

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 543 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 한국 IBM 담당자 또는 해당 지역의 IBM 지방 사무소로 문의하십시오.

IBM에 정보를 보내는 경우, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

© Copyright International Business Machines Corporation 1993, 2001. All rights reserved.

목차

이 책에 관하여	ix	스크립트 센터에서 기존 스크립트 사용	25
이 책의 사용자	x	저장된 명령 스크립트를 수행하도록 스케줄	26
이 책의 구성 방법	x	저널	26
관리 안내서의 다른 볼륨에 대한 간단한 개요	xii	작업에 대한 작업	26
관리 안내서: 계획	xii	사용권 센터	27
관리 안내서: 성능	xiv	경보 센터	28
<hr/>		클라이언트 구성 지원 프로그램	28
제1부 제어 센터를 사용한 관리.	1	성능 모니터	29
제1장 GUI 도구를 사용한 DB2 관리.	3	이벤트 모니터	31
관리 도구	4	모니터 도구 사용	31
공통 도구 기능	7	특정 시점에서 성능 모니터	34
SQL 표시 및 명령 표시	8	사전 정의된 모니터	36
관련 항목 표시	8	경보 센터에 오브젝트가 나타날 때 필요한	
DDL 생성	9	조치	38
필터	11	기간 동안 이벤트 분석	38
도움말	12	이벤트 분석기	39
제어 센터	13	SQL문 분석	42
제어 센터의 주 요소	13	조회 성능 향상	42
OS/390용 DB2에서 사용자 정의된 제어		단순 동적 SQL문 분석	43
센터 사용	15	원격 데이터베이스 관리	44
관리될 수 있는 시스템	15	사용자 관리	47
관리될 수 있는 오브젝트	16	권한 및 특권의 권한 부여와 권한 취소	47
제어 센터에서 시스템 표시	17	데이터 이동	48
OS/390용 DB2 오브젝트 관리	18	저장영역 관리	51
OS/390용 DB2 서브시스템 추가	18	테이블 및 색인 크기 계산	51
게이트웨이 연결 관리	19	테이블 공간에서 사용 가능한 공간 검사	52
제어 센터에서 수행할 수 있는 기능	20	테이블 공간으로 공간 추가	53
새로운 오브젝트 작성	21	문제점 해결	54
기존 오브젝트에 대한 작업	21	데이터 복제	55
오브젝트 찾기(OS/390용 DB2 전용)	22	LDAP(Lightweight Directory Access	
위성 관리 센터	23	Protocol) 사용	56
명령 센터	24	Java 제어 센터 사용	56
스크립트 센터	24	Java 애플릿으로서 제어 센터 수행	57
		관리용으로 Java 도구 사용	57

제2부 설계 구현	59
제2장 데이터베이스를 작성하기 전에	61
데이터베이스를 작성하기 전의 전제조건	62
DB2 시작	62
Windows NT에서 DB2 UDB 시작	63
데이터베이스 관리 프로그램의 다중 인스턴스 사용	64
스키마별 오브젝트 구성 및 그룹화	65
병렬 처리 사용	66
데이터 파티션 사용	68
DB2 중지	72
데이터베이스에서 세부사항 작성	73
데이터베이스의 논리적 특성 및 물리적 특성 설계	73
인스턴스 작성	73
사용권 관리	84
환경 변수 및 프로파일 레지스트리 설정	84
DB2 관리 서버(DAS) 작성	94
노드 구성 파일 작성	114
데이터베이스 구성 파일 작성	116
응답 파일을 사용한 구성 정보 복제	117
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	128
DCE 디렉토리 서비스	128
LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스	129
노드 그룹 작성	129
데이터베이스 복구 로그의 정의	130
유틸리티를 데이터베이스에 비인딩	131

데이터베이스 카탈로그화	131
테이블 공간 작성	133
시스템 임시 테이블 공간 작성	136
사용자 임시 테이블 공간 작성	136
노드 그룹에 테이블 공간 작성	137
원시 입출력	137
스키마 작성	140
스키마 설정	141
테이블 작성 및 데이터 처리	142
대형 오브젝트(LOB) 컬럼 고려사항	145
제한조건 정의	147
생성된 컬럼을 새 테이블에 정의	153
사용자 정의 임시 테이블 작성	154
식별 컬럼을 새 테이블에 정의	155
순차 작성	157
IDENTITY 컬럼 및 순차 비교	159
입력된 테이블 작성	160
입력된 테이블에 데이터 처리	160
계층 구조 테이블	160
다중 테이블 공간에 테이블 작성	160
파티션된 데이터베이스에서 테이블 작성	161
트리거 작성	163
트리거 종속성	165
사용자 정의 함수(UDF) 및 메소드 작성	166
함수 맵핑 작성	167
함수 템플릿 작성	168
사용자 정의 유형(UDT) 작성	170
사용자 정의 구별 유형 작성	170
사용자 정의 구조화 유형 작성	172
유형 맵핑 작성	172
뷰 작성	173
입력된 뷰 작성	176
요약 테이블 작성	176
별명 작성	179
래퍼 작성	181
서버 작성	182
서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화	183

별칭 작성	190
별칭 및 데이터 소스 오브젝트 참조.	191
별칭 및 데이터 소스 오브젝트에 대한 작업.	191
기존의 별칭 및 데이터 소스 식별	192
색인, 색인 확장 또는 색인 스펙 작성	193
색인 사용	197
CREATE INDEX문 사용.	198
사용자 정의 확장 색인 유형 작성	201
색인 유지보수의 세부사항	202
색인 검색의 세부사항	202
색인 검색 세부사항	203
색인 확장을 정의하는 시나리오	205
제4장 데이터베이스 변경	209
데이터베이스를 변경하기 전에.	209
논리 및 물리 설계 특성 변경.	209
사용권 정보 변경.	209
인스턴스 변경	209
환경 변수 및 프로파일 레지스트리 변수 변경	214
노드 구성 파일 변경.	214
데이터베이스 구성 변경.	214
데이터베이스 변경	216
데이터베이스 제거	217
노드 그룹 변경	217
테이블 공간 변경.	218
스키마 제거.	225
구조 및 내용에서 테이블 수정	225
사용자 정의 구조화 유형 변경	244
입력된 테이블의 행 삭제 및 갱신	244
기존 테이블 이름 바꾸기	244
테이블 제거.	245
사용자 정의 임시 테이블 제거	247
트리거 제거.	248
사용자 정의 함수(UDF), 유형 맵핑 또는 메소드 제거.	248

사용자 정의 유형(UDT) 또는 유형 맵핑 제거	249
뷰 변경 또는 제거	250
요약 테이블 제거.	252
랩퍼 제거	253
서버 변경 또는 제거.	254
별칭 변경 또는 제거.	254
색인, 색인 확장 또는 색인 스펙 제거	256
오브젝트 변경시의 명령문 종속성	257

제3부 데이터베이스 보안 261

제5장 데이터베이스 액세스 제어.	263
설치를 위한 사용자 ID 및 그룹 선택	264
Windows NT 플랫폼 고려사항	265
UNIX 플랫폼 고려사항.	266
일반 규칙	267
서버에 대한 인증 방법 선택	267
원격 클라이언트에 대한 인증 고려사항.	274
파티션된 데이터베이스 고려사항	274
사용자 인증을 위한 DCE 보안 서비스 사용	274
DCE용 DB2 사용자 설정 방법	275
DCE를 사용하기 위한 DB2 서버 설정 방법.	278
DCE를 사용하기 위한 DB2 클라이언트 인스턴스 설정 방법	279
DCE 보안을 사용한 DB2 제한사항	280
연합 데이터베이스 인증 처리	282
인증 설정값.	282
사용자 ID 및 암호를 데이터 소스로 전달	283
연합 데이터베이스 인증 예.	286
특권, 권한 및 권한 부여	288
시스템 관리 권한(SYSADM).	291
시스템 제어 권한(SYSCTRL).	292
시스템 유지보수 권한(SYSMAINT).	293
데이터베이스 관리 권한(DBADM)	294
LOAD 권한	295
데이터베이스 특권	296

스키마 특권	298
테이블 공간 특권	300
테이블 및 뷰 특권	300
별칭 특권	303
서버 특권	304
패키지 특권	304
색인 특권	305
순차 특권	306
데이터베이스 오브젝트에 대한 액세스 제어 특권 권한 부여	306 307
특권 권한 취소	308
오브젝트 작성 및 제거에 의한 내재적 권 한 부여 관리	310
플랜 또는 패키지의 소유권 설정	310
패키지를 통한 간접 특권 허용	311
별칭이 들어 있는 패키지를 통한 간접 특 권 허용	312
뷰를 사용하여 데이터에 대한 액세스 제어 감사 기능을 사용하여 데이터에 대한 액세 스 모니터링	313 316
데이터 암호화	317
타스크 및 필수 권한 부여	318
시스템 카탈로그 사용	320
권한 부여된 특권이 있는 권한 부여 이름 검색	321
DBADM 권한이 있는 모든 이름 검색	321
테이블에 액세스하기 위해 권한 부여된 이 름 검색	321
사용자에게 권한 부여된 모든 특권 검색	322
시스템 카탈로그 뷰 보안	323
제6장 DB2 활동 감사	325
감사 기능 작동	327
감사 기능 사용 시나리오	330
감사 기능 메시지	334
감사 기능 레코드 레이아웃	335
감사 기능 추가 정보 및 기술	353
DB2 감사 기능 활동 제어	354

제4부 데이터 이동	359
제7장 데이터 이동을 위한 유틸리티	361
제5부 복구	363
제8장 데이터베이스 복구	365
제6부 부록 및 끝머리	367
부록A. 이름 지정 규칙	369
일반 이름 지정 규칙	369
오브젝트 이름 지정 규칙	369
스키마 이름에 대한 추가 정보	371
암호에 대한 추가 정보	373
오브젝트 이름에서 분리 식별자 사용	373
연합 시스템에서 대소문자 구별 값이 보존되 는 방법	374
부록B. DCE 디렉토리 서비스 사용	377
디렉토리 오브젝트 작성	378
데이터베이스 오브젝트	378
데이터베이스 위치 지정자 오브젝트	380
경로지정 정보 오브젝트	381
각 오브젝트 클래스의 속성	383
각 속성에 관한 세부사항	384
디렉토리 서비스 보안	389
구성 매개변수 및 레지스트리 변수	391
CATALOG 및 ATTACH 명령 및 CONNECT문	393
CATALOG GLOBAL DATABASE 명 령	393
CONNECT문	393
ATTACH 명령	394
클라이언트를 데이터베이스에 연결하는 방법	394
같은 셀의 데이터베이스에 연결	396
서로 다른 셀의 데이터베이스에 연결	398
디렉토리 검색 방법	399
ATTACH 명령	399

CONNECT문	400	Windows NT용 DB2 보안 서비스	434
DCE 디렉토리 정보를 임시로 겹쳐쓰기	402	백업 도메인 제어기에서 DB2 설치	434
디렉토리 서비스 TASK	402	그룹 및 도메인 보안 인증	436
DCE 관리자 TASK	402	부록F. Windows NT 성능 모니터 사용	441
데이터베이스 관리자 TASK	403	Windows NT 성능 모니터로 DB2 등록	441
데이터베이스 사용자 TASK	405	DB2 성능 정보에 원격 액세스 사용	442
디렉토리 서비스 제한사항	406	DB2 및 DB2 Connect 성능 값 표시	443
부록C. 데이터베이스 복구를 위한 User Exit	409	원격 DB2 성능 정보 액세스	444
부록D. 다중 데이터베이스 파티션으로 명령		DB2 성능 값 재설정	445
발행	411	부록G. Windows 2000 데이터베이스 파티	
명령	411	션 서버 또는 Windows NT 작업	447
명령 설명	412	인스턴스에서의 데이터베이스 파티션 서버 나	
수행할 명령 지정	413	열	447
UNIX 기반 플랫폼에서 병렬에 있는 명령		인스턴스에 데이터베이스 파티션 서버 추가	448
수행	415	데이터베이스 파티션 변경	450
UNIX 기반 플랫폼에서 rah 프로세스 모		인스턴스에서 데이터베이스 파티션 제거	451
니터링	416	부록H. 다중 논리 노드 구성	453
추가 rah(모든 호스트 수행) 정보(Solaris		부록I. 고속 노드 간 통신	455
및 AIX 전용)	417	TCP/IP를 사용하여 고속 상호연결	456
접두부 순서	417	IBM Netfinity SP Switch 사용에 대한	
머신 목록 지정	420	전제조건	457
머신 목록에서 중복 항목 제거	421	VI를 사용하여 고속 상호연결	457
rah 명령 제어	422	가상 인터페이스(VI) 하드웨어 설정	459
UNIX 기반 플랫폼의 \$RAHDOTFILES	423	VI를 사용하여 DB2 사용	468
Windows NT에 기본 환경 프로파일 설정	424	부록J. LDAP(Lightweight Directory	
UNIX 기반 플랫폼에서 rah 문제점 판별	425	Access Protocol) 디렉토리 서비스	471
부록E. Windows NT용 DB2가 Windows		LDAP 클라이언트 및 서버 구성 지원	472
NT 보안을 사용하여 작업하는 방법	429	Windows 2000에서 활성 디렉토리 지원	473
서버 인증에 대한 샘플 시나리오	430	활성 디렉토리로 DB2 구성	473
클라이언트 인증 및 Windows NT 클라이언		IBM LDAP 환경에서 DB2 구성	474
트 머신에 대한 샘플 시나리오	431	LDAP 사용자 작성	475
클라이언트 인증 및 Windows 95 클라이언		DB2 응용프로그램용 LDAP 사용자 구성	476
트 머신에 대한 샘플 시나리오	432	설치 후 DB2 서버의 등록	476
DB2로 백업 도메인 제어기 사용	433	DB2 서버에 대한 프로토콜 정보 갱신	478
Windows NT용 DB2로 사용자 인증	434	ATTACH에 대한 노드 별명 카탈로그화	479
사용자 이름 및 그룹 이름 제한사항	434		

DB2 서버 등록 해제	479	MyExtension.java	514
데이터베이스의 등록	479	MySample.java	514
원격 서버에 접속	480	MyDatabaseActions.java	515
데이터베이스 등록 해제	481	MyInstance.java	516
지역 데이터베이스 및 노드 디렉토리에 있는		MyDB2.java	516
LDAP 항목 새로 고침	481	MyDatabases.java	517
검색	483	MySYSPLAN.java	518
호스트 데이터베이스 등록	484	MyTable.java	518
사용자 레벨에서 DB2 레지스트리 변수 설정	486	MyDBUser.java	519
설치 완료 후 LDAP 지원 사용	486	MyToolBarAction.java	520
LDAP 지원 사용 불가능	487	MyAlterAction.java	520
LDAP 지원 및 DB2 Connect	488	MyAction.java	521
보안 고려사항	488	MyDropAction.java	521
Windows 2000에서 활성 디렉토리 보안		MyCascadeAction.java	521
고려사항	489	MyCreateAction.java	522
DB2 오브젝트 클래스 및 속성이 있는 디렉			
토리 스키마 확장	490	부록L. DB2 라이브러리 사용.	523
IBM eNetwork 디렉토리 버전 2.1에 대		DB2 PDF 파일 및 인쇄된 책	523
해 디렉토리 스키마 확장	491	DB2 정보	523
Windows 2000 Active Directory 버전		PDF 책 인쇄	533
2.1에 대해 디렉토리 스키마 확장	491	인쇄된 책 주문	534
Windows 2000 Active Directory에 있는		DB2 온라인 문서	536
DB2 오브젝트	493	온라인 도움말 액세스	536
DB2가 사용하는 오브젝트 클래스 및 속		온라인 정보 보기	538
성	494	DB2 마법사 사용	540
		문서 서버 설정	541
		온라인 정보 검색	542
부록K. 제어 센터 확장.	507		
성능 고려사항	507	부록M. 주의사항.	543
패키징 고려사항	507	상표	546
인터페이스 설명	507	색인	549
CCExtension	508		
CCObject	509	IBM에 문의	563
CCMenuItem	512	제품 정보	563
CCToolBarAction	512		
사용 시나리오	513		

이 책에 관하여

세 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 명령행 처리기. 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 API(Application Programming Interface). API 사용에 관한 자세한 정보는 *Administrative API Reference*를 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 발생한 메시지를 모니터하는 경고 센터.
- 제어 센터, 경고 센터, 복제 관리의 설정값을 변경하는 도구 설정값.
- 작업이 자동으로 수행되도록 스케줄하는 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 조작원을 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 필요한 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, 자동 로드 프로그램, 가져오기 및 내보내기 유틸리티에 대해 설명합니다.

복구

- 제8장 데이터베이스 복구에서는 데이터베이스 및 테이블 공간 복구 메소드 선택 시 고려할 인수를 설명합니다. 복구 타스크에는 데이터베이스 또는 테이블 공간의 백업과 복원 및 롤 포워드 복구 사용이 포함됩니다.

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 파티션으로 명령 발행에서는 파티션된 데이터베이스 환경의 모든 파티션으로 명령을 송신하는 *db2_all*과 *rah* 셸 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 또는 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 노드 간 통신에서는 Windows NT 환경에서 DB2 Enterprise - Extended Edition으로 가상 인터페이스 아키텍처를 사용하는 방법에 대해 설명합니다.
- 부록J. LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스에서는 LDAP 디렉토리 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 마법사, 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.

데이터베이스 설계

- "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 및 실패 지원 소개"에서는 DB2가 지원하는 높은 사용 가능성 실패 지원에 대한 개요를 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 대한 개념 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 키탈로그 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 설명합니다.
- "조정자(Governor) 사용"에서는 조정자를 사용하여 데이터베이스 관리의 일부 양상을 제어하는 방법에 대해 설명합니다.

- "구성 조정"에서는 데이터베이스 시스템의 크기 증가에 연관된 일부 고려사항 및
타스크에 대해 설명합니다.
- "데이터베이스 파티션에 데이터 재분배에서는 파티션"된 데이터베이스 환경에서
파티션에 데이터를 재분배하는 데 필요한 타스크에 대해 설명합니다.
- "벤치마크 테스트"에서는 벤치마크 테스트의 개요 및 벤치마크 테스트 수행 방
법을 보여줍니다.
- "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 설명
합니다.
- "Explain 테이블 및 정의"에서는 DB2 Explain 기능에서 사용되는 테이블 및
해당 테이블의 작성 방법을 설명합니다.
- "SQL Explain 도구"에서는 DB2 explain 도구 즉, db2expln 및 dynexpln의
사용 방법을 설명합니다.
- "db2exfmt -- Explain 테이블 포맷 도구"에서는 DB2 explain 도구를 사용하
여 explain 테이블 데이터를 포맷하는 방법을 설명합니다.

제1부 제어 센터를 사용한 관리

제1장 GUI 도구를 사용한 DB2 관리

DB2 Universal Database에서는 GUI(Graphical User Interface) 도구를 제공하여 사용자 제어 센터라는 중앙 위치에서 지역 및 원격 데이터베이스를 쉽게 관리할 수 있도록 합니다.

이 장에서는 사용 가능한 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 사용자의 Java 기능 도구를 포함시키는 방법에 대해서도 설명합니다.

이 장에는 다음과 같은 내용이 들어 있습니다.

- 4 페이지의 『관리 도구』
- 7 페이지의 『공통 도구 기능』
- 13 페이지의 『제어 센터』
- 23 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』
- 44 페이지의 『원격 데이터베이스 관리』
- 47 페이지의 『사용자 관리』
- 48 페이지의 『데이터 이동』
- 51 페이지의 『저장영역 관리』
- 54 페이지의 『문제점 해결』

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

관리 도구

DB2 관리 도구는 대부분의 DB2 Universal Database 제품에 대해 선택 가능한 구성요소인 관리 클라이언트의 일부입니다. 관리 클라이언트는 DB2를 사용할 수 있는 모든 운영 체제에 대한 관리 클라이언트가 들어 있는 CD-ROM 세트에서도 사용 가능합니다. 이것으로 모든 워크스테이션에 관리 클라이언트를 설치하고 사용할 수 있습니다. 데이터베이스 서버가 지역인지 원격인지 또는 데이터베이스 서버가 수행되고 있는 운영 체제와는 관계가 없습니다. 도구를 사용하면, 명령행 처리기에서 사용할 수 있는 것과 같은 기능을 그래픽 사용자 인터페이스에서도 수행할 수 있습니다. 이러한 기능에는 DB2 명령, SQL문 또는 시스템 명령어 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 사용 가능합니다.

- 제어 센터. 제어 센터는 데이터베이스를 관리하기 위한 주 DB2 그래픽 도구입니다. 제어 센터에서 지역적으로 카탈로그화된 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 체제의 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 실행기록 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 마법사 사용 옵션이 제공됩니다. 마법사는 제어 센터에 있는 팝업 메뉴에서 호출됩니다. 이들은 수행 중인 작업에 필요한 정보를 채우는 방법에 대해 단계별로 프롬프트를 표시하고, 사용자가 제공한 정보를 근거로 계산 및 권장사항을 결정함으로써 보다 높은 레벨의 도움말을 제공합니다. 마법사는 새로운 데이터베이스 관리자 또는 데이터베이스를 가끔 관리하는 사용자인 경우에 매우 유용합니다.

DB2 Universal Database에는 다음 마법사가 있습니다.

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

- 테이블 공간 작성. 이 마법사로 새로운 테이블 공간을 작성하고 기본 저장영역 및 성능 옵션을 설정할 수 있습니다. 마법사를 호출하려면, 테이블 공간 아이콘을 선택하고 마우스 오른쪽 단추를 누른 후, 작성 → 마법사를 사용한 테이블을 선택하십시오.
- 색인 작성. 마법사를 사용하여 제공된 SQL문에 대해 작성하거나 제거할 색인을 결정하십시오. 권장사항은 지정한 워크로드에 따라 달라집니다. 색인 작성 마법사를 호출하려면, 색인 폴더를 선택하고, 마우스 오른쪽 단추를 누른 후, 작성 → 마법사를 사용한 색인을 선택하십시오.
- 성능 구성. 이 마법사는 데이터베이스, 데이터 및 시스템 목적에 관한 정보를 요청하여 데이터베이스를 조정하는 데 도움이 됩니다. 그런 다음, 데이터베이스 및 인스턴스에 대한 새로운 구성 매개변수를 작성하고, 원하면 이를 자동으로 적용하도록 합니다. 이 마법사를 호출하려면, 데이터베이스의 아이콘을 선택하고, 마우스 오른쪽 단추를 누른 후 마법사를 사용하여 구성을 선택하십시오.
- 데이터베이스 복원. 이 마법사는 데이터베이스 복구 프로세스를 진행합니다. 마법사를 호출하려면, 데이터베이스의 아이콘을 선택하고, 마우스 오른쪽 단추를 누른 후 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.
- 다중 사이트 갱신 구성. 이 마법사를 사용하면 응용프로그램이 여러 사이트를 동시에 갱신할 수 있도록 데이터베이스를 구성할 수 있습니다. 데이터가 모든 사이트에서 일정해야 하는 경우에 이렇게 하는 것이 중요합니다. 이 마법사를 호출하려면, 인스턴스를 선택하고 마우스 오른쪽 단추를 누른 후, 다중 사이트 갱신 → 마법사를 사용한 구성을 선택하십시오.

주: OS/390용 DB2 서브시스템에 대한 마법사는 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 모니터하는 도구입니다. 이 도구를 사용하여 성능 문제점을 검출하고 최적의 성능을 위해 데이터베이스를 조정합니다. 성능 모니터는 제어 센터에 있는 팝업 메뉴의 선택으로서 호출됩니다.

- 이벤트 모니터. 이벤트 모니터는 특정 이벤트가 발생한 시간에 데이터베이스의 상태를 레코딩하여 자원 사용법을 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. Visual Explain 함수를 사용하면 보다 나은 성능을 위해 SQL 조회를 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 항목 표시
- DDL 생성
- 필터
- 도움말

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 누르면 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 명령문 또는 명령을 스크립트로서 저장합니다. 이 성능으로 같은 명령문 또는 명령을 다시 수행하려는 경우, SQL문 또는 DB2 명령을 재입력하지 않아도 됩니다. 일단 SQL문 또는 DB2 명령이 스크립트에 저장되면, 스크립트를 스케줄하거나, 스크립트를 변경하거나, 명령문 또는 명령을 재입력하지 않고 비슷한 스크립트를 작성할 수 있습니다.

SQL문 또는 DB2 명령을 표시하려면, 다음을 수행하십시오.

1. 제어 센터에서 SQL문 또는 DB2 명령을 생성하는 도구에 대해 작업하는 창 또는 노트북으로 이동하십시오. **SQL 표시** 누름 버튼 또는 **명령 표시** 누름 버튼이 사용 가능으로 표시됩니다.
2. **SQL 표시** 누름 버튼 또는 **명령 표시** 누름 버튼을 누르십시오. 해당 창이 열립니다.

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

명령 표시 또는 **SQL 표시** 누름 버튼을 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 누르면, 새 명령 스크립트 창이 나타납니다. 스크립트를 저장하기 전에 SQL문 또는 DB2 명령을 편집할 수 있습니다.

관련 항목 표시

관련 항목 표시에서는 테이블, 색인, 뷰, 별명, 트리거, 테이블 공간, 사용자 정의 함수(UDF) 및 사용자 정의 유형의 관계를 보여줍니다. 예를 들어, 테이블을 선택

하고 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 판별
- 테이블 공간에 저장된 오브젝트 판별
- 한 오브젝트와 관련된 수행하려는 조치에 의해 영향을 받는 다른 오브젝트 알기. 예를 들어, 종속 뷰가 있는 테이블을 제거하려는 경우, 관련 항목 표시에서 는 동작하지 않게 될 뷰를 보여줍니다.

관련 항목 표시 기능을 사용하려면, 다음을 수행하십시오.

- 제어 센터의 내용 분할창에서 오브젝트를 선택한 후 마우스 오른쪽 단추를 누르십시오.
 - 관련 항목 표시를 선택하십시오.
 - 탭을 눌러 원하는 관련 오브젝트에 대한 페이지를 여십시오. 선택한 탭에 따라 다른 관련 오브젝트가 나열됩니다. 선택한 오브젝트와 직접 관련된 오브젝트만이 표시됩니다.
- 선택된 페이지의 관련 오브젝트에서 마우스 오른쪽 단추를 누른 후, 팝업 메뉴에서 관련 항목 표시를 선택할 수 있습니다. 선택된 페이지는 최근 선택과 관련된 오브젝트를 표시하도록 변경됩니다. 또한, 선택된 오브젝트 옆에 있는 아래쪽 화살표를 눌러 관계를 표시하기 위해 사용자가 이전에 선택한 오브젝트 목록을 표시할 수 있습니다.
- 달기를 눌러 관련 항목 표시 노트북을 닫은 후 제어 센터로 리턴하십시오.

DDL 생성

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

- 데이터베이스 오브젝트
- 권한 부여 명령문
- 테이블 공간, 노드 그룹 및 버퍼 풀

- 데이터베이스 통계

이것으로 다음을 수행할 수 있습니다.

- DDL을 저장하여 동일하게 정의된 테이블, 데이터베이스 및 또다른 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 누르면, **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 누르면, 스크립트 수행 창이 열립니다.

주: DDL문을 생성하는 것은 시스템 390의 제어 센터와 작업할 때와는 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.

1. DDL문을 생성하려는 오브젝트를 강조표시한 후 마우스 오른쪽 단추를 누르십시오.
2. **DDL** 생성을 선택하십시오. 스크립트 수행 창이 나타납니다.
3. 사용자 ID와 암호를 입력한 후 확인을 누르십시오. **db2look** 명령의 내용이 들어 있는 작업이 작성됩니다. DB2 메시지 창이 새 작업의 작업 ID와 함께 나타납니다.
4. 확인을 눌러 메시지 창을 닫으십시오.
5. 저널 노트북의 작업 실행기록 페이지를 사용하여 작업 결과를 본 후 해당 작업과 연관되어 저장된 스크립트의 내용을 보십시오.

6. 작업을 선택한 후 마우스 오른쪽 단추를 누르십시오. 팝업 메뉴에서 결과 표시를 선택하십시오. 작업 결과 창이 열립니다. **db2look** 명령의 결과물은 작업 출력 분할창에 표시됩니다.
7. 스크립트 작성을 선택하여 결과물의 스크립트를 작성하십시오. 새 명령 스크립트 창이 나타납니다.
8. 이를 다시 사용하려는 경우 새로운 스크립트를 저장하십시오.

필터

제어 센터에서 사용자는 내용 분할창에 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 지우거나 삭제해야 합니다.

표시장치 필터링

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

1. 제어 센터 맨 아래에 있는 내용 분할창 도구 모음에서 필터 아이콘을 선택하거나 뷰 메뉴 표시줄에서 필터를 선택하십시오.
2. 오브젝트 수를 줄이기 위해 사용되는 기준을 선택하십시오.
3. 필터 사용 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 분할창에 표시되는 결과물이나 결과 세트를 정의할 수 있습니다.

1. 트리에서 폴더 오브젝트를 선택한 후 마우스 오른쪽 단추를 누르십시오.
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 마우스 오른쪽 단추를 누르십시오.
3. 팝업 메뉴에서 필터를 선택하십시오. 필터 노트북이 열립니다.
4. 찾기 페이지에서, 선택된 오브젝트의 이름 또는 다른 서술적 필터 기준을 지정하십시오. 필터 결과는 제어 센터의 내용 분할창에 표시되는 선택된 오브젝트와 연관된 결과 세트입니다.
5. 찾기 페이지에서, 단일선택 버튼을 선택하여 찾기 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 눌러 정의된 필터 기준을 사용하십시오.

행 수를 기준으로 이 필터 노트북을 자동으로 호출하려면, 메뉴 표시줄에서 도구를 선택한 후, 팝업 메뉴에서 도구 설정을 선택하십시오. 행 수를 초과할 때의 필터링 선택 선택란을 사용하면 선택에서 리턴된 행의 임계값을 사전 정의할 수 있습니다. 임계값에 도달하면, 정의된 기준을 기본으로 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 모음에 뿐만 아니라 모든 창과 노트북에도 있습니다. 필드를 채우고 타스크를 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 카탈로그화되어야 합니다. 그림1에서는 제어 센터의 1차 기능을 보여줍니다 운영 체제의 차이로 인해, 시스템에 있는 제어 센터가 그림과 다르게 표시될 수 있습니다.

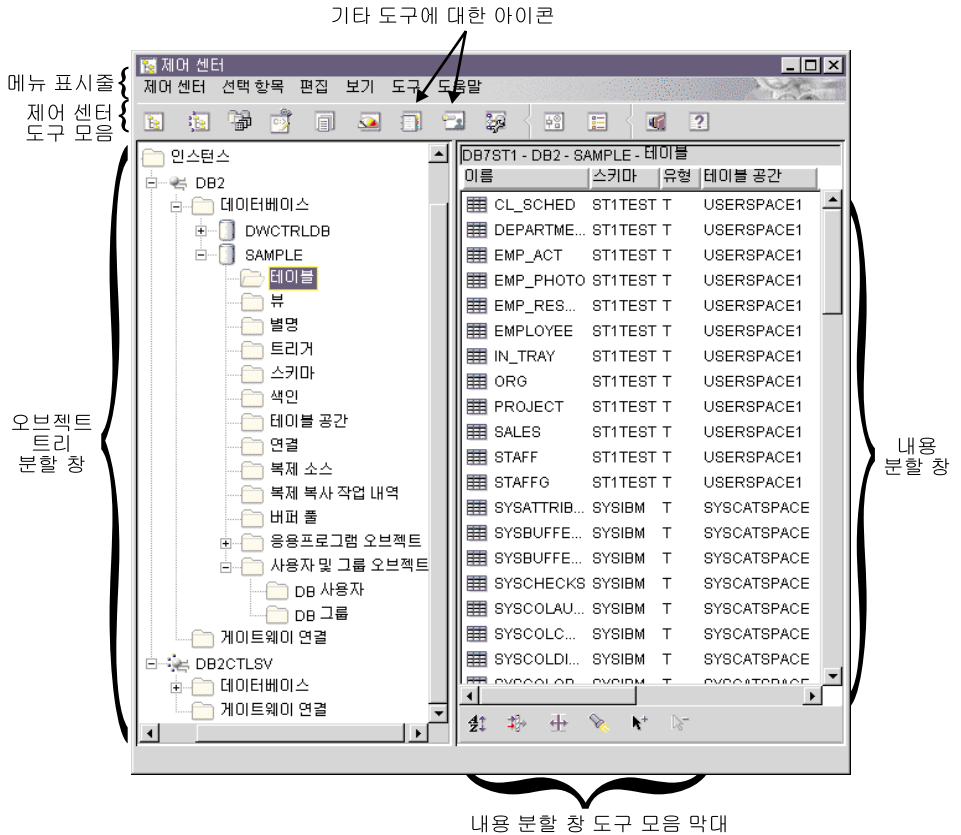


그림1. 제어 센터 기능

제어 센터의 주 요소

제어 센터의 주 요소는 다음과 같습니다.

- 메뉴 표시줄. 메뉴 표시줄은 화면의 맨 위에 있습니다. 메뉴 표시줄에서 메뉴를 선택하면 DB2 도구 종료, 그래픽 도구 액세스, 온라인 도움말 및 제품 정보에 대한 액세스 확보 같은 많은 기능을 수행할 수 있습니다. 메뉴 표시줄에 있는 각 항목을 눌러 이러한 기능에 익숙해져야 합니다.

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

제어 센터 도구 모음에서 도구 설정 아이콘을 선택하여 이러한 도구에 대한 설정을 변경할 수 있습니다.

- 오브젝트 트리. 오브젝트 트리는 화면의 왼쪽 분할창에 있습니다. 여기에서는 제어 센터에서 관리할 수 있는 모든 데이터베이스 서버와 오브젝트에 대한 아이콘을 표시합니다. 먼저 오브젝트 트리 분할창에 나타나기 전에 원격 데이터베이스 서버를 카탈로그화해야 합니다. 오브젝트 트리 분할창에 있는 일부 오브젝트에는 기타 오브젝트가 들어 있습니다. 오브젝트의 왼쪽에 있는 더하기 부호(+)는 오브젝트가 축소되었음을 나타냅니다. 더하기 부호를 눌러 이를 확장할 수 있습니다. 빼기 부호(-)는 확장될 때 오브젝트의 왼쪽에 나타납니다. 오브젝트를 축소하려면, 빼기 부호를 누르십시오.

- 내용 분할창. 내용 분할창은 화면의 오른쪽 분할창에 있습니다. 이 분할창에서는 오브젝트 트리 분할창에서 선택된 오브젝트에 들어 있는 모든 오브젝트를 표시합니다. 예를 들어, 오브젝트 트리 분할창에서 테이블 폴더를 선택하면, 모든 테이블이 내용 분할창에 표시됩니다. 데이터베이스 폴더를 선택하면, 내용 분할창이 모든 데이터베이스를 표시하기 위해 변경됩니다. 내용 분할창 도구 모음에 있는 필터 아이콘을 누른 후 필요한 정보를 지정하여 내용 분할창에 나타나는 컬럼을 필터링하거나, 도구 모음에서 도구를 선택하고 도구 설정을 선택하여 오브젝트를 필터링할 수 있습니다. 필터 사용 선택란이 내용 분할창 필터 대화 상자에서 선택되어 있는지 확인해야 합니다.

- 내용 분할창 도구 모음. 이 도구 모음은 내용 분할창의 맨 아래에 나타납니다. 이것으로 내용 분할창에 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 경계로 강조표시됩니다. 이 경계는 사용자가 입력해야 하는 필수 필드임을 나타냅니다. 값을 선택하거나 값을 입력하고 나면, 빨간색 경계가 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 데이터베이스 사용자와 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 바꾸지는 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 UNIX 플랫폼용 DB2 Universal Database 제품군에 대한 데이터베이스 오브젝트를 관리할 수 있습니다. 빠른 시작 책에서 플랫폼 고유의 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 정보를 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(UDF)
- 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
- 동의어
- 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(UDF)
- 구별 유형

이러한 오브젝트 각각에 대해 수행할 수 있는 조치를 보려면, 오브젝트 분할창에서 오브젝트를 선택한 후 마우스 오른쪽 단추를 누르십시오. 팝업 창이 함수를 나열하며 나타납니다.

제어 센터에서 시스템 표시

시스템에 카탈로그화된 모든 시스템과 설치된 DB2를 표시하려면, 다음을 수행하십시오.

1. 시스템 옆에 있는 더하기 부호(+)를 눌러 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 마우스 오른쪽 단추를 누르면, 팝업 메뉴에 있는 옵션 중 하나에 관리 서버에 접속이 호출됩니다. 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 시

작됩니다. 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, 이것은 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 제한할 수 있습니다. 또한, 특별한 환경에 맞도록 인스턴스를 조정할 수도 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 제거와 같은 기능을 수행할 수 있습니다. 그러나 제어 센터에서 OS/390용 DB2 서브시스템을 관리하기 전에, 먼저 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 목적지 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 마우스 오른쪽 단추를 누르십시오. 팝업 메뉴가 나타나 해당 오브젝트에 대해 수행할 수 있는 사용 가능한 조치를 보여줍니다. 예를 들어, 내용 보기, 이에 대한 특권 수정 및 이와 관련된 기타 오브젝트 목록 표시 뿐만 아니라 뷰를 작성, 변경 또는 제거할 수 있습니다. 수행할 수 있는 기타 기능에 대한 자세한 정보는 OS/390용 DB2 오브젝트에 대한 온라인 도움말을 참조하십시오.

게이트웨이 연결 관리

DB2 Connect 서버가 카탈로그화되면, 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 데이터베이스 간의 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 및 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.
- 데이터베이스 또는 테이블 공간을 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 연결 관리. 응용프로그램 관리
- SQL Explain을 사용하여 조회를 분석하고 액세스 플랜 참조
- 제어 센터를 통해 메뉴 및 텍스트를 표시하는 데 사용되는 글꼴 변경. 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 분할창 또는 내용 분할창에서 오브젝트를 선택한 후 마우스 오른쪽 단추를 누르십시오. 팝업 메뉴가 나타나고 해당 유형의 오브젝트에 대해 수행할 수 있는 모든 기

능을 표시합니다. 예를 들어, 테이블 폴더를 선택하면, 마법사의 도움말과는 관계없이 새로운 테이블을 작성하고, 테이블 성능을 모니터링하며, 내용 분할창에 표시되는 테이블을 필터링할 수 있습니다. 수행할 수 있는 기능은 선택한 오브젝트에 따라 다릅니다.

내용 분할창에 있는 오브젝트에서 마우스 오른쪽 단추를 눌러 특정 오브젝트에 대한 추가 기능을 수행하십시오. 예를 들어, 내용 분할창에서 테이블 중 하나를 선택하여 마우스 오른쪽 단추를 누르면, 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 마우스 오른쪽 단추를 누르십시오. 예를 들어, 테이블 아이콘을 누르십시오. 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 갖게 됩니다. 하나는 마법사를 사용하는 테이블을 작성하는 것입니다. 마법사 옵션을 선택하면, 정보를 입력하도록 프롬프트가 표시되고 작성해야 하는 선택 사항에 대한 제안사항이 제공됩니다. 마법사는 특히 데이터베이스 오브젝트를 자주 작성하지 않는 사용자 또는 새로운 사용자에게 특히 유용합니다.

기존 오브젝트에 대한 작업

오브젝트 트리 분할창에서 테이블 폴더와 같은 오브젝트를 누르면, 기존의 모든 테이블이 내용 분할창에 나타납니다. 그런 다음, 작업하려는 테이블을 선택하고, 마우스 오른쪽 단추를 눌러 해당 특정 테이블에 대해 수행하려는 기능을 호출할 수 있습니다.

제어 센터 사용에 대한 자세한 정보는, 도움말 메뉴로 이동하거나 제어 센터의 어디에서든 F1을 누르면 온라인 도움말을 얻을 수 있습니다.

오브젝트 찾기(OS/390용 DB2 전용)

찾기 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.

찾기 노트북의 찾기 페이지를 사용하여 검색 기준을 지정하십시오. 찾기 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, 마우스 오른쪽 단추로 오브젝트를 누르십시오. 팝업 메뉴에서 찾기를 선택하십시오. 찾기 노트북이 열립니다.
2. 오브젝트 유형 필드에서, 검색할 데이터베이스 오브젝트의 유형을 선택하십시오. 사용 가능한 목표 오브젝트 목록은 검색을 시작하는 오브젝트에 따라 다릅니다.
3. 찾기 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. 찾기 페이지에서 단일선택 버튼을 선택하여 찾기 페이지에 있는 필드에서 선택된 모든 조건에 부합하게 할지 또는 적어도 하나의 조건에 부합하게 할지 여부를 지정하십시오.
5. 확인을 눌러 검색 기준을 사용하십시오. 검색 결과는 결과 찾기 창에 표시됩니다. 출력 테이블의 형식은 검색한 오브젝트 유형에 따라 달라집니다.
6. 같은 기준 또는 다른 기준으로 검색을 반복하려면, 적용을 누르십시오.
7. 결과 찾기 창에 나타나는 행을 선택하고, 해당 행을 오른쪽 버튼을 눌러 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 콜렉션을 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 학습하지 않아도 됨을 의미합니다.

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

DB2 서버를 함께 그룹화함으로써, DB2 서버를 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 인증 증명서를 작성할 수 있습니다. 또한 성공 코드 세트를 정의하고, 위성 환경의 관리와 연관된 기타 기능을 수행할 수도 있습니다. 위성 환경에 대한 정보는 위성 제어 데이터베이스라고 하는 중앙 데이터베이스에 저장됩니다. 이 데이터베이스는 환경에 어떤 위성이 있는지, 각 위성이 어느 그룹에 속하는지와 위성이 수행 중인 일반 사용자 응용프로그램이 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, 일괄처

리 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 눌러 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 화면이동시키고 보고서를 생성할 수 있습니다.
- 스크립트 센터에 명령 스크립트를 작성한 후 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 한번에 발행할 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 로그인되도록 하는 장점이 있습니다.

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

1. 스크립트 → 새로 작성을 선택하십시오 새 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 누르십시오.

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 자동 수행되도록 작업을 스케줄할 수 있습니다. 작업은 스크립트를 스케줄하거나 스크립트를 수행할 때 마다 즉시 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 누르십시오. 스크립트 센터가 열립니다.
2. 스크립트 → 가져오기를 선택하십시오. 파일 찾아보기 창이 열립니다.

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

저장된 명령 스크립트를 수행하도록 스케줄

스크립트를 스케줄하려면, 다음을 수행하십시오.

1. 제어 센터 도구 모음에 있는 스크립트 센터 아이콘을 누르십시오. 스크립트 센터가 열립니다.
2. 실행하도록 스케줄하려는 스크립트를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 스케줄을 선택하십시오. 스케줄러 창이 열립니다.
3. 작업 빈도와, 완료 메시지 또는 시작할 또다른 명령 스크립트 같은 완료 조치를 선택하십시오.
4. 확인을 누르십시오. 이것으로 저널에서 추적할 수 있는 보류 중인 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. 저널에서는 다음을 수행할 수 있습니다.

- 보류 중인 작업, 수행 중인 작업 및 작업 실행기록 모니터링
- 결과 검토
- 복구 실행기록 및 경보 메시지 표시
- DB2 메시지 로그 표시

작업에 대한 작업

저널을 사용하여 작업을 수행하십시오. 저널을 열려면, 다음을 수행하십시오.

1. 스크립트 센터 도구 모음에서 저널 아이콘을 누르십시오. 저널이 열립니다.

2. 나중에 수행되도록 스케줄된 작업을 보려면, 보류 중인 작업 누름 버튼을 누르십시오. 보류 중인 작업 목록에서 작업을 보게 됩니다. 또한, 작업에 대한 모든 정보도 볼 수 있습니다. 보류 중인 작업에 대해 재스케줄, 작업과 연관된 스크립트 표시 또는 즉시 수행 등과 같은 조치를 수행할 수 있습니다. 저장된 스크립트가 수정되면, 이에 종속되는 모든 작업은 새로 수정된 작동을 상속합니다.

저널에서, 현재 수행 중인 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 사용 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
- 만기 날짜
- 등록된 사용자

- 동시 사용자
- 권한이 부여된 사용자 수
- 동시 사용자 수
- 강제 규정
- 프로세서의 수(DB2 Universal Database Enterprise Edition 및 Enterprise - Extended Edition의 경우)

경보 센터

경보 센터는 시스템을 모니터링하여 잠재적인 문제점을 알려주는 도구입니다. 임계값을 초과하여 알람 또는 경고 상태에 있는 모니터링 오브젝트를 표시하기 위해 경보 센터가 자동으로 열리도록 설정할 수 있습니다. 성능 모니터를 사용하여 임계값을 설정하며, 이것은 제어 센터에서 호출됩니다. 아이콘 색상은 경고의 심각도를 나타냅니다. 빨간 아이콘은 알람을 나타냅니다. 노란 아이콘은 경고를 나타냅니다.

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 복잡도로부터 사용자를 보호하는 동안 노드 및 데이터베이스를 카탈로그화합니다.

클라이언트 구성 지원 프로그램에서 다음과 같은 타스크를 수행할 수 있습니다.

- 데이터베이스 연결 항목을 추가, 수정 및 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 및 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. 프로파일을 이전에 구성된 머신에서 내보내어, 새로운 머신을 구성하기 위해 사용할 수 있습니다. 서버 프로파일을 제어 센터에서 내보낼 수 있으며, 클라이언트 또는 서버 프로파일은 CCA에서 내보낼 수 있습니다.
- 네트워크를 검색하십시오. CCA는 수행 중인 관리 서버를 갖는 DB2 시스템의 네트워크를 검색할 수 있습니다. 검색 및 알려진(또는 방향지정) 발견 모드가 제공됩니다. 각 발견 모드는 네트워크 구성 제한사항에 따라 다릅니다. 보통 네트워크 라우터는 검색 발견 기능(discovery) 요청이 전송되도록 허용하지 않습니다. 알려진 발견 기능(discovery)은 원하는 서버 시스템을 찾기 위해 몇 가지 정보만을 요구합니다. 이전에 게이트웨이에 정의된 AS/400 시스템 또는 호스트도 찾을 수 있습니다.
- 데이터베이스 연결을 수동으로 구성하십시오. 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 사용자 정의될 수 있는 그래픽 유틸리티입니다. 성능 모니터가 수집한 값이 허용 범위를 벗어나면 트리거가 경고하거나 알람을 보내는 임계값 또는 구역을 정의할 수 있습니다.

오브젝트 트리 분할창 또는 내용 분할창에서 오브젝트를 선택하고 마우스 오른쪽 단추를 눌러 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 녹색, 노란색 또는 빨간색으로 표시되어 모니터 상태를 나타냅니다. 색상은 사용자가 설정한 임계값으로 정의된 문제점의 심각도를 나타냅니다. 녹색은 모니터가 수행 중이며 모든 것이 양호함을 나타냅니다. 노란색은 경고이고, 모니터 중인 오브젝트가 임계값에 근접하고 있음을 나타냅니다. 빨간색은 알람이고, 모니터 중인 오브젝트가 임계값에 도달했음을 나타냅니다. DB2와 함께 제공되는 사전 정의된 모니터를 사용하거나, 자체 모니터를 작성할 수 있습니다.

성능 모니터가 수집한 정보를 보려면, 오브젝트를 마우스 오른쪽 단추로 누른 후 팝업 창에서 **모니터 활동 표시**를 선택하십시오.

성능 모니터에 있는 정보를 사용하여 다음을 수행하십시오.

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

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

기존 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 단계는 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 알람 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 데이터베이스 이벤트(예: 데이터베이스 연결 또는 SQL문)에 대한 활동 요약을 수집합니다. 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(예: 사용된 CPU 및 SQL문의 양)을 알아야 하는 경우 이벤트 모니터를 사용하십시오. 그런 다음, 이벤트 모니터에 기록된 데이터를 읽기 위해 이벤트 분석기를 사용하십시오.

각 데이터베이스 연결의 경우, 하나의 연결 이벤트 레코드가 생성됩니다. 해당 연결에서 수행되는 각 명령문의 경우, 명령문 레코드가 생성됩니다. 각 연결 이벤트 레코드는 이벤트 분석기의 연결 뷰 창에 있는 하나의 행으로 맵핑됩니다. 이 창에서는 모니터되는 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 맵핑됩니다.

모니터 도구 사용

성능 모니터 및 이벤트 분석기를 사용하면 다음과 같은 이점이 제공됩니다.

- 광범위하고, 유연성 있는 데이터 콜렉션. 200개가 넘는 성능 변수가 지원되며, 여기에는 버퍼 풀 및 입출력, 잠금 및 교착 상태, 정렬, 통신, 에이전트 및 로깅 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 경보 성능. 성능 측정을 위해, 임계값을 지정하여 예외 조건을 정의할 수 있습니다. 임계값은 성능 그래프에 있는 특정 구역에서 측정치 점을 찍어 성능 측정치가 임계값에 도달하거나 초과했음을 시각적으로 표시하는 데 사용됩니다. 임계값에 도달하면, 다음 조치 중 일부 또는 모두가 발생하도록 지정할 수 있습니다.
 - 경보 센터를 통해 통지됩니다.
 - 들을 수 있는 알람을 받습니다.
 - 프로그램이 수행됩니다.
 - 메시지가 표시됩니다.

또는, 아무런 통지를 받지 않도록 결정할 수도 있습니다.

33 페이지의 그림2에서는 모니터가 함께 작동하는 방법에 대해 설명합니다.

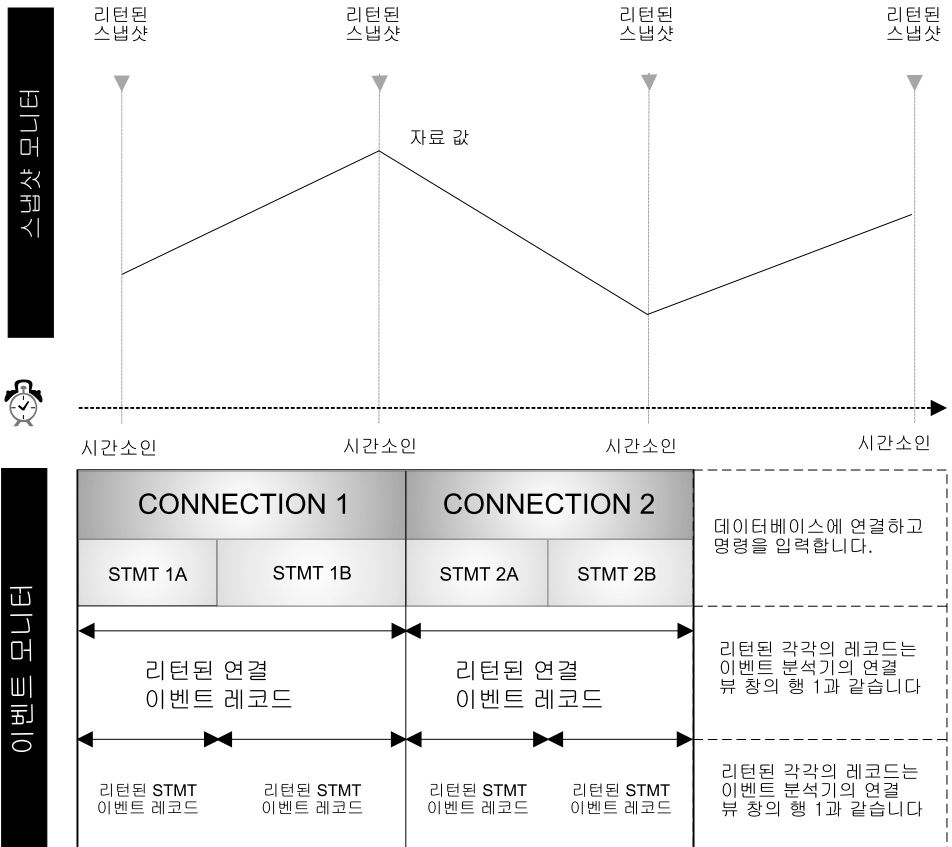


그림 2. 비교: 스냅샷 확보 및 이벤트 모니터. (이벤트 모니터, 이벤트 분석기)

데이터베이스 모니터 및 조정을 위한 고려사항

데이터베이스를 모니터하고 조정하기 전에, 다음을 수행해야 합니다.

- 목표를 정의하십시오. 예를 들어, 데이터베이스 동시성이 특수 응용프로그램이 시작될 때 줄어드는지 검사할 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 알려고 할 수도 있습니다.
- 분석할 정보를 판별하십시오. 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 다음 경우를 모니터링합니다.

- 너무 많은 데이터베이스 태스크가 최대 활동 시간에서 스케줄되는지의 여부
 - 사용자 연결 수가 많은지 여부
 - 데이터베이스 파티션(하드웨어 로그 밸런스)이 제대로 최적화되지 않았는지 여부
 - 서버가 둘 이상의 데이터베이스 서버에 대해 사용되고 있는지 여부
- 일부 눈에 띄는 효과가 있습니다. 예를 들어, 다음과 같습니다.
- 조회/응답 시간이 늦습니다.
 - 스케줄된 태스크가 제때 완료되지 않습니다.
 - 응용프로그램이 시간종료되었습니다.
- DB2와 함께 사용할 수 있는 사전 정의된 모니터를 사용할 것인지 또는 자체 모니터를 작성할 것인지 여부를 판별하십시오.

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

특정 시점에서 성능 모니터

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

다음 유형의 정보가 캡처됩니다.

- 수명이 긴 활동에 대한 정보(예: 응용프로그램을 완료하는 데 너무 오랜 시간이 소요될 때 데이터베이스 활동)
- 활동의 현재 레벨에 대한 정보를 추적하는 카운터(예: 데이터베이스에 대한 커서 열기의 수)
- 데이터베이스 활동에 대한 누적 정보(예: 데이터베이스 인스턴스가 활동 중인 동안 작성된 최대 연결 수 또는 특정 데이터베이스에 대해 실행된 SQL문의 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 검출
- 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

각각의 경우, 다양한 성능 변수를 모니터링할 수 있습니다. 스냅샷 모니터 창의 도움말 메뉴에서 사용할 수 있는 성능 변수 참조 도움말에서는 모든 성능 변수에 대해 설명합니다. 이러한 변수는 범주로 구성됩니다. 다음과 같은 범주가 있습니다.

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

- 데이터베이스 연결: 버퍼 풀 및 입출력, 잠금 및 교착 상태, 정렬, SQL문 활동 스냅샷 생성 방법에 대한 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, 등식 또는 텍스트 설명을 변경할 수 없지만, 임계값과 경보 조치는 변경할 수 있습니다. 사전 정의된 모니터를 사용하여 성능 모니터링에 대해 학습하고 사전 정의된 모니터를 복사하고 사본에서 성능 변수를 추가하거나 제거하여 사용자 고유의 모니터를 작성할 수 있습니다.

DB2와 함께 제공되는 사전 정의된 모니터는 다음과 같습니다.

- 모니터링 용량. 이 모니터를 사용하여 시스템 용량에 대한 정보를 확보하십시오. 이 모니터는 정기적으로 검사되어 시스템의 전체적인 사용을 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 판별하십시오.
- 캐쉬. 이 모니터를 사용하여 캐쉬 사용을 최적화하십시오. 최대 기간 동안 이들 값을 모니터링함으로써, 캐쉬 크기를 늘려야 하는지 여부를 판별할 수 있습니다.
- 버퍼 풀. 소형 테이블에서 이 모니터를 사용하여 자체 버퍼 풀이 필요한지 여부를 판별하십시오.
- 교착 상태. 이 모니터를 사용하여 응용프로그램이 교착 상태가 되는지 여부를 판별하십시오.
- 고속 통신 관리 프로그램. 이 모니터를 사용하여 노드 사이에서 정보를 전송하는 데 사용되는 메모리 비율을 알 수 있습니다.

- 프리페처. 이 모니터를 사용하여 시스템에 충분한 프리페처가 정의되었는지 여부를 판별하십시오.
- 디스크 성능. 이 모니터를 사용하여 입력 및 출력을 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 사용 방법에 대한 예는 성능 모니터링을 위해 제공된 온라인 도움말을 참조하십시오.

사용 가능한 모니터 목록을 보려면, 제어 센터에서 마우스 오른쪽 단추로 시스템 폴더를 누른 후, 팝업 메뉴에서 모니터 목록을 선택하십시오. 모니터 목록 창이 열립니다. 현재 연결된 JDBC 서버에 저장된 모니터를 나열합니다. 각 모니터의 경우, 모니터 이름, 설명, 상태, 기본 모니터가 있는지 그리고 모니터 작성자를 알 수 있습니다. 『모니터 상태』는 지역 시스템에서 모니터의 상태를 나타내지만, JDBC 서버에서는 나타내지 않습니다. 레벨에 대한 『기본값』은 인스턴스, 데이터베이스, 테이블, 테이블 공간 또는 연결 레벨에서 기본 모니터를 나타냅니다. 사전 정의된 모니터의 경우, 『작성자』 컬럼에는 NULLID가 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 정보를 참조하십시오.

오브젝트에 대한 기본 모니터로서 시작되는 모니터를 선택할 수 있습니다.

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

모니터 중인 오브젝트를 면밀히 관찰하려는 경우, 경보 센터를 열거나 요약 페이지에서 모니터 표시 창을 열어 빨간색 또는 노랑색 항목을 찾을 수 있습니다. 또

한, 새로운 경고 또는 알람이 추가된 경우 경고 센터가 자동으로 열릴 수 있도록 제어 센터 설정을 수정할 수도 있습니다. 경고 센터에서, 계속 모니터링하는 동안 경보를 임시로 일시중단시킬 수도 있습니다.

경보 센터에 오브젝트가 나타날 때 필요한 조치

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

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

기간 동안 이벤트 분석

이벤트 분석기는 또하나의 DB2 성능 분석 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 로그된 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 모니터링하십시오.
- 이벤트 모니터를 활성화하여 이벤트 데이터 수집을 시작하십시오. 데이터는 파일에 저장됩니다.
- 이벤트 데이터를 수집하지 않도록 이벤트 모니터를 중지시키십시오.
- 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(UOW))
- SQL문 실행
- 교착 상태 활동 검출

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 실행 파일을 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 모니터를 누른 후 메뉴에서 작성을 선택하십시오. 이벤트 모니터 작성 창이 열립니다.
3. 필드에서, 작성하는 이벤트 모니터에 대한 이름을 지정하십시오. 이 새로운 이벤트 모니터의 이름은 기존 모니터와 같아서 안됩니다. 이름에 공백은 허용되지 않습니다.

4. DB2 Universal Database Enterprise - Extended Edition 제품 전용에서, 이벤트 모니터 파일이 있는 노드를 『노드』 드롭 다운 목록에서 선택하십시오.
5. DB2 Universal Database 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 시작』이 기본 선택임을 주의하십시오.
8. 이러한 레벨에서 모니터링을 제어할 연결, 명령문 또는 트랜잭션에 대해 하나 이상의 조건을 정의하십시오.
9. 모니터가 이벤트 데이터 파일을 작성할 경로(디렉토리 이름)를 식별하십시오.
10. 옵션을 눌러 이벤트 모니터 파일 지정 옵션 창을 여십시오. 이러한 옵션은 모니터 출력이 처리되는 방법을 판별하므로, 이벤트 모니터의 성능에 영향을 줄 수 있습니다.
11. 확인을 눌러 모니터를 작성하거나, 취소를 눌러 모니터를 작성하지 않고 나가십시오.
12. 이벤트 모니터를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 이벤트 모니터링 중지를 선택하여 이벤트 모니터링을 중지시키십시오.

이렇게 하면 이벤트 모니터가 추적 파일을 작성하도록 강제됩니다. 모니터가 꺼지지 않으면, 버퍼가 가득 차거나 모든 연결이 종료될 때에만 정보가 디스크로 작성됩니다. 이벤트 모니터 창에서, 작성한 이벤트 모니터를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 이벤트 모니터 파일 보기를 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 다음, 해당 디렉토리를 지정하십시오. ...을 눌러 기존 디렉토리를 나열할 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. 확인을 눌러 디렉토리에 들어 있는 데이터 파일에 액세스하거나, 취소를 눌러 나가십시오. 모니터 기간 보기 창이 열립니다.
4. 모니터 기간을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 열기 → 연결을 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 첫 번째에 없을 수도 있습니다.)
5. 연결을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 열기 → 명령문을 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 Explain된 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 절에서, *Visual Explain* 함수 용어가 사용됩니다.

Visual Explain 함수를 사용하여 다음을 수행할 수 있습니다.

- 최적화시 사용되는 통계 보기. 그런 다음, 이러한 통계를 현재 카탈로그 통계와 비교하여 패키지 리바인드가 성능을 향상시킬 수 있을지 여부를 판별하는 데 도움을 줄 수 있습니다.
- 테이블 액세스에 색인이 사용되는지 여부 판별. 색인이 사용되고 있지 않으면, Visual Explain 함수는 색인화하여 이득을 얻는 컬럼을 판별하는 데 도움이 됩니다.
- 하나의 조회에 대한 액세스 플랜 그래프의 이전 및 이후 버전을 비교하여 다양한 조정 기법의 수행 효과 보기
- 액세스 플랜에서 각 조작에 관한 정보(총 추정 비용 및 수신된 행 수 포함) 얻기

조회 성능 향상

Visual Explain 함수를 사용하여 SQL문을 분석하고 조정을 돕습니다. Explain된 SQL문에 대한 액세스 플랜의 그래픽 뷰를 나타냅니다. 테이블과 색인 그리고 이에 대한 각각의 조작은 노드로서 표현되고, 데이터 플로우는 노드 간의 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 이해: 사용 방법, 조인 순서, 정렬 발생 및 정렬 유형
- SQL문의 성능을 향상시키는 방법 판별(예를 들어, 새로운 색인을 작성하여)
- 최적화시 사용되는 통계 보기. 그런 다음, 이러한 통계를 현재 카탈로그 통계와 비교하면 패키지 리바인드가 성능을 향상시키는지 여부를 판별하는 데 도움이 될 수 있습니다. 또한, 수집된 통계가 성능을 향상시킬 수 있는지 여부를 판별하는 데에도 도움이 됩니다.
- 테이블 액세스에 색인이 사용되는지 여부 판별. 색인이 사용되지 않는 경우, Visual Explain 함수는 조회 성능을 향상시키기 위해 색인에 포함될 수 있는 컬럼을 판별하는 데 도움이 될 수 있습니다.
- 하나의 조회에 대한 액세스 플랜 그래프의 이전 및 이후 버전을 비교하여 성능을 향상시키기 위한 다양한 조정 기법의 수행 효과 보기
- 액세스 플랜에서 각 조각에 대한 정보 확보, 총 계산된 비용과 검색된 행 수 포함

Explain된 SQL문의 액세스 플랜을 이해하기 위해 Visual Explain 기능을 사용한 후에, 색인이 해당 조회의 성능을 향상시킬 수 있는지 판별할 수 있습니다. 조회의 권장 색인을 받으려면 색인 마법사를 사용하십시오. 또는 RECOMMENDED_INDEXES EXPLAIN-mode를 사용할 수 있습니다. 색인 마법사에 대한 자세한 정보는 제어 센터로 이동한 후 정보 센터로 들어 가십시오.

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

단순 동적 SQL문 분석

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

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

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

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

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

Visual Explain 기능에 대한 도움말(도움말 메뉴에서 액세스할 수 있음)에서는 SQL문의 성능을 향상시키기 위해 액세스 플랜 그래프 창을 해석하는 방법에 대한 세부사항을 제공합니다. 온라인 도움말에는 Visual Explain 사용 방법을 학습하는데 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 체크인하여, 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 데이터베이스에 대한 네트워크를 검색하고 이들 간의 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 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 박스에 나타나도록 컴퓨터에 현재 설정된 프로토콜만 해당

6. 적절한 프로토콜 매개변수를 입력하십시오.
7. 시스템과 연관될 주석을 입력하십시오.
8. 적용을 눌러 노드 디렉토리에 시스템을 추가하십시오.

그런 다음, 해당 시스템에 대해 작업하려는 인스턴스를 추가하십시오.

1. 제어 센터에서, 방금 추가한 시스템에 속하는 인스턴스 오브젝트를 마우스 오른쪽 단추로 누르십시오.
2. 팝업 메뉴에서 추가를 선택하십시오. 인스턴스 추가 창이 열립니다.
3. 필드에 필요한 값을 입력하십시오.
4. 새로 고침 누름 버튼을 누르면 기존 인스턴스 목록이 표시됩니다.
5. 작업하려는 인스턴스를 선택하십시오.
6. 적용 누름 버튼을 누른 다음, 닫기 누름 버튼을 누르십시오.

마지막으로, 해당 인스턴스에서 작업하려는 데이터베이스를 추가하십시오.

1. 제어 센터에서, 데이터베이스 오브젝트를 마우스 오른쪽 단추로 누르십시오.
2. 팝업 메뉴에서 추가를 선택하십시오. 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 눌러 해당 인스턴스에 대한 기존 데이터베이스 목록을 표시하십시오.
5. 데이터베이스를 선택하십시오.
6. 적용 누름 버튼을 누른 다음, 닫기 누름 버튼을 누르십시오.

사용자 관리

데이터베이스 관리자로서, 데이터에 대해 사용자가 갖는 액세스 유형을 제어하거나, 해당 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 작동합니다. 사용자는 해당 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 마우스 오른쪽 단추로 누르십시오. 팝업 메뉴에서 권한 또는 특권을 선택하십시오. 권한 창 또는 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 누르십시오.
4. 창 맨 아래에서, 개별 권한 또는 특권에 대해 예, 아니오 또는 권한 부여를 선택하십시오. 권한 부여는 유효한 옵션이 있는 오브젝트에 대해서만 표시됩니다.

5. 완료되면, 적용 누름 버튼을 누르십시오.

특정 사용자에게 권한이 부여되는 오브젝트를 검토하거나 변경하려는 경우, 사용자를 선택한 후 마우스 오른쪽 단추를 눌러 오브젝트에 대한 권한을 추가 또는 변경하거나, 권한을 제거하십시오.

데이터 이동

DB2에서는 기존 소스에서 테이블로 데이터를 이동시키는 데 도움이 되도록 가져오기, 내보내기 및 로드 유틸리티를 제공합니다. 이 절에서 제공하는 정보는 데이터 이동의 간단한 개요입니다. 데이터 이동에 대한 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(예: Lotus 1-2-3 파일 또는 ASCII 파일)가 들어 있습니다. 또한 가져오기 유틸리티를 사용하면, 내보내기 유틸리티를 사용하여 저장한 테이블 또는 뷰를 재작성할 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 바꾸거나 추가할 수 있습니다.

또한 가져오기 노트북을 사용하여 입력 파일이 들어있는 새로운 테이블을 작성하거나 선택된 테이블에 있는 기존 행을 삭제하고, 입력 파일의 데이터를 사용하여 다시 데이터를 상주시킬 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 누르십시오.

가져오기 노트북의 파일 페이지를 열려면, 다음을 수행하십시오.

1. 제어 센터에서, 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 누르십시오. 기존 테이블은 내용 분할창에 표시됩니다.
3. 내용 분할창에 있는 테이블을 마우스 오른쪽 단추로 누른 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 쉼표와 같이 사용자 정의 분리문자로 구분되는 데이터를 저장하는 방법에 사용됩니다.
 - 워크시트 형식(WSF)
 - 통합 교환 형식(IXF)
PC/IXF는 데이터베이스 테이블 또는 뷰에 대한 구조적인 설명입니다. PC/IXF 형식으로 내보낸 데이터는 또다른 DB2 Universal Database 제품 데이터베이스로 가져오거나 로드될 수 있습니다.
지원되는 특정 제품 및 릴리스에 대해서는 온라인 도움말을 참조하십시오.
3. 선택적: 해당 옵션 누름 버튼을 눌러 파일 유형 수정자를 지정하십시오. 해당 형식의 옵션 창이 열립니다.
4. 가져오기 모드를 선택하십시오. 사용 가능한 가져오기 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 건너뛸 파일에 있는 레코드 수를 입력하십시오.

7. 선택적: **복합 텍스트 필드**에, 실행될 SQL문의 수를 지정하는 숫자를 입력하십시오(실행 가능한 블록에서).
8. 선택적: **십진 데이터에 내재된 소수점 삽입(IMPLIEDECIMALPOINT)** 선택란을 선택하십시오.
9. **메시지 파일 필드**에서, 가져오는 동안 발생한 경고 및 오류 메시지가 들어갈 파일 이름을 입력하십시오.

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

1. **대형 오브젝트(LOB)**를 별도의 파일에서 검색(**LOBSINFILE**) 선택란을 눌러 대형 오브젝트(LOB) 페이지에서 옵션을 사용하십시오.
2. 추가 누름 버튼을 눌러 LOB 경로 목록 박스에 별도의 LOB 파일 위치를 지정하십시오. 이러한 경로에서는 입력 파일의 LOB 컬럼에서 지정된 LOB 파일을 검색합니다(**LOB** 경로 목록 박스에 표시되는 순서대로).
3. **확인**을 눌러 다른 노트북 페이지의 기본값을 승인하고, 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 눌러 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 눌러 가져올 파일 컬럼 속성을 지정하거나 변경하십시오.

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

저장영역 관리

데이터베이스 관리자로서, 테이블과 색인의 크기를 추정하고, 테이블 공간에서 사용할 수 있는 공간량을 검사하여, 테이블 공간이 가득 차면 기존 테이블 공간에 보다 많은 공간을 추가할 필요가 있습니다.

이 절에서는 다음 방법에 대해 설명합니다.

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

테이블 및 색인 크기 계산

크기 계산 대화 상자를 호출하여 새로운 또는 기존 테이블에 또는 색인에 필요한 저장영역 공간의 양을 추정할 수 있습니다. 개별 테이블 및 색인을 선택하고 이를 마우스 오른쪽 단추로 누르거나, 테이블 작성 및 색인 작성 창에서 크기 계산을 선택하여 이 대화 상자를 엽니다. 크기는 특정 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 호출될 때, 테이블의 권장 스펙으로 미리 채워지며, 테이블 및 모든 종속 색인에 관련된 숫자가 들어 있습니다. 새로 고침 누름 버튼을 누르면 추정 크기, 크기 계산 및 최대 크기가 새 행의 총 수 및 새 행의 평균 길이 필드에 입력한 숫자를 근거로 갱신됩니다.

테이블 또는 색인 크기를 계산 중이면, 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: DB2 Universal Database Enterprise-Extended Edition 제품에 대해 크기 계산을 사용하는 경우, 크기 계산치는 테이블에 있는 데이터의 논리 크기를 기본으로 함을 확인하십시오.

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

테이블 크기를 계산하려면, 다음을 수행하십시오.

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

테이블 공간에서 사용 가능한 공간 검사

DMS 테이블 공간에서 사용할 수 있는 공간량을 검사하려면, 다음을 수행하십시오.

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

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

SMS 테이블 공간에서 사용할 수 있는 공간을 검사하려면, 운영 체제가 제공하는 기능을 사용하여 공간 사용을 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 사용에 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, 또다른 컨테이너를 추가할 수 있습니다.

1. 제어 센터에서, 컨테이너를 추가하려는 내용 분할창에 있는 테이블 공간을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오. 테이블 공간 변경 창이 열립니다.
2. 추가를 누르십시오. 컨테이너 추가 창이 열립니다.
3. 파일 또는 원시 장치 단일선택 버튼을 선택한 후, 필드를 완료하십시오. 자세한 내용은 온라인 도움말을 참조하십시오.
4. 확인을 누르십시오.

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 늘릴 수도 있습니다. UNIX 기반 시스템에 있는 SMS 테이블 공간의 경우, 해당 UNIX 기반 시스템 명령을 사용하여 테이블 공간의 크기를 늘릴 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 또다른 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 재지정된 복원을 수행할 수도 있습니다. 데이터베이스 복원 노트북에서 경로 재지정된 복원을 수행할 수 있습니다. 복원하려는 데이터베이스의 팝업 메뉴에서 복원 → 데이터베이스를 선택하십시오.

문제점 해결

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

- 간단한 방식으로 문제점 또는 오류 식별
- 증상을 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 및 Microsoft Windows 운영 체제용 진단 도구

최신 소식과 기술 문서는 <http://www.software.ibm.com/data/db2/library/> 주소의 WWW(World Wide Web)에서 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 엔터프라이즈 전체 데이터에 대한 복사 조작을 정의, 동기화, 자동화 및 관리할 수 있습니다. 호스트 시스템에서 목표 사이트로 데이터를 자동으로 전달할 수도 있습니다. 예를 들어, 데이터와 응용프로그램을 지방 사무실, 소매 아울렛, 심지어 영업 담당자의 노트북에까지 복사할 수 있습니다.

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

제어 센터를 사용하면, 복제 소스로 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 상위 레벨은 다음과 같습니다. 복제 안내 및 참조서에서 자세한 내용을 참조하십시오.

1. 복제 시나리오 설계(소스 및 목표 테이블 맵핑)
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 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 시간 제어를 설정하십시오.

LDAP(Lightweight Directory Access Protocol) 사용

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

LDAP 데이터베이스를 관리하려면, 데이터베이스를 선택한 후 마우스 오른쪽 단추를 누르십시오. 팝업 창에서 수행할 수 있는 함수를 나열합니다. LDAP에 대한 자세한 정보는 471 페이지의 『부록J. LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스』를 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 수행하려면, 정확한 JRE(Java Runtime Environment)가 설치되어야 합니다. JVM은 응용프로그램을 수행하기 위해서는 JRE가, 애플릿을 수행하기 위해서는 Java 가능 브라우저가 될 수 있습니다.

Java 응용프로그램은 머신에서 다른 응용프로그램처럼 수행되며, 정확한 JRE가 설치되도록 합니다.

Java 애플릿으로서 제어 센터 수행

Java 애플릿은 Java 가능 브라우저 내에서 수행되는 프로그램입니다. 제어 센터 애플릿 코드는 원격 머신에 상주하고, 웹 서버를 통해 클라이언트의 브라우저에 제공됩니다. 제어 센터를 Java 애플릿으로서 수행하는 경우, Windows 32비트 또는 OS/2 운영 체제에서 수행 중인 지원되는 Java 가능 브라우저를 사용하십시오. 현재, 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 작동 시 자동으로 시작하도록 설정할 수 있습니다.

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 지정한 다음, sqllib/temp를 사용하십시오. 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 센터 설치 및 구성에 대한 자세한 정보는 사용자 플랫폼의 빠른 시작 매뉴얼을 참조하십시오.

관리용으로 Java 도구 사용

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

- 오브젝트에 대해 작업할 때 추가 항목을 메뉴에 추가합니다.
- 제어 센터 도구 모음에 버튼을 추가합니다.

이 성능을 사용하려면, 적합한 레벨의 Java 소프트웨어가 설치되어 있어야 합니다. 이 기능 사용에 대한 자세한 정보는 507 페이지의 『부록K. 제어 센터 확장』을 참조하십시오.

제2부 설계 구현

제2장 데이터베이스를 작성하기 전에

데이터베이스의 설계를 판별한 후에, 데이터베이스 및 오브젝트를 작성해야 합니다. 이들 오브젝트에는 스키마, 노드 그룹, 테이블 공간, 테이블, 뷰, 랩퍼, 서버, 별칭(Nickname), 유형 맵핑, 함수 맵핑, 별명(alias), 사용자 정의 유형(UDT), 사용자 정의 함수(UDF), 트리거, 제한조건, 색인 및 패키지가 포함됩니다. 이들 오브젝트는 제어 센터에서 또는 응용프로그램의 API를 통한 명령행 처리기의 SQL문을 사용하여 작성할 수 있습니다.

SQL문에 대한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오. 명령행 처리기 명령에 대한 정보는 *Command Reference*를 참조하십시오. API에 대한 정보는 *Administrative API Reference* 매뉴얼을 참조하십시오.

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

이 장에서 제어 센터를 사용하여 타스크를 완료하는 메소드는 상자를 띄워 강조표시됩니다. 이것은 명령행을 사용하는 비교 대상 메소드가 바로 다음에 오며, 때로는 예제를 동반합니다. 일부 경우에, 오직 하나의 메소드만을 표시하는 타스크가 있을 수 있습니다. 제어 센터에서 작업할 때, 여기에 있는 개요 정보보다 더 자세히 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 타스크와 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(native) 성능 또는 DB2 UDB에 의해 제공된 성능 간의 차이점을 사용할 수 있습니다. 정확한 차이점에 대해서는 해당 빠른 시작 매뉴얼 및 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 시동에서 자동으로 시작될 수 있습니다. 서비스는 시스템에 로그인한 사용자가 없어도 실행할 수 있습니다.

데이터베이스를 작성하기 전의 전제조건

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 사용』
- 68 페이지의 『데이터 파티션 사용』
- 72 페이지의 『DB2 중지』

DB2 시작

일반적인 비즈니스를 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 SYSMOINT 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 방법을 사용하여 인스턴스를 시작하십시오.
 - a. 제어 센터를 사용하여 인스턴스를 시작하십시오.

- 1) 인스턴스 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
- 2) 시작하고자 하는 인스턴스를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 시작을 선택하십시오.

- b. 명령행을 사용하여 인스턴스를 시작하려면, 다음을 입력하십시오.

```
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 수행될 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 운영자 또는 전원 사용자 그룹의 구성원이 될 수 있습니다.

파티션된 데이터베이스 환경에서 수행될 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.

- 실행 환경에서 개발 환경을 분리시키기 위해
- 서비스하는 특정 응용프로그램에 대해 각각 개별적으로 조정하기 위해
- 관리자로부터 관련 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 단일 인스턴스에 놓일 수 있습니다.

인스턴스는 노드 디렉토리에 지역 또는 원격으로 카탈로그화됩니다. 기본 인스턴스는 DB2INSTANCE 환경 변수에 의해 정의됩니다. 데이터베이스 작성, 응용프로그램의 강제 해제, 데이터베이스의 모니터 또는 데이터베이스 관리 프로그램 구성의 갱신과 같이 인스턴스 레벨에서만 수행할 수 있는 타스크를 개발하고 유지보수하기 위해 다른 인스턴스에 접속할 수 있습니다. 기본 인스턴스가 아닌 다른 인스턴스에 접속하려고 시도할 때, 해당 인스턴스와 통신하는 방법을 판별하는 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 인스턴스 정의에 대한 정보는 사용자 플랫폼에 적합한 빠른 시작 안내서를 참조하십시오.

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

제어 센터를 사용하려면, 다음을 수행하십시오.

1. 인스턴스 폴더가 나올 때까지 오브젝트 트리를 확장하십시오.
2. 접속하려는 인스턴스를 누르십시오.
3. 선택된 인스턴스 이름을 마우스 오른쪽 단추로 누르십시오.
4. DB2 창에서 접속하려면, 사용자 ID와 암호를 입력한 후 확인을 누르십시오.

명령행을 사용하여 인스턴스를 접속하려면, 다음을 입력하십시오.

```
db2 attach to <instance name>
```

예를 들어, 다음 명령을 수행하면 노드 디렉토리 내에 이미 카탈로그화된 testdb2 라는 인스턴스에 접속됩니다.

```
db2 attach to testdb2
```

testdb2 인스턴스에 대한 유지보수 활동을 수행한 후, 다음 명령을 실행하여 해당 인스턴스에서 접속 해제할 수 있습니다.

```
db2 detach
```

스키마별 오브젝트 구성 및 그룹화

데이터베이스 오브젝트 이름은 단일 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 지정됩니다. 이러한 지정은 외부적으로 또는 내재적으로 완료됩니다.

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 세션 중에 이 값을 변경하려는 경우, SET SCHEMA문을 사용하여 또다른 스키마 이름으로 특수 레지스터를 설정할 수 있습니다. 자세한 정보는 SQL 참조서를 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 경우, 다른 스키마 이름을 사용하는 것이 바람직합니다. 스키마를 명확히 작성하는 방법에 대한 자세한 정보는 140 페이지의 『스키마 작성』을 참조하십시오.

병렬 처리 사용

구성 매개변수를 수정하여 데이터베이스 파티션 내의 또는 파티션되지 않은 데이터베이스 내의 병렬 처리를 사용하십시오. 예를 들어, 파티션 내 병렬 처리를 사용하여 멀티프로세서(SMP) 머신에서 다중 프로세서를 사용할 수 있습니다.

파티션 내 병렬 처리 사용

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 대한 자세한 정보는 *관리 안내서: 성능을 참조하십시오*.

파티션 내 조회 병렬 처리 사용

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

INTRA_PARALLEL

데이터베이스 관리 프로그램 구성 매개변수. 이 매개변수에 대한 자세한 정보는 *관리 안내서: 성능을 참조하십시오*.

DFT_DEGREE

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

DEGREE

정적 SQL에 대한 사전 처리 컴파일 또는 바인드 옵션. 자세한 정보는 *Command Reference*를 참조하십시오.

CURRENT DEGREE

동적 SQL용 특수 레지스터. 자세한 정보는 *SQL 참조서*를 참조하십시오.

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

파티션 간 조회 병렬 처리 사용

파티션 간 병렬 처리는 이 파티션의 데이터베이스 파티션 수와 데이터 분산에 기초하여 자동으로 발생합니다.

유틸리티 병렬 처리 사용

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

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

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

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

- CPU_PARALLELISM
- DISK_PARALLELISM

LOAD 명령에 대한 정보는 *데이터 이동 유틸리티 안내* 및 참조서를 참조하십시오.

자동 로드 프로그램: autoloader.cfg 파일의 LOAD 스펙에 대해 MODIFIED BY ANYORDER 매개변수를 지정하여 자동 로드 프로그램에 다중 분할 프로세스를 사용할 수 있습니다. 자세한 정보는 *데이터 이동 유틸리티 안내* 및 참조서를 참조하십시오.

색인 작성: 색인을 작성할 때 병렬 처리를 사용하려면, 다음과 같은 상태여야 합니다.

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

CREATE INDEX문에 대한 정보는 *SQL 참조서*를 참조하십시오.

데이터 파티션 사용

파티션된 데이터베이스 환경에서 수행할 때, CREATE DATABASE 명령 또는 sqlcrea() API를 사용하여 db2nodes.cfg 파일에 존재하는 모든 노드로부터 데

이터베이스를 작성할 수 있습니다. 자세한 정보는 *Command Reference* 및 *Administrative API Reference* 매뉴얼을 참조하십시오.

파티션된 데이터베이스를 작성하기 전에, 데이터베이스가 작성될 인스턴스로 지역 클라이언트로서 접속할 것인지 원격 클라이언트로서 접속할 것인지를 판별해야 합니다. 두 번째로는 인스턴스에 접속해야 합니다. 또한, 어떤 데이터베이스 파티션이 데이터베이스에 대한 카탈로그 노드가 될 것인지도 선택해야 합니다. 접속 후, `CREATE DATABASE` 명령을 실행하는 데이터베이스 파티션은 해당 특정 데이터베이스에 대한 카탈로그 노드가 됩니다.

카탈로그 노드는 모든 시스템 카탈로그 테이블이 저장되는 데이터베이스 파티션입니다. 시스템 테이블로의 모든 액세스는 이 데이터베이스 파티션을 거쳐야 합니다. 모든 연합 데이터베이스 오브젝트(래퍼, 서버, 별칭 등)는 이 노드에서 시스템 카탈로그 테이블에 저장됩니다.

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

주: 카탈로그 노드를 정기적으로 백업하고 사용자 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 첫 번째 데이터베이스가 작성될 때, 시스템 데이터베이스 디렉토리가 형성됩니다. 이것은 사용자가 작성하는 다른 데이터베이스에 대한 정보와 함께 추가됩니다. 시스템 데이터베이스 디렉토리는 `sqldbdir`이며, 홈 디렉토리 아래의 `sqllib` 디렉토리에 위치합니다. 이 디렉토리는 공유 파일 시스템(예: UNIX 플랫폼의 NFS)에 상주해야 하는데, 그 이유는 파티션된 데이터베이스를 구성하는 모든 데이터베이스 파티션용으로 하나의 시스템 데이터베이스 디렉토리만 있기 때문입니다.

또한, `sqlldbidr` 디렉토리에는 시스템 인텐션 파일도 상주합니다. 이것은 `sqldbins` 라 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. 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 매개변수로는 `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 대한 자세한 정보는 [관리 안내서: 성능을 참조하십시오](#).

데이터베이스 테이블 공간 백업

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

- 둘 이상의 목표 미디어를 사용하십시오.
- 병렬 I/O에 대해 테이블 공간을 구성하십시오.
- `BACKUP` 명령에서 `PARALLELISM` 매개변수를 사용하여 병렬 처리 수준을 지정하십시오.
- `BACKUP` 명령에서 `WITH num-buffers BUFFERS` 매개변수를 사용하여 병렬 처리 수준을 수용할 만큼 충분한 버퍼가 있는지 확인하십시오. 버퍼 수는 소유하고 있는 목표 미디어의 수에 선택된 병렬 처리 수준을 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 백업 중인 테이블 공간의 가장 큰(Extent 크기 * 컨테이너 수) 제품만큼 크게

BACKUP DATABASE 명령에 대한 정보는 *Command Reference*를 참조하십시오.

데이터베이스/테이블 공간 복원

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

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

또한, 다음과 같은 기본 복원 버퍼 크기를 사용하십시오.

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

RESTORE DATABASE 명령에 대한 정보는 *Command Reference*를 참조하십시오.

DB2 중지

db2stop 명령은 서버에서만 수행될 수 있습니다 이 명령이 수행 중일 때에는 어떠한 데이터베이스 연결도 허용되지 않지만, 인스턴스 침부가 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 **SYSMAINT** 권한이 있는 사용자 ID 및 이름으로 로그인 또는 접속하거나 인스턴스 소유자로 로그인하십시오.
2. 중지시키고자 하는 특정 데이터베이스에 연결된 모든 응용프로그램과 사용자를 표시하십시오. 중요한 응용프로그램이 수행되고 있지 않은지 확인하려면, 응용프로그램을 나열해야 합니다. 이에 대해 **SYSADM**, **SYSCTRL** 또는 **SYSMAINT** 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, **INSTHOME**은 사용할 인스턴스의 홈 디렉토리입니다.

5. 다음 방법 중 하나를 사용하여 인스턴스를 중지시키십시오.

- a. 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
- b. 중지시키려면 각 인스턴스를 누르십시오.
- c. 인스턴스를 마우스 오른쪽 단추로 선택한 후, 자동(팝업) 메뉴로부터 중지를 선택하십시오.
- d. 중지 확인 창에서, 확인을 누르십시오.

명령행을 사용하여 인스턴스를 중지시키려면, 다음을 입력하십시오.

```
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) 작성
- 노드 구성 파일 작성
- 데이터베이스 구성 파일 작성
- 응답 파일을 사용한 구성 정보 복제
- FCM 통신 작동

데이터베이스의 논리적 특성 및 물리적 특성 설계

데이터베이스를 작성하기 전에, 데이터베이스 설계에 대한 논리 및 실제 데이터베이스를 작성해야 합니다. *관리 안내서: 계획을 참조하십시오.*

인스턴스 작성

인스턴스는 데이터베이스를 카탈로그화하고 구성 매개변수를 설정하는 논리 데이터베이스 관리 프로그램 환경입니다. 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. 일단 인스턴스 디렉토리가 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 기타 모든 파일

UNIX 운영 체제에서, 인스턴스 디렉토리는 INSTHOME/sqllib/sqllib 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 인스턴스는 이름 지정 규칙 지침 내에서 원하는 것을 지정할 수 있습니다. 인스턴스 이름은 디렉토리 구조 설정에 사용됩니다.

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

- 환경 변수 DB2INSTANCE를 『DB2』로 설정합니다.
- DB2 레지스트리 변수 DB2INSTDEF를 『DB2』로 설정합니다.

UNIX에서, 기본값은 이름 지정 규칙 지침 내에서 원하는 것을 지정할 수 있습니다.

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

DB2를 사용하기 전에, 각 사용자에 대한 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 디렉토리 아래에 있는 sqllib 서브디렉토리에 위치해 있습니다. 인스턴스 소유자 또는 인스턴스의 SYSADM 그룹에 속하는 모든 사용자는 인스턴스의 모든 사용자를 위해 스크립트를 사용자 정의할 수 있습니다. 또한, 스크립트는 각 사용자를 위해 복사되고 사용자 정의할 수도 있습니다.

샘플 스크립트에는 다음을 수행하기 위한 명령문이 들어 있습니다.

- 인스턴스 소유자의 홈 디렉토리의 sqllib 서브디렉토리 아래에 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 수행하도록 할 수 있습니다.

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

- 하나의 스크립트 버전을 공유하는 사용자의 경우, 다음을 추가하십시오.

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

여기서, INSTHOME은 사용할 인스턴스의 홈 디렉토리입니다.

- 홈 디렉토리에 사용자 정의된 스크립트 버전이 있는 사용자의 경우, 다음을 추가하십시오.

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

여기서, USERHOME은 사용자의 홈 디렉토리입니다.

DB2 환경 수동 설정

주: 다음 설명은 UNIX 운영 체제 환경에만 적용됩니다.

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

- 하나의 스크립트 버전을 공유하는 사용자의 경우, 다음을 추가하십시오.

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

여기서, INSTHOME은 사용할 인스턴스의 홈 디렉토리입니다.

- 홈 디렉토리에 사용자 정의된 스크립트 버전이 있는 사용자의 경우, 다음을 추가하십시오.

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

여기서, USERHOME은 사용자의 홈 디렉토리입니다.

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

창 1의 경우,

- Bourne 또는 Korn 셸에서, 다음을 입력하십시오.

```
. /u/test/sql1lib/db2profile
```

- C 셸에서, 다음을 입력하십시오.

```
source /u/test/sql1lib/db2cshrc
```

창 2의 경우,

- 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 디렉토리 아래에 위치해 있는지 검증하십시오.

시스템의 다중 인스턴스

시스템에 둘 이상의 인스턴스가 있을 수 있습니다. 한 번에 하나의 DB2 인스턴스 내에서만 작업할 수 있습니다.

인스턴스 소유자 및 그룹, 즉 모든 인스턴스에 연관된 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 경우, 잠재적으로 인스턴스에 연관된 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. (또

한, 1차 그룹을 인스턴스 소유자의 사용자 ID 또는 사용자 이름에 지정하도록 하십시오. 그렇지 않으면, 시스템 기본값 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.

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

인스턴스 추가

OS/2에서 관리자 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.

또다른 인스턴스를 추가하려면, 다음 단계를 수행하십시오.

1. 관리자 권한이 있거나 관리자 그룹에 속하는 사용자 ID 또는 이름으로 로그인 하십시오.
2. 인스턴스를 추가하려면, 다음 방법 중 하나를 사용하십시오.
제어 센터를 사용하려면, 다음을 수행하십시오.

- | |
|--|
| <ol style="list-style-type: none">a. 원하는 시스템의 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.b. 인스턴스 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 추가를 선택하십시오.c. 정보를 완료하고, 적용을 누르십시오. |
|--|

명령행을 사용하여 인스턴스를 추가하려면, 다음을 입력하십시오.

```
db2icrt <instance_name>
```

3. 관리 서버를 작성하십시오.

또다른 DB2 인스턴스를 추가하기 위해 **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 대한 자세한 정보는 263 페이지의 『제5장 데이터베이스 액세스 제어』를 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 이외의 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 완료됩니다.

인스턴스 작성 시 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 판별 중에 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 않습니다.

- -u FencedID

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

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

- -p PortName

이 매개변수는 사용될 TCP/IP 서비스 이름 또는 포트 번호를 지정합니다. 그런 다음, 이 값은 인스턴스의 모든 데이터베이스에 대해 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 사용할 수 있습니다.

```
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 사용할 수 있습니다.

```
db2icrt db2inst1 -s client -u fencedID
```

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

인스턴스 작성 시 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 디렉토리 아래에서 작성되고 주어진 공유 이름인 DB2는 인스턴스 이름에 연결됩니다. 읽기 및 쓰기 사용권한은 도메인에 있는 모든 사용자에게 자동으로 권한 부여됩니다. 사용권한을 변경하여 디렉토리에 대한 액세스를 제한할 수 있습니다.

다른 인스턴스 프로파일 경로를 지정하면, 공유 드라이브 또는 디렉토리를 작성해야 합니다. 사용권한이 변경되지 않았다면, 이를 통해 도메인의 모든 사용자가 인스턴스를 직접 액세스하는 기회를 허용합니다.

- /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 Communication 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 - Extended 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 늘리십시오.
- 서비스로 로그인하십시오.
- 프로세스 레벨 토큰을 바꾸십시오.

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. 시스템의 인스턴스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 인스턴스 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 추가를 선택하십시오.
3. 데이터베이스 추가 창에서, 새로 고침을 누르십시오.
4. 데이터베이스 인스턴스를 보려면 드롭 다운 화살표를 누르십시오.
5. 창을 나가려면 취소를 누르십시오.

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, 다음을 입력하십시오.

```
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. 마우스 오른쪽 단추로 인스턴스를 누른 후, 팝업 메뉴에서 시작을 선택하십시오.
3. 동시에 수행하려는 모든 인스턴스를 시작할 때까지 2단계를 반복하십시오.

명령행을 사용하여 다중 인스턴스를 수행하려면, 다음을 수행하십시오.

1. 다음을 입력하여 시작하려면 DB2INSTANCE 변수를 원하는 다른 인스턴스의 이름으로 설정하십시오.

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

2. **db2start** 명령을 입력하여 인스턴스를 시작하십시오.

사용권 관리

DB2 제품에 대한 사용권 관리는 제품에 대한 온라인 인터페이스의 제어 센터에 있는 사용권 센터를 통해 주로 완료됩니다. 사용권 센터에서 사용권 정보, 통계, 등록된 사용자 그리고 설치된 각 제품의 현재 사용자를 검사할 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 레지스트리 값을 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 수행 중인 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 사용하십시오. 이들 변경사항은 다음 번에 시스템이 재부팅될 때까지 효과가 있습니다. UNIX 플랫폼에서, **set** 명령 대신 **내보내기** 명령이 사용됩니다.

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 나열되어 있습니다. 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 설정이 있습니다. 이 레벨에서 정의된 값은 전역 레벨에서 설정을 겹쳐줍니다.
- DB2 전역 레벨 프로파일 레지스트리. 특정 인스턴스에 대해 환경 변수가 설정되어 있지 않을 경우, 이 레지스트리가 사용됩니다. 이 레지스트리에는 머신 전체에 적용되는 환경 변수 설정이 있습니다. DB2 UDB EEE에는 각 머신마다 하나의 전역 레벨 프로파일이 있습니다.
- DB2 인스턴스 노드 레벨 프로파일 레지스트리. 다양한 데이터베이스 파티션 간에 데이터베이스가 분할된 시스템에서, 이 레지스트리는 모든 노드(머신)에 상주하며 노드에 데이터를 저장한 모든 인스턴스에 대한 환경 변수 설정이 들어 있습니다. 이 레벨에서 정의된 값은 인스턴스 및 전역 레벨의 비교 가능한 설정을 겹쳐줍니다.
- DB2 인스턴스 프로파일 레지스트리. 이 레지스트리에는 이 시스템이 인식하는 모든 인스턴스 이름 목록이 들어 있습니다.

사용자는 **set** 명령을 사용하여 세션 환경 변수 설정값을 변경하여 해당 세션에 대한 DB2 인스턴스 프로파일 레지스트리 환경 변수 설정을 겹쳐줄 수 있습니다.

DB2는 레지스트리 값 및 환경 변수를 검사하고, 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. 인스턴스 프로파일(아래의 db2set -i 명령 사용)로 설정된 레지스트리 값
4. 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
```

현재 또는 기본 인스턴스에 대해 정의된 모든 레지스트리 변수를 나열하려면, 다음을 사용하십시오.

```
db2set
```

프로파일 레지스트리에 정의된 모든 레지스트리 변수를 나열하려면, 다음을 사용하십시오.

```
db2set -all
```

현재 또는 기본 인스턴스에서 레지스트리 변수 값을 표시하려면, 다음을 사용하십시오.

```
db2set registry_variable_name
```

모든 레벨에서 레지스트리 변수 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 상위 레벨의 프로파일(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.

현재 또는 기본 인스턴스에서 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 사용자에게 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 기본값으로 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 파티션 또는 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, LDAP에서 이 DB2 레지스트리 변수를 설정하는 것은 Windows 플랫폼에서만 지원됩니다.

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 사용하십시오.

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

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

```
db2set -r registry_variable_name
```

인스턴스의 노드에 대한 레지스트리 변수를 전역 프로파일 레지스트리의 기본값으로 재설정하려면, 다음을 사용하십시오.

```
db2set -r registry_variable_name node_number
```

OS/2에서 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 환경 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 변수를 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 설치 중에 설정되고 사용자는 이를 수정하지 않아야 합니다. DB2INSTANCE 및 DB2INSTPROF 환경 변수의 설정은 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 적용되도록 시스템을 재부팅하십시오.

다음에 따라 다양한 프로파일 레지스트리가 배치됩니다.

- DB2 인스턴스 레벨 프로파일 레지스트리 파일은 다음 위치에 배치됩니다.

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

- DB2 전역 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

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

- 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 목표 논리 노드로 요청을 경로 지정하기 위해 사용됩니다. 이 환경 변수는 DB2 프로파일 레지스트리가 아닌 응용프로그램 또는 명령이 발행되는 세션에서 설정되어야 합니다. 이 변수가 설정되지 않으면, 목표 논리 노드는 기본적으로 머신의 포트 0으로 정의되는 논리 노드가 됩니다.

환경 변수의 설정을 판별하려면, **echo** 명령을 사용하십시오. 예를 들어, DB2PATH 환경 변수의 값을 검사하려면, 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 더블 클릭하십시오.
- 시스템 제어판의 시스템 환경 변수 세션에서, 다음을 수행하십시오.
 1. DB2INSTANCE 변수가 없을 경우, 다음을 수행하십시오.
 - 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 추가하십시오.
 - a. DB2INSTANCE 환경 변수를 선택하십시오.
 - b. 값 필드를 db2inst와 같은 인스턴스 이름으로 변경하십시오.

3. 설정을 선택하십시오.
4. 확인을 선택하십시오.
5. 이들 변경사항을 적용하려면 시스템을 재부팅하십시오.

주: DB2INSTANCE 환경 변수는 세션(프로세스) 레벨에서도 설정될 수 있습니다. 예를 들어, TEST라고 하는 두 번째 DB2 인스턴스를 시작하려면, 명령 창에서 다음 명령을 발행하십시오.

```
set db2instance=TEST
db2start
```

프로파일 레지스트리는 다음과 같이 배치됩니다.

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

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

주: *instance_name*은 작업 중인 데이터베이스 파티션에 대해 고유합니다.

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

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

- Windows NT 레지스트리의 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 작업 중인 데이터베이스 파티션에 대해 고유합니다.

DB2 UDB는 원격 머신의 인스턴스 레벨에서 DB2 UDB 레지스트리 변수에 액세스하는 성능을 제공합니다. DB2 UDB 레지스트리 변수는 세 가지 레벨인 머신 또는 전역 레벨, 인스턴스 레벨 및 노드 레벨에 저장됩니다. DB2REMOTEPEG를 사용하면 인스턴스 레벨(노드 레벨 포함)에 저장된 레지스트리 변수를 또다른 머신으로 경로 재지정할 수 있습니다. DB2REMOTEPEG를 설정하면, DB2 UDB는 DB2REMOTEPEG가 지정하는 머신에서 DB2 UDB 레지스트리 변수에 액세스합니다. 예를 들어, 다음과 같습니다.

```
db2set DB2REMOTEPEG=rmtwkstn
```

여기서, *rmtwkstn*은 원격 워크스테이션 이름입니다.

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

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

UNIX 시스템에서 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 체제에서, 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_name*으로 설정하여 실행합니다.

주: PATH 및 DB2INSTANCE를 제외한 기타 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 셸 또는 C 셸의 경우,

```
set db2instance inst1
```

DB2 프로파일 레지스트리가 적절히 관리되도록 하려면, UNIX 운영 체제에서는 다음 파일 소유권 규칙을 반드시 지켜야 합니다. (DB2 관리 서버(DAS)에 대한 정보는 94 페이지의 『DB2 관리 서버(DAS) 작성』을 참조하십시오.)

- DB2 인스턴스 레벨 프로파일 레지스트리 파일은 다음 위치에 배치됩니다.

```
INSTHOME/sql/lib/profile.env
```

이 파일의 액세스 권한 및 소유권은 다음과 같아야 합니다.

```
-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>는 현재 버전임)

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

이 파일의 액세스 권한 및 소유권은 다음과 같아야 합니다.

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

전역 레지스트리 변수를 수정하려면, 사용자가 루트 또는 DAS 인스턴스 소유자로서 로그인되어야 합니다. DB2 관리 서버에 대한 자세한 정보는 94 페이지의 『DB2 관리 서버(DAS) 작성』을 참조하십시오.

- DB2 인스턴스 노드 레벨 프로파일 레지스트리는 다음 위치에 배치됩니다.

```
INSTHOME/sql/lib/nodes/node_number.env
```

디렉토리와 이 파일의 액세스 권한 및 소유권은 다음과 같아야 합니다.

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

이 파일의 액세스 권한 및 소유권은 다음과 같아야 합니다.

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

DB2 관리 서버(DAS) 작성

DB2 관리 서버(DAS)는 다른 DB2 서버의 관리 작업을 보조하는 데에만 사용되는 특수 DB2 관리 제어점입니다. 클라이언트 구성 지원 프로그램 또는 제어 센터를 사용하려면, DAS가 수행 중이어야 합니다. DAS는 다음과 같은 관리 작업을 수행할 때 제어 센터 및 클라이언트 구성 지원 프로그램을 보조합니다.

- DB2 서버의 원격 관리를 사용합니다.
- DB2 및 운영 체제 명령 스크립트의 실행 스케줄을 설정하는 기능을 포함하여 작업 관리 기능을 제공합니다. 이러한 명령 스크립트는 사용자 정의됩니다. 제어 센터는 작업 스케줄을 정의하고, 완료된 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 제어 센터는 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 작업 또는 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
- 기타 DB2 인스턴스를 시작하거나 중지시키기 위한 조치
- 스케줄된 작업 실행
- 연결성 및 프로토콜 구성을 위한 정보 수집

DAS 통신 설정에 대한 자세한 정보는 사용자 플랫폼용 빠른 시작을 참조하십시오.

DAS 작성

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

DAS와 관련하여 설치 프로세스 중 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 경우, 지역 관리자 권한의 사용자를 작성합니다. db2admin create를 입력하십시오. (특정 사용자 계정을 원하는 경우, db2admin create를 발행할 때 『/USER:』 및 『/PASSWORD:』를 사용하십시오.)

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 후, **db2admin setid** 명령을 사용한 사용자 계정 이름과 사용자 암호를 제공하여 해당 소유권을 설정하거나 변경할 수 있습니다. 이 명령에 대한 자세한 정보는 *Command Reference*를 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 실제 접속은 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 없으므로 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.

```
dasict ASName
```

- AIX에서,

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

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

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

- Linux에서,

```
/usr/IBMDB2/<version_id>/instance/  
dasict 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.

일단 관리 서버를 작성하면, 관리 서버를 사용하여 디렉토리 구조 및 액세스 권한을 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 중지시키려면, db2admin stop을 입력하십시오.

주: Windows NT의 두 경우 모두, 이러한 명령을 사용하려면 SYSADM, SYSCTRL 또는 SYSMANT 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/sqllib/db2profile      (Bourne 또는 Korn 셸의 경우)
source INSTHOME/sqllib/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 중지시키려면, 다음을 수행하십시오.
 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 중지시키십시오.

```
db2admin stop
```

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

DAS 나열

머신에서 DAS 인스턴스 이름을 확보하려면, 다음을 입력하십시오.

```
db2admin
```

이 명령은 DAS를 시작하거나 중지시키고, 새로운 사용자 및 암호를 작성하고, DAS 인스턴스를 제거하고, 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 설치 중에 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 대한 자세한 정보는 *Command Reference*를 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 시작을 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 변경하

지 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 갖는 것이 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 같이 사용됩니다.

```
dasiupdt InstName
```

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

DAS를 제거하려면, 다음을 수행하십시오.

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

주: 이 예에서는 db2das00을 제거될 DAS의 이름으로 간주합니다.

3. db2admin drop을 사용하여 DAS를 제거하십시오.

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.
 - . INSHOME/sqllib/db2profile (Bourne 또는 Korn 셸의 경우)
 - source INSHOME/sqllib/db2cshrc (C 셸의 경우)

여기서, INSHOME은 인스턴스의 홈 디렉토리입니다.

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

```
db2admin stop
```

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

5. 로그오프하십시오.

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

```
dasidrop ASName
```

여기서, ASName은 관리 서버의 인스턴스 이름입니다. 이 명령은 설치된 DB2 버전 및 릴리스에 대해 고유한 서브디렉토리 아래에 있는 instance 서브 디렉토리에 있습니다.

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

EEE 시스템으로 DAS 설정

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

설치 중에 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 전개될 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 지원 프로그램에서 별도의 시스템으로 카탈로그화하십시오.
3. 각 새 시스템에서 동일한 인스턴스를 카탈로그화한 후, DAS를 카탈로그화할 때 사용하는 동일한 머신 이름을 매번 지정하십시오.

구성에는 두 가지 양상이 있습니다. 즉, 하나는 DB2 관리 서버(DAS)에 필요하고 다른 하나는 DB2 인스턴스로 관리되는 목표에 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:

product/version:

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

owner/user ID:

db2as

인스턴스 경로:

das_path

install/run host:

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 이름의 예가 들어 있습니다.

표 1. 지원되는 EEE 플랫폼의 경로 이름의 예

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 고유의 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 구성 지원 프로그램에서 인스턴스에 발행된 요청에 대한 조정자 노드입니다.

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

예를 들어, db2inst는 두 개의 실제 머신 또는 호스트에 분산되는 세 개의 노드로 구성됩니다. db2das를 hostA 또는 hostB에서 수행하여 최소한의 요구사항을 충족시킬 수 있습니다.

주:

1. 호스트 A에 있는 파티션 수는 해당 호스트에서 수행될 수 있는 DAS의 수와는 상관이 없습니다. 해당 호스트의 다중 논리 노드(MLN) 구성과 상관없이 hostA에서 하나의 db2as 사본만을 수행할 수 있습니다.
2. 모든 호스트에서 DAS ID db2as를 작성해야 하는 것은 아닙니다. 이것이 수행되는 호스트에만 있으면 됩니다. 또한, DAS ID의 홈 디렉토리를 모든 호스

트에 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. 102 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 리스너 프로세스 또는 디먼, db2cclist를 시작해야 합니다. 명령행에서 수동으로 하거나 또는 시스템 시동 시마다 자동으로 db2cclist를 호출하도록 시스템을 구성할 수 있습니다.

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

```
rah '<install_path>/bin/db2cc1st'
```

예를 들어, AIX에서 이 명령 호출은 다음과 같습니다.

```
rah '/usr/lpp/<v_r_ID>/bin/db2cc1st'
```

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

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

자동: 루트로서 적절한 항목

```
/etc/inittab
```

을 파일에 추가하십시오. 예를 들어, AIX에서 이 명령 호출은 다음과 같이 hostA 및 hostB에서 수행되어야 합니다.

```
mkitab "db2cc1st::once:su - db2inst -c
        /usr/lpp/<v_r_ID>/bin/db2cc1st"
```

머신이 시동될 때마다 사용자의 개입 없이도 db2cc1st가 호출됩니다.

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

리스너 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 바꿀 수 있습니다.

주: 파일은 반드시 있어야 하며 변경사항이 수행된 후 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 리스너를 수동으로 호출한 후, 실패 호스트의 syslogd 파일 /tmp/db2/user.info에서 db2cclst가 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 처리합니다. 장애가 발생하는 경우, Windows NT 레지스트리는 진단 정보를 수록합니다.

보안: DAS가 인스턴스에 대해 몇몇 관리 작업을 수행하려면, 충분한 권한을 가지고 있어야 합니다. 특히, DAS는 목표인 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 작성하고, db2as의 1차 그룹을 db2inst의 2차 그룹으로 작성하는 것으로 충분합니다. 마지막으로, 또 다른 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 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 검증하려면, 다음 명령을 DB2 인스턴스 ID db2inst 또는 DAS ID db2das에서 실행하십시오.

```
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 검증하려면, 다음 명령을 DAS ID db2as에서 실행한 후 전역(-g) 및 인스턴스(-i) 레벨에서 검사하십시오 (하나만 설정하면 됨).

```
db2set -all
```

동일한 행에서, 제어 센터와 db2inst 간에 통신을 가능하게 하려면 db2inst ID에서 다음 명령을 발행하여 DB2COMM 매개변수가 DB2 인스턴스용 『TCPIP』로 설정되어 있는지 검증하십시오.

```
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 *db2stop*을 *db2start* 다음에 발행하고, DAS의 경우는 *db2admin stop* 및 *db2admin start*를 발행하십시오.

관리 서버, 인스턴스 및 데이터베이스의 발견: 알려진 발견(Known Discovery)을 통해 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 맵핑되는 IP 주소는 클라이언트 머신에서 구

성된 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 맵핑할 수 있습니다.

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

- **KNOWN**

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

- **SEARCH**

알려진 발견의 모든 기능을 작동시키고 네트워크 검색도 작동시킵니다.

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

- **DISABLE**

발견을 사용하지 않게 합니다. 이 경우, 네트워크 검색 옵션을 『데이터베이스 마법사 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 대기하게 될 통신 프로토콜을 제어하는 데 사용되고, 해당 클라이언트가 검색 요청을 보낼 때에

도 사용됩니다. *discover_comm* 매개변수는 TCP/IP 또는 NetBIOS로 설정할 수 있습니다. 현재는 이러한 프로토콜만 지원됩니다.

DAS에서, *discover_comm*에 지정된 값은 db2comm에 설정된 값과 같거나 부속 집합이어야 합니다.

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

서버에서, *discover_comm* 매개변수는 DAS 구성 파일에 설정됩니다. 클라이언트 (또는 클라이언트로 작동하는 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 검사하려면, 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 작동하는 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, 다음을 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 센터를 사용하여,

1. 클라이언트 구성 지원 프로그램을 시작하십시오.
2. 클라이언트 설정 누름 버튼을 누르십시오.
3. 통신 탭을 선택하십시오.
4. 수정할 매개변수를 매개변수 창에서 선택하십시오.
5. 수정할 매개변수의 값을 값 상자에서 선택하십시오.
6. 확인 누름 버튼을 눌러 클라이언트 설정 창을 닫으십시오. DB2 메시지 창이 열립니다.
7. 확인 누름 버튼을 누른 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 구성을 선택하십시오.
3. 『환경』 페이지에서, *discover_inst* 매개변수를 선택하십시오.
4. 클라이언트에서 서버 인스턴스를 허용하려면, 사용을 선택한 후 확인을 누르십시오.
5. 오브젝트 트리에서 데이터베이스를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 구성을 선택하십시오.
6. 『환경』 페이지에서, *discover_db* 매개변수를 선택하십시오.
7. 데이터베이스가 클라이언트에서 허용되게 하려면, 사용을 선택한 후 확인을 누르십시오.

DAS를 설정하여 클라이언트 구성 지원 프로그램 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, 다음을 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 설정될 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 프로그램 또는 제어 센터 인터페이스에서 둘 이상의 시스템에 나타날 수 있습니다. 그러

나 시스템마다 인스턴스에 대한 다른 통신 액세스 경로가 있습니다. 사용자는 통신에 대해 다른 DB2 시스템을 조정자 노드로 선택할 수 있으므로 워크로드를 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 설정될 때 DAS가 조정자 노드로 상주하는 데이터베이스 파티션 서버를 사용합니다.

노드 구성 파일 작성

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 Edition을 사용하는 경우, 인스턴스를 작성할 때 노드 구성 파일이 작성됩니다. 노드 구성 파일을 수동으로 작성 또는 수정하려고 하지 마십시오.

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 서브디렉토리 아래의 function 서브디렉토리에 배치하십시오. (저장 프로시저어에 대한 정보는 관리 안내서: 성능의 『저장 프로시저어』를 참조하십시오.) 또다른 예외는 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 갭이 있을 수 있습니다.

일단 노드 번호가 지정되면, 이는 변경될 수 없습니다. (그렇지 않으면, 데이터가 파티션되어 있는, 파티션 맵의 정보가 절충됩니다.)

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

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

`NODEnnnn`

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

hostname

파티션 간 통신용 IP 주소의 호스트 이름(`netname`이 지정될 경우 예외가 발생합니다. 이 경우, 대부분의 통신에 `netname`이 사용되고, `hostname`은 DB2START, DB2STOP 및 `db2_all`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 조합하면 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 `sqlesetc()` API를 사용하여 DB2NODE 환경 변수로 겹쳐쓸 수 있습니다.

다중 노드를 같은 호스트에 가지고 있을 경우(즉, 하나의 호스트에 둘 이상의 *nodenum*이 있을 경우), *logical-port* 번호를 논리 노드에 0에서부터 겹 없이 오름차순으로 지정해야 합니다.

netname

이 매개변수는 선택적으로서, 각각 자체의 호스트 이름을 가지고, 활동 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 겹을 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. (예외는 편집기를 Windows NT에서 사용하지 말아야 한다는 것입니다.) 그러나 데이터 파티션 시에는 노드 번호를 변경하면 안 되므로, 파일 정보의 무결성이 보호되도록 주의해야 합니다. DB2START를 발행하면 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 잠금해제하십시오.

데이터베이스 구성 파일 작성

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

- 데이터베이스 작성시 지정 및/또는 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.

구성 파일에서 매개변수를 수동으로 변경하지 마십시오. 지원 인터페이스를 단지 사용하기만 하십시오.

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 보존하려면 **db2_all**을 사용하십시오.

응답 파일을 사용한 구성 정보 복제

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

하나 이상의 DB2 제품으로 시스템을 설치하고 환경에 맞게 매개변수를 조정한 후, *db2rspgn*을 사용하여 필요한 값을 응답 파일에서 생성할 수 있습니다. 응답 파일은 동일한 시스템을 재작성하는 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/또는

DataLinks 서버 인스턴스를 선택적으로 사용 불가능하게 할 수 있습니다. 대량 전개 발행에 대한 자세한 정보는 *Administering Satellites Guide and Reference*를 참조하십시오.

이 유틸리티의 구문에 대한 세부사항 및 생성된 응답 파일 생성 사용 방법에 대한 세부사항은 해당 빠른 시작을 참조하십시오.

FCM 통신 작동

파티션된 데이터베이스 환경에서는 FCM(Fast Communication Manager)으로 데이터베이스 파티션 간의 대부분의 통신을 다룹니다. 데이터베이스 파티션에서 FCM을 사용하고 다른 데이터베이스 파티션과의 통신을 사용하려면, 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 첫 번째 노드를 추가할 때의 db2ncrt 유틸리티

추가 정보는 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작을 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 db2puser일 경우, DB2_db2puser를 지정하십시오.

port/tcp

데이터베이스 파티션에 대해 예약하고자 하는 TCP/IP 포트

#comment

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

/etc/services 파일을 공유하는 경우, 파일에 할당된 포트 수가 인스턴스의 다중 데이터베이스 파티션의 가장 큰 수와 같거나 더 큰 수인지를 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 있는 같은 고려사항이 적용됩니다. DB2 인스턴스에 정의된 항목이 모든 /etc/services 파일에서 동일해야 합니다(파티션된 데이터베이스에 적용되지 않는 기타 항목은 같은 필요가 없다고 해도).

하나의 인스턴스의 동일한 호스트에 다중 데이터베이스 파티션이 있을 경우, 사용할 FCM에 대해 둘 이상의 포트를 정의해야 합니다. 이를 수행하려면, 두 행을 etc/services 파일에 입력하여 할당 중인 포트 범위를 지정하십시오. 첫 번째 행은 첫 번째 포트를 지정하고, 두 번째 행은 포트 블록의 끝을 나타냅니다. 다음 예에서, 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 둘다 밑줄(_) 문자도 넣어야 합니다.

제3장 데이터베이스 작성

이 장에서는 사용자 데이터베이스 설계 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 타스크도 다루었습니다.

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

데이터베이스를 작성할 때, 다음과 같은 각 타스크가 수행됩니다.

- 데이터베이스가 필요로 하는 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 작성하려면, 다음을 수행하십시오.

- | |
|--|
| <ol style="list-style-type: none">1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.2. 데이터베이스 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 데이터베이스를 선택하십시오.3. 이 타스크를 완료하려면 단계에 따르십시오. |
|--|

연관된 주석 "Personnel DB for BSchiefer Co"와 함께 다음 명령행 처리기 명령은 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』을 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 위치 지정에 대한 정보는 *Command Reference*를 참조하십시오.

데이터베이스 관리 프로그램을 작성할 때 사용자 또는 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』
- 133 페이지의 『테이블 공간 작성』
- 140 페이지의 『스키마 작성』
- 142 페이지의 『테이블 작성 및 데이터 처리』
- 163 페이지의 『트리거 작성』
- 166 페이지의 『사용자 정의 함수(UDF) 및 메소드 작성』
- 170 페이지의 『사용자 정의 유형(UDT) 작성』
- 173 페이지의 『뷰 작성』

- 176 페이지의 『요약 테이블 작성』
- 179 페이지의 『별명 작성』
- 181 페이지의 『랩퍼 작성』
- 182 페이지의 『서버 작성』
- 190 페이지의 『별칭 작성』
- 193 페이지의 『색인, 색인 확장 또는 색인 스펙 작성』

데이터베이스의 실제 구현에 관련된 추가 정보는 [관리 안내서: 계획을 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 처리 중에 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 동안 작성된 시스템 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 테이블 공간 매개변수를 지정하지 않으면, 데이터베이스 관리 프로그램은 시스템 관리 저장영역(SMS) 디렉토리 컨테이너를 사용하여 이들 테이블 공간을 작성합니다. 이 디렉토리 컨테이너는 데이터베이스용으로 작성된 서브디렉토리에 작성됩니다. (데이터베이스 실제 디렉토리에 대한 자세한 정보는 관리 안내서: 계획을 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 데이터베이스를 선택하십시오.
3. 이 작업을 완료하려면 단계에 따르십시오.

명령행을 사용하여 초기 테이블 공간을 정의하려면, 다음을 입력하십시오.

```
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 테이블 공간 정의를 지정하기만 하면 됩니다.

CREATE DATABASE 명령의 MANAGED BY절을 코드화하는 형식은 CREATE TABLESPACE 명령의 MANAGED BY절의 형식과 동일합니다. 추가 예에 대해서는 133 페이지의 『테이블 공간 작성』을 참조하십시오.

데이터베이스를 작성하기 전에 테이블 공간의 설계 및 선택에 대한 정보는 *관리 안내서: 계획* 매뉴얼을 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. 이들 테이블은 SYSCATSPACE 테이블 공간에 저장됩니다.

이들 테이블은 데이터베이스를 조작하는 동안에 갱신됩니다(예를 들어, 테이블이 작성될 때). 사용자가 이들 테이블을 명시적으로 작성하거나 제거할 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 대한 자세한 정보는 *SQL 참조서*를 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 대한 정보는 *SQL 참조서*의 『갱신 가능한 카탈로그 뷰』를 참조하십시오.

데이터베이스가 작성된 후, 323 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 이름(별명 이름이 지정되지 않으면 동일한 데이터베이스 이름으로 대신함)
- 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 대한 정보는 131 페이지의 『데이터베이스 카탈로그화』를 참조하십시오.

작성된 각 데이터베이스의 경우, 항목은 다음 정보가 들어 있는 디렉토리에 추가됩니다.

- CREATE DATABASE 명령으로 제공되는 데이터베이스 이름
- 데이터베이스 별명 이름(별명 이름이 지정되지 않으면 동일한 데이터베이스 이름으로 대신함)
- 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 sqldbdir, 즉 인스턴스의 홈 디렉토리에 있는 sqldbdir 서브디렉토리의 파일에 항상 액세스하는지 확인해야 합니다. 동일한 sqldbdir 서브디렉토리에 있는 시스템 데이터베이스 디렉토리 또는 시스템 인텐션 파일 sqldbins가 공유 파일 시스템의 또다른 파일에 대한 기호 링크일 경우, 예측할 수 없는 오류가 발생할 수 있습니다. 이들 파일은 68 페이지의 『데이터 파티션 사용』에 설명되어 있습니다.

노드 디렉토리

데이터베이스 관리 프로그램은 첫 번째 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 이기종 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 계층입니다.

지역 디렉토리과 함께 목표 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 대한 정보가 중앙 위치에 일단 기록될 수 있으며, 변경사항 또는 갱신사항이 하나의 위치에서 작성될 수 있습니다.

DCE가 DB2를 수행하기 위한 전제조건은 아니지만, 사용자가 DCE 환경에서 작업 중이면, 자세한 정보는 377 페이지의 『부록B. DCE 디렉토리 서비스 사용』을 참조하십시오.

LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스

LDAP(Lightweight Directory Access Protocol)는 디렉토리 서비스에 대한 산업 표준 액세스 방법입니다. 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 디렉토리로 데이터베이스 정보를 제공합니다. 클라이언트가 데이터베이스에 연결할 때, 카탈로그에 대한 정보는 LDAP 디렉토리에서 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 디렉토리를 검색합니다.

LDAP가 DB2를 수행하기 위한 전제조건은 아니지만, 사용자가 LDAP 환경에서 작업 중일 경우, 자세한 정보는 471 페이지의 『부록J. LDAP(Lightweight Directory Access Protocol) 디렉토리 서비스』를 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. 파티션 맵에 대한 세부사항은 *관리 안내서: 계획*을 참조하십시오.
- 파티션 맵 ID를 생성합니다.
- 레코드를 다음 카탈로그 테이블에 삽입합니다.
 - SYSCAT.NODEGROUPS
 - SYSCAT.PARTITIONMAPS
 - SYSCAT.NODEGROUPDEF

제어 센터를 사용하여 노드 그룹을 작성하려면, 다음을 수행하십시오.

1. 노드 그룹 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 노드 그룹 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성을 선택하십시오.
3. 노드 그룹 작성 창에서, 정보를 완료하고, 화살표를 사용하여 사용 가능한 노드 상자에서 선택된 노드 상자로 노드를 이동시킨 후, 확인을 누르십시오.

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 개(0에서 2)의 노드를 구성하는 데이터베이스에서 두 개의 노드(1과 2)의 노드 그룹을 작성합니다.

```
CREATE NODEGROUP mixng12 ON NODES (1,2)
```

노드 그룹 작성에 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

또한, CREATE DATABASE 명령 또는 `sqlcrea()` API는 기본 시스템 노드 그룹, IBMDEFAULTGROUP, IBMCATGROUP 및 IBMTEMPGROUP을 작성합니다. (자세한 정보는 *관리 안내서: 계획*을 참조하십시오.)

데이터베이스 복구 로그의 정의

데이터베이스 복구 로그는 새 테이블의 추가 또는 기존 테이블의 갱신을 포함한 데이터베이스에서 이루어진 모든 변경사항을 기록합니다. 이 로그는 다수의 로그 *extent*로 구성되며, 로그 파일인 분리 파일에 각각 들어 있습니다.

데이터베이스 복구 로그를 사용하여 장애(예: 시스템 전원 중단 또는 응용프로그램 오류)가 불일치 상태에서 데이터베이스를 떠나지 않은 것을 확인할 수 있습니다. 장애가 발생한 경우, 이미 내용이 변경되었으나 파악되지 않은 변경사항은 구간 복원되고, 실제로 디스크에 작성되지 않은 약속된 모든 트랜잭션은 재실행됩니다. 이러한 조치로 데이터베이스의 무결성을 확인할 수 있습니다.

자세한 정보는 *데이터 이동 유틸리티 안내 및 참조서*를 참조하십시오.

유틸리티를 데이터베이스에 바인딩

데이터베이스가 작성되면, 데이터베이스 관리 프로그램은 `db2ubind.lst`에 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 사용자의 플랫폼에 맞는 빠른 시작 매뉴얼을 참조하십시오.

유틸리티를 데이터베이스에 바인드하거나 리바인드해야 하는 경우, 명령행 처리기를 사용하여 다음 명령을 발행하십시오.

```
connect to sample
bind @db2ubind.lst
```

주: `sample` 데이터베이스에 패키지를 작성하려면 이들 파일이 상주하는 디렉토리 내에 있어야 합니다. 바인드 파일은 `SQLLIB` 디렉토리의 `BND` 서브디렉토리에 있습니다. 이 예에서, `sample`은 데이터베이스 이름입니다.

데이터베이스 카탈로그화

새로운 데이터베이스를 작성하면, 시스템 데이터베이스 디렉토리 파일에 자동으로 카탈로그화됩니다. 데이터베이스를 시스템 데이터베이스 디렉토리 파일에 명시적으로 카탈로그화하기 위해 `CATALOG DATABASE` 명령을 사용할 수도 있습니다. `CATALOG DATABASE` 명령을 사용함으로써, 다른 별명 이름으로 데이터베이스를 카탈로그화하거나, 이전에 `UNCATALOG DATABASE` 명령을 사용하여 삭제된 데이터베이스 항목을 카탈로그화할 수 있습니다.

다음 명령행 처리기 명령은 `person1` 데이터베이스를 `humanres`로 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 기타 인스턴스에서 데이터베이스를 카탈로그화할 수도 있습니다. 다음 예에서, 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 시작을 참조하십시오.

DCE(Distributed Computing Environment) 셸 디렉토리에 대한 정보는 128 페이지의 『DCE 디렉토리 서비스』 및 377 페이지의 『부록B. DCE 디렉토리 서비스 사용』을 참조하십시오.

주: 성능을 개선하기 위해, 데이터베이스 디렉토리를 포함하여 디렉토리 파일을 메모리에 캐쉬할 수 있습니다. (디렉토리 캐쉬 사용에 대한 정보는 관리 안내서 : 성능의 『디렉토리 캐쉬 지원(dir_cache)』을 참조하십시오.) 디렉토리 캐싱이 사용되면, 또다른 응용프로그램이 디렉토리에 대해 수행한 변경(예를 들어, CATALOG DATABASE 또는 UNCATALOG DATABASE 명령을 사용하여)은 응용프로그램이 재시작될 때까지 적용되지 않습니다. 명령행 처리기 세션에 의해 사용된 디렉토리 캐쉬를 새로 고치려면, db2 terminate 명령을 발행하십시오.

응용프로그램 레벨 캐쉬뿐만 아니라, 데이터베이스 관리 프로그램 레벨 캐쉬도 내재적으로 데이터베이스 관리 프로그램을 찾아보는 데 사용됩니다. 이러한 『공유』 캐쉬를 새로 고치려면, db2stop 및 db2start 명령을 발행하십시오.

디렉토리 캐쉬에 대한 자세한 정보는 관리 안내서: 성능의 『디렉토리 캐쉬 지원 (dir_cache)』을 참조하십시오.

테이블 공간 작성

데이터베이스 내에서 테이블 공간을 작성하면 컨테이너가 테이블 공간에 지정되고 정의 및 속성이 데이터베이스 시스템 카탈로그에 기록됩니다. 그런 다음, 이 테이블 공간 내에 테이블을 작성할 수 있습니다.

테이블 공간에 대한 설계 정보는 [관리 안내서: 계획을 참조하십시오](#).

CREATE TABLESPACE문의 구문은 [SQL 참조서](#)에 자세히 설명되어 있습니다. SMS 및 DMS 테이블 공간에 대한 정보는 [관리 안내서: 계획을 참조하십시오](#).

제어 센터를 사용하여 테이블 공간을 작성하려면, 다음을 수행하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 공간 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
3. 작업을 완료하려면, 마법사의 단계에 따르십시오.

명령행을 사용하여 SMS 테이블 공간을 작성하려면, 다음을 입력하십시오.

```
CREATE TABLESPACE <NAME>  
  MANAGED BY SYSTEM  
  USING ('<path>')
```

명령행을 사용하여 DMS 테이블 공간을 작성하려면, 다음을 입력하십시오.

```
CREATE TABLESPACE <NAME>  
  MANAGED BY DATABASE  
  USING (FILE'<path>' <size>)
```

다음 SQL문은 세 개의 개별 드라이브에서 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 5,000페이지인 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. (데이터베이스 실제 디렉토리에 대한 자세한 정보는 [관리 안내서: 계획을 참조하십시오.](#))

또한 지정된 경로 이름의 부분이 존재하지 않으면, 데이터베이스 관리 프로그램이 작성합니다. 서브디렉토리가 데이터베이스 관리 프로그램에 의해 작성된 경우, 서브디렉토리는 테이블 공간이 제거될 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 10,000 페이지의 세 개의 논리 볼륨을 사용하여 UNIX 기반 시스템에 DMS 테이블 공간을 작성하고, 입출력 특성을 지정합니다.

```

CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rdb1v6' 10000,
        DEVICE '/dev/rdb1v7' 10000,
        DEVICE '/dev/rdb1v8' 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 파티션된 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 가정

됩니다. 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 지정됩니다. 문자 직렬 장치는 일반적으로 『rhd0』, 『rfd0』 또는 『rmt0』와 유사한 이름으로 지정됩니다. 이들 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 판별하는 데 도움이 됩니다. OVERHEAD 및 TRANSFERRATE 매개변수에 대한 정보는 *관리 안내서: 성능의 『응용프로그램 성능 조정』*을 참조하십시오.

DB2는 병렬 I/O를 사용하는 순차 프리페치 기능을 사용하여 순차 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 수도 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 BUFFPOOL8K
```

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 설정을 수정하는 데 사용될 수 있습니다. 테이블 공간 명령문을 발행하는 트랜잭션은 시스템 카탈로그 경합을 막기 위해 가능한 빨리 확약되어야 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 세 가지 기본 테이블 공간 중 하나는 『TEMPSPACE1』이라고 하는 시스템 임시 테이블 공간입니다.

주: 시스템 임시 테이블이 이러한 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 들어, 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 임의의 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 분할되거나 파티션됩니다. 테이블 공간은 노드 그룹 안에 작성됩니다. 일단 테이블 공간이 노드 그룹에 작성되면, 계속 그 안에 남아 있어야 합니다. 또다른 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 Windows 95 및 Windows 98 운영 체제입니다. 다음 목록은 이 장치 유형을 식별하는 실제 및 논리 방법을 설명합니다.

- Windows에서 실제 하드 드라이브를 지정하려면, 다음과 같은 구문을 사용하십시오.

```
\\.\PhysicalDriveN
```

여기서, N은 시스템의 실제 드라이브 중 하나를 나타냅니다. 이 경우, N을 0, 1, 2 또는 기타 양의 정수로 바꿀 수 있습니다.

```
\\.\PhysicalDisk5
```

- Windows에서 논리 원시 파티션(즉, 포맷되지 않은 파티션)을 지정하려면, 다음과 같은 구문을 사용하십시오.

```
\\.\N:
```

여기서, N:은 시스템의 논리 드라이브 이름을 나타냅니다. 예를 들어, N:은 E: 또는 임의의 기타 드라이브 이름으로 바꿀 수 있습니다.

- 주: 로그를 장치에 작성하기 위해서는 서비스 팩 3이 설치된 Windows NT 버전 4.0이 있어야 합니다.
- UNIX 기반 플랫폼에서, /dev/rhd0와 같이 문자 직렬 장치 이름을 사용하십시오.

Linux에서 원시 입출력 사용

Linux에는 원시 입출력이 수행되기 전에 블록 장치로 바인드되어야 하는 원시 장치 노드의 풀이 있습니다. 블록 장치 바인딩 정보에 대해 원시 중앙 저장소로 작동하는 원시 장치 제어기가 있습니다. 바인딩은 raw라고 불리는 유틸리티를 사용하여 수행하며, 보통 Linux 분배기가 공급합니다.

Linux에서 원시 입출력을 설정하기 전에, 다음 사항을 요청합니다.

- 하나 이상의 여유 IDE 또는 SCSI 디스크 파티션
- Linux 커널 2.4.0 이상(그러나 일부 Linux 분산은 2.2 커널에서 원시 입출력 제공)
- /dev/rawctl 또는 /dev/raw라는 이름의 원시 장치 제어기. 없는 경우 다음과 같이 기호 링크를 작성하십시오.

```
# ln -s /dev/your_raw_dev_ctrl /dev/rawctl
```
- 원시 유틸리티, 보통 Linux 분산이 제공
- DB2 버전 7.1 FixPak 3 이상

주: 현재 원시 입출력을 지원하는 분산 중, 원시 장치 노드의 이름 지정은 다음과 같은 차이점이 있습니다.

분산	원시 장치 노드	원시 장치 제어기
RedHat 6.2	/dev/raw/raw1 to 255	/dev/rawctl
SuSE 7.0	/dev/raw1 to 63	/dev/raw

DB2는 위의 원시 장치 제어기와 대부분 원시 장치 노드의 기타 이름 중 하나를 지원합니다. Linux/390의 DB2는 원시 장치를 지원하지 않습니다.

Linux에서 원시 입출력을 구성하려면, 다음을 수행하십시오.

이 예에서, 사용하는 원시 파티션은 /dev/sda5입니다. 이 원시 파티션에는 어떠한 가치 있는 데이터도 없어야 합니다.

단계 1. 이 파티션에서 4 096바이트 페이지의 수가 필요하면 반올림하여 계산하십시오. 예를 들어, 다음과 같습니다.

```
# fdisk /dev/sda
Command (m for help): p
Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes
   Device Boot   Start    End  Blocks  Id System
 /dev/sda1            1    523  420097   83 Linux
 /dev/sda2           524   1106  4682947+   5 Extended
 /dev/sda5           524   1106  4682947   83 Linux
Command (m for help): q
#
```

/dev/sda5의 페이지 수는 다음과 같습니다.

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

단계 2. 이 파티션에 미사용 원시 장치를 바인드하십시오. 바인드는 머신이 재부팅될 때마다 매번 수행해야 하며, 루트 액세스를 요청합니다. raw -a를 사용하여 어떤 원시 장치 노드가 이미 사용 중인지 보십시오.

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

단계 3. 원시 장치 제어기 및 디스크 파티션에 전역 읽기 사용권한을 설정하십시오. 원시 장치에 다음과 같이 전역 읽기 및 쓰기 사용권한을 설정하십시오.

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+rw /dev/raw/raw1
```

단계 4. DB2에 테이블 공간을 작성하고, 디스크 파티션이 아니라 원시 장치를 지정하십시오. 예를 들어, 다음과 같습니다.

```
CREATE TABLESPACE dms1
  MANAGED BY DATABASE
  USING (DEVICE '/dev/raw/raw1' 11170736)
```

원시 장치의 테이블 공간 또한 DB2가 지원하는 모든 기타 페이지 크기에 대해 지원됩니다.

스키마 작성

데이터를 테이블에 구성하는 동안, 테이블(및 기타 관련 오브젝트)을 서로 그룹화하는 것이 좋습니다. 이 작업은 CREATE SCHEMA문을 사용하여 스키마를 정의함으로써 완료됩니다. 스키마에 대한 정보는 연결한 데이터베이스의 시스템 카탈로그 테이블에 저장됩니다. 이 정보는 다른 오브젝트가 작성될 때, 이 스키마 내에 배치될 수 있습니다.

CREATE SCHEMA문의 구문은 SQL 참조서에 자세히 설명되어 있습니다. 새로운 스키마 이름은 시스템 카탈로그에 아직 존재하지 않으며, "SYS"로 시작하지 않습니다.

사용자가 SYSADM 또는 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 실행해야 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.

스키마 내의 오브젝트로의 직접 액세스는, 스키마가 데이터베이스에서 고유성을 강요하는 데 사용되기 때문에 허용되지 않습니다. 이것은 두 사용자가 두 개의 테이블(또는 기타 오브젝트)을 동일 이름으로 작성할 수 있는 가능성을 고려할 때 명백해집니다. 고유성을 강요하는 스키마가 없는 경우, 제3의 사용자가 테이블을 조회하려고 시도하면 모호성이 존재합니다. 일부 자세한 규정 없이 사용할 테이블을 정하는 것은 불가능합니다.

테이블 이름 규정의 일부로서 스키마 이름을 입력하지 않고도 테이블을 액세스하도록 또다른 사용자를 허용하려면, 해당 사용자에 대해 뷰를 설정할 것을 요청합니다. 뷰의 정의에서는 사용자 스키마를 포함해서 완전한 테이블 이름을 정의하며, 사용자는 단순히 뷰 이름을 사용하여 조회하면 됩니다. 뷰는 뷰 정의의 일부인 사용자 스키마에 의해 완전해집니다.

제어 센터를 사용하여 스키마를 작성하십시오.

1. 스키마 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 스키마 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성을 선택하십시오.
3. 새로운 스키마에 대한 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 스키마를 작성하십시오.

```
CREATE SCHEMA <name> AUTHORIZATION <name>
```

다음은 권한 부여 ID "joe"를 가진 개별 사용자용 스키마를 작성하는 CREATE SCHEMA문의 예입니다.

```
CREATE SCHEMA joeschma AUTHORIZATION joe
```

스키마 설정

특정 DB2 연결 내에서 발행한 동적 SQL문에 규정되지 않은 오브젝트 참조로 사용할 기본 스키마를 설정하려고 할 수 있습니다. 이것은 기본적으로 사용하려는 스키마에 특수 레지스터 CURRENT SCHEMA를 설정하여 수행됩니다. 모든 사용자가 이 특수 레지스터를 설정할 수 있으며, 권한 부여는 필요하지 않습니다.

SET SCHEMA문의 구문은 SQL 참조서 매뉴얼에 자세히 설명되어 있습니다.

다음은 CURRENT SCHEMA 특수 레지스터 설정 방법에 대한 예입니다.

```
SET CURRENT SCHEMA = 'SCHEMA01'
```

이 명령문은 응용프로그램 내에서 사용될 수 있거나 대화식으로 발행될 수 있습니다. 일단 CURRENT SCHEMA 특수 레지스터 값이 설정되면, 이는 동적 SQL문의 규정되지 않은 오브젝트 참조용 규정자(스키마)로서 사용됩니다(데이터베이스 오브젝트로의 규정되지 않은 참조가 있는 CREATE SCHEMA문의 경우 예외).

CURRENT SCHEMA 특수 레지스터의 초기값은 현재 세션 사용자의 권한 부여 ID와 같습니다.

테이블 작성 및 데이터 처리

테이블 안에서 데이터를 어떻게 구성할 것인지를 판별한 후, 다음 단계는 CREATE TABLE문을 사용하여 이 테이블을 작성하는 것입니다. 테이블 설명은 연결한 데이터베이스의 시스템 카탈로그에 저장됩니다.

CREATE TABLE문의 구문은 *SQL 참조서 매뉴얼*에 자세히 설명되어 있습니다. 요약 테이블 작성에 대한 자세한 정보는 176 페이지의 『요약 테이블 작성』을 참조하십시오. 테이블, 컬럼 및 기타 데이터베이스 오브젝트의 이름 지정 규칙에 대해서는 369 페이지의 『부록A. 이름 지정 규칙』에서 자세한 내용을 참조하십시오.

CREATE TABLE문은 테이블에 이름을 부여합니다. 이 이름은 규정되거나 규정되지 않은 식별자이며, 각 컬럼에 대한 정의입니다. 테이블 공간에 단 하나의 테이블만 포함되도록, 각 테이블을 개별 테이블 공간에 저장할 수 있습니다. 테이블이 자주 제거되고 작성될 경우, 개별 테이블 공간에 저장한 다음, 테이블 대신 테이블 공간을 제거하는 것이 훨씬 더 효과적입니다. 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
- *데이터 이동 유틸리티 안내* 및 참조서에서 설명한 대로 파티션된 데이터베이스 환경에서 작동하는 경우 자동 로드 프로그램 유틸리티

테이블 안팎으로 이루어지는 데이터 이동에 대한 세부사항은 *데이터 이동 유틸리티 안내* 및 참조서에 있습니다.

데이터를 테이블에 추가하는 것은 변경을 로그하지 않고 완료될 수 있습니다. CREATE TABLE문의 NOT LOGGED INITIALLY절은 테이블에 변경을 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. 로깅은 후속 작업 단위(UOW)에서 시작됩니다.

테이블은 하나 이상의 컬럼 정의로 구성됩니다. 한 테이블에 대해 최대 500 컬럼을 정의할 수 있습니다. 컬럼은 엔터티의 속성을 나타냅니다. 컬럼에 있는 값은 모두 동일한 유형의 정보입니다. 자세한 정보는 *SQL 참조서*를 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 대한 정보는 369 페이지의 『부록A. 이름 지정 규칙』을 참조하십시오.

컬럼의 데이터 유형은 컬럼에 있는 값의 길이와 컬럼에 유효한 데이터의 종류를 나타냅니다. 데이터베이스 관리 프로그램은 문자열, 숫자, 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 170 페이지의 『사용자 정의 유형(UDT) 작성』에서 설명됩니다.

기본 속성에 대한 스펙은 값이 제공되지 않을 경우에 사용해야 할 값을 표시합니다. 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 테이블을 선택하십시오.
3. 작업을 완료하려면, 마법사의 단계에 따르십시오.

명령행을 사용하여 테이블을 작성하려면, 다음을 입력하십시오.

```
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 또다른 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 계층 구조』라고 합니다. 테이블 계층 구조에서 맨 위에 있는 테이블(상위 테이블이 없는 것)은 계층 구조의 『루트 테이블』이라고 합니다.

다음 절에서는 앞의 예를 토대로 하여 사용자가 고려해야 할 기타 옵션을 다룹니다.

- 145 페이지의 『대형 오브젝트(LOB) 컬럼 고려사항』
- 147 페이지의 『고유 제한조건 정의』

- 153 페이지의 『생성된 컬럼을 새 테이블에 정의』
- 154 페이지의 『사용자 정의 임시 테이블 작성』
- 155 페이지의 『식별 컬럼을 새 테이블에 정의』
- 157 페이지의 『순차 작성』
- 159 페이지의 『IDENTITY 컬럼 및 순차 비교』
- 160 페이지의 『입력된 테이블 작성』
- 160 페이지의 『입력된 테이블에 데이터 처리』
- 160 페이지의 『계층 구조 테이블』
- 160 페이지의 『다중 테이블 공간에 테이블 작성』
- 161 페이지의 『파티션된 데이터베이스에서 테이블 작성』

또한 조회 결과에 기초하여 정의된 테이블을 작성할 수도 있습니다. 이러한 유형의 테이블을 요약 테이블이라고 합니다. 자세한 정보는 176 페이지의 『요약 테이블 작성』을 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,
   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)
IN RESOURCE
```

LOB 컬럼이 1GB보다 크면, 기록하지 마십시오. (보통 사용자는 10MB보다 큰 LOB 컬럼을 기록하는 일은 없을 것입니다.) 컬럼 정의에 지정된 기타 옵션처럼, 기록 옵션을 변경하는 방법은 테이블을 재작성하는 것입니다.

변경사항을 기록하지 않도록 선택했다더라도, 구간 복원이 시스템 생성 오류의 결과이던 응용프로그램 요구이던 상관없이, 변경사항이 구간 복원되도록 하기 위해 LOB 컬럼이 음영처리됩니다. 음영처리는 현재의 저장영역 페이지 내용이 겹쳐쓰게 되지 않도록 하는 복구 기법입니다. 즉, 수정되지 않은 기존의 페이지는 『음영』 사본으로 보존됩니다. 트랜잭션 구간 복원을 지원하는 데 이들 사본이 더 이상 필요하지 않으면 버립니다.

주: RESTORE 및 ROLLFORWARD 명령을 사용하여 데이터베이스를 복원할 때, 『로그되지 않고』 마지막 백업 이후에 작성된 LOB 데이터는 2진 0으로 바뀝니다.

2. LOB 컬럼에 필요한 공간을 최소화하시겠습니까?

CREATE TABLE문의 COMPACT절을 사용하여 LOB 컬럼을 가능한 작게 작성할 수 있습니다. 예를 들어, 다음과 같습니다.

```
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 COMPACT)
IN RESOURCE
```

최소 설치 LOB 컬럼을 가진 테이블에 추가할 경우, 특히 LOB 값의 크기가 늘어나면(이미 만들어진 저장영역 조정 때문에) 성능 비용이 있습니다.

LOB가 SMS 테이블 공간에 위치하고 회소 파일 할당을 지원하지 않는 OS/2와 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 운영 체제와 비교하면 LOB 저장에 많은 실제 디스크 공간을 사용하지 않습니다. COMPACT 옵션은 회소 파일 할당 지원과는 별도로 훨씬 큰 실제 디스크 공간 『절약』을 가능하게 합니다. 어느 정도의 실제 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 보존되지 않고, 필요할 때마다 디스크에서 읽습니다. 디스크로부터 읽을 경우, 카탈로그의 LOB 컬럼이 사용되는 DB2의 성능이 느려질 수 있습니다. 보통 파일 시스템은 자체적으로 데이터를 저장(또는 캐싱)할 장소를 가지고 있기 때문에, 파일 컨테이너에 빌드된 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』
- 148 페이지의 『참조 제한조건 정의』
- 152 페이지의 『테이블 점검 강제 규정』

제한조건에 대한 자세한 정보는 *관리 안내서: 계획에 있는 제한조건 강제 실시의 계획에 대한 절 및 SQL 참조서*를 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 테이블의 데이터를 수정할 때 데이터베이스 관리 프로그램에 의해 자동으로 시행됩니다. 고유 제한조건은 고유 색인을 통해 시행됩니다.

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(기타 고유 제한조건과 함께)의 상위 키로 사용될 수 있습니다. 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 상태가 되도록 시행합니다. 테이블이 고유 키와 함께 작성되면, 데이터베이스 관리 프로그램은 해당 키에 대해 1차 색인을 작성합니다.

고유 제한조건으로 사용된 색인의 일부 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, 데이터를 가져오거나 로드된 후에 고유 키를 작성하십시오. 이로 인해 테이블이 로드되는 동안 색인을 유지보수하는 데 발생하는 오버헤드를 막을 수 있습니다. 그 결과, 색인은 최소량의 저장영역을 사용하게 됩니다.
- REPLACE 모드에서 로드 유틸리티를 사용하려면, 데이터를 로드하기 전에 고유 키를 작성하십시오. 이 경우, 로드 중에 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 KEY절과, CREATE TABLE 또는 ALTER TABLE

문의 REFERENCES절로 설정됩니다. 입력된 테이블에 대한 참조 제한조건의 효과에 대한 자세한 정보는 *SQL 참조서* 및 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 제한조건을 검사하여 이에 따라 관계를 유지보수합니다. 목적은 하나의 데이터베이스 오브젝트가 또다른 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 테이블 간의 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. 테이블 간에는 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 VARCHA(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,
     FIRSTNME  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, WORKDEPT 값에 대해 참조 제한조건을 정의하게 됩니다. 이 제한조건은 두 테이블의 값 사이에 참조 무결성을 시행합니다. 이 경우, EMPLOYEE 테이블에 추가되는 사원은 DEPARTMENT 테이블에서 찾을 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 삭제할 수 없음을 의미합니다.

앞의 예에서는 참조 제한조건을 추가하기 위해 CREATE TABLE문을 사용했지만, ALTER TABLE문을 사용할 수도 있습니다. 225 페이지의 『구조 및 내용에서 테이블 수정』을 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 문제점이 되지 않습니다. 나중에 외부 키를 추가할 수 있습니다(230 페이지의 『기본 키 및 외부 키 추가』 참조). 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 지정은 지정된 참조 제한조건에 따라 참조 무결성이 유지보수됨을 나타냅니다. 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 동일한 데이터 유형과 길이를 가지고 있어야 합니다. 외부 키는 제한조건 이름을 지정 받을 수 있습니다. 이름을 지정하지 않으면, 자동으로 지정됩니다. 쉽게 사용하려면, 제한조건 이름을 지정하고 시스템 생성 이름을 사용하지 않는 것이 바람직합니다.

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 파트의 값에 관계없이 항상 유효합니다.

다음과 같은 규칙이 외부 키 정의에 적용됩니다.

- 한 테이블이 여러 개의 외부 키를 가질 수 있습니다.
- 외부 키는 키의 모든 파트가 널(NULL) 입력 가능 상태일 때 널(NULL) 입력 가능합니다.
- 어떤 파트가 널(NULL) 값일 경우, 외부 키 값은 널(NULL)이 됩니다.

REFERENCES절: REFERENCES절은 테이블간의 관계에서 상위 테이블을 식별하고, 필요한 제한조건을 정의합니다. 컬럼 정의에 이를 포함시키거나, CREATE TABLE 또는 ALTER TABLE문에서 FOREIGN KEY절을 동반한 개별 절로서 이를 포함시킬 수 있습니다.

REFERENCES절을 컬럼 제한조건으로서 지정하면, 내재된 컬럼 목록이 컬럼 이름으로 구성됩니다. 다중 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 지정되어 있는 경우에는 어느 부서도 삭제될 수 없음을 의미합니다. 기타 삭제 규칙으로는 ON DELETE CASCADE, ON DELETE SET NULL 및 ON DELETE RESTRICT가 있습니다. 참조 무결성 구현시 DELETE 규칙에 대한 자세한 정보는 *관리 안내서: 계 획을 참조하십시오.*

유틸리티 조작에 대한 포함: 로드 유틸리티는 자체 참조 및 종속 테이블을 점검 보류 상태로 둬으로써 테이블에 대한 제한조건 점검을 끕니다. 로드 유틸리티가 완료되면, 꺼진 모든 테이블에 대해 제한조건 점검을 다시 켜야 합니다. 예를 들어, DEPARTMENT 및 EMPLOYEE 테이블만 점검 보류 상태에 있을 경우, 다음 명령을 실행할 수 있습니다.

```
SET INTEGRITY FOR DEPARTMENT, EMPLOYEE IMMEDIATE CHECKED
```

가져오기 유틸리티는 다음과 같은 방법으로 참조 제한조건에 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 제거하십시오. 가져오기가 완료되면, ALTER TABLE문으로 외부 키를 재작성하십시오.
- 자체 참조 제한조건을 가진 테이블로의 가져오기 성공 여부는 행이 가져오는 순서에 따라 달라집니다.

테이블 점검 강제 규정

테이블 점검 강제 규정은 테이블 점검 강제 규정이 정의된 테이블의 각 행에 대해 시행되는 검색 조건을 지정합니다. 테이블이 작성되거나 변경될 때 테이블과 점검 제한조건 정의를 연관시킴으로써 테이블에 대해 테이블 점검 강제 규정을 작성할 수 있습니다. 이 제한조건은 INSERT 또는 UPDATE문이 테이블에 있는 데이터를 수정할 때 자동으로 활성화됩니다. 테이블 점검 강제 규정은 DELETE 또는 SELECT문에 아무런 영향을 주지 않습니다. 점검 제한조건은 입력된 테이블에 연관될 수 없습니다.

제한조건 이름은 동일한 CREATE TABLE문에서 지정된 다른 제한조건과는 달라야 합니다. 제한조건 이름을 지정하지 않으면, 시스템이 제한조건에 대해 18자의 고유 식별자를 생성합니다.

테이블 점검 강제 규정은 키 고유성 또는 참조 무결성 제한조건에서 다루지 않는 데이터 무결성 규칙을 시행하는 데 사용됩니다. 일부 경우에는 테이블 점검 강제 규정이 도메인 검사를 구현하는 데 사용될 수도 있습니다. CREATE TABLE문에서 발행된 다음 제한조건은 모든 활동의 시작일이 동일한 활동의 종료일보다 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 강제 규정을 추가하기 위해 CREATE TABLE문을 사용했지만, ALTER TABLE문도 사용될 수 있습니다. 225 페이지의 『구조 및 내용에서 테이블 수정』을 참조하십시오.

생성된 컬럼을 새 테이블에 정의

생성된 컬럼은 저장된 값이 삽입 또는 갱신 조작을 통해 지정되기 보다는 표현식을 사용하여 계산되는 기본 테이블에 정의됩니다. 특정 표현식 또는 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 두 가지 경우가 있습니다.

1. 표현식의 평가가 조회 중 여러 번 수행된 경우
2. 계산이 복잡한 경우

조회 성능을 개선시키기 위해, 표현식의 결과를 포함하는 추가 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 바꿀 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 후, 색인은 생성된 컬럼을 사용하여 작성될 수 있습니다. 예를 들어, 다음과 같습니다.

```
CREATE INDEX i1 ON t1(c4)
```

조회는 생성된 컬럼의 이점을 사용할 수 있습니다. 예를 들어, 다음과 같습니다.

```
SELECT COUNT(*) FROM t1 WHERE c1 > c2
```

다음과 같이 작성할 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 작성할 수 있습니다.

```
SELECT c3 FROM t1 WHERE c3 * c1 > 100
```

생성된 컬럼을 사용하여 조회 성능을 높일 수 있습니다. 결과적으로, 생성된 컬럼은 테이블이 작성되어 데이터 상주된 후에 추가될 수 있습니다. 자세한 정보는 142 페이지의 『테이블 작성 및 데이터 처리』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 삭제됩니다.

임시 테이블을 정의할 수 있는 방법의 예는 다음과 같습니다.

```
DECLARE GLOBAL TEMPORARY TABLE gbl_temp
  LIKE emp1tab1
  ON COMMIT DELETE ROWS
  NOT LOGGED
  IN usr_tbbsp
```

이 명령문은 gbl_temp라고 하는 사용자 임시 테이블을 작성합니다. 사용자 임시 테이블은 emp1tab1의 컬럼과 동일한 이름 및 설명을 가진 컬럼으로 정의됩니다. 내재된 정의만이 컬럼 이름, 데이터 유형, 널 기능 특성 및 컬럼 기본값 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 열려 있지 않은 경우 테이블에 있는 모든 데이터가 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 대한 추가 정보는 *SQL 참조서*를 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 고유 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.

일단 테이블을 작성하면, 식별 컬럼을 포함하기 위해 테이블 설명을 변경할 수 없습니다.

식별 컬럼의 스펙을 허용하는 CREATE TABLE문에 AS IDENTITY절이 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 세 번째 컬럼은 식별 컬럼입니다. 또한 컬럼을 추가할 때 각 행을 고유하게 식별하기 위해 컬럼에서 사용되는 값을 지정할 수 있습니다. 입력된 첫 번째 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 DEFAULT입니다.

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 DEFAULT절은 기존 테이블의 내용을 복사하거나 테이블을 언로드 및 다시 로드할 의도가 있는 곳에서 데이터 보급을 위해 사용됩니다.

주: 식별 컬럼은 파티션된 데이터베이스 환경에서 현재 지원되지 않습니다.

행을 지정된 명시적 식별 컬럼 값으로 테이블에 삽입하는 경우, 그 다음 내부적으로 생성된 값은 갱신되어 있지 않아 테이블의 기존 값과 충돌할 수 있습니다. 중복 값은 오류 메시지를 생성합니다.

새 테이블에서 식별 컬럼 정의에 대한 추가 정보는 SQL 참조서를 참조하십시오.

순차 작성

순차는 값을 자동 생성하도록 하는 데이터베이스 오브젝트입니다. 순차는 고유 키 값을 생성하는 TASK에 이상적으로 맞습니다. 응용프로그램은 순차를 사용하여 데이터베이스 외부에서 고유 카운터를 생성함으로써 있을 수 있는 동시성 및 성능 문제점을 방지합니다.

식별 컬럼 속성과는 달리, 순차는 특정 테이블 컬럼에 고정되어 있지도 고유 테이블 컬럼에 바인드되어 있지도 않으며, 단지 해당 테이블 컬럼을 통해 액세스가 가능합니다.

순차는 작성되거나 변경되어, 다음 중 한 가지 방법으로 값을 생성합니다.

- 바인드하지 않고 단순 증분 또는 감축
- 사용자 정의 한계에 대한 단순 증분 또는 감축 그리고 중지
- 사용자 정의 한계에 대한 단순 증분 또는 감축 그리고 시작으로 돌아간 후 다시 시작

다음은 순차 오브젝트를 작성하는 예입니다.

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

이 예에서, 순차는 order_seq로 부릅니다. 이 순차는 1에서 시작하여 최고 한계 없이 1씩 증가합니다. 최고 한계를 지정하지 않기 때문에 초기로 되돌아가 순환하여 1에서 재시작할 이유가 없습니다. CACHE 매개변수에 연관된 번호는 순차 값의 최대수를 지정하여 데이터베이스 관리 프로그램이 사전 할당하고 메모리에 보존할 수 있도록 합니다.

생성된 순차 번호는 다음과 같은 등록 정보를 가집니다.

- 값은 0 스케일의 임의의 exact 숫자 데이터 유형을 가집니다. 이러한 데이터 유형에는 SMALLINT, BIGINT, INTEGER 및 DECIMAL이 포함됩니다.
- 연속하는 값은 임의의 지정된 정수 증분에 따라 달라질 수 있습니다. 기본 증분 값은 1입니다.

- 카운터 값은 복구 가능합니다. 카운터 값은 복구가 필요할 때 로그에서 재구성할 수 있습니다.
- 값을 캐쉬하여 성능을 향상시킬 수 있습니다. 캐쉬에 값을 사전 할당하고 저장하면, 로그에 대한 동기 입출력을 줄일 수 있는데, 이때 값은 순차에 대해 생성됩니다. 시스템 장애의 이벤트에서 파악받지 못한 모든 캐쉬 값은 절대 사용되지 않으며 잃어버린 것으로 간주됩니다. CACHE에 대해 지정된 값은 잃어버릴 수 있는 순차 값의 최대 번호입니다.

하나 이상의 순차를 포함하는 데이터베이스를 특정 시점 이전까지 복구하면, 일부 순차에 대해 중복 값 생성을 유발합니다. 가능한 중복 값을 방지하려면, 순차가 있는 데이터베이스는 특정 시점 이전까지 복구해서는 안 됩니다.

순차는 단일 노드 데이터베이스에서만 지원됩니다.

순차가 사용되는 두 가지 표현식이 있습니다.

PREVVAL 표현식은 현재 세션 내에서 가장 최근에 생성한 이전 명령문의 지정된 순차 값을 리턴합니다.

NEXTVAL 표현식은 지정된 순차의 그 다음 값을 리턴합니다. NEXTVAL 표현식이 순차의 이름을 지정할 때 새 순차 번호를 생성합니다. 그러나 한 조회 내에서 동일한 순차 이름을 지정하는 NEXTVAL 표현식의 다중 인스턴스가 있으면, 순차 카운터는 각 행의 결과에 대해 한번만 증분됩니다.

동일한 순차 번호는 두 개의 분리된 테이블에서 고유한 키 값으로 사용되는데, 첫 번째 테이블에서는 NEXTVAL 표현식이 있는 순차 번호를, 추가 테이블에서는 PREVVAL 표현식이 있는 순차 번호를 참조합니다.

예를 들어, 다음과 같습니다.

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

NEXTVAL 또는 PREVVAL 표현식은 다음과 같은 위치에서 사용합니다.

- INSERT문, VALUES절

- SELECT문, SELECT 목록
- SET 지정문
- UPDATE문, SET절
- VALUES 또는 VALUES INTO문

IDENTITY 컬럼 및 순차 비교

IDENTITY 컬럼 및 순차 사이에는 유사점이 있는 반면, 차이점 역시 있습니다. 각각의 특성은 데이터베이스 및 응용프로그램 설계시에 사용됩니다.

식별 컬럼은 다음 특성을 가집니다.

- 식별 컬럼은 테이블 작성시에만 테이블의 파트로 정의할 수 있습니다. 한번만 테이블을 작성하면, 변경하여 식별 컬럼을 추가할 수 없습니다. (그러나 기존 식별 컬럼 특성은 변경할 수 있습니다.)
- 식별 컬럼은 자동으로 단일 테이블의 값을 생성합니다.
- 식별 컬럼을 GENERATED ALWAYS로 정의하면, 사용된 값은 항상 데이터베이스 관리 프로그램이 생성합니다. 테이블의 내용을 수정하는 동안 응용프로그램은 소유한 값을 제공하도록 허용되지 않습니다.

순차 컬럼은 다음 특성을 가집니다.

- 순차 오브젝트는 임의의 한 테이블에 고정되어 있지 않은 데이터베이스 오브젝트입니다.
- 순차 오브젝트는 임의의 SQL문에서 사용되는 순차 값을 생성합니다.
- 순차 오브젝트가 응용프로그램에 사용되므로, 지정된 순차에서 다음 값의 조회를 제어하는 데 두 가지 표현식이 사용됩니다. PREVVAL 표현식은 현재 세션 내에서 이전 명령문에 대한 지정된 순차의 가장 최근 생성 값을 리턴합니다. NEXTVAL 표현식은 지정된 순차의 다음 값을 리턴합니다. 이 표현식을 사용하면, 여러 테이블 내에서 여러 SQL문에 걸쳐 동일 값을 사용할 수 있게 됩니다.

이들 특성이 두 항목에 대한 특성의 전부는 아니지만, 이들 특성은 데이터베이스 설계 및 데이터베이스를 사용하는 응용프로그램에 따라 무엇을 사용할 것인가 판별하는 데 도움을 줍니다.

입력된 테이블 작성

CREATE TABLE문의 변형을 사용하여 입력된 테이블을 작성할 수 있습니다. 입력된 테이블에 필요한 모든 정보는 [응용프로그램 개발 안내서](#)를 참조하십시오.

입력된 테이블에 데이터 처리

구조화 유형을 작성하고 해당 테이블 및 하위 테이블을 작성한 다음, 입력된 테이블에 데이터 처리를 할 수 있습니다. 입력된 테이블에 필요한 모든 정보는 [응용프로그램 개발 안내서](#)를 참조하십시오.

계층 구조 테이블

계층 구조 테이블은 입력된 테이블 계층 구조의 구현에 연관된 테이블입니다. 이는 계층 구조의 루트 테이블과 동시에 작성됩니다. 계층 구조 테이블에 필요한 모든 정보는 [응용프로그램 개발 안내서](#)를 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 사용해야만 가능합니다.

제어 센터를 사용하여 다중 테이블 공간에 테이블을 작성하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 테이블 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 테이블을 선택하십시오.
3. 테이블 이름을 입력한 후, 다음을 누르십시오.
4. 사용자 테이블의 컬럼을 선택하십시오.
5. 테이블 공간 페이지에서, 별도의 색인 공간 사용 및 별도의 long 공간을 누른 후, 정보를 지정하고 완료를 누르십시오.

명령행을 사용하여 다중 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 여러 파트를 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.

단일 테이블의 다중 DMS 테이블 공간 사용에 대한 추가 고려사항은 *관리 안내서: 계획*을 참조하십시오.

CREATE TABLE문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

파티션된 데이터베이스에서 테이블 작성

실제로 나누어지거나 파티션된 테이블을 작성하기 전에, 다음과 같은 사항을 고려해야 합니다.

- 테이블 공간은 둘 이상의 데이터베이스 파티션을 확장시킬 수 있습니다. 스캔하는 파티션 수는 노드 그룹의 파티션 수에 따라 달라집니다.

- 테이블을 동일한 테이블 공간에 배치하거나, 첫 번째 테이블 공간과 함께 동일한 노드 그룹과 연관된 또다른 테이블 공간에 배치함으로써 할당할 수 있습니다. 자세한 정보는 *관리 안내서: 계획*을 참조하십시오.

파티션된 데이터베이스 환경에서 테이블을 작성할 경우에는 *파티션 키*라는 추가 옵션이 있습니다. 파티션 키는 테이블의 정의 파트인 키입니다. 이 키는 데이터의 각 행이 저장된 파티션을 판별합니다.

이는 나중에 변경될 수 없기 때문에 적합한 파티션 키를 선택하는 것이 중요합니다. 더욱이, 고유 색인(고유 키 또는 기본 키)은 파티션 키의 상위 집합으로 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, 다음 기본값이 사용됩니다. 기본 *파티션 키가 적합한지 확인*하십시오.

- 기본 키가 CREATE TABLE문에서 지정되지 않은 경우, 기본 키의 첫 번째 컬럼은 파티션 키로 사용됩니다.
- 기본 키가 없을 경우, 긴 필드가 아닌 첫 번째 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 해당 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 데이터베이스 파티션 수를 곱한 만큼 클 수 있습니다. 테이블 공간의 페이지 크기가 8KB인 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 입력된 테이블에서 INSERT, UPDATE 또는 DELETE절과 함께 실행되는 조치 또는 이러한 절에 의해 트리거되는 일련의 조치를 정의합니다. 트리거는 다음 경우에 사용됩니다.

- 입력 데이터의 유효성 확인을 위해
- 새로 삽입된 행의 값을 생성하기 위해
- 상호 참조 목적으로 다른 테이블을 읽기 위해
- 감사 추적 목적으로 다른 테이블에 작성하기 위해

별칭이 있는 트리거를 사용할 수 없습니다.

사용자는 무결성 또는 비즈니스 규칙의 일반 양식을 지원하기 위해 트리거를 사용할 수 있습니다. 예를 들어, 트리거는 주문을 승인하기 전에 고객의 신용 한도를 검사하거나 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 작성하려면, 다음을 수행하십시오.

1. 트리거 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 트리거 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성을 선택하십시오.
3. 트리거용 정보를 지정하십시오.
4. 트리거가 호출하려는 조치를 지정한 후, 확인을 누르십시오.

명령행을 사용하여 트리거를 작성하려면, 다음을 입력하십시오.

```
CREATE TRIGGER <name>  
  <action> ON <table_name>  
  <operation>  
  <triggered_action>
```

다음 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 내용에는 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 참조서를 참조하십시오. 트리거 작성 및 사용에 대한 정보는 응용프로그램 개발 안내서를 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.

atomic 트리거를 작성할 때는 명령문의 끝 문자에 주의해야 합니다. 데이터베이스 관리 프로그램은 기본적으로 『;』 명령문의 끝 표시문자를 고려합니다. 『;』 이외의 문자를 사용할 수 있도록 atomic 트리거를 작성하려면, 스크립트에서 명령문의 끝 문자를 수동으로 편집해야 합니다. 예를 들어, 『;』은 『#』과 같은 또다른 특수 문자로 바꿀 수 있습니다.

그런 다음, 다음 중 하나를 실행해야 합니다.

- 명령 센터에서 선택된 스크립트 탭과 함께 도구—>도구 설정 메뉴에서 분리문자를 변경한 다음, 스크립트를 다음과 같이 수행하십시오.
- 명령행 처리기에서, 다음을 사용하십시오.

```
db2 -td <delimiter> -vf <script>
```

여기서, 분리문자는 대체 명령문의 끝 문자이며, <스크립트>는 내부에 새 분리 문자가 있는 수정된 스크립트입니다.

트리거 종속성

일부 다른 오브젝트에 대한 트리거의 모든 종속성이 SYSCAT.TRIGDEP 카탈로그에 기록됩니다. 트리거는 여러 오브젝트에 따라 달라질 수 있습니다. 이들 오브젝트 및 종속 트리거는 SQL 참조서의 DROP문 설명 부분에 자세히 기술되어 있습니다.

이들 오브젝트 중 하나가 제거되면, 트리거는 작동 불능이 되지만 정의는 카탈로그에 남아 있습니다. 이 트리거의 유효성을 재확인하려면, 카탈로그에서 정의를 검색하여 새로운 CREATE TRIGGER문을 제출해야 합니다.

트리거가 제거되면, 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 유효하지 않게 됩니다.

종속 오브젝트가 뷰이고 뷰가 조작 불가능하게 되면, 트리거도 작동 불능으로 표시됩니다. 작동 불능이라고 표시된 트리거에 종속된 임의의 패키지는 유효하지 않습니다. (자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 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 상속되어 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 WWW(World Wide Web)의 샘플 데이터를 테이블로 만들거나 Lotus

Notes 데이터베이스에 액세스하고 날짜, 보낸 사람, 메일 메시지 텍스트와 같은 정보를 리턴할 수 있습니다. 이 정보는 데이터베이스의 기타 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, 실행 가능한 UDF 관리 방법을 고려해야 합니다.)

UDF의 성능에 대한 통계는 SQL문을 컴파일할 때 중요합니다. 시스템 카탈로그에서 UDF 통계를 갱신하는 방법에 대한 정보는 *관리 안내서: 성능의 『사용자 정의 함수에 대한 통계 갱신』*을 참조하십시오.

CREATE FUNCTION문을 사용하여 특정 응용프로그램에 맞는 UDF를 작성하는 방법에 대한 세부사항은 *응용프로그램 개발 안내서*를 참조하십시오. UDF 구문에 대한 세부사항은 *SQL 참조서*를 참조하십시오.

함수 맵핑 작성

연합 데이터베이스에서 지역 함수 또는 지역 함수 템플릿(168 페이지의 『함수 템플릿 작성』에서 설명)를 하나 이상의 데이터 소스에서 함수와 함께 맵핑할 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 함수(UDF)를 지역 함수에 맵핑해야 합니다.
- 응용프로그램은 기본 맵핑으로 제공되는 다른 기본 작동이 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. 이들 요구사항은 리턴 값에도 적용됩니다.

CREATE FUNCTION MAPPING문을 사용하여 함수 맵핑을 작성하십시오. 예를 들어, Oracle1 서버에서 Oracle AVGNEW 함수와 DB2 동등 함수 간에 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 사용하십시오. 함수 맵핑 속성은 SYSCAT.FUNCMAPPINGS에 저장됩니다.

연합 서버는 입력 호스트 변수를 바인드하거나 LOB, LONG VARCHAR/VARGRAPHIC, DATALINK, 구별 유형 및 구조화 유형의 결과를 검색하지 않습니다. 입력 매개변수 또는 리턴된 값에 이들 유형 중 하나가 포함되면, 함수 맵핑이 작성될 수 없습니다.

함수 맵핑 사용 및 작성에 대한 추가 세부사항은 *응용프로그램 개발 안내서*를 참조하십시오. CREATE FUNCTION MAPPING 구문에 대한 세부사항은 *SQL 참조서*를 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 사용할 수 있게 합니다. 함수 맵핑에서는 DB2에 함수 템플릿 또는 기존의 비슷한 함수가 있어야 합니다.

템플릿은 단지 함수 셀일 뿐입니다. 이는 이름, 입력 매개변수 및 리턴값입니다. 함수에 대한 지역 실행 파일이 없습니다.

함수에 대한 지역 실행 파일이 없으므로, 함수를 데이터 소스에서 사용할 수는 있어도 함수 템플리트를 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 완료되어야 하므로 조회는 실패합니다. 조합 순서가 같으면, 비교 조작은 myfunc로 참조되는 기초 함수가 있는 데이터 소스에서 완료될 수 있습니다.

함수(또는 함수 템플리트)에는 데이터 소스 함수와 동일한 수의 입력 매개변수가 있어야 합니다. 연합 측의 입력 매개변수의 데이터 유형은 데이터 소스 측의 입력 매개변수의 데이터 유형과 호환 가능해야 합니다. 이들 요구사항은 리턴값에도 적용됩니다.

CREATE FUNCTION문을 AS TEMPLATE 키워드와 함께 사용하여 함수 템플리트를 작성하십시오. 템플리트가 작성된 후, CREATE FUNCTION MAPPING 문을 사용하여 템플리트를 데이터 소스에 맵핑합니다.

예를 들어, 서버 S1에서 함수 MTS1FUNC에 대한 함수 템플리트 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 템플리트 사용 및 작성에 대한 세부사항은 *응용프로그램 개발 안내서*를 참조하십시오. CREATE FUNCTION 구문에 대한 세부사항은 *SQL 참조서*를 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 또다른 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 값하고만 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*를 참조하십시오.

UDT는 대부분의 시스템 제공 함수 또는 내장 함수의 인수로는 사용될 수 없습니다. 이러한 조작 및 기타 다른 조작을 사용하려면 사용자 정의 함수(UDF)가 제공되어야 합니다.

다음 경우에만 UDT를 제거할 수 있습니다.

- 기존 테이블에 대한 컬럼 정의에서 사용되지 않습니다.
- 기존 입력된 테이블 또는 입력된 뷰의 유형으로 사용되지 않습니다.
- 제거될 수 없는 UDF 함수에서 사용되지 않습니다. 뷰, 트리거, 테이블 점검 강제 규정 또는 다른 UDF가 이에 종속적인 경우에는, UDF가 제거될 수 없습니다.

UDT가 제거되면, 종속된 다른 함수도 제거됩니다.

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 조작의 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 캐스트 또한 생성됩니다. T_EDUC 유형 값을 SMALLINT 값으로 캐스트하고 SMALLINT 값을 T_EDUC 값으로 캐스트할 수 있습니다.

CREATE DISTINCT TYPE문에 대한 완전한 구문 정보는 *SQL 참조서*를 참조하십시오. 구별 유형 작성 및 사용에 대한 정보는 *응용프로그램 개발 안내서*를 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 대한 세부사항은 *SQL 참조서*를 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. 구조화 유형에 필요한 모든 정보는 [응용프로그램 개발 안내서](#)를 참조하십시오.

유형 맵핑 작성

연합 시스템에서, 유형 맵핑을 통해 데이터 소스 테이블에 있는 특정 데이터 유형 및 뷰를 DB2 구별 데이터 유형에 맵핑할 수 있습니다. 유형 맵핑은 하나의 데이터 소스 또는 데이터 소스의 범위(유형, 버전)에 적용될 수 있습니다.

기본 데이터 유형 맵핑은 내장 데이터 소스 유형과 내장 DB2 유형에 제공됩니다. 새 데이터 유형 맵핑(사용자가 작성하는)은 SYSCAT.TYPEMAPPINGS 뷰에 나열됩니다.

CREATE TYPE MAPPING문으로 유형 맵핑을 작성합니다. 연합 데이터베이스에서 SYSADM 또는 DBADM 권한 중 하나를 보유하여 이 명령문을 사용하십시오.

유형 맵핑 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 맵핑을 작성할 수 없습니다.

유형 맵핑 사용 및 작성에 대한 세부사항은 [응용프로그램 개발 안내서](#)를 참조하십시오. CREATE TYPE MAPPING 구문에 대한 세부사항은 [SQL 참조서](#)를 참조하십시오.

뷰 작성

뷰는 하나 이상의 기본 테이블, 별칭 또는 뷰에서 파생되고 데이터 검색시 기본 테이블과 함께 교환되어 사용될 수 있습니다. 뷰에 표시된 데이터를 변경하면, 테이블의 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, **INSERT**문이 기본 테이블의 식별 컬럼을 직접 참조한 것처럼 동일한 규칙을 적용합니다. **INSERT**문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

위에서 설명된 대로 뷰를 사용하는 것 외에도, 뷰는 다음 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** 문 내의 별칭을 참조하고 다중 위치/전역 뷰(뷰는 서로 다른 시스템에 위치한 다중 데이터 소스에 있는 정보를 조인할 수 있음)를 작성할 수 있습니다.

표준 **CREATE VIEW** 구문을 사용하는 별칭을 참조하는 뷰를 작성할 때, 뷰 작성자 인증 ID 대신 기초를 이루는 오브젝트 또는 데이터 소스에 있는 오브젝트에 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 찾아보기를 줄이고 성능을 개선하기 위해서입니다. 공통 테이블 표현식에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

제어 센터를 사용하여 뷰를 작성하려면, 다음을 수행하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 뷰 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성을 선택하십시오.
3. 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 뷰를 작성하려면, 다음을 입력하십시오.

```
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 대한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

뷰를 사용하여, 응용프로그램에 대해 사용 가능한 테이블 데이터 부속 집합을 만들고, 삽입되거나 갱신될 데이터의 유효성 확인을 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 첫 번째 행에서는 뷰를 이름 지정하고 컬럼을 정의합니다. 이름 EMP_VIEW는 SYSCAT.TABLES의 스키마에서 고유해야 합니다. 뷰 이름은 그 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 검사해야 하며, 맞지 않을 경우 거부됨을 나타냅니다. 이로 인해, 데이터 무결성은 강화되지만 추가 처리가 요구됩니다. 이 절을 생략하면, 삽입 및 갱신 내용이 뷰 정의에 맞는지 검사되지 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 정의에 있는 UDF를 사용하는 뷰를 작성할 수 있습니다. 그러나 이 뷰를 갱신하여 최근 함수를 포함하려면, 뷰를 제거한 후 재작성해야 합니다. 뷰가 UDF에 종속적이면, 해당 함수는 제거될 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. 입력된 뷰에 필요한 모든 정보는 [응용프로그램 개발 안내서](#)를 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 수행될 것이라고 판별한 경우, 조회는 요약 테이블에 대해 실행되며 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. 이러한 테이블에 대한 자세한 개요 정보는 [관리 안내서: 계획](#)을 참조하십시오.

주: 요약 테이블은 정적 SQL 또는 별칭과 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 아래에서 수행 중일 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 때 만큼 많은 이름을 포함해야 합니다. 전체 선택의 결과 테이블이 중복 컬럼 이름 또는 이름 없는 컬럼을 갖는다면, 컬럼 이름 목록이 제공되어야 합니다. 이름 없는 컬럼은 상수, 함수, 표현식 또는 선택 목록의 AS절을 사용하는 이름 지정되지 않은 세트 조작에서 파생됩니다. 컬럼 이름의 목록이 지정되지 않은 경우, 테이블의 컬럼은 전체 선택의 결과 세트 컬럼의 이름을 상속합니다.

요약 테이블을 작성하는 경우, 기본 테이블이 변경될 때 요약 테이블을 자동으로 새로 고치는지 또는 REFRESH TABLE문을 사용하여 새로 고치는지를 지정하는 옵션이 제공됩니다. 기본 테이블을 변경할 때 요약 테이블을 자동으로 새로 고치려면, REFRESH IMMEDIATE 키워드를 지정하십시오. 새로 고침은 다음과 같은 경우에 즉시 유용합니다.

- 기본 테이블에 대하여 수행할 때 오래 걸리는 조화가 있습니다
- 기본 테이블은 자주 변경되지 않습니다
- 새로 고침은 비용이 적게 듭니다.

이러한 상황에서 요약 테이블은 사전 처리된 결과를 제공할 수 있습니다. 요약 테이블 새로 고침을 지연시키려면, REFRESH DEFERRED 키워드를 지정하십시오. REFRESH DEFERRED를 사용하여 지정된 요약 테이블은 기본 테이블 변경사항을 반영하지 않습니다. 요구사항이 필요하지 않은 곳에서 요약 테이블을 사용하십시오. 예를 들어, DSS 조화를 수행할 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 요약 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 지속 기간 값인 이 특수 레지스터에 허용되는 최대값입니다.

주: REFRESH DEFERRED로 정의된 요약 테이블은 정적 SQL 최적화에 사용되지 않습니다.

CURRENT REFRESH AGE 특수 레지스터를 사용하면 새로 고침이 연기된 요약 테이블을 새로 고치기 전에 동적 조회에 사용할 수 있는 시간을 지정할 수 있

습니다. CURRENT REFRESH AGE 특수 레지스터 값을 설정하려면 SET CURRENT REFRESH AGE문을 사용하면 됩니다. CURRENT REFRESH AGE 특수 레지스터 및 SET CURRENT REFRESH AGE문에 대한 자세한 정보는 *SQL* 참조서를 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 합당합니다.

활동이 소스 데이터에 영향을 주기 때문에, 시간이 지난 요약 테이블에는 더 이상 정확한 데이터가 수록되지 않습니다. REFRESH TABLE문을 사용해야 합니다. 자세한 정보는 *SQL* 참조서를 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 별칭을 참조하는 조회 결과를 기초로 한 요약 테이블을 작성할 수 없습니다.

요약 테이블 제한사항의 완전한 명령문은 *SQL* 참조서를 참조하십시오.

별명 작성

별명은 테이블, 별칭 또는 뷰를 참조하는 간접 방법입니다. 그러므로 *SQL*문은 해당 테이블 또는 뷰의 완전한 이름과는 무관합니다. 테이블 또는 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 점점 제한 조건 정의를 제외하고는, 뷰 또는 트리거 정의와 임의의 *SQL*문에서 사용할 수 있습니다.

별명은 테이블 이름이 사용되는 곳이면 어디든지 사용될 수 있으며, 별명의 체인을 따라 순환 또는 반복적으로 참조되지 않을 경우, 또다른 별명을 참조할 수 있습니다.

별명 이름은 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 별명 이름과 달라야 합니다.

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 *SQL*문이 컴파일될 때에는 반드시 있어야 합니다.

별명 또는 별명이 가리키는 오브젝트가 삭제될 경우, 별명과 관련된 모든 패키지에 올바르게 없다는 표시가 주어지고 별명에 의존한 모든 뷰와 트리거에 작동 불능이라는 표시가 주어집니다.

제어 센터를 사용하여 별명을 작성하려면, 다음을 수행하십시오.

1. 별명 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 별명 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성을 선택하십시오.
3. 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 별명을 작성하려면, 다음을 입력하십시오.

```
CREATE ALIAS <alias_name> FOR <table_name>
```

별명은 명령문 컴파일시에 테이블 또는 뷰 이름에 의해 바뀝니다. 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.
 - 기타 별명을 언급할 수 없습니다.
- MVS/ESA용 DB2에 있는 SYNONYM
 - 해당 작성자만 사용할 수 있습니다.
 - 항상 규정되지 않습니다.
 - 참조 테이블이 제거될 때 제거됩니다.
 - 테이블 또는 뷰와 함께 이름 공간을 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 센터 또는 명령행에서 랩퍼를 작성하면 연합 데이터베이스에 등록합니다.

제어 센터를 사용하여 랩퍼를 작성하려면, 다음을 수행하십시오.

1. 연합 데이터베이스 오브젝트 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 연합 데이터베이스 오브젝트 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 랩퍼 작성을 선택하십시오.
3. 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 랩퍼를 작성하려면, 다음을 입력하십시오.

```
CREATE WRAPPER <wrapper_name> LIBRARY '<library_name>'
```

다음 SQL문은 랩퍼 ORACLE8을 Windows NT 운영 체제에 등록합니다.

```
CREATE WRAPPER ORACLE8 LIBRARY 'libnet8.dll'
```

CREATE WRAPPER문 사용에 대한 세부사항은 *SQL 참조서*를 참조하십시오.

서버 작성

연합 데이터베이스에서, 서버를 작성하여 DB2에 데이터 소스를 정의하고 해당 특성인 이름, 랩퍼, 유형, 버전, 위치 및 옵션을 설명하십시오. 이 정보는 별칭을 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. 기타 데이터 소스(예: 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 서버 간의 인증 처리 차이점을 관리하는 사용자 맵핑을 작성할 수 있습니다. 사용자 맵핑은 284 페이지의 『사용자 맵핑』에 자세히 설명되어 있습니다.

서버가 제거되면, 해당 서버에 종속되어 있는 모든 오브젝트(사용자 맵핑, 별칭, 함수 맵핑, 유형 맵핑, 플랜 등)가 제거됩니다.

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 서버를 작성하려면, 다음을 수행하십시오.

1. 연합 데이터베이스 오브젝트 폴더 아래에서 서버 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 서버 폴더를 오른쪽 마우스 단추를 누른 후, 팝업 메뉴에서 서버 작성을 선택하십시오.
3. 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 서버를 작성하려면, 다음을 수행하십시오.

```
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.

CREATE SERVER문 구문에 대한 추가 세부사항은 *SQL 참조서*를 참조하십시오. CREATE SERVER문 사용에 대한 추가 세부사항은 *설치 및 구성 보충 설명서*를 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 설정을 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.
- 인증을 용이하게 합니다. 일부 서버 옵션을 설정하여 사용자 ID와 암호가 적합한 형태(대문자 또는 소문자)로 데이터 소스에 보내는지 확인할 수 있습니다. 예를 들어, `fold_id` 옵션을 설정하여 연합 서버가 사용자 ID를 데이터 소스에 보내기 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 용이하게 할 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 그 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 초기값으로 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 연결 중에 서버 옵션 값을 임시로 변경하십시오. SET SERVER OPTION문은 데이터 소스 연결 다음에 첫 번째 작업 단위(UOW) 내에서 먼저 발행되어야 합니다.

예를 들어, Oracle 서버 ORASEB1에 대한 플랜 힌트를 임시로 사용하려면, 다음 명령문을 발행하십시오.

```
SET SERVER OPTION plan_hints TO 'Y' FOR SERVER ORASEB1
```

서버 옵션 및 설정

다음 표에서는 서버 옵션과 설정할 수 있는 값에 대해 설명합니다. 달리 표시하지 않는 한, 모든 서버 옵션 값을 작은 따옴표로 묶어야 합니다.

표 2. 서버 옵션 및 설정

옵션	유효 설정	기본 설정
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 별칭 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 방향지정될 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 들어, 'TOLLESON'과 'TolLESon'이 동일하게 간주됩니다.)</p>	'N'
comm_rate	<p>연합 서버와 연관된 데이터 소스 간의 통신 비율을 지정합니다. 초당 메가바이트 단위로 표현됩니다.</p>	'2.0'
connectstring	<p>OLE DB 제공업체에 연결하기 위해 필요한 초기화 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.</p>	없음

표 2. 서버 옵션 및 설정 (계속)

옵션	유효 설정	기본 설정
cpu_ratio	연합 서버의 CPU 속도보다 데이터 소스의 CPU 속도가 얼마나 빠르 지 또는 느린지를 나타냅니다. 『1.0』의 값은 데이터 소스의 CPU와 연 합 서버의 CPU 사이에서 같은 속도를 표시합니다. <1.0의 값은 더 느린 데이터 소스 CPU를 표시합니다.> 1.0의 값은 더 빠른 데이터 소 스 CPU를 표시합니다.	‘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 보낸 사용자 ID에 적용됩니 다. 올바른 값은 다음과 같습니다. ‘U’ 연합 서버는 사용자 ID를 데이터 소스로 보내기 전에 대문자 로 변경합니다. 이는 DB2 계열 및 Oracle** 데이터 소스에 대한 논리적 선택입니다(이 표의 끝부분에 있는 주 2 참조). ‘N’ 연합 서버는 사용자 ID를 데이터 소스로 보내기 전에 아무것 도 수행하지 않습니다(이 표의 끝부분에 있는 주 2 참조). ‘L’ 연합 서버는 사용자 ID를 데이터 소스로 보내기 전에 소문자 로 변경합니다. 이들 설정 중 아무것도 사용되지 않으면, 연합 서버는 대문자로 사용자 ID를 데이터 소스로 보내려 합니다. 사용자 ID가 실패하면, 서버는 소 문자로 보내기를 시도합니다.	없음
fold_pw(이 표의 끝부분에 있 는 주 1, 3 및 4 참조)	연합 서버가 인증을 위해 데이터 소스로 보내는 암호에 적용됩니다. 올 바른 값은 다음과 같습니다. ‘U’ 연합 서버는 암호를 데이터 소스로 보내기 전에 대문자로 변 경합니다. 이는 DB2 계열 및 Oracle** 데이터 소스에 대한 논리적 선택입니다. ‘N’ 연합 서버는 암호를 데이터 소스로 보내기 전에 아무것도 수 행하지 않습니다. ‘L’ 연합 서버는 암호를 데이터 소스로 보내기 전에 소문자로 변 경합니다. 이들 설정 중 아무것도 사용되지 않으면, 연합 서버는 대문자로 암호를 데이터 소스로 보내려 합니다. 암호가 실패하면, 서버는 소문자로 보내 기를 시도합니다.	없음

표 2. 서버 옵션 및 설정 (계속)

옵션	유효 설정	기본 설정
io_ratio	연합 서버의 입출력 속도보다 데이터 소스의 입출력 속도가 얼마나 빠 른지 또는 느린지를 나타냅니다. 『1.0』의 값은 데이터 소스의 CPU와 연합 서버의 CPU 사이에서 같은 속도를 표시합니다. <1.0의 값은 더 느린 데이터 소스 CPU를 표시합니다.> 1.0의 값은 더 빠른 데이터 소 스 CPU를 표시합니다.	‘1.0’
node	데이터 소스가 RDBMS에서 인스턴스로 정의되는 이름입니다. 모든 데 이터 소스에 필요합니다. DB2 계열 데이터 소스의 경우, 이 이름은 연합 데이터베이스의 DB2 노드 디렉토리에 지정되어 있습니다. 이 디렉토리를 보려면, db2 list node directory 명령을 발행하십시오. 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.	없음
password	데이터 소스로 암호를 보낼지 여부를 지정합니다. ‘Y’ 암호를 항상 데이터 소스로 보내며 유효성을 확인합니다. 이는 기본값입니다. ‘N’ 암호를 데이터 소스로 보내지 않으며(사용자 맵핑에 관계없이) 유효성을 확인하지 않습니다. ‘ENCRYPTION’ 암호를 항상 암호화 양식을 사용하여 데이터 소스로 보내며 유 효성을 확인합니다. 암호화된 암호를 지원하는 DB2 계열 데 이터 소스에 대해서만 유효합니다.	‘Y’
plan_hints	플랜 힌트가 사용 가능한지 여부를 지정합니다. 플랜 힌트는 데이터 소 스 최적화 알고리즘에 대한 추가 정보를 제공하는 명령문입니다. 특정 조회 유형의 경우, 이 정보는 조회 성능을 향상시킬 수 있습니다. 플랜 힌트는 색인 사용 여부, 사용할 색인 또는 사용할 테이블 조인 순서를 데이터 소스 최적화 알고리즘이 결정하는 데 도움이 됩니다. ‘Y’ 데이터 소스가 플랜 힌트를 지원할 경우, 데이터 소스에서 플 랜 힌트를 사용할 수 있습니다. ‘N’ 데이터 소스에서 플랜 힌트를 사용할 수 없습니다.	‘N’

표 2. 서버 옵션 및 설정 (계속)

옵션	유효 설정	기본 설정
pushdown	<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’
varchar_no_trailing_blanks	<p>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. 별칭(뷰를 포함하여)을 사용할 가능성이 있는 모든 오브젝트도 고려하도록 하십시오.</p> <p>‘Y’ 이 데이터 소스는 DB2와 유사한 공백이 아닌 문자로 채워진 비교 의미론을 가집니다.</p> <p>‘N’ 이 데이터 소스는 DB2와 동일한 공백이 아닌 문자로 채워진 비교 의미론을 갖지 않습니다.</p>	‘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 설정은 암호에 대한 설정이 ‘N’이면 영향을 미치지 않습니다. 암호를 보내지 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 보내는 것은 가능합니다.)

서버로 통과 세션 사용

통과 세션을 통해 응용프로그램은 서버의 원시(native) 클라이언트 액세스 방법 및 원시(native) 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(별칭 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 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

별칭 작성

연합 데이터베이스에서 별칭은 데이터 소스, 테이블, 별명 및 뷰에 대한 식별자입니다. 분산 요청(DR)은 일반적으로 데이터 소스 테이블 또는 뷰가 아닌 별칭을 참조합니다.

별칭은 DB2가 위치 투명성을 제공하는 수단의 일부입니다. 별칭은 데이터 소스 위치 정보에 대한 서버 정의를 사용하여 데이터 소스를 찾고 효과적으로 액세스할 수 있습니다. 예를 들어, ALTER SERVER문은 새 별칭 없이 또는 응용프로그램 코드를 변경하지 않고 서버 성능 데이터와 모든 사용자 및 응용프로그램에 대한 버전 정보를 투명하게 갱신할 수 있습니다.

별칭은 제어 센터 또는 명령행 처리기에서 작성됩니다. 같은 데이터 소스 테이블 또는 뷰에 대해 둘 이상의 별칭을 정의할 수 있습니다.

별칭은 정적 SQL문에서 사용될 수 없습니다.

별칭을 작성하기 전에 데이터 소스에서 RUNSTATS 명령과 동등한 명령을 수행하고 데이터 소스 오브젝트에 대한 통계를 갱신하십시오. 통계적 정보는 별칭이 작성되고 연합 데이터베이스 카탈로그에 저장될 때 데이터 소스에서 수집됩니다. 이 카탈로그 데이터에는 테이블, 컬럼 정의 그리고 사용 가능한 경우 색인 정의와 통계가 들어 있습니다.

다음 SQL문은 별칭 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 사용에 대한 추가 정보는 *SQL* 참조서를 참조하십시오.

별칭 및 데이터 소스 오브젝트 참조

데이터 소스 오브젝트에 대한 참조는 일반적으로 정의된 별칭을 사용합니다. 단 통과 세션에서의 참조는 예외입니다(자세한 정보는 190 페이지의 『서버로 통과 세션 사용』 참조). 예를 들어, 데이터 소스 테이블 DB2MVS1.PERSON.DEPT에 대해 별칭 DEPT를 정의하면, SELECT * FROM DEPT문을 사용할 수 있습니다. SELECT * FROM DB2MVS1.PERSON.DEPT문은 사용할 수 없습니다.

별칭 및 데이터 소스 오브젝트에 대한 작업

대부분의 유틸리티 명령(LOAD, IMPORT, EXPORT, REORGCHK, REORGANIZE TABLE)은 별칭을 지원하지 않습니다.

COMMENT ON은 지원됩니다. 이는 연합 데이터베이스에서 시스템 카탈로그를 갱신합니다.

삽입, 갱신 및 삭제 조작은 별칭에 대해 지원되지 않습니다.

기존의 별칭 및 데이터 소스 식별

여러 개의 별칭을 작성한 후, 다음 정보를 사용하여 주어진 별칭이 해당되는 데이터 소스를 식별하거나 주어진 데이터 소스에서 모든 별칭을 식별할 수 있습니다.

별칭 및 데이터 소스 식별

이 예에서는 별칭(*PAYROLL*)과 작성자(*ACCTG*)를 알고 있지만, 데이터 소스에 대해서 추가 정보가 필요하다고 간주합니다. 다음 SQL문을 사용하여 먼저 *PAYROLL* 이 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_PAYROLL*입니다. 이제 *PAYROLL*이 *DB2_MVS*라고 하는 서버에서 *FINANCE*가 소유한 *DEPTJ35_PAYROLL*이라는 테이블의 별칭임을 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 *DB2MVSDB3*입니다. 이제 테이블 *DEPTJ35_PAYROLL*이 *REGIONW*라고 하는 노드에서 *DB2MVSDB3*이라고 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 검사하여 비슷한 정보를 찾아야 합니다. 예를 들어, 노드가 Oracle 데이터 소스를 위한 것이면, Oracle *tnsnames.ora* 파일에서 비슷한 정보를 얻을 수 있습니다.

시스템 키탈로그 뷰에 대한 세부사항은 *SQL 참조서*를 참조하십시오.

DB2에 알려진 모든 별칭 식별

다음 SQL문은 각 별칭에 대한 스키마 이름과 원격 서버를 비롯한 연합 데이터베이스에 알려진 모든 별칭의 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 컬럼의 값을 오름차순 또는 내림차순으로 지정하기 위해 작성될 수도 있습니다.

색인 확장은 구조화 유형 또는 구별 유형 컬럼을 갖는 색인과 함께 사용하는 색인 오브젝트입니다.

색인 스펙은 메타데이터 구성입니다. 이는 최적화 알고리즘에 데이터 소스 오브젝트(테이블 또는 뷰)에 대해 별칭으로 참조되는 색인이 있다는 것을 알려줍니다. 색인 스펙에는 행 위치 목록이 없습니다. 이는 색인의 설명일 뿐입니다. 최적화 알고리즘은 색인 스펙을 사용하여 별칭으로 표시되는 오브젝트에 대한 액세스를 향상시킵니다. 별칭이 처음으로 작성되면, 데이터 소스에서 기초가 되는 테이블에 대해 DB2에서 인식할 수 있는 형식으로 이루어진 색인이 있을 때 색인 스펙이 생성됩니다.

주: 필요할 경우, 테이블 별칭 또는 하나의 테이블상에 뷰가 있는 뷰 별칭에서 색인 스펙을 작성하십시오.

다음과 같은 경우에 색인 또는 색인 스펙을 수동으로 작성하십시오.

- 성능을 향상시키려는 경우. 예를 들어, 최적화 알고리즘이 특정 테이블 또는 별칭을 중첩된 루프 조인의 내부 테이블로 사용하도록 하려면, 색인이 없을 때 조인하는 컬럼에서 색인 스펙을 작성하십시오. 색인 또는 색인 스펙을 원하는 경우, 자세한 정보는 *관리 안내서: 성능을 참조하십시오.*

- 기본 테이블에 대한 색인이 해당 테이블의 별칭이 작성된 후에 추가된 경우 색인 스펙은 기본 테이블에 색인이 없을 때 작성될 수 있습니다. (DB2는 CREATE INDEX문을 발행할 때 원격 색인을 검사하지 않습니다.) 색인 스펙은 UNIQUE 키워드가 지정되어도 행의 고유성을 강요하지는 않습니다.

DB2 색인 권장 도구는 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 요청 처리 시간이 더 길어질 수 있습니다. 그러므로 잦은 액세스로 인해 분명한 이점이 있는 경우에만 색인을 사용하십시오.

색인 키의 일부인 컬럼은 255바이트로 제한됩니다.

주: DB2_INDEX_2BYTEVARLEN 레지스트리 변수는 색인 키의 파트로 지정된 255바이트보다 큰 길이의 컬럼을 허용하는 데 사용됩니다.

색인에서 컬럼의 최대 수는 16입니다. 입력된 테이블을 색인화하는 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. 마찬가지로, 역시 큰 색인 키는 요청 처리 속도를 느리게 할 수 있습니다.

색인 키는 색인이 정의된 컬럼의 컬렉션 또는 컬럼이며, 색인의 사용성을 판별합니다. 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 작성합니다.

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, 연관된 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 파트는 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 작성할지 여부를 결정할 때, 다음 질문을 고려해야 합니다. 키 삭제가 발생할 때마다 병합할 공간을 검사하는 성능 비용이 추가됩니까? 그리고 공

간이 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 기타 오브젝트(데이터베이스 관리 저장영역으로 작업할 때) 또는 디스크 공간(시스템 관리 저장영역으로 작업할 때)에서 사용될 수 있습니다. 이외에도, 온라인 재구성은 색인의 비 리프 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 대한 자세한 정보는 198 페이지의 『CREATE INDEX문 사용』을 참조하십시오.

파티션된 데이터베이스의 테이블에 대한 색인은 동일한 CREATE INDEX문을 사용하여 빌드됩니다. 이 색인은 테이블의 파티션 키에 의거하여 파티션됩니다. 테이블의 색인은 노드 그룹에 있는 각 노드의 해당 테이블의 지역 색인으로 구성됩니다. 다중 파티션 환경에서 정의된 고유 색인은 파티션 키의 상위 집합이어야 한다는 사실을 주의하십시오.

성능 추가 정보: 다음 작업을 수행하려면, 다음을 수행하십시오.

1. 테이블 작성
2. 테이블 로드
3. 색인 작성
4. RUNSTATS 수행

그런 다음, 작업 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 실제 저장영역이 순서화되지 않았기 때문에 매우 중요합니다. 클러스터링 색인이 정의되지 않은 경우 행이 삽입되면, 행이 들어갈 수 있는 가장 편리한 저장영역 위치에 이 행이 위치합니다. 특정 선택 조건에 부합하는 테이블 행을 검색하고 테이블에 색인이 없으면, 전체 테이블이 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 판별됩니다(160 페이지의 『다중 테이블 공간에 테이블 작성』 참조).

제어 센터를 사용하여 색인을 작성하려면, 다음을 수행하십시오.

1. 색인 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 색인 폴더를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 작성 → 마법사를 사용한 색인을 선택하십시오.
3. 작업을 완료하려면, 마법사의 단계에 따르십시오.

명령행을 사용하여 색인을 작성하려면, 다음을 입력하십시오.

```
CREATE INDEX <name> ON <table_name> (<column_name>)
```

다음 두 절 『색인 사용』 및 198 페이지의 『CREATE INDEX문 사용』에서는 색인 작성에 대한 자세한 정보를 제공합니다.

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 자세한 설명은 *관리 안내서: 성능의 『색인 스캔 개념』*을 참조하십시오.

CREATE INDEX문 사용

고유 키가 아닌 컬럼으로 좀더 효과적으로 검색할 수 있도록 중복(비고유 색인)을 허용하고 색인화된 컬럼에 중복 값이 존재할 수 있도록 색인을 작성할 수 있습니다.

다음 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.

두 컬럼에서 고유 색인을 작성할 수 있는데, 그 중 하나는 포함 컬럼입니다. 기본 키는 포함 컬럼이 아닌 컬럼에서 정의합니다. 두 컬럼 모두 같은 테이블의 기본 키로서 카탈로그에 표시되어 있습니다. 보통 테이블마다 한 개의 기본 키만 있습니다.

INCLUDE절은 색인 키 컬럼 세트에 추가되는 추가 컬럼을 지정합니다. 이 절에 포함된 어떤 컬럼도 고유성을 강요하려고 사용할 수 없습니다. 포함 컬럼은 색인만의 액세스를 통해 일부 조회의 성능을 개선합니다. 컬럼은 고유성을 강요하는 데 사용하는 컬럼과 구별되어야 합니다. (그렇지 않으면, 오류 메시지인 SQLSTATE 42711을 받게 됩니다.) 컬럼의 수와 길이 속성의 합계에 대한 한계는 고유 키와 색인의 모든 컬럼에 적용됩니다.

기존 색인이 기본 키 정의와 일치하는지 판별하기 위해 확인합니다(색인에서 임의의 INCLUDE 컬럼은 무시). 컬럼의 순서 또는 방향(오름차순 또는 내림차순) 권장 스펙에 무관하게 같은 세트의 컬럼을 식별하면, 색인 정의가 일치합니다. 일치하는 색인 정의라면 색인의 설명은 시스템이 요청하는 대로 1차 색인이고, 비 고유 색인이었다면 고유 색인으로 변경됨을 표시하도록 변경됩니다.

이것이 카탈로그에 표시된 것처럼 같은 테이블에서 기본 키를 두 개 이상 가질 수 있는 이유입니다.

구조화 유형으로 작업할 때 사용자 정의 색인 유형을 작성해야 합니다. 여기서는 색인 유지보수, 색인 검색 및 색인 개발 기능 정의 수단이 필요합니다. 색인 유형 작성의 요구사항에 대한 정보는 *SQL 참조서*를 참조하십시오.

다음 SQL문은 EMPLOYEE 테이블의 LASTNAME 컬럼에 INDEX1이라고 하는 클러스터링 색인을 작성합니다.

```
CREATE INDEX INDEX1 ON EMPLOYEE (LASTNAME) CLUSTER
```

데이터베이스의 내부 저장영역을 효과적으로 사용하려면, ALTER TABLE문과 연관된 PCTFREE 매개변수와 함께 클러스터링 색인을 사용하여 올바른 페이지에 새 데이터가 삽입될 수 있도록 하십시오. 올바른 페이지에 데이터가 삽입되면, 클러스터링 순서가 유지보수됩니다. 일반적으로, 테이블에서의 INSERT 활동이 더 클수록 클러스터링을 유지보수하는 데 필요한 (테이블의) PCTFREE 값이 더 커집니다. 이 색인은 데이터가 실제 페이지에 놓이는 순서를 판별하므로, 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 여유 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 대한 추가 개요 정보는 240 페이지의 『테이블 속성 변경』을 참조하십시오. ALTER TABLE에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

또한, 위의 설명은 행의 크기를 증가시키는 UPDATE의 결과인 새로운 "오버플로우" 행에 적용합니다.

CREATE INDEX문의 MINPCTUSED절은 색인 리프 페이지에서 사용한 최소 공간량에 대한 임계값을 지정합니다. 이 절이 사용되면, 온라인 색인 재구성이 이 색인에 대해 사용 가능하게 됩니다. 사용 가능하게 되면, 다음 고려사항을 사용하여 온라인 재구성이 이루어지는지를 판별하게 됩니다. 키가 이 색인의 리프 페이지에서 삭제되고 페이지에서 사용한 공간의 비율이 지정된 임계값보다 작으면, 이웃하는 색인 리프 페이지를 검사하여 두 리프 페이지에 있는 키가 단일 색인 리프 페이지로 병합될 수 있는지를 판별합니다.

예를 들어, 다음 SQL문은 온라인 재구성이 사용 가능 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 여유 공간으로 남겨둘 각 색인 페이지의 비율을 지정합니다. 색인 페이지에 여유 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 수도 있습니다. 이렇게 하면, 각 색인 페이지에서 예약된 공간을 제한할 수 있습니다.

복제된 요약 테이블을 가지고 있는 경우, 해당 기본 테이블은 고유 색인을 가져야 하며 복제된 요약 테이블을 정의하는 조화에 색인 키 컬럼을 사용하십시오. 복제된 요약 테이블에 대한 자세한 정보는 *관리 안내서: 계획을 참조하십시오.*

파티션 내 병렬 처리의 경우, 색인 작성 중에 수행된 데이터 스캐닝 및 정렬에 다중 프로세서를 사용함으로써 색인 작성 성능이 향상됩니다. 다중 프로세서는 *intra_parallel*을 YES(1) 또는 ANY(-1)로 설정하여 사용할 수 있습니다. 색인 작

성 중에 사용되는 프로세서 수는 시스템에 의해 판별되며, 구성 매개변수 `dft_degree`, `max_querydegree`, 응용프로그램 런타임 정도 또는 SQL문 컴파일 정도에 의해 영향을 받지 않습니다. 데이터베이스 구성 매개변수 `indexsort`가 NO로 설정될 경우, 색인 작성은 다중 프로세서를 사용하지 않습니다.

다중 파티션 데이터베이스에서 고유 색인은 파티션 키의 상위 집합으로 정의된 상태여야 합니다.

사용자 정의 확장 색인 유형 작성

사용자 정의 색인 유형을 지원하기 위해, DB2 Universal Database는 색인이 작업하는 방법을 구성하는 기본 구성요소에 대해 사용자의 논리를 작성하여 적용하게 합니다. 대체될 수 있는 해당 구성요소는 다음과 같습니다.

- 색인 유지보수. 이것은 색인 컬럼 내용을 색인 키에 맵핑하는 기능을 허용합니다. 그러한 맵핑은 사용자 정의 맵핑 기능을 통해 완료됩니다. 정확하게 하나의 구성 유형 컬럼이 확장 색인에 참여할 수 있습니다. 일반 색인과 달리, 확장 색인을 행당 둘 이상의 색인 항목을 가질 수 있습니다. 행당 다중 색인 항목은 문서에 있는 각 키워드의 각 색인 항목이 있는 오브젝트로서 텍스트 문서를 저장하게 할 수 있습니다.
- 색인 개발. 이것은 응용프로그램 설계자가 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.

또한, 선택적 데이터 필터 기능도 지정될 수 있습니다. 최적화 알고리즘은 사용자 정의 함수(UDF)가 평가되기 전에 폐치된 튜플(tuple)에 대해 필터를 사용합니다.

구조화 유형 또는 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 색인으로 정의된 상태
- INCLUDE 컬럼

색인 유지보수의 세부사항

CREATE INDEX EXTENSION문을 통해 색인의 조작용을 구성하는 두 개의 구성요소를 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 KEY절은 이 색인 확장으로 지원되는 소스 키 컬럼에 대한 구조화 데이터 유형 또는 구별 유형을 지정합니다. 단일 매개변수 이름 및 데이터 유형이 제공되며 소스 키 컬럼과 연관됩니다.

GENERATE KEY USING절은 색인 키를 생성하기 위해 사용되는 사용자 정의 테이블 함수를 지정합니다. 이 함수의 출력은 TARGET KEY절 스펙에 지정되어야 합니다. 이 함수의 출력은 FILTER USING절에서 지정한 색인 필터링 함수의 입력으로서 사용될 수 있습니다.

색인 검색의 세부사항

색인 검색은 검색 인수를 검색 범위에 맵핑합니다.

CREATE INDEX EXTENSION문의 WITH TARGET KEY절은 GENERATE KEY USING절에 지정된 사용자 정의 테이블 함수의 출력인 목표 키 매개변수를 지정합니다. 단일 매개변수 이름 및 데이터 유형이 제공되며 목표 키 컬럼과 연관됩니다. 이 매개변수는 GENERATE KEY USING절의 사용자 정의 테이블 함수의 RETURNS 테이블 컬럼에 대응합니다.

SEARCH METHODS절은 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 방법을 정의합니다. 더욱이,

각 검색 메소드는 특정 검색 범위에 있는 색인 항목이 사용자 정의 스칼라 함수에 의해 자세히 규정될 수 있는 방식을 정의하여 단일 값을 리턴합니다.

- **WHEN절**은 레이블을 검색 메소드와 연관시킵니다. 레이블은 색인 개발 규칙(사용자 정의 함수(UDF)의 **PREDICATES절**에 있음)에 지정된 메소드 이름과 관련된 **SQL 식별자**입니다. 하나 이상의 매개변수 이름과 데이터 유형은 범위 함수 및/또는 색인 필터링 함수에서 인수로서 사용하도록 제공됩니다. **WHEN절**은 **CREATE FUNCTION**문의 **PREDICATES절**이 입력 조회와 일치할 때 최적화 알고리즘이 취할 수 있는 조치를 지정합니다.
- **RANGE THROUGH절**은 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 USING절**은 범위 생성 함수에서 리턴된 색인 항목을 필터하기 위해 사용되는 **CASE 표현식** 또는 사용자 정의 외부 테이블 함수를 지정하는 선택적 방법입니다. 색인 필터 함수 또는 **CASE 표현식**에 의해 리턴된 값이 1이면, 색인 항목에 대응하는 행이 테이블에서 검색됩니다. 리턴되는 값이 1이 아닌 경우, 색인 항목을 버립니다. 이 기능은 두 번째 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.

PREDICATES절은 색인 확장을 검색할 수 있는(그리고 술어의 검색 조건에 대해 선택적 **SELECTIVITY절** 사용) 이 함수를 사용하여 술어를 식별합니다. **PREDICATES절**이 지정되면, 함수는 **NO EXTERNAL ACTION**을 사용한 **DETERMINISTIC**으로서 정의되어야 합니다.

- **WHEN절**은 비교 연산자(=, >, < 및 기타) 및 상수 또는 표현식(**EXPRESSION AS절** 사용)이 있는 술어에서 정의되는 함수의 특정 사용을 소개합니다. 술어가 동일한 비교 연산자와 제공된 상수 또는 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 찾아가기가 술어 평가에서 일치함을 나타냅니다. 이 절은 색인 찾아가기 이후에 데이터베이스가 필터 함수나 원래의 사용자 제공 술어 함수를 적용하지 않도록 알립니다. 색인 찾아가기가 사용되지 않는 경우, 원래의 술어 및 필터 함수가 적용되어야 합니다. `EXACT`절이 사용되지 않는 경우, 원래의 사용자 제공 술어는 색인 찾아가기 이후에 적용됩니다. `EXACT` 술어는 색인 찾아가기가 술어와 동일한 결과를 리턴할 때 유용합니다. 이것은 조회 실행시 색인 찾아가기에서 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 색

인 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. 구조화 유형(형태의 경우)을 정의하십시오. CREATE TYPE문을 사용하여 형태가 supertype 및 nullshape, 포인트, 라인이며, 다각형이 하위 유형인 유형 계층 구조를 정의하십시오. 이것은 유형 모델 공간 엔터티를 구조화했습니다. 예를 들어, 상점의 위치는 포인트이며, 강의 경로는 라인이며, 비즈니스 구역의 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. 기타 모든 속성은 이 시나리오와 관련이 없으므로 무시됩니다.
2. 색인 확장을 작성하십시오.
 - CREATE FUNCTION문을 사용하여 키 변환(gridentry), 범위 생성(gridrange) 및 색인 필터링(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 KEY절은 키 변환의 매개변수 및 데이터 유형을 식별합니다. GENERATE KEY USING절은 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 KEY절은 검색 메소드 정의를 식별합니다. WHEN절은 메소드 이름을 식별합니다. RANGE THROUGH절은 사용되는 색인의 범위를 제한하는 데 사용되는 함수를 식별합니다. FILTER USING절은 결과 색인 값에서 불필요한 항목을 제거하는 데 사용되는 함수를 식별합니다.

주: FILTER USING절은 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)

```

PREDICATES절은 각 WHEN절로 시작되는 하나 이상의 술어를 소개합니다. WHEN절은 상수 또는 EXPRESSION AS절 중 하나가 다음에 오는 비교 연산자를 사용하여 술어에 대한 스펙을 시작합니다. FILTER USING절은 결과 테이블의 추가 필터링을 수행하기 위해 사용할 수 있는 필터 함수를 식별합니다. 이것은 규정하는 행을 판별하기 위해 사용자 정의 술어가 실행되어야 하는 행의 수를 감축하는 정의된 함수(술어에서 사용)의 비용이 저렴한 버전입니다. SEARCH BY INDEX EXTENSION절은 색인 검색이 발생하는 곳을 지정합니다. 색인 검색은 색인을 검색하기 위해 사용될 수 있는 색인 확장의 검색 메소드를 사용하여 규칙 세트를 정의합니다. WHEN KEY절은 검색 규칙을 지정합니다. 검색 규칙은 검색 메소드를 통해 색인 검색을 수행하는 데 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 함수(UDF)는 중심 및 반경(각각 첫 번째와 두 번째 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 95퍼센트의 행을 필터링합니다.

제4장 데이터베이스 변경

이 장은 데이터베이스를 변경하기 전에 고려해야 하는 사항과 데이터베이스 오브젝트를 변경하거나 제거하는 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』
- 214 페이지의 『환경 변수 및 프로파일 레지스트리 변수 변경』
- 214 페이지의 『노드 구성 파일 변경』
- 214 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 정보는 [관리 안내서](#): 계획을 참조하십시오.)

사용권 정보 변경

DB2 제품 사용권 관리의 일부로서, 사용권 수를 늘려야 하는 필요성을 느끼게 됩니다. 제어 센터에 있는 사용권 센터를 사용하여 설치된 제품의 사용량을 검사하고 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 인스턴스에 있는 데이터베이스 파티션 서버에 대해 이해하고 있어야 합니다.

인스턴스 나열

제어 센터를 사용하여 시스템에서 사용할 수 있는 모든 인스턴스 목록을 확보하려면, 다음을 수행하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 목록 인스턴스를 원하는 데이터베이스를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 추가를 선택하십시오.
3. 새로 고침을 누른 후, 데이터베이스 이름 필드의 끝에서 화살표를 눌러 인스턴스 목록을 참조하십시오.
4. 취소를 누르십시오.

시스템에서 사용할 수 있는 모든 인스턴스 목록을 확보하려면, 다음을 입력하십시오.

```
db2ilist
```

현재 세션에 적용되는 인스턴스를 판별하려면(OS/2 또는 지원되는 Windows 플랫폼에서), 다음을 사용하십시오.

```
set db2instance
```

인스턴스 구성 갱신

db2iupdt 명령을 수행하면, 다음을 수행하여 지정된 인스턴스를 갱신합니다.

- 인스턴스 소유자의 홈 디렉토리 아래에 있는 sqllib 서브디렉토리의 파일을 바꿉니다.
- 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 완료됩니다. 새 데이터베이스 관리 프로그램 구성 파일이 작성되면, 이전 파일은 인스턴스 소유자의 홈 디렉토리 아래에 있는 sqllib 서브디렉토리의 backup 서브디렉토리로 백업됩니다.

db2iupdt 명령은 버전 및 릴리스 서브디렉토리의 instance 서브디렉토리에서 찾을 수 있습니다(정확한 이름은 운영 체제에 따라 다름).

명령은 다음과 같이 사용됩니다.

```
db2iupdt InstName
```

InstName은 인스턴스 소유자의 로그인 이름입니다.

이 명령과 연관된 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 허용합니다. **db2ilist**를 사용하면 존재하는 인스턴스를 표시할 수 있습니다.
- -u FencedID

분리(fenced) 사용자 정의 함수(UDF) 및 저장 프로시저가 실행되는 사용자를 지정합니다. DB2 클라이언트 또는 DB2 SDK(Software Developer's Kit)를 설치하는 경우 이것은 필요하지 않습니다. 기타 DB2 제품의 경우, 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 사용할 수 있습니다.

```
db2iupdt db2inst1
```

인스턴스 제거

제어 센터를 사용하여 인스턴스를 제거하려면, 다음을 수행하십시오.

1. 제거하려는 인스턴스를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 인스턴스 이름을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. **확정** 상자를 누른 후, **확인**을 누르십시오.

명령행을 사용하여 인스턴스를 제거하려면, 다음을 입력하십시오.

```
db2idrop <instance_name>
```

명령행을 사용하여 인스턴스를 제거하는 준비 및 세부사항은 다음과 같습니다.

1. 현재 인스턴스를 사용 중인 모든 응용프로그램을 중지시키십시오.
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 체제에서, INSTHOME/sqllib/sqllib 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 분리(fenced) 저장 프로시저어 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 제거 중인 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 재작성할 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 디렉토리 아래에 있는 `sqllib` 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 재시작해야 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 세부사항은 [관리 안내서: 성능의 『프로세서 추가를 통해 사용자 구성 확장』](#)을 참조하십시오.

데이터베이스 구성 변경

데이터베이스를 변경하기로 계획 중인 경우, 구성 매개변수 값을 검토해야 합니다. 데이터베이스 사용에 기초하여 수행 중인 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 밸런스를 맞추도록 도와줍니다.

주: 매개변수를 수정하려는 경우, 값은 다음과 같을 때까지 갱신되지 않습니다.

- 데이터베이스 매개변수의 경우, 모든 응용프로그램이 연결해제된 후에 데이터베이스에 대한 새로운 첫 번째 연결
- 데이터베이스 관리 프로그램 매개변수의 경우, 인스턴스를 중지시킨 후 다시 시작할 때

대부분의 경우, 성능 구성 마법사에 의해 권장된 값은 기본값보다 더 나은 성능을 제공하는데, 그 이유는 워크로드 및 사용자의 특정 서버에 대한 정보에 근거한 값이기 때문입니다. 그러나 값이 데이터베이스 시스템의 성능을 반드시 최적화한 것은 아니지만, 향상시킬 목적으로 설계되었음을 주의하십시오. 최적화된 성능을 얻기 위해 조정이 가능한 시작 지점으로 생각하십시오.

제어 센터를 사용하여 데이터베이스 구성을 변경하려면, 다음을 수행하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 변경하려는 인스턴스 또는 데이터베이스를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 마법사를 사용한 성능 구성을 선택하십시오.
3. 각 페이지를 누른 후, 필요하면 정보를 변경하십시오.
4. 결과 페이지를 눌러 작업을 검토하고 제시된 구성 매개변수를 적용하십시오.
5. 적용하는 갱신사항을 완료하면, 완료를 누르십시오.

명령행을 사용하여 데이터베이스 구성을 변경하려면, 다음을 입력하십시오.

```
UPDATE DBM CFG FOR <database_alias>
  USING <config_keyword>=<value>
```

하나 이상의 <config_keyword>=<value> 조합을 단일 명령에서 갱신할 수 있습니다. 데이터베이스 관리 프로그램 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 구성 방법에 대한 세부사항은 관리 안내서: 성능의 『벤치마크 테스트』 및 『DB2 구성』을 참조하십시오.

다중 파티션의 경우: 둘 이상의 파티션에 걸쳐 파티션된 데이터베이스가 있으면, 데이터베이스 구성 파일은 모든 데이터베이스 파티션에서 동일해야 합니다. SQL 컴파일러는 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 타스크를 수행해야 합니다. 이 타스크는 이전에 작성된 데이터베이스의 양상을 갱신하거나 제거합니다. 수행할 타스크에는 다음과 같은 것이 있습니다.

- 217 페이지의 『데이터베이스 제거』
- 217 페이지의 『노드 그룹 변경』
- 218 페이지의 『테이블 공간 변경』
- 225 페이지의 『스키마 제거』
- 225 페이지의 『구조 및 내용에서 테이블 수정』
- 244 페이지의 『사용자 정의 구조화 유형 변경』
- 244 페이지의 『입력된 테이블의 행 삭제 및 갱신』
- 244 페이지의 『기존 테이블 이름 바꾸기』
- 245 페이지의 『테이블 제거』
- 248 페이지의 『트리거 제거』
- 248 페이지의 『사용자 정의 함수(UDF), 유형 맵핑 또는 메소드 제거』
- 249 페이지의 『사용자 정의 유형(UDT) 또는 유형 맵핑 제거』
- 250 페이지의 『뷰 변경 또는 제거』
- 252 페이지의 『요약 테이블 제거』
- 254 페이지의 『서버 변경 또는 제거』
- 254 페이지의 『별칭 변경 또는 제거』
- 256 페이지의 『색인, 색인 확장 또는 색인 스펙 제거』
- 257 페이지의 『오브젝트 변경시의 명령문 종속성』

데이터베이스 제거

비록 데이터베이스에 있는 일부 오브젝트를 변경할 수는 있어도, 데이터베이스 자체를 변경할 수는 없습니다. 제거하거나 재작성해야 합니다. 데이터베이스를 제거하게 되면 해당되는 모든 오브젝트, 컨테이너 및 연관된 파일이 삭제되므로 파급 효과가 매우 커집니다. 제거된 데이터베이스는 데이터베이스 디렉토리에서 제거(카탈로그 해제)됩니다.

제어 센터를 사용하여 데이터베이스를 제거하려면, 다음을 수행하십시오.

1. 데이터베이스 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 데이터베이스를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 데이터베이스를 제거하려면, 다음을 수행하십시오.

```
DROP DATABASE <name>
```

다음 명령은 데이터베이스 SAMPLE을 삭제합니다.

```
DROP DATABASE SAMPLE
```

주: SAMPLE 데이터베이스의 시험을 계속하려는 경우, 이를 삭제하면 안됩니다. SAMPLE 데이터베이스를 제거한 후 이를 다시 사용하려면, 재작성해야 합니다.

노드 그룹 변경

관리 안내서: 성능의 『프로세서 추가를 통해 사용자 구성 조정』장에서 노드 그룹 변경에 대한 세부사항을 참조하십시오.

노드를 추가하거나 제거하면, 노드 그룹의 새로운 노드 세트에 현재 데이터를 재분배해야 합니다. 이를 수행하려면, REDISTRIBUTE NODEGROUP 명령을 사용하십시오. 자세한 정보는 관리 안내서: 성능의 『데이터베이스 파티션에서의 데이터 재분배』 및 *Command Reference*를 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 제거하거나 하나를 추가 작성할 수 없으며, 적어도 하나의 시스템 임시 테이블 공간이 항상 있어야 합니다. 기타 시스템 임시 테이블 공간을 작성할 수 있습니다. 또한 테이블 공간을 작성한 후에 페이지 크기 또는 테이블 공간의 extent 크기를 변경할 수 없습니다.

이 절에서는 다음과 같이 테이블 공간을 변경하는 방법에 대해 설명합니다.

- 『DMS 테이블 공간에 컨테이너 추가』
- 219 페이지의 『DMS 테이블 공간에서 컨테이너 수정』
- 221 페이지의 『파티션이 설정된 SMS 테이블 공간에 컨테이너 추가』
- 221 페이지의 『테이블 공간 이름 바꾸기』
- 222 페이지의 『테이블 공간의 상태 전환』
- 222 페이지의 『사용자 테이블 공간 제거』
- 223 페이지의 『시스템 임시 테이블 공간 제거』.

테이블 공간에 대한 설계 정보는 *관리 안내서: 계획*을 참조하십시오.

DMS 테이블 공간에 컨테이너 추가

테이블 공간에 하나 이상의 컨테이너를 추가하여 DMS 테이블 공간(MANAGED BY DATABASE절로 작성된 테이블 공간)의 크기를 증가시킬 수 있습니다.

테이블 공간의 내용이 모든 컨테이너를 통해 재조정됩니다. 재조정은 동안에도 테이블 공간에 대한 액세스는 제한되지 않습니다. 둘 이상의 컨테이너를 추가해야 할 경우, 컨테이너를 동시에 추가해야 합니다.

제어 센터를 사용하여 DMS 테이블 공간에 컨테이너를 추가하려면, 다음을 수행하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 컨테이너를 추가하려는 테이블 공간을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오
3. 추가를 누르고 정보를 완료한 후, 확인을 누르십시오.
4. 테이블 공간이 파티션된 데이터베이스 환경인 경우, 테이블 공간의 성능 매개변수를 변경하려는 경우 고급을 누르십시오.
5. 확인을 누르십시오.

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 000페이지씩)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 조정하여 DMS 테이블 공간(MANAGED BY DATABASE절로 작성된 테이블 공간)에서 컨테이너의 크기를 증가시킬 수 있습니다. 컨테이너의 크기의 새로운 최고 한계를 알면 크기 조정 메소드를 고려해야 합니다. 컨테이너의 현재 크기를 모른다면(개의치 않으면) 확장 메소드를 고려해야 합니다.

명령행을 사용하여 DMS 테이블 공간에서 하나 이상의 컨테이너의 크기를 조정하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
  RESIZE (DEVICE '<path>' <size>)
```

다음 예는 UNIX 기반 시스템의 테이블 공간에서 두 개의 장치 컨테이너(1,000페이지가 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 1,000페이지에서 2,000페이지로 증가합니다. 새로운 컨테이너를 추가하는 것과 마찬가지로, 테이블 공간의 내용이 모든 컨테이너를 통해 재조정됩니다. 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(1,000페이지가 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 1,000페이지에서 2,000페이지로 증가합니다. 새로운 컨테이너를 추가하는 것과 마찬가지로, 테이블 공간의 내용이 모든 컨테이너를 통해 재조정됩니다. 재조정하는 동안에도 테이블 공간에 대한 액세스는 제한되지 않습니다.

테이블 작성 동안 또는 작성 후에 추가되거나, 테이블 공간 작성 후에 확장된 DMS 컨테이너(파일 및 원시 장치 컨테이너 모두)는 프리페처를 통해 병렬로 수행됩니다. 이러한 컨테이너 작성 또는 크기 조정 조작의 병렬 처리에서 증가를 하려면, 시스템에서 수행 중인 프리페처의 수를 증가시킵니다. 병렬에서 완료되지 않는 단 하나의 프로세스는 이 조치의 로깅과 컨테이너 작성의 경우, 컨테이너의 태그 처리입니다.

주: CREATE TABLESPACE 또는 ALTER TABLESPACE문(기존 테이블 공간에 새 컨테이너 추가에 관하여)의 병렬 처리를 최대화하려면, 프리페처의 수가 추가된 컨테이너의 수보다 더 많은지 또는 같은지 확인하십시오.

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 『조회 최적화에서의 테이블 공간 영향』*을 참조하십시오.

파티션이 설정된 SMS 테이블 공간에 컨테이너 추가

명령행을 사용하여 SMS 테이블 공간에 컨테이너를 추가하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
  ADD ('<path>')
  ON NODE (<partition_number>)
```

번호로 지정된 파티션 및 파티션 범위의 모든 파티션(또는 노드)은 테이블 공간이 정의된 노드 그룹에 존재해야 합니다. partition_number는 오직 명시적이거나, 명령문에 대해 정확히 하나의 노드 절 범위 내에 나타납니다.

다음 예에서는 테이블 공간 『plans』가 UNIX 기반 운영 체제에서 사용한 노드 그룹인 파티션 번호 3에 새 컨테이너를 추가하는 방법을 표시합니다.

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

테이블 공간 이름 바꾸기

테이블 공간 내에서 개별 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 사용하십시오. 이전 테이블 공간 이름을 사용하는 경우, 찾지 못합니다. 마찬가지로, ROLLFORWARD DATABASE 명령으로 테이블 공간을 롤 포워드하는 경우, 새로운 이름을 사용하십시오. 이전 테이블 공간 이름을 사용하는 경우, 찾지 못합니다.

테이블 공간의 상태 전환

테이블 공간에 연관된 컨테이너가 액세스 가능한 경우, ALTER TABLESPACE 문의 SWITCH ONLINE 절은 해당 테이블 공간에서 OFFLINE 상태를 제거하는데 사용합니다. 테이블 공간은 제거된 OFFLINE 상태를 가지고 있는 반면, 나머지 데이터베이스는 여전히 위에 있고 사용되고 있습니다.

이 절의 사용에 대한 대체는 데이터베이스로부터 모든 응용프로그램을 연결해제했다가, 데이터베이스에 응용프로그램을 다시 연결하는 것입니다. 이것은 테이블 공간에서 OFFLINE 상태를 제거해 줍니다.

명령행을 사용하여 테이블 공간에서 OFFLINE 상태를 제거하려면, 다음을 입력하십시오.

```
ALTER TABLESPACE <name>
SWITCH ONLINE
```

사용자 테이블 공간 제거

사용자 테이블 공간을 제거하는 경우, 해당 테이블 공간의 모든 데이터를 삭제하고, 컨테이너를 해제하고, 카탈로그 항목을 제거하십시오. 그러면 테이블 공간의 모든 정의된 오브젝트는 제거되거나 올바르지 않은 것으로 표시됩니다.

테이블 공간을 제거하여 빈 테이블 공간에서 컨테이너를 재사용할 수 있지만, 컨테이너를 재사용하기 전에 DROP TABLESPACE 명령을 COMMIT해야 합니다.

단일 사용자 테이블 공간에서 색인 및 LOB 데이터를 비롯한 모든 테이블 데이터가 들어 있는 사용자 테이블 공간을 제거할 수 있습니다. 여러 개의 테이블 공간에 스패되어 있는 사용자 테이블 공간을 제거할 수도 있습니다. 즉, 테이블 데이터를 테이블 공간 하나에 보관하고 색인은 또다른 테이블 공간에 보관하여 모든 LOB를 세 번째 테이블 공간에 보관할 수 있습니다. 단일 명령문에서는 세 가지 테이블 공간을 동시에 모두 제거해야 합니다. 테이블이 들어 있는 모든 스패된 테

이블 공간은 이 단일 명령문의 일부가 되지 않으면, 제거 요청이 실패합니다. 스캔된 테이블 데이터가 들어 있는 테이블 공간 제거 방법에 대한 세부사항은 *SQL 참조서*를 참조하십시오.

제어 센터를 사용하여 사용자 테이블 공간을 제거하려면, 다음을 수행하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 테이블 공간을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 사용자 테이블을 제거하려면, 다음을 입력하십시오.

```
DROP TABLESPACE <name>
```

다음 SQL문은 ACCOUNTING 테이블 공간을 제거합니다.

```
DROP TABLESPACE ACCOUNTING
```

시스템 임시 테이블 공간 제거

또다른 시스템 임시 테이블 공간을 우선 작성하지 않고 시스템 임시 테이블 공간을 제거할 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 제거해야 합니다.

제어 센터를 사용하여 시스템 테이블 공간을 제거하려면, 다음을 수행하십시오.

1. 테이블 공간 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 기타 시스템 임시 테이블 공간에서 테이블 공간 아이콘을 마우스의 오른쪽 버튼을 선택한 후, 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(내용 분할창) 테이블 공간의 목록을 표시하려면 테이블 공간 폴더를 다시 누르십시오.
5. 제거하려는 시스템 임시 테이블 공간을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 누르십시오.
6. 확정 상자를 누른 후, 확인을 누르십시오.

오직 하나의 시스템 임시 테이블 공간을 가진 경우, 삭제하기 전에 또다른 것을 작성해야 합니다. 다음을 입력하여 명령행을 사용하여 완료될 수 있습니다.

```
CREATE SYSTEM TEMPORARY TABLESPACE <name>  
MANAGED BY SYSTEM USING ('<device>')
```

명령행을 사용하여 시스템 테이블 공간을 제거하려면, 다음을 입력하십시오.

```
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 제거할 수 있습니다.

```
DROP TABLESPACE TEMPSPACE1
```

테이블 공간을 제거하여 빈 테이블 공간에서 컨테이너를 재사용할 수 있지만, 컨테이너를 재사용하기 전에 DROP TABLESPACE 명령을 COMMIT해야 합니다.

임시 테이블 공간 제거

해당 테이블 공간에 정의되어 있고, 현재 선언된 테이블 공간이 없는 경우 단지 사용자 임시 테이블 공간을 제거할 수 있습니다. 테이블 공간을 제거할 때, 테이블 공간에 있는 모든 선언된 임시 테이블을 제거하려는 시도는 선언되지 않습니다.

주: 선언된 임시 테이블은 이를 선언한 응용프로그램이 데이터베이스에서 연결해제 되면 내재적으로 제거됩니다.

스키마 제거

스키마를 제거하기 전에, 해당 스키마에 있는 모든 오브젝트가 제거되거나 또다른 스키마로 이동되어야 합니다. DROP문을 시도할 때에는 스키마 이름이 카탈로그에 있어야 하고, 그렇지 않으면 오류가 리턴됩니다.

제어 센터를 사용하여 스키마를 제거하려면, 다음을 수행하십시오.

1. 스키마 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 스키마를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 스키마를 제거하려면, 다음을 입력하십시오.

```
DROP SCHEMA <name>
```

다음 예에서는 스키마 "joeschma"가 제거됩니다.

```
DROP SCHEMA joeschma RESTRICT
```

RESTRICT 키워드는 데이터베이스에서 삭제되는 스키마의 경우 어떠한 오브젝트도 지정된 스키마에 정의될 수 없다는 규칙을 시행합니다.

구조 및 내용에서 테이블 수정

테이블 구조 및 내용 수정에 필요한 TASK에는 다음이 포함됩니다.

- 226 페이지의 『기존 테이블에 컬럼 추가』
- 227 페이지의 『컬럼 정의 수정』
- 228 페이지의 『테이블 또는 뷰에서 행 제거』

- 229 페이지의 『제한조건 변경』
- 235 페이지의 『생성된 컬럼을 기존 테이블에 정의』
- 239 페이지의 『소멸성 테이블 선언』
- 239 페이지의 『파티션 키 변경』
- 240 페이지의 『테이블 속성 변경』
- 243 페이지의 『요약 테이블의 데이터 새로 고침』

테이블용 트리거를 변경할 수 없습니다. 더 이상 적합하지 않은 모든 트리거를 제거해야 하고(248 페이지의 『트리거 제거』 참조), 대체 트리거를 추가해야 합니다(163 페이지의 『트리거 작성』 참조).

기존 테이블에 컬럼 추가

컬럼 정의에는 컬럼 이름, 데이터 유형 및 필요한 제한조건이 포함됩니다.

테이블에 컬럼을 추가할 때 컬럼은 논리적으로 가장 오른쪽에 존재하는 컬럼 정의의 오른쪽에 위치합니다. 새 컬럼이 기존 테이블에 추가될 때에는, 시스템 카탈로그의 테이블 설명만이 수정되므로, 테이블로의 액세스 시간에 즉각 영향을 주지 않습니다. 기존의 레코드는 UPDATE문을 사용하여 수정될 때까지 실제로 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 컬럼 페이지를 검사하고 컬럼에 대한 정보를 완료한 후, 확인을 누르십시오.

명령행을 사용하여 기존 테이블에 컬럼을 추가하려면, 다음을 입력하십시오.

```
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 분할창에 있는 테이블 목록에서 컬럼을 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 컬럼 페이지를 검사하고 컬럼을 선택한 후, **변경**을 누르십시오.
4. 컬럼에 새 바이트 수를 길이에 입력한 후, **확인**을 누르십시오.

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 들어, 다음과 같습니다.

```
ALTER TABLE ALTER COLUMN
  COLNAM1 ADD SCOPE TYPTAB1
```

ALTER TABLE문에 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

테이블 또는 뷰에서 행 제거

행을 삭제하여 테이블 또는 뷰의 내용을 변경하면 됩니다. 뷰에서 행을 삭제하면 뷰의 기초가 되는 테이블에서 해당 행이 삭제됩니다. DELETE문은 다음과 같이 사용됩니다.

- 검색 조건에 따라 선택적으로 판별되는 한 행 이상의 삭제. 이는 검색된 *DELETE* 로 불립니다.
- 커서의 현재 위치에 따라 결정되는 정확한 한 행 삭제. 이는 위치 *DELETE*로 불립니다.

DELETE문은 응용프로그램에 임베드되거나 동적 SQL문으로 발행됩니다.

수정되고 있는 테이블이 참조 제한조건을 통해 기타의 테이블과 관련되어 있다면, 행 삭제의 수행과 함께 고려사항이 있습니다. 식별된 테이블 또는 식별된 뷰의 기본 테이블이 상위인 경우, 삭제하려고 선택된 행은 삭제 규칙인 RESTRICT와의 관계에서 어떠한 종속도 가져서는 안됩니다. 더욱이, DELETE는 삭제 규칙인 RESTRICT와의 관계에서 종속을 가지고 있는 하위 행에 연쇄해서도 안됩니다.

RESTRICT 삭제 규칙으로 삭제 조작을 막지 않으면, 선택된 행이 삭제됩니다. 선택된 행에 있는 종속적인 행에 발생하는 현상에 대한 추가 정보는 SQL 참조서를 참조하십시오.

예를 들어, 테이블(DEPARTMENT)에서 부서(DEPTNO)인 『D11』를 삭제하려면, 다음을 사용하십시오.

```
DELETE FROM department WHERE deptno='D11'
```

다중 행의 DELETE 수행 중에 오류가 발생하면, 테이블에는 아무런 변경도 일어나지 않습니다. 기존의 참조 제한조건에 의해 요청되어 검색 조건 및 모든 조작에 일치하는 모든 행 삭제를 미리 방지하는 오류가 발생하면, 테이블은 어떠한 변경도 되지 않습니다.

적절한 잠금이 이미 존재하지 않으면, 성공적인 DELETE문의 수행 중에 하나 이상의 독점 잠금이 획득됩니다. 잠금은 COMMIT 또는 ROLLBACK문 다음에 해제됩니다. 잠금은 기타의 응용프로그램이 테이블에서 조작을 수행하지 못하도록 합니다.

식별 컬럼 정의 수정

가져오기 또는 로드 조작으로 가져온 테이블을 재작성하려고 하고, 또 테이블에 IDENTITY 컬럼이 있다면, 테이블의 내용을 재작성한 다음 IDENTITY 값을 1에서 생성을 시작하도록 재설정합니다. 이렇게 다시 작성된 테이블에 새 행을 삽입할 때, 다시 1에서 시작하는 IDENTITY 컬럼을 원하지 않습니다. IDENTITY 컬럼에서 중복 값을 원하지 않습니다. 이것이 발생하지 않게 하려면, 다음을 수행해야 합니다.

1. 테이블을 재작성하십시오.
2. MODIFIED BY IDENTITYOVERRIDE절을 사용하여 테이블에 데이터를 로드하십시오. 데이터가 테이블에 로드되지만 식별 값은 행에 대해 생성되지 않습니다.
3. 다음과 같이 조회를 수행하여 IDENTITY 컬럼에 대한 마지막 카운터 값을 구하십시오.

```
SELECT MAX(<IDENTITY column>)
```

이는 테이블의 IDENTITY 컬럼 값이었던 동일 값과 함께 리턴합니다.

4. 다음과 같이 ALTER TABLE문의 RESTART절을 사용하십시오.

```
ALTER TABLE <table name> ALTER COLUMN <IDENTITY column>  
RESTART WITH <last counter value>
```

5. 테이블로 새 행을 삽입하십시오. IDENTITY 컬럼 값이 RESTART WITH절에 지정된 값을 기본으로 생성됩니다.

제한조건 변경

삭제한 다음 제한조건을 변경할 수 있으며 그 자리에 새로운 제한조건을 추가할 수 있습니다. 자세한 정보는 다음을 참조하십시오.

- 『제한조건 추가』
- 233 페이지의 『제한조건 제거』

제한조건에 대한 자세한 정보는 147 페이지의 『제한조건 정의』를 참조하십시오.

제한조건 추가

ALTER TABLE문을 사용하여 제한조건을 추가하십시오. 이 명령문에 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

제한조건에 대한 자세한 정보는 147 페이지의 『제한조건 정의』를 참조하십시오.

고유 제한조건 추가: 고유 제한조건은 기존의 테이블에 추가될 수 있습니다. 제한조건 이름은 ALTER TABLE문에 지정된 다른 제한조건과 동일할 수 없으며, 테이블 내에서 고유해야 합니다(여기에는 정의된 참조 무결성 제한조건의 이름을 포함합니다). 명령문이 수행되기 전에 기존의 데이터가 새로운 조건에 맞는지 검사합니다.

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 키 추가: 대형 테이블에 제한조건을 추가하려면, 테이블을 점검 보류 상태로 놓고, 제한조건을 추가한 후, 테이블에서 위반 행의 통합 목록을 검사하는 것이 훨씬 더 효율적입니다. SET INTEGRITY문을 사용하여 점검 보류 상태를 명시적으로 설정하십시오. 테이블이 상위 테이블인 경우, 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 기본 키 페이지에서, 기본 키로서 하나 이상의 컬럼을 선택한 후, 화살표를 눌러 이동시키십시오.
4. 선택적: 기본 키의 제한조건 이름을 입력하십시오.
5. 확인을 누르십시오.

명령행을 사용하여 기본 키를 추가하려면, 다음을 수행하십시오.

```
ALTER TABLE <name>
ADD CONSTRAINT <column_name>
PRIMARY KEY <column_name>
```

외부 키가 테이블에 추가되면, 다음과 같은 명령문이 들어 있는 패키지와 캐쉬된 동적 SQL은 올바르지 않음으로 표시됩니다.

- 외부 키가 들어 있는 테이블을 삽입하고 갱신하는 명령문
- 상위 테이블을 갱신하거나 삭제하는 명령문

자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.

제어 센터를 사용하여 외부 키를 추가하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 외부 키 페이지에서, 추가를 누르십시오.
4. 외부 키 추가 창에서, 상위 테이블 정보를 지정하십시오.
5. 하나 이상의 컬럼이 외부 키가 되도록 선택한 후, 화살표를 눌러 이동시키십시오.
6. 상위 테이블의 행이 삭제되거나 갱신될 때 종속 테이블에서 수행할 조치를 지정하십시오. 또한 외부 키에 대한 제한조건 이름을 추가할 수도 있습니다.
7. 확인을 누르십시오.

명령행을 사용하여 외부 키를 추가하려면, 다음을 수행하십시오.

```
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 TABLE문에 지정된 기타 제한조건과 동일할 수 없으며, 테이블 내에서 고유해야 합니다(여기에는 정의된 참조 무결성 제한조건의 이름을 포함합니다). 명령문이 수행되기 전에 기존의 데이터가 새로운 조건에 맞는지 검사합니다.

대형 테이블에 제한조건을 추가하려면, 테이블을 점검 보류 상태로 놓고, 제한조건을 추가한 후, 테이블에서 위반 행의 통합 목록을 검사하는 것이 훨씬 더 효율적입니다. SET INTEGRITY문을 사용하여 점검 보류 상태를 명시적으로 설정하십시오. 테이블이 상위 테이블인 경우, 모든 종속 테이블 및 하위 테이블에 대해 점검 보류가 내재적으로 설정됩니다.

테이블 점검 제한조건이 추가되면, 테이블을 삽입하거나 갱신하는 패키지 및 캐쉬된 동적 SQL이 올바르게 작동하지 않음으로 표시됩니다. 자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.

제어 센터를 사용하여 테이블 점검 제한조건을 추가하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 점검 제한조건 페이지에서, 추가를 누르십시오.
4. 점검 제한조건 추가에서, 정보를 완료한 후 확인을 누르십시오.
5. 점검 제한조건 페이지에서, 확인을 누르십시오.

명령행을 사용하여 테이블 점검 제한조건을 추가하려면, 다음을 수행하십시오.

```
ALTER TABLE <name>  
ADD CONSTRAINT <name> (<constraint>)
```

다음 SQL문은 급여와 커미션을 합해 \$25,000를 넘어야 한다는 제한조건을 EMPLOYEE 테이블에 추가합니다.

```
ALTER TABLE EMPLOYEE  
ADD CONSTRAINT REVENUE CHECK (SALARY + COMM > 25000)
```

제한조건 제거

ALTER TABLE문을 사용하여 제한조건을 제거합니다. 이 명령문에 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

제한조건에 대한 자세한 정보는 147 페이지의 『제한조건 정의』를 참조하십시오.

고유 제한조건 제거: ALTER TABLE문을 사용하여 고유 제한조건을 명시적으로 제거할 수 있습니다. 테이블의 모든 고유 제한조건 이름은 SYSCAT.INDEXES 시스템 카탈로그 뷰에 있습니다.

다음 SQL문은 EMPLOYEE 테이블에서 고유 제한조건 NEWID를 제거합니다.

```
ALTER TABLE EMPLOYEE
DROP UNIQUE NEWID
```

이 고유 제한조건을 제거하면, 제한조건에 사용된 모든 패키지 또는 캐쉬된 동적 SQL이 올바르지 않음으로 표시됩니다.

기본 키 및 외부 키 삭제: 제어 센터를 사용하여 기본 키를 제거하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 기본 키 페이지에서, 오른쪽에서 삭제할 기본 키를 선택한 후, 화살표를 눌러 왼쪽에 있는 사용 가능한 컬럼 상자로 이동하십시오.
4. 확인을 누르십시오.

명령행을 사용하여 기본 키를 제거하려면, 다음을 입력하십시오.

```
ALTER TABLE <name>
DROP PRIMARY KEY
```

외부 키 제한조건이 제거되면, 다음과 같은 사항이 들어 있는 패키지와 캐쉬된 동적 SQL문이 올바르지 않음으로 표시됩니다.

- 외부 키가 들어 있는 테이블을 삽입하고 갱신하는 명령문
- 상위 테이블을 갱신하거나 삭제하는 명령문

자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.

제어 센터를 사용하여 외부 키를 제거하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 변경을 선택하십시오.
3. 외부 키 페이지에서, 추가를 누르십시오.
4. 오른쪽에서 제거할 외부 키를 선택한 후, 화살표를 눌러 왼쪽에 있는 사용 가능한 컬럼 상자로 이동하십시오.
5. 외부 키 페이지에서, 확인을 누르십시오.

명령행을 사용하여 외부 키를 제거하려면, 다음을 수행하십시오.

```
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 제거합니다.

```
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 대한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

테이블 점검 제한조건 제거: ALTER TABLE문을 사용하여 테이블 점검 제한 조건을 명시적으로 제거 또는 변경하거나, DROP TABLE문의 결과로서 이를 내재적으로 제거할 수 있습니다.

테이블 점검 제한조건을 제거하면, 테이블에 INSERT 또는 UPDATE 종속성이 있는 모든 패키지과 캐쉬된 SQL문이 올바르지 않음으로 표시됩니다. (자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.) 테이블에 대한 모든 점검 제한조건은 SYSCAT.CHECKS 카탈로그 뷰에 있습니다. 시스템이 생성한 이름을 갖는 테이블 점검 제한조건을 제거하기 전에, SYSCAT.CHECKS 카탈로그 뷰에서 이름을 찾으십시오.

제어 센터를 사용하여 테이블 점검 제한조건을 제거하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 점검 제한조건 페이지에서, 제거할 점검 제한조건을 선택하고 제거를 누른 후, **확인**을 누르십시오.

명령행을 사용하여 테이블 점검 제한조건을 제거하려면, 다음을 수행하십시오.

```
ALTER TABLE <table_name>  
DROP CHECK <check_constraint_name>
```

다음 SQL문은 EMPLOYEE 테이블에서 테이블 점검 제한조건 REVENUE를 제거합니다.

```
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.

생성된 컬럼을 정의하려면, 다음 단계를 수행하십시오.

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. 여기에는 테이블에 대해 수행되는 작업에 근거하여 이 작업을 완료하는 여러 가지 방법이 있습니다.

- 테이블은 매우 크며 작업을 완료할 로그 공간이 충분한지 확신하지 못합니다. 데이터를 로드한 후에 그렇지만 무결성 점검을 다시 on으로 설정하기 전에, 다음을 수행해야 합니다.

```
COMMIT
```

그런 다음, db2gncol 유틸리티를 사용하여 생성된 컬럼을 설정해야 합니다. 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 대소문자가 구별됩니다. commitcount는 로그를 정리하기 위해 각 내부 확약 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, 기타 모든 매개변수가 무시됩니다.

테이블이 점점 보류 상태에 있더라도 테이블은 전체 프로세스에 대해 잠깁니다. 잠금의 이유는 점점 보류 상태에 있는 테이블에 액세스할 수 있는 기타 유틸리티가 있기 때문입니다. 잠금은 다른 유틸리티와 충돌하는 것을 방지합니다.

- 생성된 컬럼을 갱신할 로그 공간이 SET INTEGRITY에 대해 충분할 것입니다. 정상적인 경우에는 충분하지 않습니다. 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 첫 번째 메소드를 선택하지 않습니다. 데이터를 로드한 후에는 그렇지만 무결성 점검을 다시 on으로 설정하기 전에, 다음을 수행해야 합니다.

- a. 테이블에서 독점 잠금을 가져오십시오. 이것은 미확약 읽기(UR) 트랜잭션을 제외하고 모두 테이블을 액세스하지 못하게 합니다.

```
LOCK TABLE t1
```

- b. 검사되지 않은 데이터를 사용하여 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 완료함으로써 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 GENERATED
```

- c. 트랜잭션을 확약하여 처음에 로그되지 않는 옵션을 설정해제하십시오.

```
COMMIT
```

생성된 컬럼의 값은 표현식을 등식 점검 제한조건인 것처럼 적용하여 간단히 검사할 수 있습니다.

SET INTEGRITY FOR t1 IMMEDIATE CHECKED

예를 들어, **LOAD**를 사용하여 생성된 컬럼에 값이 위치하며, 값이 생성된 표현식과 일치함을 아는 경우, 값을 검사하거나 지정하지 않고 점검 보류 상태에서부터 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. 구조화 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 상속하지 않습니다.

생성된 컬럼은 키워드 **DEFAULT** 없이 삽입되거나 갱신될 수 없습니다. 삽입할 때, **DEFAULT**를 사용하면 컬럼 목록에 컬럼을 열거할 필요가 없습니다. 대신, 생성된 컬럼은 값 목록에 있는 **DEFAULT**로 설정될 수 있습니다. 갱신할 때, **DEFAULT**는 검사하지 않고도 **SET INTEGRITY**에 의해 온라인으로 위치된 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 사용시, 목표 테이블은 맵핑에서 생성된 컬럼을 사용하지 말아야 합니다. 복제할 때 두가지 선택사항이 있습니다.

- 목표 테이블은 일반 컬럼으로서 생성된 컬럼을 정의해야 합니다.
- 목표 테이블은 맵핑에서 생성된 컬럼을 생략해야 합니다.

생성된 컬럼으로 작업할 때 몇 가지 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.

소멸성 테이블 선언

소멸성 테이블은 런타임 크기가 공백에서 대형까지 변할 수 있는 내용을 가진 테이블로 정의됩니다. 이 테이블의 소멸성 또는 극도의 변경 가능성은 **RUNSTATS**로 수집된 통계 사용을 부정확하게 만듭니다. 통계는 특정 시점에서 수집되고 특정 시점만을 반영합니다. 소멸성 테이블을 사용하는 액세스 플랜을 생성하면, 이는 부정확하거나 성능이 떨어지는 결과를 낳습니다. 예를 들어, 소멸성 테이블이 비어 있을 때 통계가 모아지면, 최적화 알고리즘은 색인 스캔보다는 테이블 스캔을 사용하여 소멸성 테이블 액세스를 선호하는 경향이 있습니다.

이를 방지하려면, **ALTER TABLE**문을 사용하여 테이블을 소멸성(**volatile**)으로 선언해야 합니다. 테이블을 소멸성으로 선언하면, 최적화 알고리즘은 테이블 스캔보다 색인 스캔 사용을 고려합니다. 선언된 소멸성 테이블을 사용하는 액세스 플랜은 해당 테이블에 대해 기존의 통계에 의존하지 않습니다.

제어 센터를 사용하여 테이블을 소멸성으로 선언하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 테이블 페이지에서, 런타임시에 기본 행 수 변동 선택란을 선택한 후 **확인**을 누르십시오.

명령행을 사용하여 테이블을 『소멸성』으로 선언하려면, 다음을 수행하십시오.

```
ALTER TABLE <table_name>
VOLATILE CARDINALITY
```

파티션 키 변경

단일 파티션 노드 그룹에 있는 테이블의 파티션 키만 변경할 수 있습니다. 우선 기존의 파티션 키를 제거한 다음, 또다른 파티션 키를 작성하십시오.

다음 SQL문은 **MIXREC** 테이블에서 파티션 키 **MIX_INT**를 제거합니다.

ALTER TABLE MIXREC DROP PARTITIONING KEY

자세한 정보는 *SQL 참조서 매뉴얼*의 ALTER TABLE문을 참조하십시오.

다중 파티션 노드 그룹의 테이블에 있는 파티션 키는 변경할 수 없습니다. 이를 제거하려고 할 경우, 오류가 표시됩니다.

다중 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 제거하며, 테이블을 재작성하고 파티션 키를 재정의한 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 여유 공백 백분율(PCTFREE), 잠금 크기 또는 추가 모드와 같은 테이블 속성을 변경할 경우가 있습니다.

테이블의 각 페이지에 남아 있는 여유 공간의 양은 PCTFREE를 통해 지정되며, 클러스터링 색인의 효과적인 사용을 위한 주요 고려사항입니다. 지정할 양은 기존 데이터 및 예상 장래 데이터의 특성에 따라 달라집니다. PCTFREE는 LOAD 및 REORG로 고려되지만, 삽입, 갱신 및 가져오기 활동에 의해 무시됩니다.

PCTFREE를 더 큰 값에 설정하면 더 긴 기간 동안 클러스터링을 유지보수하지만, 추가 디스크 공간이 필요합니다.

LOCKSIZE 매개변수를 사용하여 테이블이 액세스될 때 사용되는 잠금의 크기(세분성)를 지정할 수 있습니다. 기본적으로, 테이블이 작성될 때 행 레벨 잠금이 정의됩니다. 테이블 레벨 잠금의 사용은 취득하거나 해제하는 데 필요한 잠금 수를 제한하여 조회의 성능을 향상시킬 수 있습니다.

APPEND ON을 지정하여, 테이블의 전체 성능을 향상시킬 수 있습니다. 여유 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.

식별 컬럼 변경

ALTER TABLE문으로 기존 식별 컬럼의 속성을 수정하십시오. 이 명령문(구문을 포함한)에 대한 자세한 정보는 *SQL* 참조서를 참조하십시오.

속성 컬럼이 일부의 순차 특성을 가지도록 수정하는 데에는 몇 가지 방법이 있습니다.

다음과 같이 ALTER TABLE문과 식별 컬럼에 대해 고유한 일부 타스크가 있습니다.

- RESTART는 식별 컬럼과 연관된 순차를 식별 컬럼이 원래 작성될 때 시작값으로서 내재적으로나 명시적으로 지정된 값으로 재설정합니다.
- RESTART WITH <numeric-constant>는 식별 컬럼과 연관된 순차를 정확한 숫자 상수 값으로 재설정합니다. 숫자 상수는 식별 컬럼에 지정되는 소수점의 오른쪽에 0이 아닌 숫자가 하나도 없는 임의의 양수 또는 음수 값입니다.

순차 변경

ALTER SEQUENCE문으로 기존 순차의 속성을 수정하십시오. 이 명령문(구문을 포함한)에 대한 자세한 정보는 *SQL* 참조서를 참조하십시오.

수정할 수 있는 순차 속성에는 다음이 포함됩니다.

- 장래 값 사이의 증분 변경
- 새로운 최소값 또는 최대값 설정
- 캐쉬 순차 번호의 수 변경
- 순차가 순환할 것인지 아닌지의 변경
- 순차 번호가 요청 순서대로 생성되어야 하는지의 변경
- 순차 재시작

순차 작성의 일부가 아닌 두 개의 타스크가 있습니다. 타스크는 다음과 같습니다.

- RESTART. 식별 컬럼이 작성될 때 시작값으로서 내재적으로나 명시적으로 지정된 값에 순차를 재설정합니다.

- **RESTART WITH numeric-constant.** 정확한 숫자 상수 값에 순차를 재설정합니다. 숫자 상수는 소수점의 오른쪽에 0이 아닌 숫자가 하나도 없는 임의의 양수 또는 음수 값입니다.

순차를 재시작하거나 CYCLE로 변경한 후, 중복 순차 번호를 생성할 수 있습니다. 장래 순차 번호만은 ALTER SEQUENCE문의 영향을 받습니다.

순차의 데이터 유형은 변경할 수 없습니다. 대신, 현재 순차를 제거한 다음, 새로운 데이터 유형을 지정하는 순차를 새로 작성해야 합니다.

순차 변경시, DB2가 사용하지 않는 캐쉬 순차 값은 모두 잃게 됩니다.

순차 제거

순차를 삭제하려면, DROP문을 사용하십시오. 이 명령문(구문을 포함한)에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

특정 순차는 다음을 사용하여 제거할 수 있습니다.

```
DROP SEQUENCE sequence_name
```

여기서는 `sequence_name`이 제거할 순차의 이름이고, 기존 순차를 정확히 식별하기 위해 내재적 또는 명시적 스키마 이름을 포함합니다.

IDENTITY 컬럼에 대해 시스템이 생성하는 순차는 DROP SEQUENCE문을 사용하여 제거할 수 없습니다.

순차를 한번만 제거하면, 순차의 모든 특권도 제거됩니다.

요약 테이블 등록 정보 변경

일부 제한사항이 있는 경우, 요약 테이블을 일반 테이블로 변경하거나 일반 테이블을 요약 테이블로 변경할 수 있습니다. 기타 테이블 유형을 변경할 수 없으며, 오직 일반 및 요약 테이블만을 변경할 수 있습니다. 예를 들어, 복제된 요약 테이블을 일반 테이블로 변경할 수 없으며, 역방향으로도 변경할 수 없습니다.

일반 테이블이 요약 테이블로 변경되면, 테이블은 점검 보류 상태가 됩니다. 이러한 방법으로 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 방법으로 테이블을 변경하도록 허용되지 않습니다.

일반 테이블을 요약 테이블로 변경할 때, 요약 테이블 정의의 fullselect는 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.

CREATE SUMMARY TABLE문에 대한 추가 정보는 SQL 참조서를 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 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

사용자 정의 구조화 유형 변경

구조화 유형을 작성한 후, 해당 구조화 유형과 연관된 속성을 추가하거나 제거해야 하는 경우가 있습니다. ALTER TYPE(구조화)문을 사용하여 이 작업을 수행할 수 있습니다. 구조화 유형에 필요한 모든 정보는 *응용프로그램 개발 안내서*를 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. 입력된 테이블에 필요한 모든 정보는 *응용프로그램 개발 안내서*를 참조하십시오.

기존 테이블 이름 바꾸기

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. 이름을 바꾸려는 기존의 테이블은 테이블 또는 별명 이외의 카탈로그 테이블, 요약 테이블, 입력된 테이블 또는 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 유효하지 않습니다. 마지막으로, 원래의 테이블을 언급하는 모든 별명은 수정되지 않습니다.

해당 시스템 카탈로그 테이블을 검사하여 이름 바꾸기 중인 테이블이 이들 제한사항에 의해 영향받지 않도록 해야 합니다.

패키지가 방금 이름 바뀌기된 테이블을 언급할 경우에는 리바인드되어야 합니다. 다음과 같은 경우 패키지는 내재적으로 리바인드될 수 있습니다.

- 다른 테이블이 원래의 테이블 이름을 사용하여 이름 바뀌기될 경우
- 별명 또는 뷰가 원래의 테이블 이름을 사용하여 작성될 경우

내재적 또는 명시적 리바인드가 수행되기 전에, 이 두 선택항목 중 하나가 완료되어야 합니다. 선택이 수행되지 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 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 이름 바꾸기를 선택하십시오.
3. 새 테이블 이름을 입력한 후, 확인을 누르십시오.

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 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

테이블 제거

테이블은 DROP TABLE SQL문으로 제거될 수 있습니다.

테이블이 제거되면, 제거된 테이블에 대한 정보가 들어 있는 SYSCAT.TABLES 카탈로그의 행과, 테이블에 의존하는 기타 오브젝트는 영향을 받습니다. 예를 들어, 다음과 같습니다.

- 모든 컬럼 이름이 제거됩니다.

- 테이블의 컬럼에 대해 작성된 색인이 제거됩니다.
- 테이블에 근거한 모든 뷰가 작동 불능 상태로 표시됩니다. (자세한 정보는 251 페이지의 『작동 불능 뷰 복구』를 참조하십시오.)
- 제거된 테이블과 종속 뷰에 대한 모든 특권도 내재적으로 권한 취소됩니다.
- 상위 테이블 또는 종속 테이블에 있는 모든 참조 제한조건이 제거됩니다.
- 제거된 테이블에 종속된 모든 패키지 및 캐쉬된 동적 SQL문은 올바르지 않음으로 표시되며 종속 오브젝트가 재작성될 때까지 이 상태로 남아 있습니다. 계층 구조에서 제거 중인 하위 테이블 위에 있는 모든 상위 테이블에 종속된 패키지도 여기에 포함됩니다. (자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 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, 파일을 기타 조작용으로 즉시 사용할 수 없다는 의미입니다.

제어 센터를 사용하여 테이블을 삭제하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 테이블을 제거하려면, 다음을 입력하십시오.

```
DROP TABLE <table_name>
```

다음 명령문은 DEPARTMENT라고 하는 테이블을 제거합니다.

```
DROP TABLE DEPARTMENT
```


각각의 테이블은 하위테이블이 있을 때는 제거될 수 없습니다. 그러나 테이블 계층 구조에 있는 모든 테이블은 다음 예에서처럼 단일 DROP TABLE HIERARCHY문으로 제거될 수 있습니다.

```
DROP TABLE HIERARCHY person
```

DROP TABLE HIERARCHY문은 제거될 계층 구조의 루트 테이블을 이름 지정해야 합니다.

특정 테이블을 제거하는 것과 테이블 계층 구조를 제거하는 것에는 차이가 있습니다.

- DROP TABLE HIERARCHY는 개별 DROP 테이블 명령문으로 활성화되는 삭제된 트리거를 활성화시키지 않습니다. 예를 들어, 개별 하위테이블을 제거하면, 상위테이블에 대한 삭제 트리거가 활성화됩니다.
- DROP TABLE HIERARCHY는 제거된 테이블의 개별 행에 대해 로그 항목을 작성하지 않습니다. 대신 계층 구조 제거는 단일 이벤트로 로그됩니다.

DROP문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

사용자 정의 임시 테이블 제거

DECLARE GLOBAL TEMPORARY TABLE문을 사용하여 작성된 사용자 정의 임시 테이블을 제거할 때 알아야 할 몇 가지 고려사항이 있습니다.

이러한 테이블을 제거할 때, 테이블 이름은 스키마 이름 SESSION으로 규정되어야 하며 테이블을 작성한 응용프로그램에 있어야 합니다.

패키지는 이 테이블 유형에 종속될 수 없으므로 이러한 테이블을 제거할 때 유효합니다.

사용자 정의 임시 테이블이 제거되고, 임시 테이블 작성이 사용 중인 작업 단위(UOW) 또는 저장 지점을 선행한 경우, 테이블은 기능적으로 삭제되어 응용프로그램이 테이블에 액세스할 수 없습니다. 그러나 테이블은 여전히 테이블 공간에 예약된 일부 공간을 가지며 이것은 작업 단위(UOW)가 확장되거나 저장 지점이 종료될 때까지 사용자 임시 테이블 공간을 제거하지 못하게 합니다.

DROP문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

트리거 제거

트리거 오브젝트는 DROP문을 사용하여 제거될 수 있지만, 이 프로시저어로 인해 종속 패키지는 다음과 같이 올바르지 않음으로 표시됩니다.

- 명시적 컬럼 목록 없이 갱신 트리거가 제거될 경우, 목표 테이블에서 갱신 사용이 있는 패키지가 유효하지 않음으로 표시됩니다.
- 컬럼 목록과 함께 갱신 트리거가 제거될 경우, 패키지가 CREATE TRIGGER 문의 column-name 목록 내의 적어도 하나의 컬럼에서 갱신 사용이 있을 경우에만 목표 테이블에 갱신 사용이 있는 패키지가 유효하지 않음으로 표시됩니다.
- 삽입 트리거가 제거될 경우, 목표 테이블에서 삽입 사용이 있는 패키지가 유효하지 않음으로 표시됩니다.
- 삭제 트리거가 제거될 경우, 목표 테이블에서 삭제 사용이 있는 패키지가 유효하지 않음으로 표시됩니다.

패키지는 응용프로그램이 명시적으로 바인드 또는 리바인드될 때까지 올바르지 않음으로 있거나, 패키지가 수행되면 데이터베이스 관리 프로그램이 자동으로 이를 리바인드합니다.

사용자 정의 함수(UDF), 유형 맵핑 또는 메소드 제거

사용자 정의 함수(UDF), 함수 템플릿 또는 함수 맵핑은 DROP문으로 제거될 수 있습니다.

DISABLE 맵핑 옵션으로 함수 맵핑을 사용할 수 없게 할 수 있습니다. 수행 방법에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

뷰, 트리거, 테이블 점검 제한조건 또는 또다른 UDF가 이에 종속적인 경우에는, UDF가 제거될 수 없습니다. CREATE DISTINCT TYPE문에 의해 내재적으로 생성된 함수는 제거될 수 없습니다. SYSIBM 스키마 또는 SYSFUN 스키마에 있는 함수는 제거할 수 없습니다.

기타 오브젝트는 함수 또는 함수 템플릿에 종속될 수 있습니다. 함수 맵핑을 비롯한 이러한 모든 종속성은 함수가 제거되기 전에 제거되어야 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 대한 자세한 정보는

Command Reference 매뉴얼을 참조하십시오. UDF를 제거하면, 이것을 사용한 다른 패키지 또는 캐쉬된 동적 SQL문이 유효하지 않음으로 표시됩니다.

함수 매핑을 제거하면, 패키지는 유효하지 않음으로 표시됩니다. 자동 리바인드가 일어나며 최적화 알고리즘은 지역 함수를 사용하려고 시도합니다. 지역 함수가 템플릿인 경우, 내재적 리바인드는 실패합니다.

(자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.)

사용자 정의 유형(UDT) 또는 유형 매핑 제거

DROP문을 사용하여 사용자 정의 유형(UDT) 또는 유형 매핑을 제거할 수 있습니다. UDT가 다음에 사용될 때는 제거할 수 없습니다.

- 기존 테이블 또는 뷰(구별 유형)에 대한 컬럼 정의에 사용
- 기존의 입력된 테이블 또는 입력된 뷰(구조화 유형)의 유형으로 사용
- 또다른 구조화 유형의 상위 유형으로 사용

기본 유형 매핑을 제거할 수 없습니다. 또다른 유형 매핑을 작성하여 이를 겹쳐쓰기만 할 수 있습니다.

데이터베이스 관리 프로그램은 이러한 구별 유형에 종속적인 모든 함수 제거를 시도할 것입니다. UDF가 제거될 수 없으면, UDT도 제거될 수 없습니다. 뷰, 트리거, 테이블 점검 제한조건 또는 또다른 UDF가 이에 종속적인 경우에는, UDF가 제거될 수 없습니다. UDT를 제거하면, 이것을 사용한 다른 패키지 또는 캐쉬된 동적 SQL문이 유효하지 않음으로 표시됩니다.

UDT에 대한 변환을 작성하고 UDT를 제거할 계획이라면, 변환을 제거해야 하는지 고려해야 합니다. 이는 DROP TRANSFORM문을 통해 완료됩니다. 이 명령문에 대한 세부사항은 *SQL* 참조서를 참조하십시오. 사용자 또는 기타 응용프로그램 개발자가 정의한 변환만 제거될 수 있다는 점에 유의하십시오. 내장된 변환과 연관된 그룹 정의는 제거될 수 없습니다.

사용자 정의 유형에 대한 자세한 정보는 *SQL* 참조서 및 *응용프로그램 개발 안내서* 매뉴얼을 참조하십시오.

뷰 변경 또는 제거

ALTER VIEW문은 참조 유형 컬럼이 범위를 추가하도록 변경하여 기존 뷰를 수정합니다. 뷰에 대한 다른 모든 변경을 수행할 때에는 해당 뷰를 제거한 다음, 재작성해야 합니다.

뷰를 변경할 때에는 아직 범위가 정의되지 않은 기존 참조 유형 컬럼에 범위를 추가해야 합니다. 그리고 상위 뷰로부터 컬럼을 상속할 수 없습니다.

ALTER VIEW문의 컬럼 이름의 데이터 유형은 REF(입력된 테이블 이름 또는 입력된 뷰 이름의 유형)여야 합니다.

패키지 및 캐쉬된 동적 명령문에 올바르게 표시되지 않은 표시가 되어 있더라도, 테이블 및 색인과 같은 기타 데이터베이스 오브젝트에는 영향을 주지 않습니다. 자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.

ALTER VIEW문에 대한 추가 정보는 *SQL* 참조서를 참조하십시오.

제어 센터를 사용하여 뷰를 변경하려면, 다음을 수행하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 수정하려는 뷰를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **변경**을 선택하십시오.
3. 뷰 **변경** 창에서 주석을 입력하거나 수정한 후, **확인**을 누르십시오.

명령행을 사용하여 뷰를 변경하려면, 다음을 입력하십시오.

```
ALTER VIEW <view_name> ALTER <column name>  
ADD SCOPE <typed table or view name>
```

제어 센터를 사용하여 뷰를 제거하려면, 다음을 수행하십시오.

1. 뷰 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 뷰를 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 **제거**를 선택하십시오.
3. **확정** 상자를 누른 후, **확인**을 누르십시오.

명령행을 사용하여 뷰를 제거하려면, 다음을 수행하십시오.

```
DROP VIEW <view_name>
```

다음 예에서는 EMP_VIEW를 제거하는 방법을 보여줍니다.

```
DROP VIEW EMP_VIEW
```

제거될 뷰에 종속적인 모든 뷰가 작동 불능 상태로 됩니다. (자세한 정보는 『작동 불능 뷰 복구』를 참조하십시오.)

테이블 계층 구조의 경우, 다음 예에서처럼 루트 뷰를 이름 지정하여 하나의 명령 문에서 전체 뷰 계층 구조를 제거할 수 있습니다.

```
DROP VIEW HIERARCHY VPerson
```

뷰 제거 및 작성에 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

작동 불능 뷰 복구

뷰는 다음 경우 작동 불능이 될 수도 있습니다.

- 기본 테이블의 특권을 권한 취소한 경우
- 테이블, 별명 또는 함수가 제거된 경우
- 상위 뷰가 작동 불능인 경우
- 종속인 뷰가 제거될 경우

다음 단계는 작동 불능 뷰를 복구하는 데 도움이 됩니다.

1. 뷰를 작성하기 위해 초기에 사용된 SQL문을 판별하십시오. SYSCAT.VIEW 카탈로그 뷰의 TEXT 컬럼으로부터 이 정보를 얻을 수 있습니다.
2. 동일한 뷰 이름과 동일한 정의로 CREATE VIEW문을 사용하여 뷰를 재작성 하십시오.
3. GRANT문을 사용하여 뷰에 대해 이전에 권한 부여된 모든 특권을 다시 권한 부여하십시오. (작동 불능 뷰에 권한 부여된 특권이 모두 권한 취소됨을 유의 하십시오.)

작동 불능 뷰를 복구하지 않을 경우, DROP VIEW문으로 작동 불능 뷰를 명시 적으로 제거하거나 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 제거

요약 테이블을 변경할 수는 없지만, 제거할 수는 있습니다.

테이블을 참조하는 모든 색인, 기본 키, 외부 키 및 점검 제한조건이 제거됩니다. 테이블을 참조하는 모든 뷰 및 트리거는 사용 불가능 상태가 됩니다. 제거되거나 작동 불능 상태로 표시된 오브젝트에 종속되는 모든 패키지는 유효하지 않으므로 표시됩니다. 패키지 종속성에 대한 자세한 정보는 257 페이지의 『오브젝트 변경시의 명령문 종속성』을 참조하십시오.

제어 센터를 사용하여 요약 테이블을 제거하려면, 다음을 수행하십시오.

1. 테이블 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 요약 테이블을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 요약 테이블을 제거하려면, 다음을 입력하십시오.

```
DROP TABLE <table_name>
```

다음 SQL문은 요약 테이블 XT를 제거합니다.

```
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 판별하십시오. 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. (작동 불능 요약 테이블에 권한 부여된 특권이 모두 권한 취소됨을 유의하십시오.)

작동 불능 요약 테이블을 복구하지 않으려는 경우, DROP TABLE문을 사용하여 작동 불능 요약 테이블을 명시적으로 제거하거나 정의는 다르지만 동일한 이름을 사용하여 새 요약 테이블을 작성할 수 있습니다.

작동 불능 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 제거

DROP문은 데이터베이스에서 래퍼를 제거할 수 있습니다. 다음 예에서는 DRDA 래퍼를 제거하는 방법을 보여줍니다.

```
DROP WRAPPER DRDA
```

래퍼에서 종속인 서버 정의, 사용자 정의 함수 맵핑 및 사용자 정의 데이터 유형 맵핑은 모두 제거됩니다. 제거된 서버 정의에서 종속인 사용자 정의 맵핑, 별칭, 사용자 정의 유형 맵핑 및 사용자 맵핑 또한 모두 제거됩니다. 제거된 별칭에서 종속인 임의의 색인 스펙은 제거되고 이들 별칭에 종속인 뷰는 작동 불능으로 표시됩니다. 제거된 오브젝트에서 종속인 패키지 및 작동 불능 뷰는 모두 유효하지 않습니다.

DROP 래퍼에 대해 SYSADM 또는 DBADM 권한 중 하나를 가지고 있어야 합니다.

래퍼 제거에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

서버 변경 또는 제거

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 제거될 수 있습니다. 다음 예에서는 ORALOC01 서버를 제거하는 방법을 보여줍니다.

```
DROP SERVER ORALOC01
```

데이터 소스에 상주하는 테이블 및 뷰에 대한 별칭은 모두 제거됩니다. 이들 별칭에 종속인 임의의 색인 스펙은 제거됩니다. 제거된 서버 정의에서 종속인 임의의 사용자 정의 함수(UDF) 맵핑, 사용자 정의 유형 맵핑 및 사용자 맵핑 또한 제거됩니다. 제거된 서버 정의에서 종속인 모든 패키지, 함수 맵핑, 별칭 및 색인 스펙은 유효하지 않습니다.

ALTER 또는 DROP 서버에 대해 SYSADM 또는 DBADM 권한 중 하나를 가지고 있어야 합니다.

서버 제거 및 변경에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

별칭 변경 또는 제거

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 맵핑할 수

있습니다. 이 명령문을 사용하여 컬럼 옵션을 추가할 수도 있습니다. ALTER NICKNAME 구문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

별칭이 제거되면, 해당 별칭에 작성된 뷰가 작동 불능으로 표시됩니다. 별칭이 뷰에서 참조되면, 별칭 컬럼 이름 또는 데이터 유형을 변경할 수 없습니다.

이 명령문을 사용하려면, SYSADM 또는 DBADM 권한 중 하나를 보류하거나, 별칭에 대해 CONTROL 또는 ALL 데이터베이스 특권이 있거나, ALTERIN(현재 스키마에 대해) 스키마 특권이 있거나, 연합 데이터베이스에서 별칭 정의자여야 합니다.

별칭 컬럼 변경 및 별칭 제거

다음 예에서는 컬럼의 지역 이름을 COL1에서 NEWCOL로 변경하여 별칭 TESTNN을 변경하는 방법을 보여줍니다.

```
ALTER NICKNAME TESTNN ALTER COLUMN COL1 LOCAL NAME NEWCOL
```

다음 예에서는 별칭 TESTNN을 제거하는 방법을 보여줍니다.

```
DROP NICKNAME TESTNN
```

별칭 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 설정

옵션	유효 설정	기본 설정
numeric_string	<p>‘Y’ 예, 이 컬럼은 숫자 데이터로만 이루어진 문자열을 포함합니다. 중요: 이 컬럼이 뒤 공백을 가진 숫자 문자열을 포함할 경우, ‘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 대립될 수 있는 공백을 포함하지 않음을 알려주십시오. 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 설정 (계속)

옵션	유효 설정	기본 설정
varchar_no_trailing_blanks	<p>특정 VARCHAR 컬럼에서 뒤 공백을 제거할지 여부를 나타냅니다.</p> <p>‘Y’ 예, 이 VARCHAR 컬럼에서 뒤 공백이 제거됩니다.</p> <p>‘N’ 아니오, 이 VARCHAR 컬럼에서 뒤 공백을 제거하지 않습니다.</p> <p>데이터 소스 VARCHAR 컬럼에 채워진 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 이들 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 <i>SQL 참조서 참조</i>).</p>	‘N’

색인, 색인 확장 또는 색인 스펙 제거

색인 정의, 색인 확장 또는 색인 스펙의 절을 변경할 수 없습니다. 색인 또는 색인 확장을 제거한 후 재작성해야 합니다. (색인 또는 색인 스펙을 제거해도 다른 오브젝트가 제거되지는 않지만, 일부 패키지는 유효하지 않음으로 표시될 수 있습니다.)

제어 센터를 사용하여 색인, 색인 확장 또는 색인 스펙을 제거하려면, 다음을 수행하십시오.

1. 색인 폴더를 찾을 때까지 오브젝트 트리를 확장하십시오.
2. 제거하려는 색인을 마우스 오른쪽 단추로 누른 후, 팝업 메뉴에서 제거를 선택하십시오.
3. 확정 상자를 누른 후, 확인을 누르십시오.

명령행을 사용하여 색인, 색인 확장 또는 색인 스펙을 제거하려면, 다음을 입력하십시오.

```
DROP INDEX <index_name>
```

다음 SQL문은 PH라고 하는 색인을 제거합니다.

```
DROP INDEX PH
```

다음 SQL문은 IX_MAP이라고 하는 색인 확장을 제거합니다.

```
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 시행합니다. 기본 색인이 이 색인 확장에 종속되는 경우, 제거에 실패합니다.

기본 키 또는 고유 색인 키(색인 스펙이 아닌 한)는 명시적으로 제거될 수 없습니다. 다음 방법 중 하나를 사용하여 이를 제거해야 합니다.

- 1차 색인 또는 고유 제한조건이 기본 키 또는 고유 키를 사용하여 자동으로 작성된 경우, 기본 키 또는 고유 키를 제거하면 색인이 제거됩니다. 제거는 ALTER TABLE문을 사용하여 완료됩니다.
- 1차 색인 또는 고유 제한조건이 사용자 정의된 경우, 기본 키 또는 고유 키는 ALTER TABLE문을 사용하여 먼저 제거되어야 합니다. 기본 키 또는 고유 키가 제거된 후, 색인은 더 이상 1차 색인 또는 고유 색인으로 간주되지 않고, 명시적으로 제거될 수 있습니다.

제거된 색인에 의존하는 모든 패키지 및 캐쉬된 동적 SQL문은 올바르지 않음으로 표시됩니다. 자세한 정보는 『오브젝트 변경시의 명령문 종속성』을 참조하십시오. 응용프로그램은 색인의 추가 또는 제거로 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 완전한 목록은 SQL 참조서를 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 위반되지 않도록 하기 위해 검사되어야 하는 종속 테이블이 여기에 해당됩니다. 패키지는 패키지 작성자에게 권한 부여된 특권에도 종속됩니다.

패키지 또는 캐쉬된 동적 SQL문이 오브젝트에 의존하고 해당 오브젝트가 제거되면, 패키지 또는 캐쉬된 동적 SQL문은 "올바르지 않은" 상태가 됩니다. 패키지가 사용자 정의 함수(UDF)에 의존하고 해당 함수가 제거되면, 패키지는 "작동 불능" 상태에 놓입니다.

올바르지 않은 상태에 있는 캐쉬된 동적 SQL문은 다음에 사용될 때 자동으로 다시 최적화됩니다. 명령문에서 필요한 오브젝트가 제거되면, 동적 SQL문의 실행은 오류 메시지와 함께 실패하게 됩니다.

올바르지 않은 상태에 있는 패키지는 다음에 사용될 때 내재적으로 리바인드됩니다. 이러한 패키지 또한 명시적으로 리바인드될 수 있습니다. 트리거가 제거되었기 때문에 패키지가 올바르게 않은 것으로 표시된 경우, 리바인드 패키지는 더 이상 트리거를 호출하지 않습니다.

작동 불능 상태에 있는 패키지는 사용되기 전에 명시적으로 리바인드되어야 합니다. 패키지 바인드 및 리바인드에 대한 자세한 정보는 응용프로그램 개발 안내서를 참조하십시오.

연합 데이터베이스 오브젝트에는 비슷한 종속성이 있습니다. 예를 들어, 서버를 제거하면, 해당 서버에 연관된 별칭을 참조하는 패키지 또는 캐쉬된 동적 SQL은 유효하지 않게 됩니다.

일부 경우에는, 패키지를 리바인드하는 것이 불가능할 때도 있습니다. 예를 들어, 테이블이 제거된 후 재작성되지 않으면, 패키지는 리바인드될 수 없습니다. 이 경우, 오브젝트를 재작성하거나 제거된 오브젝트를 사용하지 않도록 응용프로그램을 변경해야 합니다.

대부분의 경우, 제한조건 중 하나가 제거될 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*의 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 데이터 또는 데이터 무결성의 갑작스런 손실을 방지
- 권한 부여되지 않은 액세스로부터 가치 있는 데이터를 보호. 중요한 정보가 『권한』 없는 사용자에게 의해 액세스되지 않는지 확인하십시오.
- 권한 부여되지 않은 사용자가 악의로 데이터를 삭제하거나 고치는 등의 위해를 가하지 못하도록 방지
- 325 페이지의 『제6장 DB2 활동 감사』에서 언급한 사용자의 데이터 액세스 모니터링

이 절에는 다음과 같은 내용이 들어 있습니다.

- 264 페이지의 『설치를 위한 사용자 ID 및 그룹 선택』
- 267 페이지의 『서버에 대한 인증 방법 선택』
- 274 페이지의 『원격 클라이언트에 대한 인증 고려사항』
- 274 페이지의 『파티션된 데이터베이스 고려사항』
- 274 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』
- 282 페이지의 『연합 데이터베이스 인증 처리』
- 288 페이지의 『특권, 권한 및 권한 부여』
- 306 페이지의 『데이터베이스 오브젝트에 대한 액세스 제어』
- 318 페이지의 『타스크 및 필수 권한 부여』
- 320 페이지의 『시스템 키탈로그 사용』

보안을 위한 계획: 데이터베이스 액세스 제어 플랜의 목표를 정의하는 것으로 시작하여, 누가 어떤 환경에서 무엇에 액세스할 것인지 지정하십시오. 사용자의 계획에는 데이터베이스 기능, 기타 프로그램의 기능 및 관리 프로시저어를 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 DB2를 계획, 설치 및 구성하는 데 필요한 모든 정보를 제공합니다.

DB2의 설치를 완료하는 단계는 사용자 이름, 그룹 이름 및 암호를 필요로 합니다. 설치 동안에, 관리자는 이들 각 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.

암호는 사용자 인증시 매우 중요합니다. 운영 체제 레벨에서 아무런 인증 요구사항이 설정되지 않고 데이터베이스가 사용자 인증을 위해 운영 체제를 사용하고 있다면, 사용자는 연결할 수 있도록 허용됩니다. 예를 들어, UNIX 운영 체제에서 정의되지 않은 암호는 널(NULL)로 취급합니다. 그리고 정의된 암호가 없는 임의의 사용자도 널(NULL) 암호를 가진 것처럼 취급됩니다. 운영 체제를 바로 보면 이는 일치되며, 해당 사용자는 유효성을 확인받고 데이터베이스로 연결할 수 있습니다. 운영 체제가 데이터베이스에 대해 사용자 인증을 하도록 하려면 운영 체제 레벨에서 암호를 요청해야 합니다.

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 또는 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, 이들 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 그룹을 제어할 수 있습니다. DB2 설치 및 후속 인스턴스 및 데이터베이스 작성에 대한 보안 요구사항을 완료하려면 다음 지침에 따라야 합니다.

시스템 관리자 그룹으로 정의된 임의의 그룹이 있어야 합니다(SYSADM_GROUP을 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 기타 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 인증되도록 암호와 연관되어야 합니다. 암호 작성시 권장사항은 조직의 암호 이름 지정 지침에 따르는 것입니다.

Windows NT 플랫폼 고려사항

Windows NT용 Enterprise - Extended Edition에서 작업할 때, 시스템 관리자(SYSADM) 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 두 가지 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 설정하십시오.

Windows NT 도메인 환경에서 기본적으로, 기본 도메인 제어기(PDC)에서 관리자 그룹에 속하는 도메인 사용지만이 인스턴스에 대한 SYSADM 권한을 갖습니다. 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 둘다)를 추가하십시오. 이렇게 하려면, 다음 명령을 입력하십시오.

```
DB2STOP
DB2 UPDATE DBM CFG USING SYSADM_GROUP global_group
DB2START
```

UNIX 플랫폼 고려사항

UNIX 기반 플랫폼의 경우, 분리(fenced) 사용자 정의 함수(UDF) 및 저장 프로시저에 대한 그룹이 작성되어야 하며, 분리(fenced) UDF 또는 저장 프로시저를 사용하는 모든 사용자 ID는 이 그룹의 구성원이어야 합니다. SYSADM 그룹을 사용하여, 분리(fenced) UDF 또는 저장 프로시저 그룹의 이름이 손쉬운 식별을 보장해야 합니다. 분리(fenced) UDF 또는 저장 프로시저 그룹에 속하는 사용자 ID는 해당 기본값으로 그룹과 연관된 권한 및 특권을 가집니다.

보안상의 이유로, 분리(fenced) ID로서 인스턴스 이름을 사용하지 않도록 권장합니다. 그러나 분리(fenced) UDF 또는 저장 프로시저를 사용하려고 계획하지 않는 경우, 또다른 사용자 ID를 작성하는 대신 인스턴스 이름에 분리(fenced) ID를 설정할 수 있습니다.

권장사항은 이 그룹과 연관하여 인식되는 사용자 ID를 작성하는 것입니다. 분리(fenced) UDF 및 저장 프로시저에 대한 사용자는 인스턴스 작성 스크립트의 매

개변수로서(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 작업 중인 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.

DB2 이름 지정 규칙은 369 페이지의 『부록A. 이름 지정 규칙』을 참조하십시오.

db2icrt 명령은 인스턴스 소유자의 홈 디렉토리 아래에 기본 SQL 라이브러리(sqllib) 디렉토리를 작성합니다.

서버에 대한 인증 방법 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. 인증 데이터베이스 관리 프로그램 구성 때 개변수에 대한 자세한 정보는 *관리 안내서: 성능의 『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. 자세한 정보는 282 페이지의 『연합 데이터베이스 인증 처리』를 참조하십시오.

다음 인증 유형이 제공됩니다.

SERVER

인증이 지역 운영 체제 보안을 사용하여 서버에서 발생함을 지정합니다. 사용자 ID와 암호가 연결 또는 접속 시도 중에 지정되는 경우, 서버의 유효

한 사용자 ID 및 암호의 조합과 비교되어 사용자가 인스턴스에 액세스하는 데 허가를 받았는지 판별합니다. 이것이 기본 보안 메커니즘입니다.

주: 서버 코드는 연결이 지역인지 원격인지를 감지합니다. 지역 연결의 경우, 인증이 SERVER라면, 사용자 ID와 암호가 인증에 필요하지 않습니다.

원격 인스턴스에 SERVER 인증이 있으면, 인증이 발생하는 데에는 다음과 같은 두 가지 방법이 있습니다.

- 사용자가 사용자 ID 및 암호를 제공합니다.
- DB2가 사용자 ID 및 암호를 검색한 다음 유효성 확인을 위해 서버로 전달합니다. (사용자는 지역 머신 또는 도메인에 이미 로그인되어 있습니다.)

SERVER_ENCRYPT

서버가 암호화된 SERVER 인증 스킴을 승인하도록 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 발생함을 지정합니다. 연결 또는 접속 시도 중에 지정된 사용자 ID와 암호는 클라이언트 노드에서 유효한 사용자 ID 및 암호의 조합과 비교되어 사용자 ID가 인스턴스로의 액세스를 허가받았는지 판별합니다. 데이터베이스 서버에서 인증은 더 이상 발생하지 않습니다.

사용자가 지역 또는 클라이언트 로그인을 수행하면, 사용자는 해당 지역 클라이언트 워크스테이션에만 알려집니다.

원격 인스턴스가 CLIENT 인증을 가지고 있으면, 다른 두 매개변수 (*trust_allclnts* 및 *trust_clntauth*)가 최종 인증 유형을 판별합니다.

신뢰성 있는 클라이언트 전용에 대한 CLIENT 레벨 보안

신뢰성 있는 클라이언트는 신뢰 받는 지역 보안 시스템을 갖는 클라이언트입니다. 특히, 모든 클라이언트는 Windows 95 및 Windows 98 운영 체제를 제외하고 신뢰성 있는 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 신뢰성 있는 클라이언트 인증을 선택할 수 있습니다. 이는 모든 신뢰성 있는 플랫폼이 서버를 대신하여 사용자를 인증할 수 있음을 나타냅니다. 신뢰성 없는 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(native) 안전 보안 시스템이 없는 클라이언트로 신뢰 받을 수 있습니다.

신뢰성 있는 클라이언트라 하더라도 서버에서 완전한 인증을 받아야 할 경우가 있습니다. 신뢰성 있는 클라이언트의 유효성을 확인하는 곳을 나타내기 위해, *trust_clntauth* 구성 매개변수를 사용합니다. 이 매개변수의 기본값은 CLIENT입니다. 이 매개변수에 대한 자세한 정보는 *관리 안내서: 성능의 『DB2 구성』*을 참조하십시오.

주: 신뢰성 있는 클라이언트 전용의 경우, CONNECT 또는 ATTACH 시도 중에 사용자 ID 또는 암호가 명시적으로 제공되면, 사용자의 유효성이 클라이언트에서 확인됩니다. *trust_clntauth* 매개변수만 사용하여 USER/USING절에서 제공된 정보의 유효성을 확인하는 곳을 판별합니다.

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 클라이언트 측 인증을 수행하도록 신뢰 받을 수 있습니다. 기타 모든 클라이언트는 서버에 의해 인증되려면 사용자 ID와 암호를 제공해야 합니다.

trust_clntauth 매개변수는 위 클라이언트가 인증되는 위치를 판별하는 데 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	신뢰성 없는 비DRDA 클라이언트 인증. 암호 없음	신뢰성 없는 비DRDA 클라이언트 인증. 암호 있음	신뢰성 있는 비DRDA 클라이언트 인증. 암호 없음	신뢰성 있는 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 대한 세부사항은 보안의 *DB2 Connect* 사용자 안내서 섹션을 참조하십시오.) 이를 사용하여 데이터베이스 관리 프로그램 구성 파일의 인스턴스용으로 인증 유형을 지정하는 경우, APPC 프로토콜을 사용한 DRDA(Distributed Relational Database Architecture) 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 발생하지 않습니다. 이 값은 연결용 APPC SECURITY 매개변수가 SAME 또는 PROGRAM으로 지정된 경우에만 지원됩니다.

DCS_ENCRYPT

DB2 Connect가 암호화된 SERVER 인증 스킴을 승인하도록 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 대한 자세한 정보는 274 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』을 참조하십시오.

DCE_SERVER_ENCRYPT

서버가 DCE 인증 또는 암호화된 SERVER 인증 스킴을 승인하도록 지정합니다. 클라이언트 인증이 DCE이거나 지정되지 않으면, 클라이언트는 DCE 보안 서버를 사용하여 인증됩니다. DCE 보안에 대한 자세한 정보는 274 페이지의 『사용자 인증을 위한 DCE 보안 서비스 사용』을 참조하십시오.

클라이언트 인증이 DCE 또는 SERVER이면, 클라이언트는 사용자 ID와 암호를 서버로 전달하여 인증됩니다. 클라이언트 인증이 DCS_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 스킴으로서 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 요청되

는 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 써드 파티 인증 서비스로서 인증을 수행합니다. 이 키는 사용자 증명되며 지역 또는 네트워크 서비스가 요청될 때 모든 경우에서 사용자 식별을 검증하는 데 사용됩니다. 키는 명확한 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 스킴을 승인하도록 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 연합 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 대한 정보는 빠른 시작 매뉴얼을 참조하십시오.

2. 구성 파일 자체로의 액세스는 구성 파일의 정보에 의해 보호되므로, 인증 정보를 변경하려면 인스턴스로부터 부주의하게 사용자를 잠그지 마십시오. 다음 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 문제점을 일으킬 수 있습니다.

이러한 상황이 발생하지 않도록 할 수 있는 방법이 있습니다. 우연히 DB2 시스템으로부터 자신을 잠그게 되면, 확실한 특권을 받은 지역 운영 체제 보안 사용자를 사용하여 데이터베이스 관리 프로그램 구성 파일을 갱신하는 보통 DB2 보안 검사를 겹쳐쓸 수 있는 모든 플랫폼에서 사용 가능한 장애 안전 옵션이 있습니다. 이 사용자는 데이터베이스 관리 프로그램 구성 파일을 갱신하여 문제점을 수정하는 특권을 항상 가지고 있습니다. 그러나 이 보안 생략은 데이터베이스 관리 프로그램 구성 파일의 지역 갱신으로 제한됩니다. 원격으로 또는 기타 DB2 명령에 대해 장애 안전 사용자를 사용할 수 없습니다. 이 특수 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 검사를 전달합니다.

3. Windows NT 보안에 대한 추가 정보는 429 페이지의 『부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법』을 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 올바른 것으로 간주합니다.

파티션된 데이터베이스 고려사항

파티션된 데이터베이스에서 데이터베이스의 파티션마다 정의된 동일한 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 수행합니다. 연결 로

그인 문맥은 USER/USING절을 사용하여 CONNECT 또는 ATTACH에 제공된 사용자 ID와 암호를 사용하여 DB2 세션용으로 설정됩니다. 마지막으로, 대표 로그인 문맥은 DB2 클라이언트가 DB2 서버 응용프로그램의 일부로 사용된 경우 발생합니다. 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 가져옵니다. DB2 서버가 DCE 서버 ID를 사용하기 보다는 DCE 클라이언트의 원래 ID를 사용하여 요청을 처리할 수 있습니다. 대표 로그인 문맥을 설정하는 방법에 대한 정보는 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 제품을 구입하여 사용하는 경우, 이 두 파일을 각각 db2dces.dll 및 db2dcec.dll에 복사하십시오. 또다른 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 문의하십시오.

DCE용 DB2 사용자 설정 방법

사용자는 DCE 레지스트리에 등록되어 있어야 하고 DB2와 사용하기 전에 올바른 속성을 가지고 있어야 합니다. DCE 핵심부를 작성하는 방법에 대한 정보는 플랫폼에 맞는 DCE 문서를 참조하십시오.

DCE 인증 서버를 사용하고자 하는 각 DB2 사용자는 활성화된 클라이언트 플랫폼과 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 맵핑합니다. 권한 부여 이름은 2차 권한 부여 이름이지만, 동일한 제한사항이 적용됩니다. 그룹 작성 및 구성원에 대한 추가 정보는 DCE 문서를 참조하십시오.

ERA의 정보는 사용자의 DCE 핵심부 이름 또는 그룹 이름을 특정 서버 DCE 핵심부 이름의 DB2 권한 부여 이름에 맵핑합니다. ERA를 사용하려면, 이 속성 형식을 나타내는 ERA 스키마가 정의된 상태여야 합니다. DCE 셸마다 한 번씩 완료되어야 하고 다음 단계를 수행하여 완료됩니다.

1. 유효한 DCE 관리자로서 DCE에 로그인하십시오.
2. dcecp를 호출한 후, 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 올바르게 않은 권한 부여 이름이 주

어진 경우, DB2가 해당 사용자를 인증하려는 시도를 할 때 오류가 발생합니다. (인증이 CONNECT, ATTACH, DB2START 또는 인증을 요구하는 기타 조작에서 발생함을 기억하십시오.) 또한, 권한 부여 이름과 DCE 핵심부의 지정이 1대1의 고유한 형식인지 확인해야 합니다. DCE는 이러한 조건을 검사하지 않습니다.

DB2 클라이언트가 DB2 UDB 서버에 액세스하면, DCE 핵심부로 등록된 상태 인 경우, ERA 정보를 추가하여 핵심부 이름에서 권한 부여 이름까지 매핑을 제공해야 합니다. 사용자 또는 그룹마다 한 번씩 완료되어야 하고 다음 단계를 수행하여 완료됩니다.

- 유효한 DCE 관리자로서 DCE에 로그인하십시오.
- dcecp를 호출한 후, 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 매핑이 또다른 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 매핑되어야 합니다.

DB2 authid에 DCE 핵심부 이름을 매핑하기 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 대한 정보는 플랫폼 고유의 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 등록하십시오. 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(keytab) 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(keytab) 파일을 작성해야 합니다. 키탭(keytab) 파일을 작성하는 방법에 대한 자세한 정보는 사용자 플랫폼에 맞는 DCE 문서에 있습니다. 해당 문서에서, 키탭(keytab) 파일과 *dcecp keytab* 명령 또는 *rgy_edit* 명령과 연관된 세부사항을 참조하십시오. DB2 키탭(keytab) 파일의 이름은 *keytab.db2*여야 하고, 인스턴스용 *sqllib* 디렉토리의 *security* 서브디렉토리에 상주해야 합니다. (Intel 기반 운영 체제에서, 파일은 *sqllib* 디렉토리에 있는 *INSTANCENAME* 서브디렉토리의 *security* 서브디렉토리에 상주해야 합니다. *INSTANCENAME*은 작업하는 인스턴스의 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 보호되어야 합니다.

다음은 키탭(keytab) 파일 작성에 대한 예입니다.

- 유효한 DCE 사용자로서 DCE에 로그인하십시오.
- *rgy_edit*를 호출한 후, 프롬프트에서 다음을 입력하십시오.


```
> ktadd -p principal_name -pw principal_password \  
> -f keytab.db2
```

DCE 구성이 완료된 후, 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 완료됩니다.

```
db2 update database manager configuration using authentication DCE  
sysadm_group DCE_group_name
```

그런 다음, 유효한 DB2 DCE 사용자에게 대해 dce_login을 수행하고 DB2START를 발행하십시오.

주: DCE 인증을 사용하여 DB2를 시작하기 전에, 인스턴스에 대한 SYSADM으로 사용되는 DCE 사용자 핵심부를 정의하여 인스턴스를 시작, 중지하고 관리할 수 있는 유효한 DCE 사용자 ID를 가져야 합니다. 수행 방법에 대한 지시 사항은 275 페이지의 『DCE용 DB2 사용자 설정 방법』을 참조하십시오.

이 지시사항에 추가하여, 작성된 핵심부는 인스턴스에 대한 SYSADM_GROUP의 구성원이어야 합니다. 기본적으로, 이 그룹 이름은 명시적으로 지정된 그룹이 없는(즉, SYSADM_GROUP이 널(NULL)인 경우) DCE 인증용 DB2ADMIN이지만, 인스턴스용 인증 유형을 사용자가 선택하는 그룹 이름(인증 이름)으로 변경하기 전에 갱신될 수 있습니다. 선택하는 DCE 그룹에는, 해당 이름을 지정된 SYSADM_GROUP 권한 부여 이름에 매핑하는 정의된 ERA가 있어야 합니다.

DB2 관리 서버의 기능 중 하나는 DB2 인스턴스를 시작하는 것입니다. AUTHENTICATION = DCE이면, 인스턴스용 DB2 키탭(keytab) 파일에 사용되는 DCE 핵심부는 DB2 authid에 매핑될 유효한 DCE 핵심부를 가져야 합니다. 이 매핑은 DB2 인스턴스를 시작하기 위한 DB2 관리 서버에 필요합니다. 유효한 매핑은 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(keytab) 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, 완전한 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. 다음 제한사항은 DCE 인증을 사용할 경우에 존재합니다.

- GRANT 또는 REVOKE문을 사용하는 경우, USER 및 GROUP 키워드는 반드시 지정된 권한 부여 이름에 자격을 주도록 지정되어야 하며, 그렇지 않으면 오류가 나타납니다.
- CREATE SCHEMA문의 AUTHORIZATION절을 사용하는 경우, 지정된 권한 부여 이름의 그룹 멤버십은 이 절 다음에 나오는 명령문을 수행하는 데 필요한 권한 부여 평가시 고려되지 않을 것입니다. CREATE SCHEMA문을 실행하는 동안 권한 부여 실패가 발생합니다.
- 패키지가 패키지의 원래 바인더를 제외하고 사용자에게 의해 리바인딩되면, 원래 바인더의 특권이 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 티켓에서 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 보존될 수 있는 경우에, 어떠한 제한사항도 시행되지 않습니다. 그

러나 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 작동합니다. 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 Pipes와 결합하여 사용될 수 있으나, 두 개의 보안 메커니즘은 서로 별개로 작동합니다. 사용자는 유효한 DCE 로그인 문맥을 제공해야 하며, NT Named Pipes 지원의 기준을 충족시키는 사용자 ID에 대한 NT 도메인에 로그인해야 합니다.

위의 두 가지 예제와 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 또한 DCE 암호화 티켓에 포함된 동일한 값이 통신 계층의 암호화되지 않은 배선에서 흐름을 의미할 수도 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 보내는 가능성을 피하기 위한 것입니다. 이때 APPC 계층에서의 DCE 보안은 DB2에 의해 지원되지 않습니다.

연합 데이터베이스 인증 처리

Distributed Join 설치 기능을 설치하고 데이터베이스 관리 구성 변수 *federated*를 'YES'로 설정하면, DB2 시스템은 연합 시스템으로 작동합니다. 연합 시스템에 있는 데이터베이스 인증 설정값은 표준 DB2 정의와는 다소 차이가 있습니다. 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 작성하십시오. 또한 정확한 사용자 ID와 암호 정보를 데이터 소스에 전달하도록 서버 옵션이 제대로 설정되어 있는지도 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 설정에 따라 직접 데이터 소스로 전달됩니다. 인증은 Oracle 또는 DB2 계열 데이터 소스에서만 발생합니다.

인증이 DCS로 설정되어 있으면 주의하십시오. 인증은 클라이언트에서도 DB2에서도 완료되지 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 매핑을 정의해야 합니다. 데이터 소스가 DB2 연결을 신뢰하면, 외부 보안 시스템에서 받은 ID가 데이터 소스로 전달될 수 있으므로 사용자 매핑은 필요하지 않습니다.

기타 DB2 인증 설정도 가능하며, 하나 이상을 설정하면 데이터 소스로의 전송을 위해 DB2에서 암호를 사용할 수도 있습니다. DB2와 클라이언트 인증 설정으로 DB2로 암호가 전송되면, 해당 암호는 데이터 소스에서 추가 인증 처리에 사용할 수 있습니다. 자세한 정보는 270 페이지의 표4를 참조하십시오.

사용자 ID 및 암호를 데이터 소스로 전달

데이터 소스로의 인증 정보 전송을 제어하는 방법에는 DB2 인증, 사용자 매핑, 서버 옵션 및 APPC 보안 설정이 있습니다.

인증 설정값

이 절의 목적은 인증 설정값이 연합 시스템에서의 전역 인증 처리에 미치는 영향 정도를 명확히 하는 것입니다. (인증 설정 정의는 282 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 사

용할 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, 암호를 DB2로 보내어 데이터 소스로서의 전송에 사용될 수도 있습니다.

사용자 맵핑

DB2는 DB2로의 연결에 사용된 권한 부여 이름 또는 DB2에 정의된 권한 부여 이름을 보낼 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 승인된 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 맵핑 명령문의 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 설정으로 문자를 변환시키도록 하십시오. 자세한 정보는 『서버 옵션』을 참조하십시오.

인증 설정이 DCE이고 데이터 소스에 인증 처리(암호를 예상함)가 필요하다면 사용자 맵핑을 작성해야 합니다. DB2는 DB2 사용자 ID만을 데이터 소스로 전달합니다. 암호를 해당 사용자 ID에 맵핑한 다음, 데이터 소스로 보내야 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 183 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 보내야 하는 경우, 서버 옵션 암호를 ‘ENCRYPTION’으로 설정하십시오. DB2에서 인증 설정값이 CS_ENCRYPT 또는 SERVER_ENCRYPT이면, 암호를 ‘ENCRYPTION’으로 설정하는 것이 바람직합니다.

사용자 ID를 항상 데이터 소스로 보냅니다.

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 보내기 전에 대문자로 변환합니다.

‘N’ DB2는 권한 부여 이름 또는 암호를 변환하지 않습니다.

‘L’ DB2가 권한 부여 이름 또는 암호를 데이터 소스로 보내기 전에 소문자로 변환합니다.

널(NULL)

DB2는 먼저 권한 부여 이름과 암호를 대문자로서 보냅니다. 실패하면, 이를 소문자로 변환하여 다시 보냅니다.

널(NULL) 설정은 많은 경우를 처리할 수 있으므로 매력적으로 보일 수 있습니다. 그러나 성능면에서 보면, 연결에 대해 하나의 시도만을 수행하도록 이러한 옵션을 설정하는 것이 바람직합니다. fold_id 및 fold_pw 옵션 모두가 널(NULL)로 설정 되면, DB2가 권한 부여 이름과 암호를 보내기 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 설정

| 사용자 ID와 암호가 필요한 APPC를 통해 DRDA 데이터 소스로 연결하는 경우
| 또는 인증 설정이 DCS이고 DRDA 데이터 소스에서 인증하는 경우, APPC 보안
| 설정이 DB2와 해당 데이터 소스 간의 연결을 위해 PROGRAM으로 되어 있는지
| 확인하십시오.

연합 데이터베이스 인증 예

이 절에서는 연합 시스템 인증 및 권한 부여 단계에 대한 개요를 설명합니다. 연합 데이터베이스 인증 및 권한 부여 처리에 대한 개요는 287 페이지의 그림3을 참조하십시오.

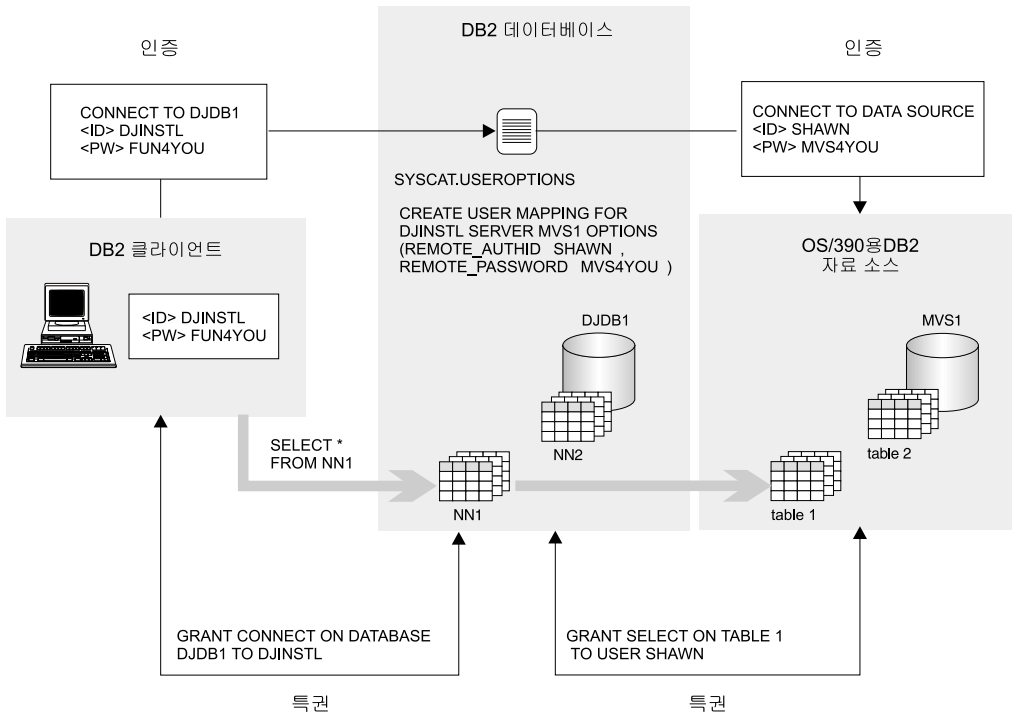


그림 3. 연합 데이터베이스 인증 및 권한 부여 처리

이 시나리오에 있는 타스크는 사용자 DJINSTL을 사용 가능하게 하여 두 개의 별칭(NN1 및 NN2)에 대해 UNION 조작을 수행하는 것입니다. 별칭은 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. 기타 데이터 소스는 DJINSTL의 ID와 암호가 같은 DB2 시스템입니다. 이 데이터 소스 DB21에서는 사용자 ID와 암호를 대문자로 보내기만 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 암호를 보내고 있는지 확인하십시오. 또한, 클라이언트와 서버 인증 유형이 같은지도 확인하십시오. DB2 서버에서 다음 명령을 입력하여 DB2 서버 인증 유형을 검사하십시오.

```
GET DATABASE MANAGER CONFIGURATION
```

클라이언트에서 다음 명령을 입력하여 클라이언트 인증 유형을 검사하십시오.

```
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, 암호를 데이터 소스로 보내는지 확인하십시오. 연합 데이터베이스 DJDB1으로 연결한 후, 다음 명령을 발행하십시오.

```
ALTER SERVER MVS1 OPTIONS (SET password 'Y')
ALTER SERVER DB21 OPTIONS (SET password 'Y')
```

암호를 적합한 문자로 DB21 데이터 소스에 보내는지 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 별칭을 선택하는 것입니다.

```
GRANT CONNECT ON DATABASE DJDB1 TO DJINSTL;
```

이제, 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 보낼 수 있습니다. 별칭이 참조하는 데이터 소스 오브젝트에 액세스하는 데 추가 과정이 필요할 수도 있습니다. (특권은 주로 별칭이 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 방법을 제공합니다. 이들이 함께 작동하여 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.

- 291 페이지의 『시스템 관리 권한(SYSADM)』
- 292 페이지의 『시스템 제어 권한(SYSCTRL)』
- 293 페이지의 『시스템 유지보수 권한(SYSMAINT)』
- 294 페이지의 『데이터베이스 관리 권한(DBADM)』
- 295 페이지의 『LOAD 권한』

다음 유형의 특권이 있습니다.

- 296 페이지의 『데이터베이스 특권』
- 298 페이지의 『스키마 특권』
- 300 페이지의 『테이블 공간 특권』
- 300 페이지의 『테이블 및 뷰 특권』
- 303 페이지의 『별칭 특권』
- 304 페이지의 『서버 특권』
- 304 페이지의 『패키지 특권』
- 305 페이지의 『색인 특권』.

290 페이지의 그림4에서는 권한과 이들의 제어 스펙 간의 관계(데이터베이스, 데이터베이스 관리 프로그램)에 대해 설명합니다.

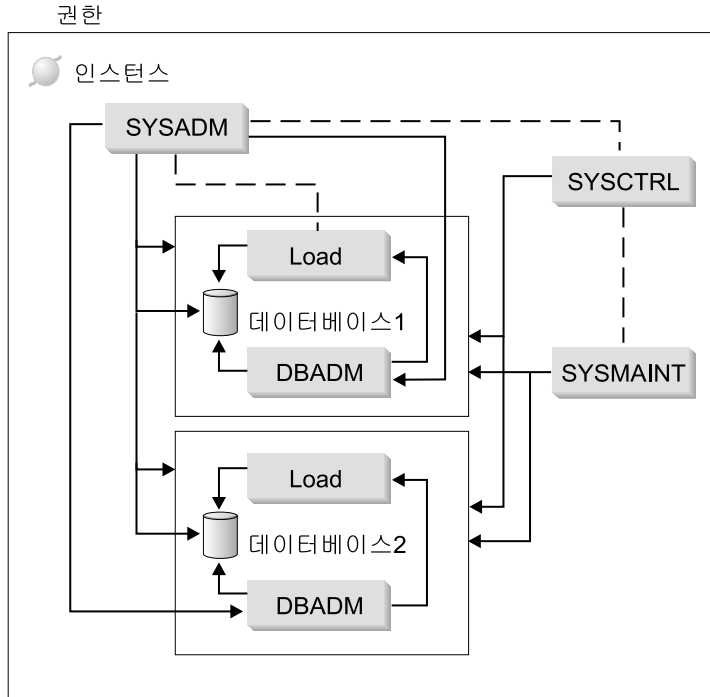


그림 4. 권한 계층 구조

사용자 또는 그룹이 다음과 같은 권한 부여 레벨 중 하나 이상에 있을 수 있습니다.

- 관리자 권한(SYSADM 또는 DBADM)은 오브젝트 세트에 대한 전반적인 특권을 부여합니다.
- 시스템 권한(SYSCTRL 또는 SYSMOINT)은 시스템 관리에 대한 전반적인 특권을 부여하지만, 데이터에 액세스하지는 못합니다.
- LOAD 권한(LOAD)은 LOAD 유틸리티 및 자동 로드 프로그램 유틸리티 특권을 제공하여 데이터를 테이블로 로드합니다.
- 소유권 특권(또한 어떤 경우에는 CONTROL 특권)은 특정 오브젝트에 대한 완전한 특권을 제공합니다.
- 개별 특권은 사용자가 특정 오브젝트에 대해 세부 함수를 수행할 수 있도록 권한 부여할 수 있습니다.

- 내재적 특권은 패키지를 실행하는 특권을 갖는 사용자에게 권한 부여할 수 있습니다. 사용자가 응용프로그램을 수행하는 동안, 패키지 내에서 사용되는 데이터 오브젝트에 대한 명시적 특권이 반드시 필요하지는 않습니다. 자세한 정보는 311 페이지의 『패키지를 통한 간접 특권 허용』을 참조하십시오.

관리자 권한(SYSADM 또는 DBADM) 또는 소유권 특권(CONTROL)이 있는 사용자는 GRANT 및 REVOKE문을 사용하여 다른 사용자에게 권한 부여하거나 권한 취소합니다 (306 페이지의 『데이터베이스 오브젝트에 대한 액세스 제어』 참조). 해당 특권이 WITH GRANT OPTION이면, 테이블, 뷰 또는 스키마 특권을 또다른 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 연관된 특권일 경우, 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 연관된 모든 권한과 특권을 자동으로 부여받게 됩니다.

특정 명령, API 또는 SQL문에 필요한 권한에 대한 정보는 *Command Reference*, *Administrative API Reference* 또는 *SQL* 참조서를 참조하십시오.

시스템 관리 권한(SYSADM)

SYSADM 권한은 최상위 레벨의 관리자 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 플

랫폼에 사용된 보안 기능을 통해 데이터베이스 관리 프로그램 외부에서 제어됩니다. 시스템 보안 기능을 사용하여 SYSADM 권한을 작성, 변경 또는 삭제하는 방법에 대한 정보는 빠른 시작을 참조하십시오.

SYSADM 권한을 가진 사용자만이 다음 기능을 수행할 수 있습니다.

- 데이터베이스 이주
- 데이터베이스 관리 프로그램 구성 파일 변경(SYSCTRL 또는 SYSMOINT 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)』
- 293 페이지의 『시스템 유지보수 권한(SYSMAINT)』
- 294 페이지의 『데이터베이스 관리 권한(DBADM)』

주: SYSADM 권한을 가지고 있는 사용자가 데이터베이스를 작성할 때, 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 취소하십시오.

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 제거
- 테이블 공간의 제거, 작성 또는 변경
- 새로운 데이터베이스로 복원

또한, SYSCTRL 권한을 갖는 사용자는 『시스템 유지보수 권한(SYSMAINT)』 권한을 가진 사용자의 기능도 수행할 수 있습니다.

SYSCTRL 권한을 갖는 사용자는 데이터베이스에 연결할 수 있는 내재적 특권도 갖게 됩니다.

주: SYSCTRL 권한을 가지고 있는 사용자가 데이터베이스를 작성할 때, 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 취소하십시오.

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, 데이터베이스의 데이터에 직접 액세스하지는 못합니다. 시스템 유지보수 권한은 사용자가 중요한 데이터가 들어 있는 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스를 유지보수하기 위한 목적으로 설계되었습니다.

SYSMANT 권한은 *sysmaint_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.

SYSMANT 이상의 시스템 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 데이터베이스 구성 파일 갱신
- 데이터베이스 또는 테이블 공간 백업
- 기존의 데이터베이스로 복원
- 롤 포워드 복구 수행
- 인스턴스 시작 또는 중지
- 테이블 공간 복원
- 추적 수행
- 데이터베이스 관리 프로그램 인스턴스 또는 해당 데이터베이스의 데이터베이스 시스템 모니터 스냅샷

SYSMANT, DBADM 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 내재적 특권도 갖게 됩니다.

데이터베이스 관리 권한(DBADM)

DBADM 권한은 관리 권한 중 두 번째로 높은 레벨입니다. 이것은 특정 데이터베이스에만 적용되며, 사용자가 특정 유틸리티를 수행하고, 데이터베이스 명령을 발행하고, 데이터베이스 테이블의 데이터에 액세스할 수 있도록 합니다. DBADM 권

한이 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 제거

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 보유한 데이터베이스에 대해서만 위의 기능을 수행할 수 있습니다.

LOAD 권한

테이블에서 INSERT 특권은 물론, 데이터베이스 레벨에서 LOAD 권한을 가지는 사용자는 LOAD 명령 또는 자동 로드 프로그램 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

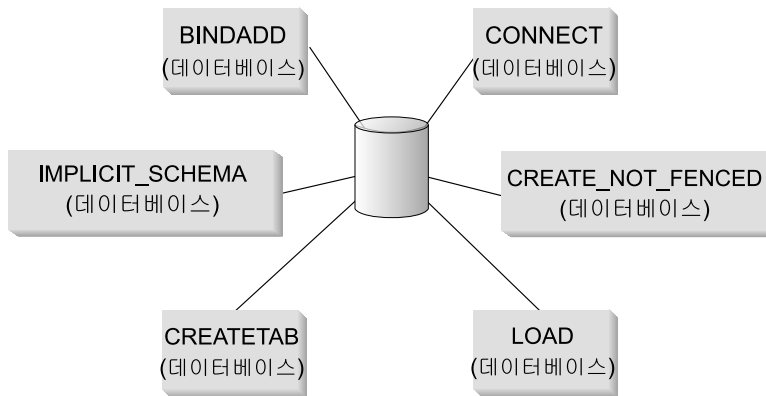


그림 5. 데이터베이스 특권

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 『비 분리』 프로시저를 작성할 수 있습니다. 『비 분리』 UDF 또는 프로시저는 데이터베이스 관리 프로그램이 이러한 UDF 또는 프로시저로부터의 저장영역 또는 제어 블록을 보호하지 않기 때문에 세심하게 테스트되어야 합니다. (결과적으로, 『비 분리』를 수행하도록 허용된 UDF 또는 프로시저가 완벽하게 작성되어 있지 않거나 테스트되지 않으면, 시스템에 심각한 문제가 발생할 수 있습니다.) (자세한 정보는 응용프로그램 개발 안내서 또는 SQL 참조서를 참조하십시오.)

- `IMPLICIT_SCHEMA`는 모든 사용자가 아직 존재하지 않는 스키마 이름과 함께 `CREATE`문을 사용하여 오브젝트를 작성함으로써 스키마를 내재적으로 작성할 수 있도록 합니다. `SYSIBM`은 내재적으로 작성된 스키마의 소유자가 되며, `PUBLIC`에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.
- `LOAD`는 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 `PUBLIC`에 권한 부여됩니다.

- `CREATETAB`
- `BINDADD`
- `CONNECT`
- `IMPLICIT_SCHEMA`
- `USERSPACE1` 테이블 공간에서의 `USE` 특권
- 시스템 카탈로그 뷰에 대한 `SELECT` 특권

어떤 특권을 제거하려면, `DBADM` 또는 `SYSADM`은 `PUBLIC`으로부터 명시적으로 특권을 권한 취소하십시오.

내재된 스키마 권한(`IMPLICIT_SCHEMA`) 고려사항

새로운 데이터베이스가 작성되거나 데이터베이스가 이전 릴리스로부터 이주될 경우, `PUBLIC`에는 `IMPLICIT_SCHEMA` 데이터베이스 권한이 부여됩니다. 이 권한이 있는 모든 사용자는 오브젝트를 작성하고 아직 존재하지 않는 스키마 이름을 지정함으로써 스키마를 작성할 수 있습니다. `SYSIBM`은 내재적으로 작성된 스키마의 소유자가 되며, `PUBLIC`에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.

데이터베이스에서 스키마 오브젝트를 내재적으로 작성할 수 있는 사용자를 제어할 필요가 있을 경우, `IMPLICIT_SCHEMA` 데이터베이스 권한을 `PUBLIC`에서 권한 취소하십시오. 일단 이렇게 되면, 스키마 오브젝트를 작성할 수 있는 방법은 다음 세 가지 밖에 없습니다.

- 모든 사용자는 `CREATE SCHEMA`문에 자신의 권한 부여 이름을 사용하여 스키마를 작성할 수 있습니다.

- DBADM 권한을 갖는 모든 사용자는 아직 존재하지 않는 모든 스키마를 명시적으로 작성할 수 있으며, 또다른 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 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 299 페이지의 그림6에 나와 있습니다.

오브젝트 특권

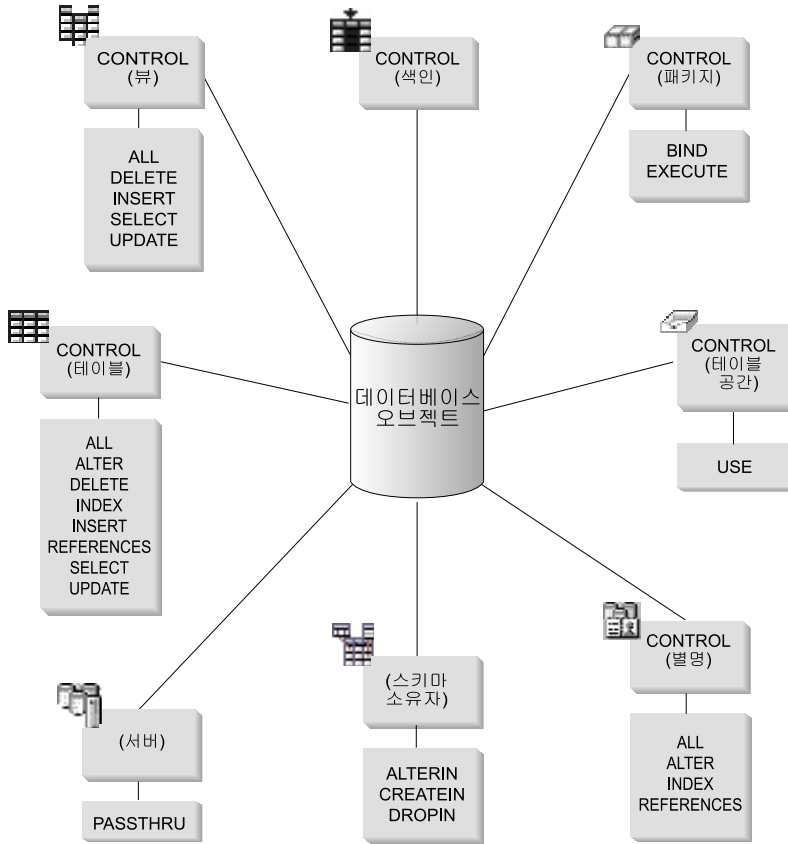


그림 6. 오브젝트 특권

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 제거하도록 합니다.

스키마 소유자는 모두 이러한 특권과 기능을 다른 사용자에게 권한 부여할 권한을 가집니다. 스키마 오브젝트 내에서 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 사용자에게 부여하는 능력을 갖습니다. 기본적으로, 데이터베이스 작성 시간에 테이블 공간 USERSPACE1에 대한 USE 특권이 PUBLIC으로 부여되지만, 이 특권은 권한 취소될 수 있습니다.

USE 특권은 SYSCATSPACE 또는 임의의 시스템 임시 테이블 공간에서 사용될 수 없습니다.

테이블 및 뷰 특권

테이블 및 뷰 특권에는 데이터베이스의 테이블 또는 뷰에 대한 조치가 포함됩니다. 사용자는 데이터베이스에 대해 CONNECT 특권을 가져야 다음과 같은 특권을 사용합니다.

- CONTROL은 테이블 또는 뷰를 제거할 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 제거할 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 제거할 수 있고, 테이블에 대해 ALTER 특권을 가지고 상위 테이블에 대해 REFERENCES 특권을 가지고 있거나 해당 컬럼에 대해 REFERENCES 특권

을 가진 사용자는 외부 키를 작성하거나 제거할 수 있습니다. ALTER 특권을 갖는 사용자는 또한 테이블에서 COMMENT ON을 수행할 수도 있습니다.

- DELETE는 사용자가 테이블 또는 뷰에서 행을 제거할 수 있도록 합니다.
- INDEX는 사용자가 데이터베이스에 색인을 작성할 수 있도록 합니다. 색인의 작성자는 색인에 대해 자동으로 CONTROL 특권을 가지게 됩니다. 자세한 정보는 305 페이지의 『색인 특권』을 참조하십시오.
- INSERT는 사용자가 테이블 또는 뷰에 행을 삽입하고 IMPORT 유틸리티를 수행하도록 합니다.
- REFERENCES는 사용자가 테이블을 관계의 상위로 지정하여, 외부 키를 작성하거나 제거할 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. CONTROL 특권으로 권한 부여된 모든 특권을 권한 취소하려면, 각 특권을 명시적으로 권한 취소하거나, REVOKE문에서 ALL 키워드를 지정하십시오. 예를 들어, 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 계층 구조의 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, 하위 테이블에 직접적으로 조작용 보류 하십시오.

테이블 계층 구조 내 테이블 간의 상위 테이블/하위 테이블 관계는 SELECT, UPDATE 및 DELETE와 같은 조작용이 해당 조작용의 목표 테이블 및 모든 해당 하위 테이블(있는 경우)의 행에 영향을 줄 것임을 의미합니다. 이 작동을 『대체 가능성』이라 부를 수 있습니다. 예를 들어, 유형이 Manager_t인 하위 테이블 Manager로 유형이 Employee_t인 Employee 테이블을 작성했다고 가정합니다. 관리자도 사원의 (특정한) 일부이므로, 구조화 유형 Employee_t 및 Manager_t 간의 유형/부속 유형 관계와 테이블 Employee 및 Manager 간의 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 조작용

```
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 조작용을 수행할 수 있습니다. 그러나 이
| 러한 사용자는 Manage 하위 테이블에 직접 SELECT 조작용을 수행할 수 없기 때
| 문에, Manager 테이블의 상속되지 않은 컬럼에는 액세스할 수 없습니다.

| 마찬가지로, Employee에 대한 UPDATE 특권을 가진 사용자는 Manager에 대해
| UPDATE 조작용을 수행할 수 있기 때문에, Manager 테이블에 대한 명시적
| UPDATE 특권을 가지고 있지 않더라도 정규 사원과 관리자 모두에게 영향을 줍
| 니다. 그러나 이러한 사용자는 Manage 하위 테이블에 직접 UPDATE 조작용을 수
| 행할 수 없기 때문에, Manager 테이블의 상속되지 않은 컬럼은 갱신할 수 없습니
| 다.

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 대한 자세한 정보는 *관리 안내서: 성능*을 참조하십시오.

뷰 특권이 판별되는 방법에 대한 정보는 *SQL 참조서* 매뉴얼의 CREATE VIEW 문을 참조하십시오.

별칭 특권

별칭 특권은 데이터베이스에 있는 별칭에 대해 조치를 취할 수 있는 특권입니다. 이러한 특권은 별칭이 참조하는 데이터 소스 오브젝트에 대한 특권에 영향을 주지 않습니다. 사용자는 데이터베이스에 대해 CONNECT 특권을 가져야 다음과 같은 특권을 사용합니다.

- CONTROL은 사용자에게 별칭 특권을 제거하고, 개별 별칭 특권을 권한 부여 및 권한 취소할 수 있는 별칭에 대한 모든 특권을 제공합니다. CONTROL을 권한 부여하려면 SYSADM 또는 DBADM 권한이 있어야 합니다. 별칭 작성자는 자동으로 별칭에 대해 CONTROL 특권을 가지게 됩니다.
- ALTER를 사용하면 별칭에서 컬럼 이름을 변경하고, 컬럼의 데이터 유형이 맵핑되는 DB2 유형을 추가 또는 변경하며, 별칭 컬럼에 대한 컬럼 옵션을 설정할 수 있습니다.
- INDEX를 사용하면 별칭에서 색인 스펙을 작성할 수 있습니다. 색인 스펙 작성자는 자동으로 색인에 대해 CONTROL 특권을 갖습니다.
- REFERENCES를 사용하면 관계에서 상위로 별칭을 지정하여 외부 키를 작성 및 제거할 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 별칭에 대한 CONTROL 특권이 권한 부여되면, 해당 별칭에 대한 다른 모든 특권은 WITH GRANT OPTION으로 자동 권한 부여됩니다. 나중에 별칭에 대한 CONTROL 특권을 권한 취소해도, 사용자는 자동으로 권한 부여된 다른 특권을 계속 보유합니다.

데이터 소스에 액세스하려면, 별칭이 참조하는 데이터 소스의 오브젝트에 대해 적합한 권한을 가지고 있어야 합니다.

사용자가 하나 이상의 별칭을 참조하는 뷰에 액세스하는 경우, 사용자는 데이터 소스에서 별칭이 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 명령문은 다음과 같습니다.

```
GRANT PASSTHRU ON SERVER ORACLE1 TO USER SHAWN
```

PASSTHRU문의 구문에 대한 완전한 정보는 SQL 참조서를 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, 제거 또는 실행할 능력과 이러한 특권을 다른 사용자에게 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.

별칭이 들어 있는 패키지를 실행할 권한을 가진 사용자는 패키지 내의 별칭에 대해 추가 특권 또는 권한 레벨이 필요하지 않습니다. 그러나 별칭이 참조하는 오브젝트가 들어 있는 데이터 소스에서 인증 검사를 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 별칭이 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 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

색인 특권

색인 또는 색인 스펙 작성자는 색인에 대해 자동으로 CONTROL 특권을 갖게 됩니다. 색인에 대한 CONTROL 특권은 색인을 제거할 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(300 페이지의 『테이블 및 뷰 특권』 참조).

순차 특권

순차의 작성자는 자동으로 USAGE 특권을 받게 됩니다. 순차를 위한 NEXTVAL 및 PREVVAl 표현식을 사용하려면 USAGE 특권이 필요합니다. 기타 사용자가 NEXTVAL 및 PREVVAl 표현식을 사용하도록 허용하려면 순차 특권을 공용(PUBLIC)에 권한 부여해야 합니다. 이로써 모든 사용자가 지정된 순차로 표현식을 사용할 수 있게 됩니다.

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 320 페이지의 『시스템 카탈로그 사용』에서 설명됩니다.

권한 부여는 다음 세 가지 방법으로 제어됩니다.

- 명시적 권한 부여는 GRANT 및 REVOKE문으로 제어되는 특권을 통해 제어됩니다.
- 내재적 권한 부여는 오브젝트를 작성 및 제거함으로써 제어됩니다.
- 간접 권한은 패키지와 연관됩니다.

이 절에는 다음과 같은 내용이 들어 있습니다.

- 307 페이지의 『특권 권한 부여』
- 308 페이지의 『특권 권한 취소』
- 310 페이지의 『오브젝트 작성 및 제거에 의한 내재적 권한 부여 관리』
- 311 페이지의 『패키지를 통한 간접 특권 허용』
- 313 페이지의 『뷰를 사용하여 데이터에 대한 액세스 제어』
- 316 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 체제에서, 사용자 또는 그룹에 특권을 권한 부여할 것인지 여부를 지정하십시오. GRANT 및 REVOKE문은 둘다 키워드 USER 및 GROUP을 지원합니다. 이들 선택적 키워드가 사용되지 않는 경우, 데이터베이스 관리 프로그램이 운영 체제 보안 기능을 검사하여 권한 부여 이름이 사용자 또는 그룹 식별 여부를 판별합니다. 권한 부여 이름이 사용자 또는 그룹 모두인 경우, 오류가 리턴됩니다.

다음 예에서는 HERON 사용자에게 EMPLOYEE 테이블에 대한 SELECT 특권을 권한 부여합니다.

```
GRANT SELECT
ON EMPLOYEE TO USER HERON
```

다음 예에서는 HERON 그룹으로 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 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

특권 권한 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. 또다른 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: 테이블 또는 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 대한 자세한 정보는 *SQL 참조서 매뉴얼*을 참조하십시오.

특권이 동일한 이름을 갖는 사용자와 그룹에 모두 권한 부여되면, 특권을 권한 취소할 때 GROUP 또는 USER 키워드를 지정하십시오. 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 경우, 해당 특권이 직접 권한 취소될 때까지 이름을 보존합니다.

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. 또다른 사용자에 의해 뷰에 대한 특권을 직접 권한 부여 받은 경우, 특권이 계속 보유됩니다.

특권을 그룹으로 권한 부여(GRANT)한 다음, 그룹의 한 구성원에게서 권한을 취소(REVOKE)하려는 상황이 있을 수 있습니다. 다음과 같은 오류 메시지 SQL0556N을 받지 않고 그렇게 하기 위해서는 오직 두 가지 방법만이 있습니다.

- 그룹으로부터 구성원을 제거하거나, 더 적은 수의 구성원이 있는 그룹 하나를 새로 작성하고 새 그룹에게 특권을 권한 부여(GRANT)합니다.
- 그룹으로부터 특권을 권한 취소(REVOKE)한 다음, 사용자에게 개별적으로 특권을 권한 부여합니다(권한 부여 ID).

명시적으로 권한 부여된 테이블(또는 뷰) 특권이 DBADM 권한을 갖는 사용자로부터 권한 취소된 경우, 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 해당 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 올바르지 않음으로 표시되지만, 해당 권한을 갖는 사용자에 의해 리바인드되면 다시 유효성이 확인될 수 있습니다. 특권이 연속적으로 응용프로그램 바인더에 다시 권한 부여될 경우 패키지도 재빌드될 수 있으며, 응용프로그램을 수행하면 내재적 리바인드가 성공적으로 트리거됩니다. 특권이 PUBLIC으로부터 권한 취소되면, PUBLIC 특권에 근거해야만 바인드 가능한 사용자가 바인드한 모든 패키지는 유효하지 않습니다. DBADM 권한이 사용자로부터 권한 취소되면, 데이터베이스 유틸리티와 연관된 패키지를 포함하여 해당 사용자가 바인드한 모든 패키지가 올바르지 않게 됩니다. 올바르지 않음으로 표시된 패키지를 사용하려고 시도하면 시스템이 패키지를 리바인드하려고 합니다. 이러한 리바인드 시도가 실패할 경우, 오류가 발생합니다(SQLCODE-727). 이 경우, 다음과 같은 권한을 갖는 사용자가 패키지를 명시적으로 리바인드해야 합니다.

- 패키지를 리바인드할 수 있는 권한
- 패키지 내에서 사용한 오브젝트에 대한 해당 권한

이들 패키지는 특권이 권한 취소될 때마다 리바인드되어야 합니다. REVOKE 및 REBIND PACKAGE문에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

하나 이상의 특권에 근거하는 트리거를 정의한 경우와, 하나 이상의 해당 특권이 없으면, 트리거는 사용될 수 없습니다.

오브젝트 작성 및 제거에 의한 내재적 권한 부여 관리

데이터베이스 관리 프로그램은 CREATE SCHEMA, CREATE TABLESPACE, CREATE VIEW 또는 CREATE INDEX문을 나타내는 사용자 또는 PREP 명령어나 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 제거될 때 제거됩니다.

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 판별되는 방법에 대한 정보는 *SQL 참조서 매뉴얼*의 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 대한 자세한 정보는 *Command Reference*를 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. 각각의 이들 조치에는 하나 이상의 특권이 필요합니다.

패키지를 바인드하는 개별 및 PUBLIC에 권한 부여된 권한은 정적 SQL이 바인드될 때 권한 부여 검사용으로 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 BIND때 지정된 경우, 이 패키지 내에서 정적 SQL문에 대한 모든 권한 부여 실패는 BIND를 실패하도록 야기하지 않으며 이러한 SQL문의 유효성이 런타임에서 재확인됩니다. 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.

별칭이 들어 있는 패키지를 통한 간접 특권 허용

패키지에 별칭에 대한 참조가 들어 있으면, 패키지 작성자와 패키지 사용자에 대한 권한 부여 처리는 다소 복잡합니다. 패키지 작성자가 별칭이 들어 있는 패키지를 바인드하는 경우, 패키지 작성자는 데이터 소스에서 별칭이 참조하는 테이블과 뷰에 대해 인증 검사 또는 특권 검사를 통과할 필요는 없습니다. 그러나 패키지 실행하는 데이터 소스에서 인증 및 권한 부여 검사를 통과해야 합니다.

예를 들어, 패키지 작성자의 .SQL 파일에 여러 SQL문이 들어 있는 것으로 가정하십시오. 하나의 정적 명령문은 지역 테이블을 참조합니다. 또다른 동적 명령문은 별칭을 참조합니다. 패키지가 바인드되면, 패키지 작성자의 인증 ID(authid)는 지역 테이블에 대한 특권을 검증하는 데 사용됩니다. 그러나 별칭이 식별하는 데이터 소스 오브젝트에 대해서는 어떠한 검사도 수행되지 않습니다. 또다른 사용자가 패키지를 수행하면, 해당 패키지에 대해 EXECUTE 특권을 가지고 있고 사용자가 테이블을 참조하는 명령문에 대한 추가 특권 검사를 통과할 필요가 없는 것으로 가정합니다. 그러나 별칭을 참조하는 명령문의 경우, 패키지를 실행하는 사용자는 데이터 소스에서 인증 검사 및 특권 검사를 통과해야 합니다.

.SQL 파일에 동적 SQL문과 테이블 및 별칭 참조의 혼합만 들어 있으면, 지역 오브젝트 및 별칭에 대한 DB2 권한 부여 검사는 비슷합니다. 패키지 사용자는 명령문 내의 지역 오브젝트(테이블, 뷰)에 대한 특권 검사를 통과해야 하며, 또한 별칭 오브젝트에 대한 특권 검사도 통과해야 합니다. (패키지 사용자는 별칭이 식별하는 오브젝트가 들어 있는 데이터 소스에서 인증 및 특권 검사를 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 맵핑을 작성하여 변경될 수 있습니다.

주: 별칭은 정적 SQL에서 지정될 수 없습니다. 별칭이 들어 있는 패키지와 함께 DYNAMICRULES 옵션(BIND로 설정된)을 사용하지 마십시오.

DB2가 DB2 계열 데이터 소스와 통신할 때 동적 SQL을 사용하므로 별칭이 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 대한 자세한 정보는 *SQL 참조서*를 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 조회에 WHERE절을 지정함으로써, 권한 부여된 사용자가 뷰를 통해 주소지정되는 행을 제한할 수 있습니다.
- 데이터 소스 테이블이나 뷰에 있는 행 또는 컬럼의 부속 집합에만 액세스. 별칭을 통해 데이터 소스에 액세스하는 경우, 별칭을 참조하는 지역 DB2 뷰를 작성할 수 있습니다. 이들 뷰는 하나 이상의 데이터 소스에서 별칭을 참조할 수 있습니다.

주: 둘 이상의 데이터 소스에 대한 별칭 참조사항이 들어 있는 뷰를 작성할 수 있으므로, 사용자는 하나의 뷰에서 다중 데이터 소스에 있는 데이터에 액세스할 수 있습니다. 이들 뷰를 *다중 위치 뷰*라고도 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 스키마

에 대한 CREATEIN 특권 또는 데이터베이스에 대한 IMPLICIT_SCHEMA 권한이 있어야 합니다. 자세한 정보는 173 페이지의 『뷰 작성』을 참조하십시오.

별칭을 참조하는 뷰를 작성하는 경우, 뷰에서 별칭이 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.

1. 데이터 소스 테이블에서 사용자가 액세스할 수 있는 컬럼에 대해 데이터 소스 뷰를 작성합니다.
2. 이 뷰에 대한 SELECT 특권을 사용자에게 권한 부여합니다.
3. 뷰를 참조하는 별칭을 작성합니다.

그런 다음, 새로운 별칭을 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 테이블의 정보를 액세스해야 할 경우가 있습니다. 예를 들어, 다음과 같습니다.

- 인사부에서는 전체 테이블을 갱신하고 볼 수 있어야 합니다.

이러한 요구사항은 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 요구사항은 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. 이 요구사항은 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

NAME	LOCATION
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta
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 보존할 수 있습니다.

SYSADM 권한은 감사 기능 관리자 도구 db2audit을 사용하는 것이 필요합니다.

DB2 감사 기능의 자세한 설명은 325 페이지의 『제6장 DB2 활동 감사』를 참조하십시오.

데이터 암호화

보안 플랜의 한 파트는 사용자의 데이터 암호화를 포함합니다. 암호화를 하려면, ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR 및 GETHINT과 같은 암호화 및 암호 해독 내장 함수를 사용하면 됩니다. 이 함수(해당 구문을 포함한)에 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

ENCRYPT 함수는 암호 기반 암호화 메소드를 사용하여 데이터를 암호화합니다. 이 함수는 또한 사용자가 암호 힌트를 캡슐화하도록 합니다. 암호 힌트는 암호화된 데이터에 임베드되어 있습니다. 한번만 암호화되면, 데이터를 암호 해독할 수 있는 유일한 방법은 올바른 암호를 사용하는 것입니다. 이 함수를 사용하려고 선택한 개발 프로그램으로 잊어버린 암호 및 사용할 수 없는 데이터의 관리에 대해 계획해야 합니다.

암호화(ENCRYPT) 함수의 결과는 첫 번째 인수와 데이터 유형이 같습니다.

VARCHAR만 암호화됩니다.

결과 선언된 길이는 다음 중 하나입니다.

- 선택적 힌트 매개변수가 지정된 경우, 데이터 인수에 42를 더한 길이
- 선택적 힌트 매개변수가 지정되지 않은 경우, 데이터 인수에 10을 더한 길이

DECRYPT_BIN 및 DECRYPT_CHAR 함수는 암호 기반 암호 해독을 사용하여 암호를 해독합니다.

DECRYPT_BIN 및 DECRYPT_CHAR 함수의 결과는 첫 번째 인수와 데이터 유형이 같습니다.

결과 선언된 길이는 원래 데이터의 길이입니다.

GETHINT 함수는 캡슐화된 암호 힌트를 리턴합니다. 암호 힌트는 데이터 소유자가 암호를 기억하도록 도와주는 문구입니다. 예를 들어, 단어 『Ocean』은 암호 "태평양(Pacific)"을 기억하는 힌트로 사용할 수 있습니다.

데이터를 암호화하는 데 사용되는 암호는 다음과 같은 두 가지 방법 중 하나로 판별합니다.

- 암호 인수. 암호는 암호화(ENCRYPT) 함수가 호출되면 명시적으로 전달되는 문자열입니다. 데이터는 주어진 암호로 암호화하고 암호 해독합니다.
- 특수 레지스터 암호. SET ENCRYPTION PASSWORD 문은 암호 값을 암호화하고 암호화된 암호를 데이터베이스 관리 프로그램으로 보내어 특수 레지스터에 저장합니다. 암호 매개변수 없이 호출된 ENCRYPT, DECRYPT_BIN 및 DECRYPT_CHAR 함수는 ENCRYPTION PASSWORD 특수 레지스터의 값을 사용합니다.

특수 레지스터의 초기값 또는 기본값은 비어 있는 문자열입니다.

암호의 유효 길이는 6에서 127까지입니다. 힌트의 유효 길이는 0에서 32까지입니다.

암호화 암호(ENCRYPTION PASSWORD) 특수 레지스터는 클라이언트에서부터 설정하고, 암호는 클라이언트에서 암호화하며 데이터베이스 서버로 보낸 다음, 암호를 해독합니다. 암호를 확실히 읽을 수 없게 하려고 데이터베이스 서버에서 암호를 다시 암호화합니다. DECRYPT_BIN 및 DECRYPT_CHAR 함수는 특수 레지스터를 암호 해독한 후 사용합니다. ENCRYPTION PASSWORD에 있는 값 또한 읽을 수 없어야 합니다. 게이트웨이 보안은 지원되지 않습니다.

타스크 및 필수 권한 부여

모든 조직이 동일한 방식으로 작업량을 분담하지는 못합니다. 표5에는 기타 일부 공통된 직책, 보통 작업에 수반되는 타스크, 이러한 타스크를 수행하는 데 필요한 권한 또는 특권이 나열되어 있습니다.

표5. 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

표 5. 공통 직책, 태스크 및 필수 권한 부여 (계속)

직책	태스크	필수 권한 부여
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. 일부 경우에는 SYSCTRL 권한
시스템 운영자	데이터베이스를 모니터링하고 백업 기능을 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 특권, 일부 테이블에 대한 특권 목록
사용자 분석자	시스템 카탈로그 뷰를 검사하여 응용프로그램에 대한 데이터 요구사항 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 다음 주 참조
정보 센터 상담원	조회 사용자의 데이터 요구사항 정의, 테이블 및 뷰를 작성하고 데이터베이스 오브젝트로 액세스를 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, 작성 중인 테이블 및 뷰의 스키마에 대해서는 CREATEIN 그리고 일부 테이블 및 뷰에 대해서는 SELECT, INSERT, UPDATE, DELETE

주: 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 유지보수됩니다. 이 시스템 카탈로그는 데이터베이스가 생성될 때 작성됩니다. 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	테이블 및 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 **SYSIBM**을 권한 준 사용자로 갖게 됩니다. **SYSADM**, **SYSMAINT** 및 **SYSCTRL**은 시스템 카탈로그에 나열되지 않습니다.

CREATE 및 **GRANT**문은 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 조희를 사용하여 특권 정보를 추출하는 방법을 보여줍니다.

- 321 페이지의 『권한 부여된 특권이 있는 권한 부여 이름 검색』
- 321 페이지의 『DBADM 권한이 있는 모든 이름 검색』
- 321 페이지의 『테이블에 액세스하기 위해 권한 부여된 이름 검색』
- 322 페이지의 『사용자에게 권한 부여된 모든 특권 검색』
- 323 페이지의 『시스템 카탈로그 뷰 보안』

권한 부여된 특권이 있는 권한 부여 이름 검색

모든 특권에 대한 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.PASSTHROUGH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

정기적으로, 이 명령문에 의해 검색되는 목록은 시스템 보안 기능에 정의된 사용자 및 그룹 이름의 목록과 비교되어야 합니다. 그러면 더 이상 유효하지 않은 해당 권한 부여 이름을 식별할 수 있습니다.

주: 원격 데이터베이스 클라이언트를 지원하고 있는 경우, 원격 클라이언트에만 권한 부여 이름이 정의되고 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 개별 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 대한 자세한 정보는 *SQL 참조서*를 참조하십시오.

시스템 카탈로그 뷰 보안

데이터베이스가 작성되는 동안, 시스템 카탈로그 뷰에 대한 `SELECT` 특권이 `PUBLIC`에 권한 부여됩니다. (`PUBLIC`에 자동으로 권한 부여되는 기타 특권은 296 페이지의 『데이터베이스 특권』을 참조하십시오.) 대부분의 경우, 보안 문제점이 나타나지 않습니다. 그러나 매우 중요한 데이터의 경우에 이들 테이블이 데이터베이스에 있는 모든 오브젝트를 서술하기 때문에, 적절하지 않을 수도 있습니다. 이 경우, `PUBLIC`으로부터 `SELECT` 특권을 권한 취소하고, 특정 사용자에게 필요한 `SELECT` 특권을 권한 부여하는 방법을 고려해 보십시오. 시스템 카탈로그 뷰에 대해 `SELECT`를 권한 부여하고 권한 취소하는 방법은 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.TBAUTH`
- `SYSCAT.PACKAGEAUTH`
- `SYSCAT.INDEXAUTH`
- `SYSCAT.COLAUTH`
- `SYSCAT.PASSTHROUGHAUTH`
- `SYSCAT.SCHEMAAUTH`

이렇게 하면, 데이터베이스에 액세스하는 모든 사용자가 사용자 특권에 대한 정보를 사용할 수 없게 됩니다. 비윤리적인 사용자가 이 정보로 데이터베이스에 대해 권한 없이 액세스할 수도 있습니다.

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 데이터 값이 들어 있습니다. 이러한 통계에 관련 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, 다음 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 작성합니다.

```
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 후속 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 활동의 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 보존됩니다. 이들 레코드의 분석은 시스템 오용을 식별하는 사용 패턴을 나타낼 수 있습니다. 일단 시스템 오용이 식별되면, 이러한 시스템 오용을 감소시키거나 줄이기 위한 조치가 취해질 수 있습니다.

감사 기능은 모든 인스턴스 레벨 활동 및 데이터베이스 레벨 활동을 기록하며 인스턴스 레벨에서 작동합니다.

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 사용권한이 설정됩니다. 기본적으로, 사용권한은 인스턴스 소유자 전용의 읽기/쓰기입니다. 이들 사용권한을 변경하지 않도록 권장합니다.

감사 기능 관리자 도구(db2audit)의 사용자는 SYSADM 권한/특권을 가져야 합니다.

감사 기능은 명시적으로 중지 및 시작되어야 합니다. 감사 기능이 시작되면, 기존 감사 구성 정보를 사용합니다. 감사 기능은 DB2 서버와 무관하므로, 인스턴스가 중지되더라도 사용 중인 상태로 남습니다. 사실, 인스턴스가 중지되면, 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 중지시킵니다.
- 기록될 감사 가능한 이벤트 범주 선택을 비롯한 감사 기능의 작동을 구성합니다.
- 현재 감사 구성의 설명을 요청합니다.
- 인스턴스에서 보류 중인 감사 레코드를 비우고 감사 로그에 작성합니다.
- 감사 로그에서 플랫폼 파일 또는 ASCII 분리 파일로 감사 레코드를 포맷팅하고 복사하여 감사 레코드를 추출합니다. 로그 레코드의 분석을 위한 준비 또는 로그 레코드의 제거를 위한 준비에서 추출이 완료됩니다.
- 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 설정이 변경되거나 감사 로그가 액세스될 때 레코드를 생성합니다.
- 권한 부여 검사(CHECKING). DB2 오브젝트 또는 기능에 액세스하거나 조작하려는 시도에 대한 권한 부여 검사 동안에 레코드를 생성합니다.
- 오브젝트 유지보수(OBJMAINT). 데이터 오브젝트를 작성하거나 제거할 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 유효성 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 둘다 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 둘다 중 어느 것을 감사하는지 여부에 따라서도 다릅니다. 이러한 이유로 감사할 이벤트를 선택하는 것은 중요합니다.

감사 기능 작동

감사 기능은 영향을 미치는 데이터베이스 인스턴스를 포함하는 감사 가능한 이벤트를 기록합니다. 이러한 이유로, 감사 기능은 DB2 인스턴스가 중지되는 경우에도 조작할 수 있는 DB2의 독립적 파트입니다. 감사 기능이 사용 중이면, 중지된 인스턴스가 시작될 때 인스턴스의 데이터베이스 이벤트 감사가 재개됩니다.

감사 로그에 감사 레코드를 작성하는 시기는 인스턴스에서의 데이터베이스 성능에 상당한 영향을 줍니다. 감사 레코드의 작성하는 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 완료됩니다. 감사 레코드를 생성하는 이벤트가 레코드를 디스크에 작성할 때까지 대기합니다. 각 레코드와 연관된 대기는 DB2 성능이 저하되게 합니다.

*audit_buf_sz*의 값이 0보다 크면, 레코드 작성은 비동기적으로 완료됩니다. *audit_buf_sz*의 값이 0보다 커지면 내부 버퍼를 작성하기 위해 몇 개의 4KB 페이지가 사용됩니다. 감사 레코드의 그룹을 디스크에 작성하기 전에 몇 개의 감사 레코드를 보존하기 위해 내부 버퍼가 사용됩니다. 감사 레코드를 감사 이벤트의 결과로서 생성하는 명령문은 레코드가 디스크에 작성될 때까지 기다리지 않고 조작을 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 작성을 강제합니다. 감사 기능의 권한이 부여된 사용자는 명시적 요청으로 감사 버퍼를 비울 수 있습니다.

동기 레코드 작성이 있는지 비동기 레코드 작성이 있는지 여부에 따라 오류 발생 시기가 다릅니다. 비동기 모드의 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 설정은 오류가 DB2와 감사 기능 간에 어떻게 관리되는지를 제어합니다. 감사 기능이 사용 중이고, ERRORTYPE 감사 기능 매개변수가 AUDIT이면, 감사 기능은 DB2의 다른 파트와 동일한 방법으로 처리됩니다. 성공적이라고 간주되는 명령문과 연관된 감사 이벤트에 대한 감사 레코드가 작성되어야 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

설정되면, db2audit로부터의 모든 오류는 무시되며 조건의 SQLCODE가 리턴됩니다. ERRORTYPE 감사 기능 매개변수(및 기타 관련 매개변수)의 추가 세부사항은 330 페이지의 『감사 기능 사용 시나리오』를 참조하십시오.

API 또는 SQL문과 DB2 인스턴스에 대한 감사 설정에 따라, 없음, 하나 또는 여러 감사 레코드가 특정 이벤트에 대해 생성될 수 있습니다. 예를 들어, SELECT 부속 조치가 있는 SQL UPDATE문은 테이블에서 UPDATE 특권에 대한 권한 부여 감사의 결과가 들어 있는 하나의 감사 레코드와 테이블에서 SELECT 특권에 대한 권한 부여 점검의 결과가 들어 있는 또다른 레코드의 결과를 가져올 수 있습니다.

동적 DML(Data Manipulation Language)문의 경우, 명령문이 준비되면 모든 권한 부여 감사에 대해 감사 레코드가 생성됩니다. 이때는 어떠한 감사 검사도 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 SQL문에 대한 명령문 특권이 검사되며 하나 이상의 새로운 감사 레코드가 작성됩니다.

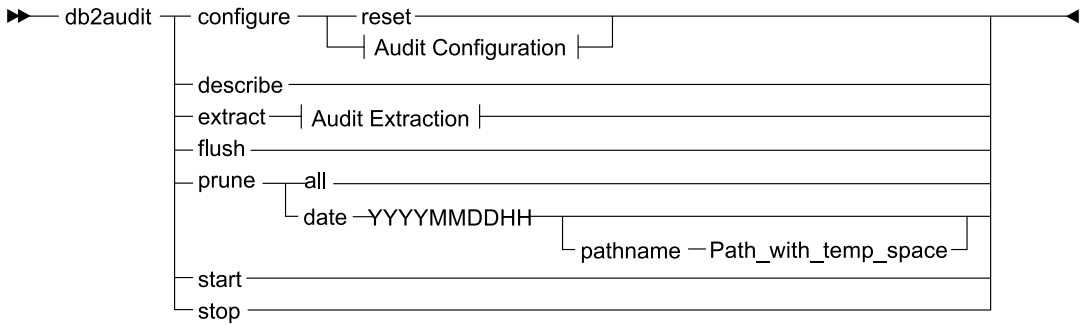
정적 DML문만 들어 있는 패키지의 경우, 감사 레코드를 생성할 수 있는 유일한 감사 가능한 이벤트는 사용자가 해당 패키지를 실행할 특권을 가지고 있는지 여부를 보기 위한 권한 부여 검사입니다. 패키지에서 정적 SQL문에 필요한 가능한 감사 레코드 작성과 권한 부여 검사는 패키지가 사전 처리 컴파일되거나 바인드될 때 수행됩니다. 패키지 내의 정적 SQL문의 실행은 감사 가능하지 않습니다. 패키지가 사용자에게 의해 명시적으로 또는 시스템에 의해 내재적으로 리바인드될 때, 정적 SQL문에 의해 필요한 권한 부여 감사에 대한 감사 레코드가 생성됩니다.

권한 부여 검사가 명령문 실행 시간에서 수행되는 명령문의 경우(예: DDL, GRANT 및 REVOKE문), 이들 명령문이 사용될 때마다 감사 레코드가 생성됩니다.

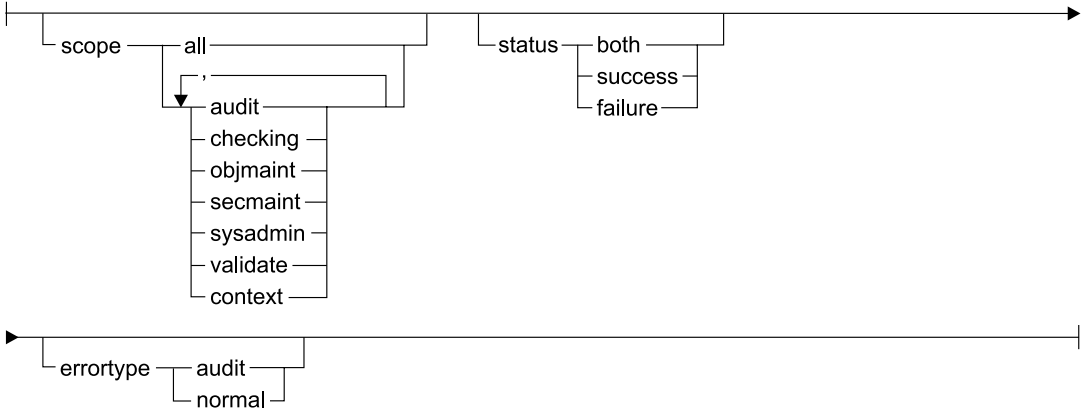
주: DDL을 실행할 때, 감사 레코드의 모든 이벤트(문맥 이벤트 제외)에 기록되는 섹션 번호는 명령문의 실제 섹션 번호와 상관없이 0이 됩니다.

감사 기능 사용 시나리오

| 다음 구문 다이어그램의 각 파트를 검토하면, 감사 기능을 사용하는 방법을 이해
| 하는 데 도움이 됩니다.



감사 구성:



감사 추출:

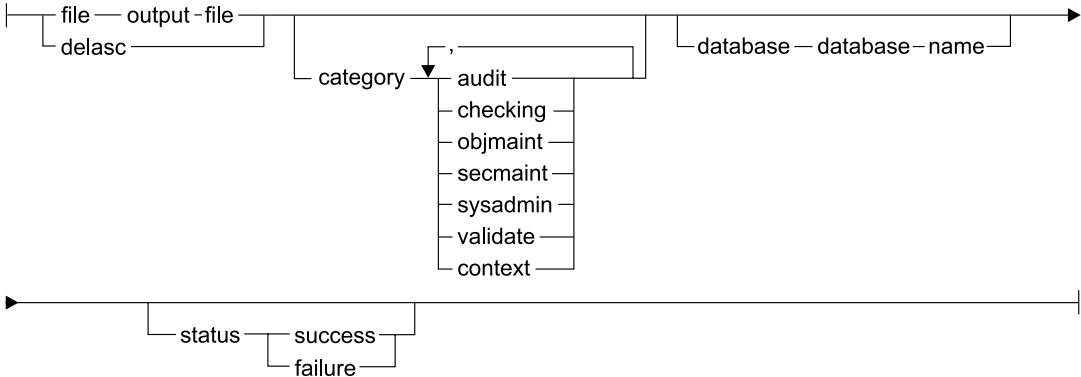


그림 7. DB2AUDIT 구문

다음은 각 매개변수의 내재된 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 사용 중일 때 발생하는 갱신은 모든 파티션에 걸쳐 DB2에 의해 수행 중인 감사에 영향을 미칩니다. 감사 기능이 시작될 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, 감사 기능은 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 로그하는지 여부를 지정합니다.

주: 조작 상태가 알려지기 전에 문맥 이벤트가 발생합니다. 그러므로 이러한 이벤트는 이 매개변수와 연관된 값과 무관하게 로그됩니다.

- **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 수행 중인 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 목적지로 감사 레코드의 이동을 허용합니다. 선택적 절이 지정되지 않으면, 모든 감사 레코드가 추출되어 플랫폼 보고서 파일에 위치합니다. 『extract』 매개변수가 지정되지 않으면, 감사 레코드는 *security* 디렉토리의 *db2audit.out*라는 파일에 위치합니다. *output_file*이 있는 경우, 오류 메시지가 리턴됩니다.

다음은 추출할 때 사용할 수 있는 가능한 옵션입니다.

- **FILE.** 추출된 감사 레코드는 파일(*output_file*)에 위치합니다.
- **DELASC.** 추출된 감사 레코드는 DB2 관계 테이블로 로드하는 데 적합한 컬럼 식별자가 있는 ASCII 형식으로 위치합니다. 출력은 각 범주에 하나씩 분리 파일에 위치합니다. 파일 이름은 다음과 같습니다.
 - *audit.del*
 - *checking.del*
 - *objmaint.del*
 - *secmaint.del*
 - *sysadmin.del*
 - *validate.del*
 - *context.del*

DELASC 선택항목은 감사 로그에서 추출할 때 기본 감사 문자열 분리문자(『0xff』)를 겹쳐쓸 수 있게 합니다. 감사 레코드를 보류할 테이블로 로드하는 준비에서 사용하려는 새로운 분리문자가 뒤에 DELASC DELIMITER를 사용하십시오. 새로운 로드 분리문자는 단일 문자(예: !) 또는 16진수(예: 0xff)를 표현하는 4바이트 문자열 중 하나입니다. 자세한 정보는 353 페이지의 『감사 기능 추가 정보 및 기술』을 참조하십시오.

- **CATEGORY.** 감사 이벤트의 지정 범주에 대한 감사 레코드가 추출됩니다. 지정되지 않으면, 모든 범주가 추출 대상이 됩니다.
- **DATABASE.** 지정 데이터베이스의 감사 레코드가 추출됩니다. 지정되지 않으면, 모든 데이터베이스가 추출 대상이 됩니다.
- **STATUS.** 지정 상태의 감사 레코드가 추출됩니다. 지정되지 않으면, 모든 레코드가 추출 대상이 됩니다.

flush 이 매개변수는 보류 중인 모든 감사 레코드가 감사 로그에 작성되도록 강제합니다.

다. 또한, 감사 상태는 감사 기능이 오류 상태에 있는 경우 『로그할 수 없음』에서 『로그 준비』 상태로 엔진에서 재설정됩니다.

prune 이 매개변수는 감사 로그에서 감사 레코드의 삭제를 허용합니다. 감사 기능이 사용 중이며 이벤트의 『감사』 범주가 감사용으로 지정된 경우, 감사 레코드는 감사 로그가 제거된 후에 로그됩니다.

다음은 제거할 때 사용할 수 있는 가능한 옵션입니다.

- ALL. 감사 로그의 모든 감사 레코드가 삭제됩니다.
- DATE *yyyymmddhh*. 지정한 날짜/시간에 발생하거나 지정한 날짜/시간 이전에 발생한 모든 감사 레코드를 감사 로그에서 삭제하도록 지정할 수 있습니다. 사용자는 감사 로그를 제거할 때 감사 기능이 임시 공간으로서 사용할

pathname

을 선택적으로 제공할 수 있습니다. 이 임시 공간은 감사 로그가 있는 디스크가 가득 차서 제거 조작할 공간이 충분하지 않을 때 감사 로그를 제거할 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 로그됩니다.

stop 이 매개변수는 감사 기능이 감사 이벤트를 중지시키게 합니다. 파티션된 DB2 인스턴스에서 이 절이 지정되면 모든 파티션에서 감사가 중지됩니다. 이벤트의 『감사』 범주가 감사에 지정된 경우, 감사 기능이 중지될 때 감사 레코드가 로그됩니다.

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 시스템에

서 공간을 해제하거나 감사 로그의 크기를 줄이도록 제거해야 합니다.

더 많은 공간이 사용 가능하면, *db2audit*를 사용하여 메모리의 데이터를 비우고 감사 프로그램을 준비 상태로 재설정하십시오. 삭제한 레코드를 복구할 수 없으면, 해당 추출이 발생했는지 또는 로그를 제거하기 전에 로그 사본을 작성했는지 확인하십시오.

| **sqlcode:** -1322
| **sqlstate:** 50830

- 파일의 저장 버전에서 복원하십시오.
- 다음을 발행하여 감사 기능 구성 파일을 재설정하십시오.

db2audit reset

SQL1323N 감사 구성 파일 액세스 중에 오류가 발생했습니다.

설명: 감사 구성 파일(db2audit.cfg)을 열 수 없거나 올바르지 않습니다. 이 오류의 가능한 이유는 db2audit.cfg 파일이 없거나 손상된 것입니다.

사용자 응답: 다음 조치 중 하나를 취하십시오.

| **sqlcode:** -1323
| **sqlstate:** 57019

감사 기능 레코드 레이아웃

DELASC 추출 옵션을 사용하여 감사 로그에서 감사 레코드가 추출되면, 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(굵은체)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.

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 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
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_FUNCTION입니다.
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

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
조정자 노드 번호	SMALLINT	조정자의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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 첫 번째 목록에 표시됩니다.
액세스가 시도됨	CHAR(18)	시도된 액세스 유형을 나타냅니다. 가능한 값은 이 표 다음에 오는 두 번째 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
오브젝트 간의 참조 제한조건 설정 시도

0x0000000000000100 CREATE
오브젝트 작성 시도

0x0000000000000200 DROP
오브젝트 제거 시도

0x0000000000000400 CREATEIN
또다른 스키마 내에서 오브젝트 작성 시도

0x0000000000000800 DROPIN
또다른 스키마 내에서 발견된 오브젝트 제거 시도

0x0000000000001000 ALTERIN
또다른 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 제거 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	조정자 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0

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
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	조정자 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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
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.

표 9. SECMAINT 이벤트에 대한 감사 레코드 레이아웃 (계속)

<pre>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;</pre>		
NAME	FORMAT	DESCRIPTION
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 ALTER TABLE with GRANT

허용된 특권 권한 부여로 테이블을 변경하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000008 DELETE TABLE

테이블을 제거하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000010 DELETE TABLE with GRANT

허용된 특권 권한 부여로 테이블을 제거하기 위해 권한 부여되거나 권한 취소된 특권

0x0000000000000020 Table Index

색인에서 권한 부여되거나 권한 취소된 특권

0x0000000000000040 Table Index with GRANT

허용된 특권 권한 부여로 색인에서 권한 부여되거나 권한 취소된 특권

0x0000000000000080 Table INSERT

테이블의 삽입에서 권한 부여되거나 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 부여되거나 권한 취소된 특권

0x0000000000000400 Table SELECT with GRANT

허용된 특권 권한 부여로 테이블의 선택에서 권한 부여되거나 권한 취소된 특권

0x0000000000000800 Table UPDATE

테이블의 갱신에서 권한 부여되거나 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 부여되거나 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 BIND 특권

0x0000000000010000 Package EXECUTE

패키지에서 권한 부여되거나 권한 취소된 EXECUTE 특권

0x0000000000020000 CREATEIN Schema

스키마에서 권한 부여되거나 권한 취소된 CREATEIN 특권

0x0000000000040000 CREATEIN Schema with GRANT

허용된 특권 권한 부여로 스키마에서 권한 부여되거나 권한 취소된 CREATEIN 특권

0x0000000000080000 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 권한

0x000000020000000 Server PASSTHRU

이 서버에 대해 통과 기능을 사용하기 위해 권한 부여되거나 권한 취소된 특권(연합 데이터베이스 데이터 소스)

0x000000010000000 Table Space USE

테이블 공간에서 테이블을 작성하기 위해 권한 부여되거나 권한 취소된 특권

0x000000020000000 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 부여되거나 권한 취소된 특권

0x0000001000000000 Column REFERENCE

테이블에서 하나 이상의 특정 컬럼에 대한 참조에서 권한 부여되거나 권한 취소된 특권

0x0000002000000000 Column REFERENCE with GRANT

특권이 허용된 테이블의 하나 이상의 특정 컬럼에 대한 참조에서 권한 부여되거나 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.
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

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
권한 부여 ID	VARCHAR(128)	감사 이벤트시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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=FOO;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 사용 중인 실행 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
조정자 노드 번호	SMALLINT	조정자 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
인증 유형	VARCHAR(32)	감사 이벤트시 인증 유형
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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	조정자 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생시 사용 중인 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 알아보기 위해 검사되는 오브젝트입니다. 예를 들어, 사용자가 컬럼을 추가하여 테이블을 변경하려고 하는 경우, CHECKING 이벤트 감사 레코드는 시도된 액세스가 『ALTER』이고 검사 중인 오브젝트 유형이 『TABLE』이었음을 나타냅니다. (주: 검사되어야 하는 것은 테이블 특권이므로 컬럼이 아닙니다.)

그러나 사용자 ID가 오브젝트를 작성 또는 바인드하거나, 오브젝트를 삭제할 수 있게 하는 데이터베이스 권한이 있는지 여부를 확인하는 것이 감사에 포함되면, 데이터베이스에 대한 검사가 있다고 하더라도, 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 작성된 다음, 또다른 OBJMAINT 이벤트 감사 레코드는 패키지의 새로운 사본의 CREATE용으로 작성됩니다.

SQL DDL(Data Definition Language)은 성공적인 것으로 로그되는 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 추출할 때, 명령문 텍스트 필드 내에서 사용하는 분리문자를 지워야 합니다. 이것은 컬럼 식별자가 있는 ASCII 파일을 추출할 때 완료될 수 있으며, 다음을 사용하여 완료됩니다.

```
db2audit extract delasc delimiter <load delimiter>
```

로드 분리문자는 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
```

추출할 때 분리문자로서 기본 로드 분리문자(『0xff』) 이외의 문자를 사용한 경우, LOAD 명령에 MODIFIED BY 옵션을 사용하십시오. 분리문자로서 『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 분리문자인 『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 수행합니다. 이와 동일한 응용프로그램이 아래 설명하는 각 예에서 사용됩니다.

극단적인 예를 표시하며 시작합니다. 모든 성공 및 실패 감사 이벤트를 감사하도록 판별했으므로, 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 작성되며, 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 수행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다. 로그에서 감사 레코드를 추출하여, 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=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;
timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;
audit event=CREATE_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;
timestamp=1998-06-24-08.42.42.018900;category=CONTEXT;audit event=COMMIT;
event correlator=3;database=FOO;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 수행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다. (그리고 이전에 *testapp*를 수행하지 않은 것으로 가정합니다.) 로그에서 감사 레코드를 추출하여, 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=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;
```

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 부여될 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 수행하면, 다음 레코드가 생성되어 감사 로그에 위치합니다. (그리고 이전에 *testapp*를 수행하지 않은 것으로 가정합니다.) 로그에서 감사 레코드를 추출하여, 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=FOO;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=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;
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;
```

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 언로드된 데이터를 LOAD 유틸리티를 사용하여 로드할 수 있습니다.

자동 로드 프로그램 유틸리티는 대량의 데이터를 나누어서, 나뉜 데이터를 파티션된 데이터베이스의 다른 파티션으로 로드합니다.

IMPORT 및 EXPORT 유틸리티는 테이블이나 뷰 및 또다른 데이터베이스 또는 스프레드시트 프로그램 간에, DB2 데이터베이스 간에, DB2 데이터베이스와 호스트 데이터베이스 간에 DB2 연결을 사용하여 데이터를 이동시킵니다.

데이터 복제(원래는 DPROPR(DataPropagator Relational))는 다른 DB2 관계형 데이터베이스의 다른 테이블에 대한 테이블 갱신사항을 자동으로 복사할 수 있는 DB2 Universal Database의 구성요소입니다.

주: 이들 주제에 대한 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.

복제에 대해서는 *복제 안내 및 참조서*를 참조하십시오.

제5부 복구

제8장 데이터베이스 복구

잘못된 것이 있을 때는 데이터베이스를 복구할 수 있어야 합니다. 문제점에는 미디어 및 저장영역 실패뿐만 아니라 전원 실패, 응용프로그램 실패가 포함됩니다. 이와 같은 문제점 발생시에 확실히 복구가 가능하게 하려면, 전체 데이터베이스나 데이터베이스를 구성하는 테이블 공간의 백업 또는 사본을 보관할 필요가 있습니다. 그러면 이 백업은 데이터베이스를 복원하는 데이터베이스 문제점을 조치하는데 사용됩니다.

문제점을 조치하는 데이터베이스의 재빌드를 복구라고 합니다. 실패 후에는 자동으로 응급 복구를 시도하여 데이터베이스를 복구합니다. 응급 복구는 데이터베이스가 불일치 상태나 사용불가 상태가 되지 않도록 보호합니다. 데이터베이스에 대한 트랜잭션은 데이터베이스 실패 발생시에 불완전하게 남겨질 수 있습니다. 응급 복구는 완료되지 않은 트랜잭션을 구간 복원하거나 완료된 트랜잭션을 파악합니다. 이러한 조치로 데이터베이스를 지속적이고 유용하게 합니다.

데이터베이스가 손상되고 데이터베이스의 내용에 대해 확신할 수 없을 때 두 가지 다른 유형의 복구가 있습니다. 손상된 데이터베이스를 복구하는 두 가지 방법은 버전 복구 및 롤 포워드 복구입니다. 읽기 전용 데이터베이스에 작업하거나 데이터베이스에 기록된 트랜잭션에 관여하지 않는 곳에 작업하면, 버전 복구만 필요합니다. 데이터베이스 백업을 했으면, 데이터베이스 사본을 적용 또는 복원합니다. 버전 복구가 호출되었습니다. 데이터베이스에 적용된 트랜잭션을 가진 데이터베이스에 작업하고 있고, 데이터베이스가 적용된 해당 변경사항 모두를 가지고 있음을 알아야 한다면, 롤 포워드 복구를 완료해야 합니다. 롤 포워드 복구는 데이터베이스의 백업 복원을 포함합니다. 그런 다음, 데이터베이스에 대해 트랜잭션을 레코딩하는 로그 레코드를 적용해야 합니다. 로그의 응용프로그램은 데이터베이스에 대한 모든 활동을 반복하여, 실패 지점의 바로 전 상태가 되도록 합니다. 이 복구 메소드를 사용하면 데이터베이스에 대한 어떠한 변경도 빠뜨리지 않습니다. 로깅은 이 복구 메소드에 대한 키입니다.

| 주: 이들 주제에 대한 모든 정보와 *Command Reference* 및 *Administrative API*
| *Reference*의 비교 가능한 주제는 데이터 복구 및 고가용성 안내 및 참조서에
| 통합되어 있습니다.

| 데이터 복구 및 고가용성 안내 및 참조서는 이들 주제에 대한 1차 단일 정보
| 소스입니다.

제6부 부록 및 끝머리

부록A. 이름 지정 규칙

다음에 관한 정보를 요청하는 이름 지정 규칙을 설명하는 절로 이동하십시오.

- 『일반 이름 지정 규칙』
- 『오브젝트 이름 지정 규칙』
- 374 페이지의 『연합 시스템에서 대소문자 구별 값이 보존되는 방법』

일반 이름 지정 규칙

다르게 지정되어 있지 않으면, 모든 이름은 다음과 같은 문자를 포함할 수 있습니다.

- A에서 Z까지. 대부분의 이름에 사용시, A에서 Z까지의 문자는 소문자에서 대문자로 변환됩니다.
- 0에서 9까지
- @, #, \$ 및 _ (밑줄)

이름은 숫자 또는 밑줄 문자로 시작할 수 없습니다.

SQL의 예약어를 테이블, 뷰, 컬럼, 색인 및 권한 부여 ID 이름으로 사용하지 마십시오. SQL 예약어의 목록에 대해서는 *SQL 참조서*를 참조하십시오.

사용자의 운영 체제와 DB2와 작업하는 곳에 따라 별개로 작업하는 기타의 특수 문자가 있습니다. 그러나 특수 문자가 작업하는 동안 제대로 작업할 것이라는 보장은 없습니다. 데이터베이스에서 오브젝트의 이름을 지정할 때, 이들 기타 특수 문자를 사용하는 것은 바람직하지 않습니다.

오브젝트 이름 지정 규칙

모든 오브젝트는 일반 이름 지정 규칙을 준수합니다. 뿐만 아니라, 일부 오브젝트는 아래에 표시된 제한사항을 가집니다.

표 15. 데이터베이스, 데이터베이스 별명 및 인스턴스 이름 지정 규칙

오브젝트	지침서
<ul style="list-style-type: none"> • 데이터베이스 • 데이터베이스 별명 • 인스턴스 	<ul style="list-style-type: none"> • 데이터베이스 이름은 카탈로그화되어 있는 위치에서 고유해야 합니다. DB2의 UNIX 기반 구현에서, 이 위치는 디렉토리 경로인데 반해 Windows 구현인 경우에는 드라이브 이름이 됩니다. • 데이터베이스 별명 이름은 시스템 데이터베이스 디렉토리에서 고유해야 합니다. 새로운 데이터베이스가 생성될 때, 데이터베이스 이름에 대해 기본값으로 별명이 따르게 됩니다. 결과적으로, 해당 이름을 가진 데이터베이스가 없다고 하더라도, 데이터베이스 별명으로 존재하는 이름을 사용하여 데이터베이스를 생성할 수 없습니다. • 데이터베이스, 데이터베이스 별명 및 인스턴스 이름은 최대 8바이트까지 가능합니다. • Windows NT 및 Windows 2000 시스템에서, 인스턴스는 서비스 이름과 동일한 이름을 가질 수 없습니다. <p>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, 또다른 언어에서 데이터베이스를 사용할 계획이 있다면, 이러한 문자를 사용하는 것은 피해 주십시오.</p>

표 16. 데이터베이스 오브젝트 이름 지정 규칙

오브젝트	지침서
<ul style="list-style-type: none"> • 별명 • 버퍼 풀 • 컬럼 • 이벤트 모니터 • 색인 • 메소드 • 노드 그룹 • 스키마 • 저장 프로시저어 • 테이블 • 테이블 공간 • 트리거 • UDF • UDT • 뷰 	<p>다음 경우를 제외하고는 최대 18바이트까지 포함할 수 있습니다.</p> <ul style="list-style-type: none"> • 테이블 이름(뷰 이름, 요약 테이블 이름, 별명 이름 및 상관 이름 등), 최대 128바이트까지 가능 • 컬럼 이름, 최대 30바이트까지 가능 • 스키마 이름, 최대 30바이트까지 가능 • 오브젝트 이름은 다음 문자도 포함할 수 있음 <ul style="list-style-type: none"> - 유효한 강조 문자(예: ö) - 다중 바이트 공간을 제외한 다중 바이트 문자(다중 바이트 환경의 경우)

스키마 이름에 대한 추가 정보

- 18바이트를 초과하는 스키마 이름을 가진 테이블은 복제할 수 없습니다.
- 사용자 정의 유형(UDT)은 8바이트를 초과하는 스키마 이름을 가질 수 없습니다.
- 스키마 이름 SYSCAT, SYSFUN, SYSIBM, SYSSTAT는 예약어이므로 사용될 수 없습니다.
- 앞으로 나타날 수도 있는 잠재적인 이주시의 문제점을 미리 방지하려면, SYS로 시작되는 스키마 이름을 사용하지 마십시오. 데이터베이스 관리 프로그램은 SYS로 시작하는 스키마 이름을 사용하여 트리거, 사용자 정의 유형 또는 사용자 정의 함수(UDF)를 작성하지 못하게 합니다.
- 스키마 이름으로서 SESSION을 사용하지 않도록 권장합니다. 선언된 임시 테이블은 SESSION으로 규정되어야 합니다. 그러므로 응용프로그램 논리가 지나

치게 복잡할 수 있는 경우에 지속적 테이블의 이름과 동일한 임시 테이블을 응용프로그램에서 선언하는 것이 가능합니다. 선언된 임시 테이블을 처리할 때를 제외하고는 스키마 SESSION 사용을 피하십시오.

표 17. 사용자, 사용자 ID 및 그룹 이름 지정 규칙

오브젝트	지침서
<ul style="list-style-type: none"> • 그룹 이름 • 사용자 이름 • 사용자 ID 	<ul style="list-style-type: none"> • 그룹 이름은 최대 8바이트까지 포함할 수 있습니다. • UNIX 기반 시스템에서 사용자 ID는 최대 8자까지 포함할 수 있습니다. • Windows에서 사용자 이름은 최대 30자까지 포함할 수 있습니다. Windows NT 및 Windows 2000은 현재 20자의 실제 한계를 가집니다. • DCE 인증을 사용하는 경우, 사용자 이름은 8자의 한계를 가집니다. • DCE 또는 클라이언트 인증을 사용하지 않을 때, 비 Windows 32비트 클라이언트가 8자를 초과하는 사용자 이름으로 Windows NT 및 Windows 2000에 연결하더라도 사용자 이름 및 암호를 명시적으로 지정하는 경우는 지원됩니다. • 다음을 이름 및 ID로 사용할 수 없습니다. <ul style="list-style-type: none"> - SQL 참조서에 나열된 USERS, ADMINS, GUESTS, PUBLIC, LOCAL 또는 모든 SQL 예약어 - IBM, SQL 또는 SYS로 시작 - 강조 문자 포함 <p>UNIX 기반 시스템에서, 그룹과 사용자가 같은 이름을 가질 수 있습니다. GRANT문에 있어서는, 그룹을 참조한 것인지 또는 사용자를 참조한 것인지의 여부를 지정해야 합니다. REVOKE문의 경우, 사용자 또는 그룹을 지정하는 것은 서로 다른 GRANTEE TYPE 값을 갖는 GRANTEE에 대해 권한 부여 카탈로그 테이블에 다중 행이 있는지 여부에 따라 달라집니다.</p> <p>Windows NT, 지역 그룹, 전역 그룹 및 사용자가 같은 이름을 가질 수 없습니다.</p> <p>OS/2에서, 그룹과 사용자가 같은 이름을 가질 수 없습니다.</p> <p>주:</p> <ol style="list-style-type: none"> 1. 사용자 ID 및 암호의 대소문자를 구별하는 운영 체제도 있습니다. 이러한 경우가 있는지 운영 체제 문서를 검사해야 합니다. 2. 성공적인 CONNECT 또는 ATTACH 후에 리턴된 권한 부여 ID는 8자로 절단됩니다. ellipsis(...)가 권한 부여 ID에 추가되고 SQLWARN 필드는 절단을 표시하는 경고를 포함합니다. 자세한 정보는 SQL 참조서의 CONNECT문을 참조하십시오.

암호에 대한 추가 정보

암호 유지보수 작업을 수행할 때 필요할 수 있습니다. 이러한 작업은 서버에서 요구되고, 많은 사용자가 서버 환경에서 작업할 수 없거나 작업하기에 용이하지 않으므로, 작업을 수행한다는 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 CONNECT 문을 사용하여 암호를 변경하십시오.

```
CONNECT TO <database> USER <userid> USING <password>
NEW <new_password> CONFIRM <new_password>
```

DB2 클라이언트 구성 지원 프로그램(CCA)의 『암호 변경』 대화 상자를 사용하여 암호를 변경할 수도 있습니다. 암호를 변경하는 이러한 방법에 대한 자세한 정보는 SQL 참조서 및 CCA 온라인 도움말을 참조하십시오.

표 18. 연합 데이터베이스 오브젝트 이름 지정 규칙

오브젝트	지침서
<ul style="list-style-type: none"> • 함수 맵핑 • 색인 스펙 • 별칭 • 서버 • 유형 맵핑 • 사용자 맵핑 • 랩퍼 	<ul style="list-style-type: none"> • 별칭, 맵핑, 색인 스펙, 서버 및 랩퍼 이름은 128바이트를 초과할 수 없습니다. • 서버, 별칭 옵션 및 옵션 설정은 255바이트로 제한됩니다. • 연합 데이터베이스 오브젝트 이름도 다음을 포함할 수 있습니다. <ul style="list-style-type: none"> - 유효한 강조 문자(예: ö) - 다중 바이트 공간을 제외한 다중 바이트 문자(다중 바이트 환경의 경우)

오브젝트 이름에서 분리 식별자 사용

키워드를 사용할 수 있습니다. 만일 SQL 키워드로도 해석될 수 있는 구문에서 키워드가 사용되면, 이것은 분리 식별자로 지정되어야 합니다.

분리 식별자를 사용함으로써, 이러한 이름 지정 규칙에 어긋나는 오브젝트를 생성하는 것이 가능해집니다. 그러나 이후의 사용이 문제점을 발생시킬 수도 있습니다. 예를 들어, 이름에 + 또는 - 부호가 포함된 컬럼을 작성하면, 해당 컬럼을 색인에

사용하여 테이블 재구성시 문제점이 발생합니다. 분리 식별자에 대한 정보는 *SQL* 참조서의 "SQL 식별자" 절을 참조하십시오.

연합 시스템에서 대소문자 구별 값이 보존되는 방법

분산 요청(DR)에서 데이터 소스에 대소문자가 구별되는 식별자 및 암호를 지정해야 합니다. 데이터 소스로 전달될 때 대소문자가 올바른지 확인하려면, 다음 지침을 따르십시오.

- 필요한 대소문자로 식별자 및 암호를 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. 데이터 소스에서 사용자 ID가 소문자로 사용될 필요가 있는 경우, 이 사용자 ID를 어떠한 문자로든지 지정하고 *fold_id* 서버 옵션을 "l"(『이 ID를 소문자로 데이터 소스에 보냅니다.』)로 설정할 수 있습니다. 데이터 소스에서 ID가 대문자로 사용될 필요가 있는 경우, 이 ID를 어떠한 문자로든지 지정하고 *fold_id*를 "u"(『이 ID를 대문자로 데이터 소스에 보냅니다.』)로 설정할 수 있습니다. 같은 방법으로, 데이터 소스에서 암호가 소문자 또는 대문자로 사용될 필요가 있다면, *fold_pw* 서버 옵션을 "l" 또는 "u"로 설정하여 이 요구사항을 충족시킬 수 있습니다.

서버 옵션에 대한 자세한 정보는 *관리 안내서: 구현183* 페이지의 『서버 옵션을 사용한 데이터 소스 정의 지원 및 인증 처리 용이화』에서 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 데이터 소스에 있는 여러 분리 식별자는 대소문자를 구별합니다. NORBASE라고 하는 데이터 소스에 상주하는 CS 뷰, "my_schema". "wkly_sal"의 DB2에 대해 NICK1 별칭을 작성한다고 가정하십시오.

UNIX 기반 시스템의 명령 프롬프트에서, 다음을 입력해야 합니다.


```
db2 'create nickname nick1 for norbase."my_schema"."wkly_sal"'
```

Windows NT 명령 프롬프트에서, 다음을 입력해야 합니다.

```
db2 create nickname nick1 for norbase.\."my_schema"\.\."wkly_sal\"
```

DB2 명령 프롬프트(대화식 모드)에서 명령문을 입력하거나 응용프로그램에서 SQL을 지정할 경우, 작은따옴표 또는 백슬래시를 사용할 필요가 없습니다. 예를 들어, UNIX 기반 시스템 또는 Windows NT의 DB2 명령 프롬프트에서는 다음을 입력해야 합니다.

```
create nickname nick1 for norbase."my_schema"."wkly_sal"
```

부록B. DCE 디렉토리 서비스 사용

DCE는 CDS(Cell Directory Service) 및 GDS(Global Directory Service)를 제 공합니다. DCE 개념 및 서비스에 대한 자세한 정보는 *Introduction to OSF DCE* 매뉴얼을 참조하십시오. DCE 디렉토리 서비스에 대한 DB2 기능은 CDS만을 지 원합니다. 이러한 지원이 있기 때문에, 사용자는 모든 클라이언트에 각 데이터베 이스, 노드 및 DCS 데이터베이스를 작성할 필요가 없습니다. 이 모든 정보는 DCE CDS에 집중됩니다.

다음 절에서는 DCE 디렉토리 서비스를 사용하여 데이터베이스를 설정하고 액세스 하는 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 겹쳐쓰기
- 디렉토리 서비스 타스크
- 디렉토리 서비스 제한사항

DCE 디렉토리 서비스는 모든 DB2 클라이언트에 의해 지원되지 않을 수도 있습 니다. 그러나 DCE 디렉토리 서비스가 DB2 클라이언트를 위해 지원된다면, 빠른 시작 매뉴얼에서 추가 정보를 제공합니다.

디렉토리 오브젝트 작성

데이터베이스 관리자가 작성해야 하는 디렉토리 오브젝트에는 다음 세 가지 유형이 있습니다.

- 『데이터베이스 오브젝트』
- 380 페이지의 『데이터베이스 위치 지정자 오브젝트』
- 381 페이지의 『경로지정 정보 오브젝트』

각각의 오브젝트는 속성을 가지고 있습니다. 속성의 완전한 설명은 383 페이지의 『각 오브젝트 클래스의 속성』을 참조하십시오.

데이터베이스 관리자가 오브젝트를 생성하기 전에, DCE 관리자는 데이터베이스 정보를 CDS 테이블에 추가하고, 데이터베이스 관리자에게 작성 특권을 권한 부여하십시오. 세부사항은 402 페이지의 『DCE 관리자 TASK』를 참조하십시오.

데이터베이스 오브젝트

각각의 목표 데이터베이스에 대해 하나의 데이터베이스 오브젝트가 요구됩니다. 오브젝트는 디렉토리 이름 및 데이터베이스 이름에 연결되는 셀 이름을 포함하는 이름을 가집니다. 예를 들어, 다음과 같습니다.

```
./.../cell_name/dir_name1/dir_name2/OBJ_NAME
```

주: 다음은 데이터베이스 이름에 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 사용해야 합니다. 명령에 대한 세부사항은 393 페이지의 『CATALOG GLOBAL DATABASE 명령』을 참조하십시오.

다음은 데이터베이스 오브젝트의 예입니다. DCE 디렉토리에 저장된 오브젝트는 시간소인과 같은 기타 정보를 갖습니다. 각 속성 왼쪽의 문자는 속성이 반드시 필요한 것인지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

```
Object name:          /.../CELL_TORONTO/subsys/database/AIXDB1
R DB_Object_Type:     D
C DB_Product_Name:    DB2 for AIX
C DB_Product_Release: V7RIM000
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 연관된 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 Connect 인스턴스의 완전한 이름입니다.

오브젝트를 작성하는 데 사용하는 DCE 명령의 예가 있습니다. 어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 402 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 = V7R1M000
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

```

그런 다음, 다음 중 하나를 수행해야 합니다.

- *dcelogin* 핵심부 암호(OS/2) 또는
- *dce_login* 핵심부 암호(UNIX, Windows 운영 체제에서)

이는 다음과 같아야 합니다.

- *cdscp < cdscp.inp*

오브젝트를 표시하려면, 다음 명령을 사용하십시오.

```
cdscp show object /./subsys/database/AIXDB1
```

데이터베이스가 데이터베이스 관리 프로그램 인스턴스와 연관된 유일한 데이터베이스라면, 데이터베이스 오브젝트는 통신 프로토콜 속성 값을 가져야 하고, 데이터베이스 위치 지정자 오브젝트는 공백이어야 합니다. 예를 들어, 다음과 같습니다.

```

Object name:                /.../CELL_TORONTO/subsys/database/MVSDB
R DB_Object type:              D
C DB_Product_Name:            DB2_for_MVS
C DB_Product_Release:         V7R1M00
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

이들 오브젝트에는 데이터베이스 관리 시스템 인스턴스 또는 DB2 Connect 인스턴스로 사용한 모든 통신 프로토콜에 대한 세부사항이 들어 있습니다. 다음에 대해 데이터베이스 위치 지정자 오브젝트가 하나씩 필요합니다.

- 데이터베이스 관리 시스템 및 DB2 Connect의 각 인스턴스
- 둘 이상의 데이터베이스와 연관이 있지만 DB2와는 연관이 없는 각 데이터베이스 관리 시스템 인스턴스
- 데이터베이스 관리 시스템과 연관되지 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 목표로 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 속성 왼쪽의 문자는 속성이 반드시 필요한 것인지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

```

Object name:                /.../CELL_TORONTO/subsys/database/AIX_INST
R DB_Object_Type:              L
C DB_Product_Name:            DB2_for_AIX
C DB_Product_Release:         V7R1M00
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 작성하는 데 사용하는 DCE 명령의 예가 있습니다. 어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 402 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 = V7R1M00
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
```

그런 다음, 다음 중 하나를 수행해야 합니다.

- dcelogin 핵심부 암호(OS/2) 또는
- dce_login 핵심부 암호(UNIX, Windows 운영 체제에서)

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 할지를 알려줍니다. 각각의 목표 데이터베이스에 대해 속성이 존재하는데, 여기에는 사용 가능한 데이터베이스 프로토콜 및 DB2 Connect 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 속성 왼쪽 문자는 속성 또는 속성 내 각 토큰이 반드시 필요한지(R), 선택적인지(O), 주석인지(C) 여부를 표시합니다.

클라이언트 그룹 1은 395 페이지의 그림8의 Client_1, Client_2 및 Client_3입니다.

```

Object name:    /.../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 명시적으로 정의되어 있지 않은 목표 데이터베이스에 액세스하는 데 사용된 공통의 라우터를 식별하는 특수 값입니다.

오브젝트를 작성하는 데 사용하는 DCE 명령의 예가 있습니다. 백슬래시(\) 문자는 계속됨을 나타내는 문자입니다.

어떠한 오브젝트를 작성하더라도, 그 전에 DCE 관리자는 402 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

```

그런 다음, 다음 중 하나를 수행해야 합니다.

- dcelogin 핵심부 암호(OS/2) 또는
- dce_login 핵심부 암호(UNIX, Windows 운영 체제에서)

이는 다음과 같아야 합니다.

- cdscp < cdscp.inp

오브젝트를 표시하려면, 다음 명령을 사용하십시오.

```
cdscp show object ./:/subsys/database/ROUTE1
```

DCE 명령에 대한 자세한 정보는 다음 OSF DCE 책을 참조하십시오.

- OSF DCE 관리 안내서
- OSF DCE 관리 참조서

각 오브젝트 클래스의 속성

DCE 환경에서, 각각의 오브젝트와 오브젝트 속성은 오브젝트 ID(OID)로 식별됩니다. 각각의 OID는 권한 할당의 계층 구조로부터 얻을 수 있는데, 가장 높은 권한은 국제 표준화 기구(ISO)입니다.

표19에서는 각 오브젝트 클래스의 속성을 보여주고, 표20에서는 해당 속성을 보여줍니다.

표 19. 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 20. 오브젝트 클래스 속성

속성 이름	OID	최소 길이	최대 길이	구문
(DAU) DB_Authentication	1.3.18.0.2.4.39	1	1024	Char
(DCO) DB_Comment	1.3.18.0.2.4.30	1	1024	Char

표 20. 오브젝트 클래스 속성 (계속)

속성 이름	OID	최소 길이	최대 길이	구문
(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 다중 값이 허용됩니다. 기타 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. 필수인 속성을 입력했는지와 올바른 값을 입력했는지를 확인하십시오.

DB_Authentication (DAU)

오브젝트가 요구하는 인증 메소드입니다. 이 속성은 DB2 서버의 데이터베이스 오브젝트에 필수입니다. 값은 CLIENT, SERVER 또는 DCE여야 합니다.

DB_Principal (DPR)

인증 메소드가 『DCE』인 경우, 이 속성에 DCE 핵심부를 입력하십시오.

DB_Comment (DCO)

문서 목적용입니다.

DB_Communication_Protocol (DCP)

각각의 값이 지원되는 네트워크 프로토콜을 설명하는 토큰으로 구성된 다중 값의 속성. 네트워크 프로토콜의 예는 TCP/IP, APPC, IPX/SPX 및 NetBIOS입니다. 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 연결 요청을 청취하기 위해 오브젝트가 사용하는 포트 번호
4. (선택적) 보안은 NONE 또는 SOCKS 중 하나일 수 있습니다.

예를 들어, 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. LU의 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 들어, appc;NETID;TARGETLU;TPNAME;MODE;PROGRAM입니다.

주: APPC의 경우, 클라이언트는 LU 이름으로 지역 제어점(CP)을 사용해야 합니다.

- (OS/2 및 지원되는 Windows 운영 체제 전용) IPX/SPX에 대한 토큰은 다음과 같습니다.
1. ipxspx
 2. 파일 서버의 이름
 3. 오브젝트의 이름

예를 들어, ipxspx;SVR_NAME;OBJ_NAME입니다.

- (OS/2 및 지원되는 Windows 운영 체제 전용) NetBIOS에 대한 토큰은 다음과 같습니다.

1. netbios
2. 서버의 노드 이름

예를 들어, netbios;SVR_NNME입니다. 여기서, 클라이언트 어댑터 번호는 레지스트리 값 *db2clientadpt* 또는 데이터베이스 관리 프로그램 구성 매개변수 *dft_client_adpt*에서 찾을 수 있습니다.

- (지원되는 Windows 운영 체제 전용) NPIPE에 대한 토큰은 다음과 같습니다.

1. NPIPE
2. 서버의 컴퓨터 이름
3. 서버의 인스턴스 이름

예를 들어, npipe;computername;instance입니다.

DB_Database_Protocol (DDP)

데이터베이스 프로토콜 또는 목표 데이터베이스에 의해 지원되는 프로토콜. 해당 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 Connect 인스턴스에 대한 것입니다.

예를 들어, /.../CELL_TORONTO/subsys/database/AIX_INST입니다.

DB_Native_Database_Name (DNN)

데이터베이스를 포함하는 인스턴스 내에서 데이터베이스가 알려져 있는 데이터베이스 이름 또는 별명. 이것은 DBMS 인스턴스에서 지역 응용프로그램이 해당 데이터베이스에 연결하기 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. 문서 목적용입니다.

DB_Product_Release (DRL)

제품 릴리스 레벨. 문서 목적용입니다.

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 이름. *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 Connect 인스턴스가 해당 목표 데이터베이스와 통신하는 데 사용하는 데이터베이스 프로토콜(예: DRDA)
3. 라우터로의 인바운드 프로토콜. 경로지정 DB2 Connect 인스턴스 오브젝트가 승인하는 데이터베이스 프로토콜(예: DB2RA)
4. 게이트웨이에서의 인증. 올바른 값은 0 또는 1입니다. 세부사항은 389 페이지의 표21을 참조하십시오.
5. DB2 Connect 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 완전한 이름. SQLCODE 맵핑을 멈추려면 NOMAP를 지정하십시오.
- D. 특정 SQLCODE가 리턴되면 응용프로그램은 DRDA 서버로부터 연결해제됩니다. SQLCODE에 대한 세부사항은 *DB2 Connect 사용자 안내서*를 참조하십시오.
- INTERRUPT_ENABLED. DRDA 서버에 연결되어 있는 동안 클라이언트가 인터럽트를 발행하면, DB2 Connect는 연결을 제거하고 작업 단위(UOW)를 구간 복원합니다.

예를 들어, 다음과 같습니다.

```
NOMAP
/u/username/sqllib/map/dcs1new.map,D
/u/username/sqllib/map/dcs1new.map,D,INTERRUPT_ENABLED
```

다음과 같이 기본값이 사용되는 위치에서 토큰의 순서를 유지하려면, 쉼표를 사용하십시오. 예를 들어, 다음과 같습니다.

```
,D
```

또는

```
,,INTERRUPT_ENABLED
```

매개변수 문자열에 대한 세부사항은 *DB2 Connect 사용자 안내서*를 참조하십시오.

6. 경로지정 서비스를 제공하는 DB2 Connect 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 연속 문자입니다.

디렉토리 서비스 보안

DB2 Connect 게이트웨이가 없는 환경에서 DCE 디렉토리 서비스를 사용할 때, 인증은 데이터베이스 서버에 액세스하는 기타 클라이언트에 사용되는 것과 같은 것입니다. 자세한 정보는 267 페이지의 『서버에 대한 인증 방법 선택』을 참조하십시오.

DB2 Connect 게이트웨이가 없는 환경에서 DCE 디렉토리 서비스를 사용할 때, DB2 Connect 관리자는 사용자 이름과 암호가 어디서 평가되는지 판별합니다. DCE 디렉토리에서, 다음을 지정하십시오.

- 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 연관된 위치 지정자 오브젝트)의 통신 프로토콜의 보안 유형
- 경로지정 정보 오브젝트의 게이트웨이 토큰에서의 인증

표21에서는 이 값으로 가능한 조합을 보여주며, 여기서 APPS를 사용하여 각 조합에 대한 유효성 확인이 수행됩니다. 다음 표에 표시되는 조합은 DCE 디렉토리 서비스를 갖는 DB2 Connect에 의해 지원됩니다.

표 21. APPS 연결을 사용하여 DCE에서의 유효한 보안 시나리오

경우	서버의 데이터베이스 오브젝트		경로지정 오브젝트	유효성 확인
	인증	보안	게이트웨이에서의 인증	
1	CLIENT	SAME	0	원격 클라이언트(또는 DB2 Connect 워크스테이션)
2	CLIENT	SAME	1	DB2 Connect 워크스테이션
3	SERVER	PROGRAM	0	DRDA 서버
4	SERVER	PROGRAM	1	DB2 Connect 워크스테이션 및 DRDA 서버
5	DCE	NONE	NOT APPLICABLE	DCE

표22에서는 이 값으로 가능한 조합을 보여주며, 여기서 TCP/IP를 사용하여 각 조합에 대한 유효성 확인이 수행됩니다. 이 표에 표시되는 조합은 DCE 디렉토리 서비스를 갖는 DB2 Connect에 의해 지원됩니다.

표 22. TCP/IP를 사용하여 DCE에서의 유효한 보안 시나리오

경우	인증	게이트웨이에서의 인증	유효성 확인
1	CLIENT		0 클라이언트
2	CLIENT		1 DB2 Connect 워크스테이션
3	SERVER		0 DRDA 서버
4	NOT APPLICABLE	NOT APPLICABLE	없음
5	DCE	NOT APPLICABLE	DCE

각 조합은 APPC 및 TCP/IP에 모두 적용할 수 있으며, 세부사항은 다음과 같습니다.

1. 사용자 이름과 암호는 원격 클라이언트에서만 유효성이 확인됩니다. (지역 클라이언트의 경우, 사용자 이름과 암호는 DB2 Connect 워크스테이션에서만 유효성이 확인됩니다.)

사용자는 사인온한 위치에서 인증되는 것이 일반적입니다. 사용자 ID를 네트워크를 통해 보내지만 암호는 보내지 않습니다. 모든 클라이언트 워크스테이션이 적절한 보안 기능을 가지고 있는 경우에만 이 유형의 보안을 사용하십시오.

2. 사용자 이름과 암호는 DB2 Connect 워크스테이션에서만 유효성이 확인됩니다. 암호를 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로 보내지만 DRDA 서버로는 보내지 않습니다.
3. 사용자 이름과 암호는 DRDA 서버에서만 유효성이 확인됩니다. 암호를 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로, DB2 Connect 워크스테이션으로부터 DRDA 서버로 보냅니다.
4. 사용자 이름과 암호는 DB2 Connect 워크스테이션과 DRDA 서버 모두에서 유효성이 확인됩니다. 암호를 네트워크를 통해 원격 클라이언트로부터 DB2 Connect 워크스테이션으로, DB2 Connect 워크스테이션으로부터 DRDA 서버로 보냅니다.

유효성 확인이 두 곳에서 수행되기 때문에 사용자 이름과 암호의 같은 세트가 DB2 워크스테이션과 DRDA 서버 두 곳에서 유지보수되어야 합니다.

5. DCE 토큰을 DCE 보안 서버에서 얻습니다.

주:

1. AIX 기반 시스템의 경우, 보안 유형 SAME을 사용하는 모든 사용자는 AIX 시스템 그룹에 속해야 합니다.
2. 원격 클라이언트가 있는 AIX 기반 시스템의 경우, DB2 Connect 워크스테이션에서 수행되는 DB2 Connect 제품의 인스턴스는 AIX 시스템 그룹에 속해야 합니다.
3. DRDA 서버로의 액세스는 자체 보안 메커니즘 또는 보안 서브시스템에 의해 제어됩니다. 예를 들어, VTAM(Virtual Telecommunications Access Method) 및 RACF(Resource Access Control Facility)에 의해 제어됩니다. 보호되는 데이터베이스 오브젝트의 액세스는 SQL의 **GRANT** 및 **REVOKE**문에 의해 제어됩니다.

구성 매개변수 및 레지스트리 변수

다음 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 연결된 데이터베이스 인스턴스 이름입니다. 인스턴스 이름이 ATTACH 명령의 목표로 사용되면, 이 이름은 다음 예에서처럼 8자 이하여야 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 완전한 이름이 될 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 수도 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 선택적 DCE 매개변수입니다. 이 매개변수는 OS/2와 지원되는 Windows 운영 체제에서 NetBIOS 프로토콜에 대한 기본 클라이언트 어댑터 수를 지정합니다. 이 숫자의 유효한 범위는 0-15입니다. 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 겹쳐쓸 수 있습니다.

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 예에서처럼 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 경로 이름이 있거나, 데이터베이스 이름에 9자 이상의 문자 또는 대소문자가 섞여 있는 경우, CATALOG GLOBAL DATABASE 명령을 사용하십시오. 데이터베이스 관리자는 데이터베이스의 DCE 이름과 디렉토리 유형 DCE를 입력합니다.

예를 들어, 다음과 같습니다.

- 경로 이름이 서로 다를 경우 즉, 다음과 같은 경우 *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 경우

```
CATALOG GLOBAL DATABASE
/.../CELL_VANCOUVER/subsys/database/DB_LONGNAME AS VANVMDB
USING DIRECTORY DCE WITH "comment-string"
```

CONNECT문

해당 DCE 디렉토리 오브젝트를 검색하려면, 클라이언트는 데이터베이스 또는 DBMS 인스턴스의 완전한 DCE 이름을 알아야 합니다. CONNECT문에 이름을 지정하는 몇 가지 방법은 다음과 같습니다.

- 다음과 같이 별명을 입력하십시오.

```
CONNECT TO VANVMDB
```

- 다음과 같이 한 부분의 이름을 입력하십시오.

CONNECT TO VMDB

이 경우, 클라이언트에 지정된 경로 이름은 서버에 지정된 경로 이름과 같아야 합니다. (경로 이름은 *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

395 페이지의 그림8에서는 두 가지 DCE 셸을 가진 데이터베이스 네트워크의 샘플 구성을 보여줍니다. `/.../CELL_TORONTO` 및 `/.../CELL_VANCOUVER`은 셸의 이름입니다. (각각의 셸에는 `././subsys/database/`라는 디렉토리가 들어 있으며, 다이어그램에는 없지만 기타 예에서 사용됩니다.)

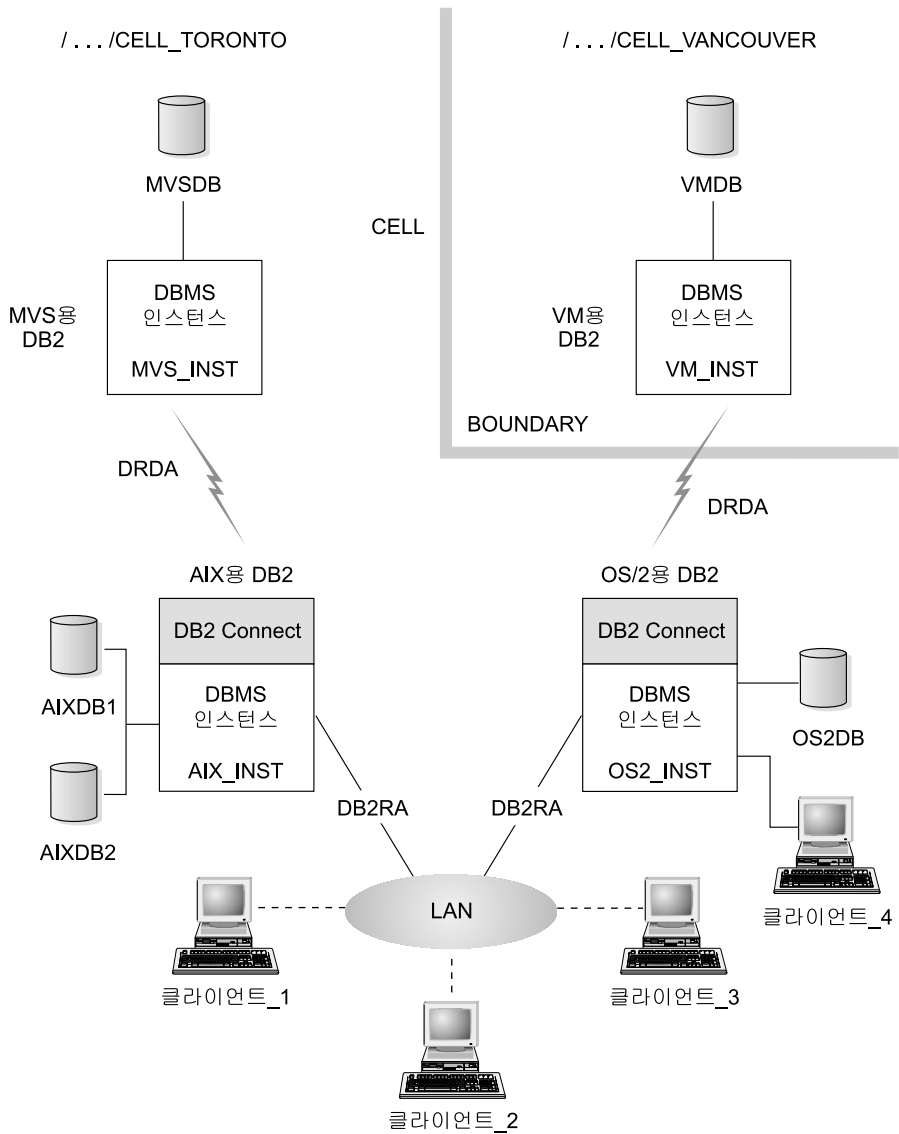


그림 8. 네트워크 데이터베이스의 구성

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 DB2의 두 데이터베이스 서버의 목표 데이터베이스 위치 지정자
- 각각의 클라이언트가 알고 있는 하나의 경로지정 정보 오브젝트. 이 속성은 MVSDDB 및 VMDB 데이터베이스에 대해 어떤 DB2 Connect 노드를 사용할지 지정합니다.

다음은 클라이언트가 데이터베이스에 연결하는 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 연결하는 방법을 보여주는 몇 가지 예를 설명합니다.

1. Client_1을 AIXDB2에 연결하십시오. 데이터베이스는 클라이언트와 같은 디렉토리 경로 이름을 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, 완전한 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. Client_1은 데이터베이스 `../../CELL_TORONTO/subsys/database/AIXDB2`를 사용하여 AIXDB2에 대한 데이터베이스 오브젝트를 확보합니다.

- b. 이 오브젝트에서, 클라이언트_1은 AIXDB2가 클라이언트_1이 사용하는 프로토콜과 같은 DB 프로토콜 DB2RA를 사용한다는 것을 알고 있습니다.
 - c. DB 프로토콜은 일치하는 것이므로, 클라이언트_1은 AIX_INST에 대한 데이터베이스 관리 시스템 위치 지정자 오브젝트를 읽고, 사용하는 통신 프로토콜 속성 값과 일치하는 값을 검색하고, 이 정보를 사용하여 데이터베이스 관리 시스템 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 라우팅 정보 오브젝트의 DCE 이름을 지정하십시오.

이벤트의 순서는 다음과 같습니다.

- a. 클라이언트_3은 데이터베이스 `/.../CELL_TORONTO/subsys/database/MVSDDB`를 사용하여 MVSDDB에 대한 데이터베이스 오브젝트를 확보합니다.
- b. 이 오브젝트에서, 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 연관된 데이터베이스 위치 지정자 오브젝트를 읽고, 통신 프로토콜을 검색하고, 라우터로 SQL CONNECT 요청을 보냅니다.
- e. 그런 다음, 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 이미 구성되어 있습니다.

- DCE 디렉토리 서비스(*dir_type* 매개변수에 DCE라고 지정하여)
- CELL_VANCOUVER 이외의 셀(구성 매개변수 *dir_path_name*을 통한). 예를 들어, 다음과 같습니다.

```
./.../CELL_TORONTO/subsys/database/
```

2. 클라이언트_3이 VMDB로 연결하기 위해서는 데이터베이스 관리자가 다음을 수행해야 합니다.

- VMDB를 지역 시스템 데이터베이스 디렉토리에 명시적으로 카탈로그화하십시오. VMDB에 대한 DCE 이름을 지역적으로 고유한 데이터베이스 별명 과 결합하고, 별명 값이 있는 CONNECT문을 입력하십시오. 예를 들어, 다음과 같습니다.

```
CATALOG GLOBAL DATABASE  
./.../CELL_VANCOUVER/subsys/database/VMDB AS VANVMD  
USING DIRECTORY DCE WITH "comment-string"
```

다음을 입력하십시오.

```
CONNECT TO VANVMD
```

- 구성 매개변수 *dft_client_comm*(또는 DB2CLIENTCOMM 레지스트리 값)에 통신 프로토콜을 지정하십시오.
- 구성 매개변수 *route_obj_name*(또는 DB2ROUTE 레지스트리 값)에 기본 라우팅 정보 오브젝트의 DCE 이름을 지정하십시오.

이벤트의 순서는 다음과 같습니다.

- a. 클라이언트_3은 시스템 데이터베이스 디렉토리에서 완전한 DCE 이름인 VANVMD를 찾습니다.
- b. Client_3은 데이터베이스 `./.../CELL_VANCOUVER/subsys/database/VMDB`를 사용하면 VMDB에 대한 데이터베이스 오브젝트를 확보합니다.
- c. 이 오브젝트에서, 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 레지스 트리 값에 정의된 이름을 사용하여 경로지정 정보 오브젝트를 확보합니다. 클라이언트는 VMDB에 대한 목표 데이터베이스 정보를 발견합니다.
- e. 클라이언트_3은 VMDB 목표 데이터베이스 정보와 연관된 데이터베이스 위치 지정자 오브젝트를 읽고, 통신 프로토콜을 검색하고, 라우터로 SQL CONNECT 요청을 보냅니다.
- f. 그런 다음, 라우터는 VMDB와의 APPC 연결을 설정합니다.

디렉토리 검색 방법

만일 DCE 디렉토리가 모든 목표 데이터베이스가 같은 디렉토리 경로 이름을 공유하는 환경에서 사용되는 경우, 클라이언트에 어떠한 지역 디렉토리도 필요하지 않습니다.

이 절에서는 디렉토리의 다음 사항이 검색되는 순서에 대해 설명합니다.

- ATTACH 명령
- CONNECT문

ATTACH 명령

400 페이지의 그림9에서는 클라이언트가 ABC_INST라는 DBMS 인스턴스에 접속할 때 디렉토리가 검색되는 방법을 보여줍니다.

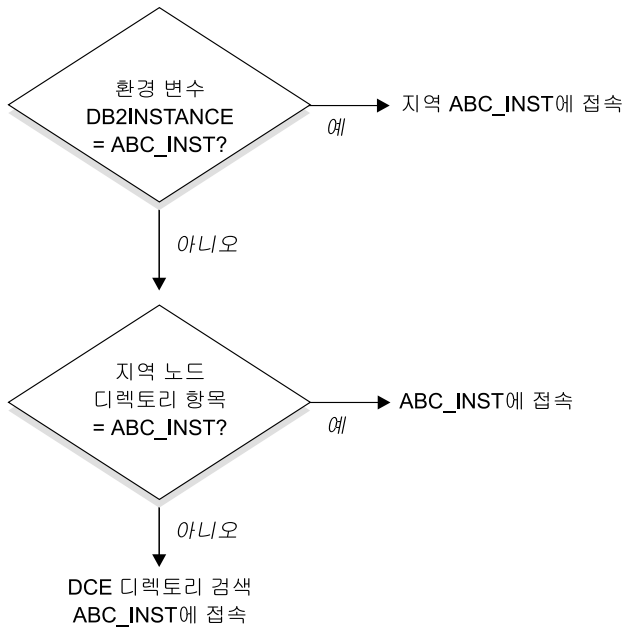


그림 9. 디렉토리가 데이터베이스에 접속하기 위해 사용되는 방법

CONNECT문

401 페이지의 그림10에서는 클라이언트가 DBTEST라는 데이터베이스에 연결될 때 디렉토리를 어떻게 검색할 것인지를 보여줍니다.

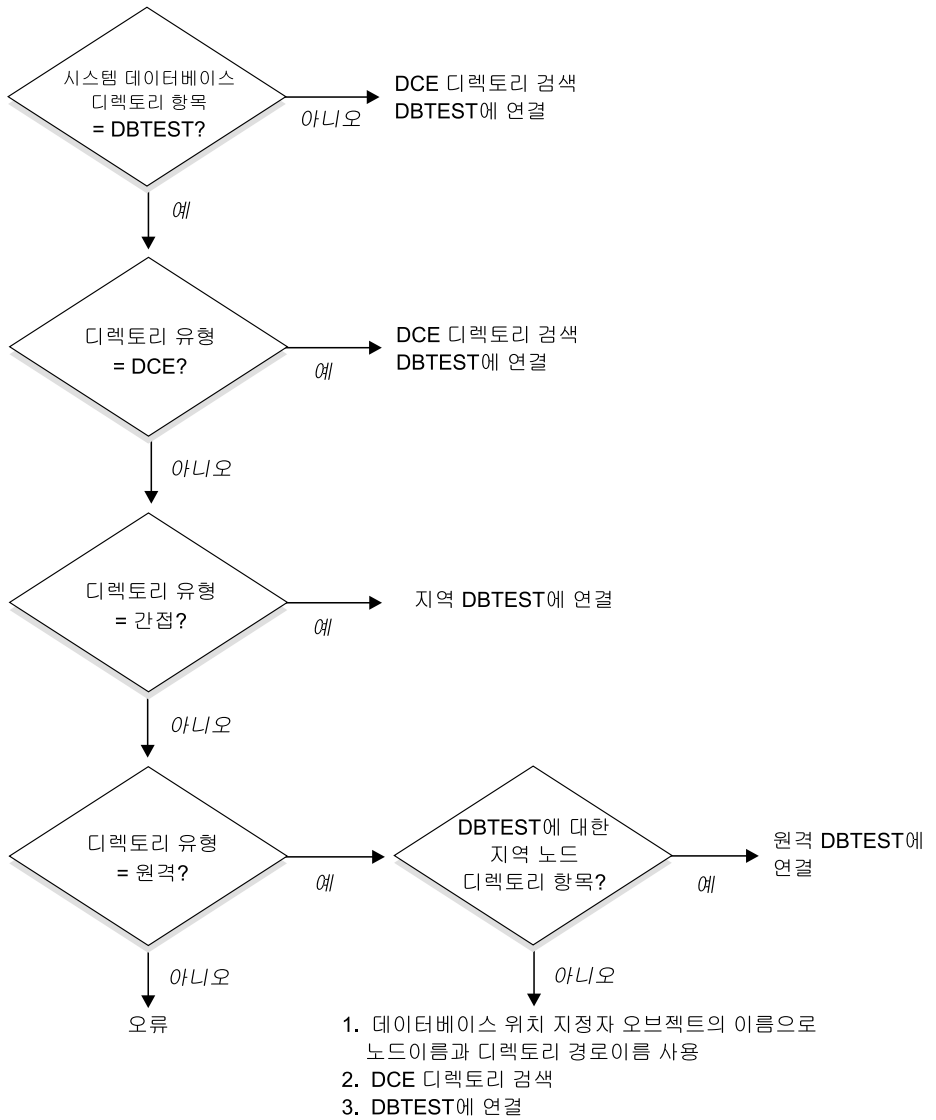


그림 10. 디렉토리가 데이터베이스에 연결하기 위해 사용되는 방법

DCE 디렉토리 정보를 임시로 겹쳐쓰기

DCE 디렉토리 정보를 겹쳐쓰기 위해 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 항목이 있는 원격 데이터베이스로서 DBTEST를 지역적으로 카탈로그화하십시오.

DCE 이름이 클라이언트의 디렉토리 경로 이름을 따르지 않는 데이터베이스에 대해 별명을 작성할 수 있습니다. 명령에 대한 세부사항은 393 페이지의 『CATALOG GLOBAL DATABASE 명령』을 참조하십시오.

디렉토리 서비스 TASK

설정이 수행되고 DCE 디렉토리를 설정 및 사용하기 위해 수행되어야 할 TASK 목록은 다음과 같습니다. 다음에서는 각 TASK에 대한 세부사항을 설명합니다.

- DCE 관리자 TASK

DCE 관리자는 새로운 데이터베이스 자원 정보가 추가될 수 있도록 DCE 디렉토리를 갱신해야 합니다.

- 데이터베이스 관리자 TASK

데이터베이스 관리자는 DCE 디렉토리를 갱신하고 DB2 설치 및 구성에 대한 정보를 제공해야 합니다.

- 데이터베이스 사용자 TASK

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 TASK

DCE 관리자는 디렉토리 오브젝트가 작성되거나 읽히기 전에 다음 TASK를 수행해야 합니다.

- DB2에 대한 디렉토리 서브트리를 지정하십시오(예: `././subsys/database`).
- 디렉토리 오브젝트를 작성하기 위해 데이터베이스 관리자에게 특권을 권한 부여하십시오.
- 디렉토리 오브젝트를 읽기 위해 데이터베이스 사용자에게 특권을 권한 부여하십시오.
- DCE 속성 테이블에 새로운 DCE 디렉토리 오브젝트 속성에 대한 정보를 추가하십시오.

CDS 속성 파일(UNIX 플랫폼에서는 `/etc/dce/cds_attributes`, OS/2에서는 `X:\opt\dcelocal\etc\cds_attr`이며, 여기서 "X"는 해당 드라이브를 나타냄)을 편집한 후 다음을 추가하십시오.

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 액세스해야 할 경우, 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 묻습니다.
- DCE 디렉토리 서브트리에 각각의 데이터베이스 관리 시스템 인스턴스에 대한 고유 이름을 지정하십시오(예: `././subsys/database/AIX_INST`).

- 각각의 데이터베이스 관리 시스템 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 변수로 임시로 겹쳐쓸 수 있습니다. 자세한 정보는 391 페이지의 『구성 매개변수 및 레지스트리 변수』를 참조하십시오.

- DCE 디렉토리 서브트리에 각각의 데이터베이스에 대해 고유 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 작성됩니다. 세 가지 유형의 오브젝트가 있습니다.
 - 각각의 목표 데이터베이스에 대해 하나의 데이터베이스 오브젝트가 요구됩니다.
 - 둘 이상의 데이터베이스와 연관된 각 DB2 Connect 인스턴스 및 각 데이터베이스 관리 시스템 인스턴스(DB2 Connect 없이)에는 데이터베이스 위치 지정자 오브젝트가 필요합니다.
 - 호스트 데이터베이스에 액세스하기 위해 라우팅 정보 오브젝트가 필요합니다.
- 각 환경에 따라, 데이터베이스 관리자는 다음 사항을 판별하십시오.
 - 어떤 데이터베이스에 액세스하고 어떤 통신 프로토콜을 사용하는지를 고려하여 클라이언트를 논리 그룹으로 어떻게 그룹화하는지
 - 경로지정 정보 오브젝트가 몇 개나 필요한지
 - 어떤 목표 데이터베이스가 각각의 오브젝트에 기록되어야 하는지

- 어떤 라우팅 정보 오브젝트가 어떤 그룹의 클라이언트에게 알려져야 하는지 오브젝트에 대한 세부사항은 378 페이지의 『디렉토리 오브젝트 작성』을 참조하십시오.

데이터베이스 사용자 task

데이터베이스 사용자는 다음 task를 수행해야 합니다.

- 데이터베이스 관리자로부터 데이터베이스의 이름을 구하십시오. 이 이름은 간단하게 일부만으로 구성된 이름일 수도 있고, 완전한 DCE 이름일 수도 있습니다.
- 필요한 경우, 레지스트리 변수에 DCE 디렉토리 서비스에 필요한 값을 지정하십시오. 클라이언트가 설정한 레지스트리 변수가 구성 매개변수를 임시로 겹쳐 쓸 수 있습니다.
 - 만일 호스트 데이터베이스 액세스가 필요한 경우, 데이터베이스 관리자로부터 라우팅 정보 오브젝트 항목의 완전한 DCE 이름을 구하십시오. 이 이름이 *route_obj_name*에 지정되어 있지 않거나 다른 이름이면, 호스트 데이터베이스에 연결하기 전에 이 이름을 DB2ROUTE 레지스트리 변수에 지정하십시오.
 - 선호되는 통신 프로토콜이 *dft_client_comm*에 지정되어 있지 않거나 다른 프로토콜인 경우, 클라이언트에 대한 통신 프로토콜을 DB2CLIENTCOMM 레지스트리 변수에 지정하십시오. 다음은 몇몇 UNIX 예입니다.

```
db2set DB2CLIENTCOMM=tcpip
db2set DB2CLIENTCOMM=appc
db2set DB2CLIENTCOMM=tcpip,appc
db2set DB2CLIENTCOMM=appc,tcpip
```

다음은 일부 OS/2 예입니다.

```
db2set DB2CLIENTCOMM=ipxspx
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc
```

일부 Windows 운영 체제는 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, 첫 번째로 지정된 것이 사용됩니다.

- *dir_path_name* 구성 매개변수 또는 DB2DIRPATHNAME 레지스트리 값에 없는 DCE 이름을 가지는 데이터베이스가 있을 경우, CATALOG GLOBAL DATABASE 명령을 사용하여 데이터베이스를 명시적으로 카탈로그화하십시오. 자세한 정보는 393 페이지의 『CATALOG GLOBAL DATABASE 명령』을 참조하십시오.
- 목표 데이터베이스에 연결하거나 데이터베이스 인스턴스에 접속하기 전에 DCE에 로그인하십시오. 로그인 명령에 대한 자세한 정보는 *OSF DCE Administration Guide*를 참조하십시오.

디렉토리 서비스 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 연결하기 위해 DCE 디렉토리 서비스를 사용할 수 없습니다.
- 지원되는 Windows 운영 체제 클라이언트만이 TCP/IP, APPC, NetBIOS, IPX/SPX 또는 NPIPE 프로토콜 일부 또는 모두를 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 제공합니다. DCE의 *cdscp show object* 명령을 사용하여 오브젝트를 표시할 수 있습니다.
- 다음 모든 조건이 존재하는 경우, 데이터베이스 관리 프로그램을 시작하기 전에 데이터베이스 관리 프로그램 인스턴스의 소유자(db2start 명령을 사용하여) DCE에 로그인해야 합니다.

- 데이터베이스 관리 프로그램 인스턴스는 *dir_type* 구성 매개변수를 통해 DCE 디렉토리 서비스를 지원하도록 구성됩니다.
- 셀 디렉토리 서비스는 DCE에 명시적으로 로깅됨으로써 읽을 수만 있습니다.
- 다음 중 하나를 지원하기 위해서는 DCE 디렉토리에 액세스해야 합니다.
 - 또다른 인스턴스에 위치해 있는 트랜잭션 관리 프로그램 데이터베이스 (*tm_database* 구성 매개변수에 의해 지정된)
 - DCE 디렉토리 서비스를 지원하지 못하거나 DCE 디렉토리 서비스를 사용하지 않도록 구성되어 있지 않은 클라이언트

주: DCE 로그인을 수행할 때, long ticket lifetime을 갖는 핵심부를 사용하십시오.

- DDCS 버전 2.2(또는 이전 버전) 게이트웨이를 사용하여 DCE 디렉토리 서비스를 사용하는 클라이언트를 DRDA 서버에 연결할 때, 게이트웨이의 지역 디렉토리에 데이터베이스 별명을 카탈로그화해야 합니다. 이 데이터베이스 별명은 클라이언트에서의 별명과 같아야 하며 동일한 데이터베이스를 나타내야 합니다.
- 지원되는 Windows 운영 체제 클라이언트를 사용하는 경우, DB2DCE.DLL이 사용됩니다. 이 파일은 *sllib* 서브디렉토리의 *bin* 서브디렉토리에 있습니다. DCE 제공업체가 Gradient**이면, 기본적으로 파일 DB2DCE.GRD는 DB2DCE.DLL과 같습니다. DCE 제공업체가 IBM이면, 파일 DB2DCE.IBM을 DB2DCE.DLL로 복사해야 합니다.

부록C. 데이터베이스 복구를 위한 User Exit

데이터 이동의 장에 있는 것과 유사한 개요 정보를 제공하십시오.

주: 이들 주제에 대한 모든 정보와 *Command Reference* 및 *Administrative API Reference*의 비교 가능한 주제는 데이터 복구 및 고가용성 안내 및 참조서에 통합되어 있습니다.

데이터 복구 및 고가용성 안내 및 참조서는 이들 주제에 대한 1차 단일 정보 소스입니다.

부록D. 다중 데이터베이스 파티션으로 명령 발행

파티션된 데이터베이스 시스템에서, 인스턴스의 머신 또는 데이터베이스 파티션 서버(노드)에서 수행되도록 명령을 발행할 수 있습니다. 이 경우, **rah** 명령 또는 **db2_all** 명령을 사용하십시오. **rah** 명령을 사용하면 인스턴스의 머신에서 수행되도록 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 방법에서 차이를 나타냅니다.
2. Windows NT에서, **rah** 명령 또는 **db2_all** 명령을 수행하려면 관리자 그룹의 구성원이 되는 사용자 계정으로 로그인하십시오.

명령의 영역을 판별하려면, *Command Reference*를 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 머신의 모든 논리 노드(데이터베이스 파티션 서버)에서 수행되는 **db2trc** 명령입니다. 모든 머신의 모든 논리 노드에서 **db2trc**를 수행하려면 **rah**를 사용하십시오.

명령

하나의 데이터베이스 파티션 서버에서 또다른 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 머신에 출력을 표시할 수 있습니다. Windows NT에서, 명령을 병렬로 수행하면, 출력은 명령이 발행된 머신에 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 구분할 수 있습니다. Windows NT에서, 앰퍼샌드(&)를 사용하여 다중 명령을 구분하십시오. 마지막 명령 다음에는 분리문자를 사용하지 않도록 하십시오.

다음 예에서는 노드 구성 파일에 지정된 모든 데이터베이스 파티션의 데이터베이스 구성을 변경하기 위해 **db2_all** 명령을 사용하는 방법을 보여줍니다. 다음과 같이 문자가 큰따옴표로 묶여 있으므로, 요청은 동시에 수행됩니다.

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ=100"
```

명령 설명

다음 명령을 사용할 수 있습니다.

명령	설명
----	----

rah	모든 머신에서 명령을 수행합니다.
------------	--------------------

db2_all	지정한 모든 데이터베이스 파티션 서버에서 명령을 수행합니다.
----------------	-----------------------------------

db2_kill	다중 데이터베이스 파티션 서버에서 수행 중인 프로세스를 즉시 중지시키고, 모든 데이터베이스 파티션 서버의 자원을 모두 정리합니다. 이 명령은 데이터베이스의 불일치를 나타냅니다. IBM 서비스의 지시가 있는 경우를 제외하고, 이 명령을 발행하지 마십시오.
-----------------	---

db2_call_stack	
-----------------------	--

UNIX 기반 플랫폼에서, 모든 데이터베이스 파티션 서버에서 수행 중인 프로세스가 syslog에 call traceback을 작성하게 합니다.

Windows NT에서, 모든 데이터베이스 파티션 서버에서 수행 중인 프로세스가 인스턴스 디렉토리의 Pxxx.nnn 파일에 call traceback을 작성하는데, 여기서 Pxxx는 프로세스 ID이고 nnn은 노드 번호입니다.

UNIX 기반 플랫폼에서, 이들 명령은 다음과 같이 내재된 특정 설정이 **rah**를 실행합니다.

- 모든 머신에서 병렬 수행
- 각각 /tmp/\$USER/db2_kill과 /tmp/\$USER/db2_call_stack에서의 버퍼 명령 출력

Windows NT에서, 이들 명령을 사용하면 모든 머신에서 **rah**가 병렬로 실행됩니다.

수행할 명령 지정

다음 방법으로 명령을 지정할 수 있습니다.

- 명령행에서 매개변수로서
- 매개변수를 지정하지 않을 경우, 프롬프트에 응답

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 묶어야 합니다.

주: UNIX 기반 플랫폼에서, 명령은 프롬프트에 입력한 것처럼 명령 실행기록에 추가해야 됩니다.

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 입력해야 합니다.

주:

1. UNIX 기반 플랫폼에서, 명령 셸이 작은따옴표가 붙은 문자열 내에 작은따옴표를 입력하는 방법을 제공하지 않는 한, 명령에 작은따옴표(')를 포함시킬 수 없습니다.
2. Windows NT에서, 명령 창이 작은따옴표가 붙은 문자열 내에 작은따옴표를 입력하는 방법을 제공하지 않는 한, 명령에 작은따옴표(')를 포함시킬 수 없습니다.

백그라운드의 stdin으로부터 읽기에 논리를 포함하는 임의의 korn-shell 셸 스크립트를 수행할 때, 프로세스가 터미널에서 중지하지 않고 읽을 수 있는 소스로 stdin의 경로를 명시적으로 재지정해야 합니다(SIGTTIN 메시지). stdin의 경로를 재지정하려면, 다음과 같은 양식으로 스크립트를 수행할 수 있습니다.

```
shell_script </dev/null &
```

제공되는 입력이 없는 경우

비슷한 방법으로, 백그라운드에서 db2_all을 수행할 때는 </dev/null을 항상 지정해야 합니다. 예를 들어, 다음과 같습니다.

```
db2_all ";run_this_command" </dev/null &
```

이렇게 함으로써 stdin의 경로를 재지정할 수 있고 터미널에서 중지되는 것을 방지할 수 있습니다.

사용자가 원격 명령으로부터의 출력에 무관할 때, 이 메소드의 대체는 다음과 같이 db2_all 접두부의 『daemonize』 옵션을 사용하는 것입니다.

```
db2_all ";daemonize_this_command" &
```


UNIX 기반 플랫폼에서 병렬에 있는 명령 수행

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.

기본적으로, 명령은 각 머신에서 순차적으로 수행되지만 특정 접두부 순서를 가지도록 명령에 접두부를 추가함으로써, 백그라운드 rshell을 사용하여 명령이 병렬로 수행되도록 지정할 수 있습니다. rshell이 백그라운드에서 수행되고 각 명령이 원격 머신의 버퍼 파일에 출력을 추가하는 경우, 이 프로세스는 다음 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 보내는 명령에 접두부로 추가 명령을 첨부하여 \$RAHBUFDIR 및 \$RAHBUFNAME을 버퍼 파일에 사용할 수 있는지 검사합니다. 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 머신에서 명령을 동시에 수행하기 전에 다음을 확인하십시오.

- 디렉토리 /tmp/\$USER이 각 머신의 ID에 대해 있는지 확인하십시오. 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 머신 ID가 **rah**를 수행하는 ID에 대한 `.rhosts` 파일에 항목을 갖는지와 **rah**를 수행하는 ID가 원격 명령을 수행하는 각 머신 ID에 대한 `.rhosts` 파일에 항목을 갖는지 확인하십시오.

UNIX 기반 플랫폼에서 **rah** 프로세스 모니터링

주: 이 절의 정보는 UNIX 기반 플랫폼에만 적용됩니다.

아직 수행 중인 원격 명령이 있거나 버퍼된 출력이 아직 누적되고 있는 경우, **rah**에 의해 시작된 프로세스는 다음 활동을 모니터링합니다.

- 명령이 수행되지 않았음을 표시하는 터미널에 메시지 작성
- 버퍼링된 출력 검색

정보 메시지는 환경 변수 `RAHWAITTIME`으로 제어되어 일정 간격으로 작성됩니다. 이를 지정하는 방법에 대한 세부사항은 도움말 정보를 참조하십시오. 모든 정보 메시지는 `RAHWAITTIME=0`을 내보내면 전혀 표시되지 않습니다.

1차 모니터링 프로세스는(`ps` 명령에 의해 표시된 대로) 명령 이름이 **rahwaitfor**이 됩니다. 첫 번째 정보 메시지에서 프로세스의 `pid`(프로세스 id)를 가리킵니다. 다른 모든 모니터링 프로세스는 **rah** 스크립트를 수행하는 **ksh** 명령(또는 기호 링크 이름)으로 나타납니다. 원하는 경우, 다음 명령으로 모든 모니터링 프로세스를 중지시킬 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, 버퍼링된 출력의 자동 표시를 방해합니다. **rah**의 단일 실행 중에 다른 시간에 실행되는 다른 모니터링 프로세스가 둘 이상 있을 수 있습니다. 그러나 현재 설정을 중지시키려 할 때에는 더 이상의 프로세스가 시작되지 않습니다.

정기적인 로그인 셸이 Korn 셸(예: `/bin/ksh`)이 아니면 **rah**를 사용할 수 있으나, 다음 특수 문자를 포함하는 명령을 입력하는 방법에는 약간 다른 규칙이 적용될 수 있습니다.

```
" unsubstituted $ "
```

자세한 정보는 rah "?"를 입력하십시오. 또한, UNIX 기반 환경에서 원격 명령을 실행하는 ID에서의 로그인 셸이 Korn 셸이 아니면 rah를 실행하는 ID에서의 로그인 셸도 Korn 셸이면 안 됩니다. (rah는 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 rah가 대형 시스템에서 tree_logic을 사용하도록 확장되었습니다. 즉, rah는 목록에 있는 노드 수를 검사하여 해당 수가 임계값을 초과할 경우, 목록의 부속 집합을 구성하여 이들 노드에 순환 호출을 보냅니다. 이들 노드에서 순환적으로 호출된 rah는 목록에 있는 모든 노드에 명령을 보내는 표준 논리(이제 "leaf-of-tree" 논리)를 따라갈 만큼 목록이 작아질 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 노드 간 전송도 감소시킵니다. (이 점은 db2_all에만 적용되며 rah에는 적용되지 않는데, 그 이유는 rah가 항상 구별된 물리 노드에만 보내기 때문입니다.)

접두부 순서

접두부 순서는 한 개 이상의 특수 문자로 되어 있습니다. 공백을 개입시키지 말고 명령 문자 바로 앞에 한 개 이상의 접두부 순서를 입력하십시오. 둘 이상의 순서를 지정하기 위해 임의의 순서로 입력을 수행할 수 있으나, 다중 문자 순서 내의 문자는 순서대로 입력하십시오. 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
	백그라운드에서 순서대로 명령을 수행합니다.
&	일부 프로세스가 여전히 수행 중이더라도 백그라운드에서 명령을 순서대로 수행하고, 모든 원격 명령이 완료된 이후 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 머신에 다시 작성합니다. 주: UNIX 기반 플랫폼에서, &를 지정하면 많은 rsh 명령이 요구되므로 성능 저하를 가져옵니다.
	백그라운드에서 명령을 병렬로 수행합니다.
&	백그라운드에서 명령을 병렬로 수행하고 위의 & 경우 대한 설명대로 모든 원격 명령이 완료된 후에 명령을 종료합니다. 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 비해 성능 저하를 가져옵니다.
]]	명령 수행 전에 사용자의 프로파일이 dot 실행을 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.

' 머신으로 명령 호출을 반향합니다.

< 이 시스템을 제외한 모든 시스템으로 보냅니다.

<<-nnn< 데이터베이스 파티션 서버 *nnn*을 제외한 전체(노드 번호 *nnn*을 제외한 db2nodes.cfg의 모든 데이터베이스 파티션 서버, 이 테이블에서 최종 접두부 다음 첫 번째 단락 참조)로 보냅니다.

<<+nnn< 데이터베이스 파티션 서버 *nnn*(노드 번호가 *nnn*인 db2nodes.cfg의 데이터베이스 파티션 서버, 이 테이블에서 최종 접두부 순차 다음 첫 번째 단락 참조)에만 보냅니다.

(blank character)

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 순서를 지정한 경우 명령을 작은따옴표로 묶거나, 큰따옴표로 묶고 앞에 \를 사용하십시오. 예를 들어, 다음과 같습니다.

```
rah ' ; mydaemon'
```

또는

```
rah " ; \ mydaemon"
```

rah 명령은 백그라운드 프로세스로 수행할 경우, 출력이 리턴되기를 기다리지 않습니다.

> <>를 머신 이름으로 대체하십시오.

" () 대신 머신 색인으로 사용하고, ## 대신 노드 번호를 사용합니다.

주:

1. 머신 색인은 데이터베이스 시스템의 머신과 연관된 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 421 페이지의 『머신 목록에서 중복 항목 제거』를 참조하십시오.

<<-nnn< 및 <<+nnn< 접두부 순서를 사용할 경우, *nnn*은 *db2nodes.cfg* 파일의 *nodenum* 값과 일치하는 1, 2 또는 3자리 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 RAHOSTFILE을 내보내거나(UNIX 기반 플랫폼) 이 변수를 설정하여(Windows NT) 머신 목록을 포함하는 파일의 경로 이름 지정
- 환경 변수 RAHOSTLIST를 내보내거나(UNIX 기반 플랫폼) 이 변수를 설정하여(Windows NT) 공백으로 구분된 일련의 이름으로 목록을 명시적으로 지정

주: 이들 환경 변수가 모두 지정되면 RAHOSTLIST가 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.

머신 목록에서 중복 항목 제거

한 머신의 다중 논리 노드(데이터베이스 파티션 서버)에서 DB2 Enterprise - Extended Edition을 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 지정하십시오. 머신을 지정하려면 **rah** 명령을 사용하십시오. 논리 노드를 지정하려면 **db2_all** 명령을 사용하십시오.

주: UNIX 기반 플랫폼에서 머신을 지정하는 경우, **rah**는 보통 다음 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 경로는 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 대한 정보는 417 페이지의 『접두부 순서』를 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 보내는 것으로 간주되므로, 중복된 항목이 제거되지 않습니다.

rah 명령 제어

다음 환경 변수를 사용하여 **rah** 명령을 제어할 수 있습니다.

표 23.

이름	의미	기본값
\$RAHBUFDIR 주: UNIX 기반 플랫폼에서만 사용 가능합니다.	버퍼 디렉토리	/tmp/\$USER
\$RAHBUFNAME 주: UNIX 기반 플랫폼에서만 사용 가능합니다.	버퍼 파일 이름	rahout
\$RAHOSTFILE (UNIX 기반 플랫폼); RAHOSTFILE (Windows NT)	호스트 목록이 들어 있는 파일	db2nodes.cfg
\$RAHOSTLIST (UNIX 기반 플랫폼); RAHOSTLIST (Windows NT)	문자열로 된 호스트 목록	\$RAHOSTFILE에서 읽음
\$RAHCHECKBUF 주: UNIX 기반 플랫폼에서만 사용 가능합니다.	"no"에 설정되면, 생략 검사	설정 안됨

표 23. (계속)

이름	의미	기본값
\$RAHSLEEPTIME (UNIX 기반 플랫폼); RAHSLEEPTIME (Windows NT)	스크립트가 병렬로 수행되는 명령으로부터 초 기 출력을 기다리는 시간(초)	db2_kill 의 경우는 86400초, 기타 모든 경우 는 200초
\$RAHWAITTIME (UNIX 기반 플랫폼); RAHWAITTIME (Windows NT)	Windows NT에서 원격 작업이 아직 수행 중 인 연속되는 검사 간의 간격(초)	45초
	UNIX 기반 플랫폼에서, 원격 작업이 아직 수 행 중임과 rah: <pid> 기다리는 중... 메 시지가 나타나는 것을 확인하는 연속되는 검 사 간의 간격(초)	
	모든 플랫폼에서, 양의 정수를 지정하십시오. 메시지 출력을 제어하기 위해 사용되는 0으로 시작하는 접두부 값의 경우, RAHWAITTIME=045를 내보내십시오.	
	rah 는 작업 완료 상태를 검출하기 위해 이 검 사를 따르지 않으므로 낮은 값을 지정할 필요 는 없습니다.	
\$RAHENV 주: UNIX 기반 플랫 폼에서만 사용 가능합 니다.	\$RAHDOTFILES=E, K, PE 또는 B이면 실 행 가능한 파일 이름을 지정하십시오.	\$ENV
\$RAHUSER(UNIX 기반 플랫폼); RAHUSER(Windows NT)	UNIX 기반 플랫폼에서, 원격 명령이 수행될 때의 사용자 ID	\$USER
	Windows NT에서, DB2 원격 명령 서비스와 연관된 로그인 계정	

주: UNIX 기반 플랫폼에서, 원격 셸이 설정한 값(있는 경우)이 아닌 **rah**가 수행되고 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 환경 변수를 설정하려면, 다음과 같이 .cshrc와 상응하는 Korn 셸 INSTHOME/.profile을 작성하고 INSTHOME/.cshrc에 지정해야 합니다.

```
setenv RAHDOTFILES P
```

또는 .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 작성되지 못하게 해야 합니다. 다음과 같이 stdout에 작성하는 행을 넣어 확인할 수 있습니다.

```
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)

문제점의 원인은 다음과 같습니다.

- **rah**는 출력을 버퍼링할 필요가 있으며 RAHCHECKBUF=no를 내보내지 않았음을 판별했습니다. 따라서, 명령을 수행하기 전에 **rah**는 모든 머신에 명령을 보내어 버퍼 디렉토리가 있는지 검사하고, 없는 경우 이를 작성합니다.
- 명령을 보내고 있는 하나 이상의 머신이 응답하지 않습니다. **rsh** 명령은 결과적으로 시간종료되지만, 시간종료 간격은 보통 60초로 매우 깁니다.

2. 다음과 같은 메시지가 수신되었습니다.

- 올바르지 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**가 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 연관된 모든 프로세스를 자체 종료할 때까지 프로세스는 정상적으로 지속됩니다. **db2_kill**

의 경우, 이것은 모든 데이터베이스 관리 프로그램의 종료를 의미합니다. 명령이 **rahwaitfor** 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 대해 다음 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_all "....."
```

어떤 방법을 사용하든지 디스크 공간에 한계가 있을 경우, 어떤 지점에서 버퍼 파일을 정리해야 하는지 확인해야 합니다. **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 메시지를 수신합니다.

```
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 전제조건 없이, **rah**는 있는 그대로 () 및 ##을 남겨둡니다. 명령 **print from ()**이 유효하지 않으므로 오류가 수신됩니다.

명령을 병렬로 수행할 때 성능 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 대한 자세한 정보는 369 페이지의 『부록A. 이름 지정 규칙』을 참조하십시오.

사용자는 지역 머신에 로그인할 수 있거나, 머신이 Windows NT 도메인에 설치될 경우 도메인에 로그인할 수 있습니다. Windows NT용 DB2는 이 옵션 둘 다를 지원합니다. 사용자를 인증하기 위해, DB2는 먼저 지역 머신을 검사한 후 현재 도메인에 대한 도메인 제어를 검사하고, 마지막으로 도메인 제어기에 알려진 신뢰성 있는 도메인을 검사합니다.

Windows NT용 DB2가 작동하는 방법을 보려면, DB2 인스턴스가 서버 인증을 요구한다고 가정하십시오. 구성은 다음과 같습니다.

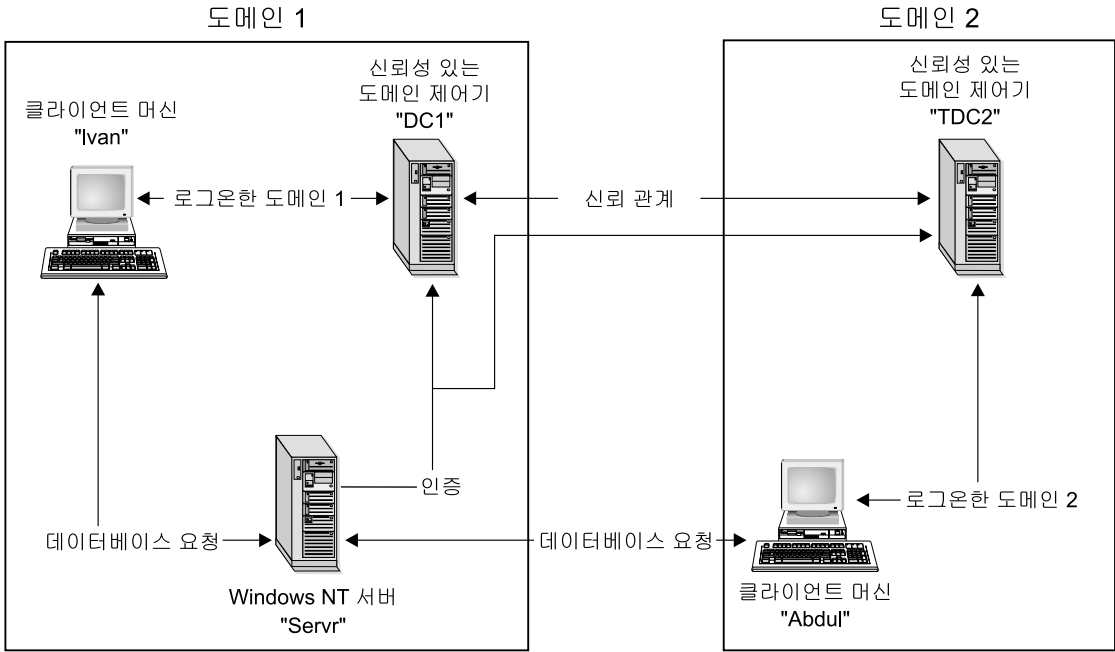


그림 11. Windows NT 도메인 사용 인증

각 머신은 클라이언트 머신에서 Windows 9x가 수행되지 않는 한, 보안 데이터베이스인 보안 액세스 관리(SAM) 데이터베이스가 있습니다. Windows 9x 머신에는 SAM 데이터베이스가 없습니다. DC1은 클라이언트 머신, Ivan 및 Windows NT 서버인 Servr용 DB2가 등록되는 도메인 제어기입니다. TDC2는 DC1과 클라이언트 머신인 Abdul용 신뢰성 있는 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 신뢰성 있는 도메인을 시도하기 전에 먼저 지역 머신(SRV3)을 검색한 후

도메인 제어기(DC1)를 검색합니다. 사용자 이름 Abdul이 TDC2에서 발견됩니다. 이 검색 순서에서는 사용자 또는 그룹에 대해 하나의 이름 공간이 필요합니다.

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.

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```

2. Windows 클라이언트 머신에서 Ivan은 DC1 도메인에 로그인합니다. (즉, DC1 SAM 데이터베이스에 알려줍니다.)
3. 그런 다음, Ivan을 SRV3에 상주시키기 위해 키탈로그에 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 신뢰성 있는 도메인을 시도하기 전에 먼저 지역 머신(Ivan)을 검색한 후 도메인 제어기(DC1)를 검색합니다. 사용자 이름 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.

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```

2. Windows 95 클라이언트 머신에서 Ivan은 DC1 도메인에 로그인합니다. (즉, DC1 SAM 데이터베이스에 알려줍니다.)
3. 그런 다음, Ivan을 SRV3에 상주시키기 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 제공업체에 대한 액세스 권한을 가지고 있으면, 일부 보안 측정은 유효성 확인된 통과 로그인에 대해 Windows 95를 구성하여 부과될 수 있습니다. 이 방법으로 Windows 95 시스템을 구성하는 방법에 대한 세부사항은 Windows 95용 Microsoft 문서를 참조하십시오.

DB2는 또한 전역 그룹도 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.

*DB2DMNBCKCTL*R 레지스트리 변수를 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 DB2로 사용자 인증

운영 체제 인증 방법 때문에 사용자 인증은 Windows NT 사용자에게 문제점을 야기할 수 있습니다. 이 절은 Windows NT용 DB2하에서 사용자 인증에 대한 고려사항을 설명합니다.

- 『사용자 이름 및 그룹 이름 제한사항』
- 『Windows NT용 DB2 보안 서비스』
- 『백업 도메인 제어기에서 DB2 설치』
- 436 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 NT하의 사용자 이름은 대소문자가 구별되지 않지만, 암호는 대소문자가 구별됩니다.
- 사용자 이름 및 그룹 이름은 대문자 및 소문자의 조합일 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 처리기에서 명령을 발행하거나, 씨드 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 대형 분산 LAN에서 매우 중요합니다. 그런 다

음, 사용자는 인증을 위해 기본 도메인 제어기(PDC)에 호출을 요청하는 대신 해당 사이트의 백업 도메인 제어기에서도 인증될 수 있습니다.

이 경우, 백업 도메인 제어기를 갖는 이점은 사용자가 더 빠르게 인증되며 LAN은 BDC가 없었을 때처럼 혼잡하지 않습니다.

다음 조건하에서 BDC에서 인증이 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 조건하에서 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 사용 중이며 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 보존하기 위해 수행 중이어야 합니다. 대형 LAN은 수행 중인 컴퓨터 브라우저가 아닐 수 있으므로 서버 관리 프로그램의 정보가 현재의 것이 아닐 수 있습니다. 이 경우, 백업 도메인 제어기에서 Windows NT용 DB2가 인증하도록 알리는 두 번째 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. (머신이 백업 도메인 제어기로서 설정되어 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 그룹 및 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_GRP_LOOKUP=local
```

– 인스턴스 설정의 경우

```
db2set -i DB2_GRP_LOOKUP=local
```

설정된 모든 DB2 프로파일 레지스트리 변수를 보려면, 다음을 입력하십시오.

```
db2set -all
```

Windows NT용 DB2가 도메인 보안으로 작업하게 하려면, 권한 및 특권을 지역 그룹에 부여하십시오. 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 머신 또는 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 신뢰성 있는 도메인에서 찾으려고 시도합니다.

다음 예는 Windows NT용 DB2가 어떻게 도메인 보안을 지원하는가를 나타냅니다. 이 첫 번째 예에서, 사용자 이름 및 지역 그룹은 동일한 도메인에 있으므로 연결이 작동합니다. 두 번째 예에서, 사용자 이름 및 지역 또는 전역 그룹은 다른 도메인에 있으므로 연결이 작동하지 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 수행 중이지만 서로 동일한 도메인에 있어야 하는 도메인에 정의될 필요가 없음을 기억하십시오.

표 24. 도메인 제어를 사용한 성공적 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 24. 도메인 제어를 사용한 성공적 연결 (계속)

Domain1	Domain2
지역 또는 전역 도메인이 스캔되지만 id2가 없습니다. 도메인 보안이 스캔됩니다.	
	사용자 이름 id2가 이 도메인에 있습니다. DB2는 이 사용자 이름에 대한 추가 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 25. 도메인 제어를 사용한 실패 연결

Domain1	Domain2
Domain2와의 신뢰 관계가 있습니다.	<ul style="list-style-type: none"> • Domain1과의 신뢰 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 스캔되며 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 25. 도메인 제어를 사용한 실패 연결 (계속)

Domain1	Domain2
<p>지역 또는 전역 그룹은 Domain2에 있으며 실제 사용자 이름은 Domain1에 정의되어 있으므로 연결이 작동하지 않습니다.</p> <p>다음과 같은 명령이 대신 발행된 경우 연결이 작동합니다. GRANT CONNECT ON db TO GROUP grp1</p>	

부록F. Windows NT 성능 모니터 사용

Windows NT용 DB2 사용자가 사용할 수 있는 두 가지 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 눌러 ‘**시작하기**’ 온라인 도움말을 참조하십시오.)

- **Windows NT 성능 모니터**

Windows NT 성능 모니터는 시스템에 등록된 모든 성능 데이터 제공업체로부터 정보를 검색하여 데이터베이스 및 시스템 성능을 모니터할 수 있게 합니다. 또한 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 등록하십시오. 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)을 설치 및 등록하려면, 다음을 입력하십시오.

```
db2perfi -i
```

DLL을 등록하면 레지스트리의 서비스 옵션에 새로운 키를 작성합니다. 한 항목은 카운터 지원을 제공하는 DLL의 이름을 지정하고, 다른 세 항목은 해당 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 언로드될 때 호출됩니다.

DB2 성능 정보에 원격 액세스 사용

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 등록하십시오. (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 DB2 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. 일단 사용자 이름과 암호 조합이 DB2에 등록되고 나면, 성능 모니터의 지역 인스턴스조차 해당 사용자 이름과 암호 사용에 대해 명시적으로 로그인합니다. 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. 등록 해제하려면, 다음을 입력하십시오.

```
db2perfr -u <username> <password>
```

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. 모니터 중인 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 성능 정보만 가져올 수 있습니다. 기본적으로, 기본 노드의 성능 정보(즉,

논리 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 활동 중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 응용프로그램**

이 오브젝트는 특정 DB2 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 DB2 응용프로그램에 대해 사용 가능합니다.

- **DB2 DCS 데이터베이스**

이 오브젝트는 특정 DCS 데이터베이스에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 DCS 응용프로그램**

이 오브젝트는 특정 DB2 DCS 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 활동 중인 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 그리고 어느 응용프로그램이 활동 중인지에 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 활동 중이라면, DB2 데이터베이스 및 DB2 응용프로그램 오브젝트가 마찬가지로 나열됩니다. Windows NT 시스템을 DB2 Connect 게이트웨이로 활동 중이고 일부 DCS 데이터베이스와 응용프로그램이 현재 활동 중이면, DB2 DCS 데이터베이스와 DB2 DCS 응용프로그램 오브젝트가 나열됩니다.

원격 DB2 성능 정보 액세스

DB2 성능 정보에 대한 원격 액세스 사용에 대해서는 이미 앞에서 언급되었습니다. 추가 대상(Add to) 상자에서, 또다른 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, **db2perfc** 프로그램을 사용하십시오. 다음을 입력하십시오.

```
db2perfc
```

기본적으로, 이 명령은 활동 중인 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 들어, 다음과 같습니다.

```
db2perfc
db2perfc dbalias1 dbalias2 ... dbaliasn

db2perfc -d
db2perfc -d dbalias1 dbalias2 ... dbaliasn
```

첫 번째 예는 활동 중인 모든 DB2 데이터베이스에 대한 성능 값을 재설정합니다. 그 다음 예는 특정 DB2 데이터베이스에 대한 값을 재설정합니다. 세 번째 예는 활동 중인 모든 DB2 DCS 데이터베이스에 대한 성능 값을 재설정합니다. 마지막 예는 특정 DB2 DCS 데이터베이스에 대한 값을 재설정합니다.

db2perf 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 `sqlmrset`가 있습니다. 자세한 정보는 *Administrative API Reference*를 참조하십시오.

부록G. Windows 2000 데이터베이스 파티션 서버 또는 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』
- 448 페이지의 『인스턴스에 데이터베이스 파티션 서버 추가』
- 450 페이지의 『데이터베이스 파티션 변경』
- 451 페이지의 『인스턴스에서 데이터베이스 파티션 제거』

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 시작 중, 수행 중, 중지 중 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: 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. 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.

또한 여러 개의 선택적 매개변수도 있습니다.

- /g:network_name

데이터베이스 파티션 서버에 대한 네트워크 이름을 지정합니다. 이 매개변수를 지정하지 않은 경우, DB2는 사용자 시스템에서 검출한 최초 IP 주소를 사용합니다.

머신에 다중 IP 주소가 있으며 데이터베이스 파티션 서버의 특정 IP 주소를 지정하려는 경우, 이 매개변수를 사용합니다. 네트워크 이름 또는 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(다중 논리 노드를 수행할 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 변경

Windows NT 또는 Windows 2000에서, **db2nchg** 명령을 사용하여 다음을 수행하십시오.

- 하나의 머신에서 또다른 머신으로 데이터베이스 파티션을 이동시킵니다.

- 머신의 TCP/IP 호스트 이름을 변경합니다.

다중 네트워크 어댑터를 사용하려고 계획하는 경우, 이 명령을 사용하여 『netname』 필드의 TCP/IP 주소를 *db2nodes.cfg* 파일에 지정해야 합니다.

- 다른 논리 포트 번호를 사용합니다.
- 데이터베이스 파티션 서버(노드)에 대해 다른 이름을 사용합니다.

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 이동시키려는 경우 이 매개변수를 지정해야 합니다. 이 매개변수를 지정하지 않은 경우, 논리 포트 번호는 변경되지 않습니다.

- /h:host_name

내부 통신을 위해 FCM에 의해 사용되는 TCP/IP 호스트 이름을 변경합니다. 이 매개변수를 지정하지 않은 경우, 호스트 이름은 변경되지 않습니다.

- /m:machine_name

데이터베이스 파티션 서버를 또다른 머신으로 이동시킵니다. 데이터베이스 파티션 서버는 인스턴스에 기존 데이터베이스가 없는 경우에만 이동될 수 있습니다.

- /g:network_name

데이터베이스 파티션 서버에 대한 네트워크 이름을 변경합니다.

머신에 다중 IP 주소가 있으며 데이터베이스 파티션 서버의 특정 IP 주소를 사용하려는 경우, 이 매개변수를 사용하십시오. 네트워크 이름 또는 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 제거

Windows NT 및 Windows 2000에서, **db2ndrop** 명령을 사용하여 데이터베이스가 없는 인스턴스에서 데이터베이스 파티션 서버(노드)를 제거합니다. 데이터베이스 파티션 서버를 제거하면, 노드 번호는 새 데이터베이스 파티션 서버에서 재사용될 수 있습니다.

인스턴스에서 데이터베이스 파티션 서버를 제거할 때에는 주의하십시오. 인스턴스에서 인스턴스를 소유하는 데이터베이스 파티션 서버 노드 0을 제거하면, 인스턴스는 사용할 수 없게 됩니다. 인스턴스를 제거하려면, **db2idrop** 명령을 사용하십시오.

주: 이 인스턴스에 데이터베이스가 있으면, **db2ndrop** 명령을 사용하지 마십시오. 대신, **db2stop drop nodenum** 명령을 사용하십시오. 이렇게 하면, 데이터베이스는 데이터베이스 파티션에서 올바르게 제거됩니다. 파일을 변경하면 파티션된 데이터베이스 시스템에 불일치가 생길 수 있으므로 db2nodes.cfg 파일을 편집하지 마십시오.

다중 논리 노드를 수행 중인 머신에서 논리 포트 0을 지정한 노드를 제거하려는 경우, 논리 포트 0에 지정된 노드를 제거하기 전에 다른 논리 포트에 지정된 다른 모든 노드를 제거해야 합니다. 각 데이터베이스 파티션 서버는 논리 포트 0에 지정된 노드를 가져야 합니다.

명령에는 다음과 같은 매개변수가 있습니다.

```
db2ndrop /n:node_number /i:instance_name
```

- /n:

데이터베이스 파티션 서버를 식별하기 위한 고유한 노드 번호. 이것은 필수 매개변수입니다. 번호는 오름차순으로 0 - 999일 수 있습니다. 노드 0은 인스턴스를 소유하는 머신을 나타낸다는 것을 상기하십시오.

- /i:instance_name

인스턴스 이름. 이 매개변수는 선택적입니다. 인스턴스 이름을 제공하지 않은 경우, 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 수행 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 수행 중이라고 합니다. 다른 인스턴스에 참여하는 경우, 이 머신은 다중 논리 노드를 호스트하고 있지 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 수행하는 구성

다중 논리 노드를 사용하는 구성은 대칭 멀티프로세서(SMP) 아키텍처를 갖는 머신에서 시스템이 조회를 수행할 때 유용합니다. 머신에서 다중 논리 노드를 구성하는 기능은 머신이 실패할 경우 유용합니다. 머신이 실패하는 경우(데이터베이스 파티션 또는 서버가 실패하도록 함), DB2START NODENUM 명령을 사용하여 또다른 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 방법 중 하나로 다중 논리 노드를 구성할 수 있습니다.

- db2nodes.cfg 파일에 논리 노드(데이터베이스 파티션)를 구성합니다. 그런 다음, DB2START 명령과 연관된 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 수행 중인 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` 파일에 충분한 포트를 정의했는지 확인해야 합니다.

부록. 고속 노드 간 통신

DB2 Universal Database Enterprise - Extended Edition 사용시, 전체 시스템 효율이 사용자 비즈니스에 중요한 통신 집중 환경에서 작업할 수 있습니다.

파티션된 환경에서 사용할 수 있는 네트워크에는 두 가지 유형이 있습니다. 한 유형은 공용 LAN에서 TCP/IP를 사용합니다. 다른 유형은 TCP/IP를 사용하거나 전용 상호연결에서 가상 인터페이스(VI) 아키텍처를 사용합니다.

공용 상호연결은 기존 TCP/IP로 작업합니다. TCP/IP는 거의 어디에서나 통신 프로토콜로서 사용 가능합니다. 이것은 근거리 통신망(LAN) 환경입니다. 이 환경에서의 이점은 추가 소유자 하드웨어 및 소프트웨어를 요구하지 않고 즉시 사용자 클러스터에 접속하도록 선택할 수 있다는 것입니다. 이 환경의 단점은 추가 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 상호연결은 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 둘다의 추가적인 각 하드웨어, 소프트웨어 및 프로토콜 비용은 물론 양쪽 네트워크에 개별적인 관리가 필요할 수 있다는 것입니다. 전용 상호 연결의 예는 100Mb/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 이러한 배열이 편리합니다. Windows NT 운영 환경 내에서, NT 도메인 제어기에 대한 통신 액세스를 위해 공용 LAN

을 보유할 필요할 있을 수 있습니다. 도메인 제어기에 대한 정보는 429 페이지의 『부록E. Windows NT용 DB2가 Windows NT 보안을 사용하여 작업하는 방법』을 참조하십시오.

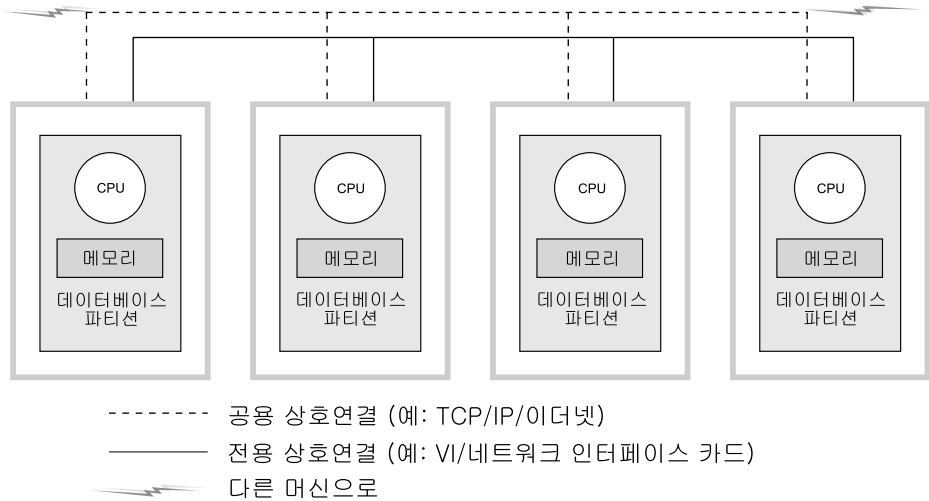


그림 12. 전용 SAN을 공용 LAN과 결합

나머지 절에는 다음 설명이 포함됩니다.

- 『TCP/IP를 사용하여 고속 상호연결』
- 457 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 URL, <http://www.ibm.com/pc/us/netfinity>를 참조하십시오.

추가 문서 및 소프트웨어 업그레이드에 대한 추가 문서는 IBM 웹 지원 사이트 <http://www.ibm.com/pc/support>를 참조하십시오.

1. 서버를 누르십시오.
2. 계열 아래에서, 클러스터링을 누르십시오.
3. 기술 정보 아래에서, 소프트웨어 업그레이드의 다운로드 가능한 파일을 누르거나 문서의 온라인 문서를 누르십시오.

IBM Netfinity SP Switch 주제를 찾아 필요한 파일을 다운로드하십시오.

IBM Netfinity SP Switch의 설정 프로시저

IBM Netfinity SP Switch를 설치하는 지시사항은 *IBM Netfinity SP Switch Installation and User's Guide*에 있습니다.

다양한 하드웨어 및 소프트웨어 구성요소(예: 서버 랙 격납장치, 호스트 어댑터 및 SP 스위치 소프트웨어)가 제공되는 하드웨어 및 소프트웨어 지시사항을 사용하여 해당 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 노드 간 통신 프로토콜입니다. VI는 SAN(System Area Network)에서 성능을 향상시키기 위해 Intel, Microsoft 및 Compaq이 합작으로 개발한 새로운 통신 아키텍처입니다. 아키텍처에 대한 자세한 정보는 <http://www.viarch.org>를 참조하십시오.

VIA가 사용 가능한 네트워크 인터페이스 카드(NIC), 스위치 및 소프트웨어 드라이버 구현을 수반하는 DB2 Universal Database에서 별도로 구입될 수 있는 제

품이 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 발생하는 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. Developer's Guide는 웹의 http://www.intel.com/design/servers/vi/developer/ia_imp_guide.htm에 들어 있습니다. Conformance Suite에 대한 정보도 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 대한 정보를 구하려면 <http://www.software.ibm.com/data>의 DB2 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 459 페이지의 『GigaNet Interconnect에 대한 프로시듀어 설정』을 참조하십시오.
- Compaq ServerNet Interconnect. 세부사항은 462 페이지의 『ServerNet Interconnect에 대한 프로시듀어 설정』을 참조하십시오.
- Fujitsu ServerNet Interconnect. 세부사항은 466 페이지의 『Synfinity Interconnect의 설정 프로시듀어』를 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 대한 정보를 구하거나 GigaNet 서비스 지원부에 문의하려면, 다음 URL, <http://www.giganet.com/>을 사용하십시오.

- Compaq ServerNet Interconnect

462 페이지의 『ServerNet Interconnect에 대한 프로시듀어 설정』에는 이 선택에 필요한 하드웨어, 소프트웨어 및 프로토콜 정보에 대한 개요가 들어 있습니다.

ServerNet 제품에 대한 정보를 알아보거나 ServerNet 서비스 지원부에 문의하려면, 다음 URL, <http://www.servernet.com/>을 사용하십시오.

- Fujitsu Synfinity Interconnect

466 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 문의하려면, 다음 URL <http://www.fujitsu.com/>을 사용하십시오.

VI를 사용하려면 DB2를 구성해야 합니다. 468 페이지의 『VI를 사용하여 DB2 사용』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 Network Interface Card
- GigaNet GNX5000 Switch
- GigaNet GNCxx11 Copper Interconnect Cables
- GigaNet cLAN Software, 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 문서도 사용해야 합니다.

각 GigaNet GNN1000는 GigaNet cLAN 소프트웨어 CD-ROM에 패키지화되어 있습니다. CD-ROM에는 GigaNet Interconnect를 설정하는 데 필요한 모든 소프트웨어가 들어 있습니다. 그밖에도 CD-ROM에는 VI 아키텍처 SDK 및 Adobe Acrobat Reader도 포함되어 있습니다. VI 아키텍처 SDK는 VI 사용 가능 응용 프로그램을 개발하고 테스트하는 데 사용됩니다. Adobe Acrobat Reader는 VI 사용 가능 응용 프로그램을 개발하는 방법을 설명하는 CD-ROM에서 문서를 보는 데 사용됩니다.

단계 요약은 다음과 같습니다.

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 카드를 설치하십시오. 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오.
2. GigaNet GNX5000 스위치 및 케이블을 설치하십시오. 설치 지시사항은 *GigaNet GNX5000 User Guide*를 참조하십시오.
3. GNX5000 스위치에 연결된 각 노드에 GigaNet GNN1000 어댑터 드라이버 소프트웨어를 설치하십시오. 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오. GigaNet이 제공하는 드라이버를 설치할 경우, 다음과 같은 추가 세부사항이 있습니다.
 - a. 이미 설치되어 있는 GNN1000 드라이버의 이전 버전을 제거하십시오. 이를 제거한 다음에는 노드를 재부트하십시오.

- b. 시작 -> 설정 -> 제어판 -> 네트워크 -> 어댑터 -> 추가를 사용하여 드라이버를 설치하십시오.
- c. 디스크 있음...을 누른 후 CD-ROM의 드라이버 디렉토리를 지정하십시오. 예를 들어, CD-ROM 드라이브가 F:이면, F:\Driver를 사용해야 합니다.
- d. 『GNN1000 NDIS 어댑터』를 선택한 다음, 확인을 누르십시오.
- 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 재부트하십시오.

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 Console을 설치하십시오. cLAN Management Console에 대한 추가 정보 및 설치 지시사항은 *GigaNet GNN1000 User Guide*를 참조하십시오. 다음은 설치 프로시듀어에 대한 추가 세부사항입니다.

- a. cLAN 소프트웨어 CD를 CD-ROM 드라이브에 삽입하십시오.
- b. CD 자동 설치 메뉴가 나타날 때까지 기다리십시오.
- c. 『cLAN Management Console 설치』를 누르십시오.
- 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 재부트하십시오.

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 『활동 중』인지 확인하십시오.
- e. 메뉴 표시줄에서 **Utilities** → **VI Throughput**을 선택하십시오 이것은 데이터가 하드웨어를 실제로 통과하는지 검사하기 위한 처리량 테스트를 수행합니다.
- f. 테스트시 사용하려고 하는 두 노드의 컴퓨터 이름을 대문자로 입력하십시오. 지역 노드를 소스 노드로 식별하십시오.
- g. **Start Measuring**을 누르십시오 데이터가 최소한 매초 65MB 비율로 전송되어야 합니다.
- h. 연결 테스트를 중지시키려면 **Stop Measuring**을 누르십시오.
- 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 절을 참조하십시오.

Windows NT용 DB2 Universal Database를 설치하고 구현하는 방법에 대한 정보는 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작을 참조하십시오.

ServerNet Interconnect에 대한 프로시듀어 설정

이 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 각 단계에서 필요한 내용의 요약이므로, 각 단계에 연관된 모든 세부사항이 여기에 나와 있지는 않습니다. 또한, 필요한 자세한 지시사항을 제공하는 각 단계별 참조 문서도 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 제거하십시오. (ServerNet을 처음 설치하는 경우이면 이 단계를 생략하십시오.)
 - a. 네트워크 제어판을 여십시오(시작 -> 설정 -> 제어판 -> 네트워크).
 - b. 어댑터 탭을 누르십시오.
 - c. Tandem ServerNet PCI 어댑터 드라이버를 제거하십시오.
 - d. 서비스 탭을 누르십시오.
 - e. SANMan을 제거하십시오.
 - f. 프로토콜 탭을 누르십시오.
 - g. Tandem ServerNet-I VI 프로토콜을 제거하십시오.
4. Tandem ServerNet PCI 어댑터 드라이버를 설치하십시오. 다음은 ServerNet이 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.
 - a. 네트워크 제어판을 여십시오(시작 -> 설정 -> 제어판 -> 네트워크).
 - b. 어댑터 탭을 누르십시오(어댑터 화면이 나타납니다).

- 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 추출 프로그램을 시작하십시오. (『ernnn.exe』는 설치될 ServerNet 드라이버의 특정 버전을 식별하는 번호 -- ERnnn.EXE -- 의 뒤에 표시되는 엔지니어링 릴리스의 이름입니다.)
 - d. 추출된 파일이 위치한 드라이브 및/또는 디렉토리로 변경하십시오. 『Spad n.n.n \ Free』 서브디렉토리로 변경하십시오. 여기서, 『n.n.n』은 제품의 특정 버전입니다. 문제점 해결 또는 개발 환경에서 작업 중인 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 나타납니다.)
 - g. 디스크 있음...을 누르십시오. (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 누르십시오 어댑터 목록이 ServerNet 어댑터를 표시하는지 확인하십시오. 닫기를 누르십시오.
 - j. 예를 선택하여 컴퓨터를 재시작하십시오 또는 아니오를 선택하고 SANMan 및 VI SDK(Software Developer's Kit)를 계속 설치하십시오.
5. SANMan을 설치하십시오. 다음은 ServerNET이 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.
- a. 네트워크 제어판을 여십시오(시작 -> 설정 -> 제어판 -> 네트워크).
 - b. 서비스 탭을 누르십시오(서비스 화면이 나타납니다).
 - 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 추출 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. 추출된 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 ServerNEt이 제공하는 소프트웨어 CD를 사용하여 설치할 경우, 필요한 추가 세부사항입니다.
- a. 네트워크 제어판을 여십시오(시작 -> 설정 -> 제어판 -> 네트워크).
 - b. 프로토콜 탭을 누르십시오. (네트워크 프로토콜 화면이 나타납니다.)
 - 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 추출 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 나타납니다.)
 - e. 디스크 있음...을 누르십시오. (디스크 삽입 화면이 나타납니다.)
 - f. 추출된 파일이 위치한 드라이브 및/또는 디렉토리로 입력하십시오.

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.

Windows NT용 DB2 Universal Database를 설치하고 구현하는 방법에 대한 정보는 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작을 참조하십시오.

Synfinity Interconnect의 설정 프로시듀어

이 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 세부사항이 여기에 나와 있지는 않습니다. 또한, 필요한 자세한 지시사항을 제공하는 각 단계별 참조 문서도 사용해야 합니다.

각 Synfinity System은 Synfinity Cluster Manager Software, 버전 1.10 CD-ROM과 함께 패키징되어 있습니다. CD-ROM에는 Synfinity Interconnect를 설정하는 데 필요한 모든 문서 및 소프트웨어가 들어 있습니다. 그밖에도 CD-ROM에는 *Synfinity Cluster Manager Software User Guide*가 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 제어판에서 볼 수 없을 수 있습니다.

단계 요약은 다음과 같습니다.

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.

설치하고 나면, Synfinity 클러스터 관리 프로그램 소프트웨어를 수행해야 합니다. 클러스터 관리 프로그램은 클러스터 플랜을 제공하며 단계별 지시사항을 통해 네트워크를 구성하도록 도우며, 최적의 라우팅 및 케이블링 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 Manager Software User Guide*를 참조하십시오.

3. Synfinity 스위치 및 케이블을 설치하십시오. *Synfinity Cluster User Guide*를 참조하십시오.
4. Synfinity 하드웨어가 작동하는지 테스트하십시오. 이것은 다음과 같은 방법으로 가능합니다.
 - a. 클러스터에 있는 임의의 시스템에서, Windows 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 c의 노드 번호입니다.
- g. "CONNECTION GOOD" 메시지가 표시되어야 합니다.
- h. "NO CONNECTION" 메시지가 표시되면 케이블링과 하드웨어 설정을 검사하고, *Synfinity Cluster Manager Software User Guide*를 참조하십시오. 또한 <http://www.fujitsu.com>에서 "기술 추가 정보"를 위한 지원 웹 페이지도 검사하십시오.

Windows NT용 DB2 Universal Database를 설치하고 구현하는 방법에 대한 정보는 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작을 참조하십시오.

VI를 사용하여 DB2 사용

자세한 설치 정보는 *Windows용 DB2 Enterprise - Extended Edition* 빠른 시작에 수록되어 있습니다.

Windows용 DB2 Enterprise - Extended Edition 빠른 시작에서 설명한 대로 DB2의 설치를 완료한 후, 다음 DB2 레지스트리 변수를 설정하고 인스턴스에 있는 각 데이터베이스 파티션에서 다음 작업을 수행하십시오.

1. DB2_VI_ENABLE=ON을 설정하십시오.

db2set 명령을 사용하여 레지스트리 변수 값을 수정하십시오. **db2_all** 명령을 사용하여 인스턴스의 모든 데이터베이스 파티션 서버의 **db2set** 명령을 수행하십시오. **db2_all** 명령을 수행하려면 관리자 그룹의 구성원이 되는 사용자 계정으로 로그인하십시오.

다음 예에서 ; 문자는 큰따옴표 안에 위치하여, 인스턴스의 모든 데이터베이스 파티션 서버에서 요청을 동시에 수행하도록 합니다.

```
db2_a11 ";db2set DB2_VI_ENABLE=ON"
```

db2_all 명령에 대한 자세한 정보는 *관리 안내서: 구현의 "다중 데이터베이스 파티션 서버로 명령 발행"*을 참조하십시오.

2. DB2_VI_DEVICE=nic0을 설정하십시오.

예를 들어, 다음과 같습니다.

```
db2_all ";db2set DB2_VI_DEVICE=nic0"
```

주: Synfinity Interconnect를 사용하여, 이 변수가 DB2_VI_DEVICE=VINIC로 설정되어야 합니다. 장치 이름(VINIC)은 대문자여야 합니다.

3. DB2_VI_VIPL=vipl.dll을 설정하십시오.

예를 들어, 다음과 같습니다.

```
db2_all ";db2set DB2_VI_VIPL=vipl.dll"
```

주: 예에서 사용한 값은 레지스트리 변수의 기본값입니다. 레지스트리 변수에 대한 자세한 정보는 *관리 안내서: 성능을 참조하십시오.*

4. 파티션 인스턴스에 db2start를 입력하십시오.
5. db2diag.log 파일을 검토하십시오. 각 파티션에 대해 『VI가 사용 가능』을 나타내는 메시지가 나타나야 합니다.
6. FCM(Fast Communication 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 상호연결 환경에서 이동 중인 경우, 더 낮은 값이 필요할 수 있습니다. 또한, Windows NT에서 DB2NTMEMSIZE 레지스트리 변수를 설정하여 DB2 기본값을 겹쳐쓸 필요가 있을 수 있습니다. 레지스트리 변수에 대한 자세한 정보는 *관리 안내서: 성능을 참조하십시오.*

부록J. LDAP(Lightweight Directory Access Protocol) 디렉토리 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 디렉토리로 데이터베이스 정보를 제공합니다. 클라이언트가 데이터베이스에 연결할 때, 카탈로그에 대한 정보는 LDAP 디렉토리에서 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 디렉토리를 검색합니다.

클라이언트가 관련 지역 디렉토리 카탈로그에서 LDAP 디렉토리만 한 번 검색하도록, 캐싱하는 메커니즘이 존재합니다. 정보가 검색되고 나면 지역 머신에 저장되거나 캐워됩니다. 동일한 정보에 대한 후속 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 디렉토리에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: DB2LDAPCACHE 레지스트리 변수는 데이터베이스 및 노드 디렉토리에만 적용할 수 있습니다.

LDAP 클라이언트 및 서버 구성 지원

다음 표는 지원되는 LDAP 클라이언트 및 서버 구성을 요약합니다.

표 26. 지원되는 LDAP 클라이언트 및 서버 구성

	IBM SecureWay 디렉토리 V3.1 및 V3.1.1	Microsoft Active Directory
IBM LDAP 클라이언트	지원	지원되지 않음
Microsoft LDAP/ADSI 클라이언트	지원	지원

IBM SecureWay 디렉토리 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 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.

- Outlook 98, Outlook 2000 또는 Outlook Express

주: Outlook Express는 Microsoft Internet Explorer의 일부로서 설치됩니다.

- Exchange Server
- Windows NT Server Service Pack 4
- Windows 98 Second Edition
- Windows 2000

Microsoft LDAP Client 지원은 ADSI(Active Directory Service Interface) 구성요소에 포함됩니다. ADSI의 최신 버전은 다음 사이트에서 다운로드될 수 있습니다.

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

Windows 98, Windows NT 또는 Windows 2000 운영 체제에서 수행할 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 활성 디렉토리 지원

DB2는 다음과 같이 활성 디렉토리를 검색합니다.

1. DB2 데이터베이스 서버는 `ibm_db2Node` 오브젝트로서 활성 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.

활성 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 98 또는 Windows NT의 경우, `wldap32.dll`이 시스템 디렉토리 아래에 있음을 검증할 필요가 있습니다.

- LDAP 지원을 사용 가능하게 하십시오. Windows 2000의 경우, LDAP 지원은 설치 프로그램으로 사용 가능합니다. Windows 98/NT의 경우, **db2set** 명령을 사용하여 DB2_ENABLE_LDAP 레지스트리 변수를 『YES』로 설정하여 명시적으로 LDAP를 사용 가능하게 하십시오.
- 활성 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 98/NT의 경우, **db2set** 명령을 사용하여 DB2_ENABLE_LDAP 레지스트리 변수를 『YES』로 설정하여 명시적으로 LDAP를 사용 가능하게 하십시오.
- LDAP 서버의 TCP/IP 호스트 이름 및 포트 번호. 이 값은 DB2LDAPHOST 응답 키워드를 사용하여 자동 설치 중 입력되거나, 사용자가 DB2SET 명령을 사용하여 나중에 수동으로 설정할 수 있습니다.

```
db2set DB2LDAPHOST=<hostname[:port]>
```

여기서 hostname은 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 식별 이름(baseDN)에 위치합니다. IBM SecureWay LDAP 디렉토리 서버 버전 3.1을 사용 중인 경우, DB2는 서버에서 동적으로 이 정보를 얻을 수 있으므로 기본 구분 이름을 구성할 필요는 없습니다. 그러나 IBM eNetwork 디렉토리 서버 버전 2.1을 사용하는 경우, DB2SET 명령을 사용하여 각 머신의 LDAP 기본 식별 이름을 구성해야 합니다.

```
db2set DB2LDAP_BASEDN=<baseDN>
```

여기서 baseDN은 LDAP 서버에 정의된 LDAP 접미부의 이름입니다. 이 LDAP 접미부는 DB2 오브젝트를 포함하는 데 사용됩니다.

- LDAP 사용자의 식별 이름(DN) 및 암호. 이것은 사용자가 LDAP를 사용하여 DB2 사용자 고유 정보를 저장하려고 계획하는 경우에만 필요합니다.

LDAP 사용자 작성

DB2는 사용자 레벨에서 DB2 레지스트리 변수 및 CLI 구성을 설정하도록 지원합니다. 이것은 AIX 및 Solaris 플랫폼에서 사용 가능하지 않습니다. 사용자 레벨 지원은 다중 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 디렉토리를 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 디렉토리로 가져오십시오. IBM LDAP 서버의 LDIF 유틸리티는 『LDIF2DB』입니다.
- IBM SecureWay LDAP 디렉토리 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 작성할 수 있도록 LDAP 사용자 오브젝트에 필수 액세스(ACL)를 부여해야 합니다. 사용자 오브젝트의 ACL을 권한 부여하려면, LDAP 디렉토리 서버 웹 관리 도구를 사용하십시오.

DB2 응용프로그램용 LDAP 사용자 구성

IBM LDAP 클라이언트로 작업할 때와 DB2를 수행하기 전에, 현재 로그인 사용자에게 대한 LDAP 사용자 식별 이름(DN) 및 암호를 구성해야 합니다. 이것은 db2ldcfg 유틸리티를 사용하여 완료될 수 있습니다.

```
db2ldcfg -u <userDN> -w <password> -> set the user's DN and password
          -r                               -> clear the user's DN and password
```

예를 들어, 다음과 같습니다.

```
db2ldcfg -u "cn=Mary Burnnet,ou=DB2 UDB Development,ou=Toronto,o=ibm,c=ca"
          -w password
```

설치 후 DB2 서버의 등록

클라이언트 응용프로그램이 DB2 서버 인스턴스로 연결하기 위해 사용하는 프로토콜 구성 정보를 공표하려면, 각 DB2 서버 인스턴스를 LDAP에 등록해야 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 또다른 별명 이름을 카탈로그화할 수 있습니다.

주: Windows 2000 도메인 환경에서 작업하는 경우, 설치 중 DB2 서버 인스턴스는 다음 정보로 활성 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 명령을 사용하여 모든 머신에 REGISTER 명령을 발행하십시오.

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 들어, 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 수행 중이고 TCP/IP를 통신 프로토콜로서 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 대한 추가 정보는 *Command Reference*를 참조하십시오.

DB2 서버에 대한 프로토콜 정보 갱신

LDAP에 있는 DB2 서버 정보는 현재 정보로 보존되어야 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.

- TCP/IP 호스트 이름 및 서비스 이름 또는 포트 번호 매개변수
- IPX 주소
- APPC 프로토콜 정보(예: TP 이름, 상대 LU 또는 모드)
- NetBIOS 워크스테이션 이름

원격 DB2 서버 프로토콜 구성 매개변수를 갱신하려면, node절이 있는 UPDATE LDAP 명령을 사용하십시오.

```
db2 update ldap
  node <node_name>
  hostname <host_name>
  svcname <tcpip_service_name>
```

UPDATE LDAP 명령에 대한 자세한 정보는 *SQL 참조서*를 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 노드 이름이 필요할 경우, CATALOG LDAP NODE 명령을 사용하여 변경하십시오. 명령은 다음과 같습니다.

```
db2 catalog ldap node <ldap_node_name>
as <new_alias_name>
```

LDAP 노드를 카탈로그 해제하려면, UNCATALOG LDAP NODE 명령을 사용하십시오. 명령은 다음과 같습니다.

```
db2 uncatalog ldap node <ldap_node_name>
```

DB2 서버 등록 해제

LDAP에서 인스턴스의 등록을 해제하면 이 인스턴스에 대한 모든 노드 또는 별명, 오브젝트 및 데이터베이스 오브젝트도 제거합니다.

지역 또는 원격 머신에 있는 DB2 서버의 등록 해제는 이 서버에 대해 지정된 LDAP 노드 이름을 필요로 합니다.

```
db2 deregister db2 server in ldap
node <node_name>
```

DB2 서버가 등록 해제될 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 클라이언트 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 디렉토리가 검색됩니다.

이 이름이 LDAP 디렉토리에 이미 있을 경우, 데이터베이스가 지역 머신에서 여전히 작성되지만 LDAP 디렉토리에 이름 지정 충돌이 있음을 나타내는 경고 메시지가 리턴됩니다. 이러한 이유로 LDAP 디렉토리에서 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 접속할 때, 노드가 지역 노드 디렉토리에 있지 않으므로 DB2는 목표 노드 항목에 대해 LDAP 디렉토리를 검색합니다. LDAP 디렉토리에서 항목이 발견되지 않으면, 원격 서버의 프로토콜 정보가 검색됩니다. 데이터베이스에 연결하며 항목이 LDAP 디렉토리에 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 디렉토리를 검색하지 않고 지역 데이터베이스 디렉토리에 있는 정보를 사용합니다.

자세히 설명하면, 클라이언트가 관련 지역 디렉토리 카탈로그에서 LDAP 디렉토리만 한 번 검색하도록, 캐싱 메커니즘이 있습니다. 정보가 검색되고 나면 지역 머신

에 저장되거나 캐쉬됩니다. 동일한 정보에 대한 후속 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 디렉토리에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: LDAP 정보의 캐싱은 사용자 레벨 CLI 또는 DB2 프로파일 레지스트리 변수에 적용할 수 없습니다. 또한, 데이터베이스, 노드 및 DCS 디렉토리에 대한 『메모리 내』 캐쉬도 있습니다. 그러나 단지 노드 디렉토리에 대해 캐쉬되는 것은 없습니다.

데이터베이스 등록 해제

데이터베이스는 다음 경우 LDAP로부터 자동으로 등록 해제됩니다.

- 데이터베이스가 제거된 경우
- 소유하는 인스턴스가 LDAP로부터 등록 해제된 경우

다음 명령을 사용하여 수동으로 LDAP로부터 데이터베이스를 등록 해제할 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 디렉토리만 한 번 검색하도록, 캐싱하는 메커니즘이 있습니다. 정보가 검색되고 나면 지역

머신에 저장되거나 캐쉬됩니다. 동일한 정보에 대한 후속 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 디렉토리에서 정보를 읽으며 지역 캐쉬가 새로 고쳐집니다.

주: LDAP 정보의 캐싱은 사용자 레벨 CLI 또는 DB2 프로파일 레지스트리 변수에 적용할 수 없습니다. 또한, 데이터베이스, 노드 및 DCS 디렉토리에 대한 『메모리 내』 캐쉬도 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 디렉토리 파티션 및 현재 활성 디렉토리 도메인을 Windows 2000 환경에서 검색합니다. 다중 LDAP 디렉토리 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 연결 또는 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 등록

호스트 데이터베이스를 LDAP에 등록할 때, 두 개의 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 카탈로그화합니다. 두 번째 경우에, 사용자는 LDAP에서 게이트웨이 서버를 등록한 후, 게이트웨이 서버의 노드 이름을 지정하는 LDAP에 호스트 데이터베이스를 카탈로그화합니다.

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 완료해야 합니다.

1. APPN 연결성을 위해서는 호스트 데이터베이스 서버를 LDAP에 등록하십시오. REMOTE 및 INSTANCE절은 임의입니다. NODETYPE절은 『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, NODETYPE절은 『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. APPN 연결성을 사용하여 LDAP에서 호스트 데이터베이스를 카탈로그화하십시오. 호스트 데이터베이스 이름은 『NIAGARA_FALLS』이며, 데이터베이스 별명 이름은 『nftcpip』입니다. GWNODE절은 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 카탈로그화하십시오.

```
db2 catalog ldap database NIAGARA_FALLS as nfappn at node nfappn
gwnode whasf authentication dcs
```

위에 표시한 등록 및 카탈로그를 완료한 후, TCP/IP를 사용하여 호스트에 연결하려는 경우 『nftcpip』에 연결합니다. APPN을 사용하여 호스트에 연결하려는 경우, 『nfappn』에 연결합니다. 클라이언트 워크스테이션에 DB2 Connect를 갖지 않은 경우, TCP/IP를 사용하여 게이트웨이를 통해 연결이 진행되며, 거기에서 『nftcpip』 또는 『nfappn』을 사용하는지 여부에 따라 TCP/IP 또는 APPN을 각각 사용하여 호스트에 연결합니다.

각 클라이언트가 수동으로 데이터베이스와 노드를 각 머신에 지역적으로 카탈로그화할 필요가 없도록 LDAP에 호스트 데이터베이스 정보를 수동으로 구성할 수 있습니다. 프로세스는 다음과 같습니다.

1. 호스트 데이터베이스 서버를 LDAP에 등록하십시오. REMOTE, INSTANCE 및 NODETYPE절을 각각 사용하여 REGISTER 명령에서 호스트 데이터베이스 서버에 대해 원격 컴퓨터 이름, 인스턴스 이름 및 노드 유형을 지정해야 합니다. REMOTE절은 호스트 이름 또는 호스트 서버 머신의 LU 이름 중 하나로 설정될 수 있습니다. INSTANCE절은 8자 이하인 문자열로 설정될 수 있습니다. 예를 들어, 인스턴스 이름은 『DB2』로 설정될 수 있습니다. NODETYPE절은 『DCS』로 설정되어야 이것이 호스트 데이터베이스 서버임을 나타낼 수 있습니다.
2. CATALOG LDAP DATABASE 명령을 사용하여 호스트 데이터베이스를 LDAP에 등록하십시오. PARMS절을 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, `-u1` 옵션을 사용하십시오.

```
db2set -u1 <variable>=<value>
```

주: 이것은 AIX 또는 Solaris에서 지원되지 않습니다.

DB2에는 캐싱하는 메커니즘이 있습니다. 사용자 레벨의 DB2 프로파일 레지스트리 변수는 지역 머신에서 캐시됩니다. `-u1` 매개변수가 지정된 경우, DB2는 항상 캐쉬에서 DB2 레지스트리 변수를 읽어옵니다. 캐시는 다음 경우에 새로 고쳐집니다.

- 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 사용

설치 프로세스 완료에 이어 일부 지점에서 LDAP 지원을 사용 가능하게 하려면, 각 머신에서 다음 프로시저를 사용하십시오.

- LDAP 지원 2진 파일을 설치하십시오. 설치 프로그램을 수행하여 사용자 설치로부터 LDAP 디렉토리 개발 지원을 선택하십시오. 설치 프로그램이 2진 파일을 설치하고 DB2 프로파일 레지스트리 변수 `DB2_ENABLE_LDAP`를 『YES』로 설정합니다.

주: Windows 98/NT 및 UNIX의 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 (선택적) 포트 번호를 선언하십시오.

```
db2set DB2LDAPHOST=<base_domain_name>[:port_number]
```


여기서 `base_domain_name`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 식별 이름(baseDN)에 위치합니다. IBM SecureWay LDAP 디렉토리 서버 버전 3.1을 사용 중인 경우, DB2는 서버에서 동적으로 이 정보를 얻을 수 있으므로 기본 식별 이름을 구성할 필요는 없습니다. 그러나 IBM eNetwork 디렉토리 서버 버전 2.1을 사용하는 경우, DB2SET 명령을 사용하여 각 머신의 LDAP 기본 식별 이름을 구성해야 합니다.

```
db2set DB2LDAP_BASEDN=<baseDN>
```

여기서 `baseDN`은 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 들어, 다음과 같습니다.

```
db2 register ldap as <node-name> protocol tcpip
```

- LDAP에 등록하고자 하는 데이터베이스가 있으면, CATALOG LDAP DATABASE 명령을 수행하십시오. 예를 들어, 다음과 같습니다.

```
db2 catalog ldap database <dbname> as <alias_dbname>
```

- LDAP 사용자의 식별 이름(DN) 및 암호를 입력하십시오. 이것은 사용자가 LDAP를 사용하여 DB2 사용자 고유 정보를 저장하려고 계획하는 경우에만 필요합니다.

LDAP 지원 사용 불가능

LDAP 지원을 사용 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_LDAP`를 『NO』로 설정하십시오.

LDAP 지원 및 DB2 Connect

LDAP 지원이 DB2 Connect 게이트웨이에서 사용 가능하며 데이터베이스가 게이트웨이 데이터베이스 디렉토리에 없는 경우, DB2는 LDAP를 찾아보며 찾은 정보를 보존하려고 시도합니다.

보안 고려사항

LDAP 디렉토리에 있는 정보를 액세스하기 전에, 응용프로그램 또는 사용자가 LDAP 서버에 의해 인증됩니다. 이 인증 프로세스는 LDAP 서버에 대한 바인딩이라고 합니다.

익명의 사용자가 정보를 추가, 삭제 또는 수정하는 것을 막기 위해 LDAP 디렉토리에 저장된 정보에 대해 액세스 제어를 적용하는 것은 중요한 일입니다.

액세스 제어는 기본적으로 상속되며 컨테이너 레벨에서 적용될 수 있습니다. 새로운 오브젝트가 작성될 때, 이것은 상위 오브젝트와 동일한 보안 속성을 상속합니다. 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 노드 항목의 경우, 모든 사용자(또는 모든 익명의 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 또다른 사용자의 프로파일에 액세스할 수 없습니다.

주: 권한 부여 검사는 LDAP 서버에 의해 항상 수행되며, DB2에 의해서는 수행되지 않습니다. LDAP 권한 부여 검사는 DB2 권한 부여와는 관계가 없습니다. SYSADM 권한을 가지는 계정 또는 권한 ID는 LDAP 디렉토리에 대해 액세스 권한이 없을 수도 있습니다.

LDAP 명령 또는 API를 수행할 때, 바인드 식별 이름(bindDN) 및 암호가 지정되지 않은 경우, DB2는 요청된 명령을 수행하는 충분한 권한이 없을 수 있는 기본 자격사항을 사용하여 LDAP 서버에 바인드하여 오류를 리턴합니다.

DB2 명령 또는 API에 대해 USER 및 PASSWORD절을 사용하여 사용자의 bindDN 및 암호를 명시적으로 지정할 수 있습니다. DB2 명령에 대한 자세한 정보는 *Command Reference*를 참조하고, DB2 API에 대한 자세한 정보는 *Administrative API Reference*를 참조하십시오.

Windows 2000에서 활성 디렉토리 보안 고려사항

DB2 데이터베이스 및 노드 오브젝트는 활성 디렉토리에 DB2 서버가 설치된 머신의 컴퓨터 오브젝트 아래에 작성됩니다. 활성 디렉토리에서 데이터베이스 서버를 등록하거나 데이터베이스를 카탈로그화하려면, 컴퓨터 오브젝트에서 오브젝트를 작성 및/또는 갱신하기에 충분한 액세스를 가질 필요가 있습니다.

기본적으로, 컴퓨터 오브젝트 아래에 있는 오브젝트는 인증된 사용자가 읽을 수 있으며 관리자(관리자, 도메인 관리자 및 엔터프라이즈 관리자 그룹에 속하는 사용자)가 갱신할 수 있습니다. 특정 사용자 또는 그룹에 대한 액세스를 부여하려면, 다음과 같이 **활성 디렉토리 사용자 및 컴퓨터 관리 콘솔(MMC)**을 사용하십시오.

1. **활성 디렉토리 사용자 및 컴퓨터 관리** 도구를 시작하십시오.
(시작—> 프로그램—> 관리 도구—> 활성 디렉토리 사용자 및 컴퓨터)
2. 보기 아래에서, **고급 기능을 선택**하십시오.
3. **컴퓨터 컨테이너**를 선택하십시오.
4. DB2가 설치된 서버 머신을 나타내는 컴퓨터 오브젝트를 마우스 오른쪽 단추로 누른 후, **등록 정보**를 선택하십시오.
5. **보안 탭**을 선택한 후, 지정된 사용자 또는 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.

기본적으로, 관리자만이 사용자 오브젝트 아래에서 오브젝트를 작성할 액세스를 갖습니다. 사용자 레벨에서 DB2 레지스트리 변수 또는 CLI 설정을 설정하게 하는 액세스를 부여하려면, 다음과 같이 **활성 디렉토리 사용자 및 컴퓨터 관리 콘솔 (MMC)**을 사용하십시오.

1. **활성 디렉토리 사용자 및 컴퓨터 관리** 도구를 시작하십시오.
(시작 → 프로그램 → 관리 도구 → 활성 디렉토리 사용자 및 컴퓨터)
2. 사용자 컨테이너 아래에서 사용자 오브젝트를 선택하십시오.
3. 사용자 오브젝트를 마우스 오른쪽 단추로 누른 후, 등록 정보를 선택하십시오.
4. 보안 탭을 선택하십시오.
5. 추가 버튼을 사용하여 목록에 사용자 이름을 추가하십시오.
6. 『쓰기』 및 『모든 하위 오브젝트 작성』 액세스를 부여하십시오.
7. 고급 설정을 사용하여, 『이 오브젝트 및 모든 하위 오브젝트』에 적용하는 사용권한을 설정하십시오.
8. 『이 오브젝트로 전파하도록 상위에서 상속가능한 사용권한 허용』 선택란을 선택하십시오.

DB2 오브젝트 클래스 및 속성이 있는 디렉토리 스키마 확장

LDAP 디렉토리 스키마는 LDAP 디렉토리 항목에 저장된 정보에 대해 오브젝트 클래스 및 속성을 정의합니다. 오브젝트 클래스는 필수 및 선택적 속성 세트로 이루어져 있습니다. LDAP 디렉토리의 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 디렉토리 버전 2.1에 대해 디렉토리 스키마 확장

IBM eNetwork 디렉토리 버전 2.1을 사용할 때, DB2가 사용하는 오브젝트 클래스와 속성이 있는 기본 스키마를 확장해야 합니다.

다음 단계를 사용하여 IBM eNetwork 디렉토리 버전 2.1에 대한 기본 스키마를 확장하십시오.

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 디렉토리 스키마에 정의되어 있는 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**를 수행하여 활성 디렉토리에 대한 스키마를 확장해야 합니다.

db2schex 프로그램은 CD-ROM 제품에 있습니다. CD-ROM에서 이 프로그램의 위치는 db2 디렉토리 및 common 서브디렉토리 아래에 있습니다. 예를 들어, 다음과 같습니다.

```
x:\db2\common
```

여기서 x:는 CD-ROM 드라이브입니다.

명령은 다음과 같이 사용됩니다.

```
db2schex
```

이 명령에 연관된 기타 선택적 절이 있습니다.

- -b UserDN
사용자 식별 이름 지정
- -w Password
바인드 암호 지정
- -u
스키마 설치 제거
- -k
오류를 무시하고 설치 제거를 계속 강제 수행

주:

1. UserDN 및 암호가 지정되지 않은 경우, **db2schex**는 현재 로그인된 사용자로서 바인드합니다.
2. userDN절은 Windows NT 사용자 이름으로서 지정될 수 있습니다.
3. 스키마를 갱신하려면, 스키마 관리자의 구성원이어야 하거나 스키마를 갱신하기 위한 권한이 위임되어 있어야 합니다.

예:

- DB2 스키마를 설치하려면, 다음을 입력하십시오.

```
db2schex
```

- DB2 스키마를 설치하고 바인드 DN 및 암호를 지정하려면, 다음을 입력하십시오.

```
db2schex -b "cn=A Name,dc=toronto1,dc=ibm,dc=com"  
-w password
```

또는

```
db2schex -b Administrator -w password
```

- DB2 스키마를 설치 제거하려면, 다음을 입력하십시오.

```
db2schex -u
```

- DB2 스키마를 설치 제거하고 오류를 무시하려면, 다음을 입력하십시오.

```
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 쓰기 가능한지 확인(즉, 레지스트리에서 안전 interlock이 제거됨)
5. 새로운 속성 작성
6. 새로운 오브젝트 클래스 작성
7. 오류를 검출하고, 오류가 발생하면 프로그램은 스키마에 대한 변경사항을 구간 복원

Windows 2000 Active Directory에 있는 DB2 오브젝트

DB2는 두 위치에서 활성 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. 이들 오브젝트에는 해당 사용자에게 고유한 정보가 들어 있습니다.

DB2가 사용하는 오브젝트 클래스 및 속성

다음 표에서는 DB2에서 사용되는 오브젝트 클래스를 설명합니다.

표 27. *cimManagedElement*

클래스	cimManagedElement
활성 디렉토리 LDAP 표시장치 이름	적용할 수 없음
활성 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
GUID(전역 고유 식별자)	b3afd63f-5c5b-11d3-b818-002035559151

표 28. *cimSetting*

클래스	cimSetting
활성 디렉토리 LDAP 표시장치 이름	적용할 수 없음
활성 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 29. eProperty

클래스	eProperty
활성 디렉토리 LDAP 표시장치 이름	ibm-eProperty
활성 디렉토리 공통 이름(cn)	ibm-eProperty
설명	사용자 환경 설정 등록 정보에 대한 응용프로그램 고유의 설정을 지정하는 데 사용됩니다.
SubClassOf	cimSetting
필수 속성(s)	
선택적 속성(s)	propertyType cisPropertyType cisProperty cesPropertyType cesProperty binPropertyType binProperty
유형	structural
OID(오브젝트 식별자)	1.3.18.0.2.6.90
GUID(전역 고유 식별자)	b3afd69c-5c5b-11d3-b818-002035559151

표 30. DB2Node

클래스	DB2Node
활성 디렉토리 LDAP 표시장치 이름	ibm-db2Node
활성 디렉토리 공통 이름(cn)	ibm-db2Node
설명	DB2 서버 설명
SubClassOf	eSap / ServiceConnectionPoint
필수 속성(s)	db2nodeName
선택적 속성(s)	db2nodeAlias db2instanceName db2Type host / dNSHostName(주 2 참조) protocolInformation/ServiceBindingInformation

표 30. DB2Node (계속)

클래스	DB2Node
유형	structural
OID(오브젝트 식별자)	1.3.18.0.2.6.116
GUID(전역 고유 식별자)	b3afd65a-5c5b-11d3-b818-002035559151
특수 주	<ol style="list-style-type: none"> 1. <i>DB2Node</i> 클래스는 IBM SecureWay 디렉토리 아래의 <i>eSap</i> 오브젝트 클래스와 Microsoft Active Directory 아래의 <i>ServiceConnectionPoint</i> 오브젝트에서 파생됩니다. 2. <i>host</i>는 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 상속된 <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 공백을 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(예: 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 않음)
- 모드
- 보안(『NONE』, 『SAME』 또는 『PROGRAM』)
- (선택적) LAN 어댑터 주소
- (선택적) 암호 변경 LU

주: Windows NT용 DB2 클라이언트에서(또는 Windows 98의 경우), APPN 정보가 지역 SNA 스택에서 구성되지 않은 경우 그리고 LAN 어댑터 주소 및 선택적 암호 변경 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 리스너는 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 컴퓨터 이름
- 서버의 인스턴스 이름

표 31. DB2Database

클래스	DB2Database
활성 디렉토리 LDAP 표시장치 이름	ibm-db2Database
활성 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
GUID(전역 고유 식별자)	b3afd659-5c5b-11d3-b818-002035559151

표 32. db2additionalParameters

속성	db2additionalParameters
활성 디렉토리 LDAP 표시장치 이름	ibm-db2AdditionalParameters
활성 디렉토리 공통 이름(cn)	ibm-db2AdditionalParameters
설명	호스트 데이터베이스 서버에 연결할 때 사용되는 모든 추가 매개변수 포함
구분	대소문자 비 구분 문자열
최대 길이	1024
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.426
GUID(전역 고유 식별자)	b3afd315-5c5b-11d3-b818-002035559151

표 33. db2authenticationLocation

속성	db2authenticationLocation
활성 디렉토리 LDAP 표시장치 이름	ibm-db2AuthenticationLocation
활성 디렉토리 공통 이름(cn)	ibm-db2AuthenticationLocation
설명	인증이 발생하는 곳 지정
구문	대소문자 비 구분 문자열
최대 길이	64
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.425
GUID(전역 고유 식별자)	b3afd317-5c5b-11d3-b818-002035559151
주	올바른 값: CLIENT, SERVER, DCS, DCE, KERBEROS, SVRENCRYPT 또는 DCSENCRYPT

표 34. db2ARLibrary

속성	db2ARLibrary
활성 디렉토리 LDAP 표시장치 이름	ibm-db2ARLibrary
활성 디렉토리 공통 이름(cn)	ibm-db2ARLibrary
설명	응용프로그램 리퀘스터 라이브러리의 이름
구문	대소문자 비 구분 문자열
최대 길이	256
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.427
GUID(전역 고유 식별자)	b3afd316-5c5b-11d3-b818-002035559151

표 35. db2databaseAlias

속성	db2databaseAlias
활성 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseAlias
활성 디렉토리 공통 이름(cn)	ibm-db2DatabaseAlias
설명	데이터베이스 별명 이름(s)
구문	대소문자 비 구분 문자열
최대 길이	1024
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.422
GUID(전역 고유 식별자)	b3afd318-5c5b-11d3-b818-002035559151

표 36. *db2databaseName*

속성	db2databaseName
활성 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseName
활성 디렉토리 공통 이름(cn)	ibm-db2DatabaseName
설명	데이터베이스 이름
구문	대소문자 비 구분 문자열
최대 길이	1024
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.421
GUID(전역 고유 식별자)	b3afd319-5c5b-11d3-b818-002035559151

표 37. *db2databaseRelease*

속성	db2databaseRelease
활성 디렉토리 LDAP 표시장치 이름	ibm-db2DatabaseRelease
활성 디렉토리 공통 이름(cn)	ibm-db2DatabaseRelease
설명	데이터베이스 릴리스 번호
구문	대소문자 비 구분 문자열
최대 길이	64
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.429
GUID(전역 고유 식별자)	b3afd31a-5c5b-11d3-b818-002035559151

표 38. *db2nodeAlias*

속성	db2nodeAlias
활성 디렉토리 LDAP 표시장치 이름	ibm-db2NodeAlias
활성 디렉토리 공통 이름(cn)	ibm-db2NodeAlias
설명	노드 별명 이름(s)
구문	대소문자 비 구분 문자열
최대 길이	1024
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.420
GUID(전역 고유 식별자)	b3afd31d-5c5b-11d3-b818-002035559151

표 39. db2nodeName

속성	db2nodeName
활성 디렉토리 LDAP 표시장치 이름	ibm-db2NodeName
활성 디렉토리 공통 이름(cn)	ibm-db2NodeName
설명	노드 이름
구문	대소문자 비 구분 문자열
최대 길이	64
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.419
GUID(전역 고유 식별자)	b3afd31e-5c5b-11d3-b818-002035559151

표 40. db2nodePtr

속성	db2nodePtr
활성 디렉토리 LDAP 표시장치 이름	ibm-db2NodePtr
활성 디렉토리 공통 이름(cn)	ibm-db2NodePtr
설명	데이터베이스를 소유하는 데이터베이스 서버를 나타내는 노드(DB2Node) 오브젝트에 대한 포인터
구문	식별 이름
최대 길이	1000
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.423
GUID(전역 고유 식별자)	b3afd31f-5c5b-11d3-b818-002035559151
특수 주	이 관계는 클라이언트가 데이터베이스에 연결하기 위해 프로토콜 통신 정보를 검색할 수 있도록 합니다.

표 41. db2gwPtr

속성	db2gwPtr
활성 디렉토리 LDAP 표시장치 이름	ibm-db2GwPtr
활성 디렉토리 공통 이름(cn)	ibm-db2GwPtr
설명	게이트웨이 서버를 나타내며 데이터베이스가 액세스될 수 있는 노드 오브젝트에 대한 포인터
구문	식별 이름
최대 길이	1000
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.424

표 41. db2gwPtr (계속)

속성	db2gwPtr
GUID(전역 고유 식별자)	b3afd31b-5c5b-11d3-b818-002035559151

표 42. db2instanceName

속성	db2instanceName
활성 디렉토리 LDAP 표시장치 이름	ibm-db2InstanceName
활성 디렉토리 공통 이름(cn)	ibm-db2InstanceName
설명	데이터베이스 서버 인스턴스의 이름
구문	대소문자 비 구분 문자열
최대 길이	256
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.428
GUID(전역 고유 식별자)	b3afd31c-5c5b-11d3-b818-002035559151

표 43. db2Type

속성	db2Type
활성 디렉토리 LDAP 표시장치 이름	ibm-db2Type
활성 디렉토리 공통 이름(cn)	ibm-db2Type
설명	데이터베이스 서버의 유형
구문	대소문자 비 구분 문자열
최대 길이	64
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.418
GUID(전역 고유 식별자)	b3afd320-5c5b-11d3-b818-002035559151
주	데이터베이스 서버의 유효한 유형: SERVER, MPP, DCS

표 44. DCEPrincipalName

속성	DCEPrincipalName
활성 디렉토리 LDAP 표시장치 이름	ibm-DCEPrincipalName
활성 디렉토리 공통 이름(cn)	ibm-DCEPrincipalName
설명	DCE 핵심부 이름
구문	대소문자 비 구분 문자열
최대 길이	2048

표 44. DCEPrincipalName (계속)

속성	DCEPrincipalName
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.443
GUID(전역 고유 식별자)	b3afd32d-5c5b-11d3-b818-002035559151

표 45. cesProperty

속성	cesProperty
활성 디렉토리 LDAP 표시장치 이름	ibm-cesProperty
활성 디렉토리 공통 이름(cn)	ibm-cesProperty
설명	이 속성의 값은 응용프로그램 고유의 환경 설정 구성 매개 변수를 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 비 구분 문자열
최대 길이	32700
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.307
GUID(전역 고유 식별자)	b3afd2d5-5c5b-11d3-b818-002035559151

표 46. cesPropertyType

속성	cesPropertyType
활성 디렉토리 LDAP 표시장치 이름	ibm-cesPropertyType
활성 디렉토리 공통 이름(cn)	ibm-cesPropertyType
설명	이 속성의 값은 구문, 의미 또는 cesProperty 속성의 모든 값의 다른 특성을 설명하기 위해 사용될 수 있습니다. 예를 들어, 『XML』의 값은 cesProperty 속성의 모든 값이 XML 구문으로서 코드화됨을 나타내기 위해 사용될 수 있습니다.
구문	대소문자 비 구분 문자열
최대 길이	128
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.308
GUID(전역 고유 식별자)	b3afd2d6-5c5b-11d3-b818-002035559151

표 47. *cisProperty*

속성	cisProperty
활성 디렉토리 LDAP 표시장치 이름	ibm-cisProperty
활성 디렉토리 공통 이름(cn)	ibm-cisProperty
설명	이 속성의 값은 응용프로그램 고유의 환경 설정 구성 매개 변수를 제공하기 위해 사용될 수 있습니다. 예를 들어, 값은 INI 파일을 포함할 수 있습니다. 이 모든 속성 값은 cisPropertyType 속성 값에서 동종이어야 합니다.
구문	대소문자 비 구분 문자열
최대 길이	32700
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.309
GUID(전역 고유 식별자)	b3afd2e0-5c5b-11d3-b818-002035559151

표 48. *cisPropertyType*

속성	cisPropertyType
활성 디렉토리 LDAP 표시장치 이름	ibm-cisPropertyType
활성 디렉토리 공통 이름(cn)	ibm-cisPropertyType
설명	이 속성의 값은 구문, 의미 또는 cisProperty 속성의 모든 값의 다른 특성을 설명하기 위해 사용될 수 있습니다. 예를 들어, 『INI 파일』의 값은 cisProperty 속성의 모든 값이 INI 파일임을 나타내기 위해 사용될 수 있습니다.
구문	대소문자 비 구분 문자열
최대 길이	128
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.310
GUID(전역 고유 식별자)	b3afd2e1-5c5b-11d3-b818-002035559151

표 49. *binProperty*

속성	binProperty
활성 디렉토리 LDAP 표시장치 이름	ibm-binProperty
활성 디렉토리 공통 이름(cn)	ibm-binProperty

표 49. binProperty (계속)

속성	binProperty
설명	이 속성의 값은 응용프로그램 고유의 환경 설정 구성 매개 변수를 제공하기 위해 사용될 수 있습니다. 예를 들어, 값에는 2진 암호화 Lotus 123 등록 정보 세트가 들어 있을 수 있습니다. 이 속성의 모든 값은 해당 binPropertyType 속성 값에서 동종이어야 합니다.
구분	2진
최대 길이	250000
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.305
GUID(전역 고유 식별자)	b3afd2ba-5c5b-11d3-b818-002035559151

표 50. binPropertyType

속성	binPropertyType
활성 디렉토리 LDAP 표시장치 이름	ibm-binPropertyType
활성 디렉토리 공통 이름(cn)	ibm-binPropertyType
설명	이 속성의 값은 구분, 의미 또는 binProperty 속성의 모든 값의 다른 특성을 설명하기 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 비 구분 문자열
최대 길이	128
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.306
GUID(전역 고유 식별자)	b3afd2bb-5c5b-11d3-b818-002035559151

표 51. PropertyType

속성	PropertyType
활성 디렉토리 LDAP 표시장치 이름	ibm-propertyType
활성 디렉토리 공통 이름(cn)	ibm-propertyType
설명	이 속성의 값은 eProperty 오브젝트의 의미 특성을 설명합니다.
구분	대소문자 비 구분 문자열
최대 길이	128

표 51. PropertyType (계속)

속성	PropertyType
다중 값	다중 값
OID(오브젝트 식별자)	1.3.18.0.2.4.320
GUID(전역 고유 식별자)	b3afd4ed-5c5b-11d3-b818-002035559151

표 52. settingID

속성	settingID
활성 디렉토리 LDAP 표시장치 이름	적용할 수 없음
활성 디렉토리 공통 이름(cn)	적용할 수 없음
설명	eProperty와 같은 오브젝트 항목에서 파생한 cimSetting를 식별하기 위해 사용될 수 있는 이름 지정 속성
구문	대소문자 비 구분 문자열
최대 길이	256
다중 값	단일 값
OID(오브젝트 식별자)	1.3.18.0.2.4.325
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 도구 모음에 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.

패키징 고려사항

Java 아카이브 파일의 규칙에 따라 확장 클래스 파일을 ZIP해야 합니다. 제어 센터 도구를 응용프로그램으로 수행하려면, ZIP 파일(db2plug.zip)이 *classpath*에 있어야 합니다. 제어 센터 도구를 애플릿으로 수행하려면, ZIP 파일이 제어 센터 HTML 파일에서 <codebase> 태그가 지정하는 곳에 있어야 합니다.

ZIP 파일은 압축 없이 빌드되고 모든 클래스 파일의 상대 경로 위치를 유지보수해야 합니다(zip -r0 db2plug.zip *.class).

인터페이스 설명

다음과 같은 인터페이스가 제공됩니다.

- CCExtension
- CCOject
- CCMenuAction

- CCToolbarAction

인터페이스는 다음 절에서 예와 함께 설명됩니다.

CCExtension

CCExtension 인터페이스는 사용자가 새 도구 모음 버튼, 새 메뉴 항목을 추가하고 이미 있는 메뉴 조치를 겹쳐써서 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 사용 하려면, 다음과 같은 설정 프로시더어를 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 해당 상대 경로 정보를 보존합니다.

2. 제어 센터를 애플릿으로 수행하려면, db2plug.zip 파일을 제어 센터 HTML 파일에서 <codebase> 태그가 지정하는 곳에 넣으십시오. 제어 센터를 응용프로그램으로 수행하려면, 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 이들이 속하는 패키지에 따라 해당 상대 디렉토리에 있어야 합니다.

CCObject

CCObject 인터페이스는 기존 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 wwwww 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 CCMenuItemAction[] CCMenuItemAction array
 */
public CCMenuItemAction[] 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 CCMenuItem
{
/**
 * 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(514 페이지의 『MySample.java』 참조)
2. 모든 데이터베이스 오브젝트의 조치 갱신(515 페이지의 『MyDatabaseActions.java』 참조)
3. 새로운 인스턴스 오브젝트 추가(516 페이지의 『MyInstance.java』 참조)
4. DB2 인스턴스의 조치 갱신(516 페이지의 『MyDB2.java』 참조)
5. 데이터베이스 폴더의 조치 갱신(517 페이지의 『MyDatabases.java』 참조)
6. SYSIBM.SYSPLAN 테이블의 조치 갱신(518 페이지의 『MySYSPLAN.java』 참조)
7. 새로운 테이블 오브젝트 추가(518 페이지의 『MyTable.java』 참조)
8. 응용프로그램 오브젝트 아래에 있는 DB_User 오브젝트의 조치 갱신(519 페이지의 『MyDBUser.java』 참조)
9. 제어 센터 도구 모음에 버튼 추가(520 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 방법에 대해 설명합니다.

제품 정보에 온라인으로 액세스하려면 정보 센터를 이용할 수 있습니다. 자세한 내용은 539 페이지의 『정보 센터로 정보에 액세스』를 참조하십시오. 웹에서 타스크 정보, DB2 책, 문제점 해결 정보, 샘플 프로그램 및 DB2 정보를 볼 수 있습니다.

DB2 PDF 파일 및 인쇄된 책

DB2 정보

다음의 표는 DB2 책을 네 개의 범주로 나눕니다.

DB2 안내 및 참조 정보

이 책에는 모든 플랫폼에 공통적인 DB2 정보가 들어 있습니다.

DB2 설치 및 구성 정보

이들 책은 특정 플랫폼에서의 DB2에 대한 것입니다. 예를 들어, OS/2, Windows 및 UNIX 기반 플랫폼에서의 DB2용으로 각각 다른 빠른 시작 책이 있습니다.

HTML 형식의 크로스 플랫폼 샘플 프로그램

이들 샘플은 응용프로그램 개발 클라이언트와 함께 설치된 샘플 프로그램의 HTML 버전입니다. 이들은 단지 정보용으로서 실제 프로그램을 바꾸지는 않습니다.

릴리스 정보

이러한 파일에는 DB2 책에 포함되지 않은 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 DB2 책 CD-ROM에서 Adobe Acrobat(PDF) 형식으로 제공됩니다. 또한 IBM에서 인쇄된 책을 주문할 수 있습니다. 534 페이지의 『인쇄된 책 주문』을 참조하십시오. 다음 표에는 주문할 수 있는 책을 보여줍니다.

OS/2 및 Windows 플랫폼에서는 `sqllib\doc\html` 디렉토리에 HTML 파일을 설치할 수 있습니다. DB2 정보는 여러 언어로 번역되었습니다. 그러나 모든 정보가 모든 언어로 번역된 것은 아닙니다. 정보를 특정 언어로 사용할 수 없을 경우에는 영문으로 제공됩니다.

UNIX 플랫폼에서는 `doc/%L/html` 디렉토리에 여러 언어 버전의 HTML 파일을 설치할 수 있습니다. 여기서 `%L`은 로케일을 나타냅니다. 자세한 내용은 빠른 시작 책을 참조하십시오.

다음의 여러 가지 방법으로 DB2 책을 구하고 정보에 액세스할 수 있습니다.

- 538 페이지의 『온라인 정보 보기』
- 542 페이지의 『온라인 정보 검색』
- 534 페이지의 『인쇄된 책 주문』
- 533 페이지의 『PDF 책 인쇄』

표 53. 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 구현, 데이터베이스 액세스, 감사, 백업 및 복구와 같은 구현에 대한 정보를 제공합니다.</p> <p>관리 안내서: 성능에서는 데이터베이스의 환경, 응용프로그램 성능 평가 및 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>	db2d0
<i>Administrative API Reference</i>	데이터베이스를 관리하는 데 사용할 수 있는 DB2 API와 데이터 구조에 대해 설명합니다. 또한 응용프로그램에서 API를 호출하는 방법에 대해 설명합니다.	SC09-2947 db2b0x70	db2b0
응용프로그램 빌드 안내서	환경 설정 정보와 Windows, OS/2 및 UNIX 기반 플랫폼에서 DB2 응용프로그램을 컴파일, 링크 및 수행하는 방법에 대한 단계별 지침을 제공합니다.	SA30-0991 db2axx70	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	<p>DB2 Universal Database 제품을 사용할 때 발생할 수 있는 APPC, CPI-C 및 SNA 감지 코드에 대한 일반 정보를 제공합니다.</p> <p>HTML 형식으로만 사용할 수 있습니다.</p>	문서 번호 없음 db2apx70	db2ap

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
응용프로그램 개발 안내서	Embedded SQL 또는 Java(JDBC 및 SQLJ)를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램을 개발하는 방법에 대해 설명합니다. 저장 프로시저어 작성, 사용자 정의 함수(UDF) 작성, 사용자 정의 유형 작성, 트리거 사용, 파티션된 환경 또는 연합 시스템에서 응용프로그램을 개발하는 등의 다양한 주제가 다루어집니다.	SA30-0992 db2a0x70	db2a0
CLI Guide and Reference	Microsoft ODBC 스펙과 호환 가능한 DB2 콜 레벨 인터페이스 및 호출 가능 SQL 인터페이스를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램을 개발하는 방법에 대해 설명합니다.	SC09-2950 db210x70	db210
Command Reference	명령행 처리기를 사용하는 방법 및 데이터베이스를 관리하기 위해 사용할 수 있는 DB2 명령에 대해 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 리퀘스터(AR)로 사용하는 방법에 대한 참조 정보 및 설치 정보를 제공합니다. 또한 DB2 Connect 응용프로그램 리퀘스터(AR)와 함께 DRDA 응용프로그램 서버를 사용하는 방법에 대해서도 상세히 설명합니다. HTML 및 PDF 형식으로만 사용할 수 있습니다.	문서 번호 없음 db2h1x70	db2h1
데이터 이동 유틸리티 안내 및 참조서	가져오기, 내보내기, 로드, 자동 로드 프로그램 및 DPROP와 같이 데이터 이동을 용이하게 해 주는 DB2 UDB 유틸리티의 사용 방법에 대해 설명합니다.	SA30-0994 db2dmx70	db2dm
Data Warehouse Center 관리 안내서	Data Warehouse Center를 사용하여 데이터 웨어하우스를 빌드 및 유지보수하는 방법에 대한 정보를 제공합니다.	SA30-1000 db2ddx70	db2dd
Data Warehouse Center 응용프로그램 통합 안내서	프로그래머들이 Data Warehouse Center 및 Information Catalog Manager를 응용프로그램과 통합하는 데 도움을 주는 정보를 제공합니다.	SA30-1001 db2adx70	db2ad

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<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 태스크 정보를 제공합니다.	SC09-2958 db2dwx70	db2dw
<i>DB2 Query Patroller User's Guide</i>	DB2 Query Patroller의 도구 및 함수를 사용하는 방법에 대해 설명합니다.	SC09-2960 db2wwx70	db2ww
용어집	DB2에서 사용되는 용어와 그 구성요소에 대한 정의를 제공합니다. HTML 형식과 SQL 참조서에서 사용할 수 있습니다.	문서 번호 없음 db2t0x70	db2t0
<i>Image, Audio</i> 및 <i>Video Extenders</i> 관리 및 프로그래밍	DB2 Extender에 대한 일반 정보와, 이미지, 오디오 및 비디오(IAV) Extenders 관리 및 구성에 대한 정보, 그리고 IAV Extenders를 사용한 프로그래밍에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1043 dmbu7x70	dmbu7
<i>Information Catalog Manager Administration Guide</i>	정보 카탈로그 관리에 대한 지침을 제공합니다.	SC26-9995 db2dix70	db2di
<i>Information Catalog Manager Programming Guide and Reference</i>	Information Catalog Manager에 대한 아키텍처 인터페이스에 대한 정의를 제공합니다.	SC26-9997 db2bix70	db2bi
<i>Information Catalog Manager</i> 사용자 안내서	Information Catalog Manager 사용자 인터페이스 사용에 대한 정보를 제공합니다.	SA30-1002 db2aix70	db2ai
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 분산 요청(DR)의 구성 및 이기종 데이터 소스에 액세스 등에 대한 정보가 들어 있습니다.	GA30-0975 db2iyx70	db2iy

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
메시지 참조서	DB2, Information Catalog Manager 및 Data Warehouse Center가 발행하는 메시지와 코드를 나열하고 수행해야 할 조치에 대해 설명합니다.	볼륨 1 GA30-0986 볼륨 2 SA30-0987	db2m0
	문서 번호(SBOF-8932)를 사용하여 두 권으로 된 메시지 참조서 책을 모두 주문할 수 있습니다.	db2m1x70 db2m2x70	
<i>OLAP Integration Server Administration Guide</i>	OLAP Integration Server의 관리 프로그램 구성요소를 사용하는 방법에 대해 설명합니다.	SC27-0782 db2dpx70	n/a
<i>OLAP Integration Server Metaoutline User's Guide</i>	표준 OLAP Metaoutline 인터페이스 (Metaoutline Assistant가 아닌)를 사용하여 OLAP Metaoutlines를 작성하고 처리하는 방법에 대해 설명합니다.	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	표준 OLAP 모델 인터페이스(Model Assistant가 아닌)를 사용하여 OLAP 모델을 작성하는 방법에 대해 설명합니다.	SC27-0783 db2lpx70	n/a
<i>OLAP Setup and User's Guide</i>	OLAP Starter Kit에 대한 구성 및 설치 정보를 제공합니다.	SA30-1074 db2lpx70	db2ip
<i>Excel용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Excel 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법에 대해 설명합니다.	SA30-1094 db2epx70	db2ep
<i>Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Lotus 1-2-3 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법에 대해 설명합니다.	SA30-1093 db2tpx70	db2tp
복제 안내 및 참조서	DB2와 함께 제공된 IBM 복제 도구에 관한 플랜, 구성, 관리 및 사용 정보를 제공합니다.	SA30-1003 db2e0x70	db2e0
<i>Spatial Extender 사용자 안내 및 참조서</i>	Spatial Extender 설치, 구성, 관리, 프로그래밍 및 문제점 해결에 대한 정보를 제공합니다. 또한 공간 데이터 개념에 대한 설명을 제공하고 Spatial Extender에 대한 특정 참조 정보(메시지 및 SQL)를 제공합니다.	SA30-1045 db2sbx70	db2sb

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
SQL 시작하기	SQL 개념을 소개하고 많은 구조와 태스크에 관한 예를 제공합니다.	SA30-0996	db2y0
			db2y0x70
SQL 참조서, 볼륨 1 및 볼륨 2	SQL 구문, 의미 및 규칙에 대해 설명합니다. 또한 릴리스 간 비호환성, 제품 제한사항 및 카탈로그 뷰에 대한 정보도 들어 있습니다. SBOF-8933 문서 번호를 사용하여 SQL 참조서를 주문할 수 있습니다.	볼륨 1 SA30-0997	db2s0
			db2s1x70
		볼륨 2 SA30-0998	
			db2s2x70
시스템 모니터 안내 및 참조서	데이터베이스와 데이터베이스 관리자에 대한 다른 종류의 정보를 수집하는 방법에 대해 설명합니다. 이 책은 데이터베이스 활동을 이해하고 성능을 향상시키며 문제점의 원인을 판별하기 위한 정보를 사용하는 방법에 대해 설명합니다.	SA30-0995	db2f0
			db2f0x70
Text Extender 관리 및 프로그래밍	DB2 Extenders에 대한 일반 정보, Text Extenders 관리 및 구성에 관한 정보, Text Extenders를 사용한 프로그래밍에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1044	desu9
			desu9x70
문제점 해결 안내서	오류의 소스를 판별하고 문제점으로부터 복구하며 DB2 고객 서비스와 상담하여 진단 도구를 사용하는 것을 도와줍니다.	GA30-0704	db2p0
			db2p0x70
새로운 기능	DB2 Universal Database 버전 7의 새로운 특성, 기능 및 향상된 내용에 대해 설명합니다.	SA30-0999	db2q0
			db2q0x70
DB2 설치 및 구성 정보			
OS/2 및 Windows용 DB2 Connect Enterprise Edition 빠른 시작	OS/2 및 Windows 32비트 운영 체제에서 DB2 Connect Enterprise Edition에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0974	db2c6
			db2c6x70

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
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 32비트 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 Linux 분산에서 DB2 Connect Personal Edition에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2962 db2c4x70	db2c4
DB2 Data Links Manager 빠른 시작	AIX 및 Windows 32비트 운영 체제용 DB2 Data Links Manager에 대한 플랜, 설치, 구성 및 타스크 정보를 제공합니다.	GA30-0980 db2z6x70	db2z6
UNIX용 DB2 Enterprise - Extended Edition 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
Windows용 DB2 Enterprise - Extended Edition 빠른 시작	Windows 32비트 운영 체제용 DB2 Enterprise - Extended Edition에 대한 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0977 db2v6x70	db2v6
OS/2용 DB2 빠른 시작	OS/2 운영 체제에서의 DB2 Universal Database에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0982 db2i2x70	db2i2
UNIX용 DB2 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Universal Database에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0984 db2ixx70	db2ix

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
Windows용 DB2 빠른 시작	Windows 32비트 운영 체제에서의 DB2 Universal Database에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0985 db2i6x70	db2i6
DB2 Personal Edition 빠른 시작	OS/2 및 Windows 32비트 운영 체제에서의 DB2 Universal Database Personal Edition에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GA30-0983 db2i1x70	db2i1
DB2 Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux 분산에서 DB2 Universal Database Personal Edition에 대한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2972 db2i4x70	db2i4
DB2 Query Patroller 설치 안내서	DB2 Query Patroller에 대한 설치 정보를 제공합니다.	GA30-0976 db2iwx70	db2iw
DB2 Warehouse Manager 설치 안내서	웨어하우스 에이전트, 웨어하우스 변환기 및 Information Catalog Manager에 대한 설치 정보를 제공합니다.	GA30-1027 db2idx70	db2id
HTML 형식의 크로스 플랫폼 샘플 프로그램			
HTML 형식의 샘플 프로그램	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 때만 사용할 수 있습니다. 프로그램에 대한 자세한 내용은 응용프로그램 빌드 안내서를 참조하십시오.	문서 번호 없음	db2hs
릴리스 정보			
DB2 Connect 릴리스 정보	DB2 Connect 책에는 포함될 수 없었던 최신 정보를 제공합니다.	#2를 참조하십시오.	db2cr
DB2 설치 정보	DB2 책에는 포함될 수 없었던 최신 설치 정보를 제공합니다.	제품 CD-ROM에서만 사용할 수 있습니다.	

표 53. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
			PDF 파일 이름
DB2 릴리스 정보	DB2 책에는 포함될 수 없었던 모든 DB2 제품 #2를 참조하십시오. 및 기능에 대한 최신 정보를 제공합니다.		db2ir

주:

1. 파일 이름의 6번째 자리에 있는 문자 *x*는 책의 언어 버전을 나타냅니다. 예를 들어, 파일 이름 db2d0e70은 관리 안내서 책의 영문 버전을 나타내며 db2d0k70은 같은 책의 한글 버전을 나타냅니다. 다음 문자는 파일 이름의 6번째 자리에 사용되어 언어 버전을 나타냅니다.

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
한국어	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 이름을 나타내고 DB2DIR은 다음과 같습니다.
 - AIX에서는 /usr/lpp/db2_07_01
 - HP-UX, PTX, Solaris 및 Silicon Graphics IRIX에서는 /opt/IBMdb2/V7.1
 - Linux에서는 /usr/IBMdb2/V7.1
- 다른 플랫폼의 경우에는 RELEASE.TXT 파일을 참조하십시오. 이 파일은 제품이 설치된 디렉토리에 있습니다. OS/2 플랫폼에서는 **IBM DB2** 폴더를 더블 클릭한 다음 릴리스 정보 아이콘을 더블 클릭할 수 있습니다.

PDF 책 인쇄

인쇄된 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 대한 자세한 내용은 525 페이지의 표53을 참조하십시오.

Adobe 웹 사이트 <http://www.adobe.com>에서 Adobe Acrobat Reader의 최신 버전을 얻을 수 있습니다.

PDF 파일은 파일 확장자가 PDF인 DB2 책 CD-ROM에 들어 있습니다. PDF 파일에 액세스하려면 다음을 수행하십시오.

1. DB2 책 CD-ROM을 삽입하십시오. UNIX 기반의 플랫폼에서는 DB2 책 CD-ROM을 마운트합니다. 마운트 프로시듀어에 대한 자세한 내용은 빠른 시작 책을 참조하십시오.
2. Acrobat Reader를 시작하십시오.
3. 다음 위치 중 하나에서 원하는 PDF 파일을 여십시오.
 - OS/2 및 Windows 플랫폼에서
 $x:\backslash\text{doc}\backslash\text{language}$ 디렉토리. 여기서 x 는 CD-ROM 드라이브를 나타내며 language 는 사용자 언어를 나타내는 2문자 국가 코드를 나타냅니다. 예를 들어, 영문인 경우에는 EN입니다.
 - UNIX 기반 플랫폼에서

`/cdrom/doc/%L` 디렉토리. 여기서 `/cdrom`은 CD-ROM의 마운트 지점을 나타내고 `%L`은 원하는 로케일의 이름을 나타냅니다.

또한 PDF 파일을 CD-ROM에서 지역이나 네트워크 드라이브로 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 페이지 <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 기능에 대한 참조 및 사용 정보를 제공합니다. 각 SBOF의 내용은 다음 표에 나열되어 있습니다.

표 54. 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, and 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 설치 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 Administration and Programming Guide • 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 사용 가능합니다. 다음의 표에서는 다양한 도움말 유형에 대해 설명합니다.

도움말 유형	내용	액세스하는 방법
명령 도움말	명령행 처리기의 명령 구문에 대해 설명합니다.	대화식 모드의 명령행 처리기에서 다음을 입력하십시오. <code>? command</code> 여기서 <code>command</code> 는 키워드이거나 전체 명령입니다. 예를 들어, <code>? catalog</code> 는 모든 CATALOG 명령에 대한 도움말을 표시하고, <code>? catalog database</code> 는 CATALOG DATABASE 명령에 대한 도움말을 표시합니다.
클라이언트 구성 지원 프로그램 도움말	창 또는 노트북에서 수행할 수 있는 task에 대해 설명합니다.	창이나 노트북에서 도움말 버튼을 누르거나 F1 키를 누르십시오.
명령 센터 도움말	도움말은 알아야 할 개요와 전제조건 정보를 포함하고, 창 또는 노트북 제어를 사용하는 방법에 대해 설명합니다.	
제어 센터 도움말		
Data Warehouse Center 도움말		
이벤트 분석기 도움말		
Information Catalog Manager 도움말		
위성 관리 센터 도움말		
스크립트 센터 도움말		

도움말 유형	내용	액세스하는 방법
메시지 도움말	메시지의 원인과 사용자가 취해야 할 조치에 대해 설명합니다.	대화식 모드의 명령행 처리기에서 다음을 입력하십시오. <code>? XXXnnnnn</code> 여기서 <code>XXXnnnnn</code> 은 유효한 메시지 식별자입니다. 예를 들어, <code>? SQL30081</code> 은 <code>SQL30081</code> 메시지에 대한 도움말을 표시합니다. 한 번에 한 화면씩 메시지 도움말을 보려면 다음을 입력하십시오. <code>? XXXnnnnn more</code> 파일에 메시지 도움말을 저장하려면 다음을 입력하십시오. <code>? XXXnnnnn > filename.ext</code> 여기서 <code>filename.ext</code> 는 메시지 도움말을 저장하려는 파일입니다.
SQL 도움말	SQL문의 구문에 대해 설명합니다.	대화식 모드의 명령행 처리기에서 다음을 입력하십시오. <code>help statement</code> 여기서 <code>statement</code> 는 SQL문입니다. 예를 들어, <code>help SELECT</code> 는 <code>SELECT</code> 문에 대한 도움말을 표시합니다. 주: SQL 도움말은 UNIX 기반 플랫폼에서 사용할 수 없습니다.
SQLSTATE 도움말	SQL 상태와 클래스 코드에 대해 설명합니다.	대화식 모드의 명령행 처리기에서 다음을 입력하십시오. <code>? sqlstate</code> 또는 <code>? class code</code> 여기서 <code>sqlstate</code> 는 유효한 5자리 숫자로 된 SQL 상태이고 <code>class code</code> 는 SQL 상태의 처음 2자리 숫자입니다. 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.

온라인 정보 보기

이 제품에 들어 있는 책은 HTML(Hypertext Markup Language) 소프트웨어 형식으로 제공됩니다. 소프트웨어 형식은 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 다음을 수행하십시오.

- DB2 관리 도구를 수행할 경우, 정보 센터를 사용하십시오.
- 브라우저에서 파일 → 페이지 열기를 누르십시오. 열린 페이지에 DB2 정보에 대한 설명과 링크가 들어 있습니다.
 - UNIX 기반 플랫폼의 경우, 다음 페이지를 여십시오.

```
INSTHOME/sqlllib/doc/%L/html/index.htm
```

여기서 %L은 로케일 이름입니다.

- 기타 플랫폼에서는 다음 페이지를 여십시오.

```
sqlllib\doc\html\index.htm
```

이 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 지침을 보려면 다음을 수행하십시오.

1. Netscape CD-ROM을 삽입하십시오.
2. UNIX 기반 플랫폼에서는 CD-ROM을 마운트하십시오. 마운트 프로시저에 대한 자세한 내용은 빠른 시작 책을 참조하십시오.

3. 설치 지침의 경우에는 CDNAVnn.txt 파일을 참조하십시오. 여기서 nn은 2문자로 된 언어 식별자입니다. 파일은 CD-ROM의 루트 디렉토리에 있습니다.

정보 센터로 정보에 액세스

정보 센터는 DB2 제품 정보에 대한 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 센터에 액세스할 수 있습니다.

정보 센터는 6개 유형의 정보를 제공합니다. 적절한 탭을 눌러 해당 유형에 제공되는 주제를 보십시오.

타스크	DB2를 사용하여 수행할 수 있는 주요 타스크.
참조	키워드, 명령 및 API와 같은 DB2 참조 정보.
책	DB2 책.
문제점 해결	오류 메시지의 범주와 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 시스템에서 웹으로의 연결을 갖고 있어야 합니다.

목록 중 하나에서 항목을 선택하면 정보 센터가 표시기를 시작하여 정보를 표시합니다. 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 검색의 경우, **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 누르고 프로그램 → IBM DB2 → 정보 → HTML 검색 서버 시작을 선택하십시오.

OS/2의 경우

OS/2용 DB2 폴더를 더블 클릭한 다음 HTML 검색 서버 시작 아이콘을 더블 클릭하십시오.

HTML 정보 검색시 그 외의 다른 문제가 발생한 경우에는 릴리스 정보를 참조하십시오.

주: 검색 기능은 Linux, PTX 및 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.

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 표에서는 마법사를 나열하고 해당 기능에 대해 설명합니다.

주: 데이터베이스 작성, 색인 작성, 다중 사이트 갱신 구성 및 성능 구성 마법사는 파티션된 데이터베이스 환경에서 사용할 수 있습니다.

마법사	도움 내용	액세스하는 방법
데이터베이스 추가	클라이언트 워크스테이션의 데이터베이스를 카탈로그화합니다.	클라이언트 구성 지원 프로그램에서 추가를 누르십시오.
데이터베이스 백업	백업 플랜의 결정, 작성 및 스케줄합니다.	제어 센터에서 백업하려는 데이터베이스를 마우스 오른쪽 단추로 누른 다음 백업 → 마법사를 사용한 데이터베이스를 선택하십시오.
다중 사이트 갱신 구성	다중 사이트 갱신, 분산 트랜잭션 또는 2단계 확약을 구성합니다.	제어 센터에서 데이터베이스 폴더를 마우스 오른쪽 단추로 누른 다음 다중 사이트 갱신을 선택하십시오.

마법사	도움 내용	액세스하는 방법
데이터베이스 작성	데이터베이스 작성 및 일부 기본 구성 작업을 수행합니다.	제어 센터에서 데이터베이스 폴더를 마우스 오른쪽 단추로 누른 다음 작성 → 마법사를 사용한 데이터베이스를 선택하십시오.
테이블 작성	기본 데이터 유형의 선택 및 테이블의 기본 키를 작성합니다.	제어 센터에서 테이블 아이콘을 마우스 오른쪽 단추로 누른 다음 작성 → 마법사를 사용한 테이블을 선택하십시오.
테이블 공간 작성	새로운 테이블 공간을 작성합니다.	제어 센터에서 테이블 공간 아이콘을 마우스 오른쪽 단추로 선택한 다음 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
색인 작성	사용자의 모든 조회를 작성하고 삭제하기 위해 색인 회합합니다.	제어 센터에서 색인 아이콘을 마우스 오른쪽 단추로 누른 다음 작성 → 마법사를 사용한 색인을 선택하십시오.
성능 구성	비즈니스 요구에 맞게 구성 매개변수를 갱신하여 데이터베이스의 성능을 조정합니다.	제어 센터에서 조정하려는 데이터베이스를 마우스 오른쪽 단추로 누른 다음 마법사를 사용한 성능 구성을 선택하십시오. 파티션된 데이터베이스 환경에 대해 데이터베이스 파티션 뷰에서 조정하려는 첫 번째 데이터베이스 파티션을 마우스 오른쪽 단추로 누른 다음 마법사를 사용한 성능 구성을 선택하십시오.
데이터베이스 복원	실패 후에 데이터베이스를 복구합니다. 사용할 백업 종류 및 재작동할 로그를 이해하는 데 도움을 줍니다.	제어 센터에서 복원하려는 데이터베이스를 마우스 오른쪽 단추로 누른 다음 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.

문서 서버 설정

기본값으로 DB2 정보는 지역 시스템에 설치됩니다. 이는 DB2 정보에 액세스해야 하는 모든 사람이 동일한 파일을 설치해야 함을 의미합니다. DB2 정보를 단일 위치에 저장하려면 다음 단계를 수행하십시오.

1. 모든 파일과 서브디렉토리를 지역 시스템의 `\sqllib\doc\html`에서 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. 자세한 내용은 설치 및 구성 보충 설명서의 부록 NetQuestion을 참조하십시오.
3. Java 버전의 정보 센터를 사용 중인 경우, 모든 HTML 파일에 대한 기본 URL을 지정할 수 있습니다. 책 목록에 대한 URL을 사용해야 합니다.
4. 책 파일을 볼 수 있게 되면 다음과 같이 자주 보는 주제 항목에 대해서는 책갈피를 설정할 수 있습니다. 다음의 페이지들을 북마크로 설정해 두면 도움이 될 것입니다.
 - 책 목록
 - 자주 사용되는 책의 목차
 - ALTER TABLE 주제와 같은 자주 참조하는 항목
 - 검색 양식

DB2 Universal Database 온라인 문서 파일을 중앙 머신에서 제공하는 방법에 대한 자세한 내용은 설치 및 구성 보충 설명서의 부록 NetQuestion을 참조하십시오.

온라인 정보 검색

HTML 파일에서 정보를 찾으려면 다음 방법 중 하나를 사용하십시오.

- 맨 위 프레임에서 검색을 누르십시오. 특정 주제를 찾으려면 검색 양식을 사용하십시오. 이 기능은 Linux, PTX 또는 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.
- 맨 위 프레임에서 색인을 누르십시오. 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 책갈피 기능을 사용하십시오.
- 특정 주제를 찾으려면 정보 센터의 검색 기능을 사용하십시오. 자세한 내용은 539 페이지의 『정보 센터로 정보에 액세스』를 참조하십시오.

부록M. 주의사항

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 사용권까지 부여하는 것은 아닙니다. 사용권에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터
전화번호: 080-023-8080

2 바이트(DBCS) 정보에 관한 사용권 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통고 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트의 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램 및 기타 프로그램(이 프로그램 포함)간의 정보 교환 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 사용권자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들면, 사용권 지불 포함)에 따라 사용할 수 있습니다.

이 책에 기술된 사용권 프로그램 및 이 프로그램에 사용가능한 모든 사용권 데이터는 IBM 고객 협약, IBM 국제 프로그래밍 사용권 협약 또는 이와 동등한 모든 협약 조건하에 IBM에서 제공됩니다.

여기에 있는 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서, 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 이러한 측정치가 일반적으로 사용가능한 시스템에서도 동일하다고는 보장하지 않습니다. 더우기, 일부 측정치는 추정을 통해 측정되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 본인의 특정 환경에 적용할 수 있는 데이터를 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 배상 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지없이 변경될 수 있습니다.

이 정보에는 일상의 업무에서 사용되는 자료와 보고의 예제가 포함되어 있을 수 있습니다. 가능한 완벽하게 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 예제에 들어 있습니다. 이들 이름은 모두 가공의 것이며, 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권:

이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 포함되어 있을 수 있습니다. 샘플 응용프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램을 개발, 사용, 마케팅 또는 배포를 목적으로 이들 샘플 프로그램을 복사, 수정 및 배포할 수 있으며 IBM에 대한 지불 의무는 없습니다. 이들 예제 프로그램은 모든 조건에서 철저히 검사된 것은 아닙니다. 따라서, IBM은 이들 프로그램의 신뢰성, 서비스 가능성 또는 기능에 대해 어떠한 보증도 하지 않습니다.

각 사본이나 이들 샘플 프로그램의 일부 또는 파생본에는 다음과 같은 저작권 주의사항을 포함시켜야 합니다.

© (귀하의 회사명) (연도). 이 코드 부분은 IBM 샘플 프로그램에 나와 있습니다.
© Copyright IBM Corp. _연도 입력_. All rights reserved.

상표

별표(*)로 표시된 다음 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 CICS	OS/390
C Set++	OS/400
C/370	PowerPC
DATABASE 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000S/370
DB2	SP
DB2 ConnectDB2 Extenders	SQL/DS
DB2 OLAP Server	SQL/400
DB2 Universal Database	System/370
Distributed Relational	System/390
Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

다음 용어는 기타 회사의 상표 또는 등록상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록상표입니다.

Java 또는 모든 Java 관련 상표와 로고 및 Solaris는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Tivoli 및 NetView는 미국 또는 기타 국가에서 사용되는 Tivoli Systems Inc.의 상표입니다.

UNIX는 미국 또는 기타 국가에서 X/Open Company Limited가 독점적인 사용권을 가진 등록상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스표입니다.

색인

[가]

- 가상 인터페이스(VI) 아키텍처 457, 459
- 감사 기능
 - 감사 이벤트 테이블 335
 - 구문 330
 - 권한/특권 325
 - 동기 레코드 쓰기 328
 - 레코드 레이아웃 335
 - 매개변수 설명 331
 - 메시지 334
 - 바동기 레코드 쓰기 328
 - 사용 시나리오 330
 - 예 354
 - 오류 조절 328
 - 이벤트 326
 - 이벤트 테이블 점검 336
 - 작동 327
 - 조치 326
 - 추가 정보 및 기술 353
 - 활동 제어 354
 - CONTEXT 이벤트 테이블 351
 - ERRORTYPE 매개변수 328
 - OBJMAINT 이벤트 테이블 341
 - SECMAINT 이벤트 테이블 342
 - SYSADMIN 이벤트 테이블 347
 - VALIDATE 이벤트 테이블 350
- 감사 추적 325
- 검색
 - 온라인 정보 539, 542
- 검색 발견
 - 추가 설정 109
- 게이트웨이 연결 19
- 경량 디렉토리 액세스 프로토콜 129
 - 검색 483
 - 노드 항목 카탈로그화 479
- 경량 디렉토리 액세스 프로토콜 129
 - (계속)
 - 데이터베이스 등록 479
 - 데이터베이스 등록해제 481
 - 디렉토리 스키마 확장 490
 - 레지스트리 변수 설정 486
 - 보안 488
 - 사용 가능 486
 - 사용 불가능 487
 - 서버 등록 해제 479
 - 오브젝트 클래스 및 속성 494
 - 원격으로 접속 480
 - 프로토콜 정보 갱신 478
 - 항목 새로 고침 481
 - 호스트 데이터베이스 구성 484
 - DB2 Connect 488
 - IBM eNetwork 디렉토리 491
 - Windows 2000 active directory 491
- 경량 디렉토리 액세스 프로토콜 (LDAP) 56, 471
- 계층 구조 테이블 160
 - 삭제 247
- 고속 통신 455
- 고유 제한조건
 - 정의 147
 - 제거 233
 - 추가 230
- 고유 제한조건 정의 147
- 고유 제한조건 제거 233
- 고유 제한조건 추가 230
- 고유 키
 - 삭제에 필요한 권리 233
 - 작성할 시기 148
 - 추가 230
- 고유 키 (계속)
 - 1차 색인 148
 - 1차 색인, 작성 194
 - DROP PRIMARY KEY절, ALTER TABLE문 233
 - 공용 상호연결 456
 - 관련 항목 오브젝트 표시 8
 - 관련됨 361
 - 구성 117
 - 관리
 - GUI 도구 사용 3
 - 관리 도구
 - 개요 4
 - 명령 센터 24
 - 스크립트 센터 24
 - 관리 서버 17, 94
 - 구성 매개변수
 - 파티션된 데이터베이스 70
 - DCE(Distributed Computing Environment) 391
 - 구조화 유형
 - 변경 244
 - 구조화 유형 변경 244
 - 권한 291
 - 권한 부여 47
 - 권한 취소 47
 - 데이터베이스 관리(DBADM) 294, 297
 - 레벨 289
 - 시스템 유지보수(SYSMAINT) 293
 - 시스템 제어(SYSCTRL) 292
 - 타스크 및 필수 권한 부여 318
 - SYSADM으로부터 DBADM 제거 292

권한 291 (계속)
 SYSCTRL로부터 DBADM 제거
 293
 권한 부여
 시스템 관리(SYSADM) 291
 신뢰성 있는 클라이언트 269
 정의 288
 권한 취소문
 내재적인 발행(issuance) 310
 보안 391
 사용 308
 예 308
 권한부여 이름
 권한 부여된 특권의 검색 322
 테이블 액세스 권한을 갖는 이름 검색
 321
 특권 정보를 검색 321
 특권 정보에 대한 뷰 작성 324
 DBADM 권한을 갖는 이름 검색
 321
 규정화된 오브젝트 이름 65
 기록(logging)
 raw 장치 137
 기본 속성 스펙 143

[나]

널(NULL) 값
 컬럼 정의 143
 노드 66
 노드 그룹에서 변경 217
 모든 노드에 걸쳐 데이터베이스 작성
 69
 카탈로그화 69
 노드 간 통신 455
 노드 구성 파일
 변경 214
 작성 114
 노드 구성 파일 변경 214
 노드 그룹
 변경 217

노드 그룹 (계속)
 작성 129
 처음 정의 123
 테이블 고려사항 161
 파티션 키, 변경 239
 IBMDEFAULTGROUP, 기본값에 따
 라 작성된 테이블 161
 노드 그룹 변경 217
 노드 레벨 프로파일 레지스트리 85
 노드 번호 115
 노드 서버 옵션 188
 논리 노드
 다중 453

[다]

다중 사이트 갱신 구성 마법사 6, 540
 다중 테이블 공간에 테이블 작성 160
 대소문자 구별명, 연합 데이터베이스
 374
 대형 오브젝트(LOB)
 컬럼 고려사항 145
 데이터
 분산 변경 217
 이동 48
 데이터 검색
 색인 196
 데이터 무결성
 고유 색인 193
 데이터 보안
 데이터베이스 액세스 제어 263
 시스템 카탈로그의 보안 323
 중요성 263
 데이터 복제 55, 361
 데이터 암호화 317
 데이터 유형
 다중 바이트 문자 세트 143
 컬럼 정의 143
 데이터 이동 361

데이터 재분배
 노드에 걸쳐 217
 데이터 전송
 개요 361
 데이터 파티션 나누기 68
 데이터베이스 복원 마법사 6
 데이터베이스 61
 데이터 분산 변경 217
 모든 노드에 걸쳐 작성 69
 변경 216
 변경 노드 그룹 217
 변경전 고려사항 209
 복구 로그 130
 원격, 관리 44
 작동 가능한 데이터 파티션 나누기
 68
 작성 121
 작성에 대한 고려사항 73
 작성하기 전 61
 제거 217
 카탈로그화 131
 패키지 종속성 257
 데이터베이스 관리 프로그램
 색인 196
 시작 62
 액세스 제어 306
 유틸리티 바인딩 131
 중지 72
 데이터베이스 관리자(DBADM) 권한
 이름 검색 321
 특권 294
 데이터베이스 구성
 변경 214
 작성된 파일 116
 데이터베이스 구성 변경 214
 데이터베이스 백업 마법사 5, 540
 데이터베이스 설계
 변경 209
 데이터베이스 액세스
 제어 263

데이터베이스 액세스 (계속)
SQL이 있는 패키지를 통한 특권
311

데이터베이스 오브젝트

액세스 제어 306

예 378

작성 378

데이터베이스 위치 지정자 오브젝트

예 380

작성 380

데이터베이스 작성 마법사 5, 540

데이터베이스 제거 217

데이터베이스 추가 마법사 540, 541

데이터베이스 카탈로그화 131

예 131

데이터베이스 파티션 서버

명령 발행 411

Windows 2000 447

Windows NT 447

도메인 보안

인증 436

동적 SQL

데이터베이스 액세스에 대한

EXECUTE 특권 311

디렉토리

노드 디렉토리 128

시스템 데이터베이스 디렉토리 127

지역 데이터베이스 디렉토리 126

디렉토리 오브젝트

오브젝트 클래스 속성 383

작성 378

디렉토리 캐쉬

카탈로그화 데이터베이스의 결과 132

[라]

라우팅 정보 오브젝트

예 382

작성 381

랩퍼 작성 181

랩퍼 제거 253

랩퍼, 작성 181

레지스트리 변수 84

변경 214

DCE(Distributed Computing

Environment) 391

레지스트리 변수 변경 214

레코드

감사 325

로그

감사 325

릴리스 정보 532

[마]

마법사 5

다중 사이트 갱신 구성 6, 540

데이터베이스 백업 5, 540

데이터베이스 복원 6, 541

데이터베이스 작성 5, 540

데이터베이스 추가 540, 541

색인 6, 541

성능 구성 6, 214, 541

타스크 완료 540

테이블 공간 작성 6, 541

테이블 작성 5, 541

마법사 성능 구성 214

머신 목록에서 중복된 항목 제거 421

메시지

감사 기능 334

명령

병렬 수행 415

CATALOG GLOBAL

DATABASE 393

명령 센터 24

명령행 처리기

데이터베이스에 바인딩 131

명시적 스키마 사용 65, 66

모니터링

rah 모니터링 416

문서 서버 설정 541

문자 직렬 장치 135

문제점 관별 425

문제점 해결 54

[바]

바인딩

데이터베이스 유틸리티 131

명령행 처리기 131

올바르지 않은 패키지의 리바인딩
310

발견

구성 113

매개변수 설정 111

서버 인스턴스 숨기기 111

백업 도메인 제거기

DB2 구성 434

범위

추가 227

범위 추가 227

별명

권한 179

사용 179

별명 작성 179

별명(MVS/ESA용 DB2) 180

별명, 작성 179

별칭

데이터 소스 뷰 313

작성 190

특권 303

패키지 특권 처리 312

별칭 변경 254

별칭 식별 192

별칭 작성 190

별칭 제거 254

병렬 처리

사용 가능 66

병렬 처리, 파티션 내

사용 가능 66

보기

온라인 정보 538

보안
 사용자 맵핑 284
 서버 옵션 284
 서비스, Windows NT 434
 연합 데이터베이스 ID 및 암호 처리 283
 연합 서버 인증 예 286
 연합 시스템에 대한 APPC 설정값 286
 인증, 연합 데이터베이스 282
 플랜 설정 263
 CLIENT 레벨 268
 DCE 디렉토리 서비스 389
 DCS 처리, 연합된 시스템 283
 복구
 개요 365, 409
 데이터베이스 작성시 로그 할당 130
 복구 로그 130
 복수 인스턴스 64
 복원 마법사 541
 뷰
 데이터 무결성 173
 데이터 보안 173
 변경 250
 사용 불가능 251
 시스템 카탈로그에 대한 내포 제거 250
 액세스 특권, 예 314
 작동 불능 요약 테이블 복구 251
 작성 173
 제거 250
 제한사항 250
 컬럼 액세스 313
 테이블에 대한 액세스 제어 313
 특권 정보 324
 행 액세스 313
 CHECK OPTION절, CREATE VIEW문 175
 뷰 변경 250
 뷰 작성 173

뷰 제거 250
 블록 구조화된 장치 135
 비고유 색인
 제거 257

[사]

사용권 관리 84
 사용권 센터 27
 사용권 정보
 변경 209
 사용자
 관리 47
 사용자 맵핑
 작성 284
 사용자 인증
 Windows NT 434
 사용자 임시 테이블 공간 136
 사용자 임시 테이블 공간 제거 225
 사용자 정의 구별 유형 작성 170
 사용자 정의 구별 유형, 작성 170
 사용자 정의 구조화 유형 작성 172
 사용자 정의 구조화 유형, 작성 172
 사용자 정의 유형 작성 170
 사용자 정의 유형 제거 249
 사용자 정의 유형(UDT)
 작성 170
 제거 249
 사용자 정의 임시 테이블 154, 247
 사용자 정의 테이블 제거 247
 사용자 정의 함수(UDF)
 비분리를 작성할 특권 296
 유형 166
 작성 166
 제거 248
 사용자 정의 함수(UDF) 작성 166
 사용자 정의 함수(UDF) 제거 248
 사용자 테이블 공간 제거 222
 색인
 고유하지 않은 198

색인 (계속)
 기본 148
 변경 256
 비1차 257
 사용된 방법 197
 사용자 정의 확장 201
 선택 207
 온라인 재구성 196, 200
 작성 193
 정의 194
 최적화 수 194
 특권 305
 1차 198
 1차 : 사용자 정의 194
 CREATE INDEX문 198
 CREATE UNIQUE INDEX문 198
 DROP INDEX문 257
 색인 마법사 6, 541
 색인 스택 제거 256
 색인 스펙 작성 193
 색인 작성 193
 색인 제거 256
 색인 키, 정의 194
 색인 확장 193
 색인 확장 작성 193
 색인 확장 제거 256
 샘플 프로그램
 크로스 플랫폼 531
 HTML 531
 서버
 작성 182
 특권 304
 서버 변경 254
 서버 옵션
 노드 188
 보안 세부사항 284
 암호 188, 285
 collating_sequence 186
 comm_rate 186
 connectstring 186

서버 옵션 (계속)

- cpu_ratio 187
- dbname 187
- fold_id 187, 285
- fold_pw 187, 285
- io_ratio 188
- plan_hints 188
- pushdown 189
- varchar_no_trailing_blanks 189

서버 작성 182

서버 제거 254

선택 207

설계, 구현 61

설정

- rah에 대한 기본 환경 프로파일 424

설치

- Netscape 브라우저 538

성능

- 값 재설정 445
- 요약 테이블 176
- 원격 정보 액세스 444
- 정보 표시 443
- 정보에 대한 원격 액세스 사용 442
- 카탈로그 정보, 경합 감소 69

성능 구성 마법사 6, 541

성능 모니터

- Windows NT 441

셸 디렉토리 서비스(CDS) 377

수치 문자열 컬럼옵션 255

순차 159

- 변경 241
- 작성 157
- 제거 242
- 특권 306

스칼라 UDF 166

스케줄

- 저장된 명령 스크립트 26

스크립트 센터 24

- 기존 스크립트 사용 25

스키마

- 개요 65
- 삭제 225
- 작성 140
- SESSION 247

스키마 설정 141

스키마 작성 140

스키마 제거 225

시스템 관리(SYSADM) 권한 291

- 개요 291
- 특권 291

시스템 데이터베이스 디렉토리

- 개요 127, 128

시스템 임시 테이블 공간 제거 223

시스템 카탈로그

- 보안 323
- 뷰 내포의 삭제 251
- 새로운 컬럼 추가 226
- 설정 125
- 이름에 대해 권한 부여된 특권의 검색 322
- 테이블 액세스 권한을 갖는 이름 검색 321
- 테이블 제거 245
- 특권 목록 320
- 특권이 있는 권한 부여 이름 검색 321
- DBADM 권한을 갖는 이름 검색 321

시스템 카탈로그 테이블

- 데이터베이스 카탈로그 노드에 저장 69

식별 컬럼 155

- 변경 241

신뢰 받지 못한 클라이언트 268

신뢰성 없는 클라이언트 269

신뢰성 있는 클라이언트

- 인증 269
- CLIENT 레벨 보안 268

[아]

암호 서버 옵션 188

암호화

- 데이터 317

암호화 암호 설정(SET ENCRYPTION PASSWORD) 318

액세스 제어 267

- 데이터베이스 관리 프로그램 306
- 데이터베이스 오브젝트 306
- 인증 267
- 테이블에 대한 뷰 313

언어 식별자

- 책 532

연합 데이터베이스

- 대소문자가 구별되는 이름 374
- 맵핑, 작성 181
- 별명, 작업 191
- 별칭 참조 191
- 별칭, 식별 192
- 별칭, 작성 190
- 사용자 맵핑, 작성 284
- 색인 스펙, 작성 193
- 서버 옵션, 보안 284
- 서버, 작성 182
- 유형 맵핑, 작성 172
- 인증 282
- 인증 예 286
- 함수 맵핑, 작성 167
- 함수 템플릿, 작성 168
- APPC 설정값 286
- DCS 설정값 283
- ID 및 암호를 데이터 소스로 전달 283

오브젝트

- 관련 항목 표시 8

오브젝트 클래스 속성

- DB_Authentication (DAU) 384
- DB_Comment (DCO) 384
- DB_Communication_Protocol 384

오브젝트 클래스 속성 (계속)	유형 맵핑, 작성 172	인증 267 (계속)
DB_Database_Locator_Name 386	응용프로그램용 LDAP 사용자 구성 476	정의 267
DB_Database_Protocol 386	이름 지정 규칙	파티션된 데이터베이스 고려사항 274
DB_Native_Database_Name 386	일반 369	DCE 디렉토리 서비스 389
DB_Object_Type 386	Windows NT 제한사항 434	DCE 보안 서비스 274
DB_Principal (DPR) 384	인스턴스	인증 유형 267
DB_Product_Name 387	개요 64	CLIENT 268
DB_Product_Release 387	갱신 210	DCE 271
DB_Target_Database_Info 387	기본값 74	DCE_SERVER_ENCRYPT 271
온라인 도움말 536	나열 82, 210	DCS 270
온라인 재구성	다중 수행 83	DCS_ENCRYPT 271
색인 196	단점 74	KERBEROS 272
온라인 정보	데이터베이스 파티션 서버 나열 447	KRB_SERVER_ENCRYPT 272
검색 542	디렉토리 74	SERVER 267
보기 538	변경 209	SERVER_ENCRYPT 268
외부 키	사용 이유 73	임시 테이블
로드 유틸리티, 참조 무결성 포함	소유자 77	사용자 정의 154
152	시작 62	사용자 정의 제거 247
복합 150	자동-시작 83	임시 테이블 공간 시스템 136
삭제에 필요한 권리 233	작성 74	입력된 뷰 작성 176
외부 키 정의에 대한 규칙 150	정의 73	입력된 뷰, 작성 176
제한조건 이름 150	제거 213	입력된 테이블
추가 230	증지 72	계층 구조 테이블 160
DROP FOREIGN KEY절, ALTER	추가 78	내용 수록 160
TABLE문 233	파티션 서버, 변경 450	작성 160
IMPORT 유틸리티, 참조 무결성 포함	파티션 서버, 제거 451	행 갱신 244
152	파티션 서버, 추가 448	행 삭제 244
요약 테이블	표시 18	입력된 테이블 갱신 244
데이터 새로 고침 243	현재 설정 82	입력된 테이블 작성 160
등록 정보 변경 242	인스턴스 레벨 프로파일 레지스트리 85	입력된 테이블에 데이터 처리 160
자동 177	인스턴스 목록 갱신 113	입력된 테이블의 행 삭제 244
자동 불능 요약 테이블 복구 252	인스턴스 사용자	
작성 176	환경 설정 75	
제거 252	인스턴스 소유자 77	
요약 테이블 등록 정보 변경 242	인스턴스 프로파일 레지스트리 85	
요약 테이블 제거 252	인증 267	
요약 테이블의 데이터 새로 고침 243	그룹 436	
원격 관리 102	도메인 보안 436	
원격 시스템 45	연합 데이터베이스 처리 282	
유형 맵핑 작성 172	원격 클라이언트 274	

[자]

자동 요약 테이블 177
자동 불능 뷰 복구 251
자동 불능 요약 테이블 복구 252
저널 열기 26
저장영역
관리 51
전역 디렉토리 서비스(GDS) 377

전역 레벨 프로파일 레지스트리 85
전용 상호연결 457, 459
점검 강제 규정 152
접두부 순서 420
정보 센터 539
정적 SQL
 데이터베이스 액세스에 대한
 EXECUTE 특권 311

제어 센터
 사용자 정의 15
 시스템 표시 17
제한사항
 Windows NT 이름지정 434
제한조건
 고유 정의 147
 변경 229
 제거 233
 추가 229
제한조건 변경 229
제한조건 이름
 외부 키 정의 151
 테이블 점검 제한조건의 정의 152
제한조건 추가 229, 233
조회 재작성
 요약 테이블 176
지역 데이터베이스 디렉토리
 개요 126

[차]

참조 제한조건
 정의 149
 테이블에 추가 230
FOREIGN KEY절,
 CREATE/ALTER TABLE문 149
PRIMARY KEY절,
 CREATE/ALTER TABLE문 149
REFERENCES절, CREATE/ALTER
 TABLE문 149
참조 제한조건 정의 148

찾기 오브젝트 22
책 523, 534
총계 함수 166
최신 정보 532
추적
 감사 325

[카]

카탈로그 노트
 설명 69
컨테이너
 수정(DMS 테이블 공간에) 219
 추가(DMS 테이블 공간에) 218
 SMS 테이블 공간에 추가 221
컬럼
 변경 227
 정의 143
 추가 226
컬럼 변경 227
컬럼 생성 153, 235
컬럼 수정 227
컬럼 옵션
 수치 문자열 255
 varchar_no_trailing_blanks 256
컬럼 UDF 166
콜 레벨 인터페이스(CLI)
 데이터베이스에 바인딩 131

크기
 계산 중 51
클라이언트

 관리 4
 신뢰 받지 못한 268
 신뢰받은 268, 269
 신뢰성 없는 269

[타]

테이블
 고유 제한조건 정의 147

테이블 (계속)
 노드 그룹 할당 129
 변경 225
 새로 이름 지정 244
 소멸성 239
 속성 변경 240
 식별 컬럼 155
 액세스를 갖는 이름 검색 321
 이름지정 142
 임시 123
 점검 강제 규정 152
 제거 245
 참조 제한조건 추가 230
 참조 제한조건의 정의 149
 컬럼 생성 153, 235
 특권 권한 취소 308
 파티션 키 변경 239
 파티션된 데이터베이스에서 작성 161
ALTER TABLE문 227
CREATE TABLE문 142
테이블 공간
 노드 그룹에서 137
 데이터 유형의 분리, 예 161
 데이터베이스 작성시 기본값 123
 변경 218
 사용 가능한 공간 검사(DMS) 52
 사용자 임시 136
 사용자 임시 제거 225
 새로 이름 지정 221
 시스템 임시 제거 223
 용량 추가 53
 임시 시스템 136
 작성 133
 장치 컨테이너 예 134
 제거 222
 추가 컨테이너 218
 컨테이너 크기재조정 219
 컨테이너 확장 219
 특권 300
 파일 시스템 컨테이너 예 133

- 테이블 공간 (계속)
 - 파일 컨테이너 예 133
 - ONLINE 상태에 대한 설정 222
- 테이블 공간 변경 218
- 테이블 공간 이름 바꾸기 221
- 테이블 공간 작성 133
- 테이블 공간 작성 마법사 6, 541
- 테이블 변경 225
- 테이블 속성 변경 240
- 테이블 수정 225
- 테이블 이름 변경 244
- 테이블 작성 142
- 테이블 작성 마법사 5, 541
- 테이블 점검 제한조건 232
 - 삭제 234
 - 정의 152
 - 추가 232
- 테이블 점검 제한조건 제거 234
- 테이블 제거 245
- 테이블 UDF 166
- 통신
 - 고속 455
- 통신 프로토콜
 - VI 아키텍처 457
- 트리거
 - 이점 164
 - 작성 163
 - 제거 248
 - 종속성 165
- 트리거 작성 163
- 트리거 제거 248
- 특권
 - 간접 특권, 별칭 312
 - 개별 290
 - 계층 289
 - 권한 부여 47
 - 권한 부여 이름 검색 321
 - 권한 취소 47
 - 권한 취소문 308
 - 권한의 권한 부여 및 취소 296

- 특권 (계속)
 - 데이터베이스 관리 프로그램 296
 - 별칭 303
 - 별칭을 가진 뷰 313
 - 뷰 300
 - 서버 304
 - 소유권(CONTROL) 290
 - 스키마 298
 - 시스템 카탈로그 목록 320
 - 요약 289
 - 이름에 대한 검색 322
 - 정보에 대해 뷰 작성 324
 - 정의 288
 - 타스크 및 필수 권한 부여 318
 - 테이블 300
 - 테이블 공간 300
 - 패키지 304
 - 패키지에 대한 내포 290
 - ALTER 301
 - BINDADD 296
 - CONNECT 296
 - CONTROL 300
 - CREATETAB 296
 - CREATE_NOT_FENCED 296
 - DELETE 301
 - IMPLICIT_SCHEMA 296
 - INDEX 305
 - INSERT 301
 - PUBLIC 297
 - REFERENCES 301
 - SELECT 301
 - USAGE 306
 - *GRANT문 307
- 특권과 권한 부여 47
- 특권과 권한 취소 47

[과]

- 파티션 내 병렬 처리
 - 사용 가능 66

- 파티션 키
 - 변경 239
 - 테이블 고려사항 161
 - 파티션 키에 파티션이 설정된 색인 196
- 파티션 키 변경 239
- 패키지
 - 사용 불가능 258
 - 소유자 311
 - 외부 키를 추가한 후에는 유효하지 않음 230
 - 제거 257
 - 종속성 257
 - 특권 304
 - 특권 권한 취소 309
 - SQL로 특권 액세스 311
- 표현식
 - NEXTVAL 158
 - PREVVAL 158
- 푸시다운 서버 조작 189
- 프로파일 레지스트리 84
- 필터 11

[하]

- 함수
 - 암호 해독(DECRYPT) 317
 - 암호화(ENCRYPT) 317
 - GETHINT 317
- 함수 맵핑 작성 167
- 함수 맵핑, 작성 167
- 함수 템플릿 작성 168
- 함수 템플릿, 작성 168
- 함수 호출
 - 선택 207
- 행
 - 제거 228
- 환경 변수 84
 - 변경 214
 - OS/2에서 설정 88
 - rah 422

환경 변수 84 (계속)
 RAHDOTFILES 423
 UNIX에서 설정 92
 Windows 95에서 설정 89
 Windows NT에서 설정 89
 환경 변수 변경 214
 활동 감사 325
 활성 디렉토리 471
 구성 473
 디렉토리 스키마 확장 491
 보안 489
 오브젝트 493
 지원 473
 히소 파일 할당 146

[숫자]

1차 색인
 고유 키에 대한 고유성 148
 삭제 257
 1차가 아닌 색인
 응용프로그램에 대한 내포 제거 257
 제거 257
 2바이트 문자 세트 사용자
 데이터 유형 143

A

ALTER COLUMN 227
 ALTER NICKNAME문, 예 255
 ALTER SERVER문, 예 254
 ALTER TABLESPACE문
 예 219
 ALTER TABLE문
 고유 제한조건 삭제의 예 233
 고유 제한조건 추가의 예 230
 점검 제한조건 삭제의 예 235
 점검 제한조건 추가의 예 232
 제한조건 추가에 대한 도움말 230
 컬럼 추가의 예 227
 키 삭제의 예 234

ALTER TABLE문 (계속)
 키 추가의 예 231
 ALTER VIEW문, 예 250
 ALTER 특권
 정의 301
 ATTACH 명령
 개요 65
 DCE(Distributed Computing
 Environment) 정보 지정 393
 audit_buf_sz 328

B

BIND 명령
 OWNER 옵션 311
 BIND 특권
 정의 305
 BINDADD 특권
 정의 296

C

CATALOG GLOBAL DATABASE
 DCE(Distributed Computing
 Environment) 정보 지정 393
 CDS 377
 CLIENT 레벨 보안 268
 CLIENT, 인증 유형 268
 collating_sequence 서버 옵션 186
 comm_rate 서버 옵션 186
 CONNECT 특권
 정의 296
 connectstring 서버 옵션 186
 CONNECT문
 DCE(Distributed Computing
 Environment) 정보 지정 393
 CONTROL 특권
 내재적인 발행(issuance) 310
 정의 300
 패키지 특권 305
 cooked 장치 135

cpu_ratio 서버 옵션 187
 CREATE ALIAS문
 사용 179
 예 180
 CREATE DATABASE 명령
 예 122
 CREATE INDEX문
 고유 색인 198
 예 198
 온라인 재구성 196, 200
 CREATE NICKNAME 190
 CREATE SERVER 182
 CREATE TABLESPACE문
 예 133
 CREATE TABLE문
 다중 테이블 공간의 사용 161
 예 144
 점검 제한조건 정의 152
 참조 제한조건 정의 149
 CREATE TRIGGER문
 예 164
 CREATE VIEW문
 예 174
 컬럼 이름의 변경 175
 CREATE WRAPPER 181
 CREATETAB 특권
 정의 296
 CREATE_NOT_FENCED 특권
 정의 296
 CURRENT SCHEMA 141
 CURRENT SCHEMA 특수 레지스터
 66
 D
 DAS 구성 갱신 113
 DataPropagator Relational(DPROPR)
 개요 361
 DAU (DB_Authentication) 384
 DB2
 Windows NT에서 시작 63

DB2 Connect	361	DB2 라이브러리 (계속)		DB_Authentication (DAU)	384
DB2 관리 서버		온라인 정보 검색	542	DB_Comment (DCO)	384
구성 갱신	113	온라인 정보 보기	538	DB_Communication_Protocol (DCP)	384
인스턴스 목록 갱신	113	인쇄된 책 주문	534	DB_Database_Locator_Name (DLN)	386
클라이언트 구성 지원 및 제어 센터 사용	112	정보 센터	539	DB_Database_Protocol (DDP)	386
DB2 관리 서버(DAS)	102	책	523	DB_Native_Database_Name (DNN)	386
개요	94	책의 언어 식별자	532	DB_Object_Type (DOT)	386
갱신	100	최신 정보	532	DB_Principal (DPR)	384
구성	99, 104	PDF 책 인쇄	533	DB_Product_Name (DPN)	387
나열	98	DB2 시작	62	DB_Product_Release (DRL)	387
레지스트리 변수	107	DB2 중지	72	DB_Target_Database_Info (DTI)	387
레지스트리 변수 고려사항	107	db2audit	330	DCE	
발견 작동 가능	108	db2audit.log	325	구성 매개변수 및 레지스트리 변수	391
보안	107	db2dmnbckctr		디렉토리 검색 방법	399
보안 고려사항	99	사용	434	디렉토리 서비스 사용	403
서비스 포트	105	db2gncol 유틸리티	236	디렉토리 서비스 제한사항	406
소유권 규칙	93	db2icrt 명령	79	디렉토리 서비스 타스크	402
시작과 중지	97	db2idrop	213	디렉토리 서비스의 개요	128
인터노드 관리 통신	105	db2ilist	210	제한사항	280
작성	95	DB2INSTANCE 환경 변수		ATTACH 명령	393, 399
제거	101	기본 인스턴스 정의	64	CATALOG GLOBAL DATABASE	393
제어 센터 통신	105	db2iupdt	210	CDS	377
통신	105	DB2LDAP_CLIENT_PROVIDER	473	CONNECT문	393, 400
파티션된 데이터베이스 시스템 설정	102	db2ldcfg 유틸리티	476	DB2 클라이언트 인스턴스 설정	279
예	102	db2lchg	450	DCE 디렉토리 정보를 임시로 겹쳐쓰기	402
파티션된 데이터베이스 시스템 (UNIX)의 인터노드 관리 통신	105	db2ncrt	448	GDS	377
파티션된 데이터베이스 시스템 (Windows NT)의 인터노드 관리 통신	107	db2ndrop	451	DCE 네트워크 데이터베이스 연결	396, 398
환경	107	db2nlist	447	작성	395
UNIX EEE 서버	105	db2nodes.cfg 파일	114	DCE(Distributed Computing Environment)	
Windows NT EEE 서버	107	db2perfc	445	DB2 사용자 설정	275
DB2 라이브러리		db2perfi	441	DCE, 인증 유형	271
구조	523	db2perfr	442		
마법사	540	db2set 명령	84, 85		
문서 서버 설정	541	db2start 명령	62		
온라인 도움말	536	db2stop 명령	72		
		db2_all	411, 412, 414		
		db2_call_stack	412		
		DB2_INDEX_2BYTEVARLEN	194		
		db2_kill	412		
		dbname 서버 옵션	187		

DCE_SERVER_ENCRYPT, 인증 유형 271

DCO (DB_Comment) 384

DCP (DB_Communication_Protocol) 384

DCS
 연합 데이터베이스 처리 283
 인증 유형 270

DCS_ENCRYPT, 인증 유형 271

DDL(Data Definition Language)
 생성 9

DDP (DB_Database_Protocol) 386

DECLARE GLOBAL TEMPORARY TABLE 154

DELETE 특권
 정의 301

DETACH 명령
 개요 65

distributed computing environment (DCE)
 보안 서비스 274
 인증 274
 DB2 서버 설정 278

DLN (DB_Database_Locator_Name) 386

DMS 테이블 공간
 작성 134

DNN (DB_Native_Database_Name) 386

DOT (DB_Object_Type) 386

DPN (DB_Product_Name) 387

DPR (DB_Principal) 384

DPROPR 361

DRL (DB_Product_Release) 387

DROP DATABASE 명령
 예 217

DROP INDEX문, 예 257

DROP NICKNAME문, 예 255

DROP SERVER문, 예 254

DROP TABLESPACE문, 예 223

DROP TABLE문
 예 245

DROP VIEW문, 예 251

DTI (DB_Target_Database_Info) 387

E

EXECUTE 특권
 동적 SQL로 데이터베이스 액세스 311
 정의 305
 정적 SQL로 데이터베이스 액세스 311

F

FCM 통신 118

fold_id 서버 옵션 187

fold_pw 서버 옵션 187

FOREIGN KEY절
 외부 키 정의에 대한 규칙 150
 참조 제한조건 150

G

GDS 377

GRANT
 예 307

GUI 도구
 administrating 사용 3

H

HTML
 샘플 프로그램 531

I

IBM eNetwork Directory
 오브젝트 클래스 및 속성 494

IBM eNetwork 디렉토리
 디렉토리 스키마 확장 491

IBMCATGROUP 노드 그룹 123

IBMDEFAULTGROUP 노드 그룹 123

IBMTEMPGROUP 노드 그룹 123

ID 및 암호를 데이터 소스로 전달 283

IDENTITY 컬럼 159

IDENTITY 컬럼의 변경 229

IMPLICIT_SCHEMA 권한 140

IMPLICIT_SCHEMA 특권
 정의 297

IMPORT 유틸리티
 데이터베이스에 바인딩 131
 참조 무결성 포함 152
 LOAD 152

INDEX 특권
 정의 301

INSERT 특권
 정의 301

io_ratio 서버 옵션 188

J

java 애플릿 56

Java 애플릿으로서의 제어 센터 57

K

Kerberos 보안 프로토콜 272

KNRBNROS 인증 유형 272

KRB_SERVER_ENCRYPT, 인증 유형 272

L

LDAP 56, 129, 471

LDAP 구성 474
 지원 472

LDAP 사용자 작성 475

LOAD 권한 295

LOAD 유틸리티
 개요 361

M

MINPCTUSED절 200

N

Netscape 브라우저
설치 538
NEXTVAL 158

O

OS/390용 DB2
서브시스템, 추가 18
오브젝트, 관리 18

P

PDF 533
PDF 책 인쇄 533
plan_hints 서버 옵션 188
PRECOMPILE 명령
OWNER 옵션 311
PREVVAL 158
PRIMARY KEY절
고유 키 추가 230
제한사항 148
PUBLIC
특권 297

R

RACF(resource access control
facility) 391
rah 412
기본 환경 프로파일 설정 424
머신 목록 지정 420
모니터링 프로세스 416
병렬 명령 수행 415
소개 411
제어 422
환경 변수 422
RAHDOTFILES 423
RAHOSTFILE 421
RAHOSTLIST 421

rah 412 (계속)
RAHWAITTIME 416
rah 명령 제어 422
RAHCHECKBUF 416
RAHTREETHRESH 417
raw 로그 137
raw 입출력 137
raw 장치 135
REFERENCES 특권
정의 301
REFERENCES절
사용 151
삭제 규칙 151
외부 키 추가 230
참조 제한조건 151
REORG 유틸리티
데이터베이스에 바인딩 131

S

SELECT 특권
정의 301
SELECT문
뷰를 선택 175
SERVER, 인증 유형 267
SERVER_ENCRYPT
인증 유형 268
SIGTTIN 414
SmartGuides
마법사 540
SMS 테이블 공간
작성 133
컨테이너 추가 221
SQL문
사용 불가능 258
표시 8
SQL문 표시 8
stdin 414
SWITCH ONLINE절 222
synonym(MVS/ESA용 DB2) 180

SYSCAT 뷰 320
SYSCATSPACE 테이블 공간 123

T

TCP/IP 456
TEMPSPACE1 테이블 공간 123

U

UPDATE 특권
정의 301
USAGE 특권 306
USERSPACE1 테이블 공간 123

V

VARCHAR 설정 227
varchar_no_trailing_blanks 서버 옵션
189
varchar_no_trailing_blanks 컬럼 옵션
256

VI

DB2 사용을 위한 설정 468
VI 아키텍처 457
VTAM(virtual telecommunications
access method) 391

W

Windows 2000 active directory
디렉토리 스키마 확장 491
오브젝트 493
Windows NT active directory
오브젝트 클래스 및 속성 494
Windows NT 성능 모니터 441
DB2 등록 441
Windows NT용 DB2 성능 카운터 441

[특수 문자]

\$RAHBUFDIR 415
\$RAHBUFSIZE 415

\$RAHCHECKBUF 415

\$RAHENV 423

*GRANT문

내재적인 발행(issuance) 310

보안 391

사용 307

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 누르십시오.

주: 일부 국가에서는 IBM 인증 공인 딜러는 IBM 지원 센터 대신 해당 딜러 지원 부서에 연락하십시오.



SA30-0988-01

