 복구 시간을 줄이기 위해서는 계속해서 데이터베이스를 주기적으로 백업해야 합니다.
- 롤 포워드 조작을 취소하고(즉, ROLLFORWARD STOP이 지정되지 않았거나 ROLLFORWARD 명령이 실패한 경우) 다시 시작해야 하면, ROLLFORWARD CANCEL을 사용하여 조작을 취소할 수 있습니다.

데이터베이스에 대해 ROLLFORWARD CANCEL을 사용하는 경우, 롤 포워드는 데이터베이스에 대한 진행 여부와 상관없이 데이터베이스를 복원 보류 상태로 둡니다.

테이블 공간에 대한 ROLLFORWARD CANCEL 활동은 다음과 같습니다.

- ROLLFORWARD CANCEL을 발행하고 롤 포워드 보류 상태인 테이블 공간 목록을 지정한 경우, 해당 테이블 공간은 복원 보류 상태가 됩니다. 이 경우, 진행중인 롤 포워드 명령은 없습니다.

주: 테이블 공간 목록을 지정하지 않으면, SQL4906이 발행됩니다.

- 다중 테이블 공간이 로그 끝까지 롤 포워드되는 중이고 목록에서 ROLLFORWARD CANCEL을 지정한 경우, 목록에 있는 테이블 공간만이 복원 보류 상태가 됩니다. 목록에 없는 테이블 공간은 롤 포워드 진행 상태로 남아 있습니다. 목록 없이 ROLLFORWARD CANCEL을 지정하면, 롤 포워드 진행 상태의 모든 테이블 공간은 복원 보류 상태에 있게 되고 ROLLFORWARD 명령은 진행되지 않습니다.
- ROLLFORWARD CANCEL을 발행하고 하나 이상의 테이블 공간이 특정 시점으로 롤 포워드 중인 경우, 해당 테이블 공간은 목록 지정 여부와는 상관 없이 모두 복원 보류 상태가 됩니다. 목록을 지정하더라도, 목록은 무시되고 롤 포워드 진행중인 모든 테이블 공간은 복원 보류 상태에 있게 되고 ROLLFORWARD 명령은 더 이상 진행되지 않습니다.

주: ROLLFORWARD CANCEL을 사용하여 수행중인 롤 포워드 조작을 취소할 수 없습니다. 이 명령을 사용하면, 완료되었지만 ROLLFORWARD STOP 이 발행되지 않았거나, 완료되기 전에 실패한 롤 포워드 조작만을 취소할 수 있습니다.

DATALINK 컬럼이 들어 있는 테이블이 있을 경우, 415 페이지의 『복원 및 롤 포워드 유틸리티 고려사항』에서 자세한 내용을 참조하십시오.

버전 2 클라이언트에서는 파티션된 데이터베이스를 롤 포워드할 수 없습니다.

ROLLFORWARD 명령 호출

ROLLFORWARD 명령을 호출하기 전에 고려해야 할 몇 가지 사항이 있습니다.

- ROLLFORWARD 명령을 시작할 때, 시간을 지정하여 데이터베이스 로그로부터 복구될 트랜잭션을 제한할 수 있습니다. BACKUP 명령의 온라인 옵션을 사용하여 작성된 백업으로부터 복원하는 경우, ROLLFORWARD 명령 시간은 온라인 백업 종료 시간보다 나중 시간이어야 합니다.
- 로그는 작업 단위(UOW) 완료와 연관된 시간소인을 사용합니다. 로그의 시간소인은 UTC(Coordinated Universal Time)를 사용하는데, 이는 다른 로그와 연관된 동일한 시간소인(예를 들어, 일광 절약시간과 연관된 시간상의 변경 때문에)을 갖지 않도록 하는 데 도움이 됩니다. 백업에 사용된 시간소인은 BACKUP

이 시작된 현지 시간에 근거합니다. 결과적으로, ROLLFORWARD 명령을 호출할 때 시간을 국제 표준 시간으로 지정해야 합니다.

주:

1. 특수 레지스터인 CURRENT TIMEZONE은 UTC 및 응용프로그램 서버 데이터베이스의 현지 시간간의 차이를 가지고 있습니다. 현지 시간은 UTC에 현재 시간대 값을 더한 것입니다.
 2. 테이블 공간을 특정 시점으로 롤 포워드중인 경우, 적어도 최소 복구 시간으로 롤 포워드해야 합니다. 최소 복구 시간은 이 테이블 공간 또는 해당 테이블의 시스템 카탈로그를 마지막으로 갱신한 시점입니다. LIST TABLESPACES SHOW DETAIL 명령을 사용하면, 테이블 공간에 대한 최소 복구 시간을 구할 수 있습니다. 이 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.
- 온라인 백업이 끝난 지점을 통과하기 전에 ROLLFORWARD 작업을 중단하면, 데이터베이스가 롤 포워드 보류 상태로 됩니다. 테이블 공간이 롤 포워드 중인 경우, 롤 포워드 진행 상태로 유지됩니다.

로드 사본 위치 파일 사용

DB2LOADREC 레지스트리 변수는 로드 사본 위치 정보로 파일을 식별하는 데 사용됩니다. 이 파일은 롤 포워드 복구 동안에 로드 사본을 위치시키는 데 사용됩니다. 이 파일에는 다음 정보가 있습니다.

- 미디어 유형
- 사용될 미디어 장치의 수
- 테이블 로드시 생성된 로드 사본의 위치
- 적용 가능한 경우, 로드 사본의 파일 이름

위치 파일이 없거나, 파일에 일치하는 항목이 없으면, 로그 레코드의 정보가 사용됩니다.

파일의 정보는 롤 포워드 복구가 발생하기 전에 겹쳐볼 수 있습니다.

주:

1. 파티션된 데이터베이스 환경에서, DB2LOADREC 레지스트리 변수는 db2profile 파일에 있어야 합니다.

- 파티션된 데이터베이스 환경에서, 로드 사본 파일이 데이터베이스 파티션 서버마다 존재해야 하고 파일 이름(경로를 포함한)은 일치해야 합니다.
- DB2LOADREC 레지스트리 변수로 식별된 파일의 항목이 유효하지 않으면, 이전 로드 사본 위치 파일이 유효하지 않은 항목을 대체할 정보를 제공하기 위해 사용됩니다.

다음 정보는 위치 파일에서 제공됩니다. 처음 5개의 매개변수는 유효한 값을 가져야 하며, 로드 사본을 식별하는 데 사용됩니다. 전체 구조는 기록된 로드 사본 각각에 대해 반복됩니다. 예를 들면, 다음과 같습니다.

```

TIMestamp      19950725182542      * Timestamp generated at load time
SCHema         PAYROLL             * Schema of table loaded
TABlename     EMPLOYEES           * Table name
DATabasename  DBT                 * Database name
DB2instance   TORONTO             * DB2INSTANCE
BUFFernumber  NULL                          * Number of buffers to be used for recovery
SESSionnumber NULL                  * Number of sessions to be used for recovery
TYPEofmedia   L                          * Type of media - L for local device
                                           A for TSM
                                           0 for other vendors

LOCationnumber 3                    * Number of locations
  ENTRY       /u/toronto/dbt.payroll.employes.001
  ENT         /u/toronto/dbt.payroll.employes.002
  ENT         /dev/rmt0

TIM           19950725192054
SCH          PAYROLL
TAB          DEPT
DAT          DBT
DB2          TORONTO
SES          NULL
BUF          NULL
TYP          A
TIM         19940325192054
SCH        PAYROLL
TAB        DEPT
DAT        DBT
DB2        TORONTO
SES        NULL
BUF        NULL
TYP        0
SHRlib     /@sys/lib/backup_vendor.a

```

주:

- 각 키워드의 처음 3자가 중요합니다. 모든 키워드는 지정된 순서로 요구됩니다. 공백 행은 허용되지 않습니다.
- 시간소인의 형식은 *yyyymmddhhmmss*입니다.
- 널(NULL)이 될 수 있는 BUF 및 SES를 제외한 모든 필드는 필수입니다. SES가 널(NULL)이면, 구성 매개변수 NUMLOADRECSES가 지정한 값이 사용됩니다. BUF가 널(NULL)이면, 기본값은 SES+2입니다.

- 위치 파일에 있는 항목 중 하나가 유효하지 않은 경우, 이전의 로드 사본 위치 파일이 이러한 항목을 제공하기 위해 사용됩니다.
- 미디어 유형은 지역 장치(테이프, 디스크 또는 디스켓의 경우 L), TSM(A) 또는 기타 벤더(O)가 될 수 있습니다. 미디어 유형이 'L'이면, 위치 항목 앞에 위치의 수가 와야 합니다. 유형이 'A'이면, 더 이상의 입력이 필요없습니다. 유형이 'O'이면, 공유 라이브러리 이름이 필요합니다. 백업 미디어로서 TSM 및 기타 벤더 제품 사용에 대해서는 425 페이지의 『Tivoli 저장영역 관리 프로그램』에서 자세한 내용을 참조하십시오.
- SHRlib 매개변수는 LOAD COPY 데이터를 저장하는 기능을 가진 라이브러리를 가리킵니다.

주: LOAD COPY NO를 수행하고 데이터베이스의 백업 복사를 하지 않거나 LOAD를 수행중에 손상을 입은 테이블 공간을 복사하지 않으면, LOAD가 실행된 후에 데이터베이스 또는 테이블 공간을 특정 시점으로 복원할 수 없습니다. 즉, 롤 포워드 복구를 사용하여 데이터베이스 또는 테이블 공간을 LOAD 후의 상태로 다시 구축할 수 없습니다. LOAD 이전의 특정 시점으로 데이터베이스 또는 테이블 공간을 복원할 수만 있습니다.

특정 로드 사본을 사용하려면, LOAD 시간소인은 데이터베이스용 복구 실행기록 파일에 기록됩니다. 파티션된 데이터베이스 환경에서, 복구 실행기록 파일은 각 데이터베이스 파티션에 대해 지역입니다.

LOAD에 대해서는 데이터 이동 유틸리티 안내 및 참조서에서 자세한 내용을 참조하십시오.

삭제된 테이블 복구

데이터가 계속 필요한 하나 이상의 테이블이 갑자기 삭제되는 경우가 있습니다. 데이터를 잃어서는 안 되지만 갑자기 삭제되는 테이블이 있으면, 이 테이블을 삭제한 다음에 복구 가능한 것으로 작성해야 합니다.

데이터베이스 롤 포워드 다음에 데이터베이스 RESTORE를 사용하여 테이블의 데이터를 복구할 수 있습니다. 이것은 데이터베이스가 크고 복구하는 동안 데이터를 사용할 수 없게 되면 시간 낭비가 될 수 있습니다. 삭제된 테이블 복구를 이용하면, 테이블 공간 레벨 복원 및 롤 포워드를 사용하여 삭제된 테이블의 데이터를 복

구할 수 있습니다. 이는 데이터베이스 레벨 복구보다 빠르게 되며, 데이터베이스는 사용자가 사용할 수 있도록 그대로 남을 수 있습니다.

삭제된 테이블을 복구 가능하게 하기 위해, 테이블이 있는 테이블 공간에서는 DROPPED TABLE RECOVERY 옵션이 활성화되어야 합니다. 이것은 ALTER TABLESPACE문을 사용하여 또는 CREATE TABLESPACE 명령문 동안에 수행될 수 있습니다. 이들 명령문에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

DROPPED TABLE RECOVERY 옵션은 테이블 공간 고유의 것이며 일반 테이블 공간용으로만 지정될 수 있습니다. 테이블 공간에 이러한 특성이 있는지 알아보기 위해, *syscat.tablespaces* 카탈로그 테이블에서 테이블 공간 이름의 DROP_RECOVERY 컬럼을 조회할 수 있습니다.

DROP TABLE문이 DROPPED TABLE RECOVERY 옵션이 『ON』인 테이블 공간에 있는 테이블에 대해 수행되면, 로그 파일에 추가 로그 항목이 작성됩니다. 로그 항목에는 삭제된 테이블을 나타내는 정보가 들어 있습니다. 항목은 또한 복구 실행기록 파일에서 작성되어 테이블을 다시 작성하는 데 사용될 수 있는 정보가 들어갑니다.

다음은 수행하여 삭제된 테이블을 복구할 수 있습니다.

1. 삭제된 테이블 ID를 확보하십시오. 이 ID는 LIST HISTORY DROPPED TABLE 명령을 사용하여 복구 실행기록 파일에서 찾을 수 있습니다. 삭제된 테이블 목록 및 테이블을 다시 작성하는 데 필요한 정보가 표시됩니다.

주: 삭제된 테이블 ID는 백업 ID 컬럼 아래의 LIST HISTORY 출력에 나열됩니다.

이 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

2. 테이블이 삭제되기 전에 수행된 데이터베이스 또는 테이블 공간 레벨 백업을 복원하십시오.
3. ROLLFORWARD 명령에서 RECOVER DROPPED TABLE 옵션을 사용하여 삭제된 후의 특정 시점으로 롤 포워드하십시오. 이 옵션 사용시 필요한 다른 정보에는 삭제된 테이블 ID와 출력 파일이 기록될 디렉토리 경로가 포함

됩니다. 디렉토리 경로는 모든 데이터베이스 파티션에서 액세스 가능하거나, 각 파티션에 존재해야 합니다. 이 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

4. 복구 실행기록 파일에서 CREATE TABLE문을 사용하여 테이블을 다시 작성하십시오.
5. ROLLFORWARD 명령으로 내보내기된 데이터를 테이블로 가져오십시오.

내보내기된 데이터는 다음 이름 지정 규약을 사용하여 파일로 기록됩니다. ROLLFORWARD 명령에서 사용자가 지정한 *export_directory*에서, 각 데이터베이스 파티션에 의해 서브디렉토리가 작성됩니다. 사용자는 롤 포워드 요청을 발행하기 전에 서브디렉토리를 작성할 수 있습니다. 이것은 특정 드라이브 또는 머신으로 데이터를 내보내기 위해 사용될 수 있습니다. 서브디렉토리의 이름은 『NODEnnnn』이며, 여기서 *nnnn*은 데이터베이스 파티션 또는 노드 번호입니다. 각 서브디렉토리에서, 데이터 파일은 『데이터』 이름으로 내보내기됩니다. 데이터 파일에는 각각의 데이터베이스 파티션에 있으므로 삭제된 테이블의 데이터가 들어 있습니다.

삭제된 테이블에서 복구할 수 있는 데이터 유형에 대해서는 몇가지 제한사항이 있습니다. 한 번에 하나의 삭제된 테이블만이 복구될 수 있습니다. 여러 삭제된 테이블을 복구하려면, 위에 표시된 복구 순서가 다른 테이블이 복구될 때마다 수행되어야 합니다. 다음을 복구할 수는 없습니다.

- LOB 또는 LONG 데이터. DROPPED TABLE RECOVERY 옵션은 LONG 테이블 공간에 대해 지원되지 않습니다. LONG 테이블 공간에 대해 이를 사용하려면 SQL628N이 리턴합니다. LOB 또는 LONG VARCHAR 컬럼이 있는 삭제된 테이블을 삭제하려는 경우, 이들 컬럼은 생성된 내보내기 파일에서 NULL로 설정됩니다. DROPPED TABLE RECOVERY 옵션은 REGULAR 테이블 공간에 대해 『ON』으로 되어야 하지만, TEMPORARY 또는 LONG 테이블 공간은 그렇지 않습니다.
- DATALINK 컬럼과 연관된 링크 파일 이름은 복구될 수 있습니다. 데이터를 가져온 후, 테이블은 DB2 Data Links Manager와 조화되어야 합니다. 파일 백업은 가비지 콜렉션이 이를 이미 삭제했는지 여부에 따라 DB2 Data Links Manager에 의해 복원될 수도 안될 수도 있습니다.

- 행 유형과 연관된 메타 정보(데이터가 복구되지만 메타 데이터는 아님). 입력된 테이블의 계층 테이블에 있는 데이터가 복구됩니다. 이 데이터에는 입력된 테이블에 나타내는 삭제된 많은 정보가 들어 있을 수 있습니다.

로그 파일 관리를 위한 고려사항

다음은 데이터베이스 로그를 관리할 때 고려해야 하는 항목입니다.

- 아카이브된 로그용 번호 체계는 S0000000.LOG로 시작하고 S9999999.LOG까지입니다(10 000 000 logs). 데이터베이스 관리 프로그램은 다음 조건에서 S0000000.LOG를 사용하여 다시 시작합니다.

- 데이터베이스 구성 파일이 롤 포워드 기능을 작동시키도록 변경될 때
- 데이터베이스 구성 파일이 롤 포워드 기능을 작동중지하도록 변경될 때
- 로그가 랩될 때. 즉, S9999999.LOG 로그가 사용된 후

롤 포워드 복구 방법이 성공하면, 롤 포워드가 사용한 마지막 로그가 절단되고 로그는 다음 순서의 로그에서부터 시작됩니다. 실제적인 효과는 롤 포워드 복구에 사용된 마지막 로그보다 큰 순차 번호를 가진 로그 경로 디렉토리의 모든 로그가 재사용된다는 것입니다. ROLLFORWARD 명령을 실행하려면 로그 사본을 작성해야 합니다(User Exit 프로그램을 사용하여 다른 위치로 로그를 복사할 수 있습니다).

다음과 같은 이유로 다른 로그에 중복되는 이름을 가질 수 있습니다.

- 데이터베이스 관리 프로그램은 로그 이름을 S0000000.LOG로 지정하기 시작합니다(위에서 설명한 대로).
- 데이터베이스 관리 프로그램은 데이터베이스를 복원한 후 로그 이름을 재사용합니다(롤 포워드 복구와 함께든 아니든).

데이터베이스 관리 프로그램은 롤 포워드 복구시 틀린 로그가 적용되지 않도록 하지만, 필요한 로그의 위치를 검출할 수 없습니다. 롤 포워드 복구에 정확한 로그를 사용할 수 있도록 해야 합니다.

- *logpath* 데이터베이스 구성 매개변수에 의해 지정된 것을 제외한 위치에 로그 파일을 이동시키는 경우, ROLLFORWARD 명령의 OVERFLOW LOG PATH 매개변수를 사용하여 추가 경로를 지정하십시오.

데이터베이스 또는 테이블 공간의 변경사항을 롤 포워드하고 있으며 롤 포워드 조작이 다음 로그를 찾을 수 없을 경우, SQLCA에 로그 이름이 리턴되어 필요

한 다음 로그 파일을 지시하고, 롤 포워드 복구가 중단됩니다. 이때, 더 이상의 로그를 사용할 수 없으면 ROLLFORWARD 명령을 사용하여 처리를 중단할 수 있습니다.

롤 포워드 복구를 종료하고(ROLLFORWARD 명령의 STOP 옵션을 지정하여) 트랜잭션의 완료가 포함된 로그가 데이터베이스 또는 테이블 공간에 적용되지 않은 경우, 완료되지 않은 트랜잭션은 구간 복원되어 데이터베이스 또는 테이블 공간이 일치된 상태인 것을 확인합니다.

- 보존 로그는 로그 경로에 있습니다. 로그 경로의 기본값은 SQLOGDIR 서브 디렉토리이지만 *newlogpath* 구성 매개변수로 변경될 수 있습니다. 이를 다른 곳에 위치시키려면, User Exit 프로그램에 대한 데이터베이스를 작동 가능하게 설정하거나 로그 경로를 *newlogpath*로 변경하십시오. 이 경우, ROLLFORWARD 명령의 OVERFLOW LOG PATH 매개변수를 사용하여 롤 포워드시 이를 지적해야 합니다.
- 데이터베이스 구성 파일을 변경함으로써 User Exit 프로그램을 작동 가능하도록 하면, 보존 로그를 테이프 드라이브와 같은 사용자 정의 저장영역 장치로 재 지정할 수 있습니다. 또한 User Exit 프로그램을 사용하여 보존 로그의 저장영역을 관리할 수 있습니다. User Exit 프로그램에 대해서는 485 페이지의 『부록C. 데이터베이스 복구를 위한 User Exit』에서 자세한 내용을 참조하십시오.
- *newlogpath* 매개변수를 변경해도 기존의 보존 로그는 영향을 받지 않습니다. 로그의 위치에 대한 기록을 가지고 있어야 합니다.
- 롤 포워드 복구용으로 활성화된 데이터베이스가 롤 포워드 복구되지 않거나 특정 시점으로 롤 포워드되어 활성화될 경우, 로그 이름이 다시 사용되기 때문에 보존 로그는 둘 이상의 서로 다른 로그 순서에 연관됩니다(402 페이지의 그림 13은 작성된 로그의 예입니다. 지금 『백업 2』를 사용하여 복원을 수행한다면, 두 로그 순서가 사용될 수 있기 때문에 각별히 주의해야 합니다). 보존 로그를 제거하기 전에, 이 로그가 필요하지 않은지 확인해야 합니다.

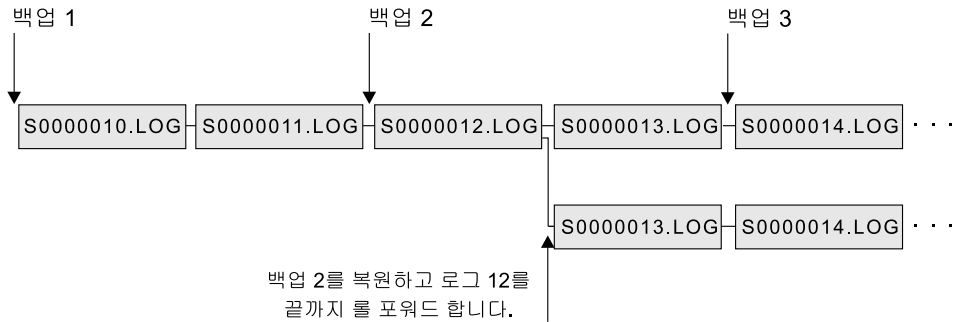


그림 13. 로그 파일 이름 재사용

- 전체 데이터베이스를 복구하는 동안 어느 한 시점까지를 롤 포워드하고 로그 중간에서 중단한 경우, 새로운 로그 순서를 작성하게 됩니다. 두 로그 순서는 결합될 수 없습니다. 첫번째 로그 순서에 걸쳐 있는 온라인 백업을 가진 경우, 첫번째 로그 순서를 사용하여 롤 포워드 복구를 완료해야 합니다.
- 복구 후에 새로운 로그 순서를 작성한 경우, 기존 로그 순서에 취해진 모든 테이블 공간 백업은 유효하지 않습니다. 이 경우, 복원은 테이블 공간 백업을 거부합니다. 백업이 더 이상 유효하지 않고(특히 온라인 백업의 경우) 복원이 성공적으로 수행되었음을 인식하는 데 실패한 인스턴스가 있을 수도 있습니다. 그러나 테이블 공간에 대한 롤 포워드가 실패하게 되며, 테이블 공간은 롤 포워드 보류 상태로 남아 있습니다.

위의 그림에서는 테이블 공간 백업인 백업 3은 맨 위 로그 순서에서 S0000013.LOG와 S0000014.LOG 사이에 완료됩니다. 데이터베이스 백업 2를 사용하여 복원하고 롤 포워드하는 경우, S0000012.LOG를 통해 롤 포워드해야 합니다. 이 다음부터는 맨 위 로그 순서나 새로운 맨 아래 로그 순서로 롤 포워드를 계속 수행할 수 있습니다. 맨 아래 순서로 롤 포워드되면, 테이블 공간 복원 및 롤 포워드 복구를 수행하는 데 테이블 공간 백업 3을 사용할 수 없습니다.

테이블 공간 백업 3을 사용하여 로그의 끝으로 테이블 공간 롤 포워드를 완료할 수 있게 하려면, 데이터베이스 백업 2를 사용하여 복원한 후 맨 위 로그 순서를 사용하여 롤 포워드해야 합니다. 일단 테이블 공간 백업 3이 복원되면, 로그의 끝으로 롤 포워드를 요청할 수 있습니다.

- 로그는 작업 단위(UOW) 완료와 연관된 시간소인을 사용합니다. 로그의 시간소인은 UTC(Coordinated Universal Time)를 사용하는데, 이는 다른 로그와 연관된 동일한 시간소인(예를 들어, 일광 절약시간과 연관된 시간상의 변경 때문에)을 갖지 않도록 하는데 도움이 됩니다. 백업에 사용된 시간소인은 현지 시간에 기초합니다. 결과적으로, ROLLFORWARD 명령을 호출할 때 시간을 국제 표준 시간으로 지정해야 합니다.

주: 특수 레지스터인 CURRENT TIMEZONE은 CUT 및 응용프로그램 서버 데이터베이스의 현지 시간간의 차이를 가지고 있습니다. 현지 시간은 CUT에 현재 시간대 값을 더한 것입니다.

원시 로그 사용

데이터베이스 로그에 대해 원시 장치를 사용할 수 있습니다. 이를 수행하는 데에는 장점과 단점 모두가 있습니다.

- 장점은 다음과 같습니다.
 - 27개 이상의 물리 드라이브를 시스템에 접속할 수 있습니다.
 - 파일 입출력 경로 길이가 더 짧습니다. 이것은 시스템에서 성능을 개선시킬 수 있습니다. 작업 로드에서 측정 가능한 이점이 있는지 여부를 평가할 벤치마크를 관리해야 합니다.
- 단점은 다음과 같습니다.
 - 장치는 다른 응용프로그램이 공유할 수 없으며, 전체 장치는 반드시 DB2에 지정되어야 합니다.
 - 장치에서 백업하거나 복사하는 운영 체제 유틸리티 또는 제3의 도구로 장치를 조작할 수 없습니다.
 - 잘못된 물리 드라이브 번호를 지정하는 경우, 기존 드라이브에서 파일 시스템이 손쉽게 지워질 수 있습니다.

newlogpath 데이터베이스 구성 매개변수로 원시 로그를 구성할 수 있습니다. 원시 장치를 지정하는 데 사용되는 구문 예는 137 페이지의 『원시 입출력』에서 자세한 내용을 참조하십시오. 그러나 이를 수행하기 전에, 위에 나열된 장점과 단점, 아래 나열된 추가 고려사항을 고려하십시오.

- 하나의 장치만을 사용할 수 있습니다. 운영 체제 레벨에서 여러 디스크에 장치를 정의할 수 있습니다. DB2는 운영 체제 호출을 수행하여 4KB 페이지의 장치 크기를 판별합니다.
여러 디스크를 사용하는 경우, 보다 큰 장치가 제공되고 결과로 인한 스트라이핑은 입출력 처리 속도를 높임으로써 성능을 향상시킵니다.
- DB2는 장치의 마지막 4KB 페이지에 기록하려고 시도합니다. 장치 크기가 2GB를 넘는 경우, 2GB를 넘는 장치를 지원하지 않는 운영 체제의 마지막 페이지

에 기록하려는 시도는 실패하게 됩니다. 이 경우, DB2는 지원되는 모든 페이지를 최대한 사용하려고 시도하게 됩니다.

장치 크기에 대한 정보는 운영 체제를 지원하는 DB2가 사용할 수 있는 장치 (4KB 페이지) 크기를 지정하는 데 사용됩니다. DB2가 기록할 수 있는 디스크 공간량을 *device-size-available*이라 합니다.

DB2는 장치의 처음 4KB 페이지를 사용하지 않습니다(일반적으로, 이 공간은 운영 체제가 다른 용도로 사용합니다). 즉, DB2가 사용할 수 있는 총 공간은 $device-size = device-size-available - 1$ 입니다.

- *logsecond* 매개변수는 사용되지 않습니다. DB2는 2차 로그를 할당하지 않습니다. 사용중인 로그 공간의 크기는 $logprimary \times logfilesiz$ 를 통해 산출되는 4KB 페이지 수입니다.
- 로그 레코드는 각 4KB의 로그 파일 크기(*logfilesiz*)를 갖는 로그 extent로 그룹화됩니다. 각 extent는 원시 장치에 하나씩 배치됩니다. 또한 각 extent는 extent 헤더용의 추가적 두 페이지로 구성됩니다. 즉, 장치가 지원할 수 있는 사용 가능한 로그 extent 수는 $device-size / (logfilesiz + 2)$ 입니다.
- 장치는 사용중인 로그 공간을 지원할 만큼 충분한 크기여야 합니다. 즉, 사용 가능한 로그 extent 수는 *logprimary* 구성 매개변수에 대해 지정된 값 이상이 어야 합니다.
- 순환 로그를 사용하는 경우, *logprimary* 구성 매개변수는 장치에 기록된 로그 extent 수를 결정합니다. 그 결과, 장치에 미사용 공간이 생기게 됩니다.
- User Exit 프로그램 없이 로그 보유(*logretain*)를 사용하는 경우, 사용 가능한 로그 extent 수 만큼 모두 사용하고 나면 갱신으로 인한 모든 조작용 로그 가득 참 오류를 수신하게 됩니다. 이 경우, 데이터베이스를 종료한 다음 오프라인 백업하여 복구할 수 있도록 해야 합니다. 데이터베이스 백업이 수행되고 나면, 장치에 기록된 로그 레코드는 유실됩니다. 즉, 조기 데이터베이스 백업을 사용하여 데이터베이스를 복원한 다음 롤 포워드할 수 없습니다. 사용 가능한 로그 extent 수까지 모두 사용하기 전에 데이터베이스 백업을 수행하면, 데이터베이스를 복원 및 롤 포워드할 수 있습니다.
- User Exit 프로그램과 함께 로그 보유(*logretain*)를 사용하는 경우, 각 로그가 로그 레코드로 가득 찰 때 각 로그 extent에 대해 User Exit 프로그램이 호출됩니다. User Exit 프로그램은 장치를 읽고 보존 로그를 파일로 저장할 수 있

어야 합니다. DB2는 로그 파일을 원시 장치로 검색하기 위해 User Exit 프로그램을 호출하지 않습니다. 그 대신, 롤 포워드 복구중에 DB2는 extent 헤더를 읽어 원시 장치에 사용될 로그 파일이 들어 있는지 여부를 판별합니다. 원시 장치에 필수 로그 파일이 없는 경우, DB2는 오버플로우 로그 경로를 검색합니다. 그래도 로그 파일이 없는 경우, DB2는 User Exit 프로그램을 호출하여 오버플로우 로그 경로로 로그 파일을 검색합니다. **rollforward** 명령에 오버플로우 로그 경로를 지정하지 않은 경우, DB2는 롤 포워드 조작중에 로그를 검색하기 위해 User Exit 프로그램을 호출하지 않습니다. User Exit 프로그램 호출에 관한 추가 정보는 492 페이지의 『UNIX 기반 또는 Windows NT 운영 체제용 형식』에서 자세한 내용을 참조하십시오.

- DPropR을 사용하고 있으며 로그를 원시 장치에 기록하는 경우, 읽기 로그 API는 로그 파일을 검색하기 위해 User Exit 프로그램을 호출하지 않습니다. 그러나, 요청된 로그 레코드를 장치에서 사용할 수 있는 경우 여전히 이 레코드가 리턴됩니다. 장치에서 가장 오래된 로그보다 이전의 로그를 요청하는 경우, 해당 로그는 리턴되지 않습니다(이 작동은 DB가 요청된 로그 레코드가 들어 있는 로그 파일을 찾을 수 없는 경우와 유사합니다).

주:

1. 원시 장치를 로그에 사용하는 경우, DPropR을 사용하지 않는 것이 좋습니다.
2. sqlurlog API를 사용하는 경우, 원시 장치를 로그에 사용할 수 없습니다.

로그 유실

- 데이터베이스를 삭제하면 현재 데이터베이스 로그 경로 디렉토리의 모든 로그가 지워집니다. 데이터베이스를 삭제하기 전에 로그의 사본을 작성해 두어야 합니다.
- 특정 시점으로 데이터베이스를 롤 포워드하는 중이면, 롤 포워드 복구에 사용되는 최종 로그와 그 다음에 오는 모든 기존의 로그가 다시 사용됩니다. 지나간 특정 시점을 복구할 능력을 유실하게 됩니다. 그러므로 특정 시점 복구를 시작하기 전에 현재 데이터베이스 로그 경로 디렉토리의 모든 로그를 복사해야 합니다.

롤 포워드 처리가 완료될 때, 마지막으로 파악된 트랜잭션이 있는 로그 파일이 절단되고 그 다음 순서의 로그부터 기록이 시작됩니다. 절단되기 전에 로그의 사

본과 상위 순차 번호를 갖는 로그 사본을 작성하지 않으면, 특정 시점을 지나간 데이터베이스를 복구할 수 없습니다(일단 정상적인 데이터베이스 활동이 롤 포워드 다음에 발생하면, 모든 후속 복구에 사용될 수 있는 새로운 로그가 작성됩니다).

- 로그 경로 디렉토리를 변경한 후 서브디렉토리를 제거하거나 로그 경로에서 호출한 서브디렉토리의 모든 로그를 지우면, 데이터베이스 관리 프로그램은 데이터베이스가 열릴 때 기본 로그 경로 `SQLLOGDIR`에서 로그를 찾습니다. 로그가 없으면, 데이터베이스는 백업 보류 상태가 되므로 먼저 데이터베이스를 백업해야 데이터베이스를 사용할 수 있습니다.

이 백업은 서브디렉토리에 빈 로그가 들어 있더라도 수행되어야 합니다.

- 온라인 백업 끝의 특정 시점을 포함하고 있는 로그를 유실하고 복원된 해당 이미지를 롤 포워드하고 있는 경우, 데이터베이스를 사용할 수 없게 됩니다. 데이터베이스를 사용할 수 있으려면, 다른 백업 및 모든 연관된 로그로부터 데이터베이스를 복원해야 합니다.

다음과 비슷한 상황이 발생할 수도 있습니다. 전체 데이터베이스에 대해 특정 시점 복구를 수행하려 하지만 복구 프로세스 동안 로그를 유실할 수도 있음을 고려합니다(이 시나리오는 최종 백업 데이터베이스 이미지의 시간과 데이터베이스를 복구하려는 시점 사이에 확장된 수의 아카이브 로그가 있는 경우 발생할 수 있습니다).

우선, 적용 가능한 모든 로그를 『안전한』 위치에 복사해야 합니다. 그런 다음, `RESTORE` 명령을 수행하여 데이터베이스에 대해 원하는 특정 시점으로 롤 포워드 복구 방법을 사용할 수 있습니다. 필요로 하는 로그 중에서 이 프로세스 동안 손상되거나 유실되는 것이 있으면, 사용자는 모든 로그의 백업 사본을 다른 위치에 보관합니다.

복구 실행기록 파일 정보

복구 실행기록 파일은 각 데이터베이스마다 작성되며 다음과 같은 경우에는 언제나 자동으로 갱신됩니다.

- 데이터베이스 또는 테이블 공간의 백업
- 데이터베이스 또는 테이블 공간의 복원

- 데이터베이스 또는 테이블 공간의 롤 포워드
- 테이블 공간의 변경
- 테이블 공간의 quiesce
- 테이블 공간의 이름 바꾸기
- 테이블의 로드
- 테이블 삭제
- 테이블 재구성
- 테이블 통계의 갱신

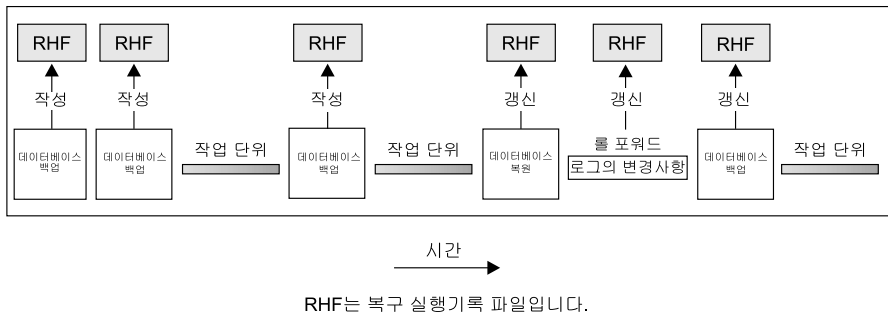


그림 14. 복구 실행기록 파일 작성 및 갱신

이 파일에서 요약된 백업 정보를 사용하여 주어진 시간에 데이터베이스의 모두 또는 일부를 복구할 수 있습니다. 이 파일의 정보에는 다음이 포함됩니다.

- 해당 항목을 고유하게 식별하는 각 항목과 연관된 식별(ID) 필드
- 복사된 데이터베이스의 부분과 방법
- 복사된 시간
- 사본의 위치(장치 정보 및 사본에 액세스하는 논리적인 방법을 명시)
- 복원이 수행된 최종 시간
- 테이블 공간의 이전 및 현재 이름을 표시하며 테이블 공간의 이름이 바뀐 시간
- 백업 상태: 사용중, 미사용중, 만기 또는 삭제
- 데이터베이스 백업이 저장되거나 롤 포워드 복구가 처리된 마지막 로그 순차 번호

복구 실행기록 파일에서 항목을 보려면, LIST HISTORY 명령을 사용하십시오. 이 명령에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

주: 복원 및 롤 포워드가 모든 로그의 끝으로 수행될 때, LIST HISTORY 명령 다음에 표시된 백업 ID는 끝 시간을 표시합니다. 즉, 백업 ID 값은 99991231235959입니다. 백업 ID는 롤 포워드가 수행될 때 이 방식으로 변환됩니다.

각 백업 조각(테이블 공간과 전체 데이터베이스 모두)은 복구 실행기록 파일의 사본을 포함하고 있습니다. 복구 실행기록 파일은 데이터베이스에 링크됩니다. 데이터베이스를 삭제하면 복구 실행기록 파일이 삭제됩니다. 데이터베이스를 새로운 위치로 복원하면, 복구 실행기록 파일이 복원됩니다. 복원은 기존의 실행기록 복구 파일을 겹쳐쓰지 않습니다.

현재 데이터베이스를 사용할 수 없거나 사용하지 않고, 연관된 복구 실행 기록 파일이 손상되거나 삭제되면 RESTORE 명령에 대한 옵션은 복구 실행기록 파일만 복원되도록 합니다. 그러면 복구 실행기록 파일은 데이터베이스를 복원하는 데 사용하는 백업의 정보를 제공하여 검토될 수 있습니다.

파일 크기는 파일에 있는 항목에 대한 보유 기간(일 단위)을 지정하는 *rec_his_retentn* 구성 매개변수에 의해 제어됩니다. 이 매개변수의 수가 0으로 설정되더라도 최신 전체 데이터베이스 백업과 복원 세트는 유지됩니다(이 사본을 제거하는 유일한 방법은 PRUNE을 FORCE 옵션과 함께 사용하는 것입니다). 보유 기간의 기본값은 366일입니다. 이 기간은 -1을 사용함으로써 무한정의 일수로 설정될 수 있습니다. 이 경우, 파일의 명시적 제거가 요구됩니다. 이 구성 매개변수에 대해서는 *관리 안내서: 성능*에서 자세한 내용을 참조하십시오.

API 함수 호출, 명령행 처리기 또는 제어 센터를 사용하여 복구 실행기록 파일에 대해 명령을 조회 및 수행할 수 있습니다. 5개의 기본 조회 및 명령은 OPEN, CLOSE, GET NEXT, UPDATE, 및 PRUNE입니다(명령 구문에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오. API 함수 호출에 대해서는 *Administrative API Reference*에서 자세한 내용을 참조하십시오. 제어 센터에 관한 자세한 정보는 사용중인 워크스테이션에서 제어 센터에 액세스하십시오).

실행기록 파일에 관한 자세한 정보는 SQLUHINFO 구조에 기록됩니다. 이 구조에 대해서는 *Administrative API Reference*에서 자세한 내용을 참조하십시오.

가비지 콜렉션

복구 실행기록 파일에서 설명한 DB2 데이터베이스 백업 수는 『DB2 가비지 콜렉션』에 의해 자동으로 모니터됩니다. 구성 매개변수 `num_db_backups`는 유지되는 "사용중" 백업 수를 정의합니다. 사용중인 백업은 데이터베이스의 현재 상태가 되도록 하기 위해 현재 로그를 사용하여 복원되거나 롤 포워드될 수 있는 것입니다. "사용중이 아님" 백업은 다른 세트의 로그 파일이 필요하므로 데이터베이스의 현재 상태에 도달하기 위해 복원되거나 롤 포워드될 수 없습니다.

다음 그림에 표시된 예에서는 `num_db_backups`가 4로 설정된 것으로 가정합니다.

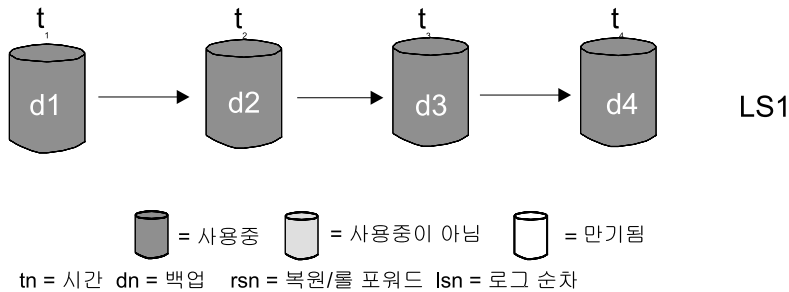


그림 15. 사용중인 데이터베이스 백업

더 이상 필요하지 않은 모든 데이터베이스 백업은 『만기됨』으로 표시됩니다. 이러한 백업은 보다 최근의 `num_db_backups`에서 정의된 대로 여러 데이터베이스 백업이 있으므로 더 이상 필요하지 않은 것으로 간주됩니다. 데이터베이스 백업이 만기되기 전에 수행된 모든 테이블 공간 백업 및 로드 백업 사본은 『만기』된 것으로 표시됩니다.

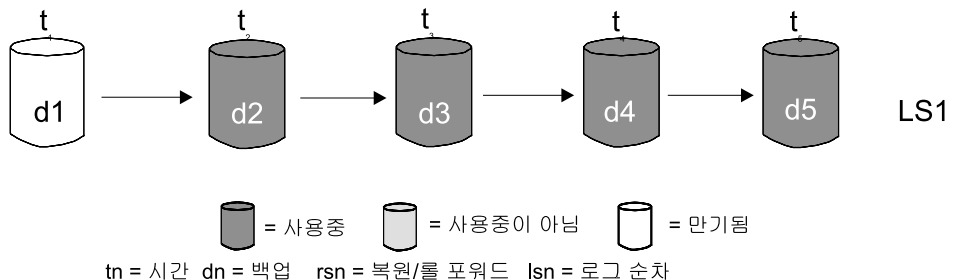


그림 16. 만기된 데이터베이스 백업

『사용중이 아님』으로 표시되고 만기된 데이터베이스 백업이 수행되기 전에 수행된 모든 데이터베이스 백업도 『만기됨』으로 표시됩니다. 연관된 모든 『사용중이 아님』 테이블 공간 백업 및 로드 백업사본도 『만기됨』으로 표시됩니다.

DATALINK 컬럼이 다음 절에서 설명된 것처럼 백업에 포함되면, DB2 Data Links Manager를 수행하는 모든 Data Links 서버는 백업이 만기되기 전에 링크되지 않은 연관된 Data Links 서버 파일의 가비지 콜렉션을 요청하기 위해 접속합니다. 실행기록 파일에 있는 정보를 근거로 이러한 백업을 물리적으로 삭제한 후, PRUNE HISTORY 명령을 사용하여 실행기록 파일에서 『만기됨』 항목을 제거할 수 있습니다. 실행기록 파일을 명시적으로 제거하지 않으려면, 다음 데이터베이스 백업때에는 『만기됨』 항목이 자동으로 제거됩니다.

DB2 가비지 콜렉션도 해당 백업이 현재 로그 체인이라고 하는 로그 순서에 해당되지 않으면 DB2 데이터베이스 또는 테이블 공간 백업에 대한 실행기록 파일 항목을 『사용중이 아님』으로 표시해야 합니다. 현재 로그 순서는 복원된 DB2 데이터베이스 백업과 처리된 로그 파일에 의해 결정됩니다. 일단 데이터베이스 백업이 복원되면, 복원된 백업 다음에 수행된 모든 데이터베이스 백업은 복원된 백업이 새로운 로그 체인을 시작하므로 『사용중이 아님』이 됩니다. 테이블 공간 백업을 복원한 후에는 『사용중이 아님』이 되고, DB2 데이터베이스의 현재 상태는 현재 로그 순서를 적용하여 도달할 수 없습니다. (DB2 데이터베이스 또는 테이블 공간 백업이 『사용중이 아님』이 되면, DB2 가비지 콜렉션에서는 해당 파일 백업 세트를 『사용중이 아님』으로 표시하도록 DB2 Data Links Manager를 수행하는 모든 Data Links 서버에게 알립니다.)

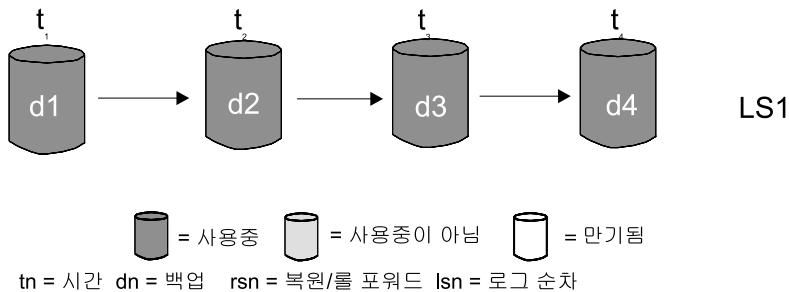


그림 17. 사용중인 데이터베이스 백업

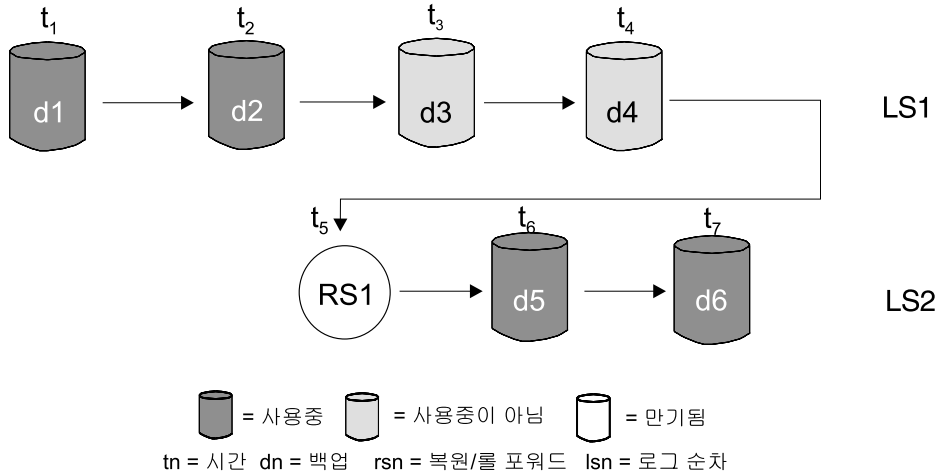


그림 18. 사용중이 아닌 데이터베이스 백업

DB2 가비지 콜렉션은 DB2 데이터베이스 백업을 완료한 후 호출됩니다. `db2_num_backups` 구성 매개변수 값은 마지막 항목에서부터 시작하여 현재 실행기록 파일을 검색하는 데 사용됩니다.

DB2 가비지 콜렉션은 로그를 롤 포워드하거나 롤 포워드하지 않고 데이터베이스 백업을 복원한 후에도 호출됩니다.

『사용중인』 데이터베이스 백업이 복원되었지만 실행기록 파일에 기록된 가장 최근의 데이터베이스 백업이 아닌 경우, 같은 로그 순서에 속하는 후속 데이터베이스 백업은 『사용중이 아님』으로 표시됩니다.

『사용중이 아님』 데이터베이스 백업이 복원되면, 현재 로그 순서에 속하는 모든 미사용 데이터베이스 백업은 다시 『사용중』으로 표시됩니다. 『사용중인』 데이터베이스 백업을 복원함으로써, 더 이상 현재 로그 순서에 있지 않은 모든 사용중인 데이터베이스 백업이 『사용중이 아님』으로 표시됩니다.

DATALINK 컬럼이 수행되는 데이터베이스 백업의 일부인 경우, DB2 가비지 콜렉션은 DB2 Data Links Manager를 수행중인 모든 Data Links 서버에 연결하여 해당 Data Links 서버에서 해당되는 파일 백업 세트를 같은 상태로 변경합니다.

모든 데이터베이스 백업 이후, `rec_his_retentn` 구성 매개변수는 실행기록 파일에서 『만기된』 항목을 제거하는 데 사용됩니다. 『만기된』 모든 백업이 제거됩니다.

PRUNE HISTORY 명령은 WITH FORCE 옵션이 발행되지 않으면 실행기록 파일에서 『만기된』 것으로 표시되는 백업만을 제거하려 할 때에는 언제든지 사용될 수 있습니다(『만기』되지 않은 백업이 제거되면, 모든 Data Links 서버가 연결되어 해당 파일 세트로 가비지 콜렉션 플래그를 요청합니다).

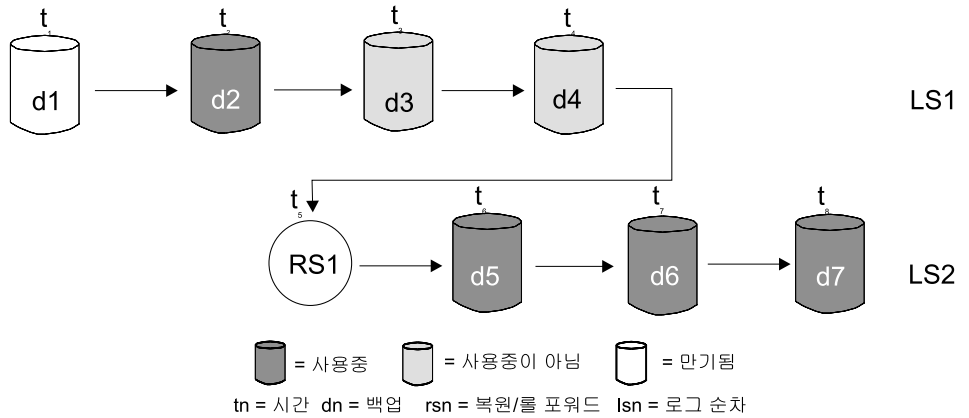


그림 19. 사용중, 사용중이 아님 및 만기된 데이터베이스 백업 혼합

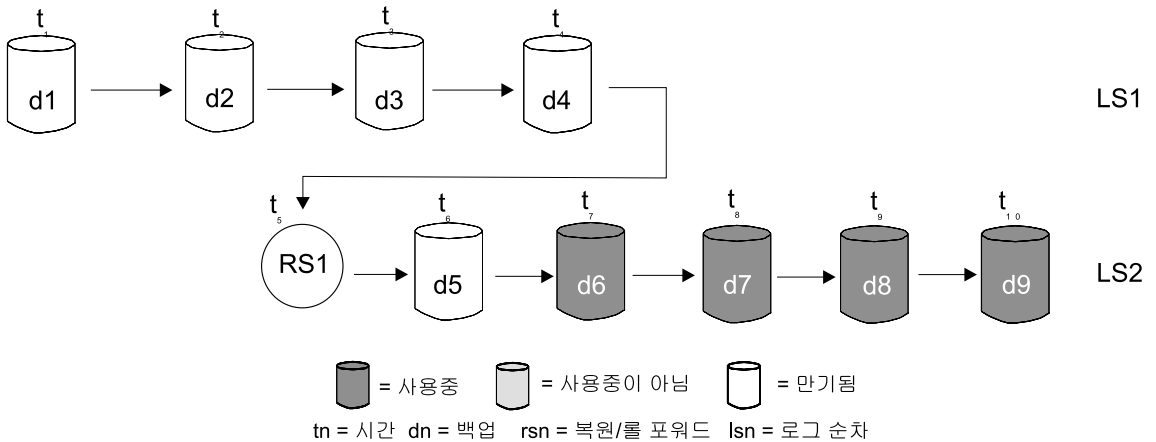


그림 20. 만기된 로그 순서

DB2 Data Links Manager 고려사항

다음 절에서는 DATALINK 컬럼이 들어 있는 테이블이 있는 경우 적용할 정보를 제공합니다. DATALINK 컬럼에 대해서는 *SQL 참조서*의 CREATE TABLE 문을 참조하십시오.

시스템 손상 복구 고려사항

응용프로그램이 DB2 Data Links Manager를 수행하는 Data Links 서버를 포함하여 SQL 요청을 발행하면(FILE LINK CONTROL 속성이 있는 DATALINK 컬럼을 사용하여), 데이터베이스 관리 프로그램은 작업을 Data Links 서버로 분배합니다. 또한, 트랜잭션에 포함된 Data Links 서버도 추적합니다. 응용프로그램이 트랜잭션에 대해 COMMIT를 발행하면, 데이터베이스 관리 프로그램은 2단계 확약 프로토콜을 사용하여 트랜잭션을 확약합니다. 1단계에서, 데이터베이스 관리 프로그램은 PREPARE 로그 레코드를 기록하고 모든 Data Links 서버로 PREPARE 요청을 분배합니다. 그런 다음, 각각의 Data Links 서버는 다음 중 하나로 응답합니다.

- YES; 데이터 링크 서버가 확약할 준비가 되었음을 나타냅니다.
- NO; 오류로 인해 데이터 링크 서버가 확약할 준비가 되어 있지 않습니다.

1단계는 모든 데이터 링크 서버가 『YES』로 응답했으므로 성공한 것으로 간주됩니다.

2단계 처리는 1단계 결과에 따라 달라집니다. 적어도 데이터 링크 서버 중 하나가 『NO』를 응답하면, 데이터베이스 관리 프로그램은 포함된 모든 데이터 링크 서버로 ABORT 요청을 전달합니다. 트랜잭션은 구간 복원되고 이유 코드가 『03』인 오류 메시지 SQL0903N이 응용프로그램으로 리턴됩니다. 그렇지 않으면, 데이터베이스 관리 프로그램은 데이터 링크 서버가 포함되지 않은 것처럼 트랜잭션을 확약합니다. 이 처리 끝에서, COMMIT 요청을 트랜잭션에 포함된 모든 데이터 링크 서버로 분배합니다.

데이터 링크 서버에서 트랜잭션을 PREPARE 상태에 놓이게 하는 오류가 발생하면, 이러한 트랜잭션을 2단계 확약중 이상 실패 트랜잭션이라고 합니다 데이터베이스 관리 프로그램은 이러한 트랜잭션 결과를 추적하고, 결국 데이터 링크 서버에서 이를 분석할 책임이 있습니다. 데이터베이스 관리 프로그램이 오류로 인해 데

이터 링크 서버에서 2단계 확약중 이상 실패 트랜잭션이 작성될 가능성이 있음을 결정할 때마다, 데이터 링크 서버의 상태를 시스템 손상 복구가 필요한 것으로 표시합니다. 이 상태에 있는 동안에는 데이터 링크 서버에 포함된 SQL 요청이 허용되지 않습니다. 이유 코드가 『03』인 SQL0357N이 SQL 요청을 작성한 응용프로그램으로 리턴됩니다.

RESTART, ACTIVATE DATABASE 또는 첫번째 CONNECT 처리시, 데이터베이스 관리 프로그램은 각각의 구성된 데이터 링크 서버로 연결하여 2단계 확약중 이상 실패 트랜잭션을 중지시키거나 확약하여 해결하려고 합니다. 데이터 링크 서버의 상태는 데이터베이스 관리 프로그램에서도 2단계 확약중 이상 실패 상태인 트랜잭션을 제외하고 2단계 확약중 이상 실패 트랜잭션 모두가 해결되면 사용 가능한 것으로 표시됩니다. 사용 가능한 상태에서는 데이터 링크 서버를 포함한 SQL 요청이 허용됩니다. 2단계 확약중 이상 실패 트랜잭션을 해결하기 위한 이러한 시도 끝에, 데이터베이스 관리 프로그램이 데이터 링크 서버에 해결이 필요한 잠재적인 2단계 확약중 이상 실패 트랜잭션이 있을 것으로 결정하게 되면, 데이터 링크 서버의 상태를 시스템 손상 복구가 필요한 것으로 표시합니다. 예를 들어, 이것은 데이터 링크 서버를 RESTART, ACTIVATE DATABASE 또는 첫번째 CONNECT 처리중에 사용할 수 없거나 데이터 링크 서버가 해당 처리 도중 오류를 발견하는 경우 발생할 수 있습니다.

데이터베이스로 구성된 데이터 링크 서버가 시스템 손상 복구가 필요한 상태에 있는 동안, 데이터베이스 관리 프로그램은 특별한 데이터 링크 서버가 포함된 SQL 요청을 허락하지 않습니다. 데이터베이스의 다른 데이터가 포함된 SQL 요청은 계속 허용됩니다. 데이터베이스 관리 프로그램은 비동기적으로 복구가 필요한 각각의 데이터 링크 서버에서 시스템 손상 복구를 시도하는 프로세스를 시작합니다. 프로세스가 시스템 손상 복구를 성공적으로 완료하면, 데이터 링크 서버 상태가 사용 가능한 것으로 표시되므로, 이를 포함한 더 이상의 SQL 요청이 허용됩니다.

백업 유틸리티 고려사항

DB2는 백업 유틸리티가 완료되는 시간에, DB2 Data Links Manager를 수행중인 데이터 링크 서버에 있는 링크 파일도 백업도록 합니다(백업 유틸리티는 온라인 또는 오프라인일 수 있으며, 백업 이미지는 데이터베이스 또는 테이블 공간일 수 있습니다). 다음 설명은 RECOVERY 매개변수를 YES로 설정한 DATALINK

컬럼에 의해 링크된 파일에만 적용됩니다(RECOVERY=NO가 지정된 DATALINK 컬럼으로 참조하는 파일은 백업되지 않습니다).

파일이 링크되면, 데이터 링크 서버는 TSM과 같은 아카이브 서버로 또는 디스크로 이들이 비동기적으로 복사되도록 스케줄합니다. 백업 유틸리티가 수행되면, DB2는 복사하도록 스케줄된 모든 파일이 복사되었는지 확인합니다. 백업 처리를 시작할 때, DB2는 또한 DB2 구성 파일에서 지정된 모든 데이터 링크 서버가 수행중인지 확인합니다. 데이터 링크 서버에 링크된 파일이 하나 이상 있으면, 백업 조장이 완료될 때까지 사용 가능해야 합니다. 백업 조장이 완료되기 전에 데이터 링크 서버를 사용할 수 없게 되면, 백업 조장은 불완전한 것으로 선언됩니다.

파일이 링크해제되면, ON UNLINK 매개변수에 지정한 값에 따라 이전 사용권한으로 리턴되거나 삭제됩니다. 백업 조장에 성공하면 데이터 링크 서버가 아카이브 서버(디스크 또는 TM)에 있는 아카이브 버전의 파일을 제거할 수 있게 됩니다. *num_db_backups* 데이터베이스 구성 매개변수는 파일의 아카이브 버전(링크해제된)이 제거되기 전에 DB2 데이터베이스 백업의 수를 지정합니다. 이 구성 매개변수에 대해서는 *관리 안내서: 성능에서* 자세한 내용을 참조하십시오.

링크해제된 파일이 제거되면, 링크해제된 파일에 대한 정보도 데이터 링크 서버 등록 테이블에서 제거됩니다.

복원 및 롤 포워드 유틸리티 고려사항

다음 정보는 테이블 공간에서 테이블에 대해 RECOVERY=YES가 정의된 DATALINK 컬럼이 있는 경우 적용됩니다. 테이블에 RECOVERY=NO 옵션이 정의된 DATALINK 컬럼이 있으면, 테이블은 복원 조작 끝에서 *Datalink_Reconcile_Pending* 상태가 됩니다. 423 페이지의 『데이터 링크 조정』에서 자세한 내용을 참조하십시오.

복원 조작 동안, DATALINK 컬럼이 있는 테이블은 다음 상태 중 하나가 됩니다.

- *Datalink_Reconcile_Not_Possible*

테이블이 *Datalink_Reconcile_Not_Possible* 상태이면, DATALINK 이외의 컬럼에 대해 제한되지 않은 조작 조치를 수행할 수 있습니다. DATALINK 컬럼이 SELECT문에 포함되면, 경고가 발행됩니다. UPDATE 호출을 DATALINK

컬럼에 대해 발행할 수 있습니다(제한사항에 대해서는 422 페이지의 『Datalink_Reconcile_Not_Possible 상태에서 테이블 제거』에서 자세한 내용을 참조하십시오). INSERT 및 DELETE문은 DATALINK 컬럼을 포함하므로 발행할 수 없습니다.

- *Datalink_Reconcile_Pending*

테이블이 *Datalink_Reconcile_Pending* 상태에 있으면, DATALINK 컬럼 이외의 컬럼에 대해 제한되지 않은 조작 조치를 수행할 수 있습니다. DATALINK 컬럼이 SELECT문에 포함되면, 경고가 발행됩니다. UPDATE, INSERT 또는 DELETE와 같은 DML문을 발행할 수 없습니다.

이 상태는 복원 또는 롤 포워드 유틸리티가 수행될 때 db2diag.log 파일에 보고됩니다. 또한, **db2dart** 명령을 사용하여 이 정보를 취득할 수도 있습니다.

데이터베이스 또는 테이블 공간을 복원하고 WITHOUT DATALINK 옵션이 지정되지 않은 경우, 복원 조작이 성공하려면 다음 조건을 충족시켜야 합니다.

- DATALINK 데이터가 들어 있는 모든 데이터 링크 서버 모두를 사용할 수 있어야 합니다.
- 백업 파일에 기록된 모든 데이터 링크 서버를 사용할 수 있어야 합니다.
- 백업 파일에 기록된 모든 DATALINK 컬럼에 대한 정보가 해당 데이터 링크 서버의 등록 테이블에 있어야 합니다.

DATALINK 컬럼에 대한 모든 정보가 등록 테이블에 기록되지 않으면, DATALINK 컬럼 정보가 누락된 테이블은 복원 조작(또는 롤 포워드 조작)이 완료된 후 *Datalink_Reconcile_Not_Possible* 상태가 됩니다.

백업이 등록 테이블에 기록되지 않으면, 이는 제공된 백업 파일이 *num_db_backups* 값보다 이전 것이므로 이미 『가비지 수집』되었음을 의미합니다. 이것은 이전 버전에서 백업의 아카이브 파일이 제거되었으므로 복원될 수 없음을 의미합니다. DATALINK 컬럼이 있는 모든 테이블은 *Datalink_Not_Possible* 상태가 됩니다.

테이블은 사용자가 사용할 수 있는 상태로 남지만, DATALINK 컬럼에 있는 값은 정확히 파일을 참조하지 않을 수 있습니다(예를 들어, DATALINK 컬럼 값에 대응되는 파일을 찾지 못할 수도 있습니다).

이러한 작동을 원하지 않는다면, 테이블 이름 TO DATALINK RECONCILE PENDING 명령에 대해 SET CONSTRAINTS를 발행하여 테이블을 점검 보류 상태가 되게 할 수 있습니다.

복원 조작 이후 테이블이 *Datalink_Reconcile_Not_Possible* 상태가 되면, 422 페이지의 『Datalink_Reconcile_Not_Possible 상태에서 테이블 제거』에서 제안된 방법 중 하나를 사용하여 DATALINK 컬럼 데이터를 수정할 수 있습니다.

주: 링크해제된 상태에서 링크된 상태로 파일을 표시하는 프로세스에서, 해당 파일은 파일 시스템에 대한 아카이브 서버에서 검색되어야 합니다. 이 프로세스 도중 오류가 발생하면(예를 들어 중복된 파일 이름으로 인해 파일 시스템으로 파일을 복사할 수 없음), 해당 테이블은 *Datalink_Reconcile_Pending* 상태가 됩니다.

데이터베이스 또는 테이블 공간을 복원하고 WITHOUT DATALINK 옵션을 지정하며 DATALINK 데이터가 들어 있는 하나 이상의 데이터 링크 서버를 사용할 수 없는 경우, 사용 불가능한 서버에 대한 DATALINK 값을 가진 테이블이 들어 있는 모든 테이블 공간은 복원 보류 상태가 됩니다.

롤 포워드 없이 오프라인 백업에서 데이터베이스 복원

주: 테이블 공간 레벨이 아닌 데이터베이스 레벨에서 롤 포워드 없이 복원만 가능합니다. 롤 포워드 없이 데이터베이스를 복원하려면, 회복 불가능한 데이터베이스(즉, 순환 로그를 사용하는 데이터베이스)를 복원하거나 복원 유틸리티를 위한 WITHOUT ROLLING FORWARD 매개변수를 지정합니다.

WITHOUT DATALINK 옵션과 함께 복원 유틸리티를 사용하는 경우, DATALINK 컬럼이 있는 모든 테이블은 *Datalink_Reconcile_Pending* 상태가 되고 복원 조작 도중 데이터 링크 서버와의 어떠한 조정도 수행되지 않습니다.

WITHOUT DATALINK 옵션을 사용하지 않고 모든 데이터 링크 서버를 사용할 수 있으며 DATALINK 컬럼에 대한 모든 정보가 등록 테이블에 모두 기록되면, 백업 파일에 기록된 각각의 데이터 링크 서버에 대해 다음이 발생합니다.

- 데이터베이스 복원에 사용된 백업 이미지 이후에 링크된 모든 파일은 링크해제된 것으로 표시됩니다(이들 파일은 백업 이미지에서 링크된 것으로 기록되지 않으므로).
- 백업 이미지 이후에 링크해제되었으나 백업 이미지가 형성될 때 링크된 모든 파일은 링크된 것으로 표시됩니다(이들 파일은 백업 이미지에서 링크된 것으로 표시되므로). 파일이 차후에 다른 데이터베이스에 있는 다른 테이블로 링크되면, 복원된 테이블은 *Datalink_Reconcile_Pending* 상태가 됩니다.

데이터베이스 및 테이블 공간 복원 및 로그 끝까지 롤 포워드

로그의 끝까지 데이터베이스 또는 테이블 공간을 롤 포워드하는 경우(즉 모든 로그가 제공된 경우), 조정 점검이 필요하지 않습니다(WITHOUT DATALINK 매개변수를 지정했는지 여부와 상관 없이). 롤 포워드 조작에 모든 로그가 제공되는지 여부가 확실하지 않거나, DATALINK 값을 조정해야 할 필요가 있다고 생각되는 경우, 다음을 수행하십시오.

1. 포함된 테이블에 대해 SQL문을 발행하십시오.

```
SET CONSTRAINTS FOR tablename TO DATALINK RECONCILE PENDING
```

이것으로 테이블은 *Datalink_Reconcile_Pending* 상태와 점검 보류 상태가 됩니다.

2. 테이블이 점검 보류 상태에 있지 않도록 하려면, 다음 SQL문을 발행하십시오.

```
SET CONSTRAINTS FOR tablename IMMEDIATE CHECKED
```

이것으로 테이블은 점검 보류 상태에서 벗어나지만, *Datalink_Reconcile_Pending* 상태는 그대로 유지됩니다. 이 상태로부터 테이블을 가져오려면 조정 유틸리티를 사용해야 합니다. 423 페이지의 『데이터 링크 조정』에서 자세한 내용을 참조하십시오.

데이터베이스 및 테이블 공간 복원 및 특정 시점까지 롤 포워드

Data Links 테이블에 대해 작업할 때에는 로그 끝 또는 특정 시점으로 롤 포워드할 수 있습니다.

특정 시점으로 롤 포워드된 테이블 공간의 테이블은 롤 포워드 조작끝에서 *Datalink_Reconcile_Pending* 상태가 됩니다. 이 상태로부터 테이블을 제거하려면 조정 유틸리티를 사용해야 합니다. 423 페이지의 『데이터 링크 조정』에서 자세한 내용을 참조하십시오.

특정 시점 롤 포워드 예

다음은 백업 및 복구를 핸들하기 위해 보유될 필요가 있는 파일을 보여주는 간단한 시나리오입니다. 이 예에서는 복구를 지원하기 위해 DB2 Data Links Manager가 보유해야 하는 파일과 함께 DATALINK 유형의 컬럼에서 단일 행 값에 대한 변경사항을 보여줍니다. 이 예의 경우, 최종 백업보다 이전 파일의 특정 시점 복구를 지원하지 않는다고 가정합니다. DB2 Data Links Manager를 수행하는 데이터 링크 서버에는 이러한 제한사항이 없습니다. 시간 3까지 fileA가 존재하는지 관찰하십시오. 이때 시간 2에서는 fileA가 링크해제되었으므로 삭제되며 이 예에 있는 데이터베이스에 대한 방침은 다음 백업이 수행될 때까지 링크해제된 파일을 보존하는 것입니다.(즉, *num_db_backups* 데이터베이스 구성 매개변수가 1로 설정됩니다.)

시간	1	2	3	4	5	6	7
활동	작성	갱신	백업	갱신	갱신	삭제	5로 복원
컬럼 값	valueA	valueB	valueB	valueC	valueD	-	valueD
링크된 파일	fileA	fileB	fileB	fileC	fileD	-	fileD
Data Links File Manager가 유지하는 기타 파일		fileA		fileB	fileB, fileC	fileB, fileC, fileD	fileB, fileC

주: 링크된 파일의 복구는 나머지 데이터베이스와 결합하여 수행됩니다.

DB2 Data Links Manager 및 복구 상호작용

다음 표에서는 사용자가 수행할 수 있는 서로 다른 유형의 복구, 복원 및 롤 포워드 처리 도중 발생하는 DB2 Data Links Manager 처리, 그리고 복구가 완료된 후 조정 유틸리티를 수행해야 하는지 여부를 보여줍니다.

복구 유형	복원중의 DB2 Data Links Manager 처리	롤 포워드중의 DB2 Data Links Manager 처리	조정
복구 불가능한 데이터베이스(logretain=NO)			
데이터베이스 복원	빠른 조정이 수행됩니다.	N/A	파일 링크의 문제점이 의심스러우면 선택적으로 수행할 수 있습니다.
WITHOUT DATALINK 옵션을 사용하여 데이터베이스 복원	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	N/A	필수
복구 가능한 데이터베이스(logretain=YES)			
WITHOUT ROLLING FORWARD 옵션을 사용하여 데이터베이스 복원	빠른 조정이 수행됩니다.	N/A	선택적
WITHOUT ROLLING FORWARD 및 WITHOUT DATALINK 옵션을 사용하여 데이터베이스 복원	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	N/A	필수
데이터베이스 복원 및 로그 끝까지 롤 포워드	조치 없음	조치 없음	선택적
WITHOUT DATALINK 옵션을 사용하여 데이터베이스 복원 및 로그 끝까지 롤 포워드	조치 없음	조치 없음	선택적
테이블 공간 복원 및 로그 끝까지 롤 포워드	조치 없음	조치 없음	선택적
WITHOUT DATALINK 옵션을 사용하여 테이블 공간 복원 및 로그 끝까지 롤 포워드	조치 없음	조치 없음	선택적
특정 시점까지 데이터베이스 복원 및 롤 포워드	조치 없음	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	필수
WITHOUT DATALINK 옵션을 사용하여 데이터베이스 복원하고 특정 시점까지 롤 포워드	조치 없음	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	필수
테이블 공간 복원 및 특정 시점까지 롤 포워드	조치 없음	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	필수

복구 유형	복원중의 DB2 Data Links Manager 처리	롤 포워드중의 DB2 Data Links Manager 처리	조정
WITHOUT DATALINK 옵션을 사용하여 테이블 공간 복원 및 특정 시점까지 롤 포워드	조치 없음	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	필수
롤 포워드 없이 다른 데이터베이스 이름, 별명, 호스트 이름 또는 인스턴스로 데이터베이스 복원(422 페이지의 주 참조)	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	N/A	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
다른 데이터베이스 이름, 별명, 호스트 이름 또는 인스턴스 및 롤 포워드로 데이터베이스 복원	조치 없음	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
롤 포워드 없이 사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서 데이터베이스 복원(422 페이지의 주 참조)	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	N/A	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서 데이터베이스 복원 및 롤 포워드	조치 없음	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서의 테이블 공간 복원 및 롤 포워드	조치 없음	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
WITHOUT DATALINK 옵션을 사용하여 롤 포워드 없이 사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서 데이터베이스 복원(422 페이지의 주 참조)	<i>Datalink_Reconcile_Pending</i> 상태의 테이블	N/A	필수

복구 유형	복원중의 DB2 Data Links Manager 처리	롤 포워드중의 DB2 Data Links Manager 처리	조정
WITHOUT DATALINK 옵션을 사용하여 롤 포워드와 함께 사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서 데이터베이스 복원	조치 없음	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.
WITHOUT DATALINK 옵션을 사용하여 롤 포워드와 함께 사용 불가능한 백업(이미지가 데이터 링크 서버에서 가비지 수집됨)에서 테이블 공간 복원	조치 없음	<i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블	선택적, 그러나 <i>Datalink_Reconcile_Not_Possible</i> 상태의 테이블은 수동으로 조정되어야 합니다.

주:

오프라인 백업 및 WITHOUT ROLLING FORWARD 옵션(logretain이 작동 상태) 또는 오프라인 백업(logretain이 작동 중지 상태)을 사용한 복원

Datalink_Reconcile_Not_Possible 상태에서 테이블 제거

DATALINK 컬럼이 있는 복원된 테이블은 *Datalink_Reconcile_Not_Possible* 상태가 됩니다.

- 테이블 공간이 *num_db_backup* 데이터베이스 구성 매개변수에 대해 지정한 값보다 더 이전 백업에서 복원되는 경우, 이 구성 매개변수에 대해서는 관리 안내서: 성능에서 자세한 내용을 참조하십시오.

DATALINK 컬럼 값이 유효하지 않더라도 DB2는 테이블이 액세스되도록 허용합니다. 불일치 DATALINK 컬럼 값이 있는 테이블에 대한 액세스를 막으려면, 테이블 이름 TO DATALINK RECONCILE PENDING 명령에 대한 SET CONSTRAINT를 발행하십시오. 다음과 같이 DATALINK 값을 갱신할 수 있습니다.

- SQL UPDATE문을 사용하여, 컬럼이 널(NULL) 입력 불가능하면 DATALINK 컬럼의 데이터 위치 부분을 길이가 0인 URL로 설정하거나, 컬럼이 널(NULL) 입력 가능하면 널(NULL)로 설정하십시오.

- 해당 데이터 링크 서버에서 파일을 복원하십시오. 그런 다음, DATALINK 컬럼 값을 읽기 위해 SELECT문을 발행하고 동일한 값으로 DATALINK 컬럼을 갱신하기 위해 UPDATE문을 발행하는 응용프로그램을 수행하십시오. DATALINK 컬럼 값이 갱신되는 동안 *Datalink_Reconcile_Not_Possible* 상태가 설정되어야 함을 기억하십시오. 갱신 조작이 완료된 후, 파일은 해당 데이터 링크 서버에서 링크된 것으로 표시됩니다.

그런 다음, 다음 명령을 발행하여 *Datalink_Reconcile_Not_Possible* 상태를 재설정합니다.

```
SET CONSTRAINTS FOR tablename DATALINK RECONCILE PENDING IMMEDIATE UNCHECKED
```

데이터 링크 조정

데이터 링크를 조정하기 위해 조정 유틸리티를 사용합니다. 유틸리티는 DB2에서 시작되며 DATALINK 컬럼 값이 참조하는 DB2 Data Links Manager를 수행하는 모든 데이터 링크 서버가 포함됩니다. 참조 파일이 데이터 링크 서버에 있는지, 아니면 링크가 재설정될 수 있는지 확인합니다. 다음 절에서는 데이터 링크를 조정할 필요가 있는지 여부를 DB2가 어떻게 검출하고 데이터 링크를 어떻게 조정하는가에 대해 설명합니다.

데이터 링크 서버 파일 참조가 없거나 재설정될 수 없는 경우, 조정 유틸리티는 이 유와 함께 오류가 있는 행 사본을 각 예외 테이블에 위치시킨 후 잘못된 행을 수정합니다. 예외 테이블이 지정되지 않으면, 다시 구성될 수 없는 파일 참조에 대한 DATALINK 컬럼 값은 컬럼 ID 및 이유와 함께 예외 보고서 파일로 복사됩니다. 예외 테이블(지정된 경우) 정보 또는 보고서를 사용하여 필요한 정정을 수행하도록 행을 갱신할 수 있습니다. 조정 유틸리티와 함께 사용되는 예외 테이블은 로드 유틸리티가 사용하는 예외 테이블과 같습니다. 로드 유틸리티에 대해서는 *데이터 이동 유틸리티 안내 및 참조서*에서 자세한 내용을 참조하십시오. 보고서는 *report.exp* 이름 지정 규약(조정 유틸리티는 .exp 확장자를 제공합니다)을 사용합니다. 예를 들어, 다음 명령문을 사용하여 조정 유틸리티를 호출할 수 있습니다.

```
db2 RECONCILE dept DLREPORT /u/scottba/report FOR EXCEPTION excptab
```

이 명령은 사용자가 작성한 dept라는 테이블을 조정하고, 예외 테이블 report.exp에 예외를 작성합니다. 조정 도중 링크해제된 파일에 대해서는 report.ulk 파일로 기록되며, 이것은 /u/scottba 디렉토리에 작성됩니다. *FOR EXCEPTION*

*excpstab*가 지정되지 않으면, 예외정보는 */u/scottba* 디렉토리에서 작성된 *report.exp* 파일로 기록됩니다. 조정 유틸리티에 대해서는 *Command Reference*에서 자세한 내용을 참조하십시오.

조정이 필요한 상황 검출

조정 유틸리티를 수행할 필요가 있는 상황은 다음과 같습니다.

- 전체 데이터베이스가 복원되며 특정 시점까지 롤 포워드됩니다. 전체 데이터베이스는 확장된 트랜잭션까지 롤 포워드되므로, 점검 보류 상태에는 테이블이 없습니다(참조 제한조건 또는 점검 제한조건에 따라). 데이터베이스의 모든 데이터를 일관된 상태로 가져옵니다. 그러나, DATALINK 컬럼은 DB2 Data Links Manager의 메타 데이터로 동기화될 수 없으며, 조정이 필요합니다.

이 상황에서, DATALINK 컬럼 데이터가 있는 테이블은 이미 *Datalink_Reconcile_Pending* 상태에 있습니다. 이들 각 테이블에 대해 조정 유틸리티를 발행해야 합니다.

- DB2 Data Links Manager를 수행하는 특별한 데이터 링크 서버는 해당 메타 데이터의 트랙을 유실합니다. 이것은 다른 이유로 발생할 수 있습니다. 예를 들면, 다음과 같습니다.
 - 데이터 링크 서버가 콜드 스타트되었습니다.
 - 데이터 링크 서버 메타 데이터가 백 레벨 상태로 복원되었습니다.

SQL UPDATE 및 DELETE와 같은 일부 상황에서, DB2는 데이터 링크 서버에서 메타 데이터의 문제점을 검출할 수 있습니다. 이러한 상황에서, DB2는 SQL문으로 실패합니다. SET CONSTRAINTS문을 사용하여 테이블을 *Datalink_Reconcile_Pending* 상태가 되게 한 다음, 해당 테이블에 대해 조정 유틸리티를 수행합니다.

- 파일 시스템은 사용 가능하지 않으며(예를 들어, 디스크 이상 때문에) 현재 상태로 복원되지 않습니다. 이 상황에서, 파일이 누락될 수 있습니다.

이와 같은 오류는 보통 데이터베이스에서 파일 참조를 얻는 파일에 액세스할 수 없을 때 응용프로그램이 발견합니다. 테이블을 *Datalink_Reconcile_Pending* 상태가 되게 하고 이에 대해 조정 유틸리티를 수행해야 합니다. 해당 DATALINK 컬럼이 RECOVERY=YES인 경우, 아카이브 서버에서 일부 파일이 복원될 수

있습니다. 어떤 경우, 조정 유틸리티는 예외 테이블 또는 예외 보고서에 예외를 기록합니다. 그런 다음, 이들 파일을 복원하거나 SQL UPDATE를 수행하여 컬럼을 수정할 수 있습니다.

조정을 위한 프로시저 요약

특정 시점 복구 때문에 또는 DB2 Data Links Manager 및 DB2 제어 정보를 수행하는 데이터 링크 서버가 일치하지 않으므로 데이터 링크를 조정할 필요가 있는 경우, 다음을 수행하십시오.

1. SET CONSTRAINTS문을 발행하여 테이블을 *Datalink_Reconcile_Pending* 상태가 되도록 하십시오(어떤 상황에서는 DB2가 이를 수행합니다).
2. 조정 유틸리티를 사용하여 링크를 해결하고, 예외 테이블 또는 예외 보고서에 있는 예외에 대한 해당 조치를 취하십시오.

Tivoli 저장영역 관리 프로그램

BACKUP 및 RESTORE 명령을 호출할 경우, 데이터베이스 또는 테이블 공간 백업을 관리하기 위해 Tivoli Storage Manager(TSM)(이전의 Tivoli ADSM) 제품을 사용하도록 지정할 수 있습니다. DB2와 함께 Tivoli ADSM 클라이언트 버전 3.1.x.3 이상을 사용할 수 있습니다. 다음 주제 항목에서는 추가 정보를 제공합니다.

- UNIX 기반 플랫폼용 Tivoli Storage Manager 클라이언트 설정
- 기타 플랫폼용 Tivoli Storage Manager 클라이언트 설정
- Tivoli Storage Manager 사용에 대한 고려사항

UNIX 기반 플랫폼용 Tivoli Storage Manager 클라이언트 설정

데이터베이스 관리 프로그램에서 TSM 옵션을 사용하기 위해서는, 먼저 다음과 같은 설정 활동이 수행되어야 합니다.

1. SunOS 및 Solaris 환경에서, 다음 단계를 수행하십시오(다른 UNIX 기반 플랫폼의 경우, 단계 426 페이지의 2에서 시작하십시오).
 - a. SunOS 5.5.1 또는 Solaris 2.5.1과 같이 필수 운영 체제 레벨이 설치되어 있는지 확인하십시오.

- b. Tivoli ADSM 클라이언트 버전 3.1.x.3 이상을 설치하십시오. 이 버전의 클라이언트를 설치하기 전에 이전 TSM 패키지 모두를 제거하십시오.
- c. TSM이 /opt/IBMDMap5, /opt/IBMDMba5 및 /opt/IBMDMSa5 디렉토리에 설치되어 있는지 확인하십시오.
- d. 없을 경우, /usr/lib 디렉토리에서 다음과 같은 기호 링크를 작성하십시오.

```
libApiDS.so -> libApiDS.so.1
libApiDS.so.1 -> /opt/IBMDMap5/api/libApiDS.so.2
```

2. TSM 사용자 구성 옵션 파일 /usr/sbin/dsm.opt 및 TSM 시스템 구성 옵션 파일 /usr/sbin/dsm.sys를 작성하거나 수정하여 사용자 환경에 맞추십시오.
3. SunOS 및 Solaris 환경에서, 다음 단계를 수행하십시오(다른 UNIX 기반 플랫폼의 경우, 단계 4를 계속하십시오).
 - a. /usr/sbin/dsm.opt 및 /usr/sbin/dsm.sys를 디렉토리 /opt/IBMDMap5로 복사하십시오.
 - b. /opt/IBMDMap5/solaris/dsmaptica를 디렉토리 /opt/IBMDMap5로 복사하십시오.
4. TSM이 사용하는 환경 변수를 설정하십시오.

DSMI_DIR API 신뢰 에이전트 파일(dsmaptica 또는 dsmtca)이 위치한 사용자 정의 디렉토리 경로를 식별합니다.

주: SunOS 및 Solaris 환경의 경우, 이것은 /opt/IBMDMap5로 설정되어야 합니다.

DSMI_CONFIG

dsm.opt 파일에 대해 사용자 정의 디렉토리 경로를 식별합니다. 이 경로에는 TSM 사용자 옵션이 들어 있습니다. 다른 두 개의 변수와 달리, 이 변수는 완전 경로 및 파일 이름을 포함해야 합니다.

주: SunOS 및 Solaris 환경의 경우, 이것은 /opt/IBMDMap5/dsm.opt로 설정되어야 합니다.

DSMI_LOG 오류 로그(`dsierror.log`)가 작성될 사용자 정의 디렉토리 경로를 식별합니다.

5. TSM 암호를 설정하십시오.

Tivoli 클라이언트가 TSM 서버와 인터페이스할 수 있으려면, 서버에 대한 암호가 있어야 합니다. 실행 가능 파일인 `dsmapiw`는 인스턴스 소유자의 `INSTHOME/sqllib/adsm` 디렉토리에 설치되어 있습니다. 이 실행 파일로 사용자는 TSM 암호를 설정 및 재설정할 수 있습니다.

`dsmapiw` 명령을 실행하려면 『root』 사용자로서 로그인되어야 합니다. 이 명령이 실행될 때, 다음과 같은 정보가 프롬프트됩니다.

- TSM 서버에 의해 인식되는 TSM 노드의 현재 암호인 기존 암호. 이 명령을 처음 실행할 때, 이 암호는 사용자의 노드가 TSM 서버에 등록될 때 TSM 관리자가 제공하는 암호입니다.
- TSM 서버에 저장될 노드의 새로운 암호인 새 암호(입력 오류를 점검하기 위해 새 암호를 두 번 입력하도록 프롬프트됩니다).

주: BACKUP 또는 RESTORE 명령을 실행하는 사용자는 이 암호를 알 필요가 없습니다. 이 명령을 수행해야 하는 유일한 경우는 초기 연결을 위해 암호를 설정할 때와 TSM 서버에 암호가 재설정되어 있는 경우입니다.

6. 데이터베이스 관리 프로그램이 수행되고 있는 경우, 다음을 수행해야 합니다.

- `db2stop` 명령을 사용하여 데이터베이스 관리 프로그램을 중지시킵니다.
- `db2start` 명령을 사용하여 데이터베이스 관리 프로그램을 시작합니다.

기타 플랫폼용 Tivoli Storage Manager 클라이언트 설정

데이터베이스 관리 프로그램에서 TSM 옵션을 사용하기 위해서는, 먼저 다음과 같은 설정 활동이 수행되어야 합니다.

1. TSM이 사용하는 환경 변수를 설정하십시오.

DSMI_DIR API 신뢰 에이전트 파일(`dsmapicta` 또는 `dsmtca`)이 위치한 사용자 정의 디렉토리 경로를 식별합니다.

DSMI_CONFIG

dsm.opt 파일에 대해 사용자 정의 디렉토리 경로를 식별합니다. 다른 두 개의 변수와 달리, 이 변수는 완전 경로 및 파일 이름을 포함해야 합니다.

DSMI_LOG 오류 로그(*dserror.log*)가 작성될 사용자 정의 디렉토리 경로를 식별합니다.

2. 사용자의 운영 체제에 적용 가능하면, TSM 시스템 구성 옵션 파일(*dsm.sys*)을 작성(또는 변경)하십시오.
3. *dsm.opt* TSM 사용자 구성 옵션 파일을 작성(또는 변경)하십시오. 환경 변수 **DSMI_CONFIG**는 이 파일을 카리킵니다.
4. TSM 암호를 설정하십시오.

Tivoli 클라이언트가 TSM 서버와 인터페이스할 수 있으려면, 서버에 대한 암호가 있어야 합니다. 실행 가능 파일인 *dsmapiw*는 인스턴스 소유자의 `\sql1lib\adsm` 디렉토리에 설치되어 있습니다 이 실행 파일로 사용자는 TSM 암호를 설정 및 재설정할 수 있습니다.

dsmapiw 명령을 실행하려면, 지역 관리자로서 로그인되어야 합니다 이 명령이 실행될 때, 다음과 같은 정보가 프롬프트됩니다.

- TSM 서버에 의해 인식되는 TSM 노드의 현재 암호인 기존 암호 이 명령을 처음 실행할 때, 이 암호는 사용자의 노드가 TSM 서버에 등록될 때 TSM 관리자가 제공하는 암호입니다.
- TSM 서버에 저장될 노드의 새로운 암호인 새 암호(입력 오류를 점검하기 위해 새 암호를 두 번 입력하도록 프롬프트됩니다).

주: BACKUP 또는 RESTORE 명령을 실행하는 사용자는 이 암호를 알 필요가 없습니다. 이 명령을 수행해야 하는 유일한 경우는 초기 연결을 위해 암호를 설정할 때와 TSM 서버에 암호가 재설정되어 있는 경우입니다.

5. 데이터베이스 관리 프로그램이 수행되고 있는 경우, 다음을 수행해야 합니다.
 - *db2stop* 명령을 사용하여 데이터베이스 관리 프로그램을 중지시킵니다.
 - *db2start* 명령을 사용하여 데이터베이스 관리 프로그램을 시작합니다.

Tivoli Storage Manager 사용에 대한 고려사항

TSM 안에서 특정 기능을 사용하려면, 기능을 사용하고 있는 오브젝트의 완전한 경로 이름을 제공해야 합니다(OS/2 및 Windows NT 플랫폼에서는 / 대신 \가 사용됨을 기억하십시오). 전체 경로 이름은 다음과 같습니다.

- 전체 데이터베이스 백업 오브젝트는 /<database>/NODEnnnn/FULL_BACKUP.timestamp.seq_no입니다.
- 테이블 공간 백업 오브젝트는 /<database>/NODEnnnn/TSP_BACKUP.timestamp.seq_no입니다.
- 로드 사본 오브젝트는 /<database>/NODEnnnn/LOAD_COPY.timestamp.seq_no입니다.

여기서, <database>는 데이터베이스 별명 이름이고, NODEnnnn은 노드 번호입니다.

주: 대문자로 표시된 이름은 표시된 대로 입력해야 합니다.

- 같은 데이터베이스 별명 이름을 사용하여 다중 백업을 가지려는 경우, 시간소인 및 순차 번호는 완전한 이름에서 구별되는 부분이 됩니다. 사용할 백업 버전을 결정하기 위해서는 TSM을 조회해야 합니다.
- 개별 백업은 TSM 그래픽 사용자 인터페이스에 알려지지 않았습니다. 백업 이미지는 TSM이 관리하는 파일 공간으로 풀(pool)됩니다. 개별 백업은 TSM API를 통하거나 이 API를 사용하는 *db2adutl*을 통해서만 조작될 수 있습니다.
- TSM 서버는 Tivoli 클라이언트가 서버의 구성 파일에서 COMMTIMEOUT 매개변수로 지정된 시간 동안 응답하지 않을 경우, 세션을 종료합니다. 다음 세 가지 요인으로 인해 이러한 시간종료 문제점이 발생할 수 있습니다.
 - COMMTIMEOUT 매개변수가 TSM 서버에서 너무 낮게 설정되어 있습니다. 예를 들어, 복원 동안 대형 DMS 테이블 공간이 작성되고 있을 경우, 시간 종료 발생할 수 있습니다.
이 매개변수에 권장된 값은 6 000초입니다.
 - 데이터베이스 관리 프로그램 백업(또는 복원) 버퍼가 너무 큼니다.
 - 데이터베이스 활동이 온라인 백업시 너무 높습니다.

- 데이터베이스 관리 프로그램은 TSM 전체 백업 옵션을 사용합니다. TSM 증분 식 백업은 지원되지 않습니다.
- throughput을 증가시키기 위해서는 다중 세션을 사용하십시오.
- UNIX 기반이 아닌 플랫폼에서, 백업 및 복원 유틸리티는 둘 이상의 TSM 세션을 허용하지 않습니다. Tivoli ADSM Client 버전 3.1.x.7(또는 이상) 또는 TSM Client 버전 3.7(또는 이상)을 사용 중인 경우, 둘 이상의 TSM 세션이 지원됩니다.

Windows 운영 체제 및 OS/2에 있는 현재 Tivoli 클라이언트(바로 위에서 언급)는 재입력이 가능하므로 여러 입출력 세션이 단일 머신에서 백업, 복원 또는 로드 유틸리티로 안전하게 작성될 수 있습니다. 그러나, 사용자는 설치된 TSM 클라이언트의 버전이 이 기능을 지원하도록 확인해야 합니다.

단일 노드 구성에서, 사용자가 다음과 같은 백업 명령을 발행하도록 시도하는 경우,

```
db2 backup db sample use tsm open 3 sessions
```

DB2는 TSM이 다중 세션을 지원하지 않음을 발견하고 SQL2032N을 리턴합니다. 같은 시나리오가 TSM을 사용한 로드 사본에도 적용됩니다.

그러나 Windows NT상의 다중 논리 노드(MLN) 구성에서, DB2는 각 논리 노드가 하나의 세션만을 작성하려고 하면 단일 머신에서 다중 세션이 사용됨을 발견 못할 수도 있음에 주의하십시오. 이러한 이유로 관련 TSM 클라이언트가 재입력을 지원함을 MLN 구성이 검증하는 것은 매우 중요합니다. 다중 논리 노드가 TSM을 사용하여 병렬로 백업, 복원 또는 로드중이면, DB2는 논리 노드가 실제로 같은 물리 하드웨어에 상주하는 경우라도 각 노드가 단일세션을 사용하려고 하면 조작을 진행하도록 합니다. 이것은 백업 시도가 실패하게 하며, 로드 프로세스를 정지시키므로, 가장 최근의 TSM 클라이언트 없이 시도해서는 안 됩니다.

TSM에서 백업 및 로그 아카이브 관리

db2adutl 유틸리티를 통해 백업, 로그, TSM을 사용하여 저장된 로드 사본 이미지를 조회, 발취 및 삭제할 수 있습니다. 유틸리티는 UNIX 플랫폼의 INSTHOME/sql1lib/misc 디렉토리에 설치되며, Intel 플랫폼의 \sql1lib\misc 디렉토리에 설치됩니다.

| 주: 백업으로 로그 파일을 유지하려고 할 수 있습니다. TSM을 사용할 때, TSM
| 의 제어에서 로그 파일을 이동하려고 합니다. 이것은 User Exit으로 수행됩니
| 다. 485 페이지의 『부록C. 데이터베이스 복구를 위한 User Exit』에서 User
| Exit를 사용하는 방법에 대한 자세한 내용을 참조하십시오.

| *db2adutl* 유틸리티를 통해 사용 가능한 모든 옵션은 다음과 같습니다.

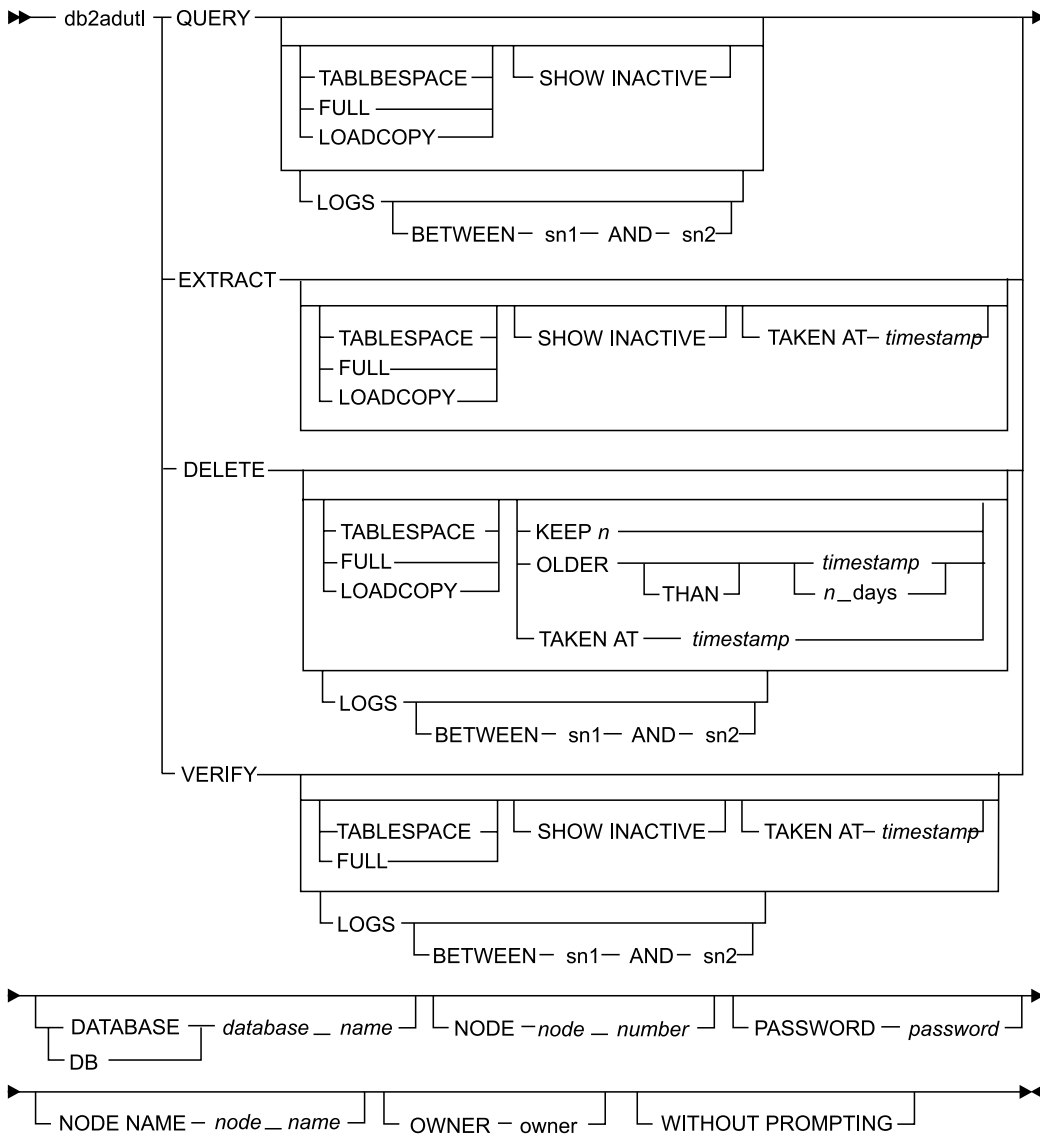


그림 21. db2adutil에 대한 구문

여기서,

QUERY

DB2 오브젝트에 대한 TSM 서버를 조회합니다.

EXTRACT

TSM 서버에서 지역 머신 및 디렉토리로 DB2 오브젝트를 복사합니다.

| **DELETE**

| 백업 오브젝트를 비활성화하거나 TSM 서버에서 로그 아카이브를 삭제합니다.

VERIFY

서버에 있는 백업 사본에 대해 일관성 점검을 수행합니다(이 매개변수를 사용하면 네트워크를 통해 전체 백업 이미지가 전송된다는 점을 유의하십시오).

TABLESPACE

테이블 공간 백업 이미지만을 포함합니다.

FULL 전체 데이터베이스 백업 이미지만을 포함합니다.

LOADCOPY

로드 사본 이미지만을 포함합니다.

LOGS 로그 아카이브 이미지만을 포함합니다.

BETWEEN *sn1* AND *sn2*

로그 순차 번호 1과 로그 순차 번호 2 사이의 로그를 사용하기 위해 이 옵션을 지정합니다.

SHOW INACTIVE

비활성화된 백업 오브젝트를 포함합니다.

TAKEN AT *timestamp*

시간소인으로 백업 이미지를 지정합니다.

KEEP *n*

시간소인으로 최근의 *n*을 제외하고 지정 유형의 모든 오브젝트를 비활성화합니다.

OLDER THAN *timestamp* 또는 *n_days*

timestamp 또는 *n*일 동안 보다 이른 시간소인을 가진 오브젝트가 비활성화됩니다.

DATABASE *database_name*

*database_name*만으로 연관된 오브젝트로 작업하도록 지정합니다.

NODE *node_number*

노드 *node_number*만으로 작성된 오브젝트로 작업하도록 지정합니다.

| **PASSWORD *password***

| 이 노드에 대한 TSM 클라이언트 암호를 지정합니다(필요한 경우). 특정 데이터

베이스가 지정되고 암호가 제공되지 않은 경우, *tsm_password* 데이터베이스 구성 매개변수에 지정한 값이 TSM에 전달되며, 그렇지 않은 경우 암호가 사용되지 않습니다.

NODENAME *node_name*

특정 TSM 노드 이름만으로 연관된 이미지로 작업하도록 지정합니다.

OWNER *owner*

*node_number*만으로 작성된 오브젝트로 작업하도록 지정합니다.

WITHOUT PROMPTING

오브젝트가 삭제되기 전에 검증하도록 프롬프트되지 않습니다.

각 명령을 사용할 때, DATABASE 매개변수를 사용하여 작업하고자 하는 데이터베이스를 선택할 수 있습니다. EXTRACT 및 DELETE 명령의 경우, WITHOUT PROMPTING 매개변수를 사용하여 선택을 확인하는 프롬프트가 제시되지 않도록 요구할 수 있습니다.

이 유틸리티의 QUERY 명령을 사용하여 백업, 로그 및 로드 사본 이미지를 나열할 수 있습니다. 백업은 전체 데이터베이스, 테이블 공간 또는 둘 다가 될 수 있습니다. 이 명령을 사용할 때 기본값은 두 가지 유형의 백업, 모든 로드 사본 이미지, 모든 로그를 나열하는 것입니다. 모든 로그를 보는 대신, 나열될 로그 범위를 선택할 수 있습니다. 사용중이 아닌 백업을 보기 위해 요청할 수도 있습니다.

이 유틸리티의 EXTRACT 명령으로 TSM으로부터 사용자의 현재 디렉토리 백업, 로그 또는 두 가지 모두를 TSM 서버에서 복사할 수 있습니다. 백업은 전체 데이터베이스, 테이블 공간 또는 둘다가 될 수 있습니다. 이 명령을 사용할 경우, 사용중인 백업 이미지 및 각 로그를 나열하는 것은 규정자가 없는 기본값입니다. 이때, 발췌할 백업 및 로그를 선택할 수 있습니다. 모든 로그를 보는 대신, 나열될 로그 범위를 선택할 수도 있습니다. 사용중이 아닌 백업을 보기 위해 요청할 수도 있습니다. 발췌할 특정백업은 TAKEN AT <timestamp> 매개변수를 사용하여 선택될 수 있습니다.

이 유틸리티의 DELETE 명령을 사용하여 로그를 삭제하거나 TSM으로부터 백업을 비활성화할 수 있습니다. 이 명령을 사용할 경우, 사용중인 백업 이미지 및 각 로그를 나열하는 것은 규정자가 없는 기본값입니다. 이때, 삭제/비활성화할 백업 및 로그를 선택할 수 있습니다. KEEP *n*으로 이 명령을 규정하여 최근의 *n* 백업을 보관할 수 있습니다. OLDER [THAN] <timestamp> 또는 *n* DAYS를 사용하여 명령을 규정할 수 있습니다. 이것은 주어진 날짜(시간소인)보다 오래되거나 지정된 날짜보다 오래된 백업을 삭제합니다. 모든 로

그를 보는 대신, 나열될 로그 범위를 선택할 수도 있습니다. 삭제할 특정 백업은
TAKEN AT <timestamp> 매개변수를 사용하여 선택될 수 있습니다.

DB2의 경우, TSM 기본 규정을 사용할 것을 권장합니다. 백업 이름 지정 규약에 대한 변
경사항으로 각 백업은 이제 고유합니다. 기존 백업을 삭제하려면, 사용중인 사본이 보관되
지 않도록 규정을 설정해야 합니다.

이 유틸리티 사용의 예는 『db2adutl 사용의 예』에서 자세한 내용을 참조하십시오.

db2adutl 사용의 예:

db2 backup database rawsamp1 use adsm

백업에 성공했습니다. 이 백업 이미지에 대한 시간소인은 19970929130942입니다.

db2adutl query

Query for database RAWSAMPL

Retrieving full database backup information.

```
full database backup image: 1, Time: 19970929130942,
                                Oldest log: S0000053.LOG, Sessions used: 1
full database backup image: 2, Time: 19970929142241,
                                Oldest log: S0000054.LOG, Sessions used: 1
```

Retrieving table space backup information.

```
table space backup image: 1, Time: 19970929094003,
                                Oldest log: S0000051.LOG, Sessions used: 1
table space backup image: 2, Time: 19970929093043,
                                Oldest log: S0000050.LOG, Sessions used: 1
table space backup image: 3, Time: 19970929105905,
                                Oldest log: S0000052.LOG, Sessions used: 1
```

Retrieving log archive information.

```
Log file: S0000050.LOG
Log file: S0000051.LOG
Log file: S0000052.LOG
Log file: S0000053.LOG
Log file: S0000054.LOG
Log file: S0000055.LOG
```

db2adutl delete full taken at 19950929130942 db rawsamp1

데이터베이스 RAWSAMPL에 대한 조회

완전한 데이터베이스 백업 정보 검색 중. 기다리십시오.

완전한 데이터베이스 백업 이미지: RAWSAMPL.0.db26000.0.19970929130942.001

이 백업 이미지를 비활성화하겠습니까(Y/N)? **y**

계속하시겠습니까(Y/N)? **y**

db2adut1 query

Query for database RAWAMPL

Retrieving full database backup information.

full database backup image: 2, Time: 19950929142241,
Oldest log: S0000054.LOG, Sessions used: 1

Retrieving table space backup information.

table space backup image: 1, Time: 19950929094003,
Oldest log: S0000051.LOG, Sessions used: 1
table space backup image: 2, Time: 19950929093043,
Oldest log: S0000050.LOG, Sessions used: 1
table space backup image: 3, Time: 19950929105905,
Oldest log: S0000052.LOG, Sessions used: 1

Retrieving log archive information.

Log file: S0000050.LOG
Log file: S0000051.LOG
Log file: S0000052.LOG
Log file: S0000053.LOG
Log file: S0000054.LOG
Log file: S0000055.LOG

호스트에서 2단계 확약중 이상 실패 트랜잭션 복구

응용프로그램이 트랜잭션 중 호스트 또는 AS/400 데이터베이스 서버에 액세스한 경우, 2단계 확약중 이상 실패 트랜잭션을 복구하는 방법에 약간의 차이가 있습니다.

호스트 또는 AS/400 데이터베이스 서버에 액세스하기 위해 DB2 Connect가 사용됩니다. DB2 Connect에 DB2 동기 지점 관리 프로그램이 구성된 경우 복구 단계는 다릅니다.

DB2 Connect에 DB2 동기 지점 관리 프로그램이 구성된 경우의 복구

일반적으로 호스트 또는 AS/400 서버에서의 2단계 확약중 이상 실패 트랜잭션 복구는 트랜잭션 관리자(TM) 및 DB2 동기 지점 관리 프로그램(SPM)에 의해 자동으로 수행됩니다. 호스트 또는 AS/400 서버에서의 2단계 확약중 이상 실패 트랜잭션은 지역 DB2 위치에서는 자원을 보유하지 않고, 트랜잭션이 해당 위치에서 2단계 확약중 이상 실패 상태로 있는 한 호스트 또는 AS/400 서버에서 자원을 보유합니다. 호스트 또는 AS/400 관리자가 경험적 방법으로 결정을 내려야 한다고 판단한 경우, 관리자는 호스트 또는 AS/400 서버에서 트랜잭션을 확약할 것인지 구간 복원할 것인지를 판별하기 위해 현지 DB2 데이터베이스 관리자(전화 등을 통

해)에게 문의할 것입니다. 이 경우, LIST DRDA INDOUBT TRANSACTIONS 명령을 사용하여 DB2 Connect 인스턴스에서의 트랜잭션 상태를 판별할 수 있습니다. SNA 통신 환경과 관련된 대부분의 상황에서 다음 단계를 지시사항으로 삼을 수 있습니다.

1. 아래와 같이 SPM에 연결하십시오.

```
db2 => connect to db2spm
```

데이터베이스 연결 정보

```
데이터베이스 제품      = SPM0500
SQL 권한 부여 ID      = CRUS
지역 데이터베이스 별명 = DB2SPM
```

2. LIST DRDA INDOUBT TRANSACTIONS 명령을 사용하여 SPM으로 알려진 2단계 협약중 이상 실패 트랜잭션을 표시하십시오. 다음의 예에서는 SPM으로 알려진 하나의 2단계 협약중 이상 실패 트랜잭션을 보여줍니다. db_name은 호스트 또는 AS/400 서버의 지역 별명입니다. partner_lu는 호스트 또는 AS/400 서버의 완전한 LU 이름입니다. 이 별명은 호스트 또는 AS/400 서버의 최상의 식별자로서, 호스트 또는 AS/400 서버에서 호출자가 제공합니다. luwid는 트랜잭션에 고유한 식별자를 제공하며, 모든 호스트 및 AS/400 서버에서 사용 가능합니다. 해당 트랜잭션이 표시되면, uow_status 필드를 사용하여 값이 C(협약)인지 R(구간 복원)인지 여부에 따라 트랜잭션 결과를 판별할 수 있습니다. LIST DRDA INDOUBT TRANSACTIONS 명령을 WITH PROMPTING 매개변수와 함께 실행하면, 해당 트랜잭션을 대화식으로 협약, 구간 복원 또는 무시할 수 있습니다. *Command Reference*에서 자세한 내용을 참조하십시오.

```
db2 => list drda indoubt transactions
DRDA Indoubt Transactions:
1.db_name: DBAS3   db_alias: DBAS3   role: AR
   uow_status: C partner_status: I partner_lu: USIBMSY.SY12DQA
corr_fok: USIBMST.STB3327L
luwid: USIBMST.STB3327.305DFDA5DC00.0001
xid: 53514C2000000017 00000000544D4442 0000000000305DFD A63055E962000000
00035F
```

3. partner_lu 및 luwid에 대한 2단계 협약중 이상 실패 트랜잭션이 표시되지 않거나 LIST DRDA INDOUBT TRANSACTIONS 명령이 다음과 같이 리턴될 경우,

```
db2 => list drda indoubt transactions
SQL1251W No data returned for heuristic query.
```

트랜잭션이 구간 복원된 것입니다.

흔하지는 않으나 또 하나의 상황이 발생할 수 있습니다. partner_lu에 대해 적절한 luwid를 가진 2단계 확약중 이상 실패 트랜잭션이 표시되었지만 uow_status가 "I"인 경우, SPM은 트랜잭션이 확약될 것인지 구간 복원될 것인지 여부를 알지 못합니다. 이 경우, DB2 Connect 워크스테이션에서 트랜잭션을 확약하거나 구간 복원하려면 WITH PROMPTING 매개변수를 사용해야 합니다. 그런 다음, DB2 Connect가 경험적 결정에 의거하여 호스트 또는 AS/400 서버와 재동기화되도록 하십시오.

DB2 Connect가 DB2 동기 지점 관리 프로그램을 사용하지 않는 경우의 복구

TCP/IP 연결을 사용하여 OS/390용 DB2를 DB2 Connect Personal Edition 또는 DB2 Connect Enterprise Edition으로부터 다중 사이트 갱신하며 DB2 동기 지점 관리 프로그램이 사용되지 않는 경우 이 절의 정보를 참조하십시오. 이 경우의 2단계 확약중 이상 실패 트랜잭션 복구는 DB2 동기 지점 관리 프로그램과 관련된 2단계 확약중 이상 실패 트랜잭션 복구와는 다릅니다. 이러한 환경에서 2단계 확약중 이상 실패 트랜잭션이 발생하는 경우, 누가 문제점을 탐지하느냐에 따라 클라이언트, 데이터베이스 서버, 트랜잭션 관리자(TM) 데이터베이스에서 경고 항목이 생성됩니다. 경고 항목은 db2alert.log 파일에 위치합니다. 경고에 관한 자세한 정보는 문제점 해결 안내서 매뉴얼을 참조하십시오.

TM, 해당 데이터베이스 및 연결을 모두 다시 사용할 수 있게 되면 곧바로 모든 2단계 확약중 이상 실패 트랜잭션의 재동기화가 자동으로 이루어집니다. 데이터베이스 서버에서 경험적 방식으로 결정이 이루어지도록 강제하기 보다는 자동 재동기화가 이루어지도록 해야 합니다. 그러나, 경험적 방식을 사용해야 한다면 다음 단계를 지시사항으로 사용하십시오.

주: DB2 동기 지점 관리 프로그램이 관련되기 때문에 LIST DRDA INDOUBT TRANSACTIONS 명령을 사용할 수 없습니다.

1. OS/390 호스트에서, DISPLAY THREAD TYPE(INDOUBT) 명령을 발행하십시오.

경험적 방식으로 완료하려는 트랜잭션을 이 목록에서 확인합니다. DISPLAY 명령의 세부사항은 *DB2 for OS/390 Command Reference*를 참조하십시오. 표시되는 LUWID는 트랜잭션 관리자 데이터베이스의 동일한 luwid에 대응될 수 있습니다.

2. 수행하려는 작업에 따라 RECOVER THREAD(<LUWID>) ACTION(ABORT|COMMIT) 명령을 발행하십시오.

RECOVER 명령의 세부사항은 *DB2 for OS/390 Command Reference*를 참조하십시오.

제6부 부록 및 끝머리

부록A. 이름 지정 규칙

다음 데이터베이스와 데이터베이스 오브젝트의 이름을 제공할 때 아래의 이름 지정 규칙을 사용하십시오.

- 데이터베이스 이름
- 데이터베이스 및 데이터베이스 별명
- 사용자 ID 및 암호
- 스키마 이름
- 그룹 및 사용자 이름
- 오브젝트 이름.

IBM SQL 또는 ISO/ANSI SQL92의 예약어를 테이블, 뷰, 컬럼, 색인 및 권한 부여 ID 이름으로 사용하지 마십시오. 적당한 이름 목록은 *SQL 참조서*에 포함되어 있습니다.

권한 부여 ID(사용자 이름 및 그룹 이름 포함)와 워크스테이션에 대한 이름 지정 규칙 및 추가 플랫폼 제한사항에 대해서는 빠른 시작 매뉴얼을 참조하십시오.

데이터베이스 이름

새로운 데이터베이스가 작성될 때마다, 데이터베이스 관리 프로그램은 데이터베이스에 대한 파일과 데이터 파일을 저장하기 위한 별도의 디렉토리를 작성합니다.

이러한 디렉토리명 체계는 SQL00001부터 SQLnnnnn까지로서, 여기서 SQL00001은 최초로 생성된 데이터베이스와 관련이 있는 제어 파일을 가지고 있으며, SQL00002는 두 번째로 생성된 데이터베이스에 대한 제어 파일을 가지고 있는 디렉토리로, 계속 이런 방식으로 이름 지정됩니다.

이러한 디렉토리는 자동으로 유지보수됩니다. 디렉토리를 이름 지정하는데 잠재적인 문제점을 피하려면, 데이터베이스 관리 프로그램이 사용한 것과 같은 이름 지

정 스키마를 사용하여 자체의 디렉토리를 작성하지 말고, 이미 데이터베이스 관리 프로그램에 의해 작성된 디렉토리를 조작하지 마십시오.

데이터베이스 및 데이터베이스 별명

데이터베이스명은 사용자가 CREATE DATABASE 명령 또는 API의 일부로 제공하는 이름을 식별합니다. 이러한 이름은 이들이 카탈로그화되어 있는 위치 안에서 고유해야 합니다. 예를 들어, DB2의 UNIX 기반 실행에서 이 위치는 디렉토리 경로인데 반해 OS/2의 실행의 경우에는 드라이브 이름이 됩니다.

데이터베이스 별명은 지역 또는 원격 데이터베이스에 주어진 지역 동의어입니다. 이러한 이름은 시스템 데이터베이스 디렉토리 내에서 고유해야 하고, 그 안에 데이터베이스 관리 프로그램의 각각의 인스턴스에 대해 모든 별명이 저장되어 있습니다. 새로운 데이터베이스가 생성될 때, 데이터베이스명에 대해 기본값으로 별명이 따르게 됩니다. 결과적으로, 해당 이름을 가진 데이터베이스가 없다고 하더라도, 데이터베이스 별명으로 존재하는 이름을 사용하여 데이터베이스를 생성할 수 없습니다.

데이터베이스 또는 데이터베이스 별명을 만들 때, 이름은 다음과 같은 규칙을 따라야 합니다.

- 1개에서 8개까지의 문자로 이루어져야 합니다.
- 다음 중 하나로 시작해야 합니다.
 - A에서 Z(소문자는 대문자로 변환합니다)
 - @, # 또는 \$
- 다음을 포함할 수 있습니다.
 - A에서 Z(소문자는 대문자로 변환합니다)
 - 0에서 9까지
 - @, #, \$ 및 _(밑줄)

주: 데이터베이스를 통신 환경에서 사용하려고 하는 경우, 잠재적인 문제점을 피하기 위해서는 데이터베이스 이름에 특수 문자 @, # 및 \$를 사용하지 마십시오. 또한, 이러한 문자가 모든 키보드에서 사용 가능한 것이 아닐 수 있으므로, 다른 국가에서 해당 데이터베이스를 사용할 계획이 있다면, 이러한 문자를

사용하는 것은 피해 주십시오. 마지막으로, Windows NT 시스템에서는 어떠한 인스턴스명도 서비스명과 같지 않음을 확인하여 주십시오.

사용자 ID 및 암호

사용자 ID 또는 암호를 작성할 때 작성하는 이름은 다음과 같습니다.

- 다음 중 어느 것도 될 수 없습니다.
 - SQL 참조서에 나열된 USERS, ADMINS, GUESTS, PUBLIC, LOCAL 또는 모든 SQL 예약어
- 다음 문자로 시작할 수 없습니다.
 - SQL, SYS 또는 IBM
- 다음을 포함할 수 있습니다.
 - A에서 Z까지

주: 사용자 ID와 암호를 대소문자 구별하는 운영 체제도 있습니다. 이러한 경우가 있는지 운영 체제 문서를 점검해야 합니다.

- 0에서 9까지
- @, # 또는 \$
- 사용자 ID는 30자를 초과할 수 없습니다.

주: 암호 유지보수 작업을 수행할 때 필요할 수 있습니다. 그러한 작업은 서버에서 요구되고, 많은 사용자가 서버 환경에서 작업할 수 없거나 작업하기에 용이하지 않으므로, 작업을 진행한다는 것은 어려운 일입니다. DB2 UDB는 서버에서 요구하지 않아도 암호를 갱신하고 확인할 수 있는 방법을 제공합니다. 예를 들어, OS/390용 DB2 버전 5는 이러한 사용자 암호 변경 방법을 지원합니다. 오류 메시지 SQL1404N 『암호가 만기되었습니다』가 수신되면, 다음과 같이 CONNECT문을 사용하여 암호를 변경하십시오.

```
CONNECT TO <database> USER <userid> USING <password>  
NEW <new_password> VERIFY <new_password>
```

DB2 클라이언트 구성 지원 프로그램(CCA)의 『암호 변경』 대화 상자를 사용하여 암호를 변경할 수도 있습니다. 암호를 변경하는 이러한 방법에 대한 자세한 내용은 SQL 참조서 및 CCA 온라인 도움말을 참조하십시오.

스키마 이름

다음 스키마 이름은 예약어이므로 사용될 수 없습니다.

- SYSCAT
- SYSFUN
- SYSIBM
- SYSSTAT

일반적으로 SYS로 시작되는 스키마 이름은 피함으로써, 앞으로 나타날 수도 있는 잠재적인 이주시의 문제점을 미리 방지해야 합니다. 데이터베이스 관리 프로그램은 SYS로 시작하는 스키마 이름을 사용하여 트리거, 사용자 정의 유형 또는 사용자 정의 함수를 작성하지 못하게 합니다.

또한 스키마 이름으로서 SESSION을 사용하지 않도록 권장합니다. 선언된 임시 테이블은 SESSION으로 규정되어야 합니다. 그러므로 응용프로그램 논리가 지나치게 복잡할 수 있는 경우에 영구 테이블의 이름과 동일한 임시 테이블을 응용프로그램에서 선언하는 것이 가능합니다. 선언된 임시 테이블을 처리할 때를 제외하고는 스키마 SESSION 사용을 피하십시오.

그룹 및 사용자 이름

UNIX 기반 시스템에서는 그룹과 사용자가 같은 이름을 가질 수 있습니다. GRANT 문에 있어서는, 그룹을 참조한 것인지 또는 사용자를 참조한 것인지를 여부를 지정해야 합니다. REVOKE문에 대해서는, 사용자 또는 그룹을 지정하는 것은 서로 다른 GRANTEETYPE 값을 갖는 GRANTEE에 대해 권한 카탈로그 테이블에 다중 행이 있는지 여부에 따라 달라집니다.

OS/2에서는 그룹과 사용자가 같은 이름을 가질 수 없습니다.

Windows NT에서는, 지역 그룹 이름, 전역 그룹 이름 및 사용자 ID가 같은 이름을 갖지 못합니다.

그룹 이름은 8자를 초과할 수 없습니다.

오브젝트 이름

데이터베이스 오브젝트에는 다음과 같은 것이 있습니다.

- 스키마
- 테이블
- 뷰
- 컬럼
- 색인
- 사용자 정의 함수(UDF)
- 사용자 정의 유형(UDT)
- 트리거
- 별명
- 테이블 공간
- 저장 프로시저어
- 메소드
- 노드 그룹
- 버퍼 풀
- 이벤트 모니터

데이터베이스 오브젝트를 이름 지정할 때, 이름은 다음과 같이 지정하십시오.

- 1개에서 18개까지의 문자(바이트)로 이루어져야 합니다.

주: 예외사항은 다음과 같습니다.

- 스키마 및 컬럼은 1에서 30자까지 허용됩니다.
- 테이블, 뷰, 상관 이름 및 별명은 1에서 128자까지 허용됩니다.
- 다음 중 하나로 시작해야 합니다.
 - A에서 Z(소문자는 대문자로 변환합니다.)
 - 유효한 강조 문자(예: ö)
 - (다중 바이트 환경에 대한) 다중 바이트 공간을 제외한 다중 바이트 문자
- 다음을 포함할 수 있습니다.

- A에서 Z(소문자는 대문자로 변환합니다.)
- 유효한 강조 문자(예: ö)
- 0에서 9까지
- @, #, \$ 및 _(밑줄)
- 다중 바이트 공간을 제외한 다중 바이트 문자(다중 바이트 환경의 경우)

키워드를 사용할 수 있습니다. 만일 SQL 키워드로도 해석될 수 있는 구문에서 키워드가 사용되면, 이것은 분리 식별자로 지정되어야 합니다. 분리 식별자에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

호환성을 극대화하기 위해서는 IBM SQL 및 ISO/ANSI SQL92 예약어를 사용하십시오. 예약어 목록에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

주:

1. 분리 식별자를 사용함으로써, 이러한 이름 지정 규칙에 어긋나는 오브젝트를 생성하는 것이 가능해집니다. 그러나, 이후의 사용이 문제점을 발생시킬 수도 있습니다. 데이터베이스의 사용 및 조작에 있어서 잠재적인 문제점을 피하려면, 위의 규칙을 반드시 지키십시오.

예를 들어, 이름에 + 또는 - 기호가 포함된 컬럼을 작성하면, 그 컬럼을 색인에 사용하여 테이블 재구성시 문제가 발생합니다.

2. 오브젝트 이름과 관련된 자국어 지원(NLS)에 대해서는 *관리 안내서: 계획*에서 자세한 내용을 참조하십시오.

연합 데이터베이스 오브젝트 이름

연합 데이터베이스 오브젝트는 다음을 포함합니다.

- 색인 스펙
- 별명
- 서버
- 랩퍼
- 함수 맵핑
- 유형 맵핑

- 사용자 맵핑

연합 데이터베이스 오브젝트를 이름 지정할 때 한계가 적용됩니다. 오브젝트명의 전체 목록, 연관된 식별자 한계 및 요구사항은 *SQL* 참조서에 있습니다. 요약한다면, 오브젝트명은

- 제한됩니다. 별명, 맵핑, 색인 스펙, 서버 및 랩퍼 이름은 128바이트를 초과할 수 없습니다.
- 다음 중 하나로 시작해야 합니다.
 - A에서 Z까지(따옴표가 없는 이름은 대문자로 변환됩니다.)
 - 유효한 강조 문자(예: ö)
 - (다중 바이트 환경에 대한) 다중 바이트 공간을 제외한 다중 바이트 문자
- 내부 이름 지정 규칙을 따라야 합니다. 다음 문자로 시작할 수 없습니다.
 - A에서 Z까지
 - 유효한 강조 문자(예: ö)
 - 0에서 9까지
 - @, #, \$ 및 _(밑줄)
 - 다중 바이트 공간을 제외한 다중 바이트 문자(다중 바이트 환경의 경우)

키워드를 사용할 수 있습니다. 만일 *SQL* 키워드로도 해석될 수 있는 구문에서 키워드가 사용되면, 이것은 분리 식별자로 지정되어야 합니다. 분리 식별자에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

호환성을 극대화하기 위해서는 *IBM SQL* 및 *ISO/ANSI SQL92* 예약어를 사용하십시오. 예약어 목록에 대해서는 *SQL* 참조서에서 자세한 내용을 참조하십시오.

옵션(서버, 별명) 및 옵션 설정값은 255바이트로 제한됩니다.

연합 시스템에서 대소문자 구별 값이 보존되는 방법

분산 요청에서, 데이터 소스에 대소문자가 구별되는 식별자 및 암호를 지정할 필요가 있습니다. 데이터 소스로 전달될 때 대소문자가 올바른지 확인하려면, 이 지시사항을 따르십시오.

- 필요한 대소문자로 단어를 지정하고 큰따옴표로 묶으십시오.
- 사용자 ID를 지정할 경우, 데이터 소스에 대해 fold_id 서버 옵션을 "n"(『아니오, 대소문자를 변경하지 마십시오.』)으로 설정하십시오. 암호를 지정할 경우, 데이터 소스에 대해 fold_pw 서버 옵션을 "n"으로 설정하십시오.

사용자 ID와 암호에 대해 다른 방법을 사용할 수 있습니다. 데이터소스에서 사용자 ID가 소문자로 사용될 필요가 있는 경우, 이 사용자 ID를 어떠한 문자로든지 지정하고 fold_pw 서버 옵션을 "1"(『이 ID를 소문자로 데이터 소스에 보냅니다.』)로 설정할 수 있습니다. 데이터 소스에서 ID가 대문자로 사용될 필요가 있는 경우, 이 ID를 어떠한 문자로든지 지정하고 fold_id를 "1"(『이 ID를 대문자로 데이터 소스에 보냅니다.』)로 설정할 수 있습니다. 같은 방법으로, 데이터 소스에서 암호가 소문자 또는 대문자로 사용될 필요가 있다면, fold_id 서버 옵션을 "1" 또는 "u"로 설정하여 이 요구사항을 충족시킬 수 있습니다.

서버 옵션에 대한 자세한 내용은 177 페이지의 『서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화』에서 "서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화"를 참조하십시오.

- 운영 체제의 명령 프롬프트에서 대소문자를 구별하는 식별자 또는 암호를 큰따옴표로 묶어 표시할 경우, 시스템이 큰따옴표를 올바르게 분석하는지 확인해야 합니다. 이를 수행하려면, 다음을 수행하십시오.

- UNIX 기반 운영 체제에서, 명령문을 작은따옴표로 묶어 표시하십시오.
- Windows NT 운영 체제에서, 각 따옴표 앞에 백슬래시(\)를 표시하십시오.

예를 들어, DB2 계열 데이터 소스에 있는 여러 구분 식별자들은 대소문자를 구별합니다. NORBASE라고 하는 데이터 소스에 상주하는 CS 뷰, "my_schema"."wkly_sal"의 DB2에 대해 NICK1 별명을 작성 한다고 가정하십시오.

Unix 기반 시스템의 명령 프롬프트에서, 다음을 입력합니다.

```
db2 'create nickname nick1 for norbase."my_schema"."wkly_sal"'
```

Windows NT 명령 프롬프트에서, 다음과 같이 입력해야 합니다.

```
db2 create nickname nick1 for norbase.\my_schema\.\wkly_sal\
```

DB2 대화식 모드 명령 프롬프트에서 명령문을 입력하거나 응용프로그램에서 SQL을 지정할 경우, 작은따옴표 또는 슬래시를 사용할 필요가 없습니다. 예를 들어, UNIX 기반 시스템 또는 Windows NT 시스템의 DB2 명령 프롬프트에서는 다음과 같이 입력하십시오.

```
create nickname nick1 for norbase."my_schema"."wkly_sal"
```

부록B. DCE 디렉토리 서비스 사용

DCE는 셀 디렉토리 서비스(CDS) 및 전역 디렉토리 서비스(GDS)를 제공합니다. DCE 개념 및 서비스에 관한 자세한 정보는 *Introduction to OSF DCE* 매뉴얼을 참조하십시오. DCE 디렉토리 서비스에 대한 DB2 기능은 CDS만을 지원합니다. 이러한 지원이 있기 때문에, 사용자는 모든 클라이언트에 각 데이터베이스, 노드 및 DCS 데이터베이스를 작성할 필요가 없습니다. 이 모든 정보는 DCE CDS에 집중됩니다.

다음 절에서는 DCE 디렉토리 서비스를 사용하여 데이터베이스를 설정하고 액세스하는 방법에 대해 설명합니다.

- 디렉토리 오브젝트 작성
- 각 오브젝트 클래스의 속성
- 디렉토리 서비스 보안
- 구성 매개변수 및 레지스트리 변수
- CATALOG 및 ATTACH 명령 및 CONNECT문
- 클라이언트를 데이터베이스에 연결하는 방법
- 디렉토리 검색 방법
- DCE 디렉토리 정보를 임시로 대체
- 디렉토리 서비스 타스크
- 디렉토리 서비스의 제한사항

DCE 디렉토리 서비스는 모든 DB2 클라이언트에 의해 지원되지 않을 수도 있습니다. 그러나 DCE 디렉토리 서비스가 DB2 클라이언트를 위해 지원된다면, 빠른 시작 매뉴얼에서 추가 내용을 제공합니다.

디렉토리 오브젝트 작성

데이터베이스 관리자가 작성해야 하는 디렉토리 오브젝트에는 다음의 세 가지 유형이 있습니다.

- 『데이터베이스 오브젝트』
- 456 페이지의 『데이터베이스 위치 지정자 오브젝트』
- 457 페이지의 『경로지정 정보 오브젝트』

각각의 오브젝트는 속성을 가지고 있습니다. 속성의 완전한 설명은 459 페이지의 『각 오브젝트 클래스의 속성』에서 자세한 내용을 참조하십시오.

데이터베이스 관리자가 오브젝트를 생성하기 전에 DCE 관리자는 데이터베이스 정보를 CDS 테이블에 추가하고, 데이터베이스 관리자에게 생성 권한을 권한 부여해야 합니다. 478 페이지의 『DCE 관리자 타스크』에서 자세한 내용을 참조하십시오.

데이터베이스 오브젝트

각각의 목표 데이터베이스에 대해 하나의 데이터베이스 오브젝트가 요구됩니다. 오브젝트는 디렉토리명 및 데이터베이스 이름에 연결되는 셀 이름을 포함하는 이름을 가집니다. 예를 들면, 다음과 같습니다.

```
./.../cell_name/dir_name1/dir_name2/OBJ_NAME
```

주: 다음은 데이터베이스명에 대한 바람직한 사항입니다. 이름은 최대 8자까지 가능하며 모든 문자는 대문자여야 합니다. 만일 대문자와 소문자가 섞여 있거나 8자를 넘어가는 경우, 별명을 할당하기 위해 CATALOG GLOBAL DATABASE 명령을 사용할 필요가 있습니다. 명령에 대해서는 469 페이지의 『CATALOG GLOBAL DATABASE 명령』에서 자세한 내용을 참조하십시오.

다음은 데이터베이스 오브젝트의 예입니다. DCE 디렉토리에 저장된 오브젝트는 시간소인과 같은 기타 정보를 갖습니다. 각 속성의 왼쪽의 문자는 속성이 반드시 필요한 것인지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

```
오브젝트 이름:                /.../CELL_TORONTO/subsys/database/AIXDB1
R DB_Object_Type:              D
C DB_Product_Name:             DB2_for_AIX
```

```

C DB_Product_Release:      V5R1M000
R DB_Native_Database_Name: AIXDBASE
R DB_Database_Protocol:   DB2RA
R DB_Authentication:      CLIENT
O DB_Communication_Protocol:
O DB_Database_Locator_Name: /.../CELL_TORONTO/subsys/database/AIX_INST
C DB_Comment:              Test_database_on_AIX

```

데이터베이스가 데이터베이스 관리 프로그램 인스턴스와 관련된 다수의 데이터베이스 중 하나라면, 데이터베이스 오브젝트는 데이터베이스 위치 지정자 오브젝트 이름을 갖고 있어야 하고, 통신 프로토콜은 공백이어야 합니다. 데이터베이스 위치 지정자 오브젝트 이름은 데이터베이스 관리 프로그램 또는 DB2 연결 인스턴스에 완전 규정화된 값입니다.

오브젝트를 생성하는 데 사용하는 DCE 명령의 예가 있습니다. 어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 478 페이지의 『DCE 관리자 타스크』에 기술된 단계를 따라야 합니다.

우선 *cdscp.inp*라는 파일에 다음을 입력해야 합니다.

```

create object /./subsys/database/AIXDB1

add object /./subsys/database/AIXDB1 DB_Object_Type           = D
add object /./subsys/database/AIXDB1 DB_Product_Name         = DB2_for_AIX
add object /./subsys/database/AIXDB1 DB_Product_Release      = V5R1M000
add object /./subsys/database/AIXDB1 DB_Native_Database_Name = AIXDBASE
add object /./subsys/database/AIXDB1 DB_Database_Protocol    = DB2RA
add object /./subsys/database/AIXDB1 DB_Authentication       = CLIENT
add object /./subsys/database/AIXDB1 DB_Database_Locator_Name = /...
/CELL_TORONTO/subsys/database/AIX_INST
add object /./subsys/database/AIXDB1 DB_Comment              = Test_database_on_AIX

```

다음 중 하나를 실행해야 합니다.

- *dcelogin* 핵심부 암호(OS/2) 또는
- *dce_login* 핵심부 암호(UNIX, Windows NT 또는 Windows 95에서)

이는 다음과 같아야 합니다.

- *cdscp < cdscp.inp*

오브젝트를 표시하려면, 다음 명령을 사용하십시오.

```
cdscp show object /./subsys/database/AIXDB1
```

데이터베이스가 데이터베이스 관리 프로그램 인스턴스와 관련된 유일한 데이터베이스라면, 데이터베이스 오브젝트는 통신 프로토콜 속성 값을 가져야 하고, 데이터베이스 위치 지정자 오브젝트는 공백이어야 합니다. 예를 들면, 다음과 같습니다.

```

오브젝트 이름:                /.../CELL_TORONTO/subsys/database/MVSDB
R DB_Object type:              D
C DB_Product_Name:            DB2_for_MVS
C DB_Product_Release:         V5R1M00
R DB_Native_Database_Name:    MVSDBASE
R DB_Database_Protocol:       DRDA
R DB_Authentication:          SERVER
O DB_Communication_Protocol:  APPC;NET1;TARGETLU1;DB2DRDA;MODE1;PROGRAM
O DB_Database_Locator_Name:
C DB_Comment:                  Test_database_on_MVS

```

데이터베이스 위치 지정자 오브젝트

이 오브젝트에는 DDMS 인스턴스 또는 DB2 연결 인스턴스로 사용한 통신 프로토콜에 대한 세부사항이 들어 있습니다. 다음에 대해 데이터베이스 위치 지정자 오브젝트가 하나씩 필요합니다.

- DBMS 및 DB2 연결의 각 인스턴스
- 둘 이상의 데이터베이스와 관련이 있지만 DB2와는 관련이 없는 각 DBMS 인스턴스
- DBMS와 관련되지 않은 각 DB2 인스턴스

오브젝트는 다음 예에서처럼, 디렉토리 이름과 데이터베이스 인스턴스 이름의 일부에 연결되는 셀 이름을 포함하는 이름을 가집니다.

```
/.../cell_name/dir_name1/dir_name2/AIX_INST
```

주: 만일 인스턴스가 ATTACH의 대상으로 사용된다면, 일부의 이름은 최대 8자까지만 가능하며 모두 대문자여야 합니다.

다음은 데이터베이스 위치 지정자 오브젝트의 예입니다. DCE 디렉토리에 저장된 오브젝트는 시간소인과 같은 기타 정보를 갖습니다. 각 속성의 왼쪽의 문자는 속성이 반드시 필요한 것인지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

```

오브젝트 이름:                /.../CELL_TORONTO/subsys/database/AIX_INST
R DB_Object_Type:              L
C DB_Product_Name:            DB2_for_AIX
C DB_Product_Release:         V5R1M00
R DB_Communication_Protocol:  TCP/IP;HOSTNAME1;1234
R DB_Communication_Protocol:  APPC;NET1;TARGETLU1;TPN1;MODE;PROGRAM
C DB_Comment:                  Test_instance_on_AIX

```

데이터베이스 오브젝트와 데이터베이스 위치 지정자 오브젝트 둘다에 속성을 정의할 때 데이터베이스 오브젝트의 값이 사용됩니다.

오브젝트를 생성하는 데 사용하는 DCE 명령의 예가 있습니다. 어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 478 페이지의 『DCE 관리자 타스크』에 기술된 단계를 따라야 합니다.

우선 *cdscp.inp*라는 파일에 다음을 입력해야 합니다.

```
create object ../subsys/database/AIX_INST

add object ../subsys/database/AIX_INST DB_Object_Type           = L
add object ../subsys/database/AIX_INST DB_Product_Name         = DB2_for_AIX
add object ../subsys/database/AIX_INST DB_Product_Release      = V5R1M00
add object ../subsys/database/AIX_INST DB_Communication_Protocol = TCPIP;
HOSTNAME1;1234
add object ../subsys/database/AIX_INST DB_Communication_Protocol = APPC;NET1;
TARGETLU;TPN1;MODE;PROGRAM
add object ../subsys/database/AIX_INST DB_Comment              = Test_instance_on_AIX
```

다음 중 하나를 실행해야 합니다.

- *dcelogin* 핵심부 암호(OS/2) 또는
- *dce_login* 핵심부 암호(UNIX, Windows NT 또는 Windows 95에서)

이는 다음과 같아야 합니다.

- *cdscp < cdscp.inp*

오브젝트를 표시하려면, 다음 명령을 사용하십시오.

```
cdscp show object ../subsys/database/AIX_INST
```

경로지정 정보 오브젝트

호스트 액세스를 위해서는 경로지정 정보 오브젝트가 필요합니다. 클라이언트에 의해 사용되는 데이터베이스 프로토콜 및 목표 데이터베이스에 의해 사용되는 데이터베이스 프로토콜이 서로 일치하지 않는 경우, 경로지정 오브젝트가 클라이언트에 어떤 DDCS 인스턴스를 사용해야 할 지를 알려줍니다. 각각의 목표 데이터베이스에 대해 속성이 존재하는데, 여기에는 사용 가능한 데이터베이스 프로토콜 및 DB2 연결 인스턴스에 대한 데이터베이스 위치 지정자 오브젝트 이름이 포함됩니다. 오브젝트는 다음과 같이 디렉토리 이름과 고유한 일부 이름으로 연결된 셀 이름을 포함하는 이름을 가집니다.

```
./.../cell_name/dir_name1/dir_name2/ROUTE1
```

다음은 경로지정 정보 오브젝트의 예입니다. DCE 디렉토리에 저장된 오브젝트는 시간소인과 같은 기타 정보를 갖습니다. 각 속성의 왼쪽 문자는 속성 또는 속성 내 각 토큰이 반드시 필요한지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

클라이언트 그룹 1은 471 페이지의 그림22의 Client_1, Client_2 및 Client_3입니다.

```

오브젝트 이름:      /.../CELL_TORONTO/subsys/database/ROUTE1
R DB_Object_Type: R
C DB_Comment:      Routing_for_client_group_1

R DB_Target_Database_Info
R Database name      = /.../CELL_TORONTO/subsys/database/MVSDB
R Outbound protocol from router = DRDA
R Inbound protocol to router   = DB2RA
R Authenticate at gateway      = 1
O Parameter string            = NOMAP,D,INTERRUPT_ENABLED
R DB_Database_Locator_Name    = /.../CELL_TORONTO/subsys/database/GW_INST

R DB_Target_Database_Info
R Database name      = *OTHERDBS
R Outbound protocol from router = DRDA
R Inbound protocol to router   = DB2RA
R Authenticate at gateway      = 0
O Parameter string            =
R DB_Database_Locator_Name    = /.../CELL_TORONTO/subsys/database/OTH_INST

```

데이터베이스 이름 *OTHERDBS는 경로지정 정보 오브젝트에 명백하게 정의되어 있지 않은 목표 데이터베이스에 액세스하는 데 사용된 공통의 라우터를 식별하는 특수 값입니다.

오브젝트를 생성하는 데 사용하는 DCE 명령의 예가 있습니다. 백슬래시(\) 문자는 계속됨을 나타내는 문자입니다.

어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 478 페이지의 『DCE 관리자 타스크』에 기술된 단계를 따라야 합니다.

우선 *cdscp.inp*라는 파일에 다음을 입력해야 합니다.

```

create object /./subsys/database/ROUTE1

add object /./subsys/database/ROUTE1 DB_Object_Type = R
add object /./subsys/database/ROUTE1 DB_Comment = Routing_for_client_group_1
add object /./subsys/database/ROUTE1 DB_Target_Database_Info = \
    /.../CELL_TORONTO/subsys/database/MVSDB;\
drda;db2ra;1;NOMAP,D,INTERRUPT_ENABLE;\
    /.../CELL_TORONTO/subsys/database/GW_INST
add object /./subsys/database/ROUTE1 DB_Target_Database_Info = \
    *OTHERDBS;drda;db2ra;0;;\
    /.../CELL_TORONTO/subsys/database/OTH_INST

```

다음 중 하나를 실행해야 합니다.

- dcelogin 핵심부 암호(OS/2) 또는
- dce_login 핵심부 암호(UNIX, Windows NT 또는 Windows 95에서)

이는 다음과 같아야 합니다.

- cdscp < cdscp.inp

오브젝트를 표시하려면, 다음 명령을 사용하십시오.

```
cdscp show object /./subsys/database/ROUTE1
```

DCE 명령에 관한 자세한 정보는 다음 DCE 문서를 참조하십시오.

- *DCE Administration Guide*
- *DCE Administration Reference*

각 오브젝트 클래스의 속성

DCE 환경에서 각각의 오브젝트와 오브젝트 속성은 오브젝트 ID(OID)로 식별됩니다. 각각의 OID는 계층적인 권한 할당 구조로부터 얻을 수 있는데, 가장 높은 권한은 국제 표준화 기구(ISO)입니다.

표15에서는 각 오브젝트 클래스의 속성을 보여주고, 표16에서는 해당 속성을 보여줍니다.

표 15. 오브젝트 속성 클래스

오브젝트 클래스	오브젝트 ID(OID)	필수 속성	선택적 속성
(DB) Database_Object	1.3.18.0.2.6.12	DAU, DOT, DDP, DNN	DCO, DPN, DRL, DLN, DCP, DPR
(DL) Database_Locator_Object	1.3.18.0.2.6.13	DOT, DCP	DCO, DPN, DRL
(RI) Routing_Information_Object	1.3.18.0.2.6.14	DOT, DTI	DCO, DPN, DRL

표 16. 오브젝트 클래스 속성

속성명	OID	최소 길이	최대 길이	구문
(DAU) DB_Authentication	1.3.18.0.2.4.39	1	1024	Char

표 16. 오브젝트 클래스 속성 (계속)

속성명	OID	최소 길이	최대 길이	구문
(DCO) DB_Comment	1.3.18.0.2.4.30	1	1024	Char
(DCP) DB_Communication_Protocol	1.3.18.0.2.4.31	1	1024	Char
(DDP) DB_Database_Protocol	1.3.18.0.2.4.32	1	1024	Char
(DLN) DB_Database_Locator_Name	1.3.18.0.2.4.33	1	1024	Char
(DNN) DB_Native_Database_Name	1.3.18.0.2.4.34	1	1024	Char
(DOT) DB_Object_Type	1.3.18.0.2.4.35	1	1	Char
(DPN) DB_Product_Name	1.3.18.0.2.4.36	1	1024	Char
(DRL) DB_Product_Release	1.3.18.0.2.4.37	1	1024	Char
(DTI) DB_Target_Database_Info	1.3.18.0.2.4.38	1	1024	Char
(DPR) DB_Principal	1.3.18.0.2.4.63	1	1024	Char

주: DCP, DDP 및 DTI에 대해서는 다중의 값이 허용됩니다. 다른 속성에 대해서는 한 개의 값만이 가능합니다.

각 속성에 관한 세부사항

다음 절에서는 각각의 속성을 설명합니다.

주: DCE 디렉토리 서비스는 항목이 DB2에 대해 유효한지 검사하지 않습니다. 입력 필수인 속성을 입력했는지와 올바른 값을 입력했는지를 확인하십시오.

DB_Authentication(DAU)

오브젝트가 요구하는 인증 방법입니다. 이 속성은 DB2 서버의 데이터베이스 오브젝트에 필수적인 속성입니다. 값은 CLIENT, SERVER 또는 DCE여야 합니다.

DB_Principal(DPR)

인증 방법이 『DCE』인 경우, 이 속성에 DCE 핵심부를 입력하십시오.

DB_Comment(DCO)

설명 기록용입니다.

DB_Communication_Protocol(DCP)

각각의 값이 지원되는 네트워크 프로토콜을 설명하는 토큰으로 구성된 다중 값의 속성. 네트워크 프로토콜의 예는 TCP/IP, APPC, IPX/SPX 및 NetBIOS가 있습니다(마지막 두 가지는 OS/2에만 적용됩니다). 각 토큰은 세미콜론으로 구분됩니다. 토큰 사이에 공백을 넣지 마십시오.

- TCP/IP에 대한 토큰은 다음과 같습니다.
 1. tcpip
 2. 목표 노드의 호스트 이름
 3. 수신되는 TCP/IP 접속 요청을 청취하기 위해 오브젝트가 사용하는 포트 번호
 4. (선택적) 보안은 NONE 또는 SOCKS입니다.

예를 들면, tcpip;HOSTNAME;1234입니다.

- APPC에 대한 토큰은 다음과 같습니다.
 1. appc
 2. 오브젝트가 속해 있는 목표의 네트워크 ID
 3. 목표가 발견될 수 있는 LU 이름
 4. LU 내의 오브젝트를 나타내는 트랜잭션 프로그램 이름 (TPN)(MVS/ESA용 DB2의 경우, TPN으로 DB2DRDA 사용)
 5. 모드 이름
 6. 목표에 의해 사용될 보안의 유형. 값은 다음과 같습니다.
 - NONE
 - PROGRAM
 - SAME

예를 들면, appc;NETID;TARGETLU;TPNAME;MODE;PROGRAM입니다.

주: APPC의 경우 클라이언트는 LU 이름으로 지역 제어점(CP)을 사용해야 합니다.

- (OS/2 및 지원되는 Windows 운영 체제만) IPX/SPX에 대한 토큰은 다음과 같습니다.
 1. ipxspx
 2. 파일 서버 이름
 3. 오브젝트 이름

예를 들면, ipxspx;SVR_NAME;OBJ_NAME입니다.

- (OS/2 및 지원되는 Windows 운영 체제만) NetBIOS에 대한 토큰은 다음과 같습니다.

1. netbios
2. 서버의 노드 이름

예를 들면, netbios;SVR_NNME입니다. 여기서, 클라이언트 어댑터 번호는 레지스트리 값 *db2clientadpt* 또는 데이터베이스 관리 프로그램 구성 매개변수 *dft_client_adpt*에서 찾을 수 있습니다.

- (Windows operating 시스템만 해당) NPIPE에 대한 토큰은 다음과 같습니다.

1. NPIPE
2. 서버의 컴퓨터 이름
3. 서버의 인스턴스 이름

예를 들면, npipe;computername;instance입니다.

DB_Database_Protocol(DDP)

데이터베이스 프로토콜 또는 목표 데이터베이스에 의해 지원되는 프로토콜. 들어갈 수 있는 값은 DB2RA 및 DRDA입니다. 다음은 두 개의 프로토콜을 추가하는 데 사용되는 *cdscp* 명령입니다.

```
add object /./subsys/database/AIXDB1 DB_Database_Protocol db2ra
add object /./subsys/database/AIXDB1 DB_Database_Protocol drda
```

DB_Database_Locator_Name(DLN)

데이터베이스 위치 지정자 오브젝트의 DCE 이름. 데이터베이스 오브젝트에서 이름은 DBMS 인스턴스를 위한 것입니다. 경로지정 정보 오브젝트에서 이름은 DB2 연결 인스턴스에 대한 것입니다.

예를 들면, /.../CELL_TORONTO/subsys/database/AIX_INST입니다.

DB_Native_Database_Name(DNN)

데이터베이스를 포함하는 인스턴스 내에서 데이터베이스가 알려져 있는 데이터베이스 이름 또는 별명. 이것은 DBMS 인스턴스상에서 지역 응용프로그램이 해당 데이터베이스에 접속하기 위해 사용할 이름입니다.

DB2 Universal Database용 데이터베이스에 대해 이름은 최대 8자까지 가능합니다. 다른 데이터베이스의 경우, 이름의 길이는 다를 수 있습니다. 예를 들어, MVS/ESA용 DB2의 데이터베이스는 18자까지도 가능합니다.

DB_Object_Type(DOT)

오브젝트의 유형. 이 속성은 모든 오브젝트에 필요하며 다음 중 하나가 될 수 있습니다.

- D** 데이터베이스 오브젝트
- L** 데이터베이스 위치 지정자 오브젝트
- R** 경로지정 정보 오브젝트

DB_Product_Name(DPN)

제품 식별. 설명 기록용입니다.

DB_Product_Release(DRL)

제품 릴리스 레벨. 설명 기록용입니다.

DB_Target_Database_Info(DTI)

각각의 값이 세미콜론으로 구분된 고정된 수의 토큰으로 구성되어 있는 다중 값 속성. 토큰 사이에 공백을 넣지 마십시오. 토큰은 다음 순서를 가져야 합니다.

1. 데이터베이스 이름. 경로지정 서비스가 제공되는 목표 데이터베이스 오브젝트의 DCE 명칭. *OTHERDBS는 경로지정 정보 오브젝트에 명시적으로 정의되어 있지 않은 모든 목표 데이터베이스에 대해 기본 게이트웨이를 지정하는 특수 값입니다.
2. 라우터로부터의 아웃바운드 프로토콜. 목표 데이터베이스가 사용하는 데이터베이스 프로토콜 또는 경로지정 DB2 연결 인스턴스가 해당 목표 데이터베이스와 통신하는 데 사용하는 데이터베이스 프로토콜. 예를 들면, DRDA가 있습니다.
3. 라우터로의 인바운드 프로토콜. 경로지정 DB2 연결 인스턴스 오브젝트가 승인하는 데이터베이스 프로토콜. 예를 들면, DB2RA가 있습니다.
4. 게이트웨이에서의 인증. 유효한 값은 0 또는 1입니다. 465 페이지의 표 17에서 자세한 내용을 참조하십시오.

5. DB2 연결 게이트웨이에 대한 고유 정보를 가지고 있는 매개변수 문자열. 이 문자열은 아래에 설명되어 있는 순서의 토큰을 가지고 있습니다. 토큰은 쉼표로 구분됩니다. 지정되지 않은 토큰에 대해서는 기본값이 사용됩니다.

- 맵 파일 이름. 기본 SQLCODE 맵핑을 겹쳐쓰는 SQLCODE 맵핑 파일의 완전히 검증된 이름. SQLCODE 맵핑을 멈추려면 NOMAP 이라고 지정하십시오.
- D. 특정 SQLCODE가 리턴되면 응용프로그램은 DRDA 서버로부터 연결이 끊어집니다. SQLCODE에 대해서는 *DB2 Connect* 사용자 안내서에서 자세한 내용을 참조하십시오.
- INTERRUPT_ENABLED. DRDA 서버에 접속되어 있는 동안 클라이언트가 인터럽트를 발행하면, DB2 연결은 연결을 해제하고 작업 단위(UOW)를 구간 복원합니다.

예를 들어, 다음과 같습니다.

```
NOMAP
/u/username/sqlllib/map/dcs1new.map,D
/u/username/sqlllib/map/dcs1new.map,D,INTERRUPT_ENABLED
```

다음과 같이 기본값이 사용되는 위치에서 토큰의 순서를 유지하려면, 쉼표를 사용하십시오. 예를 들면, 다음과 같습니다.

```
,D
```

또는

```
.,INTERRUPT_ENABLED
```

매개변수 문자열에 대해서는 *DB2 Connect* 사용자 안내서에서 자세한 내용을 참조하십시오.

6. 경로지정 서비스를 제공하는 DB2 연결 인스턴스의 DCE 이름

다음은 DB_Target_Database_Info의 예입니다.

```
./.../CELL_TORONTO/subsys/database/MVSDB;\
drda;db2ra;0;;\
./.../CELL_TORONTO/subsys/database/GW_INST
```

주: 위의 예에서 백슬래시(\)는 행 연결 문자입니다.

디렉토리 서비스 보안

DB2 Connect 게이트웨이가 없는 환경에서 DCE 디렉토리 서비스를 사용할 때, 인증은 데이터베이스 서버에 액세스하는 다른 클라이언트에 사용되는 것과 같은 것입니다. 253 페이지의 『서버에 대한 인증 방식 선택』에서 자세한 내용을 참조하십시오.

DB2 연결 게이트웨이가 없는 환경에서 DCE 디렉토리 서비스를 사용할 때, DB2 연결 관리자는 사용자 이름과 암호가 어디서 평가되는지 판별합니다. DCE 디렉토리에서 다음을 지정합니다.

- DB2 Connect 워크스테이션을 나타내는 데이터베이스 위치 지정자 오브젝트의 통신 프로토콜의 보안 유형(원격 클라이언트가 APPC를 통해 DB2 Connect Extended Edition 게이트웨이에 연결된 경우, 게이트웨이의 DCE 위치 지정자 오브젝트에서 NONE으로 보안 유형을 지정하십시오)
- 데이터베이스 오브젝트의 인증 유형
- 데이터베이스 오브젝트(또는 관련 위치 지정자 오브젝트)의 통신 프로토콜 보안 유형
- 경로지정 정보 오브젝트의 게이트웨이 토큰에서의 인증

표17에서는 이 값으로 가능한 조합을 보여주며, 여기서 APPS를 사용하여 각 조합에 대한 타당성 검사가 수행됩니다. 이 표에 표시되는 조합은 DCE 디렉토리 서비스를 갖는 DB2 연결에 의해 지원됩니다.

표 17. APPS 연결을 사용하여 DCE에서의 유효한 보안 시나리오

경우	서버의 데이터베이스 오브젝트		경로지정 오브젝트	검증
	인증	보안	게이트웨이에서의 인증	
1	클라이언트	같음	0	원격 클라이언트(또는 DB2 Connect 워크스테이션)
2	클라이언트	같음	1	DB2 Connect 워크스테이션
3	서버	프로그램	0	DRDA 서버
4	서버	프로그램	1	DB2 Connect 워크스테이션 및 DRDA 서버
5	DCE	NONE	적용할 수 없음	DCE

표18에서는 이 값으로 가능한 조합을 보여주며, 여기서 TCP/IP를 사용하여 각 조합에 대한 타당성 검사가 수행됩니다. 이 표에 표시되는 조합은 DCE 디렉토리 서비스를 갖는 DB2 연결에 의해 지원됩니다.

표 18. TCP/IP를 사용하여 DCE에서의 유효한 보안 시나리오

경우	인증	게이트웨이에서의 인증	검증
1	클라이언트	0	클라이언트
2	클라이언트	1	DB2 Connect 워크스테이션
3	서버	0	DRDA 서버
4	적용할 수 없음	적용할 수 없음	None
5	DCE	적용할 수 없음	DCE

각 조합은 APPC 및 TCP/IP에 모두 적용할 수 있으며, 자세한 정보는 다음과 같습니다.

1. 사용자 이름과 암호는 원격 클라이언트에서만 검증됩니다(지역 클라이언트의 경우, 사용자 이름과 암호는 DB2 Connect 워크스테이션에서만 검증됩니다).

사용자는 사인온한 위치에서 인증되는 것이 일반적입니다. 사용자 ID는 네트워크를 통해 전송되지만 암호는 그렇지 않습니다. 모든 클라이언트 워크스테이션이 적절한 보안 기능을 가지고 있는 경우에만 이러한 유형의 보안을 사용하십시오.

2. 사용자 이름과 암호는 DB2 Connect 워크스테이션에서만 검증됩니다. 암호는 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로 전송되지만 DRDA 서버로는 전송되지 않습니다.
3. 사용자명과 암호는 DRDA 서버에서만 검증됩니다. 암호는 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로, DB2 Connect 워크스테이션으로부터 DRDA 서버로 전송됩니다.
4. 사용자명과 암호는 DB2 Connect 워크스테이션과 DRDA 서버 모두에서 검증됩니다. 암호는 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로, DB2 Connect 워크스테이션으로부터 DRDA 서버로 전송됩니다.

검증이 두 곳에서 수행되기 때문에 사용자명과 암호의 같은 세트가 DB2 워크스테이션과 DRDA 서버 두 곳에서 유지보수되어야 합니다.

5. DCE 토큰을 DCE 보안 서버에서 얻습니다.

주:

1. AIX 시스템의 경우, 보안 유형 SAME을 사용하는 모든 사용자는 AIX 시스템 그룹에 속해야 합니다.
2. 원격 클라이언트가 있는 AIX 시스템의 경우, DB2 연결 워크스테이션에서 실행되는 DB2 연결 제품의 인스턴스는 AIX 시스템 그룹에 속해야 합니다.
3. DRDA 서버로의 액세스는 자체 보안 메커니즘 또는 보안 서브시스템에 의해 제어됩니다. 예를 들면, Virtual Telecommunications Access Method(VTAM) 및 Resource Access Control Facility(RACF)에 의해 제어됩니다. 보호되는 데이터베이스 오브젝트의 액세스는 SQL의 **GRANT** 및 **REVOKE**문에 의해 제어됩니다.

구성 매개변수 및 레지스트리 변수

다음의 구성 매개변수가 DCE 디렉토리와 함께 사용됩니다. 값의 예가 나와 있습니다. 세부사항은 관리 안내서: 성능에 있는 『DB2 구성』장의 『분산 서비스』를 참조하십시오.

- *dir_obj_name*은 *dir_path_name*과 병합된 데이터베이스 인스턴스 이름입니다. 인스턴스 이름이 ATTACH 명령의 목표로 사용되면, 이 이름은 다음 예에서처럼 8자보다 작거나 같아야 하며 모두 대문자여야 합니다.

AIX_INST

- *dir_type*은 DCE 디렉토리 서비스를 사용할지의 여부를 식별합니다. DCE 디렉토리 서비스를 사용 가능하게 하려면, 이 매개변수가 다음과 같이 설정되어야 합니다.

DCE

*dir_type*은 NONE으로 설정되어 있으며 DCE 디렉토리 서비스의 사용을 지원하지 않는 데이터베이스 클라이언트에서는 갱신될 수 없음을 주의하십시오.

- *dir_path_name*은 다음 예에서처럼 DCE 관리자가 제공하는 디렉토리 경로 이름입니다.

/./:/subsys/database/

- *route_obj_name*은 선택적 매개변수로서, 경로지정 정보 오브젝트의 DCE 디렉토리 서비스 이름을 제공합니다. 이 이름은 다음 예에서처럼 완전 규정화될 수 있습니다.

```
././subsys/database/ROUTE1
```

또는, 다음 예에서처럼 *dir_path_name*과 연결되는 한 부분의 이름이 될 수 있습니다.

```
ROUTE1
```

- *dft_client_comm*은 다음 예에서처럼 클라이언트가 사용하는 통신 프로토콜을 지정하는 선택적 DCE 매개변수입니다.

```
TCPIP
```

또한, 이 매개변수는 다음 예에서처럼 둘 이상의 프로토콜을 지정할 수 있습니다.

```
TCPIP,APPC (on UNIX-based platforms)
TCPIP,APPC,IPXSPX,NETBIOS (on OS/2 platforms)
TCPIP,APPC,IPXSPX,NETBIOS,NPIPE (on supported Windows operating systems)
```

- *dft_client_adpt*는 선택적인 DCE 매개변수입니다. 이 매개변수는 OS/2와 지원되는 Windows 운영 체제에서 NetBIOS 프로토콜에 대한 기본 클라이언트 어댑터 수를 지정합니다. 이 숫자의 유효값은 0-15까지입니다. 이 매개변수에 숫자가 아닌 값이 들어 있을 경우, 기본값은 0이 됩니다. 이 매개변수에 허용되는 범위를 벗어나는 값이 들어 있으면, 값은 기본값 0이 됩니다.

다음 매개변수의 경우, 레지스트리 변수는 매개변수 값을 겹쳐줍니다.

구성 매개변수	레지스트리 변수
<i>dir_path_name</i>	DB2DIRPATHNAME
<i>route_obj_name</i>	DB2ROUTE
<i>dft_client_comm</i>	DB2CLIENTCOMM
<i>dft_client_adpt</i>	DB2CLIENTADPT

이러한 레지스트리 변수 설정 규칙은 해당 구성 매개변수 설정 규칙과 같습니다. 예를 들어, *dft_client_comm* 매개변수처럼 DB2CLIENTCOMM은 다음 예에서 처럼 각각 쉼표로 구분되는 다중 값을 가질 수 있는 문자입니다.

CATALOG 및 ATTACH 명령 및 CONNECT문

DCE 정보는 다음과 같은 명령에서 지정되어야 합니다.

- CATALOG GLOBAL DATABASE 명령
- CONNECT문
- ATTACH 명령

CATALOG GLOBAL DATABASE 명령

클라이언트와 서버에 서로 다른 경로명이 있거나, 데이터베이스 이름이 9자 이상의 문자 또는 대문자가 섞여 있는 경우, CATALOG GLOBAL DATABASE 명령을 사용하십시오. 데이터베이스 관리자는 데이터베이스의 DCE 이름과 디렉토리 유형 DCE를 입력합니다.

예를 들면, 다음과 같습니다.

- 경로 이름이 서로 다를 경우, 즉 다음과 같을 때 *dir_path_name* = */.../CELL_TORONTO/subsys/database/*

```
CATALOG GLOBAL DATABASE
/.../CELL_VANCOUVER/subsys/database/VMDB AS VANVMDB
USING DIRECTORY DCE WITH "comment-string"
```
- 이름 DB_LONGNAME처럼 데이터베이스 이름이 9자 이상일 때

```
CATALOG GLOBAL DATABASE
/.../CELL_VANCOUVER/subsys/database/DB_LONGNAME AS VANVMDB
USING DIRECTORY DCE WITH "comment-string"
```

CONNECT문

적절한 DCE 디렉토리 오브젝트를 검색하기 위해, 클라이언트는 데이터베이스 또는 DBMS 인스턴스의 완전 규정화된 DCE 이름을 알아야 합니다. CONNECT 문에 이름을 지정하는 몇 가지 방법은 다음과 같습니다.

- 다음과 같이 별명을 입력합니다.

```
CONNECT TO VANVMDB
```
- 다음과 같이 한 부분의 이름을 입력합니다.

CONNECT TO VMDB

이 경우, 클라이언트에 지정된 경로 이름은 서버에 지정된 경로 이름과 같아야 합니다(경로명은 *dir_path_name* 구성 매개변수 또는 해당 레지스트리 값에 의해 지정됩니다).

ATTACH 명령

클라이언트의 효율적인 경로 이름은 목표 DBMS 인스턴스의 경로 이름과 같아야 합니다.

*dir_path_name*이 클라이언트 및 서버에 대한 이름이고(예: `/.../CELL_TORONTO/subsys/database/`), 데이터베이스 서버의 *dir_obj_name*이 AIX_INST이면, 인스턴스에 접속하기 위한 명령은 다음과 같습니다.

```
ATTACH TO AIX_INST
```

클라이언트를 데이터베이스에 연결하는 방법

471 페이지의 그림22에서는 두 가지 DCE 셀을 가진 데이터베이스 네트워크의 샘플 구성을 보여줍니다 `/.../CELL_TORONTO` 및 `/.../CELL_VANCOUVER`은 셀의 이름입니다(각각의 셀에는 `././subsys/database/`라는 디렉토리가 들어 있으며, 그림에는 없지만 다른 예에서 사용됩니다).

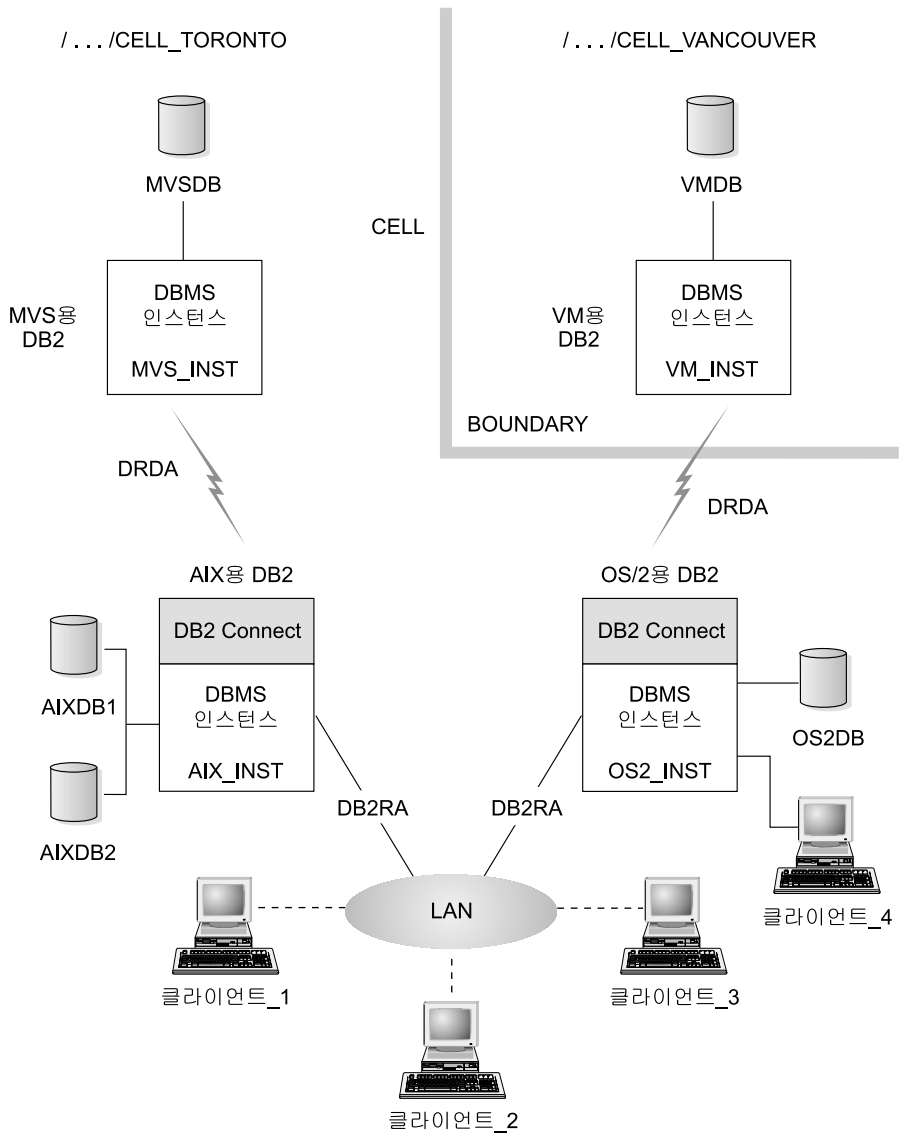


그림 22. 네트워크 데이터베이스의 구성

TORONTO 셀의 클라이언트가 두 셀의 모든 데이터베이스에 액세스할 수 있게 하려면, 데이터베이스 관리 프로그램 구성 매개변수에 값이 지정되어야 하며, 다음 오브젝트가 작성되어야 합니다.

- 데이터베이스 각각에 대한 데이터베이스 오브젝트

- AIX용 DB2 및 OS/2용 DB2에 대한 두 데이터베이스 서버의 데이터베이스 위치 지정자 오브젝트
- 각각의 클라이언트가 알고 있는 하나의 경로지정 정보 오브젝트. 이 속성은 MVSDB 및 VMDB 데이터베이스에 대해 어떤 DB2 Connect 노드를 사용할지 지정합니다.

다음은 클라이언트가 데이터베이스에 접속하는 방법의 예입니다.

- 같은 셀의 데이터베이스에 연결
- 서로 다른 셀의 데이터베이스에 연결.

이러한 예는 지정되어야 할 데이터베이스 관리 프로그램 구성 매개변수를 포함합니다.

같은 셀의 데이터베이스에 연결

이 절에서는 같은 셀에 있는 데이터베이스에 클라이언트가 접속하는 방법을 보여주는 몇 가지 예를 설명합니다.

1. Client_1을 AIXDB2에 연결하십시오. 데이터베이스는 클라이언트와 같은 디렉토리 경로를 공유합니다.

데이터베이스 관리자는 다음 작업을 수행해야 합니다.

- 구성 매개변수 *dir_path_name*(또는 DB2DIRPATHNAME 레지스트리 값)에 디렉토리 경로 이름 값을 지정하십시오.
- 구성 매개변수 *dir_type*에 디렉토리 서비스 유형 값이 DCE가 되도록 지정하십시오.
- 구성 매개변수 *dft_client_comm*(또는 DB2CLIENTCOMM 레지스트리 값)에 통신 프로토콜을 지정하십시오.

지역 시스템 데이터베이스 디렉토리는 AIXDB2를 가지고 있지 않으므로, 완전 규정화된 이름을 사용하여 DCE 디렉토리를 검색합니다. 이 이름은 구성 매개변수 *dir_path_name*(또는 D2DIRPATHNAME 레지스트리 값)에 대한 값을 AIXDB2에 연결함으로써 작성됩니다.

이벤트의 순서는 다음과 같습니다.

- a. 클라이언트_1은 데이터베이스 `./.../CELL_TORONTO/subsys/database/AIXDB2`를 사용하여 AIXDB2에 대한 데이터베이스 오브젝트를 확보합니다.

- b. 이 오브젝트, 클라이언트_1은 AIXDB2가 클라이언트_1이 사용하는 프로토콜과 같은 DB 프로토콜 DB2RA를 사용한다는 것을 알고 있습니다.
 - c. 이들 DB 프로토콜은 일치하는 것이므로 클라이언트_1은 AIX_INST에 대한 DBMS 위치 지정자 오브젝트를 읽고, 사용하는 통신 프로토콜 속성 값과 일치하는 값을 검색하고, 이 정보를 사용하여 DBMS 인스턴스와의 대화를 시작합니다.
2. 클라이언트_3을 MVSDDB에 연결하십시오. 데이터베이스는 클라이언트와 같은 디렉토리 경로 이름을 공유하며 다른 데이터베이스 프로토콜을 사용합니다.

데이터베이스 관리자는 다음 작업을 수행해야 합니다.

- 구성 매개변수 *dir_path_name*(또는 DB2DIRPATHNAME 레지스트리 값)에 디렉토리 경로 이름 값을 지정하십시오.
- 구성 매개변수 *dir_type*에 디렉토리 서비스 유형 값이 DCE가 되도록 지정하십시오.
- 구성 매개변수 *dft_client_comm*(또는 DB2CLIENTCOMM 레지스트리 값)에 통신 프로토콜을 지정하십시오.
- 구성 매개변수 *route_obj_name*(또는 DB2ROUTE 레지스트리 값)에 기본 라우트 정보 오브젝트의 DCE 이름을 지정하십시오.

이벤트의 순서는 다음과 같습니다.

- a. 클라이언트_3은 데이터베이스 `/.../CELL_TORONTO/subsys/database/MVSDDB`의 DCE 이름을 사용하여 MVSDDB에 대한 데이터베이스 오브젝트를 확보합니다.
- b. 이 오브젝트로부터, 클라이언트_3은 MVSDDB가 클라이언트_3이 사용하는 프로토콜이 아닌 DB 프로토콜 DRDA만을 사용한다는 것을 알게 됩니다.
- c. 클라이언트_3은 *route_obj_name* 구성 매개변수 또는 DB2ROUTE 레지스트리 값에 정의된 이름을 사용하여 경로지정 정보 오브젝트를 확보합니다. 클라이언트는 MVSDDB에 대한 목표 데이터베이스 정보를 발견합니다.
- d. 클라이언트_3은 MVSDDB 목표 데이터베이스 정보와 관련된 데이터베이스 위치 지정자 오브젝트를 읽고, 통신 프로토콜을 검색하고, 라우터로 SQL CONNECT 요청을 보냅니다.
- e. 그러면 라우터는 MVSDDB와의 APPC 연결을 설정합니다.

서로 다른 셀의 데이터베이스에 연결

이 절에서는 데이터베이스 프로토콜이 서로 다를 때 다른 셀에 있는 클라이언트를 데이터베이스에 연결하는 방법을 보여주는 예에 대해 설명합니다.

1. 클라이언트_3은 다음을 사용하도록 미리 구성되어 있습니다.

- *dir_type* 매개변수에 DCE라고 지정함으로써, DCE 디렉토리 서비스를 사용할 수 있습니다.
- 구성 매개변수 *dir_path_name*을 통한 CELL_VANCOUVER 이외의 셀로, 예를 들면, 다음과 같습니다.

```
./.../CELL_TORONTO/subsys/database/
```

2. 클라이언트_3이 VMDB로 접속하기 위해서는 데이터베이스 관리자가 다음을 수행해야 합니다.

- VMDB를 지역 시스템 데이터베이스 디렉토리에 명확하게 카탈로그화하십시오. VMDB에 대한 DCE 이름을 지역적으로 고유한 데이터베이스 별명과 결합하고, 별명 값을 가지고 CONNECT문을 입력하십시오. 예를 들면, 다음과 같습니다.

```
CATALOG GLOBAL DATABASE  
./.../CELL_VANCOUVER/subsys/database/VMDB AS VANVMDDB  
USING DIRECTORY DCE WITH "comment-string"
```

다음은 입력하십시오.

```
CONNECT TO VANVMDDB
```

- 구성 매개변수 *dft_client_comm*(또는 DB2CLIENTCOMM 레지스트리 값)에 통신 프로토콜을 지정하십시오.
- 구성 매개변수 *route_obj_name*(또는 DB2ROUTE 레지스트리 값)에 기본 라우트 정보 오브젝트의 DCE 이름을 지정하십시오.

이벤트의 순서는 다음과 같습니다.

- a. 클라이언트_3은 시스템 데이터베이스 디렉토리에서 완전한 DCE 이름인 VANVMDDB를 찾습니다.
- b. 클라이언트_3은 데이터베이스 `./.../CELL_VANCOUVER/subsys/database/VMDB`의 DCE 이름을 사용하여 VMDB에 대한 데이터베이스 오브젝트를 확보합니다.

- c. 이 오브젝트로부터 클라이언트_3은 VMDB가 클라이언트_3이 사용하는 프로토콜이 아닌 DB 프로토콜 DRDA만을 사용한다는 것을 알게 됩니다.
- d. 클라이언트_3은 *route_obj_name* 구성 매개변수 또는 DB2ROUTE 레지스트리 값에 정의된 이름을 사용하여 경로지정 정보 오브젝트를 확보합니다. 클라이언트는 VMDB에 대한 목표 데이터베이스 정보를 발견합니다.
- e. 클라이언트_3은 VMDB 목표 데이터베이스 정보와 관련된 데이터베이스 위치 지정자 오브젝트를 읽고, 통신 프로토콜을 검색하고, 라우터로 SQL CONNECT 요청을 보냅니다.
- f. 라우터는 VMDB와의 APPC 연결을 설정합니다.

디렉토리 검색 방법

만일 DCE 디렉토리가 모든 목표 데이터베이스가 같은 디렉토리 경로명을 공유하는 환경에서 사용되는 경우, 클라이언트에 어떠한 지역 디렉토리도 필요하지 않습니다.

이 절에서는 디렉토리의 다음 사항이 검색되는 순서에 대해 설명합니다.

- ATTACH 명령
- CONNECT문

ATTACH 명령

476 페이지의 그림23에서는 클라이언트가 ABC_INST라는 DBMS 인스턴스에 접속할 때 디렉토리가 검색되는 방법을 보여줍니다.

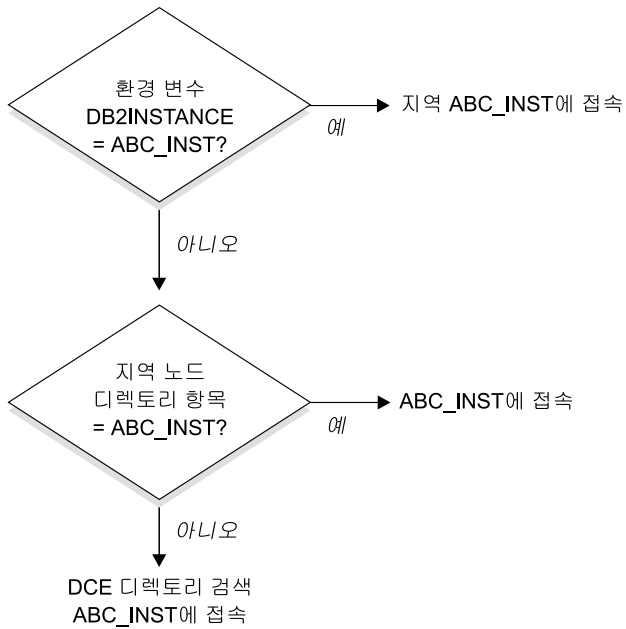


그림 23. 데이터베이스에 접속하기 위해 디렉토리가 사용되는 방법

CONNECT문

477 페이지의 그림24에서는 클라이언트가 DBTEST라는 데이터베이스에 연결될 때 디렉토리를 어떻게 검색할 것인지를 보여줍니다.

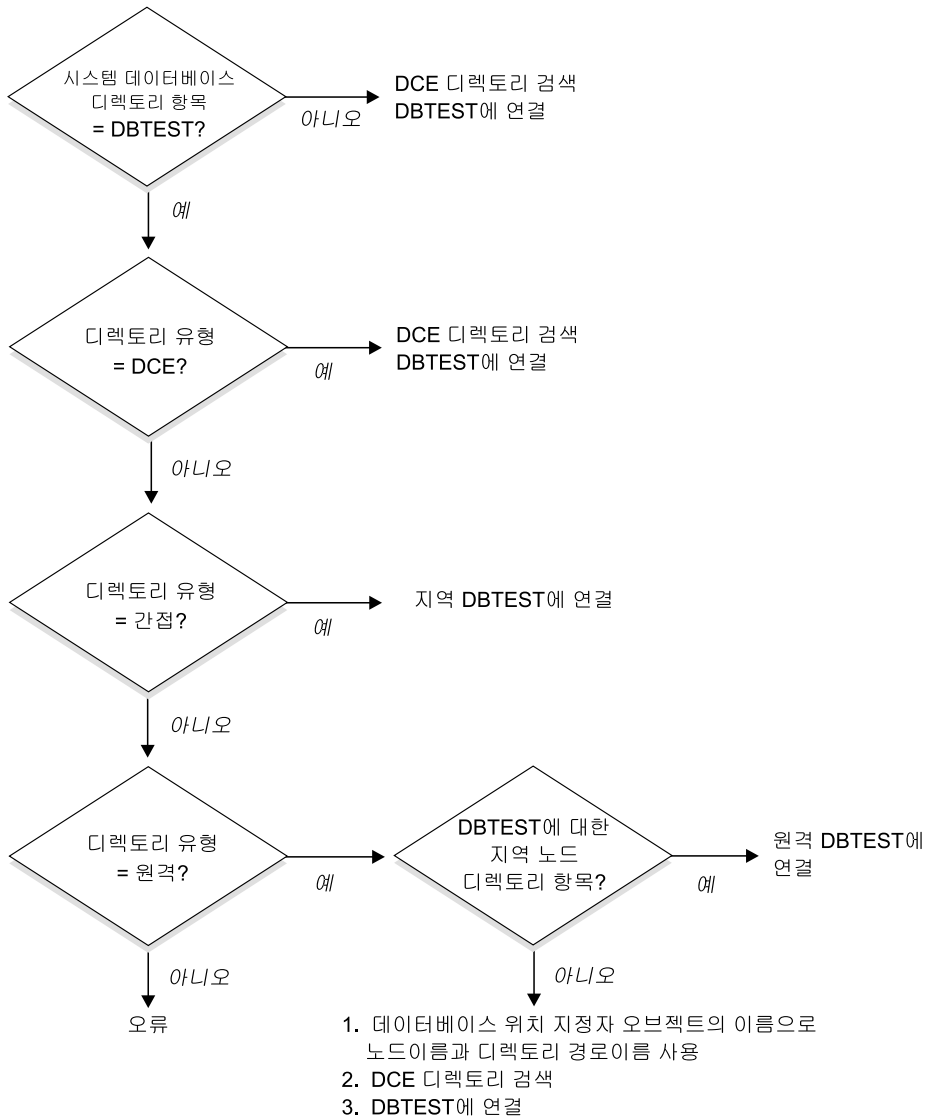


그림 24. 데이터베이스에 연결하기 위해 디렉토리가 사용되는 방법

DCE 디렉토리 정보를 임시로 대체

DCE 디렉토리 정보를 우선하기 위해서 지역 데이터베이스 디렉토리를 사용할 수 있습니다. 예를 들어, JAGUAR라고 하는 호스트에 있으면서 `././subsys/database/DBTEST`가 정의되어 있는 DBTEST에 연결하는 경우, 임시로 DBTEST를 STORM이라는 호스트의 다른 데이터베이스로 변경할 수 있습니다. STORM을 가리키는 노드 디렉토리 항목을 가지고 원격 데이터베이스로서 DBTEST를 지역적으로 카탈로그화하십시오.

DCE명이 클라이언트의 디렉토리 경로 이름을 따르지 않는 데이터베이스에 대해 별명을 만들 수 있습니다. 명령에 대해서는 469 페이지의 『CATALOG GLOBAL DATABASE 명령』에서 자세한 내용을 참조하십시오.

디렉토리 서비스 타스크

설정이 수행되고 DCE 디렉토리를 사용해야 할 타스크 목록은 다음과 같습니다. 다음에서는 각 타스크에 대한 세부사항을 설명합니다.

- DCE 관리자 타스크

DCE 관리자는 새로운 데이터베이스 자원 정보가 추가될 수 있도록 DCE 디렉토리를 갱신해야 합니다.

- 데이터베이스 관리자 타스크

데이터베이스 관리자는 DCE 디렉토리를 갱신하고 DB2 설치 및 구성에 관한 정보를 제공해야 합니다.

- 데이터베이스 사용자 타스크

데이터베이스 사용자는 DCE에 로그인하여 목표 데이터베이스 이름을 알아내야 합니다.

또한, 네트워크 관리자는 각각의 사용자 노드에 대한 네트워크 액세스도 설정합니다. 세부사항은 네트워크 문서를 참조하십시오.

DCE 관리자 타스크

DCE 관리자는 디렉토리 오브젝트가 작성되거나 읽히기 전에 다음 타스크를 수행해야 합니다.

- DB2에 대한 디렉토리 서브트리를 지정합니다(예: /./subsys/database).
- 디렉토리 오브젝트를 생성하기 위해 데이터베이스 관리자에게 특권을 권한 부여 하십시오.
- 디렉토리 오브젝트를 읽기 위해 데이터베이스 사용자에게 특권을 권한 부여하십시오.
- DCE 속성 테이블에 새로운 DCE 디렉토리 오브젝트 속성에 대한 정보를 추가 하십시오.

CDS 속성 파일(UNIX 플랫폼에서는 /etc/dce/cds_attributes, OS/2에서는 X:\op\ndcelocal\etc\cds_attr이며, 여기서 "X"는 해당 드라이브를 나타냄)을 편집한 후 다음을 첨부하십시오.

1.3.18.0.2.4.30	DB_Comment	char
1.3.18.0.2.4.31	DB_Communication_Protocol	char
1.3.18.0.2.4.32	DB_Database_Protocol	char
1.3.18.0.2.4.33	DB_Database_Locator_Name	char
1.3.18.0.2.4.34	DB_Native_Database_Name	char
1.3.18.0.2.4.35	DB_Object_Type	char
1.3.18.0.2.4.36	DB_Product_Name	char
1.3.18.0.2.4.37	DB_Product_Release	char
1.3.18.0.2.4.38	DB_Target_Database_Info	char
1.3.18.0.2.4.39	DB_Authentication	char
1.3.18.0.2.4.63	DB_Principal	char

- DCE 디렉토리 서비스를 사용하여 사용자가 데이터베이스에 액세스하고자 할 때, DCE가 수행되고 있는지를 확인하십시오.

자세한 정보는 해당 플랫폼의 DCE 문서를 참조하십시오.

데이터베이스 관리자 task

데이터베이스 관리자는 다음 task를 수행해야 합니다.

- DCE 관리자로부터 데이터베이스 자원에 대한 디렉토리 서브트리를 구하십시오 (예: /./subsys/database).
- DB2를 설치하는 동안 데이터베이스 관리 프로그램은 DCE 관리자에게 DB2가 요구하는 새로운 DCE 디렉토리 오브젝트 속성을 추가할 것인지 물으십시오.
- DCE 디렉토리 서브트리 각각의 DBMS 인스턴스에 대한 고유한 이름을 할당 하십시오(예: /./subsys/database/AIX_INST).

- 각각의 DBMS 인스턴스에 대해 DCE에 대한 데이터베이스 관리 프로그램 구성 매개변수를 지정하십시오.

- *dir_type*
- *dir_obj_name*
- *dir_path_name*
- *route_obj_name*
- *dft_client_comm*
- *dft_client_adpt*

일부 구성 매개변수는 클라이언트가 설정한 레지스트리 값으로 임시로 겹쳐쓸 수 있습니다. 467 페이지의 『구성 매개변수 및 레지스트리 변수』에서 자세한 내용을 참조하십시오.

- DCE 디렉토리 서브트리에 각각의 데이터베이스에 대해 고유한 이름을 할당하십시오. 이름을 데이터베이스 구성 파일의 *dir_obj_name* 매개변수에 지정하십시오.
- DCE *cdscp* 명령으로 오브젝트를 작성하고 표시함으로써, DCE 디렉토리 서비스에 대한 오브젝트를 작성하십시오. 오브젝트는 데이터베이스 관리자 설치 프로세스 및 데이터베이스 관리자 인스턴스 시작 프로세스와는 별도로 생성됩니다. 세 가지 유형의 오브젝트가 있습니다.
 - 각각의 목표 데이터베이스에 대해 하나의 데이터베이스 오브젝트가 요구됩니다.
 - 둘 이상의 데이터베이스와 연관된 각 DB2 Connect 인스턴스 및 각 DBMS 인스턴스(DB2 Connect 없이)에는 데이터베이스 위치 지정자 오브젝트가 필요합니다.
 - 호스트 데이터베이스 액세스를 위해서는 라우트 정보 오브젝트가 필요합니다.
- 각 환경에 따라 데이터베이스 관리자는 다음 사항을 결정해야 합니다.
 - 클라이언트가 어떤 데이터베이스에 액세스하고 어떤 프로토콜을 사용하는지를 고려하여 클라이언트를 논리 그룹으로 어떻게 묶어줄 것인지
 - 경로지정 정보 오브젝트가 몇 개나 필요한지
 - 어떤 목표 데이터베이스가 각각의 오브젝트에 기록되어야 하는지

- 어떤 라우트 정보 오브젝트가 어떤 그룹의 클라이언트에게 알려져야 하는지 오브젝트에 대해서는 454 페이지의 『디렉토리 오브젝트 작성』에서 자세한 내용을 참조하십시오.

데이터베이스 사용자 task

데이터베이스 사용자는 다음 task를 수행해야 합니다.

- 데이터베이스 관리자로부터 데이터베이스의 이름을 구하십시오. 이 이름은 간단하게 일부만으로 구성된 이름일 수도 있고, 완전 규정화된 DCE 이름일 수도 있습니다.
- 필요한 경우, 레지스트리 값에 DCE 디렉토리 서비스에 필요한 값을 지정하십시오. 클라이언트가 설정한 레지스트리 값이 구성 매개변수를 일시적으로 겹쳐 쓸 수 있습니다.
 - 만일 호스트 데이터베이스 액세스가 필요한 경우, 데이터베이스 관리자로부터 라우트 정보 오브젝트 항목의 완전한 DCE 이름을 찾으십시오. 이 이름이 *route_obj_name*에 지정되어 있지 않거나 다른 이름이면, 호스트 데이터베이스에 연결하기 전에 이 이름을 DB2ROUTE 레지스트리 값에 지정하십시오.
 - 선호되는 통신 프로토콜이 *dft_client_comm*에 지정되어 있지 않거나 다른 프로토콜인 경우, 클라이언트에 대한 통신 프로토콜을 DB2CLIENTCOMM 레지스트리 값에 지정하십시오. 다음은 몇몇 UNIX 예입니다.

```
db2set DB2CLIENTCOMM=tcpip
db2set DB2CLIENTCOMM=appc
db2set DB2CLIENTCOMM=tcpip,appc
db2set DB2CLIENTCOMM=appc,tcpip
```

다음은 몇몇 OS/2 예입니다.

```
db2set DB2CLIENTCOMM=ipxspx
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc
```

몇 가지 Windows NT 및 Windows 95 예는 다음과 같습니다.

```

db2set DB2CLIENTCOMM=npipe
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc,npipe

```

둘 이상의 통신 프로토콜이 존재할 경우, 첫번째로 지정된 것이 사용됩니다.

- *dir_path_name* 구성 매개변수 또는 DB2DIRPATHNAME 레지스트리 값에 없는 DCE 이름을 가지는 데이터베이스가 있으면, CATALOG GLOBAL DATABASE 명령을 사용하여 데이터베이스를 명시적으로 분류하십시오. 469 페이지의 『CATALOG GLOBAL DATABASE 명령』에서 자세한 내용을 참조하십시오.
- 목표 데이터베이스에 연결하거나 데이터베이스 인스턴스에 접속하기 전에 DCE에 로그인하십시오. 로그인 명령에 대해서는 *OSF DCE Administration Guide*에서 자세한 내용을 참조하십시오.

디렉토리 서비스의 제한사항

이 절에서는 지원되지 않는 사항에 대해 설명합니다.

- 모든 데이터베이스 클라이언트가 지원되지 않을 수도 있습니다. 어떤 DCE 디렉토리 서비스가 DB2 클라이언트로 지원되는지 판별하려면 빠른 시작 매뉴얼을 참조하십시오. 현재, 지원은 UNIX, OS/2 및 지원되는 Windows 운영 체제용 DB2 클라이언트에 대해서만 제공됩니다.
- 클라이언트는 OS/2용 DB2 버전 1 서버에 접속하기 위해서는 DCE 디렉토리 서비스를 사용할 수 없습니다.
- Windows 운영 체제에서 클라이언트만이 TCP/IP, APPC, NetBIOS, IPX/SPX 또는 NPIPE 프로토콜 일부 또는 전체를 사용할 수 있습니다. OS/2 클라이언트만이 TCP/IP, APPC, NetBIOS 및 IPX/SPX 프로토콜을 사용할 수 있습니다. 지원되는 모든 UNIX 클라이언트는 TCP/IP 및 APPC 프로토콜만 사용할 수 있습니다.
- LIST DATABASE(또는 NODE) DIRECTORY 명령만이 DCE 디렉토리가 아닌 지역 디렉토리로부터 항목을 제공할 수 있습니다. DCE의 *cdscp show object* 명령을 사용하여 오브젝트를 표시할 수 있습니다.

- 다음의 모든 조건이 존재하는 경우, 데이터베이스 관리 프로그램을 시작하기 전에 데이터베이스 관리 프로그램 인스턴스의 소유자는(db2start 명령을 사용하여) DCE에 로그인해야 합니다.
 - 데이터베이스 관리 프로그램 인스턴스는 *dir_type* 구성 매개변수를 통해 DCE 디렉토리 서비스를 지원하도록 구성됩니다.
 - 셸 디렉토리 서비스는 DCE에 명시적으로 로그인됨으로써 읽을 수만 있습니다.
 - 다음 중 하나를 지원하기 위해서는 DCE 디렉토리에 액세스해야 합니다.
 - 다른 인스턴스에 위치해 있는 트랜잭션 관리자 데이터베이스(*tm_database* 구성 매개변수에 의해 지정된)
 - DCE 디렉토리 서비스를 지원하지 못하거나 DCE 디렉토리 서비스를 사용하도록 구성되어 있지 않은 클라이언트

주: DCE 로그인을 수행할 때 long ticket lifetime을 갖는 principal을 사용해야 합니다.

- DDCS 버전 2.2(또는 이전 버전) 게이트웨이를 사용하여 DCE 디렉토리 서비스를 사용하는 클라이언트를 DRDA 서버로 연결할 때, 게이트웨이의 지역 디렉토리에 데이터베이스 별명을 카탈로그화해야 합니다. 이 데이터베이스 별명은 클라이언트에서의 별명과 같아야 하며 동일한 데이터베이스를 나타내야 합니다.
- Windows 운영 체제에서 클라이언트가 사용하는 경우, DB2DCE.DLL이 사용됩니다. 이 파일은 *sqllib* 서브디렉토리의 *bin* 서브디렉토리에 있습니다 DCE 제공자가 Gradient**이면 기본적으로 파일 DB2DCE.GRD는 DB2DCE.DLL과 같습니다. DCE 제공자가 IBM이면 파일 DB2DCE.IBM을 DB2DCE.DLL로 복사해야 합니다.

부록C. 데이터베이스 복구를 위한 User Exit

User Exit 프로그램을 통해 고유의 User Exit 프로그램을 개발하여 운영 체제가 직접 지원하지 않는 저장영역과 상호작용할 수 있습니다.

다음 주제에서는 User Exit 프로그램의 목적 및 고려사항에 대해 설명하고, 샘플 나감 프로그램 및 오류 처리에 대해 설명합니다.

- OS/2의 개요
- UNIX 기반 운영 체제의 개요
- User Exit 프로그램 호출
- User Exit 프로그램 샘플
- 호출 형식
- 아카이브 및 검색 고려사항
- 백업 및 복원에 대한 고려사항(OS/2용 DB2 전용)
- 오류 조절.

이미 언급되어 있는 것처럼 일부 정보는 특정 운영 체제 플랫폼에서만 적용이 가능한 것도 있습니다. 예를 들어, 백업 및 복원 User Exit는 UNIX 플랫폼에 적용할 수 없습니다.

OS/2의 개요

데이터베이스 관리 프로그램은 로그 파일을 아카이브하고 검색하기 위해, 데이터베이스를 백업하고 복원하기 위해 또는 두 가지를 위해 선택적으로 User Exit 프로그램을 호출할 수 있습니다. 한 쌍의 타스크(백업과 복원 또는 아카이브와 검색)를 위한 User Exit 프로그램을 호출하는 것은 User Exit 프로그램이 다른 쌍의 타스크를 위해 사용되는 것을 요구하지 않습니다. 예로써, 만약 사용자가 User Exit 프로그램으로 로그를 아카이브하고 검색한다면, 사용자는 User Exit 프로그램으로 데이터베이스를 백업하고 복원하도록 요구받지 않습니다.

데이터베이스 관리 프로그램은 다음 조치를 통해 User Exit 프로그램을 호출할 수 있습니다.

백업 BACKUP DATABASE 유틸리티는 명령행 처리기에서 대상 드라이브 매개변수로 0:을 지정하거나, API 호출에 대한 미디어 유형으로 U를 지정하여 User Exit 프로그램을 호출합니다. 데이터베이스 백업에 관한 추가 정보는 355 페이지의 『데이터베이스 백업』에서 자세한 내용을 참조하십시오.

복원 RESTORE DATABASE 유틸리티는 User Exit 프로그램을 호출하는 BACKUP DATABASE에 의해 사전에 저장된 데이터베이스 파일을 검색하기 위해 User Exit 프로그램을 호출합니다. RESTORE DATABASE 유틸리티는 명령행 처리기로부터의 소스 드라이브 매개변수로 0:을 지정하거나, API 호출의 매체 유형으로서 U를 지정하여 User Exit 프로그램을 호출합니다. 데이터베이스 복원에 관한 추가 정보는 365 페이지의 『데이터베이스 복원』에서 자세한 내용을 참조하십시오.

아카이브 및 검색

데이터베이스 관리 프로그램 아카이브 및 검색 함수는 데이터베이스 구성 매개변수 *userexit*가 지정되어 있는 경우, 로그 파일을 저장하고 검색하기 위해, 그리고 보존된 로그 파일의 위치를 관리하기 위해 User Exit 프로그램을 호출합니다. 파일을 아카이브하고 검색하기 위해 User Exit 프로그램을 사용하는 것은 데이터베이스가 롤 포워드 복구를 할 수 있도록 합니다(380 페이지의 『데이터베이스의 롤 포워드 변경사항』 참조).

주: *userexit* 구성 매개변수는 로그 파일을 아카이브 및 검색하는 데에만 적용됩니다.

UNIX 기반 운영 체제의 개요

데이터베이스 관리 프로그램은 데이터베이스 구성 매개변수 *userexit*가 지정된 경우, 로그 파일을 저장하고 검색하기 위해, 그리고 보존된 로그 파일의 위치를 관리하기 위해 User Exit 프로그램을 호출합니다. 파일을 아카이브하고 검색하기 위해 User Exit 프로그램을 사용하는 것은 데이터베이스가 롤 포워드 복구를 할 수 있도록 합니다(380 페이지의 『데이터베이스의 롤 포워드 변경사항』 참조).

User Exit 프로그램 호출

User Exit 프로그램을 호출하면, 데이터베이스 관리 프로그램은 수행 가능한 파일 즉, [db2uext2]에 제어를 넘겨줍니다.

주: 백업 및 복원 조작으로 [db2uext2]를 차례로 호출하는 [db2usrxt.cmd]를 처음에 호출합니다.

데이터베이스 관리 프로그램은 이 프로그램으로 매개변수를 넘겨주고 종료시에 프로그램은 데이터베이스 관리 프로그램에 리턴 코드를 넘겨줍니다. 데이터베이스 관리 프로그램은 단지 리턴 상태의 한정된 세트만 처리하므로 User Exit 프로그램이 오류 상태를 처리해야 합니다.

단 하나의 User Exit 프로그램만이 데이터베이스 관리 프로그램 인스턴스 내에서 호출될 수 있습니다. 따라서 각 프로그램은 아카이브, 검색, 백업(OS/2 전용) 및 복원(OS/2 전용)을 포함하여 수행해야 하는 모든 조치에 대한 섹션을 포함해야 합니다. User Exit 프로그램으로 넘겨진 매개변수 중 하나가 이들 조치 중 어느 것이 요구되었는지를 나타냅니다.

User Exit 프로그램 샘플

샘플 프로그램의 수는 다른 장치 또는 소프트웨어 공유를 위한 User Exit 함수의 사용을 보여주기 위해 제공됩니다. 프로그램 목록은 사용된 장치 지원 소프트웨어의 버전을 식별합니다.

사용자는 이들 프로그램을 원하는 방법대로 수정하거나 사용할 수 있습니다. 이 샘플 프로그램의 주석은 사용자의 User Exit 프로그램을 쓰는 데 필요한 전문 기술 정보를 제공합니다.

다음 주제에서는 운영 체제와 관련된 샘플 프로그램에 대한 정보를 제공합니다.

- OS/2 샘플 User Exit 프로그램
- UNIX 기반 운영 체제용 샘플 User Exit 프로그램.

OS/2 샘플 User Exit 프로그램

OS/2용 DB2에 대한 User Exit 샘플 프로그램은 \sqllib\samples\rexx 디렉토리의 인스턴스 서브디렉토리에 있습니다. 마지막 User Exit 샘플 프로그램 (dbuexit.CAD)은 예외로 \sqllib\samples\c 디렉토리의 인스턴스 서브디렉토리에 있습니다. 구현하려는 샘플은 확장자 이름이 .cmd 또는 .exe인 db2uexit의 실행 가능한 파일명으로 이름 바꾸기해야 합니다. 이러한 이름 바꾸기된 파일은 User Exit 프로그램으로 사용하기 위해 \sqllib\bin 디렉토리에 저장되어야 합니다.

제공된 샘플이 대부분 REXX 명령 파일인 반면, 사용자의 User Exit 프로그램은 다른 프로그램 언어로 기록될 수 있습니다. 실행 가능한 파일 이름은 확장자가 cmd 또는 exe인 db2uexit여야 합니다.

다음 5가지의 OS/2 샘플 프로그램이 제공됩니다.

- **db2uexit.ex1**

이 프로그램은 Seagate** Software의 Sytos Premium** 버전 2.2 프로그램을 사용하여 IBM 외부 테이프 장치에 데이터를 저장하고 검색합니다.

주: 버전 2.2의 Sytos Premium 제품만 현재 지원됩니다. 이 제품을 사용하기 위해서는 OS/2 FixPack 26이 필요합니다.

사전 정의 프로시듀어와 같은 요구사항을 확실히 하기 위해 샘플 프로그램 목록을 검토하십시오.

- **db2uexit.ex2**

이 프로그램은 Mountain 테이프 장치에 있는 데이터를 저장하고 검색하기 위해 Mountain** Corporation으로부터 제공받을 수 있는 Filesafe** 프로그램을 사용합니다.

고유한 볼륨 레이블은 동일 데이터베이스 또는 다른 데이터베이스의 다중 백업이 같은 테이프상에 저장될 수 있도록 데이터베이스의 각 백업 사본에 지정되어 있습니다. 데이터베이스가 복원될 때, 이 프로그램은 가장 최근의 백업 사본을 선택합니다. 이 기능은 백업 로그 파일을 수정함으로써 우회될 수 있습니다.

- **db2uexit.ex3**

이 프로그램은 Maynard 테이프 장치에 있는 데이터를 저장하고 검색하기 위해 Maynard** Corporation으로부터 제공받을 수 있는 MaynStream** 프로그램을 사용합니다.

MaynStream은 복원된 데이터베이스가 데이터베이스의 백업된 드라이브가 아닌 다른 드라이브로 경로를 재지정하는 것을 지원하지 않습니다.

- **db2uexit.ex4**

이 프로그램은 OS/2 XCOPY 명령을 사용합니다. 이 저장영역은 하드 디스크, 디스켓 또는 광 카트리지와 같이 OS/2에서 지원하는 어떠한 장치도 될 수 있습니다. 이 장치는 워크스테이션이 경로 재지정된 드라이브를 지원하도록 설정된 경우, LAN 경로 재지정 드라이브가 될 수 있습니다.

XCOPY는 데이터베이스를 백업하고 복원하는 데에는 사용할 수 없습니다.

- **db2uexit.CAD**

이 C 프로그램은 UNIX 기반 운영 체제의 샘플 프로그램에서 소개된 대로 데이터베이스 로그 파일을 아카이브하고 검색하기 위해 Tivoli 분산 저장영역 관리 프로그램(TSM) 샘플 프로그램과 같습니다.

UNIX 기반 운영 체제용 샘플 User Exit 프로그램

userexit 구성 매개변수로 인해 데이터베이스 관리 프로그램은 로그를 아카이브하고 검색하기 위해 User Exit 프로그램을 호출합니다. UNIX 플랫폼상의 네 개의 IBM 제공 샘플 User Exit 프로그램(디스크용, 테이프용, TSM용 및 Legato** NetWorker**용)이 있습니다. 이 프로그램을 반드시 사용해야 하는 것은 아닙니다. 사용자의 User Exit 프로그램 작성을 선택할 수도 있습니다. 샘플 프로그램은 사용자에게 User Exit 프로그램 작성시 이용할 수 있는 모델 또는 추천사항을 제공합니다. 각 샘플 프로그램의 헤더 정보에 유용한 정보가 나와 있습니다.

제공된 샘플이 C 언어로 코딩되어 있는 상태에서 User Exit 프로그램은 다른 프로그래밍 언어로 쓰여질 수 있습니다. User Exit 프로그램은 실행 가능한 파일 이름인 *db2uext2*여야 합니다.

다음 네 가지의 UNIX 기반 운영 체제 샘플 프로그램이 제공됩니다.

- **db2uext2.cadsm**

이 프로그램은 데이터베이스 로그 파일을 아카이브하고 검색하기 위해 Tivoli Storage Manager 유틸리티를 사용합니다.

- **db2uext2.ctape**

이 프로그램은 테이프 미디어를 사용하는 데이터베이스 로그 파일을 아카이브하고 검색합니다.

- **db2uext2.cdisk**

이 프로그램은 디스크 미디어를 사용하는 데이터베이스 로그 파일을 아카이브하고 검색하는 운영 체제 복사 명령을 사용합니다.

- **db2uxt2.cxbsa**

이 프로그램은 데이터베이스 로그 파일을 아카이브하고 검색하기 위해 Legato** Systems, Incorporated의 Legato NetWorker** 버전 4.2.5 유틸리티를 사용합니다. 이 프로그램은 AIX에만 적용시킬 수 있습니다.

호출 형식

데이터베이스 관리 프로그램은 요청받은 대로 User Exit 프로그램을 호출하고, 매개변수 세트를 넘겨줍니다. 이 매개변수는 문자열 또는 문자의 데이터 유형을 가집니다.

호출 형식은 다음 주제에서 설명하는 것처럼 운영 환경에 따라 달라집니다.

- OS/2용 호출 형식
- UNIX 기반 또는 Windows NT 운영 체제용 형식.

OS/2용 호출 형식

다음은 OS/2 User Exit 프로그램을 호출하기 위한 데이터베이스 관리 프로그램 형식입니다.

```
action drive db_alias log_path log_file indicator
```

action BACKUP, RESTORE, ARCHIVE 또는 RETRIEVE 값을 포함합니다.

drive BACKUP의 경우, 이 매개변수는 데이터베이스가 백업되는 드라이브를 포함합니다.

RESTORE의 경우, 이 매개변수는 데이터베이스가 복원되는 드라이브를 포함합니다.

ARCHIVE 및 RETRIEVE의 경우, 이 매개변수는 데이터베이스가 위치되어 있는 드라이브를 포함합니다.

이 매개변수 형식은 콜론이 뒤에 붙는 드라이브 이름 형태를 가집니다(예: C:).

db_alias 데이터베이스 별명과 데이터베이스에 대한 별명이 존재하지 않으면 데이터베이스 이름을 포함합니다.

log_path BACKUP의 경우, 이 매개변수는 백업될 파일 목록을 포함하고 있는 응답 파일의 완전한 이름을 포함합니다. 목록의 각 파일 이름은 완전한 이름이며 총칭 문자를 가질 수 있습니다.

RESTORE의 경우, 이 매개변수는 복원될 파일 목록인 응답 파일의 완전한 이름을 포함합니다. 목록의 각 파일 이름은 완전한 이름이며 총칭 문자를 가질 수 있습니다. 드라이브명 및 경로는 데이터베이스가 백업되는 시점에서의 원시 드라이브 및 경로입니다. 예를 들어, 응답 파일에 C:\SQLUTIL\dbname.MH1이 들어 있을 경우, dbname.MH1 파일이 C:\SQLUTIL로부터 백업된다는 의미입니다.

ARCHIVE 및 RETRIEVE의 경우, 이 매개변수에는 로그 경로 디렉토리가 포함됩니다(예: C:\SQL00001\SQLLOGDIR\).

log_file BACKUP의 경우, 이 매개변수는 BACKUP DATABASE 유틸리티에 의해 생성된 미디어 레이블을 포함합니다. 이 레이블은 데이터베이스 별명 및 시간소인으로 구성됩니다.

RESTORE의 경우, 이 매개변수는 파일이 복원될 데이터베이스 서브디렉토리의 경로 이름을 포함합니다. 드라이브 이름은 *drive* 매개변수에 나타나 있기 때문에 포함되어 있지 않습니다. 형식은 \SQLnnnnn\입니다.

ARCHIVE 및 RETRIEVE의 경우, 이 매개변수에는 로그 파일명이 포함됩니다(예: S0000001.LOG).

indicator indicator는 백업 또는 복원 조작 동안에 다중 호출을 지원하는 데 사용됩니다. 첫번째 호출은 문자 '1'의 값을 갖고, 후속 호출은 문자 '2'의 값을 가집니다.

User Exit 프로그램은 백업 또는 복원 조작 동안 여러 번 호출됩니다. 첫번째 호출은 미디어 헤더 파일(.MHN 파일)을 백업 또는 복원하고, 두번째 호출은 데이터베이스 파일의 전체 세트를 백업 또는 복원합니다.

ARCHIVE 및 RETRIEVE의 경우, 이 매개변수는 사용되지 않습니다.

UNIX 기반 또는 Windows NT 운영 체제용 형식

다음은 데이터를 아카이브하거나 검색하기 위해 UNIX 기반 또는 Windows NT 운영 체제 User Exit 프로그램을 호출할 때 사용하는 데이터베이스 관리 프로그램 형식입니다.

```
db2uext2 -OS<os> -RL<db2rel> -RQ<request>
-DB<dbname> -NN<nodenum> -LP<logpath>
-LN<logname> -AP<adsmpasswd> -SP<startpage>
-LS<logsize>
```

	os	인스턴스가 수행되고 있는 플랫폼: AIX, NT, Solaris, HP-UX, SCO, Linux, Dynix/ptx and SGI.
	db2rel	DB2 릴리스 레벨. 예, SQL07010.
	request	요청 유형. ARCHIVE 또는 RETRIEVE가 가능합니다.
	dbname	데이터베이스 이름
	nodenum	지역 노드 번호(예: 5)
	logpath	로그 파일에 완전한 경로. 경로에는 반드시 트레일링 경로 분리 문자가 있어야 합니다(예: /u/database/log/path/ 또는 d:\logpath\).
	logname	아카이브되거나 검색될 로그 파일 이름(예: S0000123.LOG)
	tsmpasswd	TSM password. 데이터베이스 구성에서 제공된다면 User Exit에 전달됩니다.

startpage 로그 extent는 장치의 오프셋 4KB 페이지의 수에서 시작됩니다.

logsize 4KB 페이지 단위의 로그 extent의 크기

주:

1. Windows NT 및 Unix만이 로그 아카이브 및 검색을 위해 User Exit를 지원합니다.
2. **-LS** 및 **-SP** 매개변수는 원래 장치가 로깅에 사용되는 경우에만 사용됩니다. 로깅을 위해 파일을 사용하는 기존의 사용자 User Exit 프로그램을 사용하는 경우, 매개변수를 변경할 필요가 없습니다.

아카이브 및 검색 고려사항

다음 고려사항은 로그 파일을 아카이브하고 검색하기 위해 User Exit 프로그램을 호출할 때 적용됩니다.

- 데이터베이스 구성 파일 매개변수 *userexit*는 데이터베이스의 롤 포워드 복구 동안에 로그 파일을 아카이브하거나 로그 파일을 검색하기 위해 데이터베이스 관리 프로그램이 User Exit 프로그램을 호출하는지 여부를 명시합니다. 로그 파일을 검색하려는 요청은 롤 포워드 데이터베이스 복구 유틸리티가 로그 경로 디렉토리에 존재하지 않는 로그 파일을 필요로 할 때 만들어집니다.

주:

1. Windows NT에서, 로그를 아카이브하기 위해 REXX User Exit를 사용할 수 없습니다.
- 아카이브 작업시에 로그 파일이 아직 활동중이고 정상 처리에 필요하더라도 파일이 채워진 상태로 User Exit에 전달됩니다. 이것은 데이터의 사본이 가능한 한 빨리 휘발성 미디어로부터 이동되어 나오는 것을 허용합니다. User Exit에 넘겨진 로그 파일은 더 이상 정상 처리에 필요하지 않게 될 때까지 로그 경로 디렉토리에 남겨져 있게 됩니다. 이 시점에서 디스크 공간은 재사용됩니다.
 - DB2는 User Exit를 시작하여 로그 파일을 아카이브할 때 읽기 모드에서 파일을 엽니다. 그러므로, User Exit는 파일이 계속 사용중일 때 파일을 삭제할 수 없습니다. DB2는 파일이 비활성화될 때 파일을 닫습니다. 파일이 비활성화될 때 User Exit가 완료되면, 로그 파일을 삭제할 수 있지만 이렇게 하는 데에는 성능 상의 비용이 듭니다.

- 로그 파일이 아카이브되었는데 그 파일이 비활동 상태인 경우, DB2는 파일을 삭제하지 않지만 이러한 파일이 필요할 때 다음 로그 파일로 이름을 변경합니다. 그러면, 새로운 로그 파일을 작성할 때(파일 이름을 바꾸는 대신) 모든 페이지가 디스크 공간에 확실하게 기록되어야 하므로 성능상의 이득을 보게 됩니다. 디스크에서 필요한 페이지를 비운 후 다시 확보하는 것보다 재사용하는 것이 더 좋습니다.
- DB2는 손상 복구 또는 구간 복원 둘다에서 로그 파일을 검색하기 위해 User Exit를 호출하지 않습니다.
- User Exit 프로그램은 롤 포워드 복구가 실패에 대한 것을 보증하지는 않지만 실패 가능성을 줄이기 위해 시도합니다. 로그 파일이 채워지면 User Exit 루틴을 위한 대기행렬에 있게 됩니다. 로그 파일이 채워지기 전에 로그를 포함하고 있는 디스크가 작업을 실패하게 되면, 로그 파일 안에 있는 데이터를 분실하게 됩니다. 또한, 파일이 아카이브 작업의 대기행렬에 있으므로, 모든 파일들이 복사되기 전에 디스크가 작업 실패를 할 수 있습니다. 그러면, 대기행렬에 있는 파일은 모두 분실됩니다.
- 각 개별 로그 파일의 구성된 크기는 User Exit에 직접적인 관련을 가집니다. 각 로그 파일이 매우 큰 경우, 아주 많은 양의 데이터가 디스크 작업 실패시 분실됩니다. 작은 로그 파일로 구성된 로그 파일은 데이터가 User Exit에 더욱 자주 넘겨지도록 하고 있습니다.

그러나, 만약 사용자가 테이프와 같이 속도가 느린 장치로 데이터를 보낸다면, 사용자는 대기행렬이 최고로 올라가는 것을 방지하기 위해 더 큰 로그 파일을 갖기 원할 수도 있습니다. 대기행렬이 완전히 채워지게 되면, 아카이브 및 검색 요청은 처리되지 않습니다. 처리는 대기행렬에 들어갈 여유가 있을 때 다시 시작됩니다. 처리되지 않은 어떠한 요청도 자동으로 다시 대기행렬에 보내지지는 않습니다.

- User Exit 프로그램에 대한 아카이브 요청은 *user exit*이 구성되고 사용중인 로그 파일이 채워지는 시간마다 발생합니다. 데이터베이스에서의 연결을 마지막으로 해제하고, 더구나 User Exit 프로그램을 부분적으로 채워진 사용 로그 파일에 호출하면, 사용중인 로그 파일이 채워지지 않을 수 있습니다.

주: 사용되지 않은 로그 공간을 해제하려면 로그 파일을 아카이브하기 전에 절단하십시오.

- 로그 파일을 포함하는 장치에 미디어 장애가 발생하는 경우, 로그 사본은 오프라인 로그 파일이 롤 포워드 복구에 의해 사용될 수 있도록 하기 위해 또 다른 장치에 만들어져야 합니다. 이 장치는 데이터베이스 데이터 파일을 포함하는 동일한 장치가 될 수 없습니다.
- 데이터베이스가 아카이브 요청에 대해 User Exit 프로그램으로부터 긍정적인 반응을 얻기 전에 폐쇄되어 버렸다면, 데이터베이스 관리 프로그램은 데이터베이스가 열릴 때 또다른 요청을 보냅니다. 그래서 로그 파일은 두 번 이상 아카이브됩니다. 사용자가 이러한 다중 아카이브 작업이 발생하는 것을 원하지 않는다면, User Exit 프로그램은 같은 파일의 아카이브에 대한 후속 요청을 허용하지 말아야 합니다.
- User Exit 프로그램이 존재하지 않는 파일(다중 아카이브 요청이 있었기 때문이거나 첫번째 아카이브가 성공적으로 끝난 후에 파일이 삭제되었기 때문에)을 아카이브하는 요청을 받은 경우 또는 존재하지 않는 파일(다른 디렉토리에 있거나 로그의 끝부분에 도달해 있기 때문에)을 검색하는 요청을 받은 경우에 User Exit 프로그램은 이 요청을 무시하고 성공적으로 실행되었다는 리턴 코드를 보내줍니다.
- User Exit는 원격 클라이언트가 DB2 서버로의 연결을 잃어버린 경우 인터럽트될 수 있습니다. 즉, User Exit를 통한 로그의 아카이브 처리중에 다른 SNA 연결 클라이언트 중 하나가 죽었거나 전원이 차단되어 신호(SIGUSR1)가 서버로 보내지는 결과를 가져올 수 있습니다. 이 서버는 신호를 User Exit에 보내어 인터럽트가 발생하도록 합니다. User Exit 프로그램은 인터럽트 점점을 위해 수정될 수 있으며, 이후 계속될 수 있습니다.
- User Exit 프로그램은 각기 다른 로그 파일이 특정 시점 복구 후에 동일명으로 존재하는 것을 허용합니다. 그것은 두 개의 로그 파일을 아카이브하고, 정확한 복구 경로에 의해 로그 파일을 결합하도록 작성되어야 합니다(400 페이지의 『로그 파일 관리를 위한 고려사항』 참조).
- 두 개 이상의 데이터베이스가 동시에 하나의 장치를 쓰고 있고 조작 중 하나를 롤 포워드 조작을 수반하고 있다면, 롤 포워드 복구를 위해 필요한 로그 파일은 현재 장치 내에 있는 미디어에 존재하지 않을 수도 있습니다. 다음의 두 상태가 발생할 수 있습니다.

- User Exit 프로그램이 데이터베이스 관리 프로그램에게 리턴 코드 0(성공한)을 넘겨주고 요청된 로그 파일이 검색되지 않으면, 데이터베이스 관리 프로그램은 롤 포워드 조장이 로그 파일의 끝까지 완전히 행해지고 롤 포워드 조장이 중단되었다고 판단합니다. 그러나, 롤 포워드 처리는 로그 파일의 끝까지 도달하지 않을 수도 있습니다.
- 만약 0 이외의 리턴 코드가 리턴되면, 데이터베이스 롤 포워드 오류 상태에 머무르게 되고, 사용자는 롤 포워드 처리를 다시 시작하거나 중단해야 합니다.

이 두 가지 상황의 발생을 방지하기 위해 사용자는 User Exit 프로그램을 호출하는 노드 상의 어떠한 다른 데이터베이스도 롤 포워드 조작중에는 열리지 않도록 하거나, 이러한 상황을 처리할 수 있도록 User Exit 프로그램을 작성할 수 있습니다.

백업 및 복원에 대한 고려사항(OS/2용 DB2 전용)

다음 고려사항은 BACKUP DATABASE 및 RESTORE DATABASE 유틸리티로부터 호출되는 User Exit 프로그램을 작성하는 경우에 적용됩니다.

- User Exit 프로그램에 의해 리턴되는 0 이외의 리턴 코드는 유틸리티의 실패를 일으키고 재시도되지 않습니다.
- 와일드카드는 완전한 파일 이름의 파일 이름에서 지원되어야 합니다. 예를 들어, C:\SQL00001*.* 및 C:*.MH*는 승인 가능한 검색 표준입니다.
- User Exit 프로그램은 각각의 행이 캐리지 리턴 또는 라인 피드에 의해 종료되는 라인별로 완전한 하나의 파일 이름을 가지는 응답 파일 형식을 처리해야만 합니다. 파일에 파일 끝(eof) 문자가 없습니다.
- 동일 데이터베이스의 다중 백업이 하나의 미디어에 설정되면, User Exit 프로그램은 복원 조작 중에 정확한 백업 버전이 선택되도록 설계되어야 합니다(488 페이지의 『OS/2 샘플 User Exit 프로그램』에 서술되어 있는 **db2uexit.ex2** 샘플을 참조하십시오).
- 하나의 백업 장치를 공유하여 동시에 수행되는 두 개의 백업 프로세스는 직렬이어야 합니다.
- 백업 이미지가 둘 이상의 미디어에 걸쳐 있게 되면, 미디어에 대한 프롬프트는 그것이 호출하는 User Exit 프로그램 또는 응용프로그램에 의해 처리되어야만

합니다. 이러한 상태를 지원하기 위해 BACKUP DATABASE 및 RESTORE DATABASE는 User Exit 프로그램을 호출하는 운영 체제 포어그라운드 세션을 시작합니다.

- User Exit 프로그램은 데이터베이스 디렉토리 내의 어떠한 서브디렉토리도 백업해서는 안됩니다.
- User Exit 프로그램을 사용하여 데이터베이스를 복원할 때 RESTORE DATABASE는 데이터베이스에 대한 완전한 제어를 필요로 합니다. 그러나, 워크스테이션은 복원되는 데이터베이스보다는 다른 데이터베이스의 활동중인 연결을 할 수 있습니다.
- 데이터베이스가 User Exit 프로그램으로 백업 또는 복원되고, 다른 작업이 동일한 테이프 장치를 사용하고 있을 때 백업 또는 복원 작업은 실패할 수 있습니다. 백업 또는 복원 작업은 다시 시작되어야 합니다. 이러한 상황을 피하기 위해 사용자는 로깅하기 위해 User Exit 프로그램을 호출하는 워크스테이션의 어떠한 다른 데이터베이스도 백업 또는 복원 작업이 진행중일 때에는 사용되지 않고, 아직 장치가 준비되지 않았으므로 나중에 User Exit 프로그램이 백업 또는 복원 작업을 재시도해야 함을 확인합니다.
- 복원 작업의 드라이브명 및 경로는 백업 작업중에 지정된 것과는 다를 수 있습니다. 예를 들어, dbname.MH1 파일이 C:\SQLUTIL로부터 백업될 경우, 이를 d:\xxx로 백업할 수 있습니다.

오류 조절

데이터베이스 관리 프로그램이 User Exit 프로그램에서 리턴 코드를 적절하게 처리할 수 있도록, 프로그램을 반드시 특정 결과를 보여줄 수 있는 특정 리턴 코드를 제공할 수 있게 코드화해야 합니다.

498 페이지의 표19에서는 User Exit 프로그램에 의해 리턴될 수 있는 리턴 코드 및 데이터베이스 관리 프로그램이 어떻게 해당 리턴 코드를 해석하는지를 보여줍니다. 만약 리턴 코드가 표에 기록되어 있지 않으면, 값이 32인 것으로 취급됩니다.

표 19. User Exit 리턴 코드 및 결과

리턴 코드	결과(주 1)	설명
0	—	성공적
4	주 2	임시 자원 오류를 만났습니다.
8	주 2	조직원 개입이 요구됩니다.
12	주 3	하드웨어 오류
16	주 3	User Exit 프로그램에 의한 오류 또는 프로그램에 의해 사용된 소프트웨어 함수에 의한 오류
20	주 3	User Exit로 넘겨진 하나 이상의 매개변수에 의한 오류. User Exit 프로그램이 주어진 매개변수를 정확하게 처리했는지를 확인합니다.
24	주 3	User Exit 프로그램을 찾을 수 없습니다. OS/2의 경우, 이 오류 메시지는 RESTORE DATABASE 작업을 완성시키기 위해 필요한 파일을 현재의 백업 미디어에서 찾을 수 없음을 의미합니다.
28	주 3	입출력 조작 실패 또는 운영 체제에 의해 발생된 오류
32(및 다른 모든 값)	주 3	User Exit 프로그램이 사용자에게 의해 종료되었습니다.

주:

1. 아카이브 및 검색 조치에만 적용합니다.
2. 아카이브 및 검색의 경우, 4 또는 8 리턴 코드는 5분 내에 재시도를 시킵니다. 동일한 로그 파일과 관련된 아카이브 요청은 5분 내에 재시도됩니다.
3. User Exit 프로그램 요청은 5분 동안 일시중지됩니다. 이 시간 동안, 리턴 코드를 일으킨 로그 파일 요청을 포함하는 모든 요청들이 무시됩니다.

요청을 처리할 때 5분간 일시중지한 후 다음 요청이 처리됩니다. 이 요청을 처리하는 동안 오류가 발생하면 새로운 User Exit 프로그램 요청에 대한 처리가 계속됩니다. 재시도를 하는 동안 8보다 큰 리턴코드가 작성될 경우, 요청은 5분간 더 일시중단됩니다. 5분간 일시중지는 문제점이 정정되거나, 데이터베이스를 중단시키고 재시작할 때까지 계속됩니다.

모든 응용프로그램이 데이터베이스로부터 연결해제된 다음 이 데이터베이스가 다시 열리면, 문제점을 일으킨 로그 파일과 관련된 요청이 반복됩니다. 올바르게 처리되면, 일시중단중에 작성된 모든 추가 요청이 처리됩니다.

User Exit를 보존 로그에 호출한 경우, 사용자의 디스크는 로그 파일로 채워지게 되고 로그 파일을 형식화하기 위해 작업을 더 하게 되기 때문에 성능이 저하될 수 있습니다. 한 번 디스크가 완전히 채워지면, 데이터베이스 관리 프로그램은 데이터베이스 변경을 위한 더 이상의 응용프로그램 요청을 받아들이지 않습니다.

User Exit 프로그램이 로그 파일을 검색하기 위해 호출되면, 롤 포워드 복구는 ROLLFORWARD DATABASE 유틸리티에 중지가 지정되어 있지 않으면 중지되지 않고 일시중단됩니다. 중지가 지정되어 있지 않으면, 사용자는 문제점을 수정하고 복구를 재개할 수 있습니다.

4. 아카이브 및 검색 조치를 위해 경고 메시지가 0, 4 및 24를 제외한 모든 리턴 코드에 대해 발행됩니다. 경고 메시지는 User Exit 프로그램으로부터 오는 리턴 코드 및 User Exit 프로그램에 제공된 입력 매개변수의 사본을 포함합니다.

User Exit 프로그램이 서버의 운영 체제 명령 처리기에 의해 호출되었기 때문에, 0 이외의 리턴 코드가 운영 체제로부터 리턴될 가능성이 있습니다. 이들 오류 코드는 다시 맵핑되지 않습니다. 해당 오류 코드의 설명은 운영 체제 메시지 도움말 정보를 참조하시기 바랍니다.

OS/2의 오류 조건

BACKUP DATABASE 및 RESTORE DATABASE 유틸리티에 의해 User Exit 프로그램에 의해 리턴된 0 이외의 리턴 코드는 유틸리티가 작업을 실패하고 아무런 재시도도 하지 않도록 합니다. 유틸리티는 일반적인 SQLCODE -2029를 보고합니다. 이 SQLCODE에 대한 메시지 텍스트는 User Exit 프로그램 또는 운영 체제에서 넘겨진 리턴 코드를 표시합니다.

부록D. 다중 데이터베이스 파티션 서버로 명령 발행

파티션된 데이터베이스 시스템에서 인스턴스의 시스템 또는 데이터베이스 파티션 서버(노드)에서 실행되도록 명령을 발행할 수 있습니다. 이 경우, **rah** 명령 또는 **db2_all** 명령을 사용하십시오. **rah** 명령을 사용하면 인스턴스의 시스템에서 실행되도록 명령을 발행할 수 있습니다. 인스턴스의 데이터베이스 파티션 서버에서 명령이 수행되게 하려면, **db2_all** 명령을 수행하십시오. 다음 절에서는 이들 명령의 개요를 제공합니다. 제공되는 정보는 파티션된 데이터베이스 시스템에만 적용됩니다.

주:

1. UNIX 기반 플랫폼에서, 로그인 셸이 Korn 셸 또는 다른 셸이 될 수 있으나, 서로 다른 셸은 특수 문자를 포함하는 명령을 처리하는 방식에서 차이를 나타냅니다.
2. Windows NT에서, **rah** 명령 또는 **db2_all** 명령을 실행하려면 관리자 그룹의 구성원이 되는 사용자 계정으로 로그인해야 합니다.

명령의 영역을 결정하려면, *Command Reference*에서 자세한 내용을 참조하십시오. 이 책에서는 명령이 단일 데이터베이스 파티션 서버에서 수행되는지 또는 이들 전체에서 수행되는지를 보여줍니다. 명령이 하나의 데이터베이스 파티션 서버에서 수행되고 있을 때 명령이 이들 전체에서 수행되게 하려면 **db2_all**을 사용하십시오. 예외는 시스템의 모든 논리 노드(데이터베이스 파티션 서버)에서 수행되는 **db2trc** 명령입니다. 모든 시스템의 모든 논리 노드에서 **db2trc**를 수행하려면 **rah**를 사용하십시오.

명령

하나의 데이터베이스 파티션 서버에서 다른 서버 이후에 연속적으로 명령을 수행할 수 있고 혹은 명령을 병렬로 수행할 수 있습니다. UNIX 기반 플랫폼에서, 명령을 병렬로 수행할 경우, 출력을 버퍼로 보내고 화면에 표시하기 위해 수집할 수 있으며(기본 활동), 명령이 발행된 시스템에 출력을 표시할 수 있습니다. Windows NT에서, 명령을 병렬로 수행하면, 출력은 명령이 발행된 시스템에 표시됩니다.

rah 명령을 사용하려면, 다음을 입력하십시오.

```
rah command
```

db2_all 명령을 사용하려면, 다음을 입력하십시오.

```
db2_all command
```

rah 구문에 대한 도움말을 보려면, 다음을 입력하십시오.

```
rah "?"
```

명령은 다중 명령을 순서대로 수행하는 것을 포함하여 대화식 프롬프트에서 입력할 수 있는 것입니다. UNIX 기반 플랫폼에서, 세미콜론(;)을 사용하여 다중 명령을 분리할 수 있습니다. Windows NT에서, 앰퍼샌드(&)를 사용하여 다중 명령을 분리하십시오. 마지막 명령 다음에는 분리 문자를 사용하지 않도록 하십시오.

다음 예에서는 노드 구성 파일에 지정된 모든 데이터베이스 파티션 서버의 데이터베이스 구성을 변경하기 위해 **db2_all** 명령을 사용하는 방법을 보여줍니다. 다음과 같이 ; 문자가 큰따옴표로 묶여 있으므로, 요청은 동시에 실행됩니다.

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ=100"
```

명령 설명

다음 명령을 사용할 수 있습니다.

명령	설명
rah	모든 시스템에서 명령을 수행합니다.
db2_all	지정한 모든 데이터베이스 파티션 서버에서 명령을 수행합니다.
db2_kill	다중 데이터베이스 파티션 서버에서 실행중인 프로세스를 즉시 중

단하고, 데이터베이스 파티션 서버의 자원을 모두 정리합니다. 이 명령으로 데이터베이스의 불일치를 보정할 수 있습니다. IBM 서비스의 지시가 있는 경우를 제외하고, 이 명령을 발행하지 마십시오.

db2_call_stack

UNIX 기반 플랫폼에서, 모든 데이터베이스 파티션 서버에서 실행 중인 프로세스가 syslog에 call traceback을 기록하게 합니다.

Windows NT에서, 모든 데이터베이스 파티션 서버에서 실행 중인 프로세스가 인스턴스 디렉토리의 Pxxxx.nnn 파일에 call traceback을 기록하는데 여기서, Pxxxx는 프로세스 ID이고 nnn은 노드 번호입니다.

UNIX 기반 플랫폼에서, 이들 명령은 다음과 같이 특정 설정이 내재된 **rah**를 실행합니다.

- 모든 시스템에서 병렬 수행
- 각각 /tmp/\$USER/db2_kill과 /tmp/\$USER/db2_call_stack에서의 버퍼 명령 출력

Windows NT에서, 이들 명령을 사용하면 모든 시스템에서 **rah**가 병렬로 수행됩니다.

수행될 명령 지정

다음 방법으로 명령을 지정할 수 있습니다.

- 명령행에서 매개변수로서
- 매개변수를 지정하지 않을 경우, 프롬프트 상태가 됩니다.

명령에 다음 특수 문자가 들어 있는 경우, 프롬프트 방법을 사용해야 합니다.

| & ; < > () { } [] unsubstituted \$

명령행에 매개변수로 명령을 지정할 때 명령이 위에 나열된 특수 문자를 포함하면, 문자를 큰따옴표로 묶으십시오.

주: UNIX 기반 플랫폼에서, 명령은 프롬프트에 입력한 것처럼 명령 실행기록에 추가됩니다.

명령의 모든 특수 문자는 정상적으로 입력할 수 있습니다(\를 제외하고 따옴표로 묶음). 명령에 \를 삽입하는 경우, 반드시 두 개의 백슬래시(\\)로 입력해야 합니다.

주: UNIX 기반 플랫폼에서, Korn 셸을 사용하고 있지 않으면 명령의 모든 특수 문자가 정상적으로 입력될 수 있습니다(" \ 대체되지 않은 \$ 및 작은따옴표(')의 경우를 제외하고 따옴표로 묶이지 않은 상태로). 명령에 이들 특수 문자 중 하나를 넣을 경우, 반드시 세 개의 백슬래시(\\\)를 앞에 입력해야 합니다. 예를 들어, 명령에 \ 하나를 넣으려면, 네 개의 백슬래시(\\\\)를 입력해야 합니다.

명령에 큰따옴표(")를 사용하려면 \\"과 같이 세 개의 백슬래시를 앞에 사용해야 합니다.

주:

1. UNIX 기반 플랫폼에서, 명령 셸이 작은따옴표가 붙은 문자열 내에 작은따옴표를 입력하는 방식을 제공하지 않는 한, 명령에 작은따옴표(')를 포함시킬 수 없습니다.
2. Windows NT에서, 명령 창이 작은따옴표가 붙은 문자열 내에 작은따옴표를 입력하는 방식을 제공하지 않는 한, 명령에 작은따옴표(')를 포함시킬 수 없습니다.

UNIX 기반 플랫폼에서 병렬에 있는 명령 수행

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.

기본적으로, 명령은 각 시스템에서 순차적으로 수행되지만 특정 접두부 순서를 가지도록 명령에 접두부를 추가함으로써, 백그라운드 rshell을 사용하여 명령이 병렬로 수행되도록 지정할 수 있습니다. rshell이 백그라운드에서 수행되고 각 명령이 원격 시스템의 버퍼 파일에 출력을 추가하는 경우, 이 프로세스는 다음의 두 단계로 출력을 검색합니다.

1. 원격 명령이 완료된 후

2. 일부 프로세스가 수행되고 있는 경우, 이 프로세스가 완료된 후에 발생할 rshell 종료 이후

버퍼 파일 이름은 기본적으로 /tmp/\$USER/rahout 또는 환경 변수 \$RAHBUFDIR/\$RAHBUFNAME에 의해 지정될 수 있습니다.

명령이 동시에 수행되도록 지정할 경우, 기본적으로 이 스크립트는 모든 호스트로 전송되는 명령에 접두부로 추가 명령을 첨부하여 \$RAHBUFDIR 및 \$RAHBUFNAME을 버퍼 파일에 사용할 수 있는지 확인합니다. 그런 다음, \$RAHBUFDIR를 작성합니다. 이 과정이 수행되지 않게 하려면 환경 변수 RAHCHECKBUF=no를 내보내십시오. 디렉토리가 존재하고 사용 가능한 지 알 경우, 이 방법을 통해 시간을 절약할 수 있습니다.

rah를 사용하여 다중 시스템에서 명령을 동시에 수행하기 전에 다음을 확인하십시오.

- 디렉토리 /tmp/\$USER이 각 시스템의 ID에 대해 있는지 확인합니다. 아직 없는 경우, 디렉토리를 작성하려면 다음을 수행하십시오.

```
rah ")mkdir /tmp/$USER"
```

- 다음과 같이 .kshrc(Korn 셸 구문) 또는 .profile에 다음 행을 추가하고 이를 현재 세션에 입력하십시오.

```
export RAHCHECKBUF=no
```

- 원격 명령을 수행하는 각 시스템 ID가 **rah**를 수행하는 ID에 대한 .rhosts 파일에 항목을 갖는지와 **rah**를 수행하는 ID가 원격 명령을 수행하는 각 시스템 ID에 대한 .rhosts 파일에 항목을 갖는지 확인하십시오.

UNIX 기반 플랫폼에서 rah 프로세스 모니터링

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.

아직 수행중인 원격 명령이 있거나 버퍼된 출력이 아직 누적되고 있는 경우, rah에 의해 시작된 프로세스는 다음 활동을 모니터링합니다.

- 명령이 수행되지 않았음을 표시하는 터미널에 메시지 기록
- 버퍼된 출력 검색

정보 메시지는 환경 변수 RAHWAITTIME으로 제어되어 일정 간격으로 기록됩니다. 이를 지정하는 방법에 대해서는 도움말 정보를 참조하십시오. 모든 정보 메시지는 RAHWAITTIME=0을 내보내면 전혀 표시되지 않습니다.

1차 모니터링 프로세스는 (ps 명령에 의해 표시된 대로) 명령 이름이 **rahwait>or** 이 됩니다. 첫번째 정보 메시지에서 프로세스의 pid(프로세스 id)를 가리킵니다. 다른 모든 모니터링 프로세스는 **rah** 스크립트를 실행하는 **ksh** 명령(또는 기호 링크명)으로 나타납니다. 원하는 경우, 다음 명령으로 모든 모니터링 프로세스를 중단할 수 있습니다.

```
kill <pid>
```

여기서 <pid>는 1차 모니터링 프로세서의 프로세스 ID입니다. 신호 수를 지정하지 마십시오. 기본값 15를 남겨 두십시오. 이 값은 원격 명령에는 전혀 영향을 주지 않으나, 버퍼된 출력의 자동 표시를 방해합니다. **rah**의 단일 실행중에 다른 시간에 실행되는 다른 모니터링 프로세스가 둘 이상 있을 수 있습니다. 그러나 현재 설정을 멈추려 할 때에는 더 이상의 프로세스가 시작되지 않습니다.

정기적인 로그인 셸이 Korn 셸(예: /bin/ksh)이 아니면 **rah**를 사용할 수 있으나, 다음의 특수 문자를 포함하는 명령을 입력하는 방법에는 약간 다른 규칙이 적용될 수 있습니다.

```
" unsubstituted $ "
```

자세한 정보는 **rah "?"**. 또한, UNIX 기반 환경에서, 원격 명령을 실행하는 ID에서의 로그인 셸이 Korn 셸이 아니면 **rah**를 실행하는 ID에서의 로그인 셸도 Korn 셸이면 안 됩니다. (**rah**는 원격 ID의 셸이 지역 ID를 근거로 하는 Korn 셸인지에 대해 결정합니다.) 이 셸은 작은따옴표로 묶인 문자열에 대해 대체 또는 특수 처리를 수행해서는 안 됩니다. 있는 그대로 두십시오.

추가 **rah**(모든 호스트 수행) 정보(Solaris 및 AIX 전용)

성능을 향상시키기 위해 **rah**가 대형 시스템에서 **tree_logic**을 사용하도록 확장되었습니다. 즉, **rah**는 목록에 있는 노드 수를 점검하여 해당 수가 임계값을 초과할 경우, 목록의 부속 집합을 구성하여 이들 노드에 순환 호출을 보냅니다. 이들 노드에서, 순환적으로 호출된 **rah**는 목록에 있는 모든 노드에 명령을 보내는 표준 논

리(이제 "leaf-of-tree" 논리)를 따라갈 만큼 목록이 작을 때까지는 같은 논리를 따라갑니다. 임계값은 환경 변수 RAHTREETHRESH에 의해 지정될 수 있거나 기본값으로 15가 됩니다.

물리 노드당 다중 논리 노드 시스템의 경우, db2_all은 특정 물리 노드에 순환 호출을 보내도록 하며, 이 노드는 같은 물리 노드에 있는 다른 논리 노드에 rsh하여 물리 노드간 전송도 감소시킵니다(이 점은 _all에만 적용되며 rah에는 적용되지 않는데, 그 이유는 rah가 항상 구별된 물리 노드에만 송신하기 때문입니다).

이 버전의 rah는 사용 가능한 옵션의 일부 소수 제한사항을 제외하고는 이전 버전(rah.sh_old로 제품에 제공된)과 거의 동일한 구문과 의미를 가집니다.

- 사용자는 ksh(kornshell)를 셸로 사용해야 합니다. 사용자가 다른 셸을 사용할 경우, rah_tree는 경고를 발행하고 트리가 아닌 다른 논리를 사용합니다.
- 명령의 반환를 요청하는 작은따옴표(') 접두부 문자는 유효하지 않으므로 무시됩니다.
- <(all-but-me) 및 >(호스트 색인에 의해 <>를 대체) 옵션이 지정되면, 호스트 색인은 트리 구조가 아닌 rah/db2_all이 대체된 것과 다릅니다.

주: () 및 ## 대체는 구 rah/db2_all에 대해서도 동일하게 작업되어야 합니다.

- 목록에 있는 모든 노드에서 사용자 ID 설정이 일치될 것을 강력히 권장합니다. 예를 들어, rah_tree 명령 또는 db2_tree 명령이 발행된 현재 작업 디렉토리가 모든 노드에 있어야 하고, rah_tree 실행 가능한 프로그램이 모든 노드의 현재 \$PATH(rah_tree 명령이 발행될 때 유효한)에 있어야 하며, rahwaitfor 실행 가능한 프로그램이 이 경로에 있어야 합니다. 노드간의 특정 환경 차이점은 묵인될 수 있지만, 많을 경우에는 묵인할 수 없습니다.
- 실행될 명령은 -o, -b, -d, or -x 문자로 시작하지 말아야 합니다. rah_tree 가 이들 문자를 플래그로 해석하기 때문입니다.
- 병렬 실행을 지정할 때, 호스트가 출력을 리턴하는 순서는 목록순으로 출력을 리턴하는 경향이 있는 비트리 구조 rah가 생성하는 순서와 다를 수 있습니다.
- rah_tree 또는 db2_tree가 호출될 때마다, 위에 기술된 대로, 대상 노드의 수를 임계값과 비교합니다. 임계값은 환경 변수에 의해 지정될 수 있습니다.

```
export RAHTREETHRESH=nn
nn은 임의의 양의 정수
```

또는, 기본값 15가 됩니다.

접두부 순서

접두부 순서는 한 개 이상의 특수 문자로 되어 있습니다. 개입 공백을 하지 말고 명령 문자 바로 앞에 한 개 이상의 접두부 순서를 입력하십시오. 둘 이상의 순서를 지정하기 위해 임의의 순서로 입력을 수행할 수 있으나, 복합어 순서 내의 문자는 순서대로 입력해야 합니다. 접두부 순서를 입력하는 경우, 다음 예에 나오는 것처럼 접두부 순서를 포함하여 전체 명령을 큰따옴표로 묶어야 합니다.

- UNIX 기반 플랫폼에서

```
rah "};ps -F pid,ppid,etime,args -u $USER"
```

- Windows NT에서

```
rah "||db2 get db cfg for sample"
```

접두부 순서는 다음과 같습니다.

순서 목적

I 백그라운드에서 순서대로 명령을 수행합니다.

I& 백그라운드에서 명령을 순서대로 수행하고, 일부 명령이 아직 수행되고 있는 경우에도 모든 원격 명령이 완료된 후에 명령을 종료합니다. 예를 들어, 이 과정은 하위 프로세스(UNIX 기반 플랫폼에서) 또는 백그라운드 프로세스(Windows NT에서)가 여전히 수행되고 있는 경우, 이후에 발생합니다. 이 경우 명령은 별도의 백그라운드 프로세스를 시작하여 명령 종료 이후에 생성된 원격 출력을 검색하고, 이를 원래 시스템에 다시 기록합니다.

주: UNIX 기반 플랫폼에서, &를 지정하면 많은 **rsh** 명령이 요구되므로 성능 저하를 가져옵니다.

II 백그라운드에서 명령을 병렬로 수행합니다.

II& 백그라운드에서 명령을 병렬로 수행하고 위의 I& 경우 대한 설명대로 모든 원격 명령이 완료된 후에 명령을 중단합니다.

주: UNIX 기반 플랫폼에서, &를 지정하면 많은 **rsh** 명령이 요구되므로 성능 저하를 가져옵니다.

; 이는 ||&과 같습니다. 이것은 사용할 수 있는 다른 짧은 양식입니다.

주: UNIX 기반 플랫폼에서, ;를 지정하면 많은 **rsh** 명령이 요구되므로 ||에 비하여 성능 저하를 가져옵니다.

] 명령 수행 전에 사용자의 프로파일이 dot 수행을 사전에 보류합니다.

주: UNIX 기반 플랫폼에서만 사용 가능합니다.

} 명령을 수행하기 전에 \$RAHENV에 이름 지정된 파일(.kshrc)의 dot 실행을 사전에 보류합니다.

주: UNIX 기반 플랫폼에서만 사용 가능합니다.

] } 명령을 실행하기 전에 \$RAHENV(.kshrc)에 이름 지정된 파일 실행 이전에 사용자 프로파일의 dot 실행을 보류합니다.

주: UNIX 기반 플랫폼에서만 사용 가능합니다.

) 사용자의 프로파일 및 \$RAHENV에 이름 지정된 파일이 수행되는 것을 막습니다.

주: UNIX 기반 플랫폼에서만 사용 가능합니다.

' 시스템으로 명령 호출을 반향합니다.

< 이 시스템을 제외한 모든 시스템으로 송신합니다.

<<-nnn<

데이터베이스 파티션 서버 *nnn*을 제외한 전체(노드 번호 *nnn*을 제외하고 db2nodes.cfg의 모든 데이터베이스 파티션 서버, 아래의 주 참조)로 전 송합니다.

<<+nnn<

데이터베이스 파티션 서버 *nnn*(노드 번호가 *nnn*인 db2nodes.cfg의 데이터베이스 파티션 서버, 아래의 주 참조)으로만 전송합니다.

stdin, stdout 및 stderr이 모두 닫혀진 채로 백그라운드에서 원격 명령을 수행합니다. 이 옵션은 백그라운드에서 명령을 수행할 때, 즉 | 또는

;를 포함하는 접두부 순서에서만 유효합니다. 이 옵션을 사용하면 명령을 좀더 빨리(원격 명령이 시작되자마자) 완료할 수 있습니다. **rah** 명령행에 이 접두부 문자를 지정한 경우 명령을 작은따옴표로 묶거나, 큰따옴표로 묶고 앞에 \를 사용하십시오. 예를 들면, 다음과 같습니다.

```
rah ';'mydaemon'
```

또는

```
rah ";\ mydaemon"
```

rah 명령은 백그라운드 프로세스로 수행할 경우, 출력이 리턴되기를 기다리지 않습니다.

> <>를 머신 이름으로 바꾸십시오.

" () 대신 머신 색인으로 사용하고 ## 대신 노드 번호를 사용합니다.

주:

1. 시스템 색인은 데이터베이스 시스템의 머신과 관련된 번호입니다. 다중 논리 노드를 수행하고 있지 않은 경우, 머신에 대한 머신 색인은 노드 구성 파일의 해당 머신에 대한 노드 번호에 해당합니다. 다중 논리 노드 환경에서 머신에 대한 머신 색인을 구하려면, 다중 논리 노드가 수행되는 머신에 대한 중복된 항목을 계산하지 않도록 하십시오. 예를 들어, MACH1이 두 개의 논리 노드에서 수행되고 MACH2도 두 개의 논리 노드에서 수행되는 경우, 노드 구성 파일에서 MACH3에 대한 노드 번호는 5가 됩니다. 그러나, MACH3에 대한 머신 색인은 3이 됩니다.

Windows NT에서, 노드 구성 파일을 편집하지 마십시오. 머신 색인을 구하려면, **db2nlist** 명령을 사용하십시오. 세부사항은 Windows용 *DB2 Enterprise - Extended Edition* 빠른 시작 매뉴얼을 참조하십시오.

2. "가 지정되면 머신 목록에서 중복된 사항이 제거되지 않습니다. 중복된 사항을 제거하려면 511 페이지의 『머신의 목록에서 중복 항목 제거』에서 자세한 내용을 참조하십시오.

<<-nnn< 및 <<+nnn< 접두부 순서를 사용할 경우, *nnn*은 *db2nodes.cfg* 파일의 *nodenum* 값과 일치하는 1, 2 또는 3자릿수 파티션 번호가 됩니다.

주: 접두부 순서는 명령의 일부로 인식해야 합니다. 명령의 일부로 접두부 순서를 지정하는 경우, 접두부 순서를 포함하여 전체 명령을 큰따옴표로 묶어야 합니다.

머신 목록 지정

기본적으로, 머신 목록은 노드 구성 파일, *db2nodes.cfg*로부터 가져옵니다. 다음을 수행하여 파일을 겹쳐쓸 수 있습니다.

- 환경 변수 *RAHOSTFILE*을 내보내거나(UNIX 기반 플랫폼) 이 변수를 설정하여(Windows NT) 머신 목록을 포함하는 파일 경로 이름 지정
- 환경 변수 *RAHOSTLIST*를 내보내거나(UNIX 기반 플랫폼) 이 변수를 설정하여(Windows NT) 공백으로 구분된 일련의 이름으로 목록을 명시적으로 지정

주: 이들 환경 변수가 모두 지정되면 *RAHOSTLIST*가 우선됩니다.

주: Windows NT에서, 노드 구성 파일에 일관되지 않은 내용이 삽입되지 않게 하려면, 이를 수동으로 편집하지 않도록 하십시오. 인스턴스에서 머신 목록을 구하려면 **db2nlist** 명령을 사용하십시오. 세부사항은 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작 매뉴얼을 참조하십시오.

머신의 목록에서 중복 항목 제거

한 머신의 다중 논리 노드(데이터베이스 파티션 서버)에서 *DB2 Enterprise - Extended Edition*를 수행하는 경우, *db2nodes.cfg* 파일에는 해당 머신에 대한 여러 개의 항목이 포함되게 됩니다. 이 상황에서 **rah** 명령에는 사용자가 이 명령을 각 머신에 대해 한번씩만 수행할지 또는 *db2nodes.cfg* 파일에 나열된 각 논리 노드에 대해 한번씩 수행할지 여부를 지정해야 합니다. 머신을 지정하려면 **rah** 명령을 사용하십시오. 논리 노드를 지정하려면 **db2_all** 명령을 사용하십시오.

주: UNIX 기반 플랫폼에서, 머신을 지정하는 경우, **rah**는 보통 다음의 예외 상황을 제외하고 머신 목록에서 중복된 머신을 제거합니다. 논리 노드를 지정할 때, **db2_all**이 명령의 다음 사항을 사전에 보류할 경우는 예외 상황입니다.

```
export DB2NODE=nnn (for Korn shell syntax)
```

여기서, *nnn*은 `db2nodes.cfg` 파일의 해당 행으로부터 가져온 노드 번호이므로, 이 명령의 라우트는 원하는 데이터베이스 파티션 서버로 지정됩니다.

논리 노드를 지정할 때에는 `<<-nnn<`과 `<<+nnn<` 접두부 순서를 사용하여 하나만을 제외한 모든 논리 노드를 포함하도록 목록을 제한하거나 하나의 데이터베이스 파티션 서버만을 지정할 수 있습니다. 카탈로그 노드에서 먼저 명령을 수행하려 하며, 명령이 완료될 때 같은 명령을 다른 모든 데이터베이스 파티션 서버에서 병렬로 수행 가능할 경우, 이 방법을 사용할 수 있습니다. 이 방법은 보통 **db2 restart database** 명령을 수행할 때 필요합니다. 이를 위해 카탈로그 노드의 노드 번호를 알아야 합니다. 접두부 순서에 대해서는 508 페이지의 『접두부 순서』에서 자세한 내용을 참조하십시오.

rah 명령을 사용하여 **db2 restart database**를 실행하면 머신 목록에서 중복된 항목이 제거됩니다. 그러나 “접두부”를 지정하는 경우, 접두부를 사용하는 것이 각 머신이 아닌 각 데이터베이스 파티션 서버에 전송을 수행하는 것으로 간주되므로, 중복된 항목이 제거되지 않습니다.

rah 명령 제어

다음 환경 변수를 사용하여 **rah** 명령을 제어할 수 있습니다.

표 20.

이름	의미	기본값
\$RAHBUFDIR 주: UNIX 기반 플랫폼에서만 사용 가능	버퍼 디렉토리	/tmp/\$USER
\$RAHBUFNAME 주: UNIX 기반 플랫폼에서만 사용 가능	버퍼 파일 이름	rahout
\$RAHOSTFILE (UNIX 기반 플랫폼); RAHOSTFILE (Windows NT)	호스트 목록이 들어 있는 파일	db2nodes.cfg

표 20. (계속)

이름	의미	기본값
\$RAHOSTLIST (UNIX 기반 플랫폼); RAHOSTLIST (Windows NT)	문자열로 된 호스트 목록	\$RAHOSTFILE에서 얻음
\$RAHCHECKBUF 주: UNIX 기반 플랫폼에서만 사용 가능	"no"에 설정되면, 바이패스 검사	설정 안됨
\$RAHSLEEPTIME (UNIX 기반 플랫폼); RAHSLEEPTIME (Windows NT)	스크립트가 병렬로 수행되는 명령으로부터 초 기 출력을 기다리는 초 단위의 시간	db2_kill의 경우는 86400초, 다른 경우는 200초
\$RAHWAITTIME (UNIX 기반 플랫폼); RAHWAITTIME (Windows NT)	Windows NT에서, 원격 작업이 아직 수행중 인 연속되는 점검간의 초 단위 간격 UNIX 기반 플랫폼에서, 원격 작업이 아직 수행중임과 rah: <pid> 기다리는 중... 메시지가 나타나는 것을 확인하는 연속되는 점검간의 초 단위 간격 모든 플랫폼에서, 양의 정수를 지정하십시오. 메시지 출력을 제어하기 위해 사용되는 0으로 시작하는 접두부 값의 경우, RAHWAITIME=045를 내보내십시오. rah 는 작업 완료 상태를 알아내기 위해 이 점검을 따르지 않으므로 낮은 값을 지정할 필요는 없습니다.	45초
\$RAHENV 주: UNIX 기반 플랫폼에서만 사용 가능	\$RAHDOTFILES=E, K, PE 또는 B이면 수행 가능한 파일 이름 지정	\$ENV
\$RAHUSER(UNIX 기반 플랫폼); RAHUSER(Windows NT)	UNIX 기반 플랫폼에서, 원격 명령이 수행될 때의 사용자 ID Windows NT에서, DB2 원격 명령 서비스와 관련된 로그인 계정	\$USER

주: UNIX 기반 플랫폼에서, 원격 셸이 설정한 값(있는 경우)이 아닌 **rah**가 실행되고 있는 \$RAHENV의 값이 사용됩니다.

UNIX 기반 플랫폼의 \$RAHDOTFILES

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.
다음은 접두부 순서가 지정되지 않을 때 수행되는 .파일입니다.

P	.profile
E	\$RAHENV에 이름 지정된 파일(.kshrc)
K	E와 같음
PE	\$RAHENV에 이름 지정된 파일(.kshrc)이 뒤에 나오는 .profile
B	PE와 같음
N	없음

주: 로그인 셸이 Korn 셸이 아닌 경우, 수행할 dot 파일을 Korn 셸 프로세스에서 실행하고 Korn 셸 구문을 준수해야 합니다. 따라서 로그인 셸이 C 셸인 경우, **rah**에 의해 실행되는 명령에 대해 .cshrc 환경변수를 설정하려면 다음과 같이 .cshrc와 상응하는 Korn 셸 INSTHOME/.profile을 작성하고 INSTHOME/.cshrc에 지정합니다.

```
setenv RAHDOTFILES P
```

.cshrc에 상응하는 Korn 셸 INSTHOME/.kshrc를 작성하고 INSTHOME/.cshrc에 지정해야 합니다.

```
setenv RAHDOTFILES E
setenv RAHENV INSTHOME/.kshrc
```

또한, tty가 없는 경우(**rsh**에 의해 호출될 때), .cshrc가 stdout에 쓰지 못하게 해야 합니다. 다음과 같이 stdout에 기록하는 행을 넣어 확인하십시오.

```
if { tty -s } then echo "executed .cshrc";
endif
```

Windows NT에 기본 환경 프로파일 설정

주: 이 절의 정보는 Windows NT에만 적용됩니다.

rah 명령에 대한 기본 환경 프로파일을 설정하려면, 인스턴스 디렉토리에 작성되는 db2rah.env 파일을 사용하십시오. 이 파일은 다음 형식을 가져야 합니다.

```
; This is a comment line
DB2INSTANCE=instancename
DB2DBDFT=database
; End of file
```

rah에 대한 환경을 초기화하는 데 필요한 모든 환경 변수를 지정할 수 있습니다.

UNIX 기반 플랫폼에서 rah 문제점 판별

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.

다음은 **rah**를 수행할 때 발생할 수 있는 문제점을 처리하는 방법입니다.

1. **rah** 정지(또는 오랜 시간 경과)

문제의 원인은 다음과 같습니다.

- **rah**는 출력을 버퍼할 필요가 있으며 RAHCHECKBUF=no를 내보내지 않았음을 알아냈습니다. 따라서, 명령을 수행하기 전에 **rah**는 모든 머신에 명령을 송신하여 버퍼 디렉토리가 있는지 점검하고 없는 경우 이를 작성합니다.
- 명령을 송신하고 있는 하나 이상의 머신이 응답하지 않습니다. **rsh** 명령은 결과적으로 시간종료되지만, 시간종료 간격은 보통 60초로 매우 깁니다.

2. 다음과 같은 메시지가 수신되었습니다.

- 올바른지 않은 로그인
- 사용권한 거부

머신 중 하나가 `.hosts` 파일에 적절히 정의된 **rah**를 수행하는 ID를 가지고 있지 않거나, **rah**를 수행하는 ID가 `.rhosts` 파일에 적절히 정의된 머신 중 하나를 가지고 있지 않습니다.

3. 명령이 머신에서 예상된 경과 시간 내에 수행되어 완료되어도, 백그라운드 rshell을 사용하여 명령을 병렬로 수행하는 경우에는 **rah**는 이를 감지하고 셸 프롬프트로 올리는 데 많은 시간이 필요합니다.

rah를 수행하는 ID가 `.rhosts` 파일에 적절히 정의된 머신 중 하나를 가지고 있지 않습니다.

4. 셸 명령행으로부터 수행할 때 **rah**가 제대로 수행되더라도, 다음과 같이 rsh를 사용하여 원격으로 **rah**를 수행하면

```
rsh somewher -l $USER db2_kill
```

rah는 완료될 수 없습니다.

이것은 정상적입니다. **rah**는 백그라운드 모니터링 프로세스를 시작하며 이 프로세스를 빠져나간 후에도 계속 수행됩니다. 수행하고 있는 명령과 관련된 모든 프로세스를 자체 종료할 때까지 프로세스는 정상적으로 지속됩니다. **db2_kill**의 경우, 이것은 모든 데이터베이스 관리 프로그램의 종료를 의미합니다. 명령이 **rahwait>or** 및 `kill <process_id>`인 프로세스를 찾아 모니터링 프로세스를 종료할 수 있습니다. 신호 수를 지정하지 마십시오. 대신, 기본값 15를 사용하십시오.

- 다중 **rah** 명령이 같은 \$RAHUSER하에서 발행되어 있지 않을 때, **rah**로부터의 출력이 제대로 표시되지 않거나 **rah**가 \$RAHBUFNAME이 존재하지 않음을 제대로 보고하지 않습니다.

이것은 **rah**의 다중 동시 실행이 출력을 버퍼링하기 위해 동일한 버퍼 파일(예 : \$RAHBUFDIR/\$RAHBUFNAME)을 사용하려 하기 때문입니다. 이 문제점을 막으려면, 각 동시 **rah** 명령에 대해다음 ksh에서처럼 \$RAHBUFNAME을 사용하십시오.

```
export RAHBUFNAME=rahout
rah ";$command_1" &
export RAHBUFNAME=rah2out
rah ";$command_2" &
```

또는, 셸이 다음과 같은 고유한 이름을 자동으로 선택하게 만드는 방법을 사용하십시오.

```
RAHBUFNAME=rahout.$$ db2_a11 "....."
```

어떤 방법을 사용하든지 디스크 공간에 한계가 있을 경우, 어떤 지점에서 버퍼 파일을 정리해야 하는지 확인하십시오. **rah**는 버퍼 파일을 지운 다음, 같은 버퍼 파일을 지정하는 다음 번에 기존 파일을 재사용해도 실행 마지막에 버퍼 파일을 지웁니다.

- 다음을 입력하면

```
rah '"print from ()'
```

다음 메시지가 수신됩니다.

```
ksh: syntax error at line 1 : (' unexpected
```


() 및 ## 대체를 위한 전제조건:

- **rah**가 아닌 **db2_all**을 사용하십시오.
- RAHOSTFIL을 내보내거나 /sql1lib/db2nodes.cfg 파일을 기본 파일로 지정하여 RAHOSTFILE이 사용되는지 확인하십시오. 이러한 전제조건이 없으면 **rah**는 있는 그대로 () 및 ##을 남겨둡니다. 명령 **print from ()**이 유효하지 않으므로 오류가 수신됩니다.

명령을 병렬로 수행할 때 성능상 추가 정보를 얻으려면, &에서 제공한 함수가 필요하지 않는 한 |&보다는 |를, ||& 보다는 ||를 사용하십시오. &를 지정하면, 더 많은 **rsh** 명령이 필요하므로 성능이 저하됩니다.

부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법

Windows NT를 설치할 때, 두 개의 관리자 사용자 이름을 작성할 수 있습니다.

- 하나는 『Administrator』라고 합니다.
- 다른 하나는 사용자가 선택하는 이름입니다. 그 이름은 관리자 권한을 가지고 있어야 하며 DB2의 이름 지정 규칙을 따라야 합니다. DB2 이름 지정 규칙에 대해서는 443 페이지의 『부록A. 이름 지정 규칙』에서 자세한 내용을 참조하십시오.

| 사용자는 지역 머신에 로그인할 수 있거나, 머신이 Windows NT 도메인에 설치
| 될 경우 도메인에 로그인할 수 있습니다. Windows NT용 DB2는 이 옵션 둘다
| 를 지원합니다. 사용자를 인증하기 위해, DB2는 먼저 지역 머신을 점검한 후 현
| 재 도메인에 대한 도메인 제어를 점검하고, 마지막으로 도메인 제어기에 알려진
| 신뢰 도메인을 점검합니다.

Windows NT용 DB2가 작동하는 방법을 보여주기 위해, DB2 인스턴스가 서버 인증을 요구한다고 가정하십시오. 구성은 다음과 같습니다.

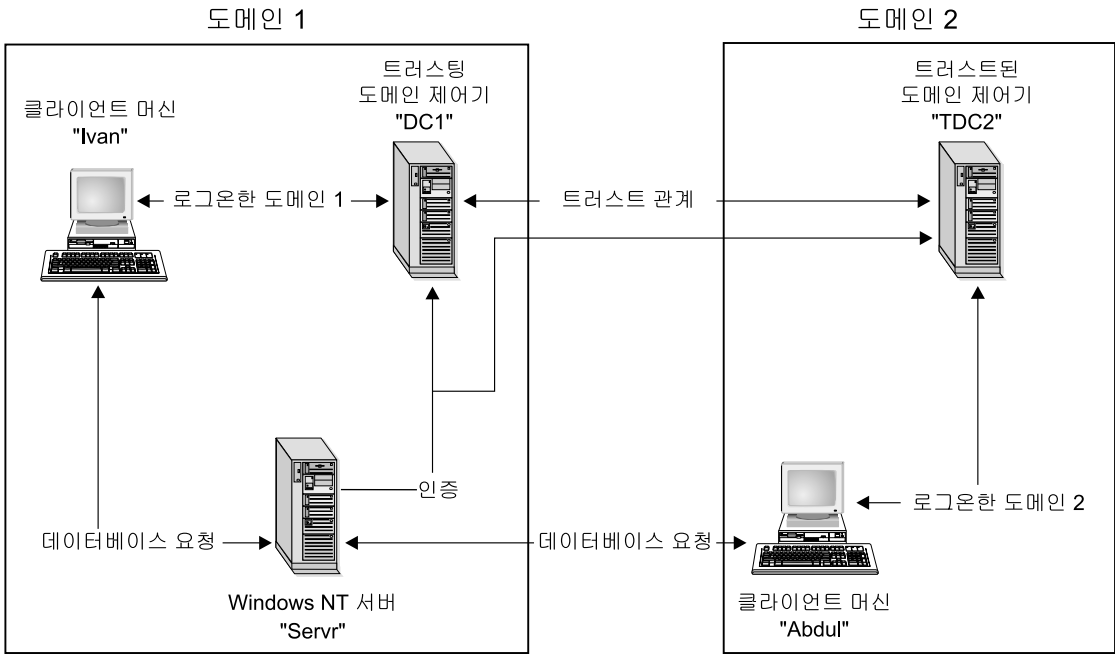


그림 25. Windows NT 도메인 사용 인증

각 머신은 클라이언트 머신에서 Windows 95가 수행되지 않는 한, 보안 데이터베이스인 보안 액세스 관리(SAM) 데이터베이스가 있습니다. Windows 95 머신에는 SAM 데이터베이스가 없습니다. DC1은 클라이언트 머신, Ivan 및 Windows NT 서버인 Servr용 DB2가 등록되는 도메인 제어기입니다. TDC2는 DC1과 클라이언트 머신인 Abdul용 신뢰 도메인으로, TDC2 도메인의 구성원입니다.

서버 인증에 대한 샘플 시나리오

1. Abdul은 TDC2 도메인에 로그인합니다(즉, TDC2 SAM 데이터베이스에 알려줍니다).
2. 그런 다음, Abdul을 SRV3에 상주하기 위해 카탈로그에 등록되는 DB2 데이터베이스에 연결합니다.

```
db2 connect to remotedb user Abdul using fredpw
```

3. SRV3은 Abdul이 알려진 곳을 판별합니다. 이 정보를 찾기 위해 사용되는 API는 신뢰 도메인을 시도하기 전에 먼저 지역 머신(SRV3)을 검색한 후 도메인

제어기(DC1)를 탐색합니다. 사용자 이름 Abdul이 TDC2에서 발견됩니다. 이 검색 순서에서는 사용자 또는 그룹에 대해 하나의 이름 공간을 필요로 합니다.

4. 그런 다음, SRV3은 다음을 수행합니다.
 - a. TDC2에 대해 사용자 이름과 암호의 유효성을 확인합니다.
 - b. TDC2에 요청하여 Abdul이 관리자인지 찾습니다.
 - c. TDC2에 요청하여 Abdul의 모든 그룹을 열거합니다.

클라이언트 인증 및 Windows NT 클라이언트 머신에 대한 샘플 시나리오

1. 관리자인 Dale은 SRV3에 로그인하여 데이터베이스 인스턴스에 대한 인증을 클라이언트로 변경합니다.

```
db2stop myinst
db2 update dbm cfg using authentication client
db2start myinst
```

2. Windows NT 클라이언트 머신에서, Ivan은 DC1 도메인에 로그인합니다(즉, DC1 SAM 데이터베이스에 알려줍니다).
3. 그런 다음, Ivan을 SRV3에 상주하기 위해 카탈로그에 등록되는 DB2 데이터베이스에 연결합니다.

```
DB2 CONNECT to remotedb user Ivan using johnpw
```

4. Ivan의 머신은 사용자 이름과 암호의 유효성을 확인합니다. 이 정보를 찾기 위해 사용되는 API는 신뢰 도메인을 시도하기 전에 먼저 지역 머신(Ivan)을 검색한 후 도메인 제어기(DC1)를 탐색합니다. 사용자명 Ivan이 DC1에서 발견됩니다.
5. 그러면, Ivan의 머신은 DC1에 대해 사용자 이름과 암호의 유효성을 확인합니다.
6. 그런 다음, SRV3은 다음을 수행합니다.
 - a. Ivan이 알려진 곳을 판별합니다.
 - b. DC1에 요청하여 Ivan이 관리자인지 찾습니다.
 - c. DC1에 요청하여 Ivan의 모든 그룹을 나열합니다.

주: DB2 데이터베이스에 연결을 시도하기 전에, DB2 보안 서비스가 시작되었는지 확인하십시오. 보안 서비스는 DB2에 의해 설치되며 Windows NT 서비스로서 수행하도록 설정됩니다. 단, 이 서비스는 자동으로 시작되지 않습니다. DB2 보안 서비스를 시작하려면, NET START DB2NTSECSERVER 명령을 입력하십시오.

클라이언트 인증 및 Windows 95 클라이언트 머신에 대한 샘플 시나리오

1. 관리자인 Dale은 SRV3에 로그인하여 데이터베이스 인스턴스에 대한 인증을 클라이언트로 변경합니다.

```
db2stop myinst
db2 update dbm cfg using authentication client
db2start myinst
```

2. Windows 95 클라이언트 머신에서, Ivan은 DC1 도메인에 로그인합니다(즉, DC1 SAM 데이터베이스에 알려줍니다).
3. 그런 다음, Ivan을 SRV3에 상주하기 위해 카탈로그에 등록되는 DB2 데이터베이스에 연결합니다.

```
db2 connect to remotedb user Ivan using johnpw
```

4. Ivan의 Windows 95 머신은 사용자 이름과 암호의 유효성을 확인할 수 없습니다. 그러므로, 사용자 이름과 암호는 유효한 것으로 가정합니다.
5. 그런 다음, SRV3은 다음을 수행합니다.
 - a. Ivan이 알려진 곳을 판별합니다.
 - b. DC1에 요청하여 Ivan이 관리자인지 찾습니다.
 - c. DC1에 요청하여 Ivan의 모든 그룹을 나열합니다.

주: Windows 95 클라이언트는 주어진 사용자 이름 및 암호의 유효성을 확인할 수 없으므로, Windows 95 환경에서의 클라이언트 인증은 안전하지 못합니다. Windows 95 머신이 Windows NT 보안 제공자에 대한 액세스 권한을 가지고 있으면, 일부 보안 측정은 확인된 연결 프로그램 로그온에 대해 Windows 95를 구성하여 부과될 수 있습니다. 이 방법으로 Windows 95 시스템을 구성하는 방법에 관한 세부사항은 Windows 95용 Microsoft 문서를 참조하십시오.

DB2는 또한 전역 그룹을 지원합니다. 전역 그룹을 사용하려면, 보안 서버에 있는 지역 그룹 내에 전역 그룹을 포함시켜야 합니다. 사용자가 구성원으로 있는 모든 그룹을 DB2가 열거할 때, 사용자가 간접적으로 구성원인 지역 그룹도 나열합니다 (자체가 하나 이상의 지역 그룹 구성원인 전역 그룹의 효력으로).

DB2로 백업 도메인 제어기 사용

DB2에 대해 사용하는 서버가 백업 도메인 제어기로도 작동할 경우, DB2를 백업 도메인 제어기를 사용하도록 구성하면 DB2를 향상시키고 네트워크 트래픽을 줄일 수 있습니다.

db2dmnbckctlr 레지스트리 값을 설정하여 백업 도메인 제어기를 DB2로 지정합니다.

DB2 서버가 백업 도메인 제어기인 도메인의 이름을 알면, 다음을 사용하십시오.

```
db2dmnbckctlr=DOMAIN_NAME
```

*DOMAIN_NAME*은 대문자여야 합니다.

DB2가 백업 도메인 제어기인 지역 머신에 대한 도메인을 판별하도록 하려면, 다음을 사용하십시오.

```
DB2DMNBCKCTLR=?
```

주: DB2는 기본적으로 기존의 백업 도메인 제어기를 사용하지 않습니다. 백업 도메인 제어기가 기본 도메인 제어기에 대해 동기화에서 벗어날 수 있어서 보안이 노출될 수 있기 때문입니다. 도메인 제어기는 기본 도메인 제어기의 보안 데이터베이스가 갱신될 때 동기화에서 벗어나지만, 변경사항은 백업 도메인 제어기에 전파되지 않습니다. 이러한 상황은 네트워크 지연이 있거나 컴퓨터 브라우저 서비스가 작동하지 않을 경우에 발생할 수 있습니다.

Windows NT용 DB2 사용자 인증

운영 체제 인증 방식 때문에 사용자 인증은 Windows NT 사용자에게 문제점을 야기할 수 있습니다. 이 절은 Windows NT용 DB2 아래에서 사용자 인증에 대한 고려사항을 설명합니다.

- 『사용자 이름 및 그룹 이름 제한사항』
- 『Windows NT용 DB2 보안 서비스』
- 『백업 도메인 제어기에서 DB2 설치』
- 526 페이지의 『그룹 및 도메인 보안 인증』

사용자 이름 및 그룹 이름 제한사항

다음은 이 환경에서의 제한사항입니다.

- 사용자 이름은 DB2 내에서 30자로 제한됩니다. 그룹 이름은 8자로 제한됩니다.
- Windows NT 아래의 사용자 이름은 대소문자가 구분되지 않지만, 암호는 대소문자가 구분됩니다.
- 사용자 이름 및 그룹 이름은 대문자 및 소문자의 결합일 수 있습니다. 그러나, DB2 내에서 사용될 때 보통 대문자로 변환됩니다. 예를 들어, 데이터베이스에 연결하고 schema1.table1 테이블을 작성하는 경우, 이 테이블은 데이터베이스 내에서 SCHEMA1.TABLE1로서 저장됩니다. 소문자 오브젝트 이름을 사용하려는 경우, 오브젝트 이름을 따옴표로 묶어 명령행 프로세서에서 명령을 발행하거나, 썬드 파티 ODBC 프론트엔드 도구를 사용하십시오.

Windows NT용 DB2 보안 서비스

DB2 Universal Database에서 사용자 이름 및 암호의 인증을 DB2 시스템 제어기로 통합했습니다. 보안 서비스는 인증 CLIENT용으로 구성되는 서버로 클라이언트를 연결할 때에만 요구됩니다.

백업 도메인 제어기에서 DB2 설치

Windows NT 환경에서 사용자는 기본 또는 백업 제어기 중 하나에서 인증될 수 있습니다. 이 기능은 하나의 중앙 기본 도메인 제어기와 각 사이트에 하나 이상의 백업 도메인 제어기(BDC)가 있는 큰 분산 LAN에서 매우 중요합니다. 그러면 사

용자는 인증을 위해 기본 도메인 제어기(PDC)에 호출을 요청하는 대신 관련 사이트의 백업 도메인 제어기에서 인증될 수 있습니다.

이 경우 백업 도메인 제어기를 갖는 이점은 사용자가 더 빠르게 인증되며 LAN은 BDC가 없었을 때처럼 혼잡하지 않습니다.

다음 조건 아래에서 BDC에서 인증을 발생할 수 있습니다.

- Windows NT용 DB2 서버는 백업 도메인 제어기에 설치됩니다.
- DB2DMNBCKCTRL 프로파일 레지스트리 변수가 제대로 설정되어 있습니다. DB2DMNBCKCTRL 프로파일 레지스트리 변수가 설정되지 않거나 공백으로 설정되어 있는 경우, Windows NT용 DB2는 기본 도메인 제어기에서 인증을 수행합니다.

DB2DMNBCKCTRL용으로 선언된 유일하게 유효한 설정은 『?』 또는 도메인 이름입니다.

DB2DMNBCKCTRL 프로파일 레지스트리 변수가 물음표로 설정되면 (DB2DMNBCKCTRL=?) Windows NT용 DB2는 다음과 같은 조건 아래에서 백업 도메인 제어기에서 인증을 수행합니다.

- cachedPrimaryDomain은 이 머신이 속하는 도메인의 이름으로 설정된 레지스트리 값입니다. **HKEY_LOCAL_MACHINE—> Software—> Microsoft—> Windows NT—> Current Version—> WinLogon** 아래에서 이 설정을 찾을 수 있습니다.
- 서버 관리 프로그램은 백업 도메인 제어기를 사용중이며 사용 가능한 것으로 표시합니다. 즉, 이 머신의 아이콘은 회색으로 표시되지 않습니다.
- DB2 Windows NT 서버의 레지스트리는 시스템이 지정된 도메인에서 백업 도메인 제어기임을 나타냅니다.

정상적인 환경에서 설정 DB2DMNBCKCTRL=?은 작동하지만 모든 환경에서 작동하는 것은 아닙니다. 도메인에서 서버에 대해 제공된 정보는 동적이며, 컴퓨터 브라우저는 이 정보를 정확하고 현재의 것으로 유지하기 위해 수행 중이어야 합니다. 큰 LAN은 수행 중인 컴퓨터 브라우저가 아닐 수 있으므로 서버 관리 프로그램의 정보는 현재의 것이 아닐 수 있습니다. 이 경우, 백업 도메인 제어기에서 Windows NT용 DB2가 인증하도록 알리는 두번째 메소드가 있습니다.

DB2DMNBCKCTLR=xxx를 설정하십시오. 여기서 xxx는 DB2 서버의 Windows NT 도메인 이름입니다. 이 설정을 사용하여, 인증은 다음과 같은 조건에 근거하여 백업 도메인 제어기에서 발생할 수 있습니다.

- cachedPrimaryDomain은 이 머신이 속하는 도메인의 이름으로 설정된 레지스트리 값입니다. **HKEY_LOCAL_MACHINE—> Software—> Microsoft—> Windows NT—> Current Version—> WinLogon** 아래에서 이 설정을 찾을 수 있습니다.
- 머신은 지정된 도메인의 백업 도메인 제어기로서 구성됩니다. 기계가 백업 도메인 제어기로서 설정되지만 또다른 도메인용인 경우, 이 설정은 오류를 발생시킵니다.

그룹 및 도메인 보안 인증

Windows NT용 DB2는 다음과 같은 유형의 그룹을 지원합니다.

- 지역 그룹
- 전역 그룹
- 지역 그룹과 전역 그룹의 구성원

Windows NT용 DB2는 사용자가 찾은 보안 데이터베이스를 사용하여 사용자가 구성원인 지역 및 전역 그룹을 열거합니다. DB2 Universal Database는 사용자 계정이 있는지 여부에 관계없이 DB2가 설치된 지역 Windows NT 서버에서 그룹 열거가 발생하게 하는 겹쳐쓰기를 제공합니다. 이 겹쳐쓰기는 다음과 같은 명령을 사용하여 이루어집니다.

- 전역 설정의 경우

```
db2set -g DB2_GROUP_LOOKUP=local
```

- 인스턴스 설정의 경우

```
db2set -i DB2_GROUP_LOOKUP=local
```

설정된 모든 DB2 프로파일 변수를 보려면, 다음을 입력하십시오.

```
db2set -all
```

Windows NT용 DB2가 도메인 보안으로 작업하게 하려면, 권한 및 특권을 지역 그룹에 부여해야 합니다. 지역 및 전역 그룹 내의 사용자 이름은 지역 또는 전역 그룹과 동일한 도메인에 정의되어야 제대로 인증됩니다.

DB2_GRP_LOOKUP 프로파일 레지스트리 변수가 지역으로 설정된 경우, DB2는 지역 머신에서만 사용자를 찾으려고 합니다. 사용자가 지역 머신에 없거나, 지역 또는 전역 그룹의 구성원으로서 정의되지 않은 경우, 인증에 실패합니다. DB2는 도메인의 또다른 머신이나 도메인 제어기에서 사용자를 찾으려고 시도하지 않습니다.

DB2_GRP_LOOKUP 프로파일 레지스트리 변수가 설정되지 않은 경우

1. DB2는 우선 동일한 머신에서 사용자를 찾으려고 시도합니다.
2. 사용자 이름이 지역적으로 정의된 경우, 사용자는 지역적으로 인증됩니다.
3. 지역적으로 사용자를 찾지 못한 경우, DB2는 도메인에서 사용자 이름을 찾은 다음 신뢰 도메인에서 찾으려고 시도합니다.

다음 예는 Windows NT용 DB2가 어떻게 도메인 보안을 지원하는가를 나타냅니다. 이 첫번째 예에서, 사용자 이름 및 지역 그룹은 동일한 도메인에 있으므로 연결이 동작합니다. 두번째 예에서, 사용자 이름 및 지역 또는 전역 그룹은 다른 도메인에 있으므로 연결이 동작하지 않습니다.

성공적인 연결의 예: 사용자 이름 및 지역 또는 전역 그룹은 동일한 도메인에 있으므로 다음 시나리오에서 연결이 작동합니다.

사용자 이름 및 지역 또는 전역 그룹은 데이터베이스 서버가 수행중이지만 서로 동일한 도메인에 있어야 하는 도메인에 정의될 필요가 없음을 기억하십시오.

표 21. 도메인 제어기를 사용한 정상적인 연결

Domain1	Domain2
Domain2와의 신뢰 관계가 있습니다.	<ul style="list-style-type: none"> • Domain1과의 신뢰 관계가 있습니다. • 지역 또는 전역 그룹 grp2가 정의됩니다. • 사용자 이름 id2가 정의됩니다. • 사용자 이름 id2는 grp2의 일부입니다.
DB2 서버는 이 도메인에서 수행합니다. 다음과 같은 DB2 명령이 발행됩니다. REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2	

표 21. 도메인 제어를 사용한 정상적인 연결 (계속)

Domain1	Domain2
지역 또는 전역 도메인이 검색되지만 id2가 없습니다. 도메인 보안이 검색됩니다.	
	사용자 이름 id2가 이 도메인에 있습니다. DB2는 이 사용자 이름에 대한 추가적인 정보를 얻습니다(즉, 이 사용자 이름은 grp2의 일부입니다).
사용자 이름 및 지역 또는 전역 그룹은 동일한 도메인에 있으므로 연결이 작동합니다.	

실패한 연결의 예: 사용자 이름이 지역 또는 전역 그룹이 정의된 도메인이 아닌 다른 도메인에 정의되어 있으므로 다음과 같은 시나리오에서는 연결이 작동하지 않습니다.

표 22. 도메인 제어를 사용한 실패 연결

Domain1	Domain2
Domain2와의 신뢰 관계가 있습니다.	<ul style="list-style-type: none"> Domain1과의 트러스트 관계가 있습니다. 지역 또는 전역 그룹 grp2가 정의됩니다.
<ul style="list-style-type: none"> 전역 그룹 grp1이 정의됩니다. 사용자 이름 id1이 정의됩니다. 사용자 이름 id1은 grp1의 일부입니다. 	
	Domain1\grp1은 grp2의 일부입니다.
DB2 서버는 이 도메인에서 수행합니다. 다음과 같은 DB2 명령이 발행됩니다. <pre>REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2</pre>	
지역 또는 전역이 검색되며 id1을 찾았습니다. DB2는 이 사용자 이름의 정보를 얻습니다(즉, 사용자 이름 id1은 grp1의 일부이며 그룹 grp1은 Domain2\grp2의 일부입니다).	
	그룹 grp2는 이 도메인에 있습니다.

| 표 22. 도메인 제어를 사용한 실패 연결 (계속)

Domain1	Domain2
지역 또는 전역 그룹은 Domain2에 있으며 실제 사용자 이름은 Domain1에 정의되어 있으므로 연결이 작동하지 않습니다. 다음과 같은 명령이 대신 발행된 경우 연결이 작동합니다. GRANT CONNECT ON db TO GROUP grp1	

부록F. Windows NT 성능 모니터 사용

Windows NT용 DB2 사용자가 사용할 수 있는 두가지 성능 모니터가 있습니다.

- **DB2 성능 모니터**

DB2 성능 모니터는 DB2 및 DB2 Connect에만 관련된 스냅샷 및 이벤트 데이터를 제공합니다(자세한 정보는 제어 센터에서 **도움말** 누름 버튼을 클릭하여 '시작하기' 온라인 도움말을 참조하십시오).

- **Windows NT 성능 모니터**

Windows NT 성능 모니터는 이 시스템에 등록된 모든 성능 데이터 제공자로부터 정보를 검색하여 데이터베이스 및 시스템 성능을 모니터링할 수 있게 합니다. Windows NT도 다음과 같은 머신 조작의 모든 양상에 대해 성능 정보 데이터를 제공합니다.

- CPU 사용
- 메모리 활용
- 디스크 활동
- 네트워크 활동

Windows NT 성능 모니터로 DB2 등록

설치 프로그램은 자동으로 DB2를 Windows NT 성능 모니터에 등록합니다.

Windows NT 성능 모니터에 액세스할 수 있는 DB2 및 DB2 Connect 성능 정보를 작성하려면, Windows NT용 DB2 성능 카운터에 대한 DLL을 등록해야 합니다. 이는 Win32 성능 API를 사용하는 다른 모든 Windows NT 응용프로그램도 성능 데이터를 확보할 수 있게 합니다.

Windows NT 성능 모니터에 Windows NT용 DB2 성능 카운터 DLL(DB2Perf.DLL)를 설치 및 등록하려면, 다음을 입력하십시오.

```
db2perfi -i
```

DLL을 등록하면 레지스트리의 서비스 옵션에 새로운 키도 작성합니다. 한 항목은 카운터 지원을 제공하는 DLL의 이름을 지정하고, 다른 세 항목은 이 DLL에 제공된 함수의 이름을 제공합니다. 이들 함수는 다음과 같습니다.

- 열기

프로세스에서 DLL이 시스템에 의해 최초로 로드될 때 호출됩니다.

- 수집

DLL에서 성능 정보를 요청하기 위해 호출됩니다.

- 닫기

DLL이 로드 해제될 때 호출됩니다.

DB2 성능 정보에 원격 액세스 작동

Windows NT용 DB2 워크스테이션이 다른 Windows NT 머신에 네트워크로 연결되어 있을 경우, 이 절에 설명되어 있는 기능을 사용할 수 있습니다.

다른 Windows NT용 DB2 머신에서 Windows NT 성능 오브젝트를 보기 위해서는 DB2에 관리자 사용자 이름과 암호를 등록해야 합니다(기본 Windows NT 성능 모니터 사용자 이름인 **SYSTEM**은 DB2 예약어이므로 사용될 수 없습니다). 이름을 등록하려면, 다음을 입력하십시오.

```
db2perfr -r username password
```

주: 사용된 username은 이름 지정 규칙에 따라야 합니다.

사용자 이름과 암호 데이터는 관리자와 SYSTEM 계정만 액세스할 수 있는 보안 하에 레지스트리에 있는 키에 보관됩니다. 관리자 암호를 레지스트리에 저장할 때 보안 문제점을 피하기 위해 데이터를 암호화합니다.

주:

1. 사용자 이름과 암호 조합이 DB2에 등록되고 나면, 성능 모니터의 지역 인스턴스 조차 해당 사용자 이름과 암호 사용에 대해 명시적으로 로그합니다. 이는 DB2에 등록된 사용자 이름 정보가 일치하지 않을 경우, 성능 모니터의 지역 세션이 DB2 성능 정보를 표시하지 않는다는 것을 의미합니다.

2. 사용자 이름과 암호 조합은 Windows NT 보안 데이터베이스에 저장된 사용자 이름과 암호 값이 일치하도록 유지보수되어야 합니다. Windows NT 보안 데이터베이스에서 사용자 이름 또는 암호가 변경되면, 원격 성능 모니터시 사용되는 사용자 이름과 암호 조합도 재설정되어야 합니다.
3. 재등록하려면, 다음을 입력하십시오.

```
db2perfr -u
```

DB2 및 DB2 Connect 성능 값 표시

성능 모니터를 사용하여 DB2 및 DB2 Connect 성능 값을 표시하려면, 추가 대상(Add to) 상자로부터 값을 표시하고자 하는 성능 카운터를 선택하기만 하십시오. 이 상자는 성능 데이터를 제공하는 성능 오브젝트 목록을 표시합니다. 한 오브젝트를 선택하여 오브젝트가 제공하는 카운터 목록을 보십시오.

성능 오브젝트는 다중 인스턴스도 가질 수 있습니다. 예를 들어, LogicalDisk 오브젝트는 『% 디스크 읽기 시간』 및 『디스크 바이트/초』와 같은 카운터를 제공하며, 머신에 있는 『C:』 및 『D:』를 포함하는 각 논리 드라이브에 대한 인스턴스도 소유합니다.

Windows NT는 다음과 같은 성능 오브젝트를 제공합니다.

- **DB2 데이터베이스 관리 프로그램**

이 오브젝트는 단일 Windows NT 인스턴스에 대한 일반 정보를 제공합니다. 모니터중인 DB2 인스턴스는 오브젝트 인스턴스로 나타납니다.

실용적 및 성능상의 이유로, 한 번에 하나의 DB2 인스턴스로부터 성능 정보만 가져올 수 있습니다. 성능 모니터가 표시하는 DB2 인스턴스는 성능 모니터 프로세스의 db2instance 레지스트리 변수에 의해 관리됩니다. 동시에 수행하는 다중 DB2 인스턴스를 가지고 있으며 둘 이상으로부터 성능 정보를 보기 원한다면, 모니터될 각 DB2 인스턴스에 대한 적당한 값에 db2instance를 설정하여, 성능 모니터의 별도 세션을 시작해야 합니다.

파티션된 데이터베이스 시스템이 수행될 때, 한 번에 하나의 DB2 인스턴스로부터 성능정보만 가져올 수 있습니다. 기본값으로, 기본 노드의 성능 정보(즉, 논

리 포트 0을 가진 노드)가 표시됩니다. 또다른 노드의 성능 정보를 보려면, 모니터되는 노드의 노드 번호에 설정된 DB2NODE 환경 변수로 성능 모니터의 각 세션을 시작해야 합니다.

- **DB2 데이터베이스**

이 오브젝트는 특정 데이터베이스에 대한 정보를 제공합니다. 정보는 현재 사용 중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 응용프로그램**

이 오브젝트는 특정 DB2 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 활동중인 각 DB2 응용프로그램에 대해 사용 가능합니다.

- **DB2 DCS 데이터베이스**

이 오브젝트는 특정 DCS 데이터베이스에 대한 정보를 제공합니다. 정보는 현재 사용중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 DCS 응용프로그램**

이 오브젝트는 특정 DB2 DCS 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 사용중인 각 DB2 DCS 응용프로그램에 대해 사용 가능합니다.

이들 오브젝트 중 어느 것이 Windows NT 성능 모니터에 의해 나열될 것인지는 무엇이 Windows NT 머신에 설치되어 있는지 및 어느 응용프로그램이 사용중인지에 따라 다릅니다. 예를 들어, DB2 UDB가 설치되어 있고 데이터베이스 관리 프로그램이 시작되어 있으면, DB2 데이터베이스 관리 프로그램 오브젝트가 나열됩니다. 머신상에서 일부 DB2 데이터베이스와 응용프로그램도 현재 사용중이라면, DB2 데이터베이스 및 DB2 응용프로그램 오브젝트가 마찬가지로 나열됩니다. Windows NT 시스템을 DB2 Connect 게이트웨이로 사용중이고 일부 DCS 데이터베이스와 응용프로그램이 현재 사용중이면, DB2 DCS 데이터베이스와 DB2 DCS 응용프로그램 오브젝트가 나열됩니다.

원격 DB2 성능 정보 액세스

DB2 성능 정보에 대한 원격 액세스 작동에 대해서는 이미 앞의 절에서 언급되었습니다. 추가 대상(Add to) 상자에서 다른 컴퓨터를 선택하여 모니터하십시오. 해당 컴퓨터에서 사용 가능한 모든 성능 오브젝트의 목록이 표시됩니다.

원격 컴퓨터의 DB2 성능 오브젝트를 모니터하기 위해서는 해당 컴퓨터에 설치되어 있는 DB2 UDB 또는 DB2 Connect 코드의 레벨이 버전 6 이상이어야 합니다.

DB2 성능 값 재설정

응용프로그램이 DB2 모니터 API를 호출할 때, 리턴된 정보는 일반적으로 DB2 서버가 시작된 이후에 누적된 값입니다. 그러나, 종종 다음을 수행하는 데 유용합니다.

- 성능 값 재설정
- 테스트 수행
- 값을 다시 재설정
- 테스트 재수행

데이터베이스 성능 값을 재설정하려면, **db2perf** 프로그램을 사용하십시오. 다음을 입력하십시오.

```
db2perf
```

기본적으로, 이 명령은 사용중인 모든 DB2 데이터베이스에 대한 성능 값을 재설정합니다. 또한, 재설정하려는 데이터베이스의 목록을 지정할 수도 있습니다. DCS 데이터베이스에 대한 성능 값이 재설정되어야 한다는 것을 지정하기 위해 -d 옵션을 사용할 수도 있습니다. 예를 들면, 다음과 같습니다.

```
db2perf
db2perf dbalias1 dbalias2 ... dbaliasn

db2perf -d
db2perf -d dbalias1 dbalias2 ... dbaliasn
```

첫번째 예는 사용중인 모든 DB2 데이터베이스에 대한 성능 값을 재설정합니다. 그 다음 예는 특정 DB2 데이터베이스에 대한 값을 재설정합니다. 세 번째 예는 사용중인 모든 DB2 DCS 데이터베이스에 대한 성능 값을 재설정합니다. 마지막 예는 특정 DB2 DCS 데이터베이스에 대한 값을 재설정합니다.

이 프로그램은 적합한 DB2 서버 인스턴스(즉 db2perf를 수행하는 세션에 있는 **db2instance**에 들어 있는)에 대한 데이터베이스 성능 정보를 현재 액세스하는 모든 프로그램에 대한 값을 재설정합니다.

또한 **db2perf**를 호출하여, **db2perf** 명령이 실행될 때 DB2 성능 정보를 원격 액세스할 때 표시되는 값을 재설정합니다.

주: 응용프로그램이 특정 데이터베이스에 대해 전역적으로가 아니라 지역적으로 보는 값을 재설정할 수 있도록 하는 DB2 API로 sqlmrset가 있습니다.

부록G. Windows 2000 데이터베이스 파티션 서버 및 Windows NT 작업

Windows NT 또는 Windows 2000 환경에서 사용자 구성의 특성을 변경하기 위해 작업할 때, 관련된 타스크는 특정 유틸리티를 사용하여 수행됩니다. 다른 운영 체제 환경은 *관리 안내서: 성능의 『프로세서 추가를 통해 사용자 구성 조정』* 장에 표시된 메소드를 사용합니다.

여기에 표시된 유틸리티는 다음과 같습니다.

- 『인스턴스에서의 데이터베이스 파티션 서버 나열』
- 538 페이지의 『인스턴스에 데이터베이스 파티션 서버 추가』
- 540 페이지의 『데이터베이스 파티션 서버 구성 변경』
- 541 페이지의 『인스턴스에서 데이터베이스 파티션 서버 삭제』

인스턴스에서의 데이터베이스 파티션 서버 나열

Windows NT 및 Windows 2000에서, **db2nlist** 명령을 사용하여 인스턴스에 참여하는 데이터베이스 파티션 서버 목록을 확보하십시오.

명령은 다음과 같습니다.

```
db2nlist
```

표시된 대로 이 명령을 사용하면, 기본 인스턴스는 현재 인스턴스가 됩니다 (DB2INSTANCE 환경 변수로 설정). 특정 인스턴스를 지정하려면, 다음을 사용하여 인스턴스를 지정할 수 있습니다.

```
db2nlist /i:instName
```

여기서, *instName*은 원하는 특정 인스턴스 이름입니다.

또한, 다음을 사용하여 선택적으로 각 파티션 서버의 상태를 요청할 수도 있습니다.

```
db2nlist /s
```

각 데이터베이스 파티션 서버의 상태는 시작중, 수행중, 중지중 또는 중지됨 중 하나일 수 있습니다.

인스턴스에 데이터베이스 파티션 서버 추가

Windows NT 및 Windows 2000에서, **db2ncrt** 명령을 사용하여 데이터베이스 파티션 서버(노드)를 인스턴스에 추가하십시오.

주: 이 인스턴스에 이미 데이터베이스가 있으면, **db2ncrt** 명령을 사용하지 마십시오. 대신, **db2start addnode** 명령을 사용하십시오. 이렇게 하면, 데이터베이스는 새 데이터베이스 파티션 서버에 올바르게 추가됩니다. 파일을 변경하면 파티션된 데이터베이스 시스템에 불일치가 생길 수 있으므로 `db2nodes.cfg` 파일을 편집하지 마십시오.

명령에는 다음과 같은 필수 매개변수가 있습니다.

```
db2ncrt /n:node_number
        /u:username,password
/p:logical_port
```

- /n:
데이터베이스 파티션 서버를 식별하기 위한 고유한 노드 번호. 번호는 오름차순으로 1 - 999일 수 있습니다.

- /u:
DB2 서비스의 로그인 계정 이름 및 암호.

- /p:logical_port
논리 포트가 0이 아닐 때 데이터베이스 파티션 서버에 사용되는 논리 포트 번호입니다. 지정되지 않은 경우, 지정된 논리 포트 번호는 0입니다.

논리 포트 매개변수는 머신에서 최초 노드를 작성할 때에만 선택적입니다. 논리 노드를 작성하는 경우, 이 매개변수를 지정하고 사용 중이지 않은 논리 포트 번호를 선택해야 합니다. 여러 제한사항이 있습니다.

- 모든 머신에는 논리 포트 0을 가진 데이터베이스 파티션 서버가 있어야 합니다.

- 포트 번호는 x:\winnt\system32\drivers\etc\ 디렉토리에 있는 서비스 파일에서 FCM 통신용으로 예약된 포트 범위를 초과할 수 없습니다. 예를 들어, 현재 인스턴스의 네 개의 포트 범위를 예약하는 경우 최대 포트 번호는 3입니다(포트 1, 2 및 3이며, 포트 0은 기본 논리 노드입니다). 포트 범위는 **db2icrt** 가 /r:base_port, end_port 매개변수와 함께 사용될 때 정의됩니다.

여러 개의 선택적 매개변수도 있습니다.

- /g:network_name

데이터베이스 파티션 서버에 대한 네트워크명을 지정합니다. 이 매개변수를 지정하지 않은 경우, DB2는 사용자 시스템에서 검출한 최초 IP 주소를 사용합니다.

머신에 여러 IP 주소가 있으며 데이터베이스 파티션 서버를 지정하려는 경우 이 매개변수를 사용하십시오. 네트워크 이름이나 IP 주소를 사용하여 *network_name* 매개변수를 입력할 수 있습니다.

- /h:host_name

호스트 이름이 지역 호스트 이름이 아닐 때 내부 통신에 대한 FCM에서 사용되는 TCP/IP 호스트 이름. 이 매개변수는 원격 머신에서 데이터베이스 파티션 서버를 추가하는 경우 필요합니다.

- /i:instance_name

인스턴스 이름. 기본값은 현재 인스턴스입니다.

- /m:machine_name

노드가 상주하는 Windows NT 워크스테이션의 컴퓨터 이름. 기본 이름은 지역 머신의 컴퓨터 이름입니다.

- /o:instance_owning_machine

인스턴스를 소유하는 머신의 컴퓨터 이름. 기본값은 지역 머신입니다. 이 매개변수는 **db2ncrt** 명령이 인스턴스를 소유하는 머신이 아닌 머신에서 호출될 때 필요합니다.

예를 들어, 새로운 데이터베이스 파티션 서버를 인스턴스 소유 머신 MYMACHIN에 있는 인스턴스 TESTMPP에 추가하고(여러 논리 노드를 실행할 수 있도록), 이 새로운 노드를 논리 포트 1을 사용하는 노드 2로서 알려지게 하려는 경우, 다음을 입력하십시오.

```
db2ncrt /n:2 /p:1 /u:my_id,my_pword /i:TESTMPP  
/M:TEST /o:MYMACHIN
```

데이터베이스 파티션 서버 구성 변경

Windows NT 또는 Windows 2000에서, **db2nchg** 명령을 사용하여 다음을 수행하십시오.

- 하나의 머신에서 또다른 머신으로 데이터베이스 파티션 서버(노드)를 이동시키십시오.
- TCP/IP 호스트 이름을 머신에서 변경합니다.
여러 네트워크 어댑터를 사용하려고 계획하는 경우, 이 명령을 사용하여 『netname』 필드의 TCP/IP 주소를 *db2nodes.cfg* 파일에 지정해야 합니다.
- 다른 논리 포트 번호를 사용하십시오.
- 데이터베이스 파티션 서버(노드)에 대해 다른 이름을 사용하십시오.

명령에는 다음과 같은 필수 매개변수가 있습니다.

```
db2nchg /n:node_number
```

매개변수 /n:은 변경하려는 데이터베이스 파티션 서버 구성의 노드 번호입니다. 이 매개변수는 필수입니다.

선택적 매개변수는 다음을 포함합니다.

- /i:instance_name
이 데이터베이스 파티션 서버가 참여하는 인스턴스를 지정합니다. 이 매개변수를 지정하지 않은 경우, 기본값은 현재 인스턴스입니다.
- /u:username,password
DB2 서비스의 로그인 계정 이름 및 암호를 변경합니다. 이 매개변수를 지정하지 않은 경우, 로그인 계정 및 암호는 동일하게 남아 있습니다.
- /p:logical_port
데이터베이스 파티션 서버에 대한 논리 포트를 변경합니다. 데이터베이스 파티션 서버를 다른 머신으로 이동하려는 경우 이 매개변수를 지정해야 합니다. 이 매개변수를 지정하지 않은 경우, 논리 포트 번호는 변하지 않습니다.
- /h:host_name

내부 통신을 위해 FCM에 의해 사용되는 TCP/IP 호스트 이름을 변경합니다.
이 매개변수를 지정하지 않은 경우, 호스트 이름은 변하지 않습니다.

- /m:machine_name

데이터베이스 파티션 서버를 또다른 머신으로 이동합니다. 데이터베이스 파티션 서버는 인스턴스에 기존 데이터베이스가 없는 경우에만 이동될 수 있습니다.

- /g:network_name

데이터베이스 파티션 서버에 대한 네트워크 이름을 변경합니다.

머신에 여러 IP 주소가 있으며 데이터베이스 파티션 서버의 특정 IP 주소를 사용하려는 경우 이 매개변수를 사용하십시오. 네트워크 이름이나 IP 주소를 사용하여 network_name을 입력할 수 있습니다.

예를 들어, 인스턴스 TESTMPP에 참여하는 노드 2에 지정된 논리 포트를 변경하고, 논리 포트 3을 사용하려면, 다음과 같은 명령을 입력하십시오.

```
db2nchg /n:2 /i:TESTMPP /p:3
```

인스턴스에서 데이터베이스 파티션 서버 삭제

Windows NT 및 Windows 2000에서, **db2ndrop** 명령을 사용하여 데이터베이스가 없는 인스턴스에서 데이터베이스 파티션 서버(노드)를 삭제합니다. 데이터베이스 파티션 서버를 삭제하면, 노드 번호는 새 데이터베이스 파티션 서버에서 다시 사용될 수 있습니다.

인스턴스에서 데이터베이스 파티션 서버를 삭제할 때에는 주의하십시오. 인스턴스에서 인스턴스를 소유하는 데이터베이스 파티션 서버 노드 0을 삭제하면, 인스턴스는 사용할 수 없게 됩니다. 인스턴스를 삭제하려면, **db2idrop** 명령을 사용하십시오.

주: 이 인스턴스에 데이터베이스가 있으면, **db2ndrop** 명령을 사용하지 마십시오. 대신, **db2stop drop nodenum** 명령을 사용하십시오. 이렇게 하면, 데이터베이스는 데이터베이스 파티션 서버에서 올바르게 제거됩니다. 파일을 변경하면 파티션된 데이터베이스 시스템에 불일치가 생길 수 있으므로 db2nodes.cfg 파일을 편집하지 마십시오.

| 여러 논리 노드를 실행 중인 머신에서 논리 포트 0을 지정한 노드를 삭제하려는
| 경우, 논리 포트 0에 지정된 노드를 삭제하기 전에 다른 논리 포트에 지정된 다른
| 모든 노드를 삭제해야 합니다. 각 데이터베이스 파티션 서버는 논리 포트 0에 지
| 정된 노드를 가져야 합니다.

| 명령에는 다음과 같은 매개변수가 있습니다.

```
db2ndrop /n:node_number
```

| • /n:

| 데이터베이스 파티션 서버를 식별하기 위한 고유한 노드 번호. 이것은 필수 매
| 개변수입니다. 번호는 오름차순으로 0 - 999일 수 있습니다. 노드 0은 인스턴
| 스를 소유하는 머신을 나타낸다는 것을 상기하십시오.

| • /i:instance_name

| 인스턴스 이름. 이 매개변수는 선택적입니다. 인스턴스 이름을 제공하지 않은 경
| 우, 기본값은 현재 인스턴스(DB2INSTANCE 레지스트리 변수로 설정)입니다.

부록H. 다중 논리 노드 구성

일반적으로, DB2 Enterprise - Extended Edition을 구성하여 각 머신에 지정된 하나의 데이터베이스 파티션 서버를 갖습니다. 그러나, 동일한 머신에서 실행 중인 여러 데이터베이스 파티션 서버를 갖는 것이 유리한 여러 상황이 있습니다. 이것은 구성에 머신보다 더 많은 노드가 들어 있을 수 있음을 의미합니다. 이 경우에, 동일한 인스턴스에 참여하는 경우 다중 논리 노드를 머신이 실행 중이라고 합니다. 다른 인스턴스에 참여하는 경우, 이 머신은 다중 논리 노드를 조정 중이지 않습니다.

다중 논리 노드 지원을 사용하여, 이러한 유형의 구성에서 선택할 수 있습니다.

- 각 머신이 오직 하나의 데이터베이스 파티션 서버를 갖는 표준 구성.
- 머신이 둘 이상의 데이터베이스 파티션 서버를 갖는 다중 논리 노드 구성.
- 다중 논리 노드가 각 여러 머신에서 실행하는 구성.

다중 논리 노드를 사용하는 구성은 대칭 멀티프로세서(SMP) 아키텍처를 갖는 머신에서 시스템이 조화를 실행할 때 유용합니다. 머신에서 다중 논리 노드를 구성하는 기능은 머신이 실패할 경우 유용합니다. 머신이 실패하는 경우(데이터베이스 파티션이나 서버가 실패하도록 함), DB2START NODENUM 명령을 사용하여 또 다른 머신에서 데이터베이스 파티션 서버를 재시작할 수 있습니다. 이것은 사용자 데이터를 사용 가능하게 합니다.

또다른 이점은 다중 논리 노드가 SMP 하드웨어 구성을 활용할 수 있다는 것입니다. 그리고, 데이터베이스 파티션은 더 작으므로, 데이터베이스 파티션 및 테이블 공간을 백업 및 복원하고, 색인을 작성하는 것과 같은 작업을 수행할 때 더 나은 성능을 얻을 수 있습니다.

다음 두 가지 방식 중 하나로 다중 논리 노드를 구성할 수 있습니다.

- db2nodes.cfg 파일에 논리 노드(데이터베이스 파티션)를 구성합니다. 그런 다음, DB2START 명령과 관련 API를 사용하여 논리 노드와 원격 노드를 모두 시작할 수 있습니다.

주: Windows NT의 경우, 시스템에 데이터베이스가 없는 경우 *db2ncrt*를 사용하여 노드를 추가하거나, 하나 이상의 데이터베이스가 있는 경우 DB2START ADDNODE 명령을 사용하여 노드를 추가해야 합니다. Windows NT 내에서, *db2nodes.cfg* 파일은 수동으로 편집될 수 없습니다.

- 다른 논리 데이터베이스 파티션(노드)이 이미 수행중인 또다른 프로세서에서 논리 노드를 재시작합니다. 이 방법을 사용하면 *db2nodes.cfg*에서 논리 데이터베이스 파티션에 대해 지정된 호스트 이름과 포트 번호를 겹쳐쓸 수 있습니다.

*db2nodes.cfg*에 논리 데이터베이스 파티션(노드)을 구성하려면, 해당 노드에 논리 포트 번호를 할당하는 항목을 파일에 작성해야 합니다. 다음 구문을 사용해야 합니다.

```
nodenumber hostname logical-port netname
```

주: Windows NT의 경우, 시스템에 데이터베이스가 없는 경우 *db2ncrt*를 사용하여 노드를 추가하거나, 하나 이상의 데이터베이스가 있는 경우 DB2START ADDNODE 명령을 사용하여 노드를 추가해야 합니다. Windows NT 내에서, *db2nodes.cfg* 파일은 수동으로 편집될 수 없습니다.

Windows NT에서의 *db2nodes.cfg* 파일의 형식은 Unix에서의 동일한 파일과 비교할 때 다릅니다. Windows NT에서, 컬럼 형식은 다음과 같습니다.

```
nodenumber hostname computername logical_port netname
```

FCM 통신의 *etc* 디렉토리의 *services* 서브디렉토리에 충분한 포트를 정의했는지 확인해야 합니다.

부록. 고속 노드간 통신

DB2 Universal Database Enterprise - Extended Edition 사용 시, 전체 시스템 효율이 사용자 비즈니스에 중요한 통신 집중 환경에서 작업할 수 있습니다.

MPP(massively parallel processing) 환경용으로 사용될 수 있는 두가지 유형의 네트워크가 있습니다. 한 유형은 공용 LAN에서 TCP/IP를 사용합니다. 다른 유형은 TCP/IP를 사용하거나 전용 상호 연결에서 가상 인터페이스(VI) 아키텍처를 사용합니다.

공용 상호 연결은 기존 TCP/IP로 작업합니다. TCP/IP는 거의 어디에서나 통신 프로토콜로서 사용 가능합니다. 이것은 근거리 통신망(LAN) 환경입니다. 이 환경에서의 이점은 추가적인 소유자 하드웨어 및 소프트웨어를 요구하지 않고 즉시 사용자 클러스터에 접속하도록 선택할 수 있다는 것입니다. 이 환경의 단점은 추가적인 클러스터 트래픽이 전체 LAN에서의 서비스 질에 영향을 미친다는 것입니다. 예를 들어, LAN에서 통신에 영향을 미치는 클러스터 내의 데이터베이스 활동에 통신 『버스트』 영향을 미칠 수 있습니다. 또한, 나머지 LAN 환경의 통신은 클러스터 내에서 데이터베이스 처리의 지속적 성능 유지보수를 어렵게 만듭니다.

전용 상호 연결은 개별 네트워크로서 작업합니다. 네트워크는 클러스터 내에서 사용 가능한 유일한 네트워크이거나 LAN 환경에 추가되어 사용될 수 있습니다. 네트워크는 클러스터의 구성원 간의 전용 통신을 제공합니다. 이것을 SNA(System Area Network)라고 합니다. 데이터베이스의 성능은 외부 통신 트래픽(LAN 환경에서와 같이)에 의해 영향을 받지 않으며, 역으로도 마찬가지입니다. 이 환경의 단점은 LAN과 SAN 둘 다의 추가적인 각 하드웨어, 소프트웨어 및 프로토콜 비용은 물론 양쪽 네트워크에 개별적인 관리가 필요할 수 있다는 것입니다. 전용 상호 연결의 예는 100 Mb/Sec 이더넷입니다.

사전에 존재하는 공용 LAN 환경을 유지보수하려고 할 수 있으나(클러스터 내에서) SAN에서 데이터를 대량 전송할 수 있는 기능을 원할 수도 있습니다. 클러스터를 넘어서 통신 액세스하려고 하는 경우 그러한 배열이 편리합니다. Windows NT 운영 환경 내에서, NT 도메인 제어기에 대한 통신 액세스를 위해 공용 LAN

을 보유할 필요할 있을 수 있습니다. 519 페이지의 『부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법』에서 도메인 제어기에 대한 자세한 내용을 참조하십시오.

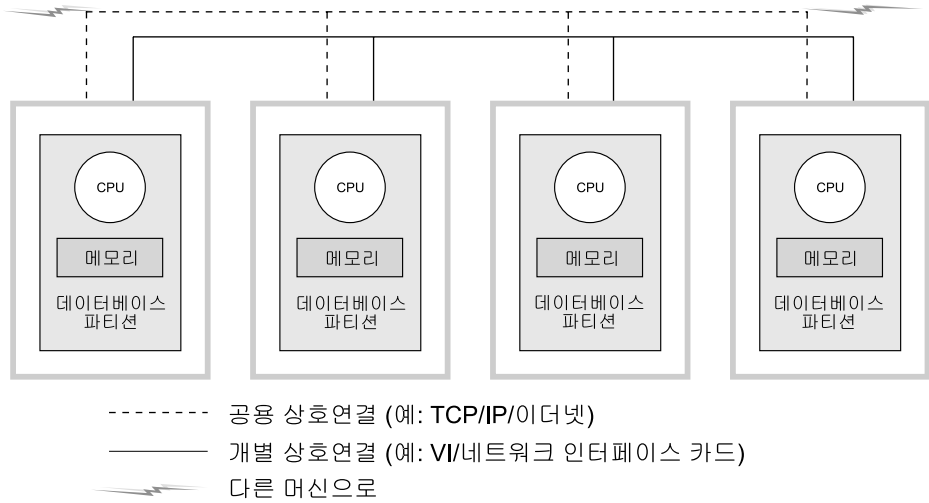


그림 26. 전용 SAN을 공용 LAN과 결합

나머지 절에는 다음 설명이 포함됩니다.

- 『TCP/IP를 사용하여 고속 상호연결』
- 547 페이지의 『VI를 사용하여 고속 상호연결』

TCP/IP를 사용하여 고속 상호연결

TCP/IP를 사용하는 네트워크 하드웨어 설정의 전제조건 예는 다음과 같습니다.

- 표준 이더넷.
 특별한 하드웨어, 소프트웨어 또는 프로토콜 요구사항은 없습니다.
- IBM Netfinity SP Switch.
 요구사항은 다음 절에 설명되어 있습니다.

IBM Netfinity SP Switch 사용에 대한 전제조건

Netfinity에 대한 자세한 내용은 다음 URL을 참조하십시오. <http://www.ibm.com/pc/us/netfinity>

추가 문서 및 소프트웨어 업그레이드에 대한 자세한 내용은 IBM 지원 웹 사이트 : <http://www.ibm.com/pc/support>를 참조하십시오.

1. 서버를 클릭하십시오
2. 계열 아래에서, 클러스터링을 클릭하십시오
3. 기술 정보 아래에서, 소프트웨어 업그레이드의 다운로드가능한 파일을 클릭하거나 문서의 온라인 문서를 클릭하십시오.

IBM Netfinity SP Switch 주제를 찾아 필요한 파일을 다운로드하십시오.

IBM Netfinity SP Switch의 설정 프로시더어

IBM Netfinity SP Switch를 설치하는 지시사항은 *IBM Netfinity SP Switch Installation and User's Guide*에 있습니다.

다양한 하드웨어 및 소프트웨어 구성요소(서버 랙 enclosure, 호스트 어댑터 및 SP 스위치 소프트웨어와 같은)가 제공되는 하드웨어 및 소프트웨어 지시사항을 사용하여 관련 구성요소를 설치, 구성 및 테스트해야 합니다.

DB2가 설치되어 추가 수정사항이 없는 경우 IBM Netfinity SP Switch를 사용합니다.

VI를 사용하여 고속 상호연결

가상 인터페이스(VI) 아키텍처는 Windows NT 대량 병렬 처리(MPP) 환경에서 TCP/IP를 대신하는 노드간 통신 프로토콜입니다. VI는 시스템 영역 네트워크(SAN)에서 성능을 향상시키기 위해 Intel, Microsoft 및 Compaq가 합작으로 개발한 새로운 통신 아키텍처입니다. <http://www.viarch.org>에서 아키텍처에 대한 자세한 내용을 참조하십시오.

VIA가 작동 가능한 네트워크 인터페이스 카드(NIC), 스위치 및 소프트웨어 드라이버 구현을 수반하는 DB2 Universal Database에서 별도로 구입될 수 있는 제품이 있습니다. 몇몇 독립 하드웨어 벤더(IHV)가 이러한 제품의 릴리스를 발표하거나, 릴리스를 발표할 계획을 가지고 있습니다.

VI 아키텍처는 TCP/IP와 비교할 때 낮은 대기, 상위 대역폭 및 더 낮은 CPU 소비를 갖습니다. 통신 중심의 환경에서, VI 아키텍처를 사용하면 전체적인 시스템

처리량이 향상됩니다. 클러스터의 노드 수가 많을수록 더 많은 양의 데이터가 전송되고 VI 아키텍처 사용으로 생기는 이점이 더욱 커집니다.

DB2 Universal Database는 *Virtual Interface Architecture Specification, Version 1.0, Intel Virtual Interface(VI) Architecture Developers' Guide, Version 1.0* 을 준수하고, 『Virtual Interface Architecture Conformance Suite』를 전달하는 VI 아키텍처 구현을 지원합니다. 이 스펙은 웹의 http://www.intel.com/design/servers/vi/the_spec/specification.htm에 들어 있습니다. 개발자 안내서는 웹의 http://www.intel.com/design/servers/vi/developer/ia_imp_guide.htm에 들어 있습니다. conformance suite에 대한 내용도 이 URL에 들어 있습니다.

IBM은 DB2 Universal Database EEE V5.2를 사용하는 가상 인터페이스(VI) 아키텍처를 지원할 것을 발표했습니다.

VI 아키텍처를 고수하고 DB2 Universal Database EEE의 지원을 받는 기타 제품에 관한 정보를 구하려면 <http://www.software.ibm.com/data>의 DB Universal Database 지원 기구에 문의하거나 1-800-237-5511(미국 및 캐나다에서만 가능)로 전화하십시오.

DB2 Universal Database로 테스트된 제품에는 다음 제품들이 있습니다.

- GigaNet Interconnect. 세부사항은 549 페이지의 『GigaNet Interconnect에 대한 프로시듀어 설정』에서 자세한 내용을 참조하십시오.
- Compaq ServerNet Interconnect. 552 페이지의 『ServerNet Interconnect에 대한 프로시듀어 설정』에서 자세한 내용을 참조하십시오.
- Fujitsu ServerNet Interconnect. 556 페이지의 『Synfinity Interconnect의 설정 프로시듀어』에서 자세한 내용을 참조하십시오.

DB2 Universal Database에서 작동할 수 있는 기타 제품이 있을 수 있습니다. 해당 제품의 벤더 및 IBM 서비스 지원부에 문의하여 다른 제품이 지원되는지 확인하십시오.

가상 인터페이스(VI) 하드웨어 설정

VI를 사용하는 네트워크 하드웨어 설정의 전제조건 예는 다음과 같습니다.

- GigaNet Interconnect.

『GigaNet Interconnect에 대한 프로시듀어 설정』에는 이 선택에 필요한 하드웨어, 소프트웨어 및 프로토콜 정보에 대한 개요가 들어 있습니다.

GigaNet 제품에 관한 정보를 구하거나 GigaNet 서비스 지원부에 연락하려면 다음 URL, <http://www.giganet.com/>을 사용하십시오.

- Compaq ServerNet Interconnect.

552 페이지의 『ServerNet Interconnect에 대한 프로시듀어 설정』에는 이 선택에 필요한 하드웨어, 소프트웨어 및 프로토콜 정보에 대한 개요가 들어 있습니다.

ServerNet 제품에 관한 정보를 알아보거나 ServerNet 서비스 지원부에 연락하려면 다음 URL, <http://www.servernet.com/>을 사용하십시오.

- Fujitsu Synfinity Interconnect.

556 페이지의 『Synfinity Interconnect의 설정 프로시듀어』에는 이 선택에 필요한 하드웨어, 소프트웨어 및 프로토콜 정보에 대한 개요가 들어 있습니다.

Fujitsu System Technologies에서, Synfinity 제품에 관한 정보를 알아보거나 Synfinity 서비스 지원부에 연락하려면 다음 URL <http://www.fjst.com/>을 사용하십시오.

VI를 사용하려면 DB2를 구성해야 합니다. 558 페이지의 『VI를 사용하여 DB2 수행』에는 VI를 사용하는 데 필요한 정보가 있습니다.

GigaNet Interconnect에 대한 프로시듀어 설정

이러한 환경을 설정하는 데 필요한 하드웨어 및 소프트웨어 목록에는 다음 제품이 포함됩니다.

- GigaNet GNN1000 네트워크 인터페이스 카드
- GigaNet GNX5000 스위치
- GigaNet GNCxx11 Copper Interconnect 케이블
- GigaNet cLAN 소프트웨어, 버전 2.0

GigaNet Interconnect가 DB2 Universal Database에서 작업할 수 있는지 확인하는 데 필요한 단계는 아래와 같습니다. 각 단계는 각 단계에서 필요한 내용의 요약이므로, 각 단계에 연관된 모든 세부사항이 여기에 나와 있지는 않습니다. 또한, 필요한 자세한 지시사항을 제공하는 각 단계별 참조 문서를 사용해야 합니다.

각 GigaNet GNN1000는 GigaNet cLAN 소프트웨어 CD-ROM에 패키징되어 있습니다. CD-ROM에는 GigaNet Interconnect를 설정하는 데 필요한 모든 소프트웨어가 들어 있습니다. 그밖에도 CD-ROM에는 VI 아키텍처 SDK 및 Adobe Acrobat Reader도 포함되어 있습니다. 이들 두 가지 항목은 VI 작동 가능 응용 프로그램을 개발하는 사용자에게만 필요합니다.

단계 요약:

1. 어댑터 카드 설치
2. 스위치 및 케이블 설치
3. 어댑터 드라이버 설치
4. cLAN Management Console 설치
5. Interconnect 테스트

단계는 다음과 같습니다.

1. GigaNet GNN1000 네트워크 인터페이스 카드 설치. 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오.
2. GigaNet GNX5000 스위치 및 케이블 설치. 설치 지시사항은 *GigaNet GNX5000 User Guide*를 참조하십시오.
3. GNX5000 스위치에 연결된 각 노드에 GigaNet GNN1000 어댑터 드라이버 소프트웨어 설치. 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오. GigaNet가 제공하는 드라이버를 설치할 경우, 다음과 같은 추가적인 세부사항이 있습니다.
 - a. 이미 설치되어 있는 GNN1000 드라이버의 이전 버전을 제거하십시오. 이를 제거한 다음에는 노드를 재부트해야 합니다.
 - b. 시작->설정->제어판->네트워크->어댑터->추가를 사용하여 드라이버를 설치하십시오.
 - c. **디스크 있음...**을 클릭한 후 CD-ROM의 드라이버 디렉토리를 지정하십시오. 예를 들어, CD-ROM 드라이브가 F:이면, F:\Driver를 사용해야 합니다.
 - d. 『GNN1000 NDIS 어댑터』를 선택한 후 확인을 클릭하십시오.
 - e. 네트워크 프로토콜을 구성하여 설치를 완료하십시오.

GigaNet 어댑터 드라이버 소프트웨어는 GigaNet의 웹 사이트인 <http://www.giganet.com>에서도 사용 가능합니다. GigaNet 웹 사이트의 지원 페이지에 있는 다운로드 및 설치 지시사항을 참조하십시오.

GNN1000 어댑터 드라이버를 설치하면 노드를 재부트해야 합니다.

4. GigaNet cLAN Management Console(GMC)은 GigaNet Interconnect의 통합성을 테스트하는 데 사용될 수 있습니다. GigaNet cLAN Management Console은 콘솔과 에이전트 두 부분으로 이루어져 있습니다. 에이전트는 클러스터의 모든 노드에 설치되어 있어야 합니다. 콘솔은 클러스터에 있는 노드에 액세스하는 모든 네트워크 노드에 설치될 수 있습니다. 가장 융통적이고 권장되는 설치 방법은 콘솔과 에이전트가 모두 클러스터의 각 노드에 설치되게 하는 것입니다.

GigaNet cLAN Management Console를 설치하십시오. cLAN Management Console에 관한 추가 정보 및 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오. 다음은 설치 프로시듀어에 대한 추가적인 세부사항입니다.

- a. cLAN 소프트웨어 CD를 CD-ROM 드라이브에 삽입하십시오.
- b. CD 자동 설치 메뉴가 나타날 때까지 기다리십시오.
- c. 『cLAN Management Console 설치』를 클릭하십시오.
- d. 클러스터의 나머지 각 노드에 대해서 이 설치 프로시듀어를 반복 실행하십시오.

GigaNet cLAN Management Console 소프트웨어는 GigaNet의 웹 사이트인 <http://www.giganet.com>에서도 사용 가능합니다. GigaNet 웹 사이트의 지원 페이지에 있는 다운로드 및 설치 지시사항을 참조하십시오.

cLAN Management Console을 설치하면 노드를 재부트해야 합니다.

5. GigaNet 하드웨어가 작동하는지 테스트하십시오. 이것은 다음과 같은 방법으로 가능합니다.
 - a. GMC를 여십시오(프로그램→GigaNet→cLAN Management Console).
 - b. LAN에서 액세스할 수 있는 모든 머신을 나타내는 대화 상자가 표시됩니다. **ESC** 키를 누르십시오.
 - c. 메뉴 표시줄에서 **Console**→**Local**을 선택하십시오.
 - d. 클러스터의 모든 구성원이 표시되고 모두 『사용중』인지 확인하십시오.

- e. 메뉴 표시줄에서 **Utilities**—>**VI Throughput**을 선택하십시오 이것은 데이터가 하드웨어를 실제로 통과하는지 점검하기 위한 처리량 테스트를 수행합니다.
- f. 테스트시 사용하려고 하는 두 노드의 컴퓨터 이름을 대문자로 입력하십시오. 지역 노드를 소스 노드로 식별하십시오.
- g. **Start Measuring**을 클릭하십시오 데이터가 최소한 매초 65MB 비율로 전송되어야 합니다.
- h. 연결 테스트를 중단하려면 **Stop Measuring**을 클릭하십시오.
- i. 지역 노드(소스)와 다른 노드(싱크(Sink))간의 처리량을 측정하여 클러스터에 있는 다른 노드에 대해 테스트를 반복하십시오.

연결 테스트가 작동하지 않는 것으로 보이면, *GigaNet GNN1000 User Guide* 및 *GigaNet GNX5000 User Guide*의 문제점 해결 섹션을 참조하십시오.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 *Windows NT용 DB2 Universal Database*를 설치하고 구현하는 방법에 대한 자세한 내용을 참조하십시오.

ServerNet Interconnect에 대한 프로시듀어 설정

이러한 환경을 설정하는 데 필요한 하드웨어 및 소프트웨어 목록에는 다음 제품이 포함됩니다.

- ServerNet PCI Adapter Driver(SPAD), (제품 ID T0089), 버전 1.3.5 이상
- ServerNet Switch 1
- ServerNet Area Network Manager(SANMan), (제품 ID T0087), 버전 1.1.3 이상

다음은 ServerNet Interconnect가 DB2 Universal Database에서 작업할 수 있는지 확인하는 데 필요한 단계입니다. 각 단계는 필요한 내용의 요약이므로, 각 단계에 연관된 모든 세부사항이 여기에 나와 있지는 않습니다. 또한, 필요한 자세한 지시사항을 제공하는 각 단계별 참조 문서를 사용해야 합니다.

다음에 나와 있는 단계는 사용자가 클러스터에서 노드를 6개까지만 사용하고 있다고 가정하고 있습니다. 6개보다 더 많은 노드를 사용해야 하는 경우에는 ServerNet에 문의하십시오.

단계는 다음과 같습니다.

1. ServerNet 네트워크 인터페이스 카드를 설치하십시오. 설치 지시사항은 *ServerNet-I Virtual Interface Software Release Document, (product ID N0031)*를 참조하십시오.
2. ServerNet Switch 1을 설치하십시오. 설치 지시사항은 *ServerNet-I Virtual Interface Software Release Document, (product ID N0031)*를 참조하십시오.
3. 이전 ServerNet 드라이버를 설치 해제하십시오(ServerNet를 처음 설치하는 경우이면 이 단계를 생략하십시오).
 - a. 네트워크 제어 패널을 여십시오(시작->설정->제어판->네트워크).
 - b. 어댑터 탭을 클릭하십시오.
 - c. Tandem ServerNet PCI 어댑터 드라이버를 제거하십시오.
 - d. 서비스 탭을 클릭하십시오.
 - e. SANMan을 제거하십시오.
 - f. 프로토콜 탭을 클릭하십시오.
 - g. Tandem ServerNet-I VI 프로토콜을 제거하십시오.
4. Tandem ServerNet PCI 어댑터 드라이버를 설치하십시오. 다음은 ServerNET가 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.
 - a. 네트워크 제어판을 여십시오(시작->설정->제어판->네트워크).
 - b. 어댑터 탭을 클릭하십시오(어댑터 화면이 표시됩니다).
 - c. 새로운 ServerNet 드라이버는 별도 드라이브 및/또는 디렉토리에 배치되어야 합니다. 그런 다음, 정확한 드라이브 및/또는 드라이버를 참조하는 명령 프롬프트에 『ernnn.exe -d』를 입력하여 자체 발췌 프로그램을 시작하십시오(『ernnn.exe』은 설치될 ServerNet 드라이버의 특정 버전을 식별하는 번호— ERnnn.EXE —의 뒤에 표시되는 엔지니어링 릴리스의 이름입니다).
 - d. 발췌된 파일이 위치한 드라이브 및/또는 디렉토리로 변경하십시오. 『Spad n.n.n \ Free』 서브디렉토리로 변경하십시오. 여기서, 『n.n.n』은 제품의 특

정 버전입니다. 문제점 해결 또는 개발 환경에서 작업 중일 경우에는 『Spad n.n.n \ Free』 서브디렉토리 대신 『Spad n.n.n \ Checked』 서브디렉토리로 변경하십시오.

- e. 『oemsetup.multi_node』 파일 이름을 『oemsetup.inf』로 변경하십시오.
- f. 어댑터 탭에서 추가를 선택하십시오(어댑터 선택 화면이 표시됩니다).
- g. 디스크 있음...을 클릭하십시오(디스크 삽입 화면이 나타납니다).
- h. oemsetup.inf 파일이 위치한 드라이브 및/또는 디렉토리를 입력하십시오.
- i. 대화 상자에 『Tandem ServerNet PCI 어댑터 드라이브』가 표시되는지 확인한 후 확인을 클릭하십시오 어댑터 목록이 ServerNet 어댑터를 표시하는지 확인하십시오. 닫기를 클릭하십시오.
- j. 예를 선택하여 컴퓨터를 재시작하십시오 또는 아니오를 선택하고 SANMan 및 VI SDK를 계속 설치하십시오.

5. SANMan을 설치하십시오. 다음은 ServerNet가 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.

- a. 네트워크 제어판을 여십시오. (시작->설정->제어판->네트워크).
- b. 서비스 탭을 클릭하십시오(서비스 화면이 표시됩니다).
- c. 새로운 ServerNet 드라이버는 별도 드라이브 및/또는 디렉토리에 배치되어야 합니다. 그런 다음, 정확한 드라이브 및/또는 드라이버를 참조하는 명령 프롬프트에 『ernnn.exe -d』를 입력하여 자체 발췌 프로그램을 시작하십시오(『ernnn.exe』은 설치될 ServerNet 드라이버의 특정 버전을 식별하는 번호— ERnnn.EXE —의 뒤에 표시되는 엔지니어링 릴리스의 이름입니다).
- d. 서비스 탭에서 추가를 선택하십시오(서비스 선택 화면이 표시됩니다).
- e. 발췌된 파일이 위치한 드라이브 및/또는 디렉토리로 변경하십시오. 『SANMan n.n.n \ Free』 서브디렉토리로 변경하십시오. 여기서, 『n.n.n』은 제품의 특정 버전입니다. 문제점 해결 또는 개발 환경에서 작업 중일 경우에는 『SANMan n.n.n \ Free』 서브디렉토리 대신 『SANMan n.n.n \ Checked』 서브디렉토리로 변경하십시오.
- f. 스위치에 켜 있는 불빛을 보고 스위치가 X 또는 Y인지 판별하십시오. 불빛 하나는 『X』를 의미하고, 다른 불빛은 『Y』를 의미합니다.

- g. X 스위치일 경우, X=1 및 Y=0를 선택하십시오. 모든 케이블이 네트워크 카드의 X 포트에 연결되어 있는지 확인하십시오.
- h. Y 스위치일 경우, X=0 및 Y=1을 선택하십시오. 모든 케이블이 네트워크 카드의 Y 포트에 연결되어 있는지 확인하십시오.
- i. 현재 머신에서 네트워크 카드가 연결된 스위치의 포트 번호를 제공하십시오.
- j. 6개의 모든 포트에 대해 『PC』를 선택하십시오.

6. 가상 인터페이스 프로토콜을 설치하십시오. 다음은 ServerNet가 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.

- a. 네트워크 제어판을 여십시오(시작->설정->제어판->네트워크).
- b. 프로토콜 탭을 클릭하십시오(네트워크 프로토콜 화면이 표시됩니다).
- c. 새로운 ServerNet 드라이버는 별도 드라이브 및/또는 디렉토리에 배치되어야 합니다. 그런 다음, 정확한 드라이브 및/또는 드라이버를 참조하는 명령 프롬프트에 『ernnn.exe -d』를 입력하여 자체 발체 프로그램을 시작하십시오(『ernnn.exe』은 설치될 ServerNet 드라이버의 특정 버전을 식별하는 번호— ERnnn.EXE —의 뒤에 표시되는 엔지니어링 릴리스의 이름입니다).
- d. 프로토콜 탭에서 추가를 선택하십시오(네트워크 프로토콜 선택 화면이 표시됩니다).
- e. 디스크 있음...을 클릭하십시오(디스크 삽입 화면이 나타납니다).
- f. 발체된 파일이 위치한 드라이브 및/또는 디렉토리로 변경하십시오.

7. ServerNet 하드웨어가 작동하는지 테스트하십시오. 사용 가능한 테스트 프로그램은 없습니다. 대신, 간단히 DB2를 사용하여 SvrerNet 하드웨어를 테스트하십시오.

하드웨어가 작동하는 것으로 보이지 않을 경우, 추가 문제점 해결 도움말은 *ServerNet-I Virtual Interface Software Release Document, (product ID N0031)*를 참조하십시오.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 Windows NT용 DB2 Universal Database를 설치하고 구현하는 방법에 대한 자세한 내용을 참조하십시오.

Synfinity Interconnect의 설정 프로시듀어

이러한 환경을 설정하는 데 필요한 하드웨어 및 소프트웨어 목록에는 다음 제품이 포함됩니다.

- Synfinity PCI Network Interface Card
- Synfinity six port Switch
- Synfinity Interconnect Cables
- Synfinity Cluster Manager Software, 버전 1.10.

Synfinity Interconnect가 DB2 Universal Database에서 작업할 수 있는지 확인하는 데 필요한 단계는 아래와 같습니다. 각 단계는 각 단계에서 필요한 내용의 요약이므로 각 단계에 연관된 모든 세부사항은 여기에 나와 있지는 않습니다. 또한, 필요한 자세한 지시사항을 제공하는 각 단계별 참조 문서를 사용해야 합니다.

각 Synfinity System은 Synfinity Cluster Manager Software, 버전 1.10 CD-ROM과 함께 패키지화되어 있습니다. CD-ROM에는 Synfinity Interconnect를 설정하는 데 필요한 모든 문서 및 소프트웨어가 들어 있습니다. 그밖에도 CD-ROM에는 Synfinity Cluster User Guide가 들어 있습니다.

다른 VI 하드웨어, 소프트웨어 및 프로토콜이 설치된 경우, Synfinity interconnect를 설치하기 전에 모든 것을 제거할 필요가 있을 수 있습니다.

Synfinity interconnect가 설치되면 외래 하드웨어로 간주되어 Windows NT 제어 패널에서 볼 수 없을 수 있습니다.

단계 요약:

1. 어댑터 카드 설치
2. Synfinity 클러스터 관리 프로그램 소프트웨어 설치
3. 스위치 및 케이블 설치
4. Interconnect 테스트

단계는 다음과 같습니다.

1. Synfinity PCI Network Interface Card를 설치하십시오. *Synfinity Cluster Manager Software User Guide*에서 설치 지시사항에 대한 자세한 내용을 참조하십시오.

2. 스위치와 연결된 노드에서 Synfinity 클러스터 관리 프로그램 소프트웨어를 설치하십시오. *Synfinity Cluster User Guide*에서 설치 지시사항에 대한 자세한 내용을 참조하십시오.

선택한 노드는 클러스터 관리 프로그램입니다. 이것은 CD에서 소프트웨어를 설치해야 하는 유일한 노드입니다.

Synfinity 클러스터 관리 프로그램 소프트웨어를 설치하고 나면, 수행해야 합니다. 클러스터 관리 프로그램은 클러스터 플랜을 제공하며 단계별 지시사항을 통해 네트워크를 구성하도록 도우며, 최적의 라우팅 및 케이블 옵션을 권고합니다. 이 단계는 케이블이 Synfinity 스위치 및 네트워크 카드에 연결되기 전에 완료되어야 합니다. 계획 프로세스의 일부로서, 클러스터 관리 프로그램은 다른 노드에서 사용할 설치가능한 디스켓을 작성하기 위해 클러스터 플랜을 사용합니다. 이것은 다른 노드에 있는 카드의 드라이버 소프트웨어를 포함합니다. *Synfinity Cluster Users guide*에서 자세한 내용을 참조하십시오.

3. Synfinity 스위치 및 케이블을 설치하십시오. *Synfinity Cluster User Guide*에서 설치 지시사항에 대한 자세한 내용을 참조하십시오.
4. Synfinity 하드웨어가 작동하는지 테스트하십시오. 이것은 다음과 같은 방법으로 가능합니다.
 - a. 클러스터에 있는 임의의 시스템에서, MS NT에서 "명령 프롬프트" 창을 여십시오.
 - b. Synfinity 클러스터 관리 프로그램 소프트웨어가 로드된 "utils" 서브디렉토리로 디렉토리를 변경하십시오.
 - c. "vitest"를 입력하고 표시된 노드 번호를 적어 놓으십시오.
 - d. 클러스터에 있는 임의의 시스템으로 이동하고, "명령 프롬프트" 창을 여십시오.
 - e. Synfinity 클러스터 관리 프로그램 소프트웨어가 이 다른 시스템에 로드된 "utils" 서브디렉토리로 디렉토리를 변경하십시오.
 - f. "vitest x"를 입력하십시오. 여기서 x는 위의 단계 3의 노드 번호입니다.
 - g. "CONNECTION GOOD" 메시지가 표시되어야 합니다.
 - h. "NO CONNECTION" 메시지가 표시되는 경우, 케이블 및 하드웨어 설정을 점검하고, *Synfinity Cluster User Guide*에서 문제점 해결에 대한 자

세한 내용을 참조하고, "Tech-tips"에 대한 지원 웹 페이지를 <http://www.fjst.com/>에서 확인하십시오.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 Windows NT용 DB2 Universal Database를 설치하고 구현하는 방법에 대한 자세한 내용을 참조하십시오.

VI를 사용하여 DB2 수행

자세한 설치 정보는 Windows용 DB2 Enterprise - Extended Edition 빠른 시작에 수록되어 있습니다.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 설명한 대로 DB2의 설치를 완료한 후에, 다음 DB2 레지스트리 변수를 설정하고 인스턴스에 있는 각 데이터베이스 파티션 서버에서 다음 타스크를 수행하십시오.

- DB2_VI_ENABLE=ON을 설정하십시오.
- DB2_VI_DEVICE=nic0을 설정하십시오.

주: Synfinity Interconnect를 사용하여, 이 변수가 DB2_VI_DEVICE=VINIC로 설정되어야 합니다. 장치 이름(VINIC)은 대문자이어야 합니다.

- DB2_VI_VIPL=vipl.dll을 설정하십시오.
- MPP 인스턴스에 db2start를 입력하십시오.
- db2diag.log 파일을 검토하십시오. 각 파티션에 대해 『VI가 작동 가능함』을 나타내는 메시지가 나타나야 합니다.
- FCM(Fast Communications Manager) 구성 매개변수를 갱신할 필요가 있을 수 있습니다. FCM과 관련된 자원 제한조건의 결과로서 문제점이 발생하는 경우, FCM 구성 매개변수의 값을 증가시켜야 합니다. FCM 구성 매개변수의 값을 증가시킨 또다른 고속 상호 연결 환경에서 이동 중인 경우, 더 낮은 값이 필요할 수 있습니다. 또한, Windows NT에서, DB2NTMEMSIZE 레지스트리 변수를 설정하여 DB2 기본값을 겹쳐쓰기 할 필요가 있을 수 있습니다. 관리 안내서: 성능에서 이 레지스트리 변수에 대한 자세한 내용을 참조하십시오.

부록J. LDAP(Lightweight Directory Access Protocol) 등록부 서비스

LDAP(Lightweight Directory Access Protocol)는 디렉토리 서비스에 대한 산업 표준 액세스 방식입니다. 디렉토리 서비스는 분산 환경 내의 다중 시스템과 서비스에 대한 자원 정보의 저장소이며, 이들 자원에 대한 클라이언트 및 서버 액세스를 제공합니다. 각 데이터베이스 서버 인스턴스는 LDAP 서버로 그 존재를 공표하며 데이터베이스가 작성될 때 LDAP 등록부로 데이터베이스 정보를 제공합니다. 클라이언트가 데이터베이스에 연결할 때, 카탈로그에 대한 정보는 LDAP 등록부에서 검색될 수 있습니다. 각 클라이언트는 각 머신에 지역적으로 카탈로그 정보를 저장하는 것이 더 이상 필요하지 않습니다. 클라이언트 응용프로그램은 데이터베이스에 연결하는 데 필요한 정보 요구를 찾기 위해 LDAP 등록부를 검색합니다.

클라이언트가 관련 지역 디렉토리 카탈로그에서 LDAP 등록부만 한 번 검색하도록, 캐쉬하는 메커니즘이 존재합니다. 정보가 검색되고 나면 지역 머신에 저장되거나 캐쉬됩니다. 동일한 정보에 대한 차후 액세스는 *dir_cache* 데이터베이스 관리 프로그램 구성 매개변수와 DB2LDAPCACHE 레지스트리 변수의 값에 근거합니다.

- DB2LDAPCACHE=NO이고 *dir_cache*=NO이면, 항상 LDAP에서 정보를 읽습니다.
- DB2LDAPCACHE=NO이고 *dir_cache*=YES이면, LDAP에서만 정보를 읽어 DB2 캐쉬로 삽입합니다.
- DB2LDAPCACHE=YES이거나 설정되지 않고, 필수 정보가 지역 캐쉬에 없는 경우, LDAP 등록부에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: DB2LDAPCACHE 레지스트리 변수는 데이터베이스 및 노드 디렉토리에만 적용할 수 있습니다.

LDAP 클라이언트 및 서버 구성 지원

다음 테이블은 지원되는 LDAP 클라이언트 및 서버 구성을 요약합니다.

표 23. LDAP 클라이언트 및 서버 구성을 지원합니다.

	IBM SecureWay 디렉토리 V3.1 및 V3.1.1	Microsoft 사용중인 디렉토리
IBM LDAP 클라이언트	지원	지원 안됨
Microsoft LDAP/ADSI 클라이언트	지원	지원

IBM SecureWay Directory 버전 3.1은 Windows NT, AIX 및 Solaris에 사용 가능한 LDAP 버전 3 서버입니다. SecureWay 디렉토리는 AIX 및 AS/400에서 기본 운영 체제의 일부로서 OS/390 보안 서버와 함께 제공됩니다.

DB2는 IBM LDAP 클라이언트를 AIX, Solaris, Windows NT, Windows 95 및 Windows 98에서 지원합니다.

Microsoft Active Directory는 LDAP 버전 3 서버이며 Windows 2000 서버 운영 체제의 일부로서 사용 가능합니다.

Microsoft LDAP Client 지원은 다음과 같은 Microsoft 제품에 포함됩니다.

1. Outlook 98, Outlook 2000 또는 Outlook Express

주: Outlook Express는 Microsoft Internet Explorer의 일부로서 설치됩니다.

2. Exchange Server
3. Windows NT Server Service Pack 4
4. Windows 98 Second Edition
5. Windows 2000

Microsoft LDAP Client 지원은 ADSI(Active Directory Service Interface) 구성요소에 포함됩니다. ADSI의 최신 버전은 <http://www.microsoft.com/adsi>에서 다운로드될 수 있습니다.

Windows 95, Windows 98, Windows NT 또는 Windows 2000 운영 체제에서 실행할 때, DB2는 IBM LDAP 클라이언트 또는 Microsoft LDAP 클라이언트 중 하나를 사용하여 IBM SecureWay 디렉토리 서버에 액세스합니다. Microsoft

LDAP 클라이언트가 사용 가능하지 않은 경우, DB2는 IBM LDAP 클라이언트를 사용하도록 시도합니다. 명시적으로 IBM LDAP 클라이언트를 선택하려면, **db2set** 명령을 사용하여 DB2LDAP_CLIENT_PROVIDER 레지스트리 변수를 『IBM』에 설정하십시오.

Windows 2000에서 사용 중인 디렉토리 지원

DB2는 다음과 같이 사용 중인 디렉토리를 검색합니다.

1. DB2 데이터베이스 서버는 **ibm_db2Node** 오브젝트로서 사용 중인 디렉토리에 발표됩니다. **ibm_db2Node** 오브젝트 클래스는 SCP(ServiceConnectionPoint 오브젝트 클래스의 서브클래스입니다. 각 **ibm_db2Node** 오브젝트에는 클라이언트 응용프로그램을 DB2 데이터베이스 서버에 연결하게 하는 프로토콜 구성 정보가 들어 있습니다. 새로운 데이터베이스가 작성되면, 데이터베이스는 **ibm_db2Node** 오브젝트 아래의 **ibm_db2Database** 오브젝트로서 활성 디렉토리에 공표됩니다.
2. 원격 데이터베이스에 연결하면, DB2 클라이언트는 **ibm_db2Database** 오브젝트에 대해 LDAP 인터페이스를 통해 활성 디렉토리를 조회합니다. 데이터베이스 서버에 연결하는 프로토콜 통신(바인딩 정보)은 **ibm_db2Database** 오브젝트가 작성되는 **ibm_db2Node** 오브젝트에서 얻습니다.

사용중인 디렉토리로 DB2 구성

Microsoft Active Directory에 액세스하려면, 다음과 같은 조건이 충족되어야 합니다.

1. DB2를 수행하는 머신은 Windows 2000 도메인에 속해야 합니다.
2. Microsoft LDAP 클라이언트가 설치되어 있습니다. Microsoft LDAP 클라이언트는 Windows 2000 운영 체제의 일부입니다. Windows 95, Windows 98 또는 Windows NT의 경우, **wldap32.dll**이 시스템 디렉토리 아래에 있음을 검증할 필요가 있습니다.
3. LDAP 지원을 사용 가능하게 하십시오. Windows 2000의 경우, LDAP 지원은 설치 프로그램으로 사용 가능합니다. Windows 95/98/NT의 경우, **db2set** 명령을 사용하여 **DB2_ENABLE_LDAP** 레지스트리 변수를 『YES』로 설정하여 명시적으로 LDAP를 사용 가능하게 해야 합니다.

4. 사용 중인 디렉토리에서 정보를 읽기 위해 DB2를 수행할 때 도메인 사용자 계정으로 로그인하십시오.

IBM LDAP 환경에서 DB2 구성

IBM LDAP 환경에서 DB2를 사용하기 전에, 각 머신에서 다음을 구성해야 합니다.

- LDAP 지원을 사용 가능하게 하십시오. Windows 2000의 경우, LDAP 지원은 설치 프로그램으로 사용 가능합니다. Windows 95/98/NT의 경우, **db2set** 명령을 사용하여 **DB2_ENABLE_LDAP** 레지스트리 변수를 『YES』로 설정하여 명시적으로 LDAP를 사용 가능하게 해야 합니다.
- LDAP 서버의 TCP/IP 호스트 이름 및 포트 번호. 이 값은 **DB2LDAPHOST** 응답 키워드를 사용하여 무인 설치 중 입력되거나, 사용자가 **DB2SET** 명령을 사용하여 나중에 수동으로 설정할 수 있습니다.

```
db2set DB2LDAPHOST=<hostname[:port]>
```

여기서 **hostname**은 LDAP 서버의 TCP/IP 호스트 이름이며 **[:port]**는 포트 번호입니다. 포트 번호가 지정되지 않으면, DB2는 기본 LDAP 포트(389)를 사용합니다.

DB2 오브젝트는 LDAP 기본 구별된 이름(baseDN)에 위치합니다. IBM SecureWay LDAP 등록부 서버 버전 3.1을 사용 중인 경우, DB2는 서버에서 동적으로 이 정보를 얻을 수 있으므로 기본 구별 이름을 구성할 필요는 없습니다. 그러나, IBM eNetwork Directory Server 버전 2.1을 사용 중이라면, **DB2SET** 명령을 사용하여 각 머신에서 LDAP 기본 구별된 이름을 구성해야 합니다.

```
db2set DB2LDAP_BASEDN=<baseDN>
```

여기서 **baseDB**은 LDAP 서버에 정의된 LDAP 접미부의 이름입니다. 이 LDAP 접미부는 DB2 오브젝트를 포함하는 데 사용됩니다.

- LDAP 사용자의 구별된 이름(DN) 및 암호. 이것은 사용자가 LDAP를 사용하여 DB2 사용자 특유의 정보를 저장하려고 계획하는 경우에만 필요합니다.

LDAP 사용자 작성

DB2는 사용자 레벨에서 DB2 레지스트리 변수 및 CLI 구성을 설정하도록 지원 합니다. 이것은 AIX 및 Solaris 플랫폼에서 사용 가능하지 않습니다. 사용자 레벨 지원은 멀티 사용자 환경에서 사용자 고유 설정을 제공합니다. 시스템 환경이나 또 다른 사용자 환경을 간섭하지 않고 각 로그인 사용자가 자신의 환경을 사용자 정의할 수 있는 Windows NT 터미널 서버가 그 예입니다.

IBM LDAP 등록부를 사용할 때, 사용자 레벨 정보를 LDAP에 저장하기 전에 LDAP 사용자를 정의해야 합니다. 다음 방법 중 하나로 LDAP 사용자를 작성할 수 있습니다.

- 사용자 오브젝트의 모든 속성을 포함할 LDIF 파일을 작성한 후, LDIF 가져오기 유틸리티를 수행하여 오브젝트를 LDAP 등록부로 가져오십시오. IBM LDAP 서버의 LDIF 유틸리티는 『LDIF2DB』입니다.
- IBM SecureWay LDAP 등록부 서버 버전 3.1에만 사용 가능한 DMT(Directory Management Tool)를 사용하여 사용자 오브젝트를 작성하십시오.

개인 오브젝트의 속성이 들어 있는 LDIF 파일이 다음과 유사하게 표시됩니다.

```
File name: newuser.ldif
```

```
dn: cn=Mary Burnnet, ou=DB2 UDB Development, ou=Toronto, o=ibm, c=ca
objectclass: ePerson
cn: Mary Burnnet
sn: Burnnet
uid: mburnnet
userPassword: password
telephonenumber: 1-416-123-4567
facsimiletelephonenumber: 1-416-123-4568
title: Software Developer
```

다음은 IBM LDIF 가져오기 유틸리티를 사용하여 LDIF 파일을 가져오는 LDIF 명령의 예입니다.

```
LDIF2DB -i newuser.ldif
```

주:

1. LDAP 서버 머신에서 LDIF2DB 명령을 수행해야 합니다.

2. LDAP 사용자가 자신의 오브젝트를 추가, 삭제, 읽기 및 쓸 수 있도록 LDAP 사용자 오브젝트에 필수 액세스(ACL)를 부여해야 합니다. 사용자 오브젝트의 ACL을 권한 부여하려면, LDAP 등록부 서버 웹 관리 도구를 사용하십시오.

DB2 응용프로그램용 LDAP 사용자 구성

IBM LDAP 클라이언트로 작업할 때와 DB2를 실행하기 전에, 현재 로그인 사용자에게 대한 LDAP 사용자 구별 이름(DN) 및 암호를 구성해야 합니다. 이것은 db2ldcfg 유틸리티를 사용하여 수행될 수 있습니다.

```
db2ldcfg -u <userDN> -w <password> -> set the user's DN and password
-r -> clear the user's DN and password
```

예를 들면, 다음과 같습니다.

```
db2ldcfg -u "cn=Mary Burnnet,ou=DB2 UDB Development,ou=Toronto,o=ibm,c=ca"
-w password
```

설치 후 DB2 서버의 등록

각 DB2 서버 인스턴스는 클라이언트 응용프로그램이 연결하기 위해 사용하는 프로토콜 구성 정보를 공표하기 위해 LDAP에 등록되어야 합니다. 데이터베이스 서버 인스턴스를 등록할 때, 노드 이름을 지정해야 합니다. 노드 이름은 클라이언트 응용프로그램이 서버에 연결하거나 접속할 때 사용됩니다. CATALOG LDAP NODE 명령을 사용하여 LDAP 노드에 대한 다른 별명을 카탈로그화할 수 있습니다.

주: Windows 2000 도메인 환경에서 작업하는 경우, 설치 중 DB2 서버 인스턴스는 다음 정보로 사용 중인 디렉토리에 자동으로 등록됩니다.

```
nodename: TCP/IP hostname
protocol type: TCP/IP
```

TCP/IP 호스트 이름이 8자보다 긴 경우, 8자로 잘립니다.

REGISTER 명령은 다음과 같습니다.

```
db2 register db2 server in ldap
as <ldap_node_name>
protocol tcpip
```


protocol절은 이 데이터베이스 서버에 연결할 때 사용하는 통신 프로토콜을 지정합니다.

다중 물리 머신을 포함하는 DB2 Universal Database EEE에 대한 인스턴스를 작성할 때, REGISTER 명령은 각 머신에 대해 한 번씩 호출되어야 합니다. rah 명령은 모든 머신에 대해 REGISTER 명령을 발행할 때 사용됩니다.

주: 각 머신은 LDAP에서 고유 이름을 가져야 하므로 동일한 ldap_node_name을 사용할 수 없습니다. REGISTER 명령에서 ldap_node_name에 대해 각 머신의 호스트 이름을 대체하고자 할 것입니다. 예를 들면, 다음과 같습니다.

```
rah ">DB2 REGISTER DB2 SERVER IN LDAP AS <> PROTOCOL TCP/IP"
```

『<>』는 rah 명령이 수행된 각 머신의 호스트 이름에 의해 대체됩니다. 다중 DB2 Universal Database EEE 인스턴스가 있는 드문 경우, 인스턴스와 호스트 색인 조합이 rah 명령에서 노드 이름으로 사용되는 경우가 있습니다.

원격 DB2 서버에 대해 REGISTER 명령이 발행될 수 있습니다. 이를 수행하려면, 원격 서버를 등록할 때 원격 컴퓨터 이름, 인스턴스 이름 및 프로토콜 구성 매개변수를 지정해야 합니다. 이 명령은 다음과 같이 사용될 수 있습니다.

```
db2 register db2 server in ldap
as <ldap_node_name>
protocol tcpip
hostname <host_name>
svcname <tcpip_service_name>
remote <remote_computer_name>
instance <instance_name>
```

컴퓨터 이름에 대해 다음 규칙이 사용됩니다.

- TCP/IP가 구성되어 있을 경우, 컴퓨터 이름은 TCP/IP 호스트 이름과 동일해야 합니다.
- APPN이 구성된 경우, 컴퓨터 이름으로서 상대 LU 이름을 사용하십시오.

고가용성 또는 실패복구 환경에서 수행중이고 TCP/IP를 통신 프로토콜을 사용할 경우, 클러스터 IP 주소가 사용되어야 합니다. 클러스터 IP 주소를 사용하여 클라이언트가 각 머신에 대해 별도의 TCP/IP 노드를 카탈로그화하지 않고도 머신에 있는 서버에 연결할 수 있도록 합니다. 클러스터 IP 주소는 다음과 같이 hostname 절을 사용하여 지정됩니다.

```
db2 register db2 server in ldap
as <ldap_node_name>
protocol tcpip
hostname n.nn.nn.nn
```

여기서, n.nn.nn.nn은 클러스터 IP 주소입니다.

REGISTER 명령에 관한 추가 정보는 *Command Reference*에서 자세한 내용을 참조하십시오.

DB2 서버에 대한 프로토콜 정보 갱신

LDAP에 있는 DB2 서버 정보는 현재 정보로 유지되어야 합니다. 예를 들어, 프로토콜 구성 매개변수 또는 서버 네트워크 주소를 변경하면 LDAP에 대해서도 갱신해야 합니다.

지역 머신의 LDAP에 있는 DB2 서버를 갱신하려면, 다음 명령을 사용하십시오.

```
db2 update ldap ...
```

갱신될 수 있는 프로토콜 구성 매개변수의 예는 다음을 포함합니다.

- 포트 번호 매개변수의 TCP/IP 호스트 이름 및 서비스 이름
- IPX 주소
- APPC 프로토콜 정보(예: TP명, 상대 LU 또는 모드)
- NetBIOS 워크스테이션 이름

원격 DB2 서버 프로토콜 구성 매개변수를 갱신하려면, node 절이 있는 UPDATE LDAP 명령을 사용하십시오.

```
db2 update ldap
node <node_name>
hostname <host_name>
svcname <tcpip_service_name>
```

UPDATE LDAP 명령에 대해서는 *SQL 참조서*에서 자세한 내용을 참조하십시오.

ATTACH에 대한 노드 별명 카탈로그화

DB2 서버에 대한 노드 이름은 LDAP의 서버를 등록할 때 지정되어야 합니다. 응용프로그램은 데이터베이스 서버에 접속하기 위해 노드 이름을 사용합니다. 노드 이름이 응용프로그램에서 하드 코딩된 경우와 같이 다른 노드명이 필요할 경우, CATALOG LDAP NODE 명령을 사용하여 변경하십시오. 이 명령은 다음과 같습니다.

```
db2 catalog ldap node <ldap_node_name>
as <new_alias_name>
```

LDAP 노드를 카탈로그 해제하려면, UNCATALOG LDAP NODE 명령을 사용하십시오. 이 명령은 다음과 같습니다.

```
db2 uncatalog ldap node <ldap_node_name>
```

DB2 서버 등록해제

LDAP에서 인스턴스의 등록을 취소하면 이 인스턴스에 관한 모든 노드 또는 별명, 오브젝트 및 데이터베이스 오브젝트도 제거합니다.

지역 또는 원격 머신에 있는 DB2 서버의 등록 취소는 이 서버에 대해 지정된 LDAP 노드 이름을 필요로 합니다.

```
db2 deregister db2 server in ldap
node <node_name>
```

DB2 서버가 등록 취소될 때, 이 DB2 서버의 동일 인스턴스를 참조하는 모든 LDAP 노드 항목 및 LDAP 데이터베이스 항목도 카탈로그 해제됩니다.

데이터베이스의 등록

인스턴스에서 데이터베이스를 작성하는 동안 데이터베이스는 자동으로 LDAP에 등록됩니다. 이 등록은 지역 머신에 데이터베이스 및 노드를 카탈로그화하지 않고도 원격 클라이언트가 데이터베이스에 연결할 수 있게 합니다. 클라이언트가 데이터베이스에 연결하려 할 때, 데이터베이스가 지역 머신의 데이터베이스 디렉토리에 존재하지 않으면, LDAP 등록부가 검색됩니다.

이 이름이 LDAP 등록부에 이미 있을 경우, 데이터베이스가 지역 머신에서 여전히 작성되지만 LDAP 등록부에 이름 지정 충돌이 있음을 나타내는 경고 메시지가 리턴됩니다. 이러한 이유로 LDAP 등록부에서 데이터베이스를 수작업으로 카탈로그화할 수 있습니다. 사용자는 CATALOG LDAP DATABASE 명령을 사용하여 LDAP에 원격 서버의 데이터베이스를 등록할 수 있습니다. 원격 데이터베이스를 등록할 때, 원격 데이터베이스 서버를 나타내는 LDAP 노드의 이름을 지정합니다. 데이터베이스를 등록하기 전에 REGISTER DB2 SERVER IN LDAP 명령을 사용하여 원격 데이터베이스 서버를 LDAP에 등록해야 합니다.

LDAP에 데이터베이스를 수작업으로 등록하려면, CATALOG LDAP DATABASE 명령을 사용하십시오.

```
db2 catalog ldap database <dbname>
      at node <node_name>
      with "My LDAP database"
```

원격 서버에 접속

LDAP 환경에서, ATTACH 명령에 LDAP 노드 이름을 사용하여 원격 데이터베이스 서버에 접속할 수 있습니다.

```
db2 attach to <ldap_node_name>
```

클라이언트 응용프로그램이 처음으로 노드 또는 데이터베이스에 연결할 때, 노드가 지역 노드 디렉토리에 있지 않으므로 DB2는 목표 노드 항목에 대해 LDAP 등록부를 검색합니다. LDAP 등록부에서 항목이 발견되지 않으면, 원격 서버의 프로토콜 정보가 검색됩니다. 데이터베이스에 연결하며 항목이 LDAP 등록부에 있는 경우, 데이터베이스 정보도 검색됩니다. 이 정보를 사용하여 DB2가 자동으로 지역 머신에 데이터베이스 항목 및 노드 항목을 카탈로그화합니다. 클라이언트 응용프로그램이 다음 번에 같은 노드 또는 데이터베이스에 접속할 때에는 LDAP 등록부를 검색하지 않고 지역 데이터베이스 디렉토리에 있는 정보를 사용합니다.

자세히 설명하면, 클라이언트가 관련 지역 디렉토리 카탈로그에서 LDAP 등록부만 한 번 검색하도록, 캐쉬하는 메커니즘이 있습니다. 정보가 검색되고 나면 지역 머신에 저장되거나 캐쉬됩니다. 동일한 정보에 대한 차후 액세스는 *dir_cache* 데이터베이스 관리 프로그램 구성 매개변수와 DB2LDAPCACHE 레지스트리 변수의 값에 근거합니다.

- DB2LDAPCACHE=NO이고 *dir_cache*=NO이면, 항상 LDAP에서 정보를 읽습니다.
- DB2LDAPCACHE=NO이고 *dir_cache*=YES이면, LDAP에서만 정보를 읽어 DB2 캐쉬로 삽입합니다.
- DB2LDAPCACHE=YES이거나 설정되지 않고, 필수 정보가 지역 캐쉬에 없는 경우, LDAP 등록부에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: LDAP 정보의 캐쉬는 사용자 레벨 CLI 또는 DB2 프로파일 레지스트리 변수에 적용할 수 없습니다. 또한, 데이터베이스, 노드 및 DCS 디렉토리에 대한 『메모리 내』 캐쉬가 있습니다. 그러나, 단지 노드 디렉토리에 대해 캐쉬되는 것은 없습니다.

데이터베이스 등록 해제

데이터베이스는 다음의 경우 LDAP로부터 자동으로 등록 해제됩니다.

- 데이터베이스가 삭제된 경우
- 소유하는 인스턴스가 LDAP로부터 등록 해제된 경우

다음 명령을 사용하여 수동으로 LDAP로부터 데이터베이스를 등록 취소할 수 있습니다.

```
db2 uncatalog ldap database <dbname>
```

지역 데이터베이스 및 노드 디렉토리에 있는 LDAP 항목 새로 고침

LDAP 정보는 변경되기 쉬우므로 지역 및 노드 디렉토리에 있는 LDAP 항목을 새로 고칠 필요가 있습니다. 지역 데이터베이스 및 노드 디렉토리는 LDAP에 있는 항목을 캐쉬하는 데 사용됩니다.

자세히 설명하면, 클라이언트가 관련 지역 디렉토리 카탈로그에서 LDAP 등록부만 한 번 검색하도록, 캐쉬하는 메커니즘이 있습니다. 정보가 검색되고 나면 지역 머신에 저장되거나 캐쉬됩니다. 동일한 정보에 대한 차후 액세스는 *dir_cache* 데이터베이스 관리 프로그램 구성 매개변수와 DB2LDAPCACHE 레지스트리 변수의 값에 근거합니다.

- DB2LDAPCACHE=NO이고 *dir_cache*=NO이면, 항상 LDAP에서 정보를 읽습니다.
- DB2LDAPCACHE=NO이고 *dir_cache*=YES이면, LDAP에서만 정보를 읽어 DB2 캐쉬로 삽입합니다.
- DB2LDAPCACHE=YES이거나 설정되지 않고, 필수 정보가 지역 캐쉬에 없는 경우, LDAP 등록부에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: LDAP 정보의 캐쉬는 사용자 레벨 CLI 또는 DB2 프로파일 레지스트리 변수에 적용할 수 없습니다. 또한, 데이터베이스, 노드 및 DCS 디렉토리에 대한 『메모리 내』 캐쉬가 있습니다. 그러나, 단지 노드 디렉토리에 대해 캐쉬되는 것은 없습니다.

LDAP 자원을 참조하는 데이터베이스 항목을 새로 고치려면, 다음 명령을 사용하십시오.

```
db2 refresh ldap database directory
```

LDAP 자원을 참조하는 지역 머신의 노드 항목을 새로 고치려면, 다음 명령을 사용하십시오.

```
db2 refresh ldap node directory
```

새로 고치는 작업의 일부로서, 지역 데이터베이스 및 노드 디렉토리에 저장되어 있는 모든 LDAP 항목이 제거됩니다. 다음에 응용프로그램이 데이터베이스 또는 노드에 액세스할 때, 응용프로그램은 LDAP로부터 직접 정보를 읽고 지역 데이터베이스 또는 노드 디렉토리에 새로운 항목을 생성합니다.

시기 적절한 방법으로 새로 고침이 수행되었는지 확인하기 위해 다음을 수행하고자 할 수 있습니다.

- 정기적으로 수행되는 새로 고침을 스케줄함
- 시스템 시동시 REFRESH 명령을 수행함
- 모든 클라이언트 머신에 대해 REFRESH 명령을 호출하기 위해 사용 가능한 관리 패키지를 사용함
- DB2LDAPCACHE=『NO』를 설정하여 데이터베이스, 노드 및 DCS 디렉토리에 LDAP 정보가 캐쉬되는 것을 피할 수 있습니다.

검색

DB2는 현재 LDAP 등록부 파티션 및 현재 사용중인 디렉토리 도메인을 Windows 2000 환경에서 검색합니다. 여러 LDAP 등록부 파티션 또는 도메인이 있는 환경에서 검색 영역을 설정할 수 있습니다. 예를 들어, 현재 파티션 또는 도메인에서 정보가 발견되지 않으면, 모든 파티션 또는 도메인의 자동 검색이 요청될 수 있습니다. 반면에 검색 영역이 지역 머신만 검색하도록 제한될 수도 있습니다.

검색 영역은 DB2 프로파일 레지스트리 변수인 DB2LDAP_SEARCH_SCOPE를 통해 제어됩니다. 검색 영역 값을 LDAP의 전역 레벨에서 설정하려면, `db2set` 명령에 『-gl』 옵션을 사용하십시오. 이 옵션은 『LDAP의 전역』을 나타냅니다.

```
db2set -gl db2ldap_search_scope=<value>
```

사용 가능한 값은 『local』, 『domain』 또는 『global』을 포함합니다. 기본값은 현재 디렉토리 파티션으로 검색 범위를 제한하는 『domain』입니다. LDAP의 검색 영역을 설정하면 전체 엔터프라이즈에 대한 기본 검색 영역을 설정할 수 있습니다. 예를 들어, 새로운 데이터가 작성된 다음에 검색 영역을 『global』로 초기화하고자 할 수 있습니다. 이는 모든 클라이언트 머신이 다른 모든 파티션이나 도메인을 검색하게 하여 특정 파티션 또는 도메인에 정의된 데이터베이스를 찾을 수 있도록 합니다. 각 클라이언트에 대한 최초 연결이나 접속 후 항목이 각 머신에 기록된 다음 검색 영역을 『local』로 변경할 수 있습니다. 일단 『local』로 변경되면, 각 클라이언트는 어떠한 파티션 또는 도메인도 스캔하지 않습니다.

주: DB2 프로파일 레지스트리 변수 DB2LDAP_SEARCH_SCOPE만 LDAP의 전역 레벨에 변수 설정을 지원하는 레지스트리 변수입니다.

호스트 데이터베이스 구성

호스트 데이터베이스를 LDAP에 등록할 때, 두개의 가능한 구성이 있습니다.

- 호스트 데이터베이스에 대한 직접 연결
- 게이트웨이를 통한 호스트 데이터베이스에 대한 연결.

처음 경우에, 사용자는 LDAP에서 호스트 서버를 등록한 후, 호스트 서버의 노드 이름을 지정하는 LDAP에 호스트 데이터베이스를 카탈로그합니다. 두번째 경우에,

사용자는 LDAP에서 게이트웨이 서버를 등록한 후, 게이트웨이 서버의 노드 이름을 지정하는 LDAP에 호스트 데이터베이스를 카탈로그합니다.

양쪽 경우를 표시하는 예로서, 다음을 고려하십시오. NIAGARA_FALLS라고 하는 호스트 데이터베이스가 있음을 가정하십시오. APPN 및 TCP/IP를 사용하여 들어오는 연결을 승인할 수 있습니다. 클라이언트가 DB2 Connect를 가지고 있지 않아서 호스트에 직접 연결할 수 없는 경우, 『goto@niagara』라고 하는 게이트웨이를 사용하여 연결합니다.

다음 단계를 수행할 필요가 있습니다.

1. APPN 연결성을 위해서는 호스트 데이터베이스 서버를 LDAP에 등록하십시오. REMOTE 및 INSTANCE 절은 임의입니다. NODETYPE 절은 『DCS』로 설정되어 이것이 호스트 데이터베이스 서버임을 나타냅니다.

```
db2 register ldap as nfappn appn network CAIBMOML partnerlu NFLU
mode IBMRDB remote mvssys instance msvinst nodetype dcs
```

2. TCP/IP 연결성을 위해서는 호스트 데이터베이스 서버를 LDAP에 등록하십시오. 서버의 TCP/IP 호스트 이름은 『myhost』이며 포트 번호는 『446』입니다. 단계 1과 유사하며, NODETYPE 절은 『DCS』로 설정되어 이것이 호스트 데이터베이스 서버임을 나타냅니다.

```
db2 register ldap as nftcpip tcpip hostname myhost svcename 446
remote mvssys instance mvsinst nodetype dcs
```

3. TCP/IP 연결성을 위해서는 DB2 Connect 게이트웨이 서버를 LDAP에 등록하십시오. 게이트웨이 서버의 TCP/IP 호스트 이름은 『niagara』이며 포트 번호는 『50000』입니다.

```
db2 register ldap as whasf tcpip hostname niagara svcename 50000
remote niagara instance goto nodetype server
```

4. TCP/IP 연결성을 사용하여 LDAP에서 호스트 데이터베이스를 카탈로그화 하십시오. 호스트 데이터베이스 이름은 『NIAGARA_FALLS』이며, 데이터베이스 별명 이름은 『nftcpip』입니다. GWNODE 절은 DB2 Connect 게이트웨이 서버의 노드 이름을 지정하기 위해 사용됩니다.

```
db2 catalog ldap database NIAGARA_FALLS as nftcpip at node nftcpip
gwnode whasf authentication dcs
```


5. APPN 연결성을 사용하여 LDAP에서 호스트 데이터베이스를 카탈로그하십시오.

```
db2 catalog ldap database NIAGARA_FALLS as nfappn at node nfappn
gwnode whasf authentication dcs
```

위에 표시한 등록 및 카탈로그를 완료한 후에, TCP/IP를 사용하여 호스트에 연결하려는 경우 『nftcpip』에 연결합니다. APPN을 사용하여 호스트에 연결하려는 경우, 『nfappn』에 연결합니다. 클라이언트 워크스테이션에 DB2 Connect를 갖지 않은 경우, TCPIP를 사용하여 게이트웨이를 통해 연결이 진행되며, 거기에서 『nftcpip』 또는 『nfappn』을 사용하는 지 여부에 따라 TCP/IP 또는 APPN을 각각 사용하여 호스트에 연결합니다.

각 클라이언트가 수작업으로 데이터베이스와 노드를 각 머신에 지역적으로 카탈로그화할 필요가 없도록 LDAP에 호스트 데이터베이스 정보를 수작업으로 구성할 수 있습니다. 이 프로세스는 다음과 같습니다.

1. 호스트 데이터베이스 서버를 LDAP에 등록하십시오. REMOTE, INSTANCE 및 NODETYPE 절을 각각 사용하여 REGISTER 명령에서 호스트 데이터베이스 서버에 대해 원격 컴퓨터 이름, 인스턴스 이름 및 노드 유형을 지정해야 합니다. REMOTE 절은 호스트 이름이나 호스트 서버 머신의 LU 이름 중 하나로 설정될 수 있습니다. INSTANCE 절은 8문자 이하인 문자열로 설정될 수 있습니다. 예를 들어, 인스턴스 이름은 『DB2』로 설정될 수 있습니다. NODE TYPE 절은 『DCS』로 설정되어야 이것이 호스트 데이터베이스 서버임을 나타낼 수 있습니다.
2. CATALOG LDAP DATABASE 명령을 사용하여 호스트 데이터베이스를 LDAP에 등록하십시오. PARMS 절을 사용하여 DRDA 매개변수를 추가 지정할 수 있습니다. 데이터베이스 인증 유형은 『DCS』로 설정되어야 합니다.

사용자 레벨에서 DB2 레지스트리 변수 설정

LDAP 환경하에서, DB2 프로파일 레지스트리 변수는 사용자의 DB2 환경을 사용자 정의할 수 있는 사용자 레벨에 설정될 수 있습니다. DB2 프로파일 레지스트리 변수를 사용자 레벨에 설정하려면, -ul 옵션을 사용하십시오.

```
db2set -ul <variable>=<value>
```

주: 이것은 AIX 또는 Solaris에서 지원되지 않습니다.

DB2에는 캐쉬하는 메커니즘이 있습니다. 사용자 레벨의 DB2 프로파일 레지스트리 변수는 지역 머신에서 캐쉬됩니다. -u1 매개변수가 지정된 경우, DB2는 항상 캐쉬에서 DB2 레지스트리 변수를 읽어옵니다. 이 캐쉬는 다음 경우에 새로 고쳐 집니다.

- 사용자 레벨에서 DB2 레지스트리 변수를 갱신하거나 재설정하는 경우
- 서버 레벨에서 LDAP 프로파일 변수를 새로 고치는 명령은 다음과 같습니다.

```
db2set -ur
```

설치 완료 후 LDAP 지원 작동 가능

설치 프로세스 완료에 이어 일부 지점에서 LDAP 지원을 작동 가능하게 하려면, 각 머신에서 다음 프로시저를 사용하십시오.

- LDAP 지원 2진 파일을 설치하십시오. 설치 프로그램을 수행하여 사용자 설치로부터 LDAP 등록부 이용 지원을 선택하십시오. 설치 프로그램이 2진 파일을 설치하고 DB2 프로파일 레지스트리 변수 DB2_ENABLE_LDAP를 『YES』로 설정합니다.

주: Windows 95/98/NT 및 UNIX 플랫폼의 경우, **db2set** 명령을 사용하여 DB2_ENABLE_LDAP 레지스트리 변수를 『YES』로 설정하여 명시적으로 LDAP를 사용 가능하게 해야 합니다.

- (UNIX 플랫폼에서만) 다음 명령을 사용하여 LDAP 서버의 TCP/IP 호스트 이름 및 (선택적인) 포트 번호를 선언하십시오.

```
db2set DB2LDAPHOST=<base_domain_name>[:port_number]
```

여기서 base_domain_name은 LDAP 서버의 TCP/IP 호스트 이름이며, [:port]는 포트 번호입니다. 포트 번호가 지정되지 않으면, DB2는 기본 LDAP 포트 (389)를 사용합니다.

DB2 오브젝트는 LDAP 기본 구분 이름(baseDN)에 위치합니다. IBM SecureWay LDAP 등록부 서버 버전 3.1을 사용 중인 경우, DB2는 서버에서 동적으로 이 정보를 얻을 수 있으므로 기본 구분 이름을 구성할 필요는 없습니

다. 그러나 IBM eNetwork 등록부 서버 버전 2.1을 사용하는 경우, DB2SET 명령을 사용하여 각 머신의 LDAP 기본 구분 이름을 구행해야 합니다.

```
db2set DB2LDAP_BASEDN=<baseDN>
```

여기서 baseDB은 LDAP 서버에 정의된 LDAP 접미부의 이름입니다. 이 LDAP 접미부는 DB2 오브젝트를 포함하는 데 사용됩니다.

- REGISTER LDAP AS 명령을 사용하여 LDAP에 DB2 서버의 현재 인스턴스를 등록하십시오. 예를 들면, 다음과 같습니다.

```
db2 register ldap as <node-name> protocol tcpip
```

- LDAP에 등록하고 싶은 데이터베이스가 있으면, CATALOG LDAP DATABASE 명령을 수행하십시오. 예를 들면, 다음과 같습니다.

```
db2 catalog ldap database <dbname> as <alias_dbname>
```

- LDAP 사용자의 구별된 이름(DN) 및 암호. 이것은 사용자가 LDAP를 사용하여 DB2 사용자 고유 정보를 저장하려고 계획하는 경우에만 필요합니다.

LDAP 지원 작동 불가능

LDAP 지원을 작동 불가능하게 하려면, 다음 프로시저어를 사용하십시오.

- DB2 서버의 각 인스턴스의 경우, LDAP로부터 DB2 서버를 등록 해제하십시오.

```
db2 deregister db2 server in ldap node <nodename>
```

- DB2 프로파일 레지스트리 변수 DB2_ENABLE_LDAP를 『NO』로 설정하십시오.

LDAP 지원 및 DB2 연결

LDAP 지원이 DB2 Connect 게이트웨이에서 사용 가능하며 데이터베이스가 게이트웨이 데이터베이스 디렉토리에 없는 경우, DB2는 LDAP를 찾아보며 찾은 정보를 유지하려고 시도합니다.

보안 고려사항

LDAP 등록부에 있는 정보를 액세스하기 전에, 응용프로그램 또는 사용자가 LDAP 서버에 의해 인증됩니다. 이 인증 프로세스는 LDAP 서버에 대한 바인딩이라고 합니다.

불분명한 사용자가 정보를 추가, 삭제 또는 수정하는 것을 막기 위해 LDAP 등록부에 저장된 정보에 대해 액세스 제어를 적용하는 것은 중요한 일입니다.

액세스 제어는 기본적으로 계승되며 컨테이너 레벨에서 적용될 수 있습니다. 새로운 오브젝트가 작성될 때, 이것은 상위 오브젝트와 동일한 보안 속성을 계승합니다. LDAP 서버에 사용할 수 있는 관리 도구를 사용하여 컨테이너 오브젝트에 대한 액세스 제어를 정의할 수 있습니다.

기본적으로, 액세스 제어는 다음과 같이 정의됩니다.

- LDAP에 있는 데이터베이스 및 노트 항목의 경우, 모든 사용자(또는 모든 불분명한 사용자)가 읽기 액세스 권한을 가집니다. 디렉토리 관리자 및 오브젝트의 소유자 또는 작성자만 읽기/쓰기 액세스 권한을 가집니다.
- 사용자 프로파일의 경우, 프로파일 소유자와 디렉토리 관리자는 읽기/쓰기 액세스 권한을 가집니다. 디렉토리 관리자 권한을 가지고 있지 않은 사용자는 다른 사용자의 프로파일에 액세스할 수 없습니다.

주: 권한 부여 점검은 LDAP 서버에 의해 항상 수행되며, DB2에 의해서는 수행되지 않습니다. LDAP 권한 부여 점검은 DB2 권한 부여와는 관계가 없습니다. SYSADM 권한을 가지는 계정 또는 권한 부여 ID는 LDAP 등록부에 대해 액세스 권한이 없을 수도 있습니다.

LDAP 명령 또는 API를 실행할 때, 바인드 구분 이름(bindDN) 및 암호가 지정되지 않은 경우, DB2는 요청된 명령을 수행하는 충분한 권한이 없을 수 있는 기본 자격사항을 사용하여 LDAP 서버에 바인딩하여 오류를 리턴합니다.

DB2 명령 또는 API에 대해 USER 및 PASSWORD 절을 사용하여 사용자의 bindDN 및 암호를 명시적으로 지정할 수 있습니다. *Command Reference*에서 DB2 명령에 대한 자세한 내용을 참조하고 *Administrative API Reference*에서 DB2 API에 대한 자세한 내용을 참조하십시오.

Windows 2000에서 사용중인 디렉토리 구성 고려사항

DB2 데이터베이스 및 노트 오브젝트는 사용중인 디렉토리에 DB2 서버가 설치된 머신의 컴퓨터 오브젝트 아래에 작성됩니다. 사용중인 디렉토리에서 데이터베이스 서버를 등록하거나 데이터베이스를 카탈로그화하려면, 컴퓨터 오브젝트에서 오브젝트를 작성하고/하거나 갱신하기에 충분한 액세스를 가질 필요가 있습니다.

기본값으로, 컴퓨터 오브젝트 아래에 있는 오브젝트는 권한이 부여된 사용자가 읽을 수 있으며 관리자(관리자, 도메인 관리자 및 엔터프라이즈 관리자 그룹에 속하는 사용자)가 갱신할 수 있습니다. 특정 사용자나 그룹에 대한 액세스를 권한 부여하려면, 다음과 같이 사용중인 디렉토리 사용자 및 컴퓨터 관리 콘솔(MMC)을 사용하십시오.

1. 사용중인 디렉토리 사용자 및 컴퓨터 관리 도구를 시작하십시오
(시작→ 프로그램→ 관리 도구→ 사용 중인 디렉토리 사용자 및 컴퓨터)
2. 보기 아래에서, 고급 기능을 선택하십시오
3. 컴퓨터 컨테이너를 선택하십시오
4. DB2가 설치된 서버 머신을 나타내는 컴퓨터 오브젝트를 마우스 오른쪽 버튼으로 누르고 등록 정보를 선택하십시오
5. 보안 탭을 선택한 후, 지정된 사용자나 그룹에 필수 액세스를 추가하십시오

사용자 레벨에서의 DB2 레지스트리 변수 및 CLI 설정은 사용자 오브젝트 아래의 DB2 등록 정보 오브젝트에 유지보수됩니다. 사용자 레벨에서 DB2 레지스트리 변수 또는 CLI 설정을 설정하려면, 사용자가 사용자 오브젝트 아래에서 오브젝트를 작성할 충분한 액세스를 가져야 합니다.

기본값으로, 관리자만이 사용자 오브젝트 아래에서 오브젝트를 작성할 액세스를 갖습니다. 사용자 레벨에서 DB2 레지스트리 변수나 CLI 설정을 설정하게 하는 액세스 권한을 부여하려면, 다음과 같이 사용 중인 디렉토리 사용자 및 컴퓨터 관리 콘솔(MMC)을 사용하십시오.

1. 사용 중인 디렉토리 사용자 및 컴퓨터 관리 도구를 시작하십시오
(시작→ 프로그램→ 관리 도구→ 사용 중인 디렉토리 사용자 및 컴퓨터)
2. 오브젝트 컨테이너에서 사용자 오브젝트를 선택하십시오.
3. 사용자 오브젝트를 마우스 오른쪽 버튼으로 누르고 등록 정보를 선택하십시오

4. 보안 탭을 선택하십시오
5. 『쓰기』 및 『모든 하위 오브젝트 작성』 액세스를 자체에 추가하십시오
6. 『이 오브젝트로 전파하도록 상위에서 상속가능한 사용권한 허용』 선택란을 선택하십시오

DB2 오브젝트 클래스 및 속성이 있는 디렉토리 스키마 확장

LDAP 등록부 스키마는 LDAP 등록부 항목에 저장된 정보에 대해 오브젝트 클래스 및 속성을 정의합니다. 오브젝트 클래스는 필수 및 선택적 속성 세트로 이루어져 있습니다. LDAP 등록부의 모든 항목은 연관된 오브젝트 클래스를 가집니다.

DB2가 LDAP에 정보를 저장하려면, 먼저 LDAP 서버에 대한 디렉토리 스키마가 DB2가 사용하는 오브젝트 클래스와 속성을 포함해야 합니다. 새로운 오브젝트 클래스 및 속성을 기본 스키마에 추가하는 프로세스를 디렉토리 스키마 확장이라고 합니다.

주: IBM SecureWay LDAP Directory v3.1을 사용할 경우, DB2가 필요로 하는 모든 오브젝트 클래스 및 속성은 이 기본 스키마에 들어 있습니다. DB2 오브젝트 클래스 및 속성이 있는 기본 스키마를 확장할 필요는 없습니다.

IBM eNetwork Directory 버전 2.10에 대해 디렉토리 스키마 확장

IBM eNetwork Directory 버전 2.1을 사용할 때, DB2가 사용하는 오브젝트 클래스와 속성이 있는 기본 스키마를 확장시켜야 합니다.

다음 단계를 사용하여 IBM eNetwork Directory 버전 2.1에 대한 기본 스키마를 확장시키십시오.

1. DB2 속성 정의 파일 db2.at와 오브젝트 클래스 정의 파일 db2.oc를 시스템 속성 및 오브젝트 클래스 정의 파일 slapd.at.conf와 slapd.oc.conf가 있는 같은 디렉토리에 복사하십시오. DB2 속성 및 오브젝트 클래스 정의 파일은 sqllib 서브디렉토리의 cfg 서브디렉토리에서 찾을 수 있습니다. 시스템 속성 및 오브젝트 클래스 정의 파일은 %LDAPHome% 서브디렉토리의 etc 서브디렉토리에 있습니다.

2. DB2 속성 및 오브젝트 클래스 정의 파일을 검토하십시오. 사용자의 현재 LDAP 등록부 스키마에 정의되어 있던 모든 오브젝트 클래스 및 속성에 주석 표시를 하십시오.
3. slapd.oc.conf 파일 끝에 다음과 같은 행을 추가하십시오.


```
include db2.oc
```
4. slapd.at.conf 파일 끝에 다음과 같은 행을 추가하십시오.


```
include db2.at
```
5. LDAP 서버를 재시작하십시오.

Windows 2000 Active Directory 버전 2.1에 대해 디렉토리 스키마 확장

DB2가 Windows 2000 Active Directory에 정보를 저장하기 전에, 디렉토리 스키마가 새로운 DB2 오브젝트 클래스 및 속성을 포함하도록 확장될 필요가 있습니다. 새로운 오브젝트 클래스 및 속성을 디렉토리 스키마에 추가하는 프로세스를 스키마 확장이라고 합니다.

Windows 2000 도메인의 일부인 임의 머신에서 DB2의 처음 설치 이전에 DB2 스키마 설치 프로그램 **db2schex**을 실행하여 사용 중인 디렉토리에 대한 스키마를 확장해야 합니다.

db2schex 프로그램은 CD-ROM 제품에 있습니다. CD-ROM에서 이 프로그램의 위치는 db2 디렉토리 및 common 서브디렉토리 아래에 있습니다. 예를 들면, 다음과 같습니다.

```
x:\db2\common
```

여기서 x:는 CD-ROM 드라이브입니다.

명령은 다음과 같이 사용됩니다.

```
db2schex
```

이 명령에 관련된 기타 선택적 절이 있습니다.

- **-b UserDN**
사용자 구분 이름을 지정

- -w 암호
바인드 암호를 지정
- -u
스키마를 설치 제거
- -k
오류를 무시하고 설치 제거를 계속 강제 수행하려면.

주:

1. UserDN 및 암호가 지정되지 않은 경우, **db2schex**는 현재 로그인 사용자로
서 바인드합니다.
2. userDN 절은 Windows NT 사용자 이름으로서 지정될 수 있습니다.
3. 스키마를 갱신하려면, 스키마 관리자의 구성원이어야 하거나 스키마를 갱신하
기 위한 권한이 위임되어 있어야 합니다.

예:

- DB2 스키마를 설치하려면.
`db2schex`
- DB2 스키마를 설치하고 바인드 DN 및 암호를 지정하려면.
`db2schex -b "cn=A Name,dc=toronto1,dc=ibm,dc=com"
-w password`

또는,

- `db2schex -b Administrator -w password`
- DB2 스키마를 설치 제거하려면.
`db2schex -u`
- DB2 스키마를 설치 제거하고 오류를 무시하려면.
`db2schex -u -k`

활성 디렉토리의 DB2 스키마 설치 프로그램은 다음과 같은 작업을 수행합니다.

주:

1. 어느 서버가 스키마 마스터인지 검출
2. 스키마 마스터인 도메인 제어기에 바인드

3. 사용자가 클래스와 속성을 스키마에 추가할 수 있는 권한을 가지고 있는지 확인
4. 스키마 마스터가 쓰기가 가능한지 확인(즉, 레지스트리에서 안전 interlock이 제거됨)
5. 새로운 속성 작성
6. 새로운 오브젝트 분류 작성
7. 오류를 검출하고, 오류가 발생하면 프로그램은 스키마에 대한 변경사항을 구간 복원

Windows 2000 Active Directory에 있는 DB2 오브젝트

DB2는 두 위치에서 사용 중인 디렉토리에 오브젝트를 작성합니다.

1. DB2 데이터베이스 및 노드 오브젝트는 DB2 서버가 설치된 머신의 컴퓨터 오브젝트 아래에 작성됩니다. Windows NT 도메인에 속하지 않는 DB2 서버 머신의 경우, DB2 데이터베이스 및 노드 오브젝트는 『시스템』 컨테이너 아래에 작성됩니다.
2. 사용자 레벨에서의 DB2 레지스트리 변수 및 CLI 설정은 사용자 오브젝트 아래의 DB2 등록 정보 오브젝트에 저장됩니다. 이 오브젝트에는 해당 사용자에게 고유한 정보가 들어 있습니다.

DB2가 사용하는 오브젝트 클래스 및 속성

다음 표에서는 DB2에서 사용되는 오브젝트 클래스를 설명합니다.

표 24. *cimManagedElement*

클래스	cimManagedElement
사용중인 디렉토리 LDAP 표시장치명	적용할 수 없음
사용중인 디렉토리 공통 이름(cn)	적용할 수 없음
설명	많은 시스템 관리 오브젝트 클래스의 기본 클래스를 IBM 스키마에서 제공합니다.
SubClassOf	top
필수 속성(s)	
선택적 속성(s)	설명
유형	abstract
OID(오브젝트 식별자)	1.3.18.0.2.6.132

표 24. *cimManagedElement* (계속)

클래스	cimManagedElement
GUID(전역 고유 식별자)	b3afd63f-5c5b-11d3-b818-002035559151

표 25. *cimSetting*

클래스	cimSetting
사용중인 디렉토리 LDAP 표시장치 이름	적용할 수 없음
사용중인 디렉토리 공통 이름(cn)	적용할 수 없음
설명	구성 및 설정의 기본 클래스를 IBM 스키마에서 제공합니다.
SubClassOf	cimManagedElement
필수 속성(s)	
선택적 속성(s)	settingID
유형	abstract
OID(오브젝트 식별자)	1.3.18.0.2.6.131
GUID(전역 고유 식별자)	b3afd64d-5c5b-11d3-b818-002035559151

표 26. *eProperty*

클래스	eProperty
사용중인 디렉토리 LDAP 표시장치 이름	ibm-eProperty
사용중인 디렉토리 공통 이름(cn)	ibm-eProperty
설명	사용자 선호 등록 정보에 대한 응용프로그램 고유의 설정을 지정하는 데 사용됩니다.
SubClassOf	cimSetting
필수 속성(s)	
선택적 속성(s)	propertyType cisPropertyType cisProperty cesPropertyType cesProperty binPropertyType binProperty

표 26. *eProperty* (계속)

클래스	eProperty
유형	structural
OID(오브젝트 식별자)	1.3.18.0.2.6.90
GUID(전역 고유 식별자)	b3afd69c-5c5b-11d3-b818-002035559151

표 27. *DB2Node*

클래스	DB2Node
사용 중인 디렉토리 LDAP 표시장치 이름	ibm-db2Node
사용 중인 디렉토리 공통 이름(cn)	ibm-db2Node
설명	DB2 서버 설명
SubClassOf	eSap / ServiceConnectionPoint
필수 속성(s)	db2nodeName
선택적 속성(s)	db2nodeAlias db2instanceName db2Type 호스트 / dNSHostName(참고 2 참조) protocolInformation/ServiceBindingInformation
유형	structural
OID(오브젝트 식별자)	1.3.18.0.2.6.116
GUID(전역 고유 식별자)	b3afd65a-5c5b-11d3-b818-002035559151
특별 주의사항	<ol style="list-style-type: none"> 1. <i>DB2Node</i> 클래스는 IBM SecureWay 등록부 아래의 <i>eSap</i> 오브젝트 클래스와 Microsoft Active Directory 아래의 <i>ServiceConnectionPoint</i> 오브젝트에서 파생됩니다. 2. 호스트는 IBM SecureWay 환경에서 사용됩니다. <i>dNSHostName</i> 속성은 Microsoft Active Directory 아래에서 사용됩니다. 3. <i>protocolInformation</i>은 IBM SecureWay 환경 아래에서만 사용됩니다. Microsoft Active Directory의 경우, <i>ServiceConnectionPoint</i> 클래스에서 계승된 <i>ServiceBindingInformation</i> 속성은 프로토콜 정보를 포함하기 위해 사용됩니다.

DB2Node 오브젝트에 있는 *protocolInformation*(IBM SecureWay Directory에서) 또는 *ServiceBindingInformation*(Microsoft Active Directory에서) 속성에는 DB2 데이터베이스 서버에 바인드할 통신 프로토콜 정보가 들어 있습니다. 지원되는 네트워크 프로토콜을 설명하는 토큰으로 구성됩니다. 각 토큰은 세미콜론으로 구분됩니다. 토큰 사이에는 공간을 두지 않습니다. 별표(*)는 선택적 매개변수를 지정하기 위해 사용될 수 있습니다.

TCP/IP에 대한 토큰은 다음과 같습니다.

- 『TCPIP』
- 서버 호스트 이름 또는 IP 주소
- 서비스 이름(svcname) 또는 포트 번호 (e.g. 50000)
- (선택적) 보안(『NONE』 또는 『SOCKS』)

APPN에 대한 토큰은 다음과 같습니다.

- 『APPN』
- 네트워크 ID
- 상대 LU
- 트랜잭션 프로그램(TP) 이름(지원 응용프로그램 TP만이 서비스 TP(HEX의 TP)를 지원하지 않습니다)
- 모드
- 보안(『NONE』, 『SAME』 또는 『PROGRAM』)
- (선택적) LAN 어댑터 주소
- (선택적) 암호 변경 LU

주: Windows NT용 DB2 클라이언트에서(또는 Windows 95 또는 Windows 98의 경우), APPN 정보가 지역 SNA 스택에서 구성되지 않은 경우, 그리고 LAN 어댑터 주소 및 선택적 변경 암호 LU가 LDAP에 있는 경우 DB2 클라이언트는 이 정보를 사용하여 스택 구성 방법을 아는 경우 SNA 스택을 구성합니다. 이 지원은 AIX용 DB2 또는 Solaris용 DB2 클라이언트에서 사용 가능하지 않습니다.

IPX/SPX에 대한 토큰은 다음과 같습니다.

- 『IPXSPX』

- IPX 주소

IPX/SPX 대기자는 AIX 및 Solaris용 DB2 서버(클라이언트가 아닌)에서 사용 가능합니다. NetBIOS 및 NPIPE는 AIX 및 Solaris에서 지원되지 않습니다.

NETBIOS에 대한 토큰은 다음과 같습니다.

- 『NETBIOS』

- 서버 NetBIOS 워크스테이션 이름

Named Pipe에 대한 토큰은 다음과 같습니다.

- 『NPIPE』

- 서버의 컴퓨터명

- 서버의 인스턴스명

표 28. DB2Database

클래스	DB2Database
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2Database
사용중인 디렉토리 공통 이름(cn)	ibm-db2Database
설명	DB2 데이터베이스 설명
SubClassOf	top
필수 속성(s)	db2databaseName db2nodePtr
선택적 속성(s)	db2databaseAlias db2additionalParameters db2ARLibrary db2authenticationLocation db2gwPtr db2databaseRelease DCEPrincipalName
유형	structural
OID(오브젝트 식별자)	1.3.18.0.2.6.117

표 28. DB2Database (계속)

클래스	DB2Database
GUID(전역 고유 식별자)	b3afd659-5c5b-11d3-b818-002035559151

표 29. db2additionalParameters

속성	db2additionalParameters
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2AdditionalParameters
사용중인 디렉토리 공통 이름(cn)	ibm-db2AdditionalParameters
설명	호스트 데이터베이스 서버에 연결할 때 사용되는 모든 추가 매개변수 포함
구문	대소문자 비구분 문자열
최대 길이	1024
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.426
GUID(전역 고유 식별자)	b3afd315-5c5b-11d3-b818-002035559151

표 30. db2authenticationLocation

속성	db2authenticationLocation
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2AuthenticationLocation
사용중인 디렉토리 공통 이름(cn)	ibm-db2AuthenticationLocation
설명	인증이 발생하는 곳 지정
구문	대소문자 비구분 문자열
최대 길이	64
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.425
GUID(전역 고유 식별자)	b3afd317-5c5b-11d3-b818-002035559151
주의사항	유효한 값: CLIENT, SERVER, DCS, DCE, KERBEROS, SVRENCRYPT 또는 DCSRENCRYPT

표 31. db2ARLibrary

속성	db2ARLibrary
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2ARLibrary
사용중인 디렉토리 공통 이름(cn)	ibm-db2ARLibrary
설명	응용프로그램 요청자 라이브러리의 이름
구문	대소문자 비구분 문자열

표 31. db2ARLibrary (계속)

속성	db2ARLibrary
최대 길이	256
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.427
GUID(전역 고유 식별자)	b3afd316-5c5b-11d3-b818-002035559151

표 32. db2databaseAlias

속성	db2databaseAlias
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseAlias
사용중인 디렉토리 공통 이름(cn)	ibm-db2DatabaseAlias
설명	데이터베이스 별명 이름(s)
구문	대소문자 비구분 문자열
최대 길이	1024
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.422
GUID(전역 고유 식별자)	b3afd318-5c5b-11d3-b818-002035559151

표 33. db2databaseName

속성	db2databaseName
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseName
사용중인 디렉토리 공통 이름(cn)	ibm-db2DatabaseName
설명	데이터베이스명
구문	대소문자 비구분 문자열
최대 길이	1024
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.421
GUID(전역 고유 식별자)	b3afd319-5c5b-11d3-b818-002035559151

표 34. db2databaseRelease

속성	db2databaseRelease
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseRelease
사용중인 디렉토리 공통 이름(cn)	ibm-db2DatabaseRelease
설명	데이터베이스 릴리스 번호

표 34. db2databaseRelease (계속)

속성	db2databaseRelease
구문	대소문자 비구분 문자열
최대 길이	64
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.429
GUID(전역 고유 식별자)	b3afd31a-5c5b-11d3-b818-002035559151

표 35. db2nodeAlias

속성	db2nodeAlias
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2NodeAlias
사용중인 디렉토리 공통 이름(cn)	ibm-db2NodeAlias
설명	데이터베이스 별명 이름(s)
구문	대소문자 비구분 문자열
최대 길이	1024
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.420
GUID(전역 고유 식별자)	b3afd31d-5c5b-11d3-b818-002035559151

표 36. db2nodeName

속성	db2nodeName
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2NodeName
사용중인 디렉토리 공통 이름(cn)	ibm-db2NodeName
설명	노드명
구문	대소문자 비구분 문자열
최대 길이	64
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.419
GUID(전역 고유 식별자)	b3afd31e-5c5b-11d3-b818-002035559151

표 37. db2nodePtr

속성	db2nodePtr
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2NodePtr
사용중인 디렉토리 공통 이름(cn)	ibm-db2NodePtr

표 37. db2nodePtr (계속)

속성	db2nodePtr
설명	데이터베이스를 소유하는 데이터베이스 서버를 나타내는 노드(DB2Node) 오브젝트에 대한 포인터
구문	구별된 이름
최대 길이	1000
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.423
GUID(전역 고유 식별자)	b3afd31f-5c5b-11d3-b818-002035559151
특별 주의사항	이 관계는 클라이언트가 데이터베이스에 연결하기 위해 프로토콜 통신 정보를 검색할 수 있도록 합니다.

표 38. db2gwPtr

속성	db2gwPtr
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2GwPtr
사용중인 디렉토리 공통 이름(cn)	ibm-db2GwPtr
설명	게이트웨이 서버를 나타내며 데이터베이스가 액세스될 수 있는 노드 오브젝트에 대한 포인터
구문	구별된 이름
최대 길이	1000
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.424
GUID(전역 고유 식별자)	b3afd31b-5c5b-11d3-b818-002035559151

표 39. db2instanceName

속성	db2instanceName
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2InstanceName
사용중인 디렉토리 공통 이름(cn)	ibm-db2InstanceName
설명	데이터베이스 서버 인스턴스의 이름
구문	대소문자 비구분 문자열
최대 길이	256
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.428
GUID(전역 고유 식별자)	b3afd31c-5c5b-11d3-b818-002035559151

표 40. db2Type

속성	db2Type
사용중인 디렉토리 LDAP 표시장치 이름	ibm-db2Type
사용중인 디렉토리 공통 이름(cn)	ibm-db2Type
설명	데이터베이스 서버 유형
구문	대소문자 비구분 문자열
최대 길이	64
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.418
GUID(전역 고유 식별자)	b3afd320-5c5b-11d3-b818-002035559151
주의사항	데이터베이스 서버의 유효한 유형: SERVER, MPP, DCS

표 41. DCEPrincipalName

속성	DCEPrincipalName
사용중인 디렉토리 LDAP 표시장치 이름	ibm-DCEPrincipalName
사용중인 디렉토리 공통 이름(cn)	ibm-DCEPrincipalName
설명	DCE 핵심부명
구문	대소문자 비구분 문자열
최대 길이	2048
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.443
GUID(전역 고유 식별자)	b3afd32d-5c5b-11d3-b818-002035559151

표 42. cesProperty

속성	cesProperty
사용중인 디렉토리 LDAP 표시장치 이름	ibm-cesProperty
사용중인 디렉토리 공통 이름(cn)	ibm-cesProperty
설명	이 속성의 값은 응용프로그램 고유의 선호 구성 매개변수를 제공하기 위해 사용될 수 있습니다. 예를 들어, 값은 XML 형식화 데이터를 포함할 수 있습니다. 이 모든 속성 값은 cesPropertyType 속성 값에서 동종이어야 합니다.
구문	대소문자 비구분 문자열
최대 길이	32700
복수 값	다중 값

표 42. cesProperty (계속)

속성	cesProperty
OID(오브젝트 식별자)	1.3.18.0.2.4.307
GUID(전역 고유 식별자)	b3afd2d5-5c5b-11d3-b818-002035559151

표 43. cesPropertyType

속성	cesPropertyType
사용중인 디렉토리 LDAP 표시장치 이름	ibm-cesPropertyType
사용중인 디렉토리 공통 이름(cn)	ibm-cesPropertyType
설명	이 속성 값은 구문, 의미 또는 cesProperty 속성의 모든 값의 다른 특성을 기술하기 위해 사용될 수 있습니다. 예를 들어, 『XML』의 값은 cesProperty 속성의 모든 값이 XML 구문으로서 코드화됨을 나타내기 위해 사용될 수 있습니다.
구문	대소문자 비구분 문자열
최대 길이	128
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.308
GUID(전역 고유 식별자)	b3afd2d6-5c5b-11d3-b818-002035559151

표 44. cisProperty

속성	cisProperty
사용중인 디렉토리 LDAP 표시장치 이름	ibm-cisProperty
사용중인 디렉토리 공통 이름(cn)	ibm-cisProperty
설명	이 속성의 값은 응용프로그램 특유의 선호 구성 매개변수를 제공하기 위해 사용될 수 있습니다. 예를 들어, 값은 INI 파일을 포함할 수 있습니다. 이 모든 속성 값은 cisPropertyType 속성 값에서 동종이어야 합니다.
구문	대소문자 비구분 문자열
최대 길이	32700
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.309
GUID(전역 고유 식별자)	b3afd2e0-5c5b-11d3-b818-002035559151

표 45. *cisPropertyType*

속성	cisPropertyType
사용중인 디렉토리 LDAP 표시장치 이름	ibm-cisPropertyType
사용중인 디렉토리 공통 이름(cn)	ibm-cisPropertyType
설명	이 속성 값은 구문, 의미 또는 cisProperty 속성의 모든 값의 다른 특성을 기술하기 위해 사용될 수 있습니다. 예를 들어, 『INI 파일』의 값은 cisProperty 속성의 모든 값이 INI 파일임을 나타내기 위해 사용될 수 있습니다.
구문	대소문자 비구분 문자열
최대 길이	128
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.310
GUID(전역 고유 식별자)	b3afd2e1-5c5b-11d3-b818-002035559151

표 46. *binProperty*

속성	binProperty
사용중인 디렉토리 LDAP 표시장치 이름	ibm-binProperty
사용중인 디렉토리 공통 이름(cn)	ibm-binProperty
설명	이 속성의 값은 응용프로그램 특유의 선호 구성 매개변수를 제공하기 위해 사용될 수 있습니다. 예를 들어, 값에는 2진 암호화 Lotus 123 등록 정보 세트가 들어 있을 수 있습니다. 이 속성의 모든 값은 관련된 binPropertyType 속성 값에서 동종이어야 합니다.
구문	2진
최대 길이	250000
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.305
GUID(전역 고유 식별자)	b3afd2ba-5c5b-11d3-b818-002035559151

표 47. *binPropertyType*

속성	binPropertyType
사용중인 디렉토리 LDAP 표시장치 이름	ibm-binPropertyType
사용중인 디렉토리 공통 이름(cn)	ibm-binPropertyType

표 47. binPropertyType (계속)

속성	binPropertyType
설명	이 속성 값은 구분, 의미 또는 binProperty 속성의 모든 값의 다른 특성을 기술하기 위해 사용될 수 있습니다. 예를 들어, 『Lotus 123』의 값은 binProperty 속성의 모든 값이 2진 코드화 Lotus 123 등록 정보임을 나타내기 위해 사용될 수 있습니다.
구분	대소문자 비구분 문자열
최대 길이	128
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.306
GUID(전역 고유 식별자)	b3afd2bb-5c5b-11d3-b818-002035559151

표 48. PropertyType

속성	PropertyType
사용중인 디렉토리 LDAP 표시장치 이름	ibm-propertyType
사용중인 디렉토리 공통 이름(cn)	ibm-propertyType
설명	이 속성의 값은 eProperty 오브젝트의 의미 특성을 설명합니다.
구분	대소문자 비구분 문자열
최대 길이	128
복수 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.320
GUID(전역 고유 식별자)	b3afd4ed-5c5b-11d3-b818-002035559151

표 49. settingID

속성	settingID
사용중인 디렉토리 LDAP 표시장치 이름	적용할 수 없음
사용중인 디렉토리 공통 이름(cn)	적용할 수 없음
설명	eProperty와 같은 오브젝트 항목에서 파생한 cimSetting를 식별하기 위해 사용될 수 있는 이름 지정 속성
구분	대소문자 비구분 문자열
최대 길이	256
복수 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.325

표 49. *settingID* (계속)

속성	settingID
GUID(전역 고유 식별자)	b3afd596-5c5b-11d3-b818-002035559151

부록K. 제어 센터 확장

버전 7에서는 추가 기능을 제공하기 위해 새로운 플러그인 아키텍처를 사용하여 DB2 Universal Database 제어 센터를 확장할 수 있습니다.

플러그인 아키텍처의 개념은 제어 센터 팝업 메뉴에서 지정된 오브젝트에 대해 항목을 추가하는 능력을 제공하고 툴 바에 새 버튼을 추가하는 것입니다. 사용자가 구현해야 하는 Java 인터페이스 세트는 이 도구와 함께 제공됩니다. 이 인터페이스는 제어 센터에 포함될 추가 조치를 전달하기 위해 사용됩니다.

성능 고려사항

플러그인 확장 파일(db2plug.zip)은 제어 센터 도구가 시동될 때 로드됩니다. 이 ZIP 파일의 크기에 따라 도구의 시동 시간이 길어질 수 있지만, 플러그인 ZIP 파일이 대부분의 사용자에게 작을 것이고 충격이 최소한일 것이라고 예상합니다.

팩 고려사항

Java 아카이브 파일의 규칙에 따라 확장 클래스 파일을 ZIP해야 합니다. 제어 센터 도구를 응용프로그램으로 수행하려면, ZIP 파일(db2plug.zip)이 클래스 경로에 있어야 합니다. 제어 센터 도구를 애플릿으로 수행시키려면, ZIP 파일이 제어 센터 HTML 파일에서 <codebase> 태그가 지정하는 곳에 있어야 합니다.

ZIP 파일은 압축 없이 구축되고 모든 클래스 파일의 상대 경로 위치를 유지보수해야 합니다(zip -r0 db2plug.zip *.class).

인터페이스 설명

다음과 같은 인터페이스가 제공됩니다.

- CCExtension
- CCOject
- CCMenuAction

- CCToolbarAction

이들 인터페이스는 다음 절에서 예와 함께 설명됩니다.

CCExtension

CCExtension 인터페이스는 사용자가 새 툴바 버튼, 새 메뉴 항목을 추가하고 이미 있는 메뉴 조치를 대체하여 제어 센터 사용자 인터페이스를 확장할 수 있게 합니다.

외부 인터페이스는 다음과 같이 정의됩니다.

```
public interface CCExtension
{
    /**
     * Get an array of CCOBJECT subclass objects which define
     * a list of objects to be inserted or overridden in the
     * Control Center
     * @return CCOBJECT[] CCOBJECT subclass objects array
     */
    public CCOBJECT[] getObjects();

    /**
     * Get an array of CCToolbarAction subclass objects which represent
     * a list of buttons to be added to the Control Center
     * main toolbar.
     * @return CCToolbarAction[] CCToolbarAction subclass objects array
     */
    public CCToolbarAction[] getToolbarActions();
}
```

CCExtension을 사용하려면, "com.ibm.db2.tools.cc.navigator" 패키지를 가져오기 하여 이 인터페이스를 구현하는 Java 클래스를 작성하십시오. 새로운 클래스는 getObjects() 및 getToolbarActions() 메소드의 구현을 제공해야 합니다.

getObjects() 메소드는 사용자가 새로운 메뉴 조치를 추가하거나 사전 정의된 메뉴 조치 세트를 제거하려고 하는 기존 오브젝트를 정의하는 CCOBJECT의 배열을 리턴합니다.

getToolbarActions() 메소드는 제어 센터 기본 도구 모음에 추가될 CCToolbarAction의 배열을 리턴합니다.

단일 CCExtension 서브클래스 파일 또는 다중 CCExtension 서브클래스 파일이 제어 센터 확장을 정의하는 데 사용될 수 있습니다. 제어 센터가 이 확장을 사용하게 하려면, 다음과 같은 설정 프로시디어를 사용하십시오.

1. 모든 CCExtension 서브클래스 파일이 있는 "db2plug.zip" 파일을 작성하십시오. 이 파일은 압축되어서는 안 됩니다. 예를 들어, CCExtension 파일이 플러그인 패키지에 있고 이들이 플러그인 디렉토리에 위치하고 있으면, 다음 명령을 사용하십시오.

```
zip -r0 db2plug.zip plugin\*.class
```

이 명령은 모든 플러그인 패키지 클래스 파일을 db2plug.zip 파일에 넣고 상대 경로 정보를 보존합니다.

2. 제어 센터를 애플릿으로 수행시키려면, db2plug.zip 파일을 제어 센터 HTML 파일에서 <codebase> 태그가 지정하는 곳에 넣으십시오. 제어 센터를 응용프로그램으로 수행시키려면, db2plug.zip 파일을 CLASSPATH 환경 변수가 지정하는 디렉토리에 넣으십시오.

다중 아카이브 파일을 지원하는 브라우저의 경우, 제어 센터 HTML 페이지의 아카이브 목록에 "db2plug.zip"을 추가만 하십시오. 그렇지 않으면, 모든 CCExtension, CCOBJECT, CCToolbarAction 및 CCMenuAction 서브클래스 파일이 그들이 속하는 패키지에 따라 해당하는 상대 디렉토리에 있어야 합니다.

CCObject

CCObject 인터페이스는 이미 있는 오브젝트의 메뉴 조치의 작동을 변경할 수 있게 합니다.

외부 인터페이스는 다음과 같이 정의됩니다.

```
public interface CCOBJECT
{
    /**
     * The following static constants defines a list of object type
     * available to be added to the Control Center tree.
     */
    public static final int UDB_SYSTEMS_FOLDER           = 0;
    public static final int UDB_SYSTEM                   = 1;
    public static final int UDB_INSTANCES_FOLDER         = 2;
    public static final int UDB_INSTANCE                 = 3;
    public static final int UDB_DATABASES_FOLDER         = 4;
    public static final int UDB_DATABASE                 = 5;
    public static final int UDB_TABLES_FOLDER            = 6;
```

```

public static final int UDB_TABLE = 7;
public static final int UDB_TABLESPACES_FOLDER = 8;
public static final int UDB_TABLESPACE = 9;
public static final int UDB_VIEWS_FOLDER = 10;
public static final int UDB_VIEW = 11;
public static final int UDB_ALIASES_FOLDER = 12;
public static final int UDB_ALIAS = 13;
public static final int UDB_TRIGGERS_FOLDER = 14;
public static final int UDB_TRIGGER = 15;
public static final int UDB_SCHEMAS_FOLDER = 16;
public static final int UDB_SCHEMA = 17;
public static final int UDB_INDEXES_FOLDER = 18;
public static final int UDB_INDEX = 19;
public static final int UDB_CONNECTIONS_FOLDER = 20;
public static final int UDB_CONNECTION = 21;
public static final int UDB_REPLICATION_SOURCES_FOLDER = 22;
public static final int UDB_REPLICATION_SOURCE = 23;
public static final int UDB_REPLICATION_SUBSCRIPTIONS_FOLDER = 24;
public static final int UDB_REPLICATION_SUBSCRIPTION = 25;
public static final int UDB_BUFFERPOOLS_FOLDER = 26;
public static final int UDB_BUFFERPOOL = 27;
public static final int UDB_APPLICATION_OBJECTS_FOLDER = 28;
public static final int UDB_USER_DEFINED_DISTINCT_DATATYPES_FOLDER = 29;
public static final int UDB_USER_DEFINED_DISTINCT_DATATYPE = 30;
public static final int UDB_USER_DEFINED_DISTINCT_FUNCTIONS_FOLDER = 31;
public static final int UDB_USER_DEFINED_DISTINCT_FUNCTION = 32;
public static final int UDB_PACKAGES_FOLDER = 33;
public static final int UDB_PACKAGE = 34;
public static final int UDB_STORE_PROCEDURES_FOLDER = 35;
public static final int UDB_STORE_PROCEDURE = 36;
public static final int UDB_USER_AND_GROUP_OBJECTS_FOLDER = 37;
public static final int UDB_DB_USERS_FOLDER = 38;
public static final int UDB_DB_USER = 39;
public static final int UDB_DB_GROUPS_FOLDER = 40;
public static final int UDB_DB_GROUP = 41;
public static final int UDB_DRDA_TABLE = 42;

public static final int S390_SUBSYSTEMS_FOLDER = 43;
public static final int S390_SUBSYSTEM = 44;
public static final int S390_BUFFERPOOLS_FOLDER = 45;
public static final int S390_BUFFERPOOL = 46;
public static final int S390_VIEWS_FOLDER = 47;
public static final int S390_VIEW = 48;
public static final int S390_DATABASES_FOLDER = 49;
public static final int S390_DATABASE = 50;
public static final int S390_TABLESPACES_FOLDER = 51;
public static final int S390_TABLESPACE = 52;
public static final int S390_TABLES_FOLDER = 53;
public static final int S390_TABLE = 54;
public static final int S390_INDEXES_FOLDER = 55;
public static final int S390_INDEX = 56;
public static final int S390_STORAGE_GROUPS_FOLDER = 57;
public static final int S390_STORAGE_GROUP = 58;
public static final int S390_ALIASES_FOLDER = 59;
public static final int S390_ALIAS = 60;
public static final int S390_SYNONYMS_FOLDER = 61;
public static final int S390_SYNONYM = 62;
public static final int S390_APPLICATION_OBJECTS_FOLDER = 63;
public static final int S390_COLLECTIONS_FOLDER = 64;

```

```

public static final int S390_COLLECTION = 65;
public static final int S390_PACKAGES_FOLDER = 66;
public static final int S390_PACKAGE = 67;
public static final int S390_PLANS_FOLDER = 68;
public static final int S390_PLAN = 69;
public static final int S390_PROCEDURES_FOLDER = 70;
public static final int S390_PROCEDURE = 71;
public static final int S390_DB_USERS_FOLDER = 72;
public static final int S390_DB_USER = 73;
public static final int S390_LOCATIONS_FOLDER = 74;
public static final int S390_LOCATION = 75;
public static final int S390_DISTINCT_TYPES_FOLDER = 76;
public static final int S390_DISTINCT_TYPE = 77;
public static final int S390_USER_DEFINED_FUNCTIONS_FOLDER = 78;
public static final int S390_USER_DEFINED_FUNCTION = 79;
public static final int S390_TRIGGERS_FOLDER = 80;
public static final int S390_TRIGGER = 81;
public static final int S390_SCHEMAS_FOLDER = 82;
public static final int S390_SCHEMA = 83;
public static final int S390_CATALOG_TABLES_FOLDER = 84;
public static final int S390_CATALOG_TABLE = 85;

public static final int DCS_GATEWAY_CONNECTIONS_FOLDER = 86;
public static final int DCS_GATEWAY_CONNECTION = 87;

/**
 * Total number of object types
 */
public static final int NUM_OBJECT_TYPES = 88;

/**
 * Get the name of these object
 * The function returns the name of this object. This name
 * can be of three types:
 * (1) Fully qualified name
 * Syntax: xxxxx-yyyyy-zzzzz
 * where xxxxx-yyyyy is the fully quality name of the
 * parent object and zzzzz is the name of the new object.
 * Note: Parent and child object name is separated by '-' character.
 * If a schema name is required to identify object, the fully
 * qualified name is represented by xxxxx-yyyyy-wwwww.zzzzz
 * where wwwwww is the schema name.
 * Only the behavior of the object that matches this fully
 * qualified name will be affected.
 * (2) Parent fully qualified name
 * Syntax: xxxxx-yyyyy
 * where xxxxx-yyyyy is the fully qualified name of the
 * parent object.
 * When the object type is folder (ie. DATABASES_FOLDER), the
 * getName() should only return the fully qualified name of the
 * folder's parent.
 * Only the behavior of the object that match this name
 * and the specific type return by the getType() function will be
 * affected.
 * (3) null
 * Syntax: null
 * If null is return, the CCActions returns by the getActions()
 * call will be applied to all objects of type returns by the
 * getType() call.

```

```

    * @return String object name
    */
    public String getName();

    /**
     * Get the type of this object
     * @return int return one of the static type constants defined
     * in this interface
     */
    public int getType();

    /**
     * Get the CCMenu Action array which defines the list of menu actions
     * to be created for the selected object
     * @return CCMenuAction[] CCMenuAction array
     */
    public CCMenuAction[] getMenuActions();

    /**
     * Check if this object is editable.
     * If not, the Alter related menu items will be removed from
     * the object's popup menu return boolean If false, the Alter
     * menu item will be removed from the object's popup menu
     */
    public boolean isEditable();

    /**
     * Check if this object is configurable.
     * If not, the configuration related menu items will be
     * removed from the object's popup menu return boolean If
     * false, the Configuration related menu item will be removed
     * from the object's popup menu
     */
    public boolean isConfigurable();
}

```

주: 이 경우, CCOBJECT에 있는 마지막 두 메소드(isEditable() 및 isConfigurable())는 항상 **true**를 리턴해야 합니다.

CCMenuItem

CCMenuItem 인터페이스는 제어 센터 오브젝트에 의해 사용되는 새로운 조치를 정의할 수 있게 합니다.

외부 인터페이스는 다음과 같이 정의됩니다.

```

public interface CCMenuAction
{
    /**
     * Get the name of this action
     * @return String Name text on the menu item
     */
    public String getMenuText();
}

```

```

    /**
     * Invoked when an action occurs. Use the getActionCommand()
     * method of the ActionEvent to get the fully qualified name of
     * the invoked Control Center object.
     * @param e Action event
     */
    public void actionPerformed(ActionEvent e);
}

```

CCToolBarAction

CCToolbarAction 인터페이스는 제어 센터 도구 모음에 새로운 조치를 정의할 수 있게 합니다.

외부 인터페이스는 다음과 같이 정의됩니다.

```

public interface CCToolbarAction
{
    /**
     * Get the name of this action
     * @return String Name text on the menu item, or toolbar
     * button hover help
     */
    public String getHoverHelpText();

    /**
     * Get the icon for the toolbar button
     * Any toolbar CCAction should implement this function and return
     * a valid ImageIcon object. Otherwise, the button will have no icon.
     * @return ImageIcon Icon to be displayed
     */
    public ImageIcon getIcon();

    /**
     * Invoked when an action occurs.
     * @param e Action event
     */
    public void actionPerformed(ActionEvent e);
}

```

사용 시나리오

다음 예에 있는 코드는 다음과 같습니다.

1. SAMPLE 데이터베이스의 조치 갱신(603 페이지의 『MySample.java』 참조)
2. 모든 데이터베이스 오브젝트의 조치 갱신(604 페이지의 『MyDatabaseActions.java』 참조)

3. 새로운 인스턴스 오브젝트 추가(604 페이지의 『MyInstance.java』 참조)
4. DB2 인스턴스의 조치 갱신(605 페이지의 『MyDB2.java』 참조)
5. 데이터베이스 폴더의 조치 갱신(606 페이지의 『MyDatabases.java』 참조)
6. SYSIBM.SYSPLAN 테이블의 조치 갱신(607 페이지의 『MySYSPLAN.java』 참조)
7. 새로운 테이블 오브젝트 추가(607 페이지의 『MyTable.java』 참조)
8. 응용프로그램 오브젝트 아래에 있는 DB_User 오브젝트의 조치 갱신(608 페이지의 『MyDBUser.java』 참조)
9. 제어 센터 도구 모음에 버튼 추가(609 페이지의 『MyToolbarAction.java』 참조)

기본 확장 파일은 MyExtension.java입니다. 모든 클래스 파일은 플러그인 디렉토리에 저장되고 다음 명령으로 ZIP됩니다.

```
zip -r0 db2plug.zip plugin
```

출력된 db2plug.zip 파일은 제어 센터가 응용프로그램 또는 애플릿으로 수행되는 지에 따라 CLASSPATH 또는 코드기준 디렉토리에 배치됩니다.

MyExtension.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyExtension implements CCExtension
{
    public CCObject[] getObjects()
    {
        CCObject[] objs = new CCObject[10];
        objs[0] = new MySample();
        objs[1] = new MyDatabaseActions();
        objs[2] = new MyInstance();
        objs[3] = new MyDB2();
        objs[4] = new MyDatabases();
        objs[5] = new MySYSPLAN();
        objs[6] = new MyTable();
        objs[7] = new MyDBUser();
        return objs;
    }

    public CCAction[] getActions()
```

```

    {
        CCAction[] actions = new CCAction[1];
        actions[0] = new MyToolbarAction();
        return actions;
    }
}

```

MySample.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MySample implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE";
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

MyDatabaseActions.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabaseActions implements CCOBJECT
{
    public String getName()
    {
        return null;
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] acts = new CCACTION[2];
        acts[0] = new MyDropAction();
        acts[1] = new MyAction();
        return acts;
    }
}
```

MyInstance.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyInstance implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - MyInstance";
    }
}
```



```

public int getType()
{
    return INSTANCE;
}

public javax.swing.ImageIcon getIcon()
{
    return null;
}

public boolean isNew()
{
    return true;
}

public CCAction[] getActions()
{
    CCAction[] acts = new CCAction[2];
    acts[0] = new MyAlterAction();
    acts[1] = new MyAction();
    return null;
}
}

```

MyDB2.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDB2 implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2";
    }

    public int getType()
    {
        return INSTANCE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }
}

```

```

public boolean isNew()
{
    return false;
}

public CCAction[] getActions()
{
    CCAction[] acts = new CCAction[3];
    acts[0] = new MyAlterAction();
    acts[1] = new MyAction();
    acts[2] = new MyCascadeAction();
    return acts;
}
}

```

MyDatabases.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabases implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - Databases";
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[1];
        acts[0] = new MyCreateAction();
    }
}

```

```
        return acts;
    }
}
```

MySYSPLAN.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MySYSPLAN implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - SYSPLAN";
    }

    public int getType()
    {
        return TABLE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] acts = new CCACTION[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}
```

MyTable.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;
```

```

public class MyTable implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - MyTable";
    }

    public int getType()
    {
        return TABLE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return true;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] acts = new CCACTION[2];
        acts[0] = new MyALTERACTION();
        acts[1] = new MyACTION();
        return acts;
    }
}

```

MyDBUser.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDBUser implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - TEST-DB Users";
    }

    public int getType()
    {
        return DB_USER;
    }
}

```

```

    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

MyToolbarAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;
import javax.swing.*;

public class MyToolbarAction extends CCAction
{
    public MyToolbarAction()
    {
        super("MyToolbarAction");
    }

    public ImageIcon getIcon()
    {
        return <Your icon>;
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
            objectName );
        return true;
    }
}

```

MyAlterAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAlterAction extends CCAction
{
    public MyAlterAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My alter action performed, object name = " +
            objectName );
        return true;
    }
}
```

MyAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAction extends CCAction
{
    public MyAction()
    {
        super("MyAction");
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
            objectName );
        return true;
    }
}
```

MyDropAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDropAction extends CCAction
{
```

```

public MyDropAction()
{
    super(1);
}

public boolean actionPerformed(String objectName)
{
    System.out.println( "My drop action performed, object name = " +
                        objectName );
    return true;
}
}

```

MyCascadeAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCascadeAction extends CCAction
{
    public MyCascadeAction()
    {
        super(11,2);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My cascade action performed, object name = " +
                            objectName );
        return true;
    }
}

```

MyCreateAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCreateAction extends CCAction
{
    public MyCreateAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My create action performed, object name = " +
                            objectName );
        return true;
    }
}

```

부록L. DB2 라이브러리 사용

DB2 Universal Database 라이브러리는 온라인 도움말, 책(PDF 및 HTML) 및 샘플 프로그램이 HTML 형식으로 구성됩니다. 이 절에서는 제공되는 정보 및 액세스하는 방법을 설명합니다.

제품 정보에 온라인으로 액세스하려면, 정보 센터를 이용할 수 있습니다. 628 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오. 웹에서 타스크 정보, DB2 책, 문제점 해결 정보, 샘플 프로그램 및 DB2 정보를 열람할 수 있습니다.

DB2 PDF 파일 및 인쇄된 책

DB2 정보

다음의 테이블은 DB2 책을 4개의 범주로 나눕니다.

DB2 안내 및 참조 정보

이 책에는 모든 플랫폼에 공통적인 DB2 정보가 들어 있습니다.

DB2 설치 및 구성 정보

이 책에는 특정 플랫폼의 DB2를 위한 것입니다. 예를 들어, OS/2, Windows 및 UNIX 플랫폼에서의 DB2용으로 각각 다른 빠른 시작 책이 있습니다.

플랫폼간 샘플 프로그램(HTML)

이 샘플들은 응용프로그램 개발 클라이언트와 함께 설치된 샘플 프로그램의 HTML 버전입니다. 이들은 단지 정보용으로서 실제 프로그램을 대체하지는 않습니다.

릴리스 정보

이러한 파일에는 DB2 책에 포함될 수 없었던 최신 정보가 포함되어 있습니다.

설치 매뉴얼, 릴리스 정보 및 지습서는 제품 CD-ROM의 HTML 디렉토리에서 볼 수 있습니다. 대부분의 책은 단지 보기용으로 제품 CD-ROM에서 HTML 형식으로 제공되고 보기와 인쇄용으로 제품 CD-ROM에서 PDF 형식으로 제공됩니다. 또한 IBM에서 인쇄된 책을 주문하려면 623 페이지의 『인쇄된 책 주문』에서 자세한 내용을 참조하십시오. 다음 테이블에는 주문할 수 있는 책을 보여줍니다.

OS/2 및 Windows 플랫폼에서는 `sqllib\doc\html` 디렉토리에 HTML 파일을 설치할 수 있습니다. DB2 정보는 여러 나라 언어로 번역되었습니다. 하지만, 모든 정보가 모든 나라의 언어로 번역된 것은 아닙니다. 정보가 특정 나라의 언어로 사용할 수 없을 경우에는 영문으로 제공됩니다.

UNIX 플랫폼에서는 `doc/%L/html` 디렉토리에 여러 나라 언어 버전의 HTML 파일을 설치할 수 있습니다. 여기서 `%L`은 해당 언어의 로케일을 나타냅니다. 빠른 시작 책에서 보다 자세한 내용을 참조하십시오.

다음의 여러 가지 방법으로 DB2 책을 구하고 정보를 액세스할 수 있습니다.

- 627 페이지의 『정보 온라인 보기』
- 631 페이지의 『정보 온라인 검색』
- 623 페이지의 『인쇄된 책 주문』
- 623 페이지의 『PDF 책 인쇄』

표 50. DB2 정보

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
DB2 안내 및 참조 정보			
관리 안내서	<p>관리 안내서: 계획에서는 데이터베이스의 개념에 대한 개요, 논리적 또는 물리적인 데이터베이스 설계와 같은 설계에 대한 정보 그리고 고가용성에 대한 정보를 제공합니다.</p> <p>관리 안내서: 구현에서는 사용자의 설계, 데이터베이스 액세스, 감사, 백업 및 복구와 같은 구현에 대한 정보를 제공합니다.</p> <p>관리 안내서: 성능에서는 데이터베이스의 환경, 응용프로그램 성능 평가 및 성능 조정에 대한 정보를 제공합니다.</p> <p>사용자는 문서 번호 SBOF-8934를 사용하여 세 권으로 된 <i>관리 안내서</i> 책을 주문할 수 있습니다.</p>	<p>SA30-0990 db2d1x70</p> <p>SA30-0988 db2d2x70</p> <p>SA30-0989 db2d3x70</p>	<p>db2d0</p>
<i>Administrative API Reference</i>	데이터베이스를 관리하는 데 사용할 수 있는 DB2 API와 데이터 구조에 대해 설명합니다. 또한 응용프로그램에서 API를 호출하는 방법을 설명합니다.	<p>SC09-2947 db2b0x70</p>	db2b0
응용프로그램 빌드 안내서	환경 설정 정보와 Windows에서 DB2 응용프로그램을 컴파일, 링크 및 수행하기 위한 지시 사항이 단계별로 제공되어 있습니다.	<p>SA30-0991 db2axx70</p>	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	DB2 Universal Database 제품을 사용할 때 발생할 수 있는 APPC, CPI-C 및 SNA 센스 코드에 관한 일반 정보를 제공합니다.	<p>문서 번호가 없습니다. db2apx70</p>	db2ap
응용프로그램 개발 안내서	Embedded SQL 또는 Java(JDBC 및 SQLJ)를 사용하여 DB2 데이터베이스를 액세스하는 응용프로그램을 개발하는 방법을 설명합니다. 저장 프로시저어 작성, 사용자 정의 함수 작성, 사용자 정의 유형 작성, 트리거 사용, 파티션된 환경 또는 연합 시스템에서 응용프로그램을 개발하는 등의 다양한 주제가 다루어집니다.	<p>SA30-0992 db2a0x70</p>	db2a0

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>CLI Guide and Reference</i>	DB2 콜 레벨 인터페이스와 Microsoft ODBC 스펙과 호환 가능한 호출 가능 SQL 인터페이스를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램의 개발 방법에 대해 설명합니다.	SC09-2950 db210x70	db210
<i>Command Reference</i>	명령행 프로세서를 사용하는 방법을 설명하고 데이터베이스를 관리하기 위해 사용할 수 있는 DB2 명령을 설명합니다.	SC09-2951 db2n0x70	db2n0
연결성 보충 설명서	AS/400용 DB2, OS/390용 DB2, MVS용 DB2 또는 VM용 DB2를 DB2 Universal Database 서버와의 DRDA 응용프로그램 리퀘스터로 사용하는 방법에 대한 참조 정보 및 설치 정보를 제공합니다. 또한 DB2 Connect AR(응용프로그램 리퀘스터)과 함께 DRDA AS(응용프로그램 서버)를 사용하는 방법에 대해서도 상세히 설명합니다. HTML 및 PDF 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2h1x70	db2h1
데이터 이동 유틸리티 안내 및 참조서	Import, Export, Load, AutoLoader 및 DPROP와 같이 데이터 이동을 용이하게 해 주는 DB2 UDB 유틸리티의 사용 방법에 대해 설명합니다.	SA30-0994 db2dmx70	db2dm
<i>Data Warehouse Center</i> 관리 안내서	Data Warehouse Center를 사용하여 데이터 웨어하우스를 구축 및 유지보수하는 방법을 제공합니다.	SA30-1000 db2ddx70	db2dd
<i>Data Warehouse Center</i> 응용프로그램 통합 안내서	프로그래머들이 Data Warehouse Center 및 Information Catalog Manager를 응용프로그램과 통합하는 데 도움을 주는 정보를 제공합니다.	SA30-1001 db2adx70	db2ad
<i>DB2 Connect</i> 사용자 안내서	DB2 Connect 제품에 대한 개념, 프로그래밍 및 일반 사용 정보를 제공합니다.	SA30-0993 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	DB2 Query Patroller 시스템의 조작 개요, 특정 조작 및 관리 정보, 관리 그래픽 사용자 인터페이스 유틸리티에 대한 타스크 정보를 제공합니다.	SC09-2958 db2dwx70	db2dw

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Query Patroller User's Guide</i>	DB2 Query Patroller의 도구 및 함수를 사용하는 방법을 설명합니다.	SC09-2960	db2ww
		db2wwx70	
용어집	DB2에서 사용되는 용어와 그 구성요소에 대한 정의를 제공합니다.	문서 번호가 없습니다.	db2t0
	HTML 형식과 SQL 참조서에서 사용할 수 있습니다.	db2t0x70	
<i>Image, Audio 및 Video Extenders 관리 및 프로그래밍</i>	DB2 Extender에 대한 일반적인 정보와 이미지, 오디오 및 비디오(IAV)의 관리 및 구성에 대한 정보 그리고 IAV extenders를 사용한 프로그램에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1043	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	정보 카탈로그 관리에 대한 지시사항을 제공합니다.	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	Information Catalog Manager에 대한 아키텍처 인터페이스에 대한 정의를 제공합니다.	SC26-9997	db2bi
		db2bix70	
<i>Information Catalog Manager 사용자 안내서</i>	Information Catalog Manager 사용자 인터페이스 사용에 대한 정보를 제공합니다.	SA30-1002	db2ai
		db2aix70	
설치 및 구성 보충 설명서	플랫폼 특정 DB2 클라이언트의 플랜, 설치 및 설정에 대해 설명합니다. 또한 바인딩, 클라이언트 및 서버 통신의 설정, DB2 GUI 도구, DRDA AS, 분산 설치 및 이중 데이터 소스에 대한 분산 요구와 액세스 방식의 구성에 대한 정보가 들어 있습니다.	GA30-0975	db2iy
		db2iyx70	
메시지 참조서	DB2, Information Catalog Manager 및 Data Warehouse Center에서 발행하는 메시지와 코드를 나열하고 수행해야 할 조치에 대해 설명합니다.	볼륨 1 GA30-0986	db2m0
		db2m1x70	
	문서 번호(SBOF-8932)를 사용하여 두 권으로 된 메시지 참조서 책을 모두 주문할 수 있습니다.	볼륨 2 GA30-0987	
		db2m2x70	

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>OLAP Integration Server Administration Guide</i>	OLAP 통합 서버의 관리 프로그램 구성요소를 사용하는 방법을 설명합니다.	SC27-0787	n/a
		db2dpx70	
<i>OLAP Integration Server Metaoutline User's Guide</i>	표준 OLAP Metaoutline 인터페이스 (Metaoutline Assistant가 아닌)를 사용하여 OLAP Metaoutlines을 작성하고 사용하는 방법을 설명합니다.	SC27-0784	n/a
		db2upx70	
<i>OLAP Integration Server Model User's Guide</i>	표준 OLAP 모델 인터페이스(Model Assistant가 아닌)를 사용하여 OLAP 모델을 작성하는 방법을 설명합니다.	SC27-0783	n/a
		db2lpx70	
<i>OLAP 설치 및 사용자 안내서</i>	OLAP Starter Kit에 대한 구성 및 설치 정보를 제공합니다.	SA30-1074	db2ip
		db2ipx70	
<i>Excel용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Excel 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0564	db2ep
		db2epx70	
<i>Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Lotus 1-2-3 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0565	db2tp
		db2tpx70	
<i>복제 안내 및 참조서</i>	DB2와 함께 제공된 IBM 복제 도구에 관한 플랜, 구성, 관리 및 사용 정보를 제공합니다.	SA30-1003	db2e0
		db2e0x70	
<i>Spatial Extender 사용자 안내 및 참조서</i>	Spatial Extender 설치, 구성, 관리, 프로그래밍 및 문제 해결에 대한 정보를 제공합니다. 또한 공간 데이터 개념에 대한 설명을 제공하고 Spatial Extender에만 고유하게 적용되는 참조 정보(메시지 및 SQL)를 제공합니다.	SA30-1045	db2sb
		db2sbx70	
<i>SQL 시작하기</i>	SQL 개념을 소개하고, 많은 구조와 타스크에 관한 예를 보여줍니다.	SA30-0996	db2y0
		db2y0x70	
<i>SQL 참조서, 볼륨 1 및 볼륨 2</i>	SQL 구문, 의미 그리고 언어 규칙에 대해 설명합니다. 또한 릴리스 간 비호환성, 제품 제한 사항 및 카탈로그 뷰에 대한 정보도 들어 있습니다.	볼륨 1 SA30-0997	db2s0
		db2s1x70	
	SBOF-8933 문서 번호를 사용하여 SQL 참조서를 주문할 수 있습니다.	볼륨 2 SA30-0998	
		db2s2x70	

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
시스템 모니터 안내 및 참조 서	데이터베이스와 데이터베이스 관리 프로그램에 관한 여러 종류의 정보를 수집하는 방법에 대해 설명합니다. 이 책은 데이터베이스 활동을 이해하고, 성능을 향상시키고, 문제점의 원인을 판별하기 위한 정보를 사용하는 방법을 설명합니다.	SA30-0995 db2f0x70	db2f0
Text Extender 관리 및 프로그래밍	DB2 extenders에 관한 일반적인 정보와 Text extenders 관리 및 구성에 관한 정보, Text extenders를 사용한 프로그래밍에 관한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1044 desu9x70	desu9
문제점 해결 안내서	오류의 출처를 판별하고 문제점으로부터 회복하고, DB2 고객 서비스와 상담하여 진단 도구를 사용하는 것을 도와줍니다.	GA30-0704 db2p0x70	db2p0
새로운 기능	DB2 Universal Database, 버전 7의 새로운 특성, 기능 및 향상된 내용을 설명합니다.	SA30-0999 db2q0x70	db2q0
DB2 설치 및 구성 정보			
OS/2 및 Windows용 DB2 Connect Enterprise Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Enterprise Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0974 db2c6x70	db2c6
UNIX용 DB2 Connect Enterprise Edition 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Connect Enterprise Edition에 대한 플랜, 이주, 설치, 구성 및 타스크 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0973 db2cyx70	db2cy
DB2 Connect Personal Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 모든 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0981 db2c1x70	db2c1
DB2 Connect Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2962 db2c4x70	db2c4

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Data Links Manager</i> 빠른 시작	AIX 및 Windows 32 비트 운영 체제용 DB2 Data Links Manager에 대한 플랜, 설치, 구성 및 타스크 정보를 제공합니다.	GA30-0980 db2z6x70	db2z6
<i>UNIX용 DB2 Enterprise - Extended Edition</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Enterprise - Extended Edition 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0978 db2v3x70	db2v3
<i>Windows용 DB2 Enterprise - Extended Edition</i> 빠른 시작	Windows 32 비트 운영 체제용 DB2 Enterprise - Extended Edition에 관한 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0977 db2v6x70	db2v6
<i>OS/2용 DB2</i> 빠른 시작	OS/2 운영 체제에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0982 db2i2x70	db2i2
<i>UNIX용 DB2</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0984 db2ixx70	db2ix
<i>Windows용 DB2</i> 빠른 시작	Windows 32 비트 운영 체제에서 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0985 db2i6x70	db2i6
<i>DB2 Personal Edition</i> 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서의 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GA30-0983 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	지원되는 모든 Linux에서 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller</i> 설치 안내서	DB2 Query Patroller에 관한 설치 정보를 제공합니다.	GA30-0976 db2iw70	db2iw

표 50. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Warehouse Manager</i> 설치 안내서	웨어하우스 에이전트, 웨어하우스 변환기 및 Information Catalog Manager에 관한 설치 정보를 제공합니다.	GA30-1027 db2idx70	db2id
플랫폼간 샘플 프로그램(HTML)			
샘플 프로그램(HTML)	DB2가 지원하는 모든 플랫폼에서 프로그래밍 언어에 대한 샘플 프로그램이 HTML 형식으로 제공됩니다. 이 샘플 프로그램은 정보용으로만 제공됩니다. 모든 샘플을 모든 프로그래밍 언어로 사용할 수 있는 것은 아닙니다. HTML 샘플은 DB2 응용프로그램 개발 클라이언트가 설치될 때에 사용할 수 있습니다.	문서 번호가 없습니다.	db2hs
릴리스 정보			
<i>DB2 Connect</i> 릴리스 정보	DB2 Connect 책에는 포함될 수 없었던 최신 정보를 제공합니다.	#2를 참조하십시오.	db2cr
<i>DB2</i> 설치 정보	DB2 책에는 포함될 수 없었던 최신 설치 정보를 제공합니다.	제품 CD-ROM에서만 사용할 수 있습니다.	
<i>DB2</i> 릴리스 정보	DB2 책에는 포함될 수 없었던 모든 DB2 제품 및 기능에 대한 최신 정보를 제공합니다.	#2를 참조하십시오.	db2ir

주:

1. 파일 이름의 6번째 자리에 있는 문자 *x*는 책의 언어 버전을 나타냅니다. 예를 들면, 파일 이름 db2d0e70은 관리 안내서 책의 영문 버전을 나타내며 db2d0k70은 같은 책의 한글 버전을 나타냅니다. 다음 문자는 언어 버전을 나타내기 위해 파일 이름의 6번째 자리에 사용됩니다.

언어	식별자
브라질 포르투갈어	b
불가리아어	u
체코어	x
덴마크어	d
네덜란드어	q
영어	e

핀란드어	y
프랑스어	f
독일어	g
그리스어	a
헝가리어	h
이탈리아어	i
일본어	j
한글	k
노르웨이어	n
폴란드어	p
포르투갈어	v
러시아어	r
중국어	c
슬로베니아어	l
스페인어	z
스웨덴어	s
대만어	t
터키어	m

2. DB2 책에 포함되어 있지 않을 수 있는 최신 정보는 릴리스 정보에서 HTML 형식과 ASCII 파일로 사용할 수 있습니다. HTML 버전은 정보 센터와 제품 CD-ROM에서 사용할 수 있습니다. ASCII 파일을 보려면,

- UNIX 기반 플랫폼의 경우에는 Release.Notes 파일을 참조하십시오. 이 파일은 DB2DIR/Readme/%L 디렉토리에 있으며 여기서, %L은 로케일 이름이고 DB2DIR은 다음과 같습니다.
 - AIX에서는 /usr/lpp/db2_07_01
 - HP-UX, PTX, Solaris 및 Silicon Graphics IRIX에서는 /opt/IBMd2/V7.1
 - Linux에서는 /usr/IBMd2/V7.1
- 다른 플랫폼의 경우에는 RELEASE.TXT 파일을 참조하십시오. 이 파일은 제품이 설치된 디렉토리에 있습니다. OS/2 플랫폼에서는 **IBM DB2** 폴더를 더블 클릭하고 릴리스 정보 아이콘을 더블 클릭할 수 있습니다.

PDF 책 인쇄

책의 사본을 원하는 경우 DB2 책 CD-ROM에 있는 PDF 파일을 인쇄할 수 있습니다. Adobe Acrobat Reader를 사용하여 책 전체나 특정 페이지를 인쇄할 수 있습니다. 라이브러리에 있는 각 책의 파일 이름에 대해서는 615 페이지의 표50에서 자세한 내용을 참조하십시오.

Adobe 웹 사이트인 <http://www.adobe.com>에서 Adobe Acrobat Reader의 최신 버전을 얻을 수 있습니다.

PDF 파일은 파일 확장자가 PDF로서 DB2 책 CD-ROM에 들어 있습니다. PDF 파일을 액세스하려면,

1. DB2 책 CD-ROM을 삽입하십시오. UNIX 기반의 플랫폼에서는 DB2 책 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 *빠른 시작* 책에서 자세한 내용을 참조하십시오.
2. Acrobat Reader를 시작하십시오.
3. 다음 위치에서 원하는 PDF 파일을 여십시오.
 - OS/2 및 Windows 플랫폼에서:
x:\doc\language 디렉토리. 여기서 *x*는 CD-ROM 드라이브를 나타내며 *language*는 사용자 언어를 나타내는 2문자 국가 코드를 나타냅니다. 예를 들면 영문인 경우에는 EN입니다.
 - UNIX 기반 플랫폼에서:
/cdrom/doc/%L 디렉토리. 여기서 */cdrom*은 CD-ROM의 마운트 위치이고 *%L*은 원하는 로케일의 이름입니다.

또한 PDF 파일을 CD-ROM에서 지역이나 네트워크로 파일을 복사하고 거기서 읽을 수도 있습니다.

인쇄된 책 주문

인쇄된 DB2 책은 책 주문 번호(SBOF)를 사용하여 세트나 날권으로 주문할 수 있습니다. 인쇄본을 주문하려면, IBM 협력업체 또는 영업 대표에게 문의하십시오. 또한 웹 사이트 <http://www.elink.ibm.com/pbl/pbl>에서도 책을 주문할 수 있습니다.

두 종류의 책 세트를 사용할 수 있습니다. SBOF-8935는 DB2 Warehouse Manager에 대한 참조 및 사용에 관한 정보를 제공합니다. SBOF-8931은 다른 모든 DB2 Universal Database 제품과 특징에 대한 참조 및 사용 정보를 제공합니다. 각 SBOF의 내용은 다음 테이블에 나열되어 있습니다.

표 51. 인쇄된 책 주문

SBOF 번호	포함된 책
SBOF-8931	<ul style="list-style-type: none"> • 관리 안내서: 계획 • 관리 안내서: 구현 • 관리 안내서: 성능 • Administrative API Reference • 응용프로그램 빌드 안내서 • 응용프로그램 개발 안내서 • CLI Guide and Reference • Command Reference • 데이터 이동 유틸리티 안내 및 참조서 • Data Warehouse Center 관리 안내서 • Data Warehouse Center 응용프로그램 통합 안내서 • DB2 Connect 사용자 안내서 • 설치 및 구성 보충 설명서 • Image, Audio 및 Video Extenders 관리 및 프로그래밍 • 메시지 참조서, 볼륨 1 및 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP 설정 및 사용자 안내서 • Excel용 OLAP Spreadsheet Add-in 사용자 안내서 • Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서 • 복제 안내 및 참조서 • Spatial Extender 관리 및 프로그래밍 안내서 • SQL 시작하기 • SQL 참조서, 볼륨 1 및 2 • 시스템 모니터 안내 및 참조서 • Text Extender 관리 및 프로그래밍 • 문제점 해결 안내서 • 새로운 기능
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager 사용자 안내서 • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 온라인 문서

온라인 도움말 액세스

온라인 도움말은 모든 DB2 구성요소에서 사용할 수 있습니다. 다음의 테이블에서는 다양한 도움말 유형을 설명합니다.

도움말의 유형	내용	액세스하는 방법
명령 도움말	명령행 처리기 내의 명령 구문을 설명합니다.	대화식 모드인 명령행 처리기에서, 다음을 입력하십시오. <i>? command</i> 여기서, <i>command</i> 는 키워드이거나 전체 명령입니다. 예를 들어, <i>? catalog</i> 는 모든 CATALOG 명령에 대한 도움말을 표시하고, <i>? catalog database</i> 는 CATALOG DATABASE 명령에 대한 도움말을 표시합니다.
클라이언트 구성 지원 프로그래밍 도움말	창 또는 노트북에서 수행할 수 있는 작업을 설명합니다. 도움말은 알아야 할 개요와 전체조건 정보를 포함하고, 창 또는 노트북 제어를 사용하는 방법을 설명합니다.	창이나 노트북에서, 도움말 버튼을 누르거나 F1 키를 누르십시오.
명령 센터 도움말		
제어 센터 도움말		
Data Warehouse Center 도움말		
이벤트 분석기 도움말		
Information Catalog Manager 도움말		
위성 관리 센터 도움말		
스크립트 센터 도움말		

도움말의 유형	내용	액세스하는 방법
메시지 도움말	메시지의 원인과 사용자가 취해야 할 조치를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn</code></p> <p>여기서, <code>XXXnnnnn</code>은 유효한 메시지 식별자입니다.</p> <p>예를 들어, <code>? SQL30081</code>은 <code>SQL30081</code> 메시지에 대한 도움말을 표시합니다.</p> <p>한 번에 한 화면씩 메시지 도움말을 보려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn more</code></p> <p>파일에 메시지 도움말을 저장하려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn > filename.ext</code></p> <p>여기서, <code>filename.ext</code>는 메시지 도움말을 저장하려는 파일입니다.</p>
SQL 도움말	SQL문의 구문을 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>help statement</code></p> <p>여기서, <code>statement</code>는 SQL문입니다.</p> <p>예를 들어, <code>help SELECT</code>는 <code>SELECT</code>문에 대한 도움말을 표시합니다.</p> <p>주: SQL 도움말은 UNIX 기반 플랫폼에서 사용할 수 없습니다.</p>
SQLSTATE 도움말	SQL 상태 및 클래스 코드를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? sqlstate</code> 또는 <code>? class code</code></p> <p>여기서, <code>sqlstate</code>는 유효한 5자리 숫자로 된 SQL 상태이고 <code>class code</code>는 SQL 상태의 처음 2자리 숫자입니다.</p> <p>예를 들어, <code>? 08003</code>은 <code>08003</code> SQL 상태에 대한 도움말을 표시하고, <code>? 08</code>은 <code>08</code> 클래스 코드에 대한 도움말을 표시합니다.</p>

정보 온라인 보기

이 제품에 들어 있는 책은 HTML(Hypertext Markup Language) 소프트웨어 형식으로 제공됩니다. 소프트웨어는 정보를 검색할 수 있게 하고 관련된 정보로 링크하는 하이퍼텍스트를 제공합니다. 또한, 사이트에서 라이브러리를 공유하는 것도 더 쉬워집니다.

HTML 버전 3.2 스펙을 따르는 브라우저로 온라인 책 또는 샘플 프로그램을 볼 수 있습니다.

온라인 책 또는 샘플 프로그램을 보려면:

- DB2 관리 도구를 수행할 경우, 정보 센터를 사용하십시오.
- 브라우저에서, 파일 → 페이지 열기를 클릭하십시오. 열린 페이지에 DB2 정보에 대한 설명과 링크가 들어 있습니다.
 - UNIX 기반 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

여기서 %L은 로케일 이름입니다.

- 다른 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
sql1lib\doc\html\index.htm
```

경로는 DB2가 설치된 드라이브에 있습니다.

정보 센터를 설치하지 않은 경우, **DB2 정보** 아이콘을 더블 클릭하여 페이지를 열 수 있습니다. 사용하는 시스템에 따라, 주 제품 폴더나 Windows 시작 메뉴에 아이콘이 있습니다.

Netscape 브라우저 설치

웹 브라우저를 설치하지 않은 경우, 제품 상자에 있는 Netscape CD-ROM에서 Netscape를 설치할 수 있습니다. 설치하는 방법에 대한 자세한 지시 사항은 다음을 수행하십시오.

1. Netscape CD-ROM을 삽입하십시오.
2. UNIX 기반의 플랫폼에서는 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 빠른 시작 책에서 자세한 내용을 참조하십시오.

3. 설치 지침서는 CDNAVnn.txt 파일을 참조하십시오. 여기서, nn은 2문자로 된 언어 식별자입니다. 파일은 CD-ROM의 루트 디렉토리에 있습니다.

정보 센터로 정보에 액세스

정보 센터는 DB2 제품 정보로의 빠른 액세스를 제공합니다. 정보 센터는 DB2 관리 도구를 사용할 수 있는 모든 플랫폼에서 사용할 수 있습니다.

정보 센터 아이콘을 더블 클릭하여 정보 센터를 열 수 있습니다. 사용하는 시스템에 따라 아이콘은 주 제품 폴더나 Windows 시작 메뉴의 정보 폴더에 있습니다.

또한 DB2 Windows 플랫폼에서 도구 모음이나 도움말 메뉴를 사용하여 정보 센터를 액세스할 수 있습니다.

정보 센터는 6개 유형의 정보를 제공합니다. 적당한 탭을 클릭하여 그 유형에서 지원하는 주제를 보십시오.

타스크 DB2를 사용하여 수행할 수 있는 키 타스크.

참조 키워드, 명령 및 API와 같은 DB2 참조 정보.

책 DB2 책.

문제점 해결 오류 메시지의 종류와 복구 조치.

샘플 프로그램 DB2 응용프로그램 개발 클라이언트와 함께 제공되는 샘플 프로그램. DB2 응용프로그램 개발 클라이언트를 설치하지 않은 경우, 이 탭은 표시되지 않습니다.

웹 월드 와이드 웹에서의 DB2 정보. 이 정보에 액세스하려면, 사용자의 시스템으로부터 웹으로의 연결이 있어야 합니다.

목록 중 하나에서 항목을 선택할 때, 정보 센터는 정보를 표시하기 위해 표시기를 시작합니다. 표시기는 사용자가 선택하는 정보의 종류에 따라, 시스템 도움말 표시기, 편집기 또는 웹브라우저가 될 수 있습니다.

정보 센터는 찾기 기능을 제공하므로 목록을 찾지 않고도 특정 주제를 찾을 수 있습니다.

전체 텍스트 검색을 위해서는 **DB2 온라인 정보 검색** 검색 양식으로 연결된 정보 센터의 하이퍼텍스트 링크를 따라 검색하십시오.

HTML 검색 서버는 보통 자동으로 시작됩니다. HTML 정보에서 검색 기능이 작동하지 않으면, 다음 방법 중 하나를 사용하여 검색 서버를 시작할 수 있습니다.

Windows의 경우:

시작을 클릭하고 프로그램 → IBM DB2 → 정보 → HTML 검색 서버 시작을 선택하십시오.

OS/2 경우:

OS/2용 DB2 폴더를 더블 클릭하고 HTML 검색 서버 시작 아이콘을 더블 클릭하십시오.

HTML 정보를 검색하면서 다른 문제가 생길 경우, 릴리스 정보를 참조하십시오.

주: 검색 기능은 Linux, PTX 및 Silicon Graphics IRIX 환경에서는 작동하지 않습니다.

DB2 마법사 사용

마법사는 한 번에 한 단계씩 각 작업을 수행하게 함으로써 특정 관리 작업을 완료하는 데 도움을 줍니다. 마법사는 제어 센터 및 클라이언트 구성 지원 프로그램을 통해 사용할 수 있습니다. 다음 테이블에서는 마법사를 나열하고 그 기능을 설명합니다.

주: 데이터베이스 작성, 색인 작성, 다중 사이트 갱신 구성 및 성능 구성 마법사는 파티션된 데이터베이스 환경에서 사용할 수 있습니다.

마법사	도움 대상	액세스하는 방법
데이터베이스 추가	클라이언트 워크스테이션의 데이터베이스를 카탈로그화합니다.	클라이언트 구성 지원 프로그램에서 추가를 클릭하십시오.
데이터베이스 백업	백업 계획을 결정하고, 작성하고, 일정을 세웁니다.	제어 센터에서 백업하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 백업 → 마법사를 사용한 데이터베이스 백업을 선택하십시오.
다중 사이트 갱신 구성	다중 사이트 갱신, 분선 트랜잭션 또는 2 단계 확약을 구성합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭하고 다중 사이트 갱신을 선택하십시오.

마법사	도움 대상	액세스하는 방법
데이터베이스 작성	데이터베이스를 작성한 다음, 몇 가지 기본적인 구성 작업을 수행합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭한 다음 작성 → 마법사를 사용한 데이터베이스 작성을 선택하십시오.
테이블 작성	기본 데이터 유형을 선택한 다음, 테이블에 대한 기본 키를 작성합니다.	제어 센터에서 테이블 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 테이블을 선택하십시오.
테이블 공간 작성	새로운 테이블 공간을 작성합니다.	제어 센터에서 테이블 공간 아이콘을 마우스의 오른쪽 버튼으로 선택하고 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
색인 작성	사용자의 모든 조회를 작성하고 삭제하기 위해 색인화합니다.	제어 센터에서 색인 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 색인을 선택하십시오.
성능 구성	업무 요구조건에 맞게 구성 매개변수를 갱신하여 데이터베이스의 성능을 조정합니다.	제어 센터에서 성능을 조정하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오. 파티션된 데이터베이스에 대해 데이터베이스 파티션 뷰로부터 성능을 조정하려는 첫번째 파티션을 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오.
데이터베이스 복원	실패 후에 데이터베이스를 복구합니다. 사용할 백업 위치 및 재작동할 로그 기록을 이해하는 데 도움을 줍니다.	제어 센터에서 복원하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.

문서 서버 설정

기본 값으로 DB2 정보는 지역 시스템에 설치됩니다. 이는 DB2 정보에 액세스해야 하는 모든 사람이 동일한 파일을 설치해야 함을 의미합니다. DB2 정보를 한 위치에 저장하려면, 다음과 같이 하십시오.

1. 지역 시스템의 `\sqllib\doc\html`에 있는 모든 파일과 서브디렉토리를 웹 서버로 복사하십시오. 각 책은 책을 구성하는 데 필요한 모든 HTML 및 GIF 파일이 들어 있는 서브디렉토리를 가집니다. 디렉토리 구조가 변경되지 않게 하십시오.
2. 새로운 위치에 있는 파일을 찾으려면 웹 서버를 구성하십시오. 보다 자세한 정보는 `설치 및 구성 보충 설명서`의 부록 `NetQuestion`을 참조하십시오.
3. Java 버전의 정보 센터를 이용하는 경우, 모든 HTML 파일에 대한 기본 URL을 지정할 수 있습니다. 책 목록에 대해서는 URL을 사용해야 합니다.
4. 책 파일을 열람할 수 있게 되면, 다음과 같이 자주 열람하는 주제 항목에 대해서는 북마크를 설정할 수 있습니다. 다음의 페이지들을 북마크로 설정해 두면 도움이 될 것입니다.
 - 책 목록
 - 자주 이용하는 책의 목차
 - ALTER TABLE 주제와 같은 자주 참조하는 항목
 - 검색 양식

DB2 Universal Database 온라인 문서 파일을 중앙 시스템에서 제공하는 방법에 대한 정보를 보려면 `설치 및 구성 보충 설명서`의 부록 `NetQuestion`을 참조하십시오.

정보 온라인 검색

HTML 파일에서 정보를 찾으려면, 다음 방법 중 하나를 사용하십시오.

- 맨 위 프레임에서 검색을 클릭하십시오. 특정 주제를 찾으려면 검색 양식을 사용하십시오. 이 기능은 Linux, PTX 또는 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.
- 맨 위 프레임에서 색인을 클릭하십시오. 책에서 특정 주제를 찾으려면 색인을 사용하십시오.
- 책에서 특정 주제를 찾으려면 목차나 도움말의 색인 또는 HTML 책을 표시하고 웹 브라우저의 찾기 기능을 사용하십시오.
- 특정 주제로 빨리 리턴하려면 웹 브라우저의 북마크 기능을 사용하십시오.

- 특정 주제를 찾으려면 정보 센터의 검색 기능을 사용하십시오. 628 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오.

부록M. 주의사항

IBM은 이 책에서 논의된 제품, 서비스 또는 기능을 다른 나라에서는 제공하지 않을 수 있습니다. 현재 사용자가 사용할 수 있는 제품 및 서비스에 대한 정보는 해당 지역의 IBM 영업대표에게 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스만을 사용해야 함을 의미하지는 않습니다. IBM의 지적 소유권을 침해하지 않는 기능상으로 동등한 타사의 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나, 타사 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대한 특허를 보유하고 있거나 출원중일 수 있습니다. 이 책을 제공한다고 해서 그러한 특허에 대한 사용권까지를 부여하는 것은 아닙니다. 특허 사용권에 대한 문의는 다음 주소로 하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
지적 재산권부

2바이트(DBCS) 정보에 관한 사용권 문의는 사용자 국가의 IBM 지적 재산권부나 다음 주소로 서면 문의하십시오.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 사항은 영국이나 이 조항이 현지법과 상충되는 나라에는 적용되지 않습니다. IBM에서는 이 책을 명시적 또는 암시적인 어떠한 종류의 보증없이 『있는 그대로』 제공하므로, 판매 가능성을 보장하거나 특정 목적에 적합한지 여부에 대해서는 책임질 수 없습니다. 일부 국가에서는 특정 거래의 명시적 또는 암시적인 보증을 부인하는 문장을 허용하지 않으므로, 이 사항이 사용자에게 적용되지 않을 수도 있습니다.

이 책에는 기술상 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 개정판에 통합됩니다. IBM은 사전 통지없이 언제든지 이 책에 설명된 제품과 프로그램을 개선 및 변경할 수 있습니다.

이 책에서 타사의 웹 사이트를 언급한 것은 단지 편의를 위해서일 뿐이며 이런 웹 사이트를 추천하려는 의도는 아닙니다. 이런 웹 사이트의 데이터가 이 IBM 제품에 대한 데이터의 일부는 아니므로 이런 웹 사이트 사용에 대한 책임은 사용자가 져야 합니다.

IBM은 독자가 제공한 정보를 적절한 방식으로 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

이 프로그램의 사용권자가 (i) 독립적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함한) 사이의 정보 교환과 (2) 교환된 정보의 공동 사용을 목적으로 그 프로그램에 대한 정보를 원하는 경우, 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
소프트웨어 사업부

이러한 정보는 특정한 기간 및 조건하에 사용 가능하며 어떤 경우에는 사용료를 지불해야 합니다.

이 책에 기술된 사용권 프로그램 및 이 프로그램에 사용 가능한 모든 사용권 데이터는 IBM 고객 협약, IBM 국제 프로그래밍 사용권 협약 또는 이와 동등한 모든 협약 조건하에 IBM에서 제공됩니다.

여기에 제시된 어떠한 성능 데이터는 주위 환경에 따라 결정될 수 있습니다. 따라서, 다른 운영 체제에서 제시된 결과 값과 다를 수 있습니다. 몇몇 측정값은 개발 단계에서 얻은 값일 수 있습니다. 따라서 일반적인 사용자 시스템에서 얻은 값과 다를 수 있습니다. 또한 몇몇 측정값은 보외법을 통해 측정된 값입니다. 실제 값과는 다를 수 있습니다. 이 책의 사용자는 사용자의 특정 환경에 맞게 적용가능한 데이터를 변경해야 합니다.

타사 제품과 관련된 정보는 해당 제품의 공급자, 공개 발표 또는 기타 공개적으로 사용 가능한 소스에서 확보한 것입니다. IBM은 이들 제품을 검사하지 않았고 성능상의 정확성, 호환성 또는 타사 제품과 관련된 기타 주장을 확인할 수 없습니다. 타사 제품의 성능에 관한 문제는 해당 제품의 공급자에게 제기되어야 합니다.

IBM이 제시하는 방향 또는 의도에 관한 어떠한 언급도 특별한 통지없이 변경될 수 있습니다.

이 정보는 일상적인 비즈니스 처리에 사용되는 데이터와 보고서의 예가 들어 있을 수 있습니다. 보다 구체적으로 예를 나타내기 위해 특정 개인, 회사, 상표 또는 제품 이름이 언급되는 경우가 있습니다. 여기서 언급된 이름은 가상의 이름이며 실제 비즈니스 업체가 사용하는 이름 및 주소와 유사하다면 우연인 것입니다.

사용권:

이 정보에는 여러 운영 체제에서 프로그래밍 소스 언어로 예제 응용프로그램이 들어 있을 수 있습니다. 사용자는 이들 예제 프로그램을 IBM에게 비용을 지급하지 않고 복사, 수정 및 분배할 수 있습니다. 이들 예제 프로그램은 모든 조건에서 철저히 검사되지 않았습니다. 따라서, IBM은 이들 프로그램에 대해 어떠한 보증도 할 수 없습니다.

이들 예제 프로그램의 각각의 복사본이나 특정 부분은 다음과 같은 사용권 주의 사항을 포함해야 합니다.

© (사용자 회사 이름) (년도). 이 코드의 일부는 IBM Corp. 예제 프로그램에서 발췌된 것입니다. © Copyright IBM Corp. (년도 입력). All rights reserved.

등록 상표

별표(*)로 표시된 다음의 용어는 전세계에서 IBM의 상표입니다.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

다음 용어는 해당 회사의 상표 또는 등록 상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록 상표입니다.

Java 또는 모든 Java 관련 상표 및 로고 그리고 Solaris는 전세계에서 Sun Microsystems, Inc.의 상표입니다.

Tivoli 및 NetView는 전세계에서 Tivoli Systems Inc.의 상표입니다.

UNIX는 전세계에서 X/Open Company Limited가 독점권을 갖는 등록 상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스 표시입니다.

색인

[가]

- 가비지 콜렉션 409
- 가상 인터페이스(VI) 548
- 가상 인터페이스(VI) 아키텍처 547
- 감사 기능
 - 감사 이벤트 테이블 317
 - 구문 312
 - 권한/특권 307
 - 동기 레코드 쓰기 310
 - 레코드 레이아웃 317
 - 매개변수 설명 313
 - 메시지 316
 - 비동기 레코드 쓰기 310
 - 사용 시나리오 312
 - 예 336
 - 오류 처리 310
 - 이벤트 308
 - 이벤트 테이블 점검 318
 - 작동 309
 - 조치 308
 - 추가 정보 및 기술 335
 - 활동 제어 336
 - CONTEXT 이벤트 테이블 333
 - ERRORTYPE 매개변수 310
 - OBJMAINT 이벤트 테이블 323
 - SECMAINT 이벤트 테이블 324
 - SYSADMIN 이벤트 테이블 329
 - VALIDATE 이벤트 테이블 331
- 검색
 - 온라인 정보 628, 631
 - 검색 발견
 - 추가 설정 110
 - 게이트웨이 연결 19
 - 경량 디렉토리 액세스 프로토콜 56, 129, 559
 - 검색 571
 - 노드 항목 카탈로그화 567
 - 데이터베이스 등록 567
 - 데이터베이스 등록해제 569
 - 디렉토리 스키마 확장 578
 - 레지스트리 변수 설정 573
 - 보안 576
 - 서버 등록해제 567
 - 오브젝트 클래스 및 속성 581
 - 원격으로 접속 568
 - 작동 가능 574
 - 작동 불가능 575
 - 프로토콜 정보 갱신 566
 - 항목 새로 고침 569
 - 호스트 데이터베이스 구성 571
 - DB2 연결 575
 - IBM eNetwork Directory 578
 - Windows 2000 Active Directory 579
- 계층구조 테이블 155
 - 삭제 234
- 고속 통신 545
- 고유 제한조건
 - 삭제 222
 - 정의 145
 - 추가 218
- 고유 제한조건 삭제 222
- 고유 제한조건 정의 145
- 고유 제한조건 추가 218
- 고유 키
 - 삭제에 필요한 권리 222
 - 작성할 시기 146
 - 추가 219
- 고유 키 (계속)
 - 1차 색인 146
 - 1차 색인, 작성 188
 - DROP PRIMARY KEY절, ALTER TABLE문 222
 - 고유하지 않은 색인
 - 삭제 244
 - 공용 상호연결 546
 - 관련 내용 표시 8
 - 관련됨
 - 구성 118
 - 관리 도구
 - 개요 4
 - 명령 센터 24
 - 스크립트 센터 24
 - 관리 서버 17, 94
 - 관리 클라이언트 4
 - 구성 매개변수
 - 데이터베이스 기록 382
 - 파티션된 데이터베이스 70
 - AUTORESTART 349
 - DCE 467
 - 구조화 유형
 - 변경 231
 - 구조화 유형 변경 231
 - 권한 276
 - 데이터베이스 관리(DBADM) 279, 282
 - 레벨 274
 - 시스템 유지보수(SYSMAINT) 278
 - 시스템 제어(SYSCtrl) 277
 - 타스크 및 필수 권한 부여 301
 - BACKUP 명령에 필요한 357
 - RESTORE 명령에 필요한 366

권한 276 (계속)

- ROLLFORWARD 명령에 필요한 392
- SYSADM으로부터 DBADM 제거 277
- SYSCTRL으로부터 DBADM 제거 278

권한 및 특권의 취소 47

권한 부여

- 시스템 관리(SYSADM) 276
- 정의 273
- 트러스트된 클라이언트 255

권한 부여문

- 내재적인 발행(issuance) 294
- 사용 291

권한 취소문

- 내재적인 발행(issuance) 294
- 사용 292
- 예 292

권한과 특권 권한 부여 47

권한부여 이름

- 권한 부여된 특권의 검색 304
- 테이블 액세스 권한을 갖는 이름 검색 303
- 특권 정보를 검색 303
- 특권 정보에 대한 뷰 작성 306
- DBADM 권한이 있는 이름 검색 303

규정화된 오브젝트 이름 65

기록(logging)

- raw 장치 137

기본 속성 스펙 141

[나]

널(NULL) 값

- 컬럼 정의 141

노드 66

- 노드 그룹에서 변경 209
- 데이터 노드 목록 판정 356

노드 66 (계속)

- 모든 노드에 걸쳐 데이터베이스 작성 69
- 사용중인 데이터베이스 파티션 서버의 트랜잭션 복구 350
- 실패한 데이터베이스 파티션 서버, 복구 354
- 오류데이터베이스 파티션 서버에서의 트랜잭션 복구 351
- 카탈로그화 69

노드 구성 파일

- 변경 206
- 작성 115

노드 구성 파일 변경 206

노드 그룹

- 데이터베이스 파티션 서버가 사용중일 때 트랜잭션 복구 350
- 변경 209
- 실패한 데이터베이스 파티션 서버 복구 354
- 오류 데이터베이스 파티션 서버에서의 트랜잭션 복구</ idxterm> 351
- 작성 129
- 처음 정의 123
- 테이블 고려사항 156
- 파티션 키, 변경 228
- IBMDEFAULTGROUP, 기본값에 따라 작성된 테이블 156

노드 그룹 변경 209

노드 레벨 프로파일 레지스트리 85

노드 번호 115

노드 서버 옵션 182

노드간 통신 545

[다]

다중 논리 노드 543

다중 데이터베이스 파티션 서버로 명령 발행 501

다중 사이트 갱신 구성 마법사 629

다중 사이트 갱신 구성> 6

다중 인스턴스 64

다중 테이블 공간에 테이블 작성 155

대소문자 구별명, 연합 데이터베이스 450

대형 오브젝트(LOB)

- 컬럼 고려사항 143

데이터

- 분산 변경 209

데이터 검색

- 색인 190

데이터 무결성

- 고유 색인 187

데이터 보안

- 데이터베이스 액세스 제어 249
- 시스템 카탈로그의 보안 305
- 중요성 249

데이터 복제 54

데이터 유형

- 다중 바이트 문자 세트 141
- 컬럼 정의 141

데이터 이동 48, 343

데이터 재분배

- 노드에 걸쳐 209

데이터 전송

- 개요 343

데이터 파티션 나누기 69

데이터베이스 61

- 데이터 노드 목록 판정 356
- 데이터 분산 변경 209
- 데이터베이스 파티션 서버가 사용중일 때 트랜잭션 복구 350
- 롤 포워드 변경사항 380
- 모든 노드에 걸쳐 작성 69
- 백업 355
- 변경 208
- 변경 노드 그룹 209
- 변경전 고려사항 201
- 복구 로그 130
- 복원 365
- 삭제 208

데이터베이스 61 (계속)	데이터베이스 백업	데이터베이스 파티션 서버 (계속)
시스템 손상(crash) 복구 351	공정 디스크 미디어 364	Windows NT 537
실폐한 데이터베이스 파티션 서버 복구 354	데이터베이스 백업 마법사 629	도메인 보안
오류 데이터베이스 파티션 서버에서의 트랜잭션 복구 351	데이터베이스 별명 443	인증 526
오브젝트 이름 지정 규칙 443	이름 지정 규칙 444	동시처리 제어
이름 지정 규칙 444	BACKUP 명령 361	BACKUP 명령 362
작동 가능한 데이터 파티션 나누기 69	RESTORE 명령 367	동적 SQL
작성 121	데이터베이스 복원	데이터베이스 액세스에 대한 EXECUTE 특권 295
작성에 대한 고려사항 73	개요 365	디렉토리
작성하기 전 61	데이터베이스 파티션 서버가 사용중일 때 트랜잭션 복구 350	노드 디렉토리 127
재시작 후 불일치 351	실폐한 데이터베이스 파티션 서버 복구 354	시스템 데이터베이스 디렉토리 127
카탈로그화 131	오류 데이터베이스 파티션 서버에서의 트랜잭션 복구 351	지역 데이터베이스 디렉토리 126
트랜잭션 복구, 개요 349	트랜잭션 복구, 개요 349	디렉토리 오브젝트
패키지 종속성 244	RESTORE 명령 365	오브젝트 클래스 속성 459
데이터베이스 관리 프로그램 349	데이터베이스 복원 마법사 6	작성 454
데이터베이스 파티션 서버가 사용중일 때 트랜잭션 복구 350	데이터베이스 삭제 208	디렉토리 캐쉬
색인 190	데이터베이스 설계	카탈로그화 데이터베이스의 결과 132
시작 62	변경 201	
실폐한 데이터베이스 파티션 서버 복구 354	데이터베이스 시드 370	
액세스 제어 290	데이터베이스 액세스	
오류 데이터베이스 파티션 서버에서의 트랜잭션 복구 351	제어 249	
유틸리티 바인딩 131	SQL이 있는 패키지를 통한 특권 295	
이름 지정 규칙 443	데이터베이스 오브젝트	
중지 72	액세스 제어 290	
트랜잭션 복구, 개요 349	예 454	
데이터베이스 관리자(DBADM) 권한	이름 지정 규칙 447	
이름 검색 303	작성 454	
특권 279	데이터베이스 위치 지정자 오브젝트	
데이터베이스 구성	예 456	
변경 206	작성 456	
작성된 파일 117	데이터베이스 작성 마법사 5, 629	
데이터베이스 구성 변경 206	데이터베이스 추가 마법사 629, 630	
데이터베이스 로그	데이터베이스 카탈로그화 131	
구성 매개변수 382	예 131	
	데이터베이스 파티션 서버	
	Windows 2000 537	

[라]

라우팅 정보 오브젝트
예 458
작성 457
램퍼 삭제 240
램퍼 작성 175
램퍼, 작성 175
레지스트리 변수
변경 206
DB2LOADREC 395
DCE 467
레지스트리 변수 변경 206
레코드
감사 307
로그
감사 307
관리 400
시간소인의 사용 402
식별 400
위치 401

- 로그 (계속)
 - 유실 405
- 로그 순서 410
- 로그 유실 405
- 로그 파일 검색(retrieve)
 - OS/2용 486
 - UNIX 시스템용 486
- 로그 파일들 아카이브
 - OS/2용 486
 - UNIX 시스템용 486
- 롤 포워드 복구 372
 - 계획 392
 - 테이블 공간 롤 포워드 386
 - 필요한 권한 392
 - 호출 394
- 릴리스 정보 622

[마]

- 마법사
 - 다중 사이트 갱신 구성 629
 - 데이터베이스 백업 629
 - 데이터베이스 복원 630
 - 데이터베이스 작성 629
 - 데이터베이스 추가 629, 630
 - 색인 630
 - 성능 구성 206, 630
 - 타스크 완료 629
 - 테이블 공간 작성 630
 - 테이블 작성 630
- 마법사 성능 구성 206
- 머신 목록에서 중복된 항목 제거 511
- 메시지
 - 감사 기능 316
- 명령 센터 24
- 명령행 처리기
 - 데이터베이스에 바인딩 131
- 명시적 스키마 사용 65, 66
- 문서 서버 설정 630
- 문자 직렬 장치 135
- 문제점 판별 515

문제점 해결 53

[바]

- 바인딩
 - 데이터베이스 유틸리티 131
 - 명령행 처리기 131
 - 유효하지 않은 패키지의 리바인드 293
- 발견
 - 구성 114
 - 매개변수 설정 112
 - 서버 인스턴스 숨기기 111
- 백업 355
 - 계획 357
 - 로그 순서 410
 - 로그 체인 410
 - 만기 409
 - 미사용 409
 - 버퍼 361
 - 사용중 409
 - 이미지 363
 - 전략 플랜 설정 357
 - 정보 표시 365
 - 컨테이너 이름 363
 - 표시 유틸리티 365
 - 호출 360
 - quiesce 357
- 백업 데이터베이스 Wizard 5
- 백업 도메인 제어기
 - 사용하도록 DB2 구성 524
- 버전 복구 355
- 범위
 - 추가 217
- 범위 추가 217
- 별명
 - 권한 173
 - 데이터 소스 뷰 297
 - 사용 173
 - 이름 지정 규칙 447
 - 작성 184

별명 (계속)

- 특권 287
 - 패키지 특권 처리 295
- 별명 변경 241
- 별명 삭제 241
- 별명 식별 185
- 별명 작성 173, 184
- 별명(MVS/ESA용 DB2) 174
- 별명, 작성 173
- 병렬 명령 수행 504
- 병렬 처리
 - 작동 기능 66
- 병렬 처리, 파티션 내
 - 작동 기능 66
- 보기
 - 온라인 정보 627
- 보안
 - 사용자 맵핑 269
 - 서버 옵션 270
 - 연합 데이터베이스 ID 및 암호 처리 269
 - 연합 서버 인증 예 271
 - 연합 시스템에 대한 APPC 설정값 271
 - 인증, 연합 데이터베이스 세부사항 267
 - 플랜 설정 249
 - CLIENT 레벨 254
 - DCE 디렉토리 서비스 465
 - DCS 처리, 연합된 시스템 268
- 보안 서비스
 - Windows NT 524
- 복구
 - 개요 347
 - 데이터베이스 작성시 로그 할당 130
 - 롤 포워드 372
 - 버전 355
 - 삭제된 테이블 397
 - 시스템 손상(crash) 348
 - 실행기록 파일 406

복구 (계속)

- 일관성 있는 데이터베이스 348
- 특정 시점 405
- 2단계 요약 프로토콜 350
- DB2 Data Links Manager와의 상호 작용 419

복구 로그 130

복구 실행기록 파일 406

복수 인스턴스

- Tivoli Storage Manager과 함께 사용 429

복원

- 계획 366
- 기존 데이터베이스 370
- 버퍼 367
- 새로운 데이터베이스 371
- 재지정 369
- 호출 367

복원 마법사 630

뷰

- 데이터 무결성 167
- 데이터 보안 167
- 변경 237
- 사용 불가능 238
- 삭제 237
- 시스템 카탈로그에 대한 내포 삭제 237
- 액세스 특권, 예 298
- 이름 지정 규칙 447
- 작동 불능 요약 테이블 복구 238
- 작성 167
- 제한사항 237
- 컬럼 액세스 297
- 테이블에 대한 액세스 제어 296
- 특권 정보 306
- 행 액세스 297
- CHECK OPTION절, CREATE VIEW문 169

뷰 변경 237

뷰 삭제 237

뷰 작성 167

블록 구조화된 장치 135

비차 색인

- 삭제 244
- 응용프로그램에 대한 내포 삭제 244

[사]

사용 가능한 공간 점검(DMS) 52

사용 중인 디렉토리 지원 561

사용권 관리 84

사용권 센터 27

사용권 정보

- 변경 201

사용자 관리 47

사용자 맵핑, 작성 269

사용자 인증

- Windows NT 524

사용자 임시 테이블 공간 136

사용자 임시 테이블 공간 삭제 215

사용자 정의 구별 유형 작성 165

사용자 정의 구별 유형(UDT)

- 삭제 236
- 이름 지정 규칙 447
- 작성 164

사용자 정의 구별 유형, 작성 165

사용자 정의 구조화 유형 작성 166

사용자 정의 구조화 유형, 작성 166

사용자 정의 유형 삭제 236

사용자 정의 유형 작성 164

사용자 정의 임시 테이블 152, 235

사용자 정의 테이블 삭제 235

사용자 정의 함수(UDF)

- 비분리를 작성할 특권 281
- 삭제 236
- 유형 160
- 이름 지정 규칙 447
- 작성 160

사용자 정의 함수(UDF) 삭제 236

사용자 정의 함수(UDF) 작성 160

사용자 정의된 제어 센터 15

사용자 테이블 공간 삭제 213

사용자 ID

- 이름지정 445

사용중인 디렉토리 559

- 디렉토리 스키마 확장 579

보안 577

- 사용하도록 구성 561
- 오브젝트 581

삭제된 테이블

- 복구 397

색인

- 고유하지 않은 192
- 기본 146
- 변경 243
- 비차 244
- 사용된 방법 191
- 사용자 정의 확장 194
- 선택 200
- 온라인 재구성 189, 193
- 이름 지정 규칙 447
- 작성 187
- 정의 188
- 최적화 수 188
- 특권 290
- 1차 192
- 1차 : 사용자 정의 188
- CREATE INDEX문 192
- CREATE UNIQUE INDEX문 192
- DROP INDEX문 243

색인 마법사 630

색인 삭제 243

색인 스택 삭제 243

색인 스펙 작성 187

색인 작성 187

색인 키, 정의 188

색인 확장 187

색인 확장 삭제 243

색인 확장 작성 187

색인 Wizard. 6

샘플 프로그램
 상호 플랫폼 621
 HTML 621
 샘플 User Exit 프로그램
 개요 487
 OS/2용 488
 UNIX 시스템용 489
 서버
 작성 176
 특권 288
 서버 변경 241
 서버 삭제 241
 서버 옵션
 노드 182
 보안 세부사항 270
 암호 182, 270
 푸시다운 182
 collating_sequence 180
 comm_rate 180
 connectstring 180
 cpu_ratio 181
 dbname 181
 fold_id 181, 270
 fold_pw 181, 270
 io_ratio 181
 plan_hints 182
 varchar_no_trailing_blanks 183
 서버 작성 176
 선택 200
 설계, 구현 61
 설치
 Netscape 브라우저 627
 성능
 요약 테이블 170
 카탈로그 정보, 경합 감소 69
 성능 구성 마법사 630
 성능 모니터
 Windows NT 531
 성능 정보
 값 재설정 535

성능 정보 (계속)
 원격 액세스 작동 532
 원격에 액세스 534
 표시 533
 셀 디렉토리 서비스(CDS) 453
 수직 문자열 컬럼옵션 242
 스칼라 UDF 160
 스크립트 센터 24
 기존 스크립트 사용 25
 스키마
 개요 65
 삭제 215
 이름 지정 규칙 446
 작성 138
 SESSION 235
 스키마 삭제 215
 스키마 설정 139
 스키마 작성 138
 시간소인
 로그 402
 시스템 관리(SYSADM) 권한 276
 개요 276
 특권 276
 시스템 데이터베이스 디렉토리
 개요 127
 시스템 손상(crash) 복구 348
 트리거 351
 시스템 임시 테이블 공간 삭제 214
 시스템 카탈로그
 보안 305
 뷰 내포의 삭제 238
 새로운 컬럼 추가 216
 설정 125
 이름에 대해 권한 부여된 특권의 검색
 304
 테이블 삭제 233
 테이블 액세스 권한을 갖는 이름 검색
 303
 특권 목록 302

시스템 카탈로그 (계속)
 특권이 있는 권한 부여 이름 검색
 303
 DBADM 권한을 갖는 이름 검색
 303
 시스템 카탈로그 테이블
 데이터베이스 카탈로그 노드에 저장
 69
 식별 컬럼 153
 신뢰 받지 못한 클라이언트 254, 255
 [아]
 암호
 변경 445
 이름지정 445
 암호 변경 445
 암호 서버 옵션 182
 액세스 제어 253
 데이터베이스 관리 프로그램 290
 데이터베이스 오브젝트 290
 인증 253
 테이블에 대한 뷰 296
 언어 식별자
 책 621
 연합 데이터베이스
 대소문자가 구별되는 이름 450
 램프, 작성 175
 별명 참조 185
 별명, 식별 185
 별명, 작성 184
 별명, 작업 185
 사용자 맵핑, 작성 269
 색인 스펙, 작성 187
 서버 옵션, 보안 270
 서버, 작성 176
 오브젝트 이름 448
 유형 맵핑, 작성 166
 인증 세부사항 267
 인증 예 271
 함수 맵핑, 작성 162

연합 데이터베이스 (계속)	외부 키 (계속)	인스턴스 17 (계속)
함수 템플릿, 작성 163	DROP FOREIGN KEY절, ALTER	다중 수행 83
APPC 설정값 271	TABLE문 222	단점 74
DCS 설정값 268	IMPORT 유틸리티, 참조 무결성 포함	데이터베이스 파티션 서버 나열 537
ID 및 암호를 데이터 소스로 전달	150	디렉토리 74
269	LOAD 유틸리티, 참조 무결성 포함	변경 201
오류 조절	149	사용 이유 73
OS/2를 위한 User Exit 프로그램	요약 테이블	소유자 77
499	데이터 새로 고침 231	시작 62
User Exit 프로그램 497	등록 정보 변경 230	자동-시작 83
오류 처리	삭제 239	작성 74
가득 찬 로그 382	작동 불능 요약 테이블 복구 240	정의 73
액세스 오류, BACKUP 명령 362	작성 170	제거 204
액세스 오류, RESTORE 명령 369	요약 테이블 등록 정보 변경 230	중지 72
BACKUP시 시스템 장애 362, 374	요약 테이블 삭제 239	파티션 서버 변경 540
오브젝트 이름, 연합 데이터베이스 448	요약 테이블의 데이터 새로 고침 231	파티션 서버 삭제 541
오브젝트 클래스 속성	원격 관리 102	파티션 서버 추가 538
DB_Authentication(DAU) 460	원격 데이터베이스 관리 45	현재 설정 82
DB_Comment(DCO) 460	원격 시스템 45	인스턴스 레벨 프로파일 레지스트리 85
DB_Communication_Protocol 460	위치 21	인스턴스 목록 갱신 113
DB_Database_Locator_Name 462	유형 맵핑 작성 166	인스턴스 사용자
DB_Database_Protocol 462	유형 맵핑, 작성 166	환경 설정 75
DB_Native_Database_Name 462	응용프로그램 349	인스턴스 소유자 77
DB_Object_Type 463	데이터베이스 파티션 서버 오류 발생	인스턴스 프로파일 레지스트리 85
DB_Principal(DPR) 460	감지 354	인증 253
DB_Product_Name 463	데이터베이스 파티션 서버가 사용중일	그룹 526
DB_Product_Release 463	때 트랜잭션 복구 350	도메인 보안 526
DB_Target_Database_Info 463	오류 데이터베이스 파티션 서버에서의	연합 데이터베이스 처리 267
온라인 도움말 625	트랜잭션 복구 351	원격 클라이언트 259
온라인 재구성	트랜잭션 복구, 개요 349	정의 253
색인 189	응용프로그램용 LDAP 사용자 구성 564	파티션된 데이터베이스 고려사항 260
온라인 정보	이름지정	DCE 디렉토리 서비스 465
검색 631	Windows NT 제한사항 524	DCE 보안 서비스 260
보기 627	이미지	인증 유형 253
외부 키	백업 363	CLIENT 254
복합 148	인스턴스 17	DCE 257
삭제에 필요한 권리 222	개요 64	DCE_SERVER_ENCRYPT 257
외부 키 정의에 대한 규칙 148	갱신 202	DCS 256
제한조건 이름 148	기본값 74	DCS_ENCRYPT 256
추가 219	나열 82, 202	KERBEROS 257

인증 유형 253 (계속)
 KRB_SERVER_ENCRYPT 258
 SERVER 253
 SERVER_ENCRYPT 254
 임시 테이블
 사용자 정의 152
 사용자 정의 삭제 235
 임시 테이블 공간 시스템 136
 입력된 뷰 작성 170
 입력된 뷰, 작성 170
 입력된 테이블
 계층구조 테이블 155
 내용 수록 155
 작성 154
 행 갱신 231
 행 삭제 231
 입력된 테이블 갱신 231
 입력된 테이블 작성 154
 입력된 테이블에 데이터 수록 155
 입력된 테이블의 행 삭제 231

[자]

자동 재시작 349
 자동 불능 뷰 복구 238
 자동 불능 요약 테이블 복구 240
 저널 열기 26
 저장된 명령 스크립트 스케줄 26
 저장영역 관리 51
 전역 디렉토리 서비스(GDS) 453
 전역 레벨 프로파일 레지스트리 85
 전용 상호연결 547, 548
 점검 제한조건 정의 150
 접두부 순서 511
 정보 센터 628
 정적 SQL
 데이터베이스 액세스에 대한
 EXECUTE 특권 295
 제어 센터
 시스템 표시 17

제한사항
 Windows NT 이름지정 524
 제한조건
 고유 정의 145
 변경 218
 삭제 221
 추가 218
 제한조건 변경 218
 제한조건 이름
 외부 키 정의 149
 테이블 점검 제한조건 정의 150
 제한조건 추가 218, 221
 조정 보류 상태 417
 조정자 노드에서 트랜잭션 복구 350
 조회 제작성
 요약 테이블 170
 지역 데이터베이스 디렉토리
 개요 126

[차]

참조 제한조건
 정의 146
 테이블에 추가 219
 FOREIGN KEY절,
 CREATE/ALTER TABLE문 146
 PRIMARY KEY절,
 CREATE/ALTER TABLE문 146
 REFERENCES절, CREATE/ALTER
 TABLE문 146
 참조 제한조건 정의 146
 책 613, 623
 총계 함수 160
 최소 복구 시간 395
 최신 정보 622
 추적
 감사 307

[카]

카탈로그 노드
 설명 69
 컨테이너
 수정(DMS 테이블 공간에) 211
 추가(DMS 테이블 공간에) 210
 컨테이너 이름 363
 컬럼
 변경 217
 이름 지정 규칙 447
 정의 141
 추가 216
 컬럼 변경 217
 컬럼 생성 151, 224
 컬럼 수정 217
 컬럼 옵션
 수치 문자열 242
 varchar_no_trailing_blanks 243
 컬럼 UDF 160
 코드 페이지

RESTORE 명령 369
 콜 레벨 인터페이스
 데이터베이스에 바인딩 131
 크기 계산 51
 클라이언트
 데이터베이스 백업, 제한사항 375
 신뢰 받지 못한 254, 255
 신뢰받은 254

[타]

테이블
 고유 제한조건 정의 145
 노드 그룹 할당 129
 변경 216
 삭제 233
 새로 이름 지정 231
 속성 변경 229
 식별 컬럼 153

테이블 (계속)	테이블 공간 컨테이너	트러스트된 클라이언트
엑세스를 갖는 이름 검색 303	재정의 369	인증 254, 255
이름 지정 규칙 447	테이블 공간으로 공간 추가 52	CLIENT 레벨 보안 254
이름지정 140	테이블 변경 216	트리거
임시 123	테이블 삭제 233	삭제 235
점검 제한조건 정의 150	테이블 속성 변경 229	이름 지정 규칙 447
참조 제한조건 추가 219	테이블 수정 216	이점 158
참조 제한조건의 정의 147	테이블 이름 변경 231	작성 158
컬럼 생성 151, 224	테이블 작성 140	종속성 160
특권 권한 취소 292	테이블 작성 마법사 5, 630	트리거 삭제 235
파티션 키 변경 228	테이블 점검 제한조건 220	트리거 작성 158
파티션된 데이터베이스에서 작성 156	삭제 223	특권
취발성 228	정의 150	간접 특권, 별명 295
ALTER TABLE문 217	추가 220	개별 275
CREATE TABLE문 140	테이블 점검 제한조건 삭제 223	계층 274
테이블 공간	테이블 UDF 161	권한 부여 이름 검색 303
기존 데이터베이스로 복원 371	테이프 백업	권한의 권한 부여 및 취소 281
노드 그룹에서 137	계획 359	데이터베이스 관리 프로그램 281
데이터 유형의 분리, 예 156	테이프 시스템	별명 287
데이터베이스 작성시 기본값 123	백업 고려사항 358	별명을 가진 뷰 297
변경 209	테이프 장치	뷰 284
사용자 임시 136	SCO 358	서버 288
사용자 임시 삭제 215	통신	소유권(CONTROL) 275
삭제 213	고속 545	스키마 283
시스템 임시 삭제 214	통신 프로토콜	시스템 카탈로그 목록 302
이름 바꾸기 212	VI 아키텍처 547	요약 274
이름 지정 규칙 447	트랜잭션 349	이름에 대한 검색 304
임시 시스템 136	사용중인 데이터베이스 파티션 서버의	정보에 대해 뷰 작성 306
작성 132	오류 복구 350	정의 273
장치 컨테이너 예 134	실패한 데이터베이스 파티션 서버 복	타스크 및 필수 권한 부여 301
추가 컨테이너 210	구 354	테이블 284
컨테이너 크기재조정 211	오류 348	테이블 공간 284
컨테이너 확장 211	오류 데이터베이스 파티션 서버에서의	패키지 289
특권 284	오류 복구 351	패키지에 대한 내포 275
파일 시스템 컨테이너 예 133	오류 복구, 개요 349	ALTER 285
파일 컨테이너 예 133	트랜잭션 실패	BINDADD 281
테이블 공간 변경 209	실패한 데이터베이스 파티션 서버 복	CONNECT 281
테이블 공간 이름 바꾸기 212	구 354	CONTROL 284
테이블 공간 작성 132	오류 데이터베이스 파티션 서버에서의	CREATETAB 281
테이블 공간 작성 마법사 6, 630	오류 복구 351	CREATE_NOT_FENCED 281

특권 (계속)
 DELETE 285
 IMPLICIT_SCHEMA 281
 INDEX 290
 INSERT 285
 PUBLIC 282
 REFERENCES 285
 REVOKE문 292
 SELECT 285
 *GRANT문 291

함수 호출
 선택 200
 환경 변수 84
 변경 206
 OS/2에서 설정 88
 UNIX에서 설정 92
 Windows 95에서 설정 90
 Windows NT에서 설정 90
 환경 변수 변경 206
 활동 감사 307
 희소 파일 할당 144

ALTER TABLE문 (계속)
 키 삭제의 예 223
 키 추가의 예 220
 ALTER 특권, 정의 285
 ATTACH 명령
 개요 65
 DCE 정보 지정 469
 AUDIT_BUF_SZ 310

[파]

파티션 내 병렬 처리
 작동 가능 66
 파티션 키
 변경 228
 테이블 고려사항 156
 파티션 키에 파티션이 설정된 색인
 190
 파티션 키 변경 228
 패키지
 삭제 244
 소유자 294
 외부 키를 추가한 후에는 유효하지 않
 음 219
 작동 불능 245
 종속성 244
 특권 289
 특권 권한 취소 293
 SQL로 특권 액세스 295
 푸시다운 서버 조작 182
 프로파일 레지스트리 84
 필터 11

[하]

함수 맵핑 작성 162
 함수 맵핑, 작성 162
 함수 템플릿 작성 163
 함수 템플릿, 작성 163

[숫자]

1차 색인
 고유 키에 대한 고유성 146
 삭제 244
 2단계 요약 프로토콜 350
 2단계 요약중 이상 실패 트랜잭션
 호스트에서 복구 436
 DB2 동기 지점 관리 프로그램을 사용
 하는 경우 복구 436
 DB2 동기 지점 관리 프로그램을 사용
 하지 않는 경우 복구 438
 2바이트 문자 세트 사용자
 데이터 유형 141

A

ALTER COLUMN 217
 ALTER NICKNAME문, 예 242
 ALTER SERVER문, 예 241
 ALTER TABLESPACE문
 예 211
 ALTER TABLE문
 고유 제한조건 삭제의 예 222
 고유 제한조건 추가의 예 218
 점검 제한조건 삭제의 예 224
 점검 제한조건 추가의 예 221
 제한조건 추가에 대한 도움말 219
 컬럼 추가의 예 217

B

BACKUP DATABASE 유틸리티
 OS/2를 위한 User Exit 프로그램
 486
 User Exit 프로그램에 대한 고려사항
 496
 User Exit를 위한 오류 조절 499
 BACKUP 명령
 개요 355
 고려사항 356
 데이터베이스 별명 제한사항 361
 동시처리 제어 362
 버퍼 361
 시스템 장애 362, 374
 액세스 오류, 오류 처리 362
 작성된 디스크 출력 363
 작성된 테이프 출력 364
 필요한 권한 357
 DB2 Data Links Manager 고려사항
 414
 Tivoli Storage Manager과 함께 사용
 425
 BIND 명령
 OWNER 옵션 294
 BIND 특권
 정의 289
 BINDADD 특권, 정의 281

C

CATALOG GLOBAL DATABASE 명령
DCE 정보 지정 469
CDS 453
CLIENT 레벨 보안 254
CLIENT, 인증 유형 254
collating_sequence 서버 옵션 180
comm_rate 서버 옵션 180
CONNECT 특권, 정의 281
CONNECT문
DCE 정보 지정 469
CONTROL 특권
내재적인 발행(issuance) 294
정의 284
패키지 특권 289
cooked 장치 135
cpu_ratio 서버 옵션 181
CREATE ALIAS문
사용 173
예 174
CREATE DATABASE 명령
예 122
CREATE INDEX문
고유 색인 192
예 192
온라인 재구성 189, 193
CREATE NICKNAME 184
CREATE SERVER 176
CREATE TABLESPACE문
예 133
CREATE TABLE문
다중 테이블 공간의 사용 156
예 142
점검 제한조건의 정의 150
참조 제한조건의 정의 147
CREATE TRIGGER문
예 159

CREATE VIEW문
예 168
컬럼 이름의 변경 169
CREATE WRAPPER 175
CREATETAB 특권, 정의 281
CREATE_NOT_FENCED 특권, 정의 281
CURRENT SCHEMA 139
CURRENT SCHEMA 특수 레지스터 66

D

DAS 구성 갱신 114
Datalink_Reconcile_Not_Possible 상태 416
DataPropagator Relational(DPROPR) 개요 343
DAU(DB_Authentication) 460
DB2
Windows NT에서 시작 63
DB2 Connect 343
DB2 Data Links Manager
가비지 콜렉션 410
고려사항 413
단계 413
데이터베이스 복원 418
로그 끝으로 데이터베이스 롤 포워드 418
로그 끝으로 테이블 공간 롤 포워드 418
롤 포워드 없이 오프라인 백업에서 데이터베이스 복원 417
롤 포워드 유틸리티 고려사항 415
링크 파일 415
링크해제된 파일 415
백업 유틸리티 고려사항 414
복구 상호작용 419
복원 유틸리티 고려사항 415
시스템 손상(crash) 복구 413
조정 423

DB2 Data Links Manager (계속)
조정 프로시저어 425
조정을 필요로하는 상황 검출 424
테이블 공간 복원 418
특정 시점 롤 포워드 예 419
특정 시점으로 데이터베이스 롤 포워드 418
특정 시점으로 테이블 공간 롤 포워드 418
2 단계 요약 413
2단계 요약중 이상 실패 트랜잭션 414
Datalink_Reconcile_Not_Possible 상태 416
Datalink_Reconcile_Not_Possible 상태에서 테이블 제거 422
Datalink_Reconcile_Pending 상태 417
DB2 관리 서버
구성 갱신 114
인스턴스 목록 갱신 113
클라이언트 구성 지원 및 제어 센터 사용 113
DB2 관리 서버(DAS) 102
개요 94
갱신 100
구성 99, 104
나열 99
레지스트리 변수 108
레지스트리 변수 고려사항 108
발견 작동 기능 109
보안 107
보안 고려사항 100
서비스 포트 105
소유권 규칙 93
시작과 중지 98
인터노드 관리 통신 105
작성 96
제거 101
제어 센터 통신 105

DB2 관리 서버(DAS) 102 (계속)	db2dmnbckctlr	DB_Native_Database_Name(DNN)
통신 105	사용 524	462
파티션된 데이터베이스 시스템 설정	db2gncol 유틸리티 225	DB_Object_Type(DOT) 463
102	db2icrt 명령 79	DB_Principal(DPR) 460
예 103	db2idrop 204	DB_Product_Name(DPN) 463
파티션된 데이터베이스 시스템(UNIX)	db2ilist 202	DB_Product_Release(DRL) 463
의 인티노드 관리 통신 106	DB2INSTANCE 환경 변수	DB_Target_Database_Info(DTI) 463
파티션된 데이터베이스 시스템	기본 인스턴스 정의 64	DCE
(Windows NT)의 인티노드 관리통	db2iupdt 202	구성 매개변수 및 레지스트리 변수
신 107	DB2LDAP_CLIENT_PROVIDER 561	467
환경 108	db2ldcfg 유틸리티 564	디렉토리 검색 방법 475
UNIX EEE 서버 106	DB2LOADREC 395	디렉토리 서비스 사용 479
Windows NT EEE 서버 107	db2nchg 540	디렉토리 서비스 제한사항 482
DB2 동기 지점 관리 프로그램	db2nrcrt 538	디렉토리 서비스 타스크 478
2단계 요약중 이상 실패 트랜잭션 복	db2ndrop 541	디렉토리 서비스의 개요 128
구 436	db2nlist 537	보안 서비스 260
DB2 라이브러리	db2nodes.cfg 파일 115	인증 260
구성 방법 613	db2perfc 535	제한사항 265
마법사 629	db2perfi 531	ATTACH 명령 469, 475
문서 서버 설정 630	db2perfr 532	CATALOG GLOBAL DATABASE
온라인 도움말 625	db2set 명령 84, 86	명령 469
온라인 정보 검색 631	db2start 명령 62	CDS 453
온라인 정보 보기 627	DB2START로 충돌 복구 트리거 351	CONNECT문 469, 476
인쇄된 책 주문 623	db2stop 명령 72	DB2 사용자 설정 261
정보 센터 628	db2uexit	DB2 서버 설정 263
책 613	OS/2를 위한 User Exit 프로그램	DB2 클라이언트 인스턴스 설정 265
책에 대한 언어 식별자 621	488	DCE 디렉토리 정보를 임의로 겹쳐쓰
최신 정보 622	UNIX 시스템을 위한 User	기 478
PDF 책 인쇄 623	Exit 489	GDS 453
DB2 시작 62	db2_all 501, 502	DCE 네트워크 데이터베이스
DB2 중지 72	db2_call_stack 502	연결 472, 474
db2adutl 유틸리티 430	db2_kill 502	작성 471
DELETE 명령 435	dbname 서버 옵션 181	DCE, 인증 유형 257
EXTRACT 명령 434	DB_Authentication(DAU) 460	DCE_SERVER_ENCRYPT, 인증 유형
QUERY 명령 434	DB_Comment(DCO) 460	257
db2adutl 유틸리티 예 435	DB_Communication_Protocol(DCP)	DCO(DB_Comment) 460
db2audit 312	460	DCP(DB_Communication_Protocol)
db2audit.log 307	DB_Database_Locator_Name(DLN)	460
db2ckbkp 365	462	DCS
	DB_Database_Protocol(DDP) 462	연합 데이터베이스 처리 268

DCS (계속)
 인증 유형 256
 DCS_ENCRYPT, 인증 유형 256
 DDL 생성 9
 DDP(DB_Database_Protocol) 462
 DECLARE GLOBAL TEMPORARY
 TABLE 152
 DELETE 특권, 정의 285
 DETACH 명령
 개요 65
 DLN(DB_Database_Locator_Name)
 462
 DMS 테이블 공간
 작성 134
 DNN(DB_Native_Database_Name)
 462
 DOT(DB_Object_Type) 463
 DPN(DB_Product_Name) 463
 DPR(DB_Principal) 460
 DPROPR 343
 DRL(DB_Product_Release) 463
 DROP DATABASE 명령
 예 208
 DROP INDEX문, 예 243
 DROP NICKNAME문, 예 242
 DROP SERVER문, 예 241
 DROP TABLESPACE문, 예 213
 DROP TABLE문
 예 233
 DROP VIEW문, 예 238
 DSMI_CONFIG 426, 427
 DSMI_DIR 426, 427
 DSMI_LOG 427, 428
 DTI(DB_Target_Database_Info) 463

E

EXECUTE 특권
 동적 SQL로 데이터베이스 액세스
 295
 정의 289

EXECUTE 특권 (계속)
 정적 SQL로 데이터베이스 액세스
 295

F

FCM 통신 118
 fold_id 서버 옵션 181
 fold_pw 서버 옵션 181
 FOREIGN KEY절
 외부 키 정의에 대한 규칙 148
 참조 제한조건 148

G

GDS 453
 GRANT문
 보안 467
 GRANT문, 예 291
 GUI 도구를 사용한 관리 3

H

HTML
 샘플 프로그램 621

I

IBM eNetwork Directory
 디렉토리 스키마 확장 578
 오브젝트 클래스 및 속성 581
 IBMCATGROUP 노드 그룹 123
 IBMDEFAULTGROUP 노드 그룹 123
 IBMTEMPGROUP 노드 그룹 123
 ID 및 암호를 데이터 소스로 전달 269
 IMPLICIT_SCHEMA 권한 138
 IMPLICIT_SCHEMA 특권, 정의 281
 IMPORT 유틸리티
 데이터베이스에 바인딩 131
 참조 무결성 포함 150
 LOAD 149

INDEX 특권, 정의 285
 INSERT 특권, 정의 285
 io_ratio 서버 옵션 181

J

java 애플릿 56
 Java 애플릿으로서의 제어 센터 57

K

Kerberos 보안 프로토콜 257
 KNRBNROS 인증 유형 257
 KRB_SERVER_ENCRYPT, 인증 유형
 258

L

LDAP 56, 129, 559
 LDAP 구성 562
 LDAP 구성 지원 560
 LDAP 사용자 작성 563
 LIST NODES CMD
 데이터베이스 백업, 데이터 노드 목록
 판정 356
 LIST NODES 명령, 데이터베이스를 백
 업할 경우 사용 356
 LOAD 권한 280
 LOAD 유틸리티
 개요 343

logbufsz 구성 매개변수 384
 logfilsiz 구성 매개변수 383
 logprimary 구성 매개변수 382
 logretain 구성 매개변수 385
 logsecond 구성 매개변수 382

M

mincommit 구성 매개변수 384
 MINPCTUSED절 193

N

Netscape 브라우저
 설치 627

newlogpath 구성 매개변수 385

O

OS/2 User Exit

개요 485

검색 고려사항 493

로그 파일 검색 486

로그 파일 아카이브 486

샘플 User Exit 프로그램 488

아카이브 고려사항 493

오류 조절 499

호출 487

호출 형식 490

BACKUP DATABASE 고려사항 496

BACKUP DATABASE 유틸리티 486

db2uexit 487

db2uexit.CAD 489

db2uexit.ex1 488

db2uexit.ex2 488

db2uexit.ex3 489

db2uexit.ex4 489

RESTORE DATABASE 고려사항 496

RESTORE DATABASE 유틸리티 486

OS/390용 DB2 서브시스템 추가 18

OS/390용 DB2 오브젝트 관리 18

P

PDF 623

PDF 책 인쇄 623

plan_hints 서버 옵션 182

PRECOMPILE 명령

OWNER 옵션 294

PRIMARY KEY절

고유 키 추가 219

제한사항 145

PUBLIC

특권 282

R

RACF 467

RACF(resource access control facility) 467

rah 501, 502

rah 명령 제어 512

rah 프로세스 모니터링 505

RAHDOTFILES 514

RAHOSTFILE 511

RAHOSTLIST 511

RAHWAITTIME 505

rah에 대한 기본 환경 프로파일 설정 514

rah에 대한 머신 목록 지정 511

raw 로그 137

raw 입출력 137

raw 장치 135

REFERENCES 특권, 정의 285

REFERENCES절

사용 149

삭제 규칙 149

외부 키 추가 219

참조 제한조건 149

REORG 유틸리티

데이터베이스에 바인딩 131

RESTART DATABASE 명령 349

RESTORE DATABASE 유틸리티

OS/2를 위한 User Exit 프로그램 486

User Exit 프로그램에 대한 고려사항 496

User Exit를 위한 오류 조절 499

RESTORE 명령

개요 365

고려사항 365

데이터베이스 별명 제한사항 367

RESTORE 명령 (계속)

롤 포워드 복구에서 사용 378

버퍼 367

액세스 오류, 오류 처리 369

코드 페이지 제한사항 369

필요한 권한 366

DB2 Data Links Manager 고려사항 415

DB2 Data Links Manager, 롤 포워드 없이 데이터베이스 복원 417

Tivoli Storage Manager과 함께 사용 425

REVOKE문

보안 467

ROLLFORWARD 명령

구성 파일 매개변수 지원 382

로그 관리 고려사항 400

백업 고려사항 373

복원 고려사항 376

시간소인 394

DB2 Data Links Manager 고려사항 415

DB2 Data Links Manager, 로그 끝까지 롤 포워드 418

DB2 Data Links Manager, 특정 시점 롤 포워드 예 419

DB2 Data Links Manager, 특정 시점으로 롤 포워드 418

S

SCO UnixWare 7

테이프 장치 358

SELECT 특권, 정의 285

SELECT문

뷰를 선택 169

SERVER, 인증 유형 253

SERVER_ENCRYPT, 인증 유형 254

SmartGuides

마법사 629

SMS 테이블 공간
 작성 133

SQL 표시 7

sqluback
 지원 357

SQL문
 작동 불능 245

synonym(MVS/ESA용 DB2) 174

SYSCAT 뷰 302

SYSCATSPACE 테이블 공간 123

T

TAKEN AT 매개변수 368

TCP/IP 546

TEMPSPACE1 테이블 공간 123

Tivoli Storage Manager (TSM)
 백업 및 로그 아카이브 관리 430
 백업 제한사항 430
 사용 429
 사용자 옵션 파일(Intel에서) 428
 시간 종료 문제 해결 429
 시스템 옵션 파일(Intel에서) 428
 암호 설정(Intel에서) 428
 암호 설정(UNIX 기반 플랫폼) 427
 클라이언트 설정(Intel에서) 427
 클라이언트 설정(UNIX 기반 플랫폼) 425
 환경 변수(Intel에서) 427
 환경 변수(UNIX 기반 플랫폼) 426

BACKUP 명령과 함께 사용 425

RESTORE 명령과 함께 사용 425

U

UNIX User Exit
 개요 486
 검색 고려사항 493
 로그 파일 검색 486
 로그파일 아카이브 486
 샘플 User Exit 프로그램 489

UNIX User Exit (계속)
 아카이브 고려사항 493
 오류 조절 497
 호출 487
 호출 형식 492
 db2uexit 487
 db2uexit.cadsm 490
 db2uexit.cdisk 490
 db2uexit.ctape 490
 db2uxt2.cxbsa 490

UPDATE 특권, 정의 285

User Exit

개요 485

아카이브 및 검색 고려사항 493

오류 조절 497

BACKUP DATABASE 유틸리티 496

RESTORE DATABASE 유틸리티 496

User Exit를 위한 호출 형식
 OS/2용 490
 UNIX 시스템용 492

userexit 구성 매개변수 386

USERSPACE1 테이블 공간 123

V

VARCHAR 설정 217

varchar_no_trailing_blanks 서버 옵션 183

varchar_no_trailing_blanks 컬럼 옵션 243

VI 아키텍처 547

VTAM 467

VTAM(virtual telecommunications access method) 467

W

Windows 2000 Active Directory
 디렉토리 스키마 확장 579

Windows 2000 Active Directory (계속)
 오브젝트 581

Windows NT 사용중인 디렉토리
 오브젝트 클래스 및 속성 581

Windows NT 성능 모니터 531
 DB2 등록 531

Windows NT용 DB2 성능 카운터 531

Wizard 성능 구성 6

Wizards 5

[특수 문자]

\$RAHBUFDIR 505

\$RAHBUFNAME 505

\$RAHCHECKBUF 505

\$RAHENV 513

IBM에 문의

기술적인 문제가 발생한 경우에는 DB2 고객 지원 센터에 문의하기 전에 문제점 해결 안내서에서 제안한 조치를 검토하고 실행해 보십시오. 이것은 DB2 고객 지원 부서로 하여금 사용자를 보다 더 잘 지원할 수 있도록 사용자가 모을 수 있는 정보를 제공합니다.

DB2 Universal Database 제품에 대한 정보나 주문은 그 지역의 IBM 영업 대표나 공인 IBM 소프트웨어 재판매업자에게 문의하십시오.

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 고객 지원을 받으려면, 1-800-237-5511.
- 사용 가능한 서비스 옵션을 알려면, 1-888-426-4343.

제품 정보

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 제품 주문이나 일반 정보를 얻으려면, 1-800-IBM-CALL (1-800-426-2255)이나 1-800-3IBM-OS2 (1-800-342-6672).
- 책에 대한 주문은 1-800-879-2755.

<http://www.ibm.com/software/data/>

DB2 월드 와이드 웹 페이지에는 새로운 소식, 제품 설명, 교육 일정 등에 관한 현재의 DB2 정보를 제공합니다.

<http://www.ibm.com/software/data/db2/library/>

DB2 제품 및 서비스 기술 라이브러리는 빈도 높은 질문(FAQ), 수정사항(fixes), 책 및 최신 DB2 기술 정보에 대한 액세스를 제공합니다.

주: 이러한 정보는 영어로만 제공됩니다.

<http://www.elink.ibm.com/pbl/pbl/>

여기에서는 책을 웹 사이트에서 주문할 수 있는 방법을 제공합니다.

<http://www.ibm.com/education/certify/>

IBM 웹 사이트에서 기술 전문 인증 프로그램은 DB2를 포함하여 다른 IBM 제품의 기술 전문 인증 테스트 정보를 제공합니다.

<ftp.software.ibm.com>

anonymous로 로그인하십시오. /ps/products/db2 디렉토리에서, DB2와 많은 관련 제품에 관한 데이터, 수정사항, 도구 등을 찾을 수 있습니다.

<comp.databases.ibm-db2>, <bit.listserv.db2-l>

이러한 인터넷 뉴스 그룹으로 사용자는 DB2 제품에 대한 자신의 사용 경험을 토론할 수 있습니다.

Compuserve에서, GO IBMDB2

이 명령을 입력하여 IBM DB2 계열 포럼을 액세스하십시오. 모든 DB2 제품이 이러한 포럼을 통해 지원됩니다.

미국 외 지역에서 IBM에 연락하는 방법에 관한 정보는 *IBM Software Support Handbook*의 Appendix A를 참조하십시오. 이 문서에 액세스하려면, 웹 사이트 <http://www.ibm.com/support/>로 가서 페이지 맨 밑에 있는 IBM Software Support Handbook 링크를 클릭하십시오.

주: 일부 국가에서는 IBM 공인 딜러는 IBM 지원 센터 대신 해당 딜러 지원 부서에 연락해야 합니다.



Printed in Australia

SA30-0988-00

