

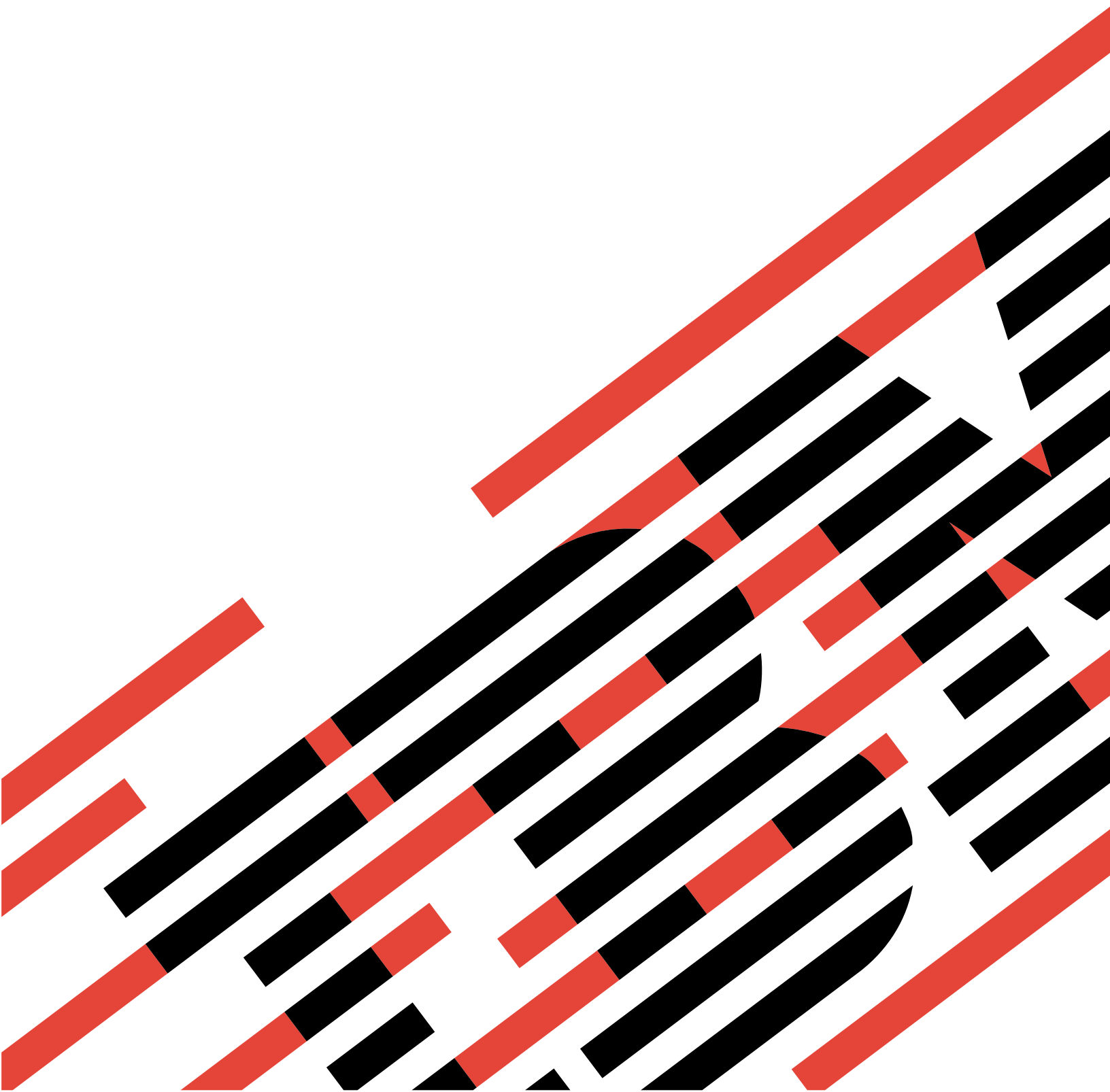
IBM

@server

iSeries

DB2 Multisystem

버전 5





@server

iSeries

DB2 Multisystem

버전 5

목차

DB2 Multisystem 정보 v
 이 책의 사용자 v
 코드 면책사항 관련 정보 v

제 1 장 DB2 Multisystem 소개 1
 DB2 Multisystem을 사용할 경우의 이점 2
 DB2 Multisystem: 기본 용어 및 개념 2

제 2 장 DB2 Multisystem에서 노드 그룹 소개 5
 DB2 Multisystem에서 노드 그룹 작업 방법 5
 DB2 Multisystem에서 노드 그룹 명령 사용 이전에
 완료할 TASK 6
 DB2 Multisystem에서 CRTNODGRP 명령을 사용하
 여 노드 그룹 작성 7
 DB2 Multisystem에서 DSPNODGRP 명령을 사용하
 여 노드 그룹 표시 9
 DB2 Multisystem에서 CHGNODGRPA 명령을 사
 용하여 노드 그룹 변경 10
 DB2 Multisystem에서 DLTNODGRP 명령을 사용
 하여 노드 그룹 삭제 12

제 3 장 DB2 Multisystem에서 분산 파일 작성 13
 DB2 Multisystem에서 분산 실제 파일 작성 13
 DB2 Multisystem에서 분산 파일 작성 또는 작업
 시 제한사항 15
 DB2 Multisystem에서 분산 파일 사용 16
 DB2 Multisystem에서 분산 파일에 대한 CL 명
 령 발행 18
 CL 명령: DB2 Multisystem에서 분산 파일에
 대해 실행이 허용됨 18
 CL 명령: DB2 Multisystem에서 분산 파일의
 로컬 부분에만 영향을 미침 18
 CL 명령: DB2 Multisystem에서 분산 파일의
 모든 부분에 영향을 미침 20
 DB2 Multisystem에서 고려사항 저널링 21
 DB2 Multisystem에서 분산 파일에 대한
 CPYF(파일 복사) 명령 사용 22
 DB2 Multisystem에서의 분할 23
 DB2 Multisystem에서의 분할 계획 24
 DB2 Multisystem에서 분할 키 선택 25
 DB2 Multisystem에서 자료 분배의 사용자 정의 26

**제 4 장 DB2 Multisystem에서 사용할 수 있는 스
 칼라 함수** 27
 DB2 Multisystem에서 PARTITION 27
 DB2 Multisystem에서 PARTITION 예 27
 DB2 Multisystem에서 HASH 28
 DB2 Multisystem에서 HASH 예 29
 DB2 Multisystem에서 NODENAME 29
 DB2 Multisystem에서 NODENAME 예 29
 DB2 Multisystem에서 NODENUMBER 30
 DB2 Multisystem에서 NODENUMBER 예 30
 DB2 Multisystem에서 특수 레지스터 31
 DB2 Multisystem에서 RRN(상대 레코드 번호)부
 여 함수 31

제 5 장 DB2 Multisystem에서 성능 및 확장성 33
 DB2 Multisystem을 사용해야 하는 이유 33
 DB2 Multisystem에서 성능 향상을 위한 추가 정
 보 35
 데이터베이스 확장을 지원하는 DB2 Multisystem 35
 시스템을 네트워크에 추가하기 위한 재분산 문제 36

**제 6 장 DB2® Multisystem에서 성능 향상을 위한
 조희 설계** 39
 DB2 Multisystem에서 최적화 개요 40
 DB2 Multisystem에서 단일 파일 조희 구현 및 최적
 화 40
 DB2 Multisystem에서 레코드 배열 구현 및 최적화 42
 DB2 Multisystem에서 UNION 및 DISTINCT 절
 구현과 최적화 43
 DB2 Multisystem에서 DSTDTA 및
 ALWCOPYDTA 매개변수 처리 43
 DB2 Multisystem에서 결합 구현 및 최적화 44
 DB2 Multisystem에서 상호 위치된 결합 44
 DB2 Multisystem에서 경로가 지시된 결합 45
 DB2 Multisystem에서 재분할된 결합 46
 DB2 Multisystem에서 브로드캐스트 결합 48
 DB2 Multisystem에서 결합 최적화 48
 DB2 Multisystem에서 결합 필드에 대한 분할
 키 49
 DB2 Multisystem에서 그룹화 구현 및 최적화 49
 DB2 Multisystem에서의 1단계 그룹화 50
 DB2 Multisystem에서의 2단계 그룹화 50
 DB2 Multisystem에서 그룹화 및 결합 51

DB2 Multisystem에서 부속 조회 지원	52	DB2 Multisystem에서의 ASYNCJ(비동기 작업 사용) 매개변수	59
DB2 Multisystem에서의 액세스 계획	52	APYRMT(리모트 적용) 매개변수	60
DB2 Multisystem에서의 재사용가능한 열린 자료 경 로(ODP)	52	성능 고려사항 요약.	60
DB2 Multisystem에서의 임시 결과 출력기.	54	참고 문헌	63
임시 출력기 작업: DB2 Multisystem에서의 장점	55	색인.	65
임시 출력기 작업: DB2 Multisystem에서의 단점	55		
DB2 Multisystem에서의 임시 출력기 제어.	56		
DB2 Multisystem에서의 Optimizer 메시지	56		
DB2 Multisystem에서의 CHGQRYA(조회 속성 변 경) 명령 변경사항	59		

DB2 Multisystem 정보

이 책에서는 분산 관계형 데이터베이스(RDB) 파일, 노드 그룹 및 분할과 같은 DB2 Multisystem의 기본 개념에 대해 설명합니다. 이 책은 여러 iSeries 서버에서 구획화된 데이터베이스 파일을 작성하고 사용하는 데 필요한 정보를 제공합니다. 시스템 구성 방법, 파일 작성 방법, 그리고 파일이 어플리케이션에서 사용될 수 있는 방법에 대한 정보가 제공됩니다.

DB2 Multisystem 책을 읽어야 하는지 알려면 『이 책의 사용자』를 참조하십시오.

이 책의 사용자

이 책은 많은 양의 자료가 들어 있는 데이터베이스를 관리하는 시스템 관리자나 데이터베이스 관리자를 위한 것입니다. 이 책의 사용자는 데이터베이스를 작성하고 사용하는 방법을 잘 알고 있어야 하며 데이터베이스 관리와 시스템 관리에 익숙해야 합니다.

DB2 Multisystem을 시작하려면 1 페이지의 제 1 장 『DB2 Multisystem 소개』로 가십시오.

코드 면책사항 관련 정보

이 책에는 많은 프로그래밍 예가 나옵니다.

사용자들은 자신의 필요에 따라 본문에서 제공하는 프로그램 예와 유사한 기능을 생성할 수 있으며 IBM에서는 사용자들에게 그러한 모든 프로그래밍 코드 예를 사용할 수 있는 배타적 사용권을 부여합니다.

IBM이 제공하는 모든 샘플 코드는 예제용입니다. 이 예들은 모든 환경에서 완벽하게 테스트된 것이 아닙니다. 따라서 IBM은 이 프로그램의 신뢰성, 서비스 기능성 또는 기능을 보장하지 않습니다.

여기에 나오는 모든 프로그램들은 어떠한 보증 없이 "현상태대로" 제공됩니다. 비침해, 판매 기능성, 특정 목적에의 적합성에 대한 암시적 보증을 명시적으로 부인합니다.

제 1 장 DB2 Multisystem 소개

DB2 Multisystem은 거의 무제한적인 데이터베이스용 확장성 옵션을 제공하는 병렬 처리 기술입니다. DB2 Multisystem을 사용하면, 여러 iSeries 서버(최대 32대)를 함께 "공유 없음" 클러스터에 연결할 수 있습니다. (공유 없음은 쌍을 이룬 네트워크의 각 시스템이 자체의 고유한 기본 메모리와 디스크 기억장치를 소유하고 관리하는 것을 의미합니다.) 일단 시스템이 연결되면, 데이터베이스 파일을 연결된 각 시스템상의 기억장치 전체에 분산시킬 수 있습니다. 데이터베이스 파일에는 시스템 세트에서 분할된(분배된) 자료가 있을 수 있으며, 각 시스템은 그 파일 내의 모든 자료에 대한 액세스를 갖습니다. 아직은 사용자들에 대해, 이 파일이 사용자들의 시스템에 있는 로컬 파일처럼 동작합니다. 사용자들의 시각에서 볼 때, 데이터베이스가 단일 데이터베이스로 표시되므로, 사용자는 네트워크의 모든 시스템에서 병렬로 조회할 수 있고 파일 내의 자료에 대한 실시간 액세스를 갖습니다.

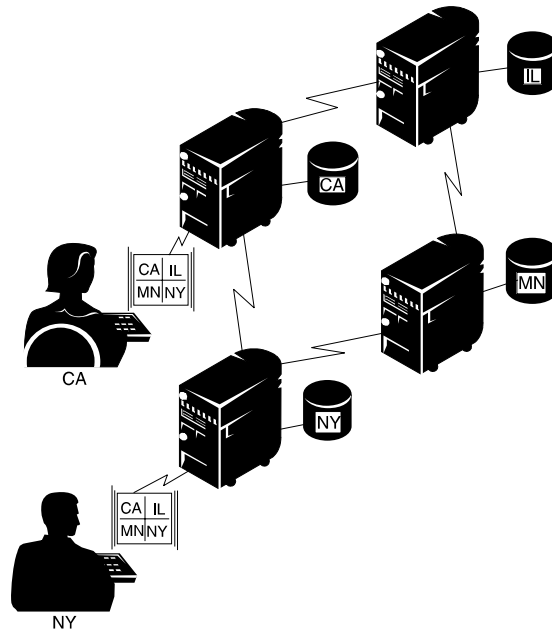


그림 1. 시스템에서의 데이터베이스 파일 분배

이 병렬 처리 기술은 하나의 서버에서 과중한 사용이 있어도 네트워크의 다른 연결된 시스템에서 성능이 떨어지지 않음을 의미합니다. 큰 자료 볼륨을 갖고 있고 조회를 실행해야 하는 경우, DB2 Multisystem은 사용가능한 가장 효율적인 방법 중 하나로 해당 조회를 실행하는 방법을 제공합니다. 대부분의 경우, 로컬 파일에 대해 더 이상 조회를 실행할 수 없지만 여러 서버에서 병렬로 실행되기 때문에 조회 성능이 향상됩니다.

아직 DB2 Multisystem을 설치하지 않은 경우, 추가 사용권 프로그램을 설치하기 위한 절차에 필요한 정보가 수록되어 있는 *Software Installation*을 참조하십시오. DB2 Multisystem을 설치하려면, OS/400 오퍼레이팅 시스템에 대한 설치가능한 옵션의 리스트에 있는 옵션 27을 사용하십시오.

자세한 안내 정보는 다음을 참조하십시오.

- 『DB2 Multisystem을 사용할 경우의 이점』
- 『DB2 Multisystem: 기본 용어 및 개념』

DB2 Multisystem을 사용할 경우의 이점

몇 가지 방법으로 DB2 Multisystem을 사용해 혜택을 받을 수 있습니다.

- 병렬로 실행하여 조회 성능을 향상시킬 수 있습니다. (각 조회 부분이 서로 다른 서버에서 동시에 실행됩니다.)
- 모든 서버가 모든 자료에 액세스할 수 있으므로 자료 복제에 대한 필요성이 거의 없습니다.
- 보다 큰 데이터베이스 파일을 수용할 수 있습니다.
- 어플리케이션은 더 이상 리모트 자료 위치에 관여하지 않습니다.
- 증가시켜야 할 경우, 더 많은 시스템에서 파일을 재분산할 수 있고 어플리케이션이 새로운 시스템에서 변경되지 않은 상태로 실행될 수 있습니다.

DB2 Multisystem으로, 동일한 입/출력(I/O) 메소드(GET, PUT 및 UPDATE)와, 과거에 사용했던 파일 액세스 방식을 사용할 수 있습니다. 추가적인 또는 다른 I/O 메소드나 파일 액세스 방식은 필요하지 않습니다.

어플리케이션은 사용자가 OptiConnect를 사용하지 않는 한, 현재 사용하는 연결 방법이 무엇이든지 변경하지 말아야 하고, 작성하는 분산 파일에 대해서도 작업할 것입니다. OptiConnect의 경우, OptiConnect 제어기 설명을 사용해야 합니다. OptiConnect에 대한 자세한 정보는 *OptiConnect for OS/400* 책을 참조하십시오.

DB2 Multisystem: 기본 용어 및 개념

분산 파일은 여러 iSeries 서버에 분산되어 있는 데이터베이스 파일입니다. 이 섹션에서는 DB2 Multisystem에 의한 분산 파일 작성 및 사용에 대한 설명에서 사용되는 기본 개념 중 일부에 대해 설명합니다. 각 용어와 개념은 다음 장들에서 보다 자세하게 설명되어 있습니다.

하나의 분산 파일을 갖는 각 서버 시스템을 노드라고 합니다. 각 서버는 관계형 데이터베이스(RDB) 디렉토리에서 정의된 이름에 의해 식별됩니다.

하나 이상의 분산 파일을 포함하게 될 시스템 그룹을 노드 그룹이라고 합니다. 노드 그룹은 자료가 분산될 노드 리스트를 포함하는 시스템 오브젝트입니다. 시스템은 여러 노드 그룹에 있는 노드가 될 수 있습니다.

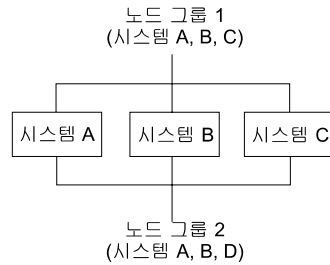


그림 2. 노드 그룹

파일은 분할을 통해 노드 그룹의 모든 시스템에 걸쳐 분배됩니다.

파티션 번호는 0 - 1023 사이의 숫자입니다. 각 파티션 번호는 노드 그룹의 노드에 할당됩니다. 각 노드에는 많은 파티션 번호가 할당될 수 있습니다. 노드와 파티션 번호 사이의 상호 관계는 파티션 맵에 저장됩니다. 파티션 맵은 노드 그룹 오브젝트의 부분으로도 저장됩니다. 노드 그룹을 작성할 때 파티션 맵을 제공할 수 있습니다. 제공하지 않으면, 시스템은 디폴트 맵을 생성하게 됩니다.

분할 파일을 사용하여 파티션 맵을 정의합니다. 분할 파일은 각 파티션 번호에 대해 노드 번호를 정의하는 실제 파일입니다.

분할 키는 분산되는 파일에 있는 하나 이상의 필드로 구성됩니다. 분할 키는 노드 그룹에서 특정 값을 갖는 행을 실제로 포함할 노드를 판별하기 위해 사용됩니다. 이것은 해싱(레코드에 대한 분할 키의 값을 취하여 이를 파티션 번호에 맵핑하는 오퍼레이팅 시스템 기능)을 사용하여 수행됩니다. 해당 파티션 번호에 상응하는 노드가 레코드 저장에 사용됩니다.

다음 예는 두 시스템에 대한 분산표와 유사하게 보일 수도 있는 파티션 번호와 노드를 나타내고 있습니다. 표의 분할 키는 LASTNAME입니다.

파티션 번호	노드
0	SYSA
1	SYSB
2	SYSA
3	SYSB
...	

그림 3. 파티션 맵. 파티션 번호 0에는 SYSA, 파티션 번호 1에는 노드 SYSB, 파티션 번호 2에는 SYSA, 파티션 번호 3에는 SYSB가 있습니다. 이러한 패턴이 반복됩니다.

분할 키의 해싱은 파티션 번호에 해당하는 숫자를 판별하게 됩니다. 예를 들어, 값이 'Andrews'인 레코드는 파티션 번호 1에 해싱될 수 있고, 값이 'Anderson'인 레코드는 파티션 번호 2에 해싱될 수 있습니다. 그림 3에 표시된 파티션 맵을 참조할 경우, 파티션 번호 1에 대한 레코드는 SYSB에 저장되는 반면, 파티션 번호 2에 대한 레코드는 SYSA에 저장됩니다.

제 2 장 DB2 Multisystem에서 노드 그룹 소개

데이터베이스 파일을 iSeries 서버 세트에서 볼 수 있도록 하려면, 먼저 파일을 작동시킬 시스템 그룹(노드 그룹)을 정의해야 합니다.

노드 그룹에는 2 - 32개까지의 노드가 정의될 수 있습니다. 노드 그룹에 대해 정의되는 노드 수는 데이터베이스 파일이 작성되는 시스템 수를 결정합니다. 로컬 시스템은 노드 그룹에서 지정된 시스템 중 하나이어야 합니다. 시스템이 노드 그룹을 작성할 때, 시스템은 각 노드에 대해 번호 1에서 시작하여 번호를 할당합니다.

노드 그룹에 대해 작업하려면 다음 주제를 참조하십시오.

- 『DB2 Multisystem에서 노드 그룹 작업 방법』
- 6 페이지의 『DB2 Multisystem에서 노드 그룹 명령 사용 이전에 완료할 TASK』
- 7 페이지의 『DB2 Multisystem에서 CRTNODGRP 명령을 사용하여 노드 그룹 작성』
- 9 페이지의 『DB2 Multisystem에서 DSPNODGRP 명령을 사용하여 노드 그룹 표시』
- 10 페이지의 『DB2 Multisystem에서 CHGNODGRPA 명령을 사용하여 노드 그룹 변경』
- 12 페이지의 『DB2 Multisystem에서 DLTNODGRP 명령을 사용하여 노드 그룹 삭제』

DB2 Multisystem에서 노드 그룹 작업 방법


노드 그룹은 작성된 시스템에 저장되는 시스템 오브젝트(*NODGRP)입니다. 이것은 분산 오브젝트가 아닙니다. *NODGRP 시스템 오브젝트에는 그룹 내의 시스템에 대한 모든 정보와 자료 파일의 자료가 분할(분배)되어야 하는 방법에 대한 정보도 포함되어 있습니다. 디폴트 분할은 각 시스템(노드)이 자료의 동등한 공유를 수신하게 합니다.

분할은 해시 알고리즘을 사용하여 처리됩니다. 노드 그룹이 작성될 때, 범위 0 - 1023 사이의 파티션 번호가 그 노드 그룹에 연관됩니다. 디폴트 분할을 사용하면, 동일한 수의 파티션들이 노드 그룹 내의 각 노드에 할당됩니다. 자료가 파일에 추가될 때, 분할 키의 자료가 해싱되고, 그 결과 파티션 번호가 생성됩니다. 자료의 전체 레코드는 해시 전용이 아니며, 분할 키의 자료는 해시 알고리즘을 통해 해시됩니다. 결과로 생성되는 파티션 번호와 연관되는 노드는 실제로 자료 레코드가 상주하게 되는 곳입니다. 그러므로, 디폴트 분할을 사용할 경우 자료 레코드가 충분하고 값 범위가 크면 각 노드가 동등한 자료 공유를 저장합니다.

각 노드가 동등한 자료 공유를 수신하는 것을 원하지 않거나, 자료의 특정 부분을 갖는 시스템을 제어하려는 경우, PTNFILE(파티션 파일) 매개변수를 사용하는 CRTNODGRP(노드 그룹 작성) 명령으로 사용자 분할 계획을 지정하거나, 추후에 CHGNODGRPA(노드 그룹 속성 변경) 명령을 사용하여 분할 계획을 변경함으로써 자료가 분할되는 방법을 변경할 수 있습니다. PTNFILE 매개변수를 사용하면 노드 그룹 내의 각각의 파티션에 대해 노드 번호를 설정할 수 있습니다. 다시 말하면, PTNFILE 매개변수를 사용하여 노드 그룹 내의 시스템에서 자료가 분할될 방법을 조정할 수 있습니다. (PTNFILE 매개변수는 7 페이지의 『DB2 Multisystem에서 CRTNODGRP 명령을 사용하여 노드 그룹 작성』의 예에서 사용됩니다.) 분할에 대한 자세한 정보는 23 페이지의 『DB2 Multisystem에서의 분할』을 참조하십시오.

노드 그룹은 시스템 오브젝트이므로, SAVOBJ(오브젝트 저장) 명령과 RSTOBJ(오브젝트 복원) 명령을 사용하여 저장되고 복원될 수 있습니다. 작성된 시스템이나 노드 그룹 내의 시스템 중 어느 하나로 노드 그룹 오브젝트를 복원할 수 있습니다. 노드 그룹 오브젝트가 노드 그룹에 없는 시스템으로 복원될 경우, 그 오브젝트는 사용할 수 없게 됩니다.

DB2 Multisystem에서 노드 그룹 명령 사용 이전에 완료할 TASK

CRTNODGRP(노드 그룹 작성) 명령이나 노드 그룹 명령을 사용하기 전에, 사용중인 분산 관계형 데이터베이스(RDB) 네트워크가 적절하게 설정되었는지 확인해야 합니다. 이것이 새로운 분산 관계형 데이터베이스 네트워크인 경우, 네트워크 구성 방법에 대해서는 분산 데이터베이스 프로그래밍  주제를 참조하십시오.

네트워크의 한 시스템이 로컬(*LOCAL) 시스템으로 정의되어 있는지 확인해야 합니다. WRKRDBDIRE(RDB(관계형 데이터베이스) 디렉토리 항목에 대한 작업) 명령을 사용하여 항목에 대한 세부사항을 표시하십시오. 로컬 시스템이 정의되어 있지 않으면, ADDRDBDIRE(RDB 디렉토리 항목 추가) 명령의 RMTLOCNAME(리모트 위치명) 매개변수에 대해 *LOCAL을 지정하여 확인할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADDRDBDIRE RDB(MP000) RMTLOCNAME(*LOCAL) TEXT ('New York')
```

MP000이라고 하는 New York의 iSeries 서버가 관계형 데이터베이스(RDB) 디렉토리에 로컬 시스템으로 정의되어 있습니다. 이 경우 네트워크 구성에 시스템명이나 로컬 위치명으로 단 하나의 로컬 관계형 데이터베이스만 iSeries 서버에 정의할 수 있습니다. 이렇게 하면 데이터베이스명을 식별하고 이를 분산 관계형 데이터베이스 네트워크의 특정 시스템에 상호 연관시키는 데 도움이 될 수 있습니다(특히, 네트워크가 복잡한 경우).

DB2 Multisystem의 경우, 사용자가 정의하는 노드 그룹 내에서 파일들을 iSeries 서버에 적절하게 분배하려면, 노드 그룹의 모든 노드(시스템)에서 일관적인 리모트 데이터베이스(RDB)명을 갖고 있어야 합니다.

예를 들어, 노드 그룹에 세 개의 시스템을 보유할 계획이면, 각 시스템의 RDB 디렉토리에 최소한 세 개의 항목이 있어야 합니다. 각 시스템에서, 세 개의 이름이 모두 같아야 합니다. 세 개의 시스템 각각에서, 항목은 *LOCAL로 식별되는 로컬 시스템에 대해 존재합니다. 다른 두 항목에는 적절한 리모트 위치 정보가 포함되어 있습니다.

DB2 Multisystem에서 CRTNODGRP 명령을 사용하여 노드 그룹 작성

이 섹션에서는 두 개의 CL 명령 예를 사용하여, CRTNODGRP(노드 그룹 작성) 명령으로 노드 그룹을 작성하는 방법을 나타내고 있습니다.

다음 예에서, 디폴트 분할(시스템 전체에 걸쳐 동등한 분할)을 갖는 노드 그룹이 작성됩니다.

```
CRTNODGRP NODGRP(LIB1/GROUP1) RDB(SYSTEMA SYSTEMB SYSTEMC SYSTEMD)
          TEXT('Node group for test files')
```

이 예에서, 명령은 네 개의 노드를 포함하는 노드 그룹을 작성합니다. 각 노드는 정의된 RDB 항목(이전에 ADDRDBDIRE 명령을 사용하여 관계형 데이터베이스 디렉토리에 추가된 항목)이어야 하며 하나의 노드가 로컬(*LOCAL)로 정의되어야 하는 점에 유의하십시오.

분할 속성은 디폴트로 각 노드 번호에 파티션들의 1/4을 할당합니다. 이 노드 그룹을 CRTPF(실제 파일 작성) 명령의 NODGRP 매개변수에 사용하여 분산 파일을 작성할 수 있습니다. 분산 파일에 대한 자세한 정보는 제 3 장 『DB2 Multisystem에서 분산 파일 작성』을 참조하십시오.

주: 자료의 1/4은 자료 내의 변화로 인해 각 시스템에서 반드시 분할되지 않아도 됩니다. 예를 들어, 분할 키로 지역 코드 필드를 선택하여 전화 지역 코드에 기초하여 자료를 분할할 것을 결정합니다. 이 상황이 발생된 바와 같이, 사용자 업무의 60%가 507 지역 코드로부터 제공됩니다. 507 지역 코드에 대한 파티션을 갖고 있는 시스템은 자료 중 25%가 아닌 60%를 수신하게 됩니다.

다음 예에서, 지정된 분할을 갖는 노드 그룹이 PTNFILE(분할 파일) 매개변수를 사용하여 작성됩니다.

```
CRTNODGRP NODGRP(LIB1/GROUP2) RDB(SYSTEMA SYSTEMB SYSTEMC)
          PTNFILE(LIB1/PTN1)
          TEXT('Partition most of the data to SYSTEMA')
```

이 예에서, 명령은 세 개의 노드(SYSTEMA, SYSTEMB, SYSTEMC)를 포함하는 노드 그룹을 작성합니다. 분할 속성은 PTN1 파일로부터 취합니다. 이 파일은 보다 높은 백분율의 레코드가 강제로 특정 시스템에 위치되도록 설정할 수 있습니다.

이 예에서 PTN1 파일이 분할 파일입니다. 이 파일은 분산 파일은 아니지만, 사용자 분할 계획을 설정하는 데 사용할 수 있는 정규 로컬 실제 파일입니다. 분할 파일에는 하

나의 2바이트 2진 필드가 있어야 합니다. 분할 파일에는 각 레코드가 유효한 노드 번호를 포함하는 1024개의 레코드들이 포함되어야 합니다.

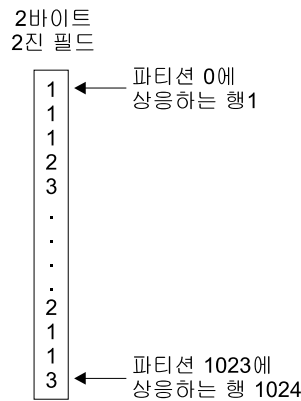


그림 4. 분할 파일 PTNFILE의 내용에 대한 예

노드 그룹에 세 개의 노드가 있을 경우, 분할 파일의 모든 레코드는 번호 1, 2 또는 3을 갖고 있어야 합니다. 노드 번호는 RDB명이 CRTNODGRP(노드 그룹 작성) 명령에 지정된 순서로 할당됩니다. 자료의 보다 높은 백분율은 분할 파일의 해당 노드 번호를 포함하는 더 많은 레코드를 갖도록 하여 특정 노드에 대해 강요할 수 있습니다. 이것은 실제로 각 시스템에 상주할 자료량에 관련하여 분할을 사용자 정의하는 방법입니다. 특정 노드에 상주하는 특정 값에 대해 분할을 사용자 정의하려면, CHGNODGRPA(노드 그룹 속성 변경) 명령을 사용하십시오. 자세한 정보는 10 페이지의 『DB2 Multisystem에서 CHGNODGRPA 명령을 사용하여 노드 그룹 변경』을 참조하십시오.

노드 그룹 정보는 분산 파일에 저장되므로, 파일은 노드 그룹에 포함되는 RDB 디렉토리 항목의 변경이나 노드 그룹의 변경에 대해 즉시 민감하게 대응하지는 않습니다. 노드 그룹과 EDB 디렉토리 항목을 수정할 수 있지만, CHGPF 명령을 사용하고 변경된 노드 그룹을 지정할 때까지 파일은 작동을 변경하지 않게 됩니다.


또 다른 개념은 비주얼 노드의 개념입니다. 노드 그룹 내의 비주얼 노드에 파일 오브젝트(파일이 여러 노드에 걸쳐 분산되도록 하는 메커니즘의 일부)가 포함되지만, 자료는 포함되지 않습니다. 비주얼 노드는 항상 파일 오브젝트의 현재 레벨을 보존합니다. 비주얼 노드에는 전혀 저장된 자료가 없습니다. 이와 반대로, 노드(간혹 자료 노드라고 함)에는 자료가 있습니다. 노드 그룹에서 비주얼 노드를 사용하는 방법의 예로서, 영업 총괄 매니저가 사용하는 iSeries 시스템을 사용자 노드 그룹의 일부로 만들어 보겠습니다. 총괄 매니저의 경우 정기적으로 조회를 실행하지는 않지만, 특정 조회를 실행해야 할 상황이 발생합니다. 이때에는 서버에서 조회를 실행하고 실시간 자료에 액세스하여 해당 조회의 결과를 수신할 수 있습니다. 따라서 자료를 서버에 저장시키지 않아도 시스템이 비주얼 노드이므로 총괄 매니저가 수시로 조회를 실행할 수 있습니다.

노드를 비주얼 노드가 되도록 지정하려면, CRTNODGRP(노드 그룹 작성) 명령에서 PTNFILE 매개변수를 사용해야 합니다. 분할 파일에 특정 노드 번호에 대한 어떤 레코드도 포함하지 않을 경우, 그 노드가 비주얼 노드입니다.

DB2 Multisystem에서 DSPNODGRP 명령을 사용하여 노드 그룹 표시

DSPNODGRP(노드 그룹 표시) 명령은 노드 그룹의 노드(시스템)를 표시합니다. 또한, 노드 그룹에 대한 분할 계획도 표시합니다. (분할에 대해서는 나중에 23 페이지의 『DB2 Multisystem에서의 분할』에서 설명합니다.)

다음 예는 GROUP1이라고 하는 노드 그룹과, 그 노드 그룹과 연관되는 분할 계획을 표시하는 방법을 나타내고 있습니다. 이 정보는 사용자의 워크스테이션에 표시됩니다.

DSPNODGRP 명령에 대한 자세한 정보는 제어 언어  주제를 참조하십시오.

```
DSPNODGRP NODGRP(LIB1/GROUP1)
```

노드 그룹명을 지정하고 DSPNODGRP 명령을 하면, 노드 그룹 표시 화면이 나타납니다. 이 화면은 시스템의 이름(관계형 데이터베이스 열에 나열됨)과 그 시스템에 할당되는 노드 번호를 나타내고 있습니다. 이것은 어떤 시스템에 어떤 노드 번호를 갖고 있는지를 판별하는 직접적인 방법입니다.

노드 그룹 표시	
노드 그룹: GROUP1	라이브러리: LIB1
관계형 데이터베이스	노드 번호
SYSTEMA	1
SYSTEMB	2
SYSTEMC	3
맨 아래 F3=나감 F11=분할 자료 F12=취소 F17=맨 위 F18=맨 아래	

그림 5. 노드 그룹 표시: 노드 그룹에서 데이터베이스 대 노드 번호의 관계 화면

각 파티션 번호에 할당된 노드 번호를 보려면, 노드 그룹 표시 화면에서 F11(분할 자료) 키를 사용하십시오. 다음 화면은 각 파티션 번호에 할당된 노드 번호를 나타내고

있습니다. 시스템과 노드 번호(또는 노드 번호 대 시스템) 사이의 맵핑은 DSPNODGRP 명령을 사용하여 쉽게 수행될 수 있습니다.

노드 그룹 표시	
노드 그룹:	GROUP1 라이브러리: LIB1
파티션 번호	노드 번호
0	1
1	2
2	3
3	1
4	2
5	3
6	1
7	2
8	3
9	1
10	2
11	3
12	1
13	2
14	3
계속...	
F3=나감	F11=노드 자료 F12=취소 F17=맨 위 F18=맨 아래

그림 6. 노드 그룹 표시: 파티션 번호 대 노드 번호 관계


다음 예는 GROUP2라고 하는 노드 그룹에 있는 시스템들의 리스트와 연관되는 분할 계획을 인쇄합니다.

```
DSPNODGRP NODGRP(LIB1/GROUP2) OUTPUT(*PRINT)
```

DB2 Multisystem에서 CHGNODGRPA 명령을 사용하여 노드 그룹 변경

CHGNODGRPA(노드 그룹 속성 변경) 명령은 노드 그룹에 대한 자료 분할 속성을 변경합니다. 노드 그룹에는 1024개의 파티션이 있는 표가 포함되어 있으며 각 파티션에는 노드 번호가 있습니다. 노드 번호는 노드 그룹이 작성될 때 할당되었으며 CRTNODGRP(노드 그룹 작성) 명령의 RDB 매개변수에 지정된 관계형 데이터베이스에 상응합니다. 유효한 노드 번호 값과, 노드 번호 및 관계형 데이터베이스명 사이의 상호 관계를 보려면 DSPNODGRP(노드 그룹 표시) 명령을 사용하십시오.

CHGNODGRPA 명령은 지정된 노드 그룹을 사용하여 작성된 기존의 분산 파일에는 영향을 주지 않습니다. 변경된 노드 그룹이 사용되도록 하려면, 새로운 파일을 작성할 때 또는 CHGPF(실제 파일 변경) 명령에서 변경된 노드 그룹을 지정해야 합니다.

CHGNODGRPA 명령에 대한 자세한 정보는 Information Center에서 제어 언어  주제를 참조하십시오.

다음 첫 번째 예는 LIB1 라이브러리에서 GROUP1이라고 하는 노드 그룹의 분할 속성을 변경하는 방법을 나타내고 있습니다.

```
CHGNODGRPA NODGRP(LIB1/GROUP1) PTNNBR(1019)
          NODNBR(2)
```

이 예에서, 분할 번호 1019가 지정되고, 1019로 해싱되는 레코드가 노드 번호 2에 기록됩니다. 이렇게 되면 노드 그룹 내의 특정 노드에 특정 파티션 번호를 지시하는 방법이 제공됩니다.

두 번째 예는 GROUP2라고 하는 노드 그룹의 분할 속성을 변경합니다. (GROUP2는 라이브러리 탐색 리스트 *LIBL을 사용하여 찾을 수 있습니다.) CMPDTA(비교 자료 값) 매개변수에 지정된 값이 해싱되고, 결과로 생성되는 파티션 번호가 기존 노드 번호에서 노드 번호 3으로 변경됩니다. (해싱과 분할에 대해서는 23 페이지의 『DB2 Multisystem에서의 분할』에 설명되어 있습니다.)

```
CHGNODGRPA NODGRP(GROUP2) CMPDTA('CHICAGO')
          NODNBR(3)
```


이 노드 그룹을 사용하여 작성되고 문자 필드로 구성되는 분할 키를 갖고 있는 파일은 노드 번호 3의 분할 키에 'CHICAGO'를 포함하는 레코드를 저장하게 됩니다. 분할 키에서 여러 필드가 있는 파일을 허용하려면, CMPDTA(자료 비교) 매개변수에 300까지의 값을 지정하면 됩니다.

CMPDTA 매개변수에 값을 입력할 때, 문자 자료는 대소문자가 구분되어야 합니다. 즉, 'Chicago'와 'CHICAGO'는 같은 파티션 번호를 생성하지 않습니다. 숫자 자료는 숫자 디지털로 간단하게 입력해야 합니다. 소수점, 선행 제로 또는 후미 제로는 사용하지 마십시오.

모든 값은 파티션 번호를 획득하기 위해 해싱되는데, 파티션 번호는 NODNBR(노드 번호) 매개변수에 지정되는 노드 번호와 연관됩니다. 완료 메시지 CPC3207의 텍스트는 변경된 파티션 번호를 나타내고 있습니다. CHGNODGRPA 명령을 많은 다른 값들에 대해 여러 번 내면 같은 파티션 번호를 두 번 변경할 수 있는 가능성이 증가됩니다. 이러한 상황이 발생하는 경우, 가장 최근의 변경에 지정된 노드 번호가 노드 그룹에 대해 유효합니다.

자료 분산의 사용자 정의에 관한 추가 정보는 26 페이지의 『DB2 Multisystem에서 자료 분배의 사용자 정의』를 참조하십시오.

DB2 Multisystem에서 DLTNODGRP 명령을 사용하여 노드 그룹 삭제

DLTNODGRP(노드 그룹 삭제) 명령은 이전에 작성된 노드 그룹을 삭제합니다. 이 명령은 노드 그룹을 사용하여 작성된 파일에는 영향을 주지 않습니다. DLTNODGRP 명령에 대한 자세한 정보는 Information Center에서 제어 언어  주제를 참조하십시오.

다음 예는 GROUP1이라는 노드 그룹을 삭제하는 방법을 나타내고 있습니다. 이 노드 그룹으로 작성된 파일에는 영향을 주지 않습니다.

```
DLTNODGRP NODGRP(LIB1/GROUP1)
```

노드 그룹의 삭제가 해당 노드 그룹을 사용하여 작성된 파일에는 영향을 주지 않더라도, 그 파일들을 사용한 후에 노드 그룹을 삭제하지 않는 것이 좋습니다. 일단 노드 그룹이 삭제되면, 노드와 분할 계획을 표시하기 위해 더 이상 DSPNODGRP 명령을 사용할 수 없습니다. 그러나, TYPE(*NODGRP)이 지정된 DSPFD(화면 파일 설명) 명령을 사용하여 특정 파일과 연관되는 노드 그룹을 볼 수 있습니다.

제 3 장 DB2 Multisystem에서 분산 파일 작성

분산 파일은 여러 iSeries 서버에 분산되어 있는 데이터베이스 파일입니다. 분산 파일은 갱신할 수 있으며, SQL, 조회 툴 및 DSPPFM(실제 파일 멤버 표시) 명령과 같은 방법을 통해 액세스할 수 있습니다. 데이터베이스 조작의 경우, 분산 파일은 대부분 로컬 파일로 취급됩니다. 분산 파일에 대한 CL 명령 변경사항에 대해서는 18 페이지의 『DB2 Multisystem에서 분산 파일에 대한 CL 명령 발행』을 참조하십시오.

분산 파일로 데이터베이스 파일을 작성하려면, CRTPF(실제 파일 작성) 명령이나 SQL CREATE TABLE문을 사용하면 됩니다. 이 장에는 이 두 방법에 대한 자세한 설명이 수록되어 있습니다. CRTPF 명령에는 두 개의 매개변수인 NODGRP(노드 그룹)과 PTNKEY(분할 키)가 있는데, 이 매개변수들은 파일을 분산 파일로 작성합니다. 분산 데이터베이스 파일은 CRTPF 명령을 실행할 때 지정한 이름과 라이브러리를 갖는 오브젝트(PF 속성이 *FILE)로 표시될 것입니다.

기존의 비분산 데이터베이스 실제 파일을 분산 파일로 변경하려면, CHGPF(실제 파일 변경) 명령을 사용하면 됩니다. CHGPF 명령에서, NODGRP 및 PTNKEY 매개변수를 사용하여 분산 파일을 변경할 수 있습니다. CHGPF 명령을 통해 NODGRP 및 PTNKEY 매개변수를 지정하여 기존의 분산 데이터베이스 파일에 대한 자료 분할 속성을 변경할 수도 있습니다. 이 매개변수에 값을 지정하면 노드 그룹의 분할표에 따라 자료가 재분산됩니다.

주: 분산 파일로 변경되는 실제 파일을 기초로 하는 모든 논리 파일도 분산 파일이 됩니다. 큰 파일이나 대규모 네트워크의 경우, 자료 재분산에 시간이 소요될 수 있어서 자주 수행하지 않아야 합니다.

파일에 대해 작업하려면 다음 주제를 참조하십시오.

- 『DB2 Multisystem에서 분산 실제 파일 작성』
- 16 페이지의 『DB2 Multisystem에서 분산 파일 사용』
- 23 페이지의 『DB2 Multisystem에서의 분할』
- 26 페이지의 『DB2 Multisystem에서 자료 분배의 사용자 정의』

DB2 Multisystem에서 분산 실제 파일 작성

이 섹션에서는 CRTPF 명령이나 SQL CREATE TABLE문을 사용하여 분산 실제 파일을 작성하는 방법에 대해 설명합니다.

분할된 파일을 작성하려는 경우, CRTPF(실제 파일 작성) 명령에서 다음 매개변수들을 지정해야 합니다.


- NODGRP 매개변수에 대한 노드 그룹명
- 분할 키로 사용될 필드(PTNKEY 매개변수 사용)

분할 키는 각 자료 레코드가 실제로 상주할 위치(어떤 노드 상에)를 판별합니다. CRTPF(실제 파일 작성) 명령이 실행되거나 SQL CREATE TABLE문이 실행될 때 분할 키를 지정합니다. 각 레코드에 대해 분할 키를 구성하는 필드들의 값은 레코드가 위치될 곳을 판별하기 위해 HASH 알고리즘에 의해 처리됩니다.

단일 필드 분할 키를 갖고 있는 경우, 해당 필드에서 같은 값을 갖는 모든 레코드들이 같은 시스템에 상주하게 됩니다.

분산 실제 파일을 작성하려면, 사용자 프로파일이 노드 그룹 내의 모든 노드에 존재하고 사용자 프로파일이 모든 노드에서 분산 파일을 작성하기 위해 필요한 권한을 가지고 있어야 합니다. 특정 라이브러리에서 분산 파일을 작성해야 하는 경우, 해당 라이브러리가 노드 그룹 내의 모든 노드에 존재하고 사용자 프로파일이 그 라이브러리에서 파일을 작성하기 위해 필요한 권한을 가지고 있어야 합니다. 이 요인들 중 어느 하나가 성립되지 않으면, 파일은 작성되지 않습니다.

시스템이 구성될 수 있는 방법은 리모트 시스템에서 사용되는 사용자 프로파일에 영향을 줄 수 있습니다. 사용자 프로파일이 리모트 시스템에서 사용되도록 하려면, 해당 시스템이 보안 위치로 구성되어야 합니다. 시스템이 보안 위치로 구성되어 있는지 판별하려면, WRKCFGL(구성 리스트에 대한 작업) 명령을 사용하십시오. 자세한 정보는

Information Center에서 분산 데이터베이스 프로그래밍  주제를 참조하십시오.

다음 예는 분할되고(NODGRP 매개변수를 사용하여 지정) 사원 번호(EMPNUM) 필드에 단일 분할 키를 갖고 있는 PAYROLL 실제 파일을 작성하는 방법을 나타내고 있습니다.

```
CRTPF FILE(PRODLIB/PAYROLL) SCRFILE(PRODLIB/DDS) SRCMBR(PAYROLL)
      NODGRP(PRODLIB/PRODGROUP) PTNKEY(EMPNUM)
```

CRTPF 명령이 실행될 때, 시스템은 분산 파일과 연관되는 로컬 자료를 보유하기 위해 분산 실제 파일을 작성합니다. CRTPF 명령은 또한 노드 그룹에 지정된 모든 리모트 시스템에 대해서도 실제 파일을 작성합니다.

실제 파일의 소유권과 모든 시스템에서의 공용 권한이 일치합니다. 이 일관성에는 CRTPF 명령의 AUT 매개변수에 지정된 권한도 포함됩니다.

SQL CREATE TABLE문을 사용하여 노드 그룹과 분할 키를 지정할 수도 있습니다. 다음 예에서, PAYROLL이라고 하는 SQL 표가 작성됩니다. 이 예에서는 IN *nodgroup-name*절과 PARTITIONING KEY절이 사용됩니다.

```

CREATE TABLE PRODLIB/PAYROLL
      (EMPNUM INT, EMPLNAME CHAR(12), EMPFNAME CHAR (12))
IN PRODLIB/PRODGROUP
PARTITIONING KEY (EMPNUM)

```

PARTITIONING KEY절을 지정하지 않으면, 1차 키(정의된 경우)의 첫 번째 열이 첫 번째 분할 키로 사용됩니다. 1차 키가 정의되어 있지 않으면, 자료 유형이 날짜, 시간, 시간소인 또는 부동 소수점 숫자가 없는 표에 대해 정의된 첫 번째 열이 분할 키로 사용됩니다.

파일이 분할되어 있는지 확인하려면, DSPFD(화면 파일 설명) 명령을 사용하십시오. 파일이 분할되어 있으면, DSPFD 명령은 노드 그룹의 이름을 보여주고, 파일 오브젝트에 저장된 노드 그룹의 세부사항을 보여주며(전체 파티션 맵을 포함하여), 분할 키의 필드들을 나열합니다.

DB2 Multisystem과 함께 분산 파일을 사용할 때 알아야 할 제한사항 리스트는 『DB2 Multisystem에서 분산 파일 작성 또는 작업시 제한사항』을 참조하십시오.

DB2 Multisystem에서 분산 파일 작성 또는 작업시 제한사항

분산 파일을 작성하거나 분산 파일에 대해 작업할 때 다음 제한사항에 유의해야 합니다.

- FCFO(먼저 변경된 것이 먼저 출력됨) 액세스 경로는 여러 노드에 걸쳐 분할되므로 사용할 수 없습니다.
- 분산 파일에는 최대 하나의 멤버가 있을 수 있습니다.
- 분산 파일은 임시 라이브러리(QTEMP)에서 허용되지 않습니다.
- 분할 키의 자료는 제한되는 갱신 기능을 갖습니다. 일반적으로, 분할 키를 선택할 때, 값이 갱신되지 않는 필드를 선택해야 합니다. 분할 키에 대한 갱신은 그러한 갱신으로 레코드가 다른 노드로 분할되지 않는다면 허용됩니다.
- 날짜, 시간, 시간소인 또는 부동 소수점 숫자 필드는 분할 키에서 사용할 수 없습니다.
- 소스 실제 파일은 지원되지 않습니다.
- 외부 서술 파일은 분산 파일에 대해 지원되고, 프로그램 서술 파일은 지원되지 않습니다.
- 액세스 경로가 고유하면, 분할 키는 고유 키 액세스 경로의 서브세트이어야 합니다.
- 제한조건이 지원되고, 상위 및 외부 키 파일 모두의 노드 그룹이 동일하며 분할 키의 모든 필드들이 그 제한조건에 포함될 경우에만 참조 제한조건이 지원됩니다. 분할 키는 제한조건 필드들의 서브세트이어야 합니다. 또한, 고유한 1차 제한조건에 대해 액세스 경로가 고유하면, 분할 키는 고유 키 액세스 경로의 서브세트이어야 합니다.

- CRTPF 명령에서, 시스템 매개변수에는 *LCL 값이 지정되어야 합니다(CRTPF SYSTEM(*LCL)). SYSTEM(*RMT)은 허용되지 않습니다.
- 논리 파일이 분산 파일을 거쳐 작성될 때, 논리 파일도 분산 파일이 됩니다. 즉, 특정 노드에서 실제 파일의 한 부분만을 거쳐 로컬 논리 파일을 빌드할 수 없음을 의미합니다. 뷰가 결합이며 기본이 되는 모든 실제 파일이 같은 노드 그룹을 갖고 있지 않을 경우, SQL 뷰는 이에 대한 예외가 됩니다. 이 경우, 뷰는 로컬 시스템에서만 작성됩니다. 이 뷰가 분산되지 않더라도 뷰를 조회할 경우, 뷰가 작성되었던 노드가 아닌 모든 노드로부터 자료가 검색됩니다.
결합 파일은 SQL로만 작성될 수 있습니다.
DDS 작성 논리 파일의 경우, 하나의 기초 파일만 허용됩니다.
- CCSID(코드화 문자 세트 ID)와 SRTSEQ(정렬 순서) 표는 원래의 시스템으로부터 분석됩니다.
- VLR(가변 길이 레코드) 처리는 지원되지 않습니다. 이것은 가변 길이 필드가 분산 파일에 대해 지원되지 않음을 의미하는 것은 아닙니다. 이 제한사항은 파일이 열릴 때 VLR 처리를 요구하는 언어와 어플리케이션에 대해서만 언급한 것입니다.
- EOFDLY(파일 끝 지연) 처리가 지원되지 않습니다.
- DFU(자료 파일 유틸리티)는 레코드에 액세스하기 위해 상대 레코드 번호(RRN)를 사용하므로 분산 파일에 대해 작업하지 않습니다.
- 분산 파일은 독립 보조 기억장치 풀(IASP)에 있는 라이브러리 안으로 작성할 수 없습니다.

DB2 Multisystem에서 분산 파일 사용

파일이 작성되고 나면, 자료가 분할되고 파일들이 동시 레벨에 남아 있는지 시스템이 확인합니다. 파일이 작성되면 자동으로 발생하는 활동은 다음과 같습니다.

- 파일에 대해 작성되는 모든 색인이 모든 노드에서 작성됩니다.
- 권한 변경사항이 모든 노드에 송신됩니다.
- 시스템은 파일이 이동되지 못하게 하고 해당 라이브러리가 재명명되지 못하게 합니다.
- 파일 자체가 재명명될 경우, 새 이름이 모든 노드에 반영됩니다.
- ALCOBJ(오브젝트 할당), RGZPFM(실제 파일 멤버 재구성) 및 STRJRNPF(저널 실제 파일 시작) 명령과 같은 일부 명령은 이제 파일의 모든 부분에 영향을 미칩니다. 이로써 분할된 파일에 대해 작업할 때 로컬 파일의 개념을 유지보수할 수 있습니다. 이러한 CL 명령의 전체 리스트를 보려면 20 페이지의 『CL 명령: DB2 Multisystem에서 분산 파일의 모든 부분에 영향을 미침』을 참조하십시오.

노드 그룹의 어느 노드에서든지 ALCOBJ(오브젝트 할당) 명령을 낼 수 있습니다. 이렇게 하면 모든 부분이 잠겨지고 로컬 파일을 할당할 때 동일한 무결성이 부여되도록 합니다. 이러한 모든 조치는 각 노드에서 명령을 입력하지 못하도록 하는 시스템에 의해 처리됩니다.

STRJRNPF(저널 실제 파일 시작) 명령의 경우, 저널은 각 시스템에서 시작됩니다. 그러므로, 각 시스템에는 고유한 저널 및 저널 리시버가 있어야 합니다. 각 시스템에는 고유한 저널 항목이 있습니다. 이 저널 항목을 사용하는 회복이 각 시스템에서 별도로 수행되어야 합니다. 저널을 시작하고 종료하기 위한 명령은 노드 그룹의 모든 시스템에 동시에 영향을 줍니다. 추가 정보에 대해서는 21 페이지의 『DB2 Multisystem에서 고려사항 저널링』을 참조하십시오.

- DMPOBJ(오브젝트 덤프), SAVOBJ(오브젝트 저장), WRKOBJLCK(오브젝트 잠금에 대한 작업)과 같은 일부 명령은 명령이 발행된 시스템의 파일 부분에만 영향을 미칩니다. 이러한 CL 명령의 전체 리스트를 보려면 18 페이지의 『CL 명령: DB2 Multisystem에서 분산 파일의 로컬 부분에만 영향을 미침』을 참조하십시오.

파일이 분산 파일로 작성되면, 실제로 파일을 열 경우 모든 리모트 시스템에 대한 연결과 파일의 로컬 부분이 열립니다. 파일이 작성되고 나면, 노드 그룹의 어느 시스템에서든지 그 파일에 액세스할 수 있습니다. 시스템은 또한 파일 I/O TASK(예: GETS, PUT 및 UPDATES)를 완료하기 위해 사용해야 하는 노드와 레코드를 판별합니다. 사용자가 실제로 영향을 주거나 이러한 활동을 지정하지 않아도 됩니다.

분산 관계형 데이터베이스 구조(DRDA[®])와 분산 자료 관리(DDM) 요구는 목표 분산 파일이 될 수 있다는 점에 유의하십시오. DRDA 또는 DDM을 사용하여 리모트 시스템의 데이터베이스 파일에 액세스하는 이전에 분산된 어플리케이션은 해당 데이터베이스 파일이 분산 파일이 되도록 변경된 경우에도 계속 작동할 수 있습니다.

레코드의 도달 순서는 로컬 데이터베이스 파일의 경우와 분산 데이터베이스 파일에 대해 서로 다르다는 사실에 유의해야 합니다.

분산 파일은 시스템에서 실제로 분산되므로, 레코드의 도달 순서나 상대 레코드 번호(RRN)에 의존할 수 없습니다. 로컬 데이터베이스 파일의 경우, 레코드들은 순서대로 처리됩니다. 예를 들어, 자료를 로컬 데이터베이스 파일에 삽입하고 있을 경우, 그 자료는 첫 번째 레코드에서 시작하여 마지막 레코드에 이르기까지 계속 삽입됩니다. 모든 레코드는 순서대로 삽입됩니다. 개인 노드의 레코드들은 로컬 파일에 대해 레코드가 삽입되는 것과 같은 방법으로 삽입됩니다.


자료가 로컬 데이터베이스 파일로부터 읽혀질 경우, 그 자료는 첫 번째 레코드에서 시작하여 마지막 레코드에 이르기까지 계속 읽혀집니다. 이것은 분산 데이터베이스 파일에 대해서는 적용되지 않습니다. 분산 데이터베이스 파일의 경우, 자료는 첫 번째 노드의 레코드들(첫 번째 레코드에서 마지막 레코드로), 두 번째 노드의 레코드들 순으로 읽

허집니다. 예를 들어, 레코드 27까지 읽고 더 이상 읽지 않을 경우 이것은 단일 레코드를 읽은 것을 의미합니다. 분산 파일의 경우, 노드 그룹의 각 노드에는 자체 레코드 27(동일하지 않은)이 포함될 수도 있습니다.

CL 명령을 발행하는 것에 관한 자세한 정보는 『DB2 Multisystem에서 분산 파일에 대한 CL 명령 발행』을 참조하십시오.

DB2 Multisystem에서 분산 파일에 대한 CL 명령 발행

분산 파일에 시스템 오브젝트 유형 *FILE이 있으므로, 실제 파일에 액세스하는 많은 CL 명령이 분산 파일에 대해 실행될 수 있습니다. 그러나, 일부 CL 명령의 작동은 분산 파일 대 비분산 파일에 대해 발행될 때 변경됩니다. 다음 섹션들에서는 CL 명령이 분산 파일에 대해 작동하는 방법을 설명합니다. CL 명령에 관한 모든 정보는

Information Center에서 제어 언어  주제를 참조하십시오.

DB2 Multisystem과 함께 사용할 수 있는 특정 유형의 CL 명령에 대해 알려면 다음 주제를 참조하십시오.

- 『CL 명령: DB2 Multisystem에서 분산 파일에 대해 실행이 허용됨』
- 『CL 명령: DB2 Multisystem에서 분산 파일의 로컬 부분에만 영향을 미침』
- 20 페이지의 『CL 명령: DB2 Multisystem에서 분산 파일의 모든 부분에 영향을 미침』

CL 명령: DB2 Multisystem에서 분산 파일에 대해 실행이 허용됨

다음 CL 명령 또는 특정 매개변수들은 분산 파일에 대해 실행할 수 없습니다.

- CHGLFM(논리 파일 멤버 변경)의 SHARE 매개변수
- CHGPFM(실제 파일 멤버 변경)의 SHARE 매개변수
- CRTDUPOBJ(중복 오브젝트 작성)
- INZPFM(실제 파일 멤버 초기화)
- MOV OBJ(오브젝트 이동)
- POSDBF(데이터베이스 파일 위치지정)
- RMV M(멤버 제거)
- 분산 파일을 포함하는 라이브러리에 대한 RNMLIB(라이브러리 재명명)
- RNMM(멤버 재명명)
- IFS(통합 파일 시스템) 명령, COPY

CL 명령: DB2 Multisystem에서 분산 파일의 로컬 부분에만 영향을 미침

다음 CL 명령들은 실행시 로컬 시스템(명령이 실행되었던 시스템)에 위치해 있는 분산 파일의 부분에만 영향을 줍니다.

- APYJRNCHG(저널된 변경사항 적용). 이 명령에 대한 추가 정보는 21 페이지의 『DB2 Multisystem에서 고려사항 저널링』을 참조하십시오.
- DSPOBJD(오브젝트 설명 표시)
- DMPOBJ(오브젝트 덤프)
- ENDJRNAP(저널 액세스 경로 종료)
- RMVJRNCHG(저널된 변경사항 제거). 이 명령에 대한 추가 정보는 21 페이지의 『DB2 Multisystem에서 고려사항 저널링』을 참조하십시오.
- RSTOBJ(오브젝트 복원)
- SAVOBJ(오브젝트 저장)
- STRJRNAP(저널 액세스 경로 시작)

SBMRMTCMD(리모트 명령 제출) 명령을 사용하여 분산 파일과 연관되는 모든 리모트 시스템에 대해 CL 명령을 낼 수 있습니다. 로컬 시스템에서 CL 명령을 발행한 후 같은 명령을 분산 파일에 대해 SBMRMTCMD 명령을 통해 발행할 경우, 분산 파일에 대한 모든 시스템에서 CL 명령을 실행할 수 있습니다. 분산 파일의 모든 부분에서 자동으로 실행되는 CL 명령에 대해 이를 수행할 필요가 없습니다. 자세한 정보는 20 페이지의 『CL 명령: DB2 Multisystem에서 분산 파일의 모든 부분에 영향을 미침』을 참조하십시오.

DSPFD(화면 파일 설명) 명령을 사용하여 분산 파일에 대한 노드 그룹 정보를 표시할 수 있습니다. DSPFD 명령은 노드 그룹의 이름, 분할 키의 필드들, 그리고 노드 그룹의 완전한 설명을 나타내고 있습니다. 이 정보를 표시하려면, DSPFD 명령의 TYPE 매개변수에 대해 *ALL 또는 *NODGRP를 지정해야 합니다.

DSPPFM(실제 파일 멤버 표시) 명령을 사용하여 분산 파일의 로컬 레코드를 표시할 수 있습니다. 그러나, 로컬 자료뿐만 아니라 리모트 자료를 표시하려면, 명령의 FROMRCD(from 레코드) 매개변수에 *ALLDATA를 지정해야 합니다.

분산 파일에 대해 SAVOBJ(오브젝트 저장) 명령이나 RSTOBJ(오브젝트 복원) 명령을 사용할 경우, 분산 파일의 각 부분들은 별도로 저장되고 복원되어야 합니다. 파일 부분은 분산 파일의 일부로 유지보수될 경우에만 저장되었던 시스템에 다시 복원될 수 있습니다. 필요할 경우, ALLOBJ(오브젝트 할당) 명령을 사용하여 저장 프로세스 도중에 파일에 대한 갱신을 막기 위해 파일의 모든 부분을 잠글 수 있습니다.

다음 조건이 만족될 경우, 파일이 복원될 때 시스템이 논리 파일을 분산시킵니다.

- 논리 파일이 비분산 파일로 저장되었습니다.
- 기초 파일이 분배될 때 논리 파일이 시스템에 복원됩니다.

파일의 저장된 부분도 로컬 파일을 작성하는 데 사용할 수 있습니다. 이를 수행하려면, 분산 파일이 작성될 때 사용된 노드 그룹에 없었던 시스템이나 다른 라이브러리로 파

일 부분을 복원해야 합니다. 분산 파일의 모든 레코드를 로컬 파일로 가져가려면, 파일의 각 부분을 같은 방법으로 복원한 후 레코드를 총계 파일로 복사해야 합니다. 레코드를 총계 파일로 복사하려면 CPYF(파일 복사) 명령을 사용하십시오.

CL 명령: DB2 Multisystem에서 분산 파일의 모든 부분에 영향을 미침

다음 CL 명령은 실행시 분산 파일의 모든 부분에 영향을 줍니다. 시스템에서 이러한 명령을 실행할 때, 명령은 노드 그룹 내의 모든 노드에서 자동으로 실행됩니다.

이 규약은 각 시스템에 같은 명령을 입력하지 않고 노드 그룹에서 일관성이 유지되도록 합니다. 권한 변경사항에 대해, 노드 그룹에서 일부 비일관성이 발생할 수 있습니다. 예를 들어, 사용자 ID가 노드 그룹의 한 시스템에서 삭제될 경우, 노드 그룹에서 일관성을 유지할 수 있는 능력이 손실됩니다.

권한 오류는 개별적으로 처리됩니다.

다음 명령들은 분산 파일의 모든 부분에 영향을 줍니다.

- ADDLFM(논리 파일 멤버 추가)
- ADDPFCST(실제 파일 제한조건 추가)
- ADDPFM(실제 파일 멤버 추가)
- ADDPFTRG(실제 파일 트리거 추가)
- ALCOBJ(오브젝트 할당)
- CHGLF(논리 파일 변경)
- CHGOBJOWN(오브젝트 소유자 변경)
- CHGPF(실제 파일 변경)
- CHGPFCST(실제 파일 제한조건 변경)
- CLRPFM(실제 파일 멤버 지우기)
- CPYF(파일 복사). 이 명령에 대한 추가 정보는 22 페이지의 『DB2 Multisystem에서 분산 파일에 대한 CPYF(파일 복사) 명령 사용』을 참조하십시오.
- CRTLF(논리 파일 작성)
- DLCOBJ(오브젝트 할당해제)
- DLTF(파일 삭제)
- ENDJRNP(저널 실제 파일 변경사항 종료). 이 명령에 대한 추가 정보는 21 페이지의 『DB2 Multisystem에서 고려사항 저널링』을 참조하십시오.
- GRTOBJAUT(오브젝트 권한 부여)
- RMVPFCST(실제 파일 제한조건 제거)
- RMVPFTRG(실제 파일 트리거 제거)
- RNMOBJ(오브젝트 재명명)
- RGZPFM(실제 파일 멤버 재구성)

- RVKOBJAUT(오브젝트 권한 취소)
- STRJRNPf(저널 실제 파일 시작). 이 명령에 대한 추가 정보는 『DB2 Multisystem에서 고려사항 저널링』을 참조하십시오.

이 명령들의 경우, 분산 파일 이외의 오브젝트가 참조될 때, 각 시스템에서 해당되는 오브젝트를 작성하는 것은 사용자의 책임입니다. 예를 들어, ADDPFTRG(실제 파일 트리거 추가) 명령을 사용할 때, 트리거 프로그램이 모든 시스템에 존재하는지 확인해야 합니다. 그렇지 않으면, 오류가 발생합니다. 이와 동일한 개념이 STRJRNPf(저널 실제 파일 시작) 명령에 적용됩니다. 이 때, 저널이 모든 시스템에 존재해야 합니다.

사용자 프로파일이 리모트 노드에 존재하지 않는데 GRTOBJAUT 명령이나 RVKOBJAUT 명령을 낼 경우, 권한이 부여되거나 프로파일이 존재하는 모든 노드에서 취소되어 프로파일이 존재하지 않는 노드에서 무시됩니다.

DB2 Multisystem에서 고려사항 저널링: STRJRNPf(저널 실제 파일 시작) 및 ENDJRNPf(저널 실제 파일 종료) 명령이 다른 시스템에 분산되더라도, 각 시스템에 독립적으로 아울러 각 시스템의 고유 저널 리시버에 대해 실제 저널이 발생합니다.

예로서, 분산 파일이 있는 두 시스템(A 및 B)을 갖고 있습니다. 시스템 A와 시스템 B 모두에서 저널과 리시버를 작성해야 하고, 그 저널명과 라이브러리는 두 시스템에서 같아야 합니다. STRJRNPf 명령을 낼 때, 명령은 두 시스템 모두에서 분산되어 두 시스템 모두에서 저널링이 시작됩니다. 그러나, 시스템 A의 저널 리시버에는 시스템 A에 상주하는 파일 부분에 대해 발생하는 변경사항에 대한 자료만 포함됩니다. 시스템 B의 저널 리시버에는 시스템 B에 상주하는 파일 부분에 대해 발생하는 변경사항에 대한 자료만 포함됩니다.

이렇게 되면 저장 및 복원 전략과 백업 전략에 영향을 줍니다. 예를 들면 다음과 같습니다.

- STRJRNPf 명령을 발행하고 나면, 파일의 노드 그룹에 있는 각 시스템으로부터 데이터베이스 파일을 저장해야 합니다.
- 각 시스템에서 표준 저널 관리를 실습해야 합니다. 각 시스템에 대한 디스크 공간 사용을 관리할 수 있도록 적절하게 저널 리시버를 변경하고 저장해야 합니다. 아니면, 시스템 변경-저널 관리 지원을 사용할 수 있습니다.

주: 저널의 이름만 각 시스템에서 같아야 합니다. 속성은 같지 않습니다. 그러므로, 예를 들어 다른 시스템에서 서로 다른 저널 리시버 임계값(threshold)을 지정하여, 각각의 시스템에서 사용가능한 디스크 공간을 반영할 수 있습니다.

- 분산 데이터베이스 파일을 회복해야 하는 경우, 분산 파일 부분이 상주했던 시스템으로부터 저널 리시버를 사용해야 합니다. 해당 저널 리시버로부터, APYJRNCHG(저널된 변경사항 적용) 명령을 사용하여 저널 변경사항을 적용하거나, RMVJRNCHG(저널된 변경사항 제거) 명령을 사용하여 저널 변경사항을 제거할 수 있습니다.

- 적용할 한 시스템에서 저널 리시버를 사용하거나 다른 시스템에서 파일 부분에 대한 저널된 변경사항을 제거할 수 없게 됩니다. 이것은 각 시스템에 있는 파일의 각 부분에 해당되는 고유한 저널 ID(JID)가 있기 때문입니다.

자세한 정보는 백업 및 회복을 참조하십시오.

DB2 Multisystem에서 분산 파일에 대한 CPYF(파일 복사) 명령 사용: CPYF(파일 복사) 명령이 발행될 때, 시스템은 될 수 있는 한 빠르게 CPYF 명령을 실행하려고 합니다. 지정된 명령 매개변수, 복사에 관련되는 파일 속성, 그리고 복사될 레코드의 크기 및 개수가 명령 실행 속도에 영향을 미칩니다.

자료를 분산 파일에 복사할 때, 복사 명령의 실행은 CPYF 명령에서 FROMFILE, TOFILE, FROMMBR, TOMBR, MBROPT 및 FMTOPT(*NONE) 또는 FMTOPT(*NOCHK) 매개변수만을 사용함으로써 개선될 수 있습니다. 또한, FROMFILE(from-file)과 TOFILE(to-file) 매개변수에서는 널(null) 허용 필드를 포함하는 파일을 지정해서는 안됩니다. 일반적으로, 복사 명령의 구문이 간단할수록 가장 빠른 복사 조작이 확보될 가능성이 더 커집니다. 가장 빠른 복사 방법이 분산 파일에 복사되는 동안 사용될 경우, CPC9203 메시지가 발행됩니다. 이 메시지에는 각 노드에 복사될 레코드 수가 표시됩니다. 일반적으로, 이 메시지가 발행되지 않으면 가장 빠른 방법이 수행되지 않은 것입니다.

분산 파일에 복사할 때, 가장 빠른 복사가 사용될 때와 사용되지 않을 때 사이의 다음과 같은 차이점을 고려해 보십시오.

1. 가장 빠른 복사의 경우, 레코드는 각 노드에 대해 버퍼링됩니다. 버퍼가 가득 차게 되면, 특별한 노드로 보내집니다. 레코드가 노드 버퍼 중 어느 하나에 위치된 후에 오류가 발생할 경우, 시스템은 현재 노드 버퍼에 있는 모든 레코드를 해당되는 노드로 보내려고 합니다. 시스템이 레코드를 특정 노드로 보내는 중에 오류가 발생할 경우, 시스템이 모든 노드 버퍼를 보내려고 할 때까지 처리는 다음 노드로 계속됩니다.

다시 말하면, 오류가 있는 특정 레코드 다음에 오는 레코드들은 분산 파일에 기록될 수도 있습니다. 이러한 조치는 여러 노드에서 동시에 블록화가 수행되기 때문에 발생합니다. 오류가 있는 레코드 다음에 오는 레코드들을 분산 파일에 기록하지 않으려면, CPYF 명령에서 ERRLVL(*NOMAX)나, 1 이상의 값을 갖는 ERRLVL을 지정하여 가장 빠른 복사가 사용되지 못하게 할 수 있습니다.

가장 빠른 복사가 사용되지 않을 경우, to-file 열기가 SEQONLY(*NO) 열기가 되거나 강제로 이렇게 되지 않는 한 레코드 블록화가 시도됩니다.

2. 가장 빠른 복사가 사용될 경우, 멤버의 열기가 SEQONLY(*NO)로 변경되었음을 알리는 메시지가 발행되지만, 분산된 to-file은 레코드 블록화에 대해 허용되도록 두 번 열립니다. SEQONLY(*NO)로의 변경에 대한 메시지는 무시해야 합니다.

3. 가장 빠른 복사 방법이 사용될 경우, 각 노드에 복사될 레코드 수를 표시하는 여러 메시지가 발행될 수도 있습니다. 그 다음에, 복사된 총 레코드 수를 표시하는 메시지가 송신됩니다.

가장 빠른 복사가 사용되지 않을 경우, 복사된 총 레코드 수의 메시지만 송신됩니다. 각 노드에 복사될 레코드 수를 나열하는 메시지는 송신되지 않습니다.

다음은 분산 파일에서, 또는 분산 파일로 복사할 때의 제한사항입니다.

- FROMRCD 매개변수는 분산 파일로부터 복사할 때 *START 또는 1의 값으로만 지정할 수 있습니다. TORCD 매개변수는 분산 파일로부터 복사할 때 디폴트 값 *END 이외의 값으로 지정할 수 없습니다.
- MBROPT(*UPDADD) 매개변수는 분산 파일로 복사할 때 허용되지 않습니다.
- COMPRESS(*NO) 매개변수는 to-file이 분산 파일이고 from-file이 데이터베이스 삭제 가능 파일일 경우 허용되지 않습니다.
- 인쇄 리스트를 복사할 경우, 제공된 RCDNBR 값은 레코드가 분산 파일 레코드일 때 특정 노드의 파일에서 레코드의 위치입니다. 리스트에서 같은 레코드 번호가 여러 번 표시될 수 있습니다. 각 레코드는 서로 다른 노드로부터의 레코드가 됩니다.

DB2 Multisystem에서의 분할

분할은 노드 그룹의 노드들에 파일을 분산시키는 프로세스입니다. 분할은 해시 알고리즘을 사용하여 수행됩니다. 새 레코드가 추가될 경우, 해시 알고리즘이 분할 키의 자료에 적용됩니다. 그러면, 해시 알고리즘의 결과(0 - 1023 사이의 숫자)가 분할 맵에 적용되어 레코드가 상주하게 될 노드를 판별합니다.

파티션 맵은 또한 조회 최적화, 갱신, 삭제 및 결합에도 사용됩니다. 파티션 맵은 특정 키 값이 특정 노드에 강제로 맵핑되도록 사용자가 정의할 수 있습니다.

예를 들어, I/O 중에 시스템은 분할 키 필드의 값에 해시 알고리즘을 적용합니다. 결과는 파일에 저장된 파티션 맵에 적용되어 레코드를 저장하는 노드를 판별합니다.

다음 예는 이러한 개념들이 서로 어떻게 관련되는지 나타내고 있습니다.

사원 번호가 분할 키이고 레코드가 사원 번호 56000에 대한 데이터베이스에 입력됩니다. 56000의 값은 해시 알고리즘에 의해 처리되고 결과는 파티션 번호 733입니다. 노드 그룹 오브젝트의 일부이고 작성시 분산 파일에 저장된 파티션 맵에는 파티션 번호 733에 대한 노드 번호 1이 들어 있습니다. 그러므로, 이 레코드는 실제로 노드 번호 1이 할당된 노드 그룹의 시스템에 저장됩니다. 분할 키(PKNKEY 매개변수)는 사용자가 분할된(분산된) 파일을 작성할 때 지정한 것입니다.

분할 키의 필드는 널(null) 허용이 가능할 수 있습니다. 그러나, 분할 키 내에 널값(null value)을 포함하는 레코드는 항상 파티션 번호 0으로 해싱됩니다. 분할 키 내에 널값

수가 유의할만큼 있는 파일들은 파티션 번호 0에서 자료 왜곡을 초래할 수 있습니다. 널값을 갖는 모든 레코드가 파티션 번호 0으로 해싱되기 때문입니다.

노드 그룹 오브젝트와 분할된 분산 관계형 데이터베이스 파일을 작성한 후에, DSPNODGRP 명령을 사용하여 파티션 번호와 노드명 사이의 관계를 볼 수 있습니다. 파티션 번호, 노드 그룹 및 시스템명 표시에 대한 자세한 정보는 9 페이지의 『DB2 Multisystem에서 DSPNODGRP 명령을 사용하여 노드 그룹 표시』를 참조하십시오.

분산 파일을 작성할 때, 분할 키 필드는 CRTPF(실제 파일 작성) 명령의 PTNKEY 매개변수나, SQL CREATE TABLE문의 PARTITIONING KEY절에 지정됩니다. 자료 유형이 DATE, TIME, TIMESTAMP 및 FLOAT인 필드들은 분할 키로 허용되지 않습니다.



파티션 구현에 관해서는 다음 주제를 참조하십시오.

- 『DB2 Multisystem에서의 분할 계획』
- 25 페이지의 『DB2 Multisystem에서 분할 키 선택』

DB2 Multisystem에서의 분할 계획

대부분의 경우, 분할 및 분할 키를 사용하려고 하는 것에 대해 먼저 계획해야 합니다. 다른 시스템에 배치하기 위해 자료를 조직적으로 나누는 방법은 무엇인가? 조회에서 자주 결합하려고 하는 자료는 무엇인가? 선택사항을 수행할 때 의미있는 선택은 무엇인가? 필요한 자료를 얻기 위해 분할 키를 설정하는 가장 효율적인 방법은 무엇인가?

분할을 계획할 때, 가장 빠른 시스템이 대부분의 자료를 수신하도록 분할을 설정해야 합니다. 데이터베이스 성능을 향상시키려면 시스템이 SMP(Symmetric Multiprocessing)를 사용하는 것을 고려해야 합니다. 조회 Optimizer가 분산 액세스 계획을 빌드할 때, 이 Optimizer는 요구하는 노드에서 레코드 수를 계산하여 그 수에 총 노드 수를 곱합니다. SMP 시스템에 대부분의 레코드를 두는 것이 장점이 있긴 하지만, Optimizer는 계산을 위해 각 노드에서 같은 수의 레코드를 사용하므로 그러한 장점 중 일부를 상쇄

시킬 수 있습니다. SMP에 관해 알려면 SQL 프로그래밍 개념  및 데이터베이스 프로그래밍  주제를 참조하십시오.

분할에 영향을 주고자 하는 경우, 그렇게 할 수 있습니다. 예를 들어, 사용자의 사업에서 작업을 완료하기 위해 특정 시스템을 사용하는 지역 판매 부서가 있을 경우, 분할을 사용하여 각 지역으로부터의 로컬 자료가 강제로 해당 지역에 대해 적절한 시스템에 저장되도록 할 수 있습니다. 그러므로, Northwest United States 영역에 있는 사원이 사용하는 시스템에 Northwest 영역에 대한 자료가 포함되게 됩니다.

분할을 설정하려면, CRTPF 명령의 PTNFILE 및 PTNMBR 매개변수를 사용하면 됩니다. CHGNODGRPA(노드 그룹 속성 변경) 명령을 사용하여 이미 분할된 파일을 재분산시킬 수 있습니다. 자세한 정보는 26 페이지의 『DB2 Multisystem에서 자료 분배의 사용자 정의』를 참조하십시오.

큰 파일에 대해 수행되는 조회의 경우 성능은 최적으로 향상됩니다. 트랜잭션 처리에 대해 사용율이 높지만 조회에 드물게 사용되는 파일들은 분할을 위해 최적의 후보가 될 수 없으며 로컬 파일로 남겨 두어야 합니다.

결합 처리에 대해, 특정 필드의 두 파일을 자주 결합할 경우, 해당 필드를 두 파일에 대한 분할 키로 만들어야 합니다. 또한, 그 필드들의 자료 유형이 같은지 확인해야 합니다.

DB2 Multisystem에서 분할 키 선택

가장 효율적인 방식으로 시스템이 분할된 파일을 처리하도록 하기 위해, 분할 키를 설정하거나 사용할 때 다음 추가 정보를 고려해야 합니다.

- 최상의 분할 키는 많은 다른 값을 갖는 키이므로, 분할 활동으로 인해 자료 레코드의 분산까지 초래됩니다. 고객 번호, 성, 청구 번호, 우편 번호(지역 우편 주소 코드) 및 전화 지역 코드는 분할 키로 사용할 좋은 범주의 예입니다.

성별은 남성 또는 여성의 두 가지 선택만 존재하므로 분할 키로서 좋지 않은 선택의 예입니다. 성별은 너무 많은 자료가 노드에서 분산되는 대신 단일 노드로 분산되도록 합니다. 또한, 조회를 수행할 때 분할 키로 성별을 사용하면 시스템이 너무 많은 자료 레코드들을 통해 처리해야 합니다. 이것은 비효율적입니다. 자료의 다른 필드는 조회 범위를 좁게 하여 더 효율적이 되게 합니다. 성별을 기초로 하는 분할 키는 특정 값에 기초하여 분산되기 보다는 자료 분산까지 원할 경우에 좋지 않은 선택입니다.

로컬 파일을 분산 파일로 변경하려고 준비할 때, HASH 함수를 사용하여 자료가 분산될 수 있는 방법을 구할 수 있습니다. 로컬 파일이나 다양한 열에 대해 HASH 함수를 사용할 수 있으므로, 실제로 파일을 분산 파일로 변경하기 전에 다른 분할 키를 시도할 수 있습니다. 예를 들어, 파일의 ZIP 코드 필드를 사용할 계획이 있으면, 해당 필드를 사용하여 HASH 함수를 실행함으로써 각 파티션 번호로 HASH 처리할 레코드 수를 알 수 있습니다. 이렇게 되면 분할 키 필드를 선택하거나, 노드 그룹에서 분할 맵을 작성할 때 도움이 될 수 있습니다.

- 자주 갱신해야 하는 필드는 선택하지 마십시오. 분할 키 필드에 대한 제한사항은 다른 노드로의 레코드 갱신이 강요되지 않을 경우에만 값이 갱신될 수 있다는 것입니다.
- 분할 키에서 많은 필드를 사용하지 마십시오. 최상의 선택은 하나의 필드를 사용하는 것입니다. 많은 필드를 사용하면 시스템이 I/O 시간에 강제로 더 많은 작업을 수행하게 합니다.

- 고정 길이 문자나 정수와 같은 간단한 자료 유형을 분할 키로 선택하십시오. 이러한 고려사항은 해싱이 간단한 자료 유형의 단일 필드에 대해 수행되므로 성능 향상에 기여합니다.
- 분할 키를 선택할 때, 일반적으로 실행하는 조회의 결합 및 그룹화 기준을 고려해야 합니다. 예를 들어, 결합에 관여하는 파일에 대해 결합 필드로 사용되지 않은 필드를 선택하면 결합 성능에 불리하게 영향을 미칠 수 있습니다. 분산 파일을 포함하는 조회 실행에 대해서는 제 6 장 『DB2® Multisystem에서 성능 향상을 위한 조회 설계』를 참조하십시오.

DB2 Multisystem에서 자료 분배의 사용자 정의

시스템이 자료 배치에 대한 책임을 지고 있으므로, 사용자는 레코드가 실제로 상주하는 곳을 알지 않아도 됩니다. 그러나, 특정 레코드가 항상 특정 시스템에 저장되도록 하려면, CHGNODGRPA(노드 그룹 속성 변경) 명령을 사용하여 그 레코드들이 상주하는 곳을 지정할 수 있습니다.

예로서, 55902 ZIP 코드에 대한 모든 레코드가 Minneapolis, Minnesota의 시스템에 상주하기를 원할 경우, CHGNODGRPA 명령을 낼 때, 55902 ZIP 코드와 Minneapolis에 있는 로컬 노드의 시스템 노드 번호를 지정합니다.

이 때, 55902 ZIP이 노드 그룹을 변경하였지만, 자료는 여전히 이전처럼 분산됩니다. CHGNODGRPA 명령은 기존 파일에 영향을 주지 않습니다. 분할된 파일이 작성될 때, 그 분할된 파일은 그 당시 노드 그룹으로부터의 정보 사본을 저장합니다. 노드 그룹은 분할된 파일에 영향을 주지 않고 변경되거나 삭제될 수 있습니다. 재분산될 레코드에 대한 변경사항이 적용되도록, 새로운 노드 그룹을 사용하여 분산 파일을 재작성하거나, CHGPF(실제 파일 변경) 명령을 사용하고 새로운 또는 갱신된 노드 그룹을 지정할 수 있습니다.

CHGPF 명령을 사용하여 다음을 수행할 수 있습니다.

- 이미 분할된 파일 재분산
- 분할 키 변경(예: 전화 지역 코드에서 분기 ID로)
- 분산 파일이 되도록 로컬 파일 변경
- 분산 파일을 로컬 파일로 변경



주: 또한 CHGNODGRPA 명령을 사용하여 이미 분할된 파일을 재분산해야 합니다. CHGNODGRPA 명령을 선택적으로 CHGPF 명령과 함께 사용하여 다른 작업을 수행할 수 있습니다.

로컬 파일을 분산 파일로, 또는 분산 파일을 로컬 파일로 변경하는 작업에 대해서는 36 페이지의 『시스템을 네트워크에 추가하기 위한 재분산 문제』를 참조하십시오.

제 4 장 DB2 Multisystem에서 사용할 수 있는 스칼라 함수

DB2 Multisystem의 경우, 분산 파일에 대해 작업할 때 사용할 수 있도록 새로운 스칼라 함수가 제공됩니다. 이 함수들은 파일에서 자료를 분산시키는 방법을 판별하고 파일이 분산된 후에 자료가 있는 곳을 판별할 때 도움이 됩니다. 분산 파일에 대해 작업할 때, 데이터베이스 관리자는 이 함수들을 유용한 디버깅 툴로 사용하기 위해 찾을 수도 있습니다.

이러한 스칼라 함수로는 PARTITION, HASH, NODENAME, NODENUMBER이 있습니다. SQL이나 OPNQRYF(조회 파일 열기) 명령을 통해 이 함수들을 사용할 수 있습니다.

SQL 스칼라 함수의 구문에 대한 정보는 SQL 참조서  를 참조하십시오. OPNQRYF 스칼라 함수 구문에 대한 정보는 Informaton Center에서 제어 언어  주제를 참조하십시오.

- 『DB2 Multisystem에서 PARTITION』
- 28 페이지의 『DB2 Multisystem에서 HASH』
- 29 페이지의 『DB2 Multisystem에서 NODENAME』
- 30 페이지의 『DB2 Multisystem에서 NODENUMBER』
- 31 페이지의 『DB2 Multisystem에서 특수 레지스터』

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 PARTITION

PARTITION 함수를 통해, 분산 관계형 데이터베이스(RDB)의 특정 행이 저장되는 파티션 번호를 판별할 수 있습니다. 파티션 번호를 알면 노드 그룹에서 해당 파티션 번호를 포함하는 노드를 판별할 수 있습니다.

PARTITION의 코드 예는 『DB2 Multisystem에서 PARTITION 예』를 참조하십시오.

DB2 Multisystem에서 PARTITION 예

- EMPLOYEE 표의 모든 행에 대해 PARTITION 번호를 찾습니다.

주: 이것은 자료 왜곡이 있는지 판별하기 위해 사용될 수 있습니다.

SQL문:

```
SELECT PARTITION(CORPDATA.EMPLOYEE), LASTNAME
FROM CORPDATA.EMPLOYEE
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE))
FORMAT(FNAME)
MAPFLD((PART1 '%PARTITION(1)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

- 파티션 번호가 100인 모든 행에 대해 EMPLOYEE 표에서 EMPNO(사원 번호)를 선택합니다.

SQL문:

```
SELECT EMPNO
FROM CORPDATA.EMPLOYEE
WHERE PARTITION(CORPDATA.EMPLOYEE) = 100
```

OPNQRYF 명령:

```
OPNQRYF FILE((EMPLOYEE)) QRYSLT('%PARTITION(1) *EQ 100')
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

- EMPLOYEE와 DEPARTMENT 표를 결합하고, 두 표의 행이 같은 파티션 번호를 갖고 있는 결과의 모든 행을 선택합니다.

SQL문:

```
SELECT *
FROM CORPDATA.EMPLOYEE X, CORPDATA.DEPARTMENT Y
WHERE PARTITION(X)=PARTITION(Y)
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
FORMAT(FNAME)
JFLD((1/PART1 2/PART2 *EQ))
MAPFLD((PART1 '%PARTITION(1)')
(PART2 '%PARTITION(2)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 HASH

HASH 함수는 해시 함수를 지정된 표현식에 적용하여 파티션 번호를 리턴합니다.

HASH의 코드 예는 29 페이지의 『DB2 Multisystem에서 HASH 예』를 참조하십시오.

DB2 Multisystem에서 HASH 예

- HASH 함수를 사용하여 분할 키가 EMPNO 및 LASTNAME으로 구성된 경우의 파티션을 판별합니다. 이 조회는 EMPLOYEE의 모든 행에 대해 파티션 번호를 리턴합니다.

SQL문:

```
SELECT HASH(EMPNO, LASTNAME)
FROM CORPDATA.EMPLOYEE
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE))
          FORMAT(FNAME)
          MAPFLD((HASH '%HASH(1/EMPNO, 1/LASTNAME)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 NODENAME

NODENAME 함수를 통해, 분산 관계형 데이터베이스(RDB)의 특정 행이 저장되는 관계형 데이터베이스(RDB)의 이름을 판별할 수 있습니다. 노드명을 알면 그 행을 포함하는 시스템명을 판별할 수 있습니다. 이것은 특정 행을 특정 노드로 재분산시킬 경우에 유용합니다.

NODENAME의 코드 예는 『DB2 Multisystem에서 NODENAME 예』를 참조하십시오.

DB2 Multisystem에서 NODENAME 예

- EMPLOYEE 표의 모든 행에 대해 노드명과 파티션 번호를 찾고, 상응하는 EMPNO 열의 값을 각 행에 대해 찾습니다.

SQL문:

```
SELECT NODENAME(CORPDATA.EMPLOYEE), PARTITION(CORPDATA.EMPLOYEE), EMPNO
FROM CORPDATA.EMPLOYEE
```

- EMPLOYEE 표의 모든 레코드에 대해 노드명을 찾습니다.

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE))
          FORMAT(FNAME)
          MAPFLD((NODENAME '%NODENAME(1)'))
```

- EMPLOYEE 및 DEPARTMENT 표를 결합하고, EMPNO(사원 번호)를 선택한 후 결합에 관련된 각 행이 시작된 노드를 판별합니다.

SQL문:

```
SELECT EMPNO, NODENAME(X), NODENAME(Y)
FROM CORPDATA.EMPLOYEE X, CORPDATA.DEPARTMENT Y
WHERE X.DEPTNO=Y.DEPTNO
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
          FORMAT(FNAME)
          JFLD((EMPLOYEE/DEPTNO DEPARTMENT/DEPTNO *EQ))
          MAPFLD((EMPNO 'EMPLOYEE/EMPNO')
                 (NODENAME1 '%NODENAME(1)')
                 (NODENAME2 '%NODENAME(2)'))
```

- EMPLOYEE와 DEPARTMENT 표를 결합하고, 두 표의 행이 같은 노드에 있는 결과의 모든 행을 선택합니다.

SQL문:

```
SELECT *
  FROM CORPDATA.EMPLOYEE X, CORPDATA.DEPARTMENT Y
 WHERE NODENAME(X)=NODENAME(Y)
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
          FORMAT(FNAME)
          JFLD((1/NODENAME1 2/NODENAME2 *EQ))
          MAPFLD((NODENAME1 '%NODENAME(1)')
                 (NODENAME2 '%NODENAME(2)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 NODENUMBER

NODENUMBER 함수를 통해, 분산 관계형 데이터베이스(RDB)의 특정 행이 저장되는 노드 번호를 판별할 수 있습니다. 노드 번호는 노드 그룹이 작성될 때 노드 그룹 내에서 각 노드에 대해 할당되는 고유한 번호입니다. 노드 번호를 알면 그 행을 포함하는 시스템명을 판별할 수 있습니다. 이것은 특정 행을 특정 노드로 재분산시킬 경우에 유용합니다.

NODENUMBER의 코드 예는 『DB2 Multisystem에서 NODENUMBER 예』를 참조하십시오.

DB2 Multisystem에서 NODENUMBER 예

- CORPDATA.EMPLOYEE가 분산표인 경우, 각 행에 대한 노드 번호와 사원명이 리턴됩니다.

SQL문:

```
SELECT NODENUMBER(CORPDATA.EMPLOYEE), LASTNAME
  FROM CORPDATA.EMPLOYEE
```

OPNQRYF 명령:

```
OPNQRYF FILE((CORPDATA/EMPLOYEE))
          FORMAT(FNAME)
          MAPFLD((NODENAME '%NODENUMBER(1)')
                (LNAME '1/LASTNAME'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 특수 레지스터

DB2 Multisystem의 경우, 특수 레지스터의 모든 인스턴스는 조회를 리모트 노드로 송신하기 전에 조정 담당자 노드에서 분석됩니다. (조정 담당자 노드는 조회가 개시된 시스템입니다.) 이 방법에서 모든 노드는 일관성 있는 특수 레지스터 값으로 조회를 실행합니다.

다음은 특수 레지스터에 대한 규칙입니다.

- CURRENT SERVER는 항상 조정 담당자 노드의 관계형 데이터베이스(RDB)명을 리턴합니다.
- USER 특수 레지스터는 조정 담당자 노드에서 작업을 실행하는 사용자 프로파일을 리턴합니다.
- CURRENT DATE, CURRENT TIME 및 CURRENT TIMESTAMP는 조정 담당자 노드에 있는 24시간 시계에서 제공됩니다.
- CURRENT TIMEZONE은 조정 담당자 노드에서의 시스템 값 QUTCOFFSET의 값입니다.

분산 파일에서 노드에 저장된 행의 상대 레코드 번호를 리턴시키려면 『DB2 Multisystem에서 RRN(상대 레코드 번호)부여 함수』를 참조하십시오.

DB2 Multisystem에서 RRN(상대 레코드 번호)부여 함수

RRN 함수는 분산 파일에서 노드에 저장된 행의 상대 레코드 번호를 리턴합니다. 이것은 RRN이 분산 파일에 대해 고유하지 않음을 의미합니다. 파일의 고유한 레코드는 RRN을 NODENAME 또는 NODENUMBER와 결합시킬 경우에 지정됩니다.

제 5 장 DB2 Multisystem에서 성능 및 확장성

DB2 Multisystem을 사용하면, 데이터베이스 용량을 증가시키고, 조회 실행을 향상시키며, 보다 쉬운 방법을 통한 리모트 데이터베이스 액세스를 제공할 수 있습니다.

DB2 Multisystem을 사용할 때, 사용자들과 어플리케이션들은 로컬 시스템으로부터 파일에 액세스해야 합니다. 분산 파일 대 로컬 파일의 자료 액세스가 가능하도록 코드를 변경할 필요가 없습니다. 분산 관계형 데이터베이스 구조™(DRDA) 및 분산 자료 관리(DDM)와 같은 기능으로, 사용자의 액세스는 해당 리모트 자료에 액세스할 수 있도록 명시적으로 리모트 파일이나 리모트 시스템에 대해 지시되어야 합니다. DB2 Multisystem은 사용자들에게 투명한 방식으로 리모트 액세스를 처리합니다.

DB2 Multisystem은 또한 데이터베이스 확장을 위한 간단한 방법을 제공합니다.

DB2 Multisystem의 성능과 확장성에 관해 알려면 다음 주제를 참조하십시오.

- 『DB2 Multisystem을 사용해야 하는 이유』
- 35 페이지의 『데이터베이스 확장을 지원하는 DB2 Multisystem』

DB2 Multisystem을 사용해야 하는 이유

특정 조회에 대해 성능 향상은 상당히 중요할 수 있습니다. 많은 양의 자료를 갖고 있지만 상대적으로 결과 세트가 작은 조회의 경우, 성능은 파일이 분산되는 시스템 수에 거의 비례한다는 것이 테스트에서 밝혀졌습니다. 예를 들어, 상위 10명의 수익 작성자에 대해 조회할 5백만(5 000 000) 레코드 파일을 갖고 있을 경우, DB2 Multisystem을 사용하면 조회에 대한 응답 시간이 두 시스템에서 동등하게 파일을 분할하여 거의 50%로 단축됩니다. 세 개의 시스템에서는 응답 시간으로 단일 시스템에서 조회를 실행할 때의 시간의 1/3이 소요됩니다. 이러한 최적의 경우에 대한 시나리오는 노드 사이에 자료가 이동되어야 하는 복잡한 결합에는 적용되지 않습니다.

파일이 아주 작거나 기본적으로 단일 레코드 읽기/쓰기 처리에 사용될 경우, 파일을 분할할 때 실현되는 성능상의 이득은 약간 있거나 거의 없습니다. 오히려, 약간의 성능 감소가 발생할 수도 있습니다. 이러한 경우, 조회 실행은 실제 연결 속도에 더 의존하게 됩니다. 그러나, 이러한 상황에서도 노드 그룹에 있는 모든 시스템의 사용자들은 자료가 분산되더라도 익숙한 일반적인 “로컬 파일” 데이터베이스 방법을 사용하여 자료에 액세스할 수 있다는 장점을 갖고 있습니다. 모든 환경에서, 사용자들은 이러한 로컬 시스템 투명성의 이점과 노드 그룹에 있는 시스템들에서의 자료 중복성을 없앨 수 있는 가능성을 갖고 있습니다.

다른 병행 피처는 DB2 UDB 대칭형 멀티프로세싱입니다. SMP(대칭적 다중 처리)로, 분할된 파일이 처리되고 시스템 중 어느 것이 멀티프로세서 시스템일 경우, 성능 이득 관점에서 배가되는 효과를 거둘 수 있습니다. 파일을 세 개의 시스템에 분할하고 각 시스템이 4중 프로세서 시스템일 경우, DB2 Multisystem 및 SMP의 기능들이 함께 작동합니다. 이전의 5백만 레코드 예를 사용하면, 응답 시간으로 병행 피처를 사용하지 않고 조회를 실행했을 때의 약 1/12에 해당되는 시간이 소요됩니다. 파일 크기와 조회의 세부사항이 실제로 보는 개선점에 영향을 줄 수 있습니다.

조회를 수행할 때, 조회를 실행하기 위해 대량 작업이 병렬로 수행되고 이로써 조회 처리의 전체 성능이 향상됩니다. 시스템은 조회를 나누고 적절한 시스템에서 조회의 적절한 부분들을 처리합니다. 이렇게 하면 가장 효율적으로 처리할 수 있게 되고, 이것은 자동으로 수행되므로, 이러한 효율적인 처리가 발생하도록 하기 위해 사용자가 지정할 것이 없어지게 됩니다.

주: 일부 조회의 성능은 향상되지 않을 수도 있습니다. 특히, 자료 볼륨이 큰 자료가 이동되어야 할 경우에 그렇습니다.

각 노드는 실제로 해당 노드에 저장되는 레코드들만 처리해야 합니다. 조회에서 분할 키에 대해 선택사항을 지정할 경우, 조회 Optimizer는 하나의 노드만 조회되어야 하는 것을 판별할 수도 있습니다. 다음 예에서, ZIP 코드 필드는 ORDERS 파일에 대한 SQL 문 내에서 분할 키입니다.

```
SELECT NAME, ADDRESS, BALANCE FROM PRODLIB/ORDERS WHERE ZIP='48009'
```

명령문이 실행될 때, Optimizer는 하나의 노드만 조회되어야 함을 판별합니다. 48009 ZIP 코드를 포함하는 모든 레코드가 같은 노드로 분산된다는 점에 유의하십시오.

다음 SQL문 예에서, 노드 그룹에서 모든 iSeries 서버의 프로세서 기능을 사용하여 명령문을 병렬로 처리할 수 있습니다.

```
SELECT ZIP, SUM(BALANCE) FROM PRODLIB/ORDERS GROUP BY ZIP
```

관련 자료를 포함하는 시스템에 대해서만 Optimizer가 직접 I/O 요구를 할 수 있게 할 때의 또 다른 장점은 하나 이상의 시스템이 사용 중이 아닐 때도 여전히 조회가 실행될 수 있다는 것입니다. 예로서, 각 업무 지점이 서로 다른 시스템에 해당 자료를 저장하고 있는 것과 같이 분할된 파일을 들 수 있습니다. 하나의 시스템이 사용가능하지 않을 경우, 파일 I/O 조작은 나머지 지점과 연관되는 자료에 대해 계속 수행될 수 있습니다. 사용 중이 아닌 지점에 대해서는 I/O 요구가 실패합니다.

Optimizer는 2단계 확약 프로토콜을 사용하여 자료의 무결성을 보장합니다. 여러 시스템에 액세스되므로, 확약 제어를 요구할 경우 모든 시스템이 보호되는 대화를 사용하게 됩니다. 보호되는 대화라는 것은 트랜잭션 중간이나 단일 데이터베이스 조작시 시스템 장애가 발생할 경우, 그 지점까지 수행된 모든 변경사항이 구간 복원됨을 의미합니다.

보호되는 대화가 사용될 경우, 일부 확약 제어 옵션은 성능 향상을 위해 리모트 노드에서 변경됩니다. 결과 대기 옵션이 Y로 설정되고, 보우트(Vote) 읽기 전용 허용 옵션이 Y로 설정되어 있습니다. 성능을 더 향상시키려면, QTNCHGCO(확약 옵션 변경) API를 사용하여 조희가 개시되는 시스템에서 결과 대기 옵션을 N으로 설정할 수 있습니다. 확약 옵션 값에 대한 설명은 Information Center에서 OS/400 API 주제를 참조하십시오.

확약 제어에 대한 자세한 정보는 백업 및 회복 책을 참조하십시오.

최상의 성능을 보장하는 방법에 관해 알려면 『DB2 Multisystem에서 성능 향상을 위한 추가 정보』를 참조하십시오.

DB2 Multisystem에서 성능 향상을 위한 추가 정보

최상의 성능을 보장하는 한 가지 방법은 OVRDBF(데이터베이스 파일 대체) 명령의 DSTDTA(자료 분배) 매개변수에 대해 *BUFFERED를 지정하는 것입니다. 이것은 시스템이 잠재적으로 파일에 대해 중간 갱신이 수행되더라도 될 수 있는 한 빠르게 분산 파일로부터 자료를 검색하도록 지시합니다. DSTDTA(*BUFFERED)는 파일이 읽기 전용으로 열릴 때 매개변수에 대한 디폴트 값입니다.

DSTDTA 매개변수의 다른 값으로는 *CURRENT와 *PROTECTED가 있습니다. *CURRENT는 다른 사용자가 파일을 갱신할 수 있게는 하지만, 성능이 다소 저하됩니다. DSTDTA(*BUFFERED)는 파일이 갱신을 위해 열릴 때의 디폴트 값입니다. *PROTECTED는 *CURRENT의 경우와 유사한 성능을 제공하지만, *PROTECTED는 파일이 열려 있는 동안 다른 사용자들이 갱신하지 못하게 합니다.

데이터베이스 확장을 지원하는 DB2 Multisystem

분산 관계형 데이터베이스 파일을 사용하여 간편하게 iSeries 서버 구성을 확장할 수 있습니다. DB2 Multisystem 이전에, 하나의 시스템에서 두 개의 시스템으로 진행하길 원한 경우, 몇 개의 데이터베이스 문제를 해결해야 했습니다. 사용자들의 반을 새로운 시스템으로 이동한 경우, 자료의 반도 그 새로운 시스템으로 이동하고자 할 것입니다. 이때, 모든 데이터베이스 관련 어플리케이션을 재작성해야 합니다. 어플리케이션이 자료가 상주하는 곳을 알아야 하기 때문입니다. 어플리케이션을 재작성한 후에, 분산 관계형 데이터베이스 구조(DRDA)나 분산 자료 관리(DDM)와 같은 일부 리모트 액세스를 사용하여 시스템들에 분산된 파일들에 액세스해야 합니다. 그렇지 않으면, 일부 자료 복제 기능을 사용해야 합니다. 이를 수행한 경우, 자료의 복수 사본이 존재하게 되어, 더 많은 기억장치가 사용되고 시스템도 동시에 파일들의 복수 사본을 저장하는 작업을 수행해야 합니다.

DB2 Multisystem으로, 구성에 새로운 시스템들을 추가하는 프로세스가 아주 간단해졌습니다. 데이터베이스 파일은 시스템들을 교차하여 분할됩니다. 그러면, 어플리케이션이 새로운 시스템으로 이동됩니다. 어플리케이션은 변경되지 않은 상태로 있습니다. 어플리

케이션의 프로그래밍을 변경하지 않아도 됩니다. 사용자들은 이제 새로운 시스템에서 실행할 수 있으며 즉시 같은 자료에 대한 액세스 권한을 갖습니다. 나중에 추가 증대가 필요하면, 추가 시스템들을 포함하는 새로운 노드 그룹에서 파일을 재분산시키면 됩니다.

시스템을 네트워크에 추가할 때 알아야 할 사항에 관해서는 『시스템을 네트워크에 추가하기 위한 재분산 문제』를 참조하십시오.

시스템을 네트워크에 추가하기 위한 재분산 문제

노드 그룹에서 파일을 재분산하는 것은 아주 간단한 프로세스입니다. CHGPF(실제 파일 변경) 명령을 사용하여 파일의 새로운 노드 그룹이나 파일의 신규 분할 키를 지정할 수 있습니다. CHGPF 명령은 로컬 파일을 분산 파일로 만들거나, 분산 파일을 로컬 파일로 만들거나, 아니면 분산 파일을 다른 노드 세트에서 또는 다른 분할 키를 사용하여 재분산시키기 위해 사용됩니다.

재분산 프로세스에는 파일에 있는 거의 모든 레코드의 이동이 포함될 수 있다는 점에 유의하십시오. 매우 큰 파일의 경우, 이것은 파일의 자료를 사용할 수 없는 장기 프로세스가 될 수 있습니다. 적절한 계획 없이 또는 자주 파일 재분산을 수행해서는 안 됩니다.

로컬 실제 파일을 분산 파일로 변경하려면, CRTPF 명령에서 NODGRP(노드 그룹)과 PTNKEY(분할 키) 매개변수를 지정해야 합니다. 이 명령을 내리면 파일이 노드 그룹의 노드들에 걸쳐 분산되도록 변경되고, 기존 자료도 PTNKEY 매개변수에 지정된 분할 키를 사용하여 분산됩니다.

분산 파일을 로컬 파일로 변경하려면, CHGPF 명령에 NODGRP(*NONE)를 지정해야 합니다. 이렇게 하면 파일의 모든 리모트 부분이 삭제되고 강제로 모든 자료가 다시 로컬 시스템으로 보내지도록 합니다.

분산 파일의 분할 키를 변경하려면, CHGPF 명령의 PTNKEY 매개변수에 원하는 필드를 지정하십시오. 이렇게 해도 파일이 분산되는 시스템에 영향을 주지 않습니다. 이를 수행하면 모든 자료가 재분산됩니다. 해싱 알고리즘이 신규 분할 키에 적용되어야 하기 때문입니다.

파일이 분산되어야 하는 새로운 시스템 세트를 지정하려면, CHGPF 명령의 NODGRP(노드 그룹) 매개변수에 노드 그룹명을 지정하십시오. 이렇게 하면 파일이 이 새로운 시스템 세트에서 분산됩니다. PTNKEY 매개변수에 신규 분할 키를 지정할 수 있습니다. PTNKEY 매개변수를 지정하지 않거나 *SAME을 지정할 경우, 기존의 분할 키가 사용됩니다.

CHGPF 명령은 노드 그룹이 새로운 시스템들을 추가한 경우 파일의 새로운 부분들의 작성을 처리합니다. CHGPF 명령은 시스템이 새로운 노드 그룹에 없을 경우에 파일의

여러 부분들에 대한 삭제를 처리합니다. 노드 그룹을 삭제하고 재작성한 후 CRTPF 명령을 사용하여 파일을 재분산시키려면, 노드 그룹명이 처음 파일이 작성될 때 사용된 것과 같은 경우에도 CHGPF 명령의 NODGRP 매개변수에 노드 그룹명을 지정해야 합니다. 이렇게 하면 시스템이 노드 그룹을 찾아서 파일을 재분산시키기를 원한다는 것을 나타냅니다. 그러나, NODGRP 매개변수에 노드 그룹을 지정하고 그 노드 그룹이 현재 파일에 저장된 노드 그룹과 같다는 것을 시스템이 인식할 경우, PTNKEY도 지정하지 않은 한 재분산이 실행되지 않습니다.

참조 제한 조건이 있는 파일들의 경우, CHGPF 명령을 사용하여 상위 파일과 종속 파일을 분산 파일로 만들려면 다음을 수행해야 합니다.





- 참조 제한 조건을 제거하십시오. 제한 조건을 제거하지 않으면, 사용자가 분산하는 작업이 먼저 제한 조건 오류를 발견하게 됩니다. 그 이유는 참조 제한 조건 관계의 다른 파일이 아직 분산 파일이 아니기 때문입니다.
- CHGPF 명령을 사용하여 두 파일을 모두 분산 파일로 만드십시오.
- 참조 제한 조건을 다시 추가하십시오.

제 6 장 DB2[®] Multisystem에서 성능 향상을 위한 조회 설계

이 장에서는 분산 파일을 사용하는 조회를 실행할 때 더 효율적으로 조회 자원을 사용하도록 조회를 설계하기 위한 몇 가지 지침을 제공합니다. 또한, 분산 파일을 사용하는 조회가 구현되는 방법에 대해서도 설명합니다. 이 정보는 분산 환경에서 더 효율적으로 실행되도록 조회를 조정하는 데 사용할 수 있습니다.

분산 파일은 SQL, OPNQRYF(조회 파일 열기) 명령, 또는 시스템의 조회 인터페이스를 사용하여 조회할 수 있습니다. 조회는 단일 파일 조회 또는 결합 조회가 될 수 있습니다. 결합에서 분산 파일과 로컬 파일의 조합을 사용할 수 있습니다.

이 장에서는 사용자가 비분산 환경에서 조회를 실행하고 최적화하는 작업에 익숙해 있다고 가정합니다. 이 주제들에 대한 추가 정보를 원할 경우,

- SQL 사용자라면 SQL 참조서  와 SQL 프로그래밍 개념  주제를 참조하십시오.
- SQL 사용자가 아니라면 데이터베이스 프로그래밍  과 CL 참조서  주제를 참조하십시오.

이 장은 또한 병렬 처리를 사용하고 자료 이동을 최적화하여 분산 조회의 실행을 개선하는 방법도 나타내고 있습니다.

- 40 페이지의 『DB2 Multisystem에서 최적화 개요』
- 40 페이지의 『DB2 Multisystem에서 단일 파일 조회 구현 및 최적화』
- 42 페이지의 『DB2 Multisystem에서 레코드 배열 구현 및 최적화』
- 43 페이지의 『DB2 Multisystem에서 UNION 및 DISTINCT 절 구현과 최적화』
- 43 페이지의 『DB2 Multisystem에서 DSTDTA 및 ALWCPYDTA 매개변수 처리』
- 44 페이지의 『DB2 Multisystem에서 결합 구현 및 최적화』
- 49 페이지의 『DB2 Multisystem에서 그룹화 구현 및 최적화』
- 52 페이지의 『DB2 Multisystem에서 부속 조회 지원』
- 52 페이지의 『DB2 Multisystem에서의 액세스 계획』
- 52 페이지의 『DB2 Multisystem에서의 재사용가능한 열린 자료 경로(ODP)』
- 54 페이지의 『DB2 Multisystem에서의 임시 결과 출력기』
- 56 페이지의 『DB2 Multisystem에서의 Optimizer 메시지』
- 59 페이지의 『DB2 Multisystem에서의 CHGQRYA(조회 속성 변경) 명령 변경사항』
- 60 페이지의 『성능 고려사항 요약』

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 최적화 개요

분산 조회는 분산 레벨 및 로컬 레벨에서 최적화됩니다.

- 분산 레벨에서의 최적화는 조회를 가장 효율적인 단계로 나누어서 그 단계들을 처리할 노드를 결정하는 데 초점을 맞춘 것입니다.

분산 Optimizer는 분산 조회에 대해 고유합니다. 분산 Optimizer는 이 장에서 설명합니다.

- 로컬(단계) 레벨에서의 최적화는 비분산 환경에서 발생하는 최적화와 같습니다. 이것은 아마도 사용자가 익숙한 Optimizer입니다. 로컬 레벨 최적화는 이 장에 간략하게 설명되어 있습니다.

분산 최적화의 기본 가정은 각 자료 노드에 저장된 레코드 수는 대략 같으며 분산 조회의 모든 시스템 구성이 유사하다는 것입니다. 분산 Optimizer의 결정은 시스템과 조정 담당자 노드 시스템의 자료 통계를 기초로 합니다.

분산 조회에서 여러 단계를 요구할 경우, 임시 결과 파일이 사용됩니다. 임시 결과 파일은 특정 조회 단계의 결과를 포함시키기 위해 사용되는 시스템 작성 임시 파일(라이브러리 QRECOVERY에 저장됨)입니다. 임시 결과 파일의 내용은 다음 조회 단계에 대한 입력으로 사용됩니다.

DB2 Multisystem에서 단일 파일 조회 구현 및 최적화

단일 분산 파일을 조회하면 모든 노드나 분산 파일의 노드 서브세트의 자료가 조회됩니다. 이를 수행하기 위해, 조회가 지정된 시스템(조정 담당자 노드)은 조회를 보낼 파일의 노드를 판별합니다. 그 노드들은 조회를 실행하고 조회된 레코드를 조정 담당자 노드에 리턴합니다.

이 장에 있는 모든 예에서는 분산 파일로 DEPARTMENT 및 EMPLOYEE를 사용합니다. 이 파일들의 노드 그룹은 SYSA, SYSB 및 SYSC로 구성됩니다. 자료는 부서 번호에서 분할됩니다.

다음 SQL문은 DEPARTMENT 분산 파일을 작성합니다.

```
CREATE TABLE DEPARTMENT
  (DEPTNO CHAR(3) NOT NULL,
  DEPTNAME VARCHAR(20) NOT NULL,
  MGRNO CHAR(6),
  ADMRDEPT CHAR(3) NOT NULL)
  IN NODGRP1 PARTITIONING KEY(DEPTNO)
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

다음은 DEPARTMENT 표를 나타낸 것입니다. 표의 머리말로 노드, 레코드 번호, DEPTNO, DEPTNAME, MGRNO, ADMRDEPT가 사용되었습니다.

노드	레코드 번호	DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
SYSA	1	A00	Support Services	000010	A00
SYSB	2	A01	Planning	000010	A00
SYSC	3	B00	Accounting	000050	B00
SYSA	4	B01	Programming	000050	B00

다음 SQL문은 EMPLOYEE 분산 파일을 작성합니다.

```
CREATE TABLE EMPLOYEE
(EMPNO CHAR(6) NOT NULL,
FIRSTNME VARCHAR(12) NOT NULL,
LASTNAME VARCHAR(15) NOT NULL,
WORKDEPT CHAR(3) NOT NULL,
JOB CHAR(8),
SALARY DECIMAL(9,2)
IN NODGRP1 PARTITIONING KEY(WORKDEPT)
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

노드	레코드 번호	EMPNO	FIRSTNME	LASTNAME	WORK		
					DEPT	JOB	SALARY
SYSA	1	000010	Christine	Haas	A00	관리자	41250
SYSA	2	000020	Sally	Kwan	A00	사무원	25000
SYSB	3	000030	John	Geyer	A01	계획자	35150
SYSB	4	000040	Irving	Stern	A01	사무원	32320
SYSC	5	000050	Michael	Thompson	B00	관리자	38440
SYSC	6	000060	Eileen	Henderson	B00	회계사	33790
SYSA	7	000070	Jennifer	Lutz	B01	프로그래머	42325
SYSA	8	000080	David	White	B01	프로그래머	36450

다음에 오는 것은 위에서 정의된 분산 파일 EMPLOYEE를 사용하는 조회입니다. 이 때, 필드 SALARY에서 색인 EMPIDX가 작성됩니다. 조회는 SYSA에서 입력됩니다.

```
SQL: SELECT * FROM EMPLOYEE WHERE SALARY > 40000
```

```
OPNQRYF: OPNQRYF FILE((EMPLOYEE)) QRYSLT('SALARY > 40000')
```

이 경우, SYSA는 위의 조회를 SYSA를 포함하는 모든 EMPLOYEE 노드에 보냅니다. 각 노드는 조회를 실행하고 레코드들을 SYSA에 리턴합니다. 분산 색인이 파일 EMPLOYEE의 필드 SALARY에 존재하므로, 각 노드에서 수행되는 최적화에 의해 색인 사용 여부가 결정됩니다.

다음 예에서, 조회는 SYSA에 지정되어 있지만, 그 조회는 EMPLOYEE 파일이 존재하는 노드의 서브세트에 보내집니다. 이 경우에, 조회는 SYSA에서만 로컬로 실행됩니다.

```
SQL: SELECT * FROM EMPLOYEE WHERE WORKDEPT = 'A00'
```

```
OPNQRYF: OPNQRYF FILE((EMPLOYEE)) QRYSLT('WORKDEPT = 'A00')
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

분산 조회 Optimizer는 이 조회에 대한 분할 키 WORKDEPT를 포함하는 분리가능한 레코드 선택, WORKDEPT = 'A00'이 있음을 판별합니다. Optimizer는 'A00' 값을 해싱하고 이 해싱 값을 기초로 이 조건을 만족하는 모든 레코드가 위치될 노드를 찾습니다. 이 경우, 이 조건을 만족하는 모든 레코드는 SYSA에 있으므로, 조회는 그 노드로만 보내집니다. 조회는 SYSA에서 시작되었으므로, SYSA에서 로컬로 실행됩니다.

다음은 조회가 실행되는 여러 노드를 서브세트로 나눕니다.

- 분할 키의 모든 필드는 분리가능한 레코드 선택이어야 합니다.
- 모든 술어는 등호(=) 연산자를 사용해야 합니다.
- 분할 키의 모든 필드는 리터럴과 비교되어야 합니다.

주: 성능 상의 이유로, 조회가 특정 노드로 보내지도록 하려면 분할 키와 일치하는 레코드 선택 술어를 지정해야 합니다. 스칼라 함수 NODENAME, PARTITION 및 NODENUMBER가 포함된 레코드 선택도 조회를 특정 노드로 보낼 수 있습니다.

DB2 Multisystem에서 레코드 배열 구현 및 최적화

조회에서 순서를 지정할 경우, 각 노드가 병렬로 순서화를 수행할 수 있도록 조회와 함께 순서화 기준이 송신됩니다. 최종 병합이나 정렬이 조정 담당자 노드에서 수행되는지의 여부는 사용자가 지정하는 조회의 유형에 따라 결정됩니다.

각 노드로부터 수신된 순서화된 레코드를 병합하는 것이 최적입니다. 병합은 조정 담당자 노드에 의해 레코드가 수신되는 대로 발생합니다. 병합이 정렬에 비해 갖는 성능상의 주요 장점은 모든 노드로부터 모든 레코드를 정렬하지 않아도 레코드가 리턴될 수 있다는 것입니다.

각 노드로부터 수신된 순서화된 레코드들을 정렬하면 임의의 레코드가 리턴되기 전에 각 노드로부터의 모든 레코드가 읽히고 정렬되어 임시 결과 파일에 기록됩니다.

병합은 순서화가 지정되고 UNION과 최종 그룹화가 요구되지 않은 경우에 발생할 수 있습니다. 그렇지 않으면, 순서화 조회에 대해 최종 정렬이 조정 담당자 노드에서 수행됩니다.

ALWCPYDTA(자료 복사 허용) 매개변수는 분산 조회의 각 노드가 순서화 기준을 처리하는 방법에 영향을 미칩니다. ALWCPYDTA 매개변수는 OPNQRYP(조회 파일 열기) 및 STRSQL(SQL 시작) CL 명령에서 지정되고 CRTSQLxxx(SQLxxx 작성) 사전컴파일러 명령에서도 지정됩니다.

- ALWCPYDTA(*OPTIMIZE)는 각 노드가 정렬이나 색인을 사용하여 순서화 기준을 구현하는 것을 선택할 수 있게 합니다. 이 옵션은 최적입니다.

- OPNQRYP 명령 및 조회 API(QQQQRY)의 경우, ALWCPYDTA(*YES) 또는 ALWCPYDTA(*NO)는 각 노드가 지정된 순서화 필드와 정확히 일치하는 색인을 사용하게 합니다. 이것은 Optimizer가 로컬 파일에 대해 순서화를 처리하는 방법보다 더 제한적입니다.

DB2 Multisystem에서 UNION 및 DISTINCT 절 구현과 최적화

통합된 SELECT문이 분산 파일을 참조할 경우, 이 명령문은 분산 조회로 처리됩니다. 이 명령문의 처리는 병렬로 발생할 수 있습니다. 그러나, 통합된 각 SELECT의 레코드들은 통합 조사를 수행하기 위해 조정 담당자 노드로 다시 되돌려집니다. 이 경우, 통합 연산자들은 연속적으로 처리됩니다.

ORDER BY절이 통합 조회와 함께 지정될 경우, 각 노드로부터의 모든 레코드는 조정 담당자 노드에서 수신되어 임의의 레코드가 리턴되기 전에 정렬됩니다.

DISTINCT절이 분산 조회에 대해 지정될 경우, ORDER BY절을 추가하면 ORDER BY절을 지정하지 않은 경우보다 더 빨리 레코드가 리턴됩니다. DISTINCT를 ORDER BY와 함께 사용하면 각 노드가 레코드들을 병렬로 순서화할 수 있습니다. 조정 담당자 노드에서의 최종 병합은 각 노드로부터 순서화된 레코드를 읽어서 적절한 순서로 그 레코드들을 병합한 후 최종 정렬을 수행하지 않고 중복되는 레코드를 제거합니다.

DISTINCT절을 ORDER BY절 없이 지정할 경우, 각 노드로부터의 모든 레코드들은 정렬이 수행되는 조정 담당자 노드로 송신됩니다. 중복되는 레코드들은 정렬된 레코드들이 리턴되는 대로 제거됩니다.

DB2 Multisystem에서 DSTDTA 및 ALWCPYDTA 매개변수 처리

ALWCPYDTA(자료 복사 허용) 매개변수는 OVRDBF(데이터베이스 파일 대체) 명령의 DSTDTA(자료 분배) 매개변수에 대해 지정된 값을 변경할 수 있습니다.

대체 명령에서 실제 자료(DSTDTA(*CURRENT))를 사용하도록 지정하였으나 다음 중 어느 하나에 해당될 경우,

- 임시 사본이 요구되고 ALWCPYDTA(*YES)가 지정되었습니다
- 또는, 더 나은 성능을 위해 임시 사본이 선택되었고 ALWCPYDTA(*OPTIMIZE)가 지정되었습니다

DSTDTA가 DSTDTA(*BUFFERED)로 변경됩니다.

DSTDTA(*BUFFERED)가 사용자에게 대해 승인 가능하지 않고 조회에 임시 사본이 요구되지 않을 경우, ALWCPYDTA(*YES)를 지정하여 DSTDTA(*CURRENT)가 유효하도록 해야 합니다.

DB2 Multisystem에서 결합 구현 및 최적화

비분산 결합 조회에 대한 실행시 고려사항 외에, 분산 파일을 포함하는 조회에 대해 추가 실행시 고려사항이 있습니다. 결합은 자료가 파티션 호환가능일 경우에만 수행될 수 있습니다. 분산 조회 Optimizer는 노드 사이의 자료 이동을 수반할 수도 있는 자료 파티션을 호환가능하게 하는 계획을 생성합니다.

자료는 양쪽 파일의 분할 키에 있는 자료가 같은 노드 그룹을 사용하여 같은 노드로 해싱할 경우 파티션 호환가능합니다. 예를 들어, 큰 정수 필드나 작은 정수 필드에 저장된 동일한 숫자 값이 같은 값으로 해싱됩니다.

다음 자료 유형은 파티션 호환가능합니다.

- 큰 정수(4바이트), 작은 정수(2바이트), 팩형 10진수 및 존(zone) 숫자.
- 고정 길이 및 가변 길이 SBCS 문자와 DBCS 개방, DBCS 선택 또는 DBCS 전용.
- 고정 길이 및 가변 길이 그래픽.

날짜, 시간, 시간소인 및 부동 소수점 숫자 자료 유형은 분할 키가 될 수 없기 때문에 파티션 호환가능하지 않습니다.

분산 파일이 관련되는 결합은 네 가지 유형으로 분류됩니다. 네 가지의 유형은 상호 위치된 결합, 경로가 지시된 결합, 재분할된 결합 및 브로드캐스트 결합입니다. 다음 섹션들에서 결합 유형을 정의하고 다른 결합 유형에 대한 예를 제공합니다.

- 『DB2 Multisystem에서 상호 위치된 결합』
- 45 페이지의 『DB2 Multisystem에서 경로가 지시된 결합』
- 46 페이지의 『DB2 Multisystem에서 재분할된 결합』
- 48 페이지의 『DB2 Multisystem에서 브로드캐스트 결합』

또한 48 페이지의 『DB2 Multisystem에서 결합 최적화』에서 분산 조회 Optimizer가 결합(join)을 처리하는 방식에 관한 정보를 알 수 있습니다.

DB2 Multisystem에서 상호 위치된 결합

상호 위치된 결합에서는, 결합될 파일들의 해당 레코드가 같은 노드에 존재합니다. 결합될 파일들의 분할 키 값은 파티션 호환가능합니다. 결합을 수행하기 위해 어떤 자료도 다른 노드로 이동될 필요가 없습니다. 이 방법은 분할 키의 모든 필드가 결합 필드이고 결합 연산자가 =(등호)인 조회에 대해서만 유효합니다. 또한, 첫 번째 파일의 분할 키에 대한 n번째 필드(분할 키의 필드 수에 대해 n=1인)는 두 번째 파일의 분할 키에 대한 n번째 필드에 결합되고, n번째 필드의 자료 유형은 파티션 호환가능해야 합니다. 분할 키의 모든 필드는 결합에 관련되어야 합니다. 분할 키의 필드들을 포함하지 않는 추가 결합 술어는 상호 위치된 결합을 수행하는 데 영향을 주지 않습니다.

다음 예에서, 결합 술어가 두 파일의 분할 키 필드들을 포함하고 필드들이 파티션 호환 가능하므로, 상호 위치된 결합이 수행될 수 있습니다. 즉, DEPTNO 및 WORKDEPT 의 일치되는 값이 같은 노드에 위치됩니다.

```
SQL:      SELECT DEPTNAME, FIRSTNME, LASTNAME
           FROM DEPARTMENT, EMPLOYEE
           WHERE DEPTNO=WORKDEPT
```

```
OPNQRYF: OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(JOINFMT)
           JFLD((DEPTNO WORKDEPT *EQ))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

이 조회에 의해 리턴되는 레코드들:

다음은 조회 결과를 나타낸 것입니다. 이 표의 머리말에는 DEPTNAME, FIRSTNAME, LASTNAME이 사용되었습니다.

DEPTNAME	FIRSTNME	LASTNAME
Support Services	Christine	Haas
Support Services	Sally	Kwan
Planning	John	Geyer
Planning	Irving	Stern
Accounting	Michael	Thompson
Accounting	Eileen	Henderson
Programming	Jennifer	Lutz
Programming	David	White

다음 예에서, 추가 결합 술어 MGRNO=EMPNO는 분할 키가 여전히 결합 술어에 포함되므로 상호 위치된 결합을 수행하는 데 영향을 주지 않습니다.

```
SQL:      SELECT DEPTNAME, FIRSTNME, LASTNAME
           FROM DEPARTMENT, EMPLOYEE
           WHERE DEPTNO=WORKDEPT AND MGRNO=EMPNO
```

```
OPNQRYF: OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(JOINFMT)
           JFLD((DEPTNO WORKDEPT *EQ) (MGRNO EMPNO *EQ))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

이 조회에 의해 리턴되는 레코드들:

DEPTNAME	FIRSTNME	LASTNAME
Support Services	Christine	Haas
Accounting	Michael	Thompson

DB2 Multisystem에서 경로가 지시된 결합

경로가 지시된 결합에서는, 파일 중 최소한 하나의 분할 키가 결합 필드로 사용됩니다. 이 결합 필드는 다른 파일의 분할 키와 대응하지 않습니다. 한 파일의 레코드들이 두 번째 파일의 파티션 맵과 노드 그룹을 사용하여 결합 필드 값의 해싱을 기초로 두 번째

파일의 노드로 보내집니다. 레코드들이 임시 분산 파일을 통해 두 번째 파일의 노드로 이동되고 나면, 상호 위치된 결합이 사용되어 자료를 결합합니다. 이 방법은 분할 키의 모든 필드가 파일들 중 최소한 하나에 대한 결합 필드인 equijoin 조회에 대해서만 유효합니다.

다음 조회에서, 결합 필드 (WORKDEPT)는 파일 EMPLOYEE에 대한 분할 키입니다. 그러나, 결합 필드 (ADMRDEPT)는 DEPARTMENT에 대한 분할 키가 아닙니다. 자료를 이동하지 않고 결합을 시도한 경우, DEPARTMENT의 레코드 2가 EMPLOYEE의 레코드 1과 2에 결합되어야 하고 이 레코드들이 서로 다른 노드에 저장되어야 하므로, 결과 레코드가 누락되게 됩니다.

```
SQL:      SELECT DEPTNAME, FIRSTNME, LASTNAME
           FROM DEPARTMENT, EMPLOYEE
           WHERE ADMRDEPT = WORKDEPT AND JOB = 'Manager'
```

```
OPNQRYF:  OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(JOINFMT)
           QRYSLT('JOB *EQ 'Manager')
           JFLD((ADMRDEPT WORKDEPT *EQ))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

조회를 실행하기 위해 필요한 DEPARTMENT의 레코드들이 읽히고, ADMRDEPT의 자료가 분할 맵과 노드 그룹 EMPLOYEE를 사용하여 해싱됩니다. 다음과 같은 임시 파일이 작성됩니다.

이전 노드	신규 노드	DEPTNAME	ADMRDEPT(신규 분할 키)
SYSA	SYSA	Support Services	A00
SYSB	SYSA	Planning	A00
SYSC	SYSC	Accounting	B00
SYSA	SYSC	Programming	B00

이 임시표는 다음을 생성하기 위해 EMPLOYEE에 결합됩니다. 결합은 ADMRDEPT가 WORKDEPT와 파티션 호환가능하므로 작동됩니다.

DEPTNAME	FIRSTNME	LASTNAME
Support Services	Christine	Haas
Planning	Christine	Haas
Accounting	Michael	Thompson
Programming	Michael	Thompson

DB2 Multisystem에서 재분할된 결합

재분할된 결합에서는, 파일의 분할 키가 결합 필드로 사용되지 않습니다. 두 파일의 레코드들은 각 파일의 결합 필드 값을 해싱하여 이동해야 합니다. 파일의 분할 키 필드

중 어느 것도 결합 기준에 포함되지 않으므로, 파일들은 하나 이상의 결합 필드를 포함하는 신규 분할 키에서 해싱하여 재분할되어야 합니다. 이 방법은 equijoin 조회에 대해서만 유효합니다.

```
SQL:      SELECT DEPTNAME, FIRSTNME, LASTNAME
           FROM DEPARTMENT, EMPLOYEE
           WHERE MGRNO = EMPNO
```

```
OPNQRYF: OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(JOINFMT)
           JFLD((MGRNO EMPNO *EQ))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

이 예에서, 자료는 MGRNO와 EMPNO 모두 분할 키가 아니므로 재분산되어야 합니다.

DEPARTMENT로부터의 자료가 재분산됩니다.

이전 노트	신규 노트	DEPTNAME	MGRNO(신규 분할 키)
SYSA	SYSB	Support Services	000010
SYSB	SYSB	Planning	000010
SYSC	SYSC	Accounting	000050
SYSA	SYSC	Programming	000050

EMPLOYEE로부터의 자료가 재분산됩니다.

이전 노트	신규 노트	FIRSTNME	LASTNAME	EMPNO(신규 분할 키)
SYSA	SYSB	Christine	Haas	000010
SYSA	SYSC	Sally	Kwan	000020
SYSB	SYSA	John	Geyer	000030
SYSB	SYSB	Irving	Stern	000040
SYSC	SYSC	Michael	Thompson	000050
SYSC	SYSA	Eileen	Henderson	000060
SYSA	SYSB	Jennifer	Lutz	000070
SYSA	SYSC	David	White	000080

이 조회에 의해 리턴되는 레코드들:

DEPTNAME	FIRSTNME	LASTNAME
Support Services	Christine	Haas
Planning	Christine	Haas
Accounting	Michael	Thompson
Programming	Michael	Thompson

DB2 Multisystem에서 브로드캐스트 결합

브로드캐스트 결합에서는, 한 파일의 선택된 모든 레코드가 결합 수행 이전에 다른 파일의 모든 노드로 송신 또는 브로드캐스트됩니다. 이것은 equijoin 이외의 모든 조화에 대해 사용되는 결합 방법입니다. 이 방법은 또한 결합 기준에 날짜, 시간, 시간소인 또는 부동 소수점 숫자와 같은 자료 유형을 갖는 필드가 사용될 경우에도 사용됩니다.

다음 예에서, 분산 조회 Optimizer는 선택사항 JOB = 'Manager'를 적용하면 더 작은 레코드 세트가 브로드캐스트되므로 EMPLOYEE를 브로드캐스트할 것을 결정합니다. 노드 그룹의 각 노드에 있는 임시 파일에는 선택된 모든 레코드들이 있습니다. (레코드들은 각 노드에서 중복됩니다.)

```
SQL:      SELECT DEPTNAME, FIRSTNME, LASTNAME
           FROM DEPARTMENT, EMPLOYEE
           WHERE DEPTNO <> WORKDEPT AND JOB = 'Manager'
```

```
OPNQRYF:  OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(JOINFMT)
           QRYSLT('JOB *EQ 'Manager')
           JFLD((DEPTNO WORKDEPT *NE))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

분산 조회 Optimizer는 다음의 선택된 두 개의 레코드를 각 노드로 송신합니다.

이전 노드	신규 노드	FIRSTNME	LASTNAME	WORKDEPT
SYSA	SYSA, SYSB, SYSC	Christine	Haas	A00
SYSC	SYSA, SYSB, SYSC	Michael	Thompson	B00

이 조화에 의해 리턴되는 레코드들:

DEPTNAME	FIRSTNME	LASTNAME
Support Services	Michael	Thompson
Planning	Christine	Haas
Planning	Michael	Thompson
Accounting	Christine	Haas
Programming	Christine	Haas
Programming	Michael	Thompson

DB2 Multisystem에서 결합 최적화

분산 조회 Optimizer는 분산 파일을 결합하기 위한 계획을 생성합니다. 분산 조회 Optimizer는 파일 크기, 각 파일에 대해 선택된 예상되는 레코드 수 및 가능한 분산 결합의 유형을 찾은 후에, 조회를 여러 단계로 구분합니다. 각 단계에서 다음 단계에 대한 입력으로 사용될 중간 결과 파일이 작성됩니다.

최적화시, 분산 결합의 유형을 기초로 각 결합 단계에 대한 비용이 계산됩니다. 비용은 부분적으로 해당 결합 단계에 대해 필요한 자료 이동량에 반영됩니다. 비용은 최종 분산 계획을 판별하는 데 사용됩니다.

각 단계에서 될 수 있는 한 많은 처리가 완료됩니다. 예를 들어, 주어진 단계에 대해 분리된 레코드 선택사항은 해당 단계에서 수행되고, 가능한 한 많은 파일이 각 단계에 대해 결합됩니다. 각 결합 단계에는 한 가지 이상의 분산 결합 유형이 관련될 수 있습니다. 상호 위치된 결합 및 경로가 지시된 결합은 필요한 파일을 먼저 지시하여 상호 위치된 결합 하나로 통합될 수 있습니다. 경로가 지시된 결합과 재분할된 결합은 먼저 모든 파일을 지시하고 결합을 수행하여 통합될 수 있습니다. 경로가 지시된 결합과 재분할된 결합은 실제로 하나 이상의 파일이 결합 발생 이전에 지시되는 상호 위치된 결합입니다.

분산 파일을 로컬 파일과 결합할 경우, 분산 조회 Optimizer는 분산 파일 결합시 계산되는 비용과 마찬가지로 비용을 계산합니다. 이 비용을 기초로, 분산 조회 Optimizer는 다음 중 하나를 수행할 것을 선택할 수 있습니다.

- 모든 로컬 파일을 분산 파일의 자료 노드로 브로드캐스트한 후 상호 위치된 결합을 수행합니다.
- 모든 로컬 및 분산 파일을 가장 큰 분산 파일의 자료 노드로 브로드캐스트한 후 상호 위치된 결합을 수행합니다.
- 분산 파일이 조정 담당자 노드로 돌아가도록 지시하고 그곳에서 결합을 수행합니다.

결합(join) 필드에서의 분할 키에 관해 알려면 『DB2 Multisystem에서 결합 필드에 대한 분할 키』를 참조하십시오.

DB2 Multisystem에서 결합 필드에 대한 분할 키

결합 유형에 대한 이전 섹션에서, 상호 위치된 결합을 제외한 모든 분산 결합 유형에 대해 자료 이동이 필요하다는 것을 알 수 있습니다. 자료 이동의 필요성을 없애고 성능을 최대화하려면, 상호 위치된 결합이 가능하도록 모든 조회를 작성해야 합니다. 다시 말해서, 분산 파일의 분할 키가 파일을 결합하는 데 함께 사용되는 필드들과 일치해야 합니다. 자주 실행되는 조회의 경우, 분할 키가 순서화 또는 그룹화 기준과 일치하는 것보다 결합 필드와 일치하도록 하는 것이 더 중요합니다.

DB2 Multisystem에서 그룹화 구현 및 최적화

분산 파일을 사용하는 조회에서 그룹화에 대한 구현 방법은 분할 키가 그룹화 기준에 포함되는지 여부에 따라 결정됩니다. 그룹화는 1단계 그룹화 또는 2단계 그룹화를 사용하여 구현됩니다.

- 50 페이지의 『DB2 Multisystem에서의 1단계 그룹화』
- 50 페이지의 『DB2 Multisystem에서의 2단계 그룹화』

그룹화와 결합을 사용하는 것에 대해 알려면 51 페이지의 『DB2 Multisystem에서 그룹화 및 결합』을 참조하십시오.

DB2 Multisystem에서의 1단계 그룹화

분할 키로부터의 모든 필드가 GROUP BY 필드이면, 1단계 그룹화를 사용하여 그룹화를 수행할 수 있습니다. 그룹에 대한 모든 자료가 같은 노드에 있기 때문입니다. 다음은 1단계 그룹화의 예입니다.

```
SQL:      SELECT WORKDEPT, AVG(SALARY)
           FROM EMPLOYEE
           GROUP BY WORKDEPT
```

```
OPNQRYF:  OPNQRYF FILE((EMPLOYEE)) FORMAT(GRPFMT)
           GRPFLD(WORKDEPT)
           MAPFLD((AVGSAL '%AVG(SALARY)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

WORKDEPT는 분할 키이며 그룹화 필드이므로, WORKDEPT의 모든 유사 값들은 같은 노드에 있습니다. 예를 들어, A00의 모든 값은 SYSA에 있고, A01의 모든 값은 SYSB에 있으며, B00의 모든 값은 SYSC에 있습니다. 그리고 B01의 모든 값은 SYSA에 있습니다. 그룹화는 모든 세 개의 노드에서 병렬로 수행됩니다.

1단계 그룹화를 구현하려면, 분할 키의 모든 필드가 그룹화 필드이어야 합니다. 추가 비분할 키 필드들도 그룹화 필드가 될 수 있습니다.

DB2 Multisystem에서의 2단계 그룹화

분할 키가 그룹화 필드에 포함되지 않으면, 필드의 유사값들이 같은 노드에 위치되지 않으므로 2단계 그룹화를 사용하여 그룹화를 수행해야 합니다. 다음은 2단계 그룹화의 예입니다.

```
SQL:      SELECT JOB, AVG(SALARY)
           FROM EMPLOYEE
           GROUP BY JOB
```

```
OPNQRYF:  OPNQRYF FILE((EMPLOYEE)) FORMAT(GRPFMT2)
           GRPFLD(JOB)
           MAPFLD((AVGSAL '%AVG(SALARY)'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

이 예에서, JOB이 'Clerk'인 그룹의 경우 'Clerk'의 값이 EMPLOYEE 분산 파일에서 서로 다른 두 개의 노드에 있다는 점에 유의하십시오. 그룹화는 세 개의 모든 노드에서 병렬로 그룹화를 먼저 수행함으로써 구현됩니다. 그 결과, 초기 그룹화는 조정 담당자 노드에서 임시 파일에 위치됩니다. 조회가 수정되고, 최종 그룹화 결과 세트를 얻기 위해 조정 담당자 노드에서 다시 그룹화가 실행됩니다.

전체 파일 그룹화(필드별 그룹이 아님)는 항상 2단계를 사용하여 구현됩니다.

조회에 HAVING절이나 OPNQRYF 명령의 GRPSLT(그룹 선택 표현식) 매개변수가 포함될 경우, 첫 번째 그룹화 단계로부터의 모든 그룹은 조정 담당자 노드로 리턴됩니다. 그러면, HAVING절이나 GRPSLT 매개변수가 두 번째 그룹화 단계의 부분으로 처리됩니다.

조회에 DISTINCT 열(총계) 함수가 포함되어 있고 2단계 그룹화가 요구되는 경우, 첫 번째 단계에서 어떤 그룹화도 수행되지 않습니다. 그 대신, 모든 레코드가 조정 담당자 노드로 리턴되고, 모든 그룹화가 두 번째 단계의 부분으로 조정 담당자 노드에서 실행됩니다.

DB2 Multisystem에서 그룹화 및 결합

조회에 결합이 있는 경우, 구현될 수 있는 그룹화 유형을 판별하기 위해 사용되는 분할 키는 결합을 구현하기 위해 요구되었던 자료의 재분할을 기초로 합니다.

다음 예에서, 재분할된 결합은 그룹화 이전에 수행되므로 신규 분할 키인 MGRNO가 생성됩니다. 이제는 MGRNO가 분할 키이므로, 그룹화는 1단계 그룹화를 사용하여 수행될 수 있습니다.

```
SQL:      SELECT MGRNO, COUNT(*)
           FROM DEPARTMENT, EMPLOYEE
           WHERE MGRNO = EMPNO
           GROUP BY MGRNO
```

```
OPNQRYF:  OPNQRYF FILE((DEPARTMENT) (EMPLOYEE)) FORMAT(GRPFMT2)
           JFLD((MGRNO EMPNO *EQ))
           GRPFLD(MGRNO)
           MAPFLD((CNTMGR '%COUNT'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.


다음 예에서, 재분할된 결합은 그룹화 이전에 수행되므로 신규 분할 키인 EMPNO가 생성됩니다. 이제는 EMPNO가 WORKDEPT 대신 분할 키이므로, 그룹화는 1단계 그룹화를 사용하여 수행될 수 없습니다.

```
SQL:      SELECT WORKDEPT, COUNT(*)
           FROM DEPARTMENT, EMPLOYEE
           WHERE MGRNO = EMPNO
           GROUP BY WORKDEPT
```

```
OPNQRYF:  OPNQRYF FILE((DEPARTMENT) (EMPLOYEE)) FORMAT(GRPFMT3)
           JFLD((MGRNO EMPNO *EQ))
           GRPFLD(WORKDEPT)
           MAPFLD((CNTDEPT '%COUNT'))
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

DB2 Multisystem에서 부속 조회 지원

분산 파일을 부속 조회에 지정할 수 있습니다. 부속 조회는 자체의 고유한 탐색 조건을 포함할 수 있으며, 이 탐색 조건들은 또 부속 조회를 포함할 수 있습니다. 그러므로, SQL 문에 부속 조회 계층이 포함될 수 있습니다. 부속 조회를 포함하는 계층의 요소들은 포함하는 부속 조회들보다 높은 레벨에 있습니다. 부속 조회 사용에 대한 자세한 정보에 대해서는 SQL 프로그래밍 개념  주제를 참조하십시오.

DB2 Multisystem에서의 액세스 계획

분산 파일을 참조하는 조회들에 대해 저장된 액세스 계획은 로컬 파일에 저장된 액세스 계획과 다릅니다. 분산 조회에 대해 저장된 액세스 계획은 조회가 여러 단계로 분할되는 방법과 조회의 각 단계가 실행되는 노드에 관한 정보를 포함하는 분산 액세스 계획입니다. 각 노드에서 단계가 로컬로 구현되는 방법에 관한 내용은 액세스 계획에 저장되지 않습니다. 이 정보는 실행 시간에 구축됩니다.

DB2 Multisystem에서의 재사용가능한 열린 자료 경로(ODP)

재사용가능한 열린 자료 경로(ODP)에는 분산 조회에 대한 특별 고려사항이 있습니다. 분산 조회의 대부분의 다른 측면들처럼, ODP에는 분산과 로컬이라는 두 레벨이 있습니다.

분산 ODP는 조정 ODP입니다. 분산 ODP는 조회를 사용자에게 연관시켜서 로컬 ODP를 제어합니다. 로컬 ODP는 조회에 관련된 각 시스템에 위치되어 있으며, 분산 ODP를 통해 요구합니다.

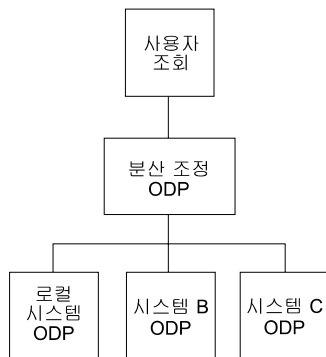


그림 7. 열린 자료 경로(ODP)의 예

예를 들어, SQL FETCH를 수행하기 위한 요구가 있는 경우, 분산 ODP에 대해 요구됩니다. 그러면, 시스템은 해당 요구를 취하여 로컬 ODP에 대해 적절한 레코드 검색을 수행합니다.

분산 조회로, 분산 ODP에 대해 재사용가능하지만, 아직 하나 이상의 로컬 ODP에 대해서는 재사용할 수 없습니다. 그러나, 분산 조회 ODP가 재사용가능하지 않으면, 로컬 ODP는 항상 재사용가능하지 않습니다. 이것은 다음을 위해 허용됩니다.

- 각 로컬 시스템은 해당되는 로컬 ODP를 여는 최상의 방법을 결정할 수 있습니다(재사용가능 대 재사용불가능).
- 로컬 ODP 방법에 관계 없이, 분산 ODP는 통신과 같은 활동 자원을 유지보수하기 위해 될 수 있는 한 많이 재사용가능하도록 열릴 수 있습니다.

시스템은 기본적인 로컬 ODP가 재사용가능하지 않을 경우에도 가능할 때마다 분산 ODP를 재사용가능하게 만들려고 합니다. 이러한 상황이 발생할 경우, 시스템은 다음과 같이 ODP 화면정리를 처리합니다.

- 각 로컬 ODP를 통해 순환합니다.
- 재사용가능 로컬 ODP의 화면정리를 수행합니다.
- 『하드』 닫기 및 재사용불가능 ODP의 다시 열기를 수행합니다.
- 필요한 나머지 분산 ODP 자체의 화면정리를 완료합니다.

분산 ODP는 로컬 ODP를 재사용할 수 없게 만드는 것(LIKE절의 변수나, index-from-index 작성이 수행될 수 있도록 재사용가능하지 않음을 Optimizer가 선택) 중 하나에 의해 영향을 받지 않으므로, 로컬 ODP보다 더 자주 재사용가능하게 됩니다. 분산 ODP를 재사용가능하게 하지 않는 경우는 로컬 ODP에 영향을 미치는 것들의 서브세트입니다. 이 서브세트에는 다음이 포함됩니다.

- 정렬 이외의 목적으로 임시 파일 사용. 이들을 다단계 분산 조회라고 하며, Optimizer 디버그 메세지 CPI4343 신호가 이 경우들에 대해 보내집니다.
- 조회되는 파일들에 영향을 줄 수 있는 라이브러리 리스트 변경사항.
- 조회되는 파일에 영향을 줄 수 있는 OVRDBF 변경사항.
- 특별 레지스터 USER 또는 CURRENT TIMEZONE에 대한 값 변경사항.
- 작업 CCSID 변경사항.
- 발행되는 RCLRSC(자원 재생) 명령.

로컬 ODP의 재사용가능성은 비분산 조회 ODP에 대해 이미 존재하는 일부 상황에 의해 영향을 받습니다. 그러므로, 로컬 조회 ODP에 적용되는 것과 같은 고려사항이 적용됩니다.

DB2 Multisystem에서의 임시 결과 출력기

임시 결과 출력기(임시 출력기)는 항상 활동 상태인 시스템에서 개시된 작업입니다. 시스템에서 임시 출력기는 QQQTEMP1 및 QQQTEMP2 작업의 쌍입니다. 임시 출력기는 조회를 실행중인 작업들로부터의 요구를 처리합니다. 이 요구들은 실행할 조회(조회 단계의)와 조회 결과로부터 채울 시스템 임시 파일의 이름으로 구성됩니다. 임시 출력기는 요구를 처리하고 임시 파일을 채웁니다. 그 다음에, 이 중간 임시 파일이 요구 작업에 사용되어 원래의 조회를 완료합니다.

다음 예는 임시 결과 출력기를 필요로 하는 조회와 그 조회를 처리하기 위해 필요한 단계들을 나타내고 있습니다.

```
SQL:      SELECT COUNT(*)
           FROM DEPARTMENT a, EMPLOYEE b
           WHERE a.ADMRDEPT = b.WORKDEPT
           AND b.JOB = 'Manager'

OPNQRYF:  OPNQRYF FILE((DEPARTMENT) (EMPLOYEE))
           FORMAT(FMTFILE)
           MAPFLD((CNTFLD '%COUNT'))
           JFLD((1/ADMRDEPT 2/WORKDEPT))
           QRYSLT('2/JOB = 'Manager')
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

WORKDEPT는 EMPLOYEE의 분할 키이지만, ADMRDEPT는 DEPARTMENT의 분할 키가 아닙니다. 조회는 두 단계에서 처리되어야 하므로, Optimizer는 조회를 다음 단계들로 분할합니다.

```
INSERT INTO SYS_TEMP_FILE
SELECT a.DEPTNAME, a.ADMRDEPT
FROM DEPARTMENT a
```

그리고

```
SELECT COUNT(*) FROM SYS_TEMP_FILE x, EMPLOYEE b
WHERE x.ADMRDEPT = b.WORKDEPT AND b.JOB = 'Manager'
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

작업에 대해 임시 출력기가 허용될 경우(CHGQRYA(조회 속성 변경) 옵션에 의해 제어됨), Optimizer는 다음을 수행합니다.

1. 임시 파일(SYS_TEMP_FILE)을 라이브러리 QRECOVERY에 작성합니다.
2. SYS_TEMP_FILE을 채우는 요구를 임시 출력기로 송신합니다.
3. 계속해서 최종 조회 열기를 완료합니다(임시 출력기가 임시 파일을 채우는 동안).
4. 최종 조회가 열리면, 제어를 해당되는 호출자에게 넘기기 전에 임시 출력기가 임시 파일 채우기를 완료할 때까지 기다립니다.

이 명령에 대한 내용은 59 페이지의 『DB2 Multisystem에서의 CHGQRYA(조회 속성 변경) 명령 변경사항』을 참조하십시오.

DB2 Multisystem과 함께 임시 결과 출력기를 사용하는 것에 관해 알려면 다음을 참조하십시오.

- 『임시 출력기 작업: DB2 Multisystem에서의 장점』
- 『임시 출력기 작업: DB2 Multisystem에서의 단점』
- 56 페이지의 『DB2 Multisystem에서의 임시 출력기 제어』

임시 출력기 작업: DB2 Multisystem에서의 장점

요구 처리시 임시 출력기를 사용할 경우의 장점은 기본 작업이 조회를 다른 단계를 처리하는 동안 동시에(병렬로) 임시 출력기가 해당되는 요구를 처리할 수 있는 것입니다.

다음은 임시 출력기를 사용할 경우 제공되는 몇 가지 성능상의 장점입니다.

- 기본 작업이 쉽게 병렬 처리할 수 없는 더 복잡한 단계(예: 조회 최적화, 색인 분석)에 대해 계속될 수 있는 반면, 임시 출력기는 해당되는 조회 단계 완료시 완전한 SMP 병렬 지원을 사용할 수 있습니다.
- 분산 파일 처리에서 통신 상호 작용을 요구하므로, 송수신에 대해 대기하면서 보통 많은 시간이 소요되어 임시 출력기로 오프로드될 수 있어서, 다른 작업을 수행하기 위해 기본 작업을 떠나게 됩니다.

임시 출력기 작업: DB2 Multisystem에서의 단점

임시 출력기에는 조회에 대한 유용성을 판별할 때 고려해야 하는 단점도 있습니다.

- 임시 출력기는 별도의 작업입니다. 따라서, 다음과 같이 기본 작업과의 충돌이 발생할 수 있습니다.

- 기본 작업에 자체에 대해 잠긴 파일이 있을 수 있습니다. 이 경우, 임시 출력기는 파일에 액세스할 수 없어서 해당되는 조회 단계를 완료할 수 없습니다.
- 기본 작업이 확약 제어 하에 분산 파일을 작성하였으나 아직 그 작성을 확약하지 않았을 수 있습니다. 이 경우, 임시 출력기는 파일에 액세스할 수 없습니다.
- 임시 출력기에 기본 작업과 같은 방법으로 처리할 수 없는 상황이 발생할 수 있습니다. 예를 들어, 조회 메시지 신호가 보내진 경우, 기본 작업이 메시지를 무시하고 계속할 것을 선택할 수 있는 반면, 임시 출력기는 취소해야 할 수도 있습니다.
- 임시 출력기는 시스템의 모든 작업에 의해 공유됩니다. 몇 개의 작업이 임시 출력기에 대한 요구를 갖고 있는 경우, 출력기가 그 요구를 처리하려고 하는 동안 요구가 대기행렬에 놓일 수 있습니다.

주: 시스템은 세 개의 사용 중인 임시 출력기 작업 쌍과 함께 적재됩니다.

- 임시 출력기가 관여할 경우(조회 단계가 별도의 작업에서 실행되므로), 조회를 분석하려는 시도(예를 들어, 디버그 메시지를 통해)가 복잡해질 수 있습니다.

주: 시스템은 확약 제어 *CS 또는 *ALL 하에서 실행되는 조회에 임시 출력기가 사용될 수 없게 합니다. 왜냐하면 기본 작업이 파일에서 잠긴 레코드들의 밖에서 임시 출력기가 잠גיע 하는 그러한 레코드들을 갖고 있어서 완료할 수 없기 때문입니다.

DB2 Multisystem에서의 임시 출력기 제어

기본적으로, 조회에서는 임시 출력기가 사용되지 않습니다. 그러나, 임시 출력기 사용은 CHGQRYA(조회 속성 변경) 명령을 사용하여 작동할 수 있게 됩니다.

CHGQRYA 명령의 ASYNCJ(비동기 작업 사용) 매개변수는 임시 출력기를 제어하기 위해 사용됩니다. ASYNCJ 매개변수에는 다음과 같은 적용가능한 옵션들이 있습니다.

- *DIST 또는 *ANY는 임시 출력기 작업이 분산 파일을 포함하는 데이터베이스 조회에 사용될 수 있게 합니다.
- *LOCAL 또는 *NONE은 임시 출력기가 분산 파일의 조회에 사용되지 못하게 합니다.

DB2 Multisystem에서의 Optimizer 메시지

OS/400® 분산 조회 Optimizer는 작업이 디버그 모드에 있을 때 현재 조회 처리에 관한 정보 메시지를 제공합니다. 분산 조회가 처리된 방법을 보여주는 이 메시지는 기존의 Optimizer 메시지 이외의 메시지입니다. 이 메시지들은 OPNQRYP(조회 파일 열기) 명령, DB2 UDB 조회 관리자 및 SQL 개발 킷, 대화식 SQL, 삽입된 SQL 및 iSeries 고급 언어(HLL)에 표시됩니다. 모든 메시지는 작업 로그에 표시됩니다. 작업을 디버그 모드에 두기만 하면 됩니다.

데이터베이스 관리자가 작업 기록부에 기록한 정보용 메시지를 사용하여 분산 조회의 실행을 평가할 수 있습니다. 데이터베이스 관리자는 다음과 같은 분산 메시지 또는 적절할 경우 기존의 Optimizer 메시지를 송신할 수 있습니다. 앰퍼샌드(&) 변수(&1, &X)는 메시지가 작업 기록부에 표시될 때 다른 대체 값이나 오브젝트명을 포함하는 대체 변수입니다.

- CPI4341 분산 조회 수행 중.
- CPI4342 조회에 대한 분산 결합 수행 중.
- CPI4343 분산 조회 단계 &2 중 &1에 대한 Optimizer 디버그 메시지.
- CPI4345 임시 분산 결과 파일 &4가 조회에 대해 구축됨.

이 메시지들은 분산 조회가 실행되는 방법에 대한 피드백을 제공합니다. 어떤 경우에는, 조회가 더 빠르게 실행되도록 돕기 위한 개선을 나타냅니다. 메시지에 대한 원인 및 사용자 응답은 아래에서 부연 설명되어 있습니다. 실제 메시지 도움말은 더 복잡하므로 각 메시지의 의미와 응답을 판별해야 할 경우에만 사용해야 합니다.

각 메시지에 대한 자세한 설명은 다음과 같습니다.

CPI4341

분산 조회 수행 중.

이 메시지는 단일 분산 파일이 조회되었으나 여러 단계에서 처리되지 않았음을 나타냅니다. 이 메시지는 조회가 실행된 파일의 노드를 나열합니다.

CPI4342

조회에 대한 분산 결합 수행 중.

이 메시지는 분산 결합이 발생하였음을 나타냅니다. 이 메시지는 또한 결합이 실행된 노드와 함께 결합된 파일들도 나열합니다.

CPI4343

분산 조회 단계 &2 중 &1에 대한 Optimizer 디버그 메시지.

이 메시지는 분산 조회가 여러 단계에서 처리되었음을 나타내고 현재 단계 번호를 나열합니다. 이 메시지 다음에는 해당 단계에 대한 모든 Optimizer 메시지들이 나옵니다.

CPI4345

임시 분산 결과 파일 &4가 조회에 대해 구축됨.

이 메시지는 임시 분산 결과 파일이 작성되었음을 나타내고 임시 파일이 요구되었던 원인으로 이유 코드를 나열합니다. 이 메시지는 또한 임시 파일이 작성된 파일과 노드를 작성하기 위해 사용되었던 분할 키도 나타내고 있습니다.

다음 예는 분산 조회가 처리된 방법을 판별하기 위해 생성된 분산 Optimizer 메시지를 살펴보는 방법을 나타내고 있습니다. 예에서는 분산 파일로 EMPLOYEE 및 DEPARTMENT가 사용됩니다.

```

SQL:      SELECT A.EMPNO, B.MGRNO, C.MGRNO, D.EMPNO
          FROM   EMPLOYEE A, DEPARTMENT B, DEPARTMENT C, EMPLOYEE D
          WHERE  A.EMPNO=B.MGRNO
                AND B.ADMRDEPT=C.DEPTNO
                AND C.DEPTNO=D.WORKDEPT

OPNQRYF:  OPNQRYF FILE((EMPLOYEE) (DEPARTMENT) (DEPARTMENT) (EMPLOYEE))
          FORMAT(JFMT)
          JFLD((1/EMNO 2/MGRNO *EQ)
              (2/ADMRDEPT 3/DEPTNO)
              (3/DEPTNO 4/WORKDEPT))

```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

다음과 같은 분산 Optimizer 메시지 리스트가 생성됩니다.

- CPI4343 분산 조회 단계 4중 1에 대한 Optimizer 디버그 메시지는 다음과 같음:
 - CPI4345 임시 분산 결과 파일 QQTDF0001이 조회에 대해 구축됨.
파일 B가 임시 파일 *QQTDF0001로 보내졌습니다.
- CPI4343 분산 조회 단계 4중 2에 대한 Optimizer 디버그 메시지는 다음과 같음:
 - CPI4342 조회에 대한 분산 결합 수행 중.
파일 B, C 및 *QQTDF0001이 결합되었습니다. 이것은 상호 위치된 결합(파일 B와 C 사이)과 경로가 지시된 결합(파일 *QQTDF0001에 대해)의 조합이었습니다.
 - CPI4345 임시 분산 결과 파일 QQTDF0002가 조회에 대해 구축됨.
임시 분산 파일 *QQTDF0002가 파일 B, C 및 *QQTDF0001을 결합한 결과를 포함하기 위해 작성되었습니다. 이 파일이 보내졌습니다.
- CPI4343 분산 조회 단계 4중 3에 대한 Optimizer 디버그 메시지는 다음과 같음:
 - CPI4345 임시 분산 결과 파일 QQTDF0003이 조회에 대해 구축됨.
파일 A가 임시 파일 *QQTDF0003으로 보내졌습니다.
- CPI4343 분산 조회 단계 4중 4에 대한 Optimizer 디버그 메시지는 다음과 같음:
 - CPI4342 조회에 대한 분산 결합 수행 중.
파일 *QQTDF0002와 *QQTDF0003이 결합되었습니다. 이것은 재분할된 결합입니다. 두 파일 모두 결합 발생 이전에 보내졌기 때문입니다.

성능을 위해 조회를 조정할 때 사용하고자 할 수 있는 추가 툴로는 CL 명령인, PRTSQLINF(SQL 정보 인쇄)와 CHGQRYA(조회 속성 변경)가 있는데, PRTSQLINF 명령은 SQL 프로그램 및 패키지에 적용됩니다.

DB2 Multisystem에서의 CHGQRYA(조회 속성 변경) 명령 변경사항

CHGQRYA 명령에는 분산 조회에 대해 적용가능한 두 개의 매개변수인 ASYNCJ(비동기 작업 사용) 및 APYRMT(리모트 적용)가 있습니다.

- 『DB2 Multisystem에서의 ASYNCJ(비동기 작업 사용) 매개변수』
- 60 페이지의 『APYRMT(리모트 적용) 매개변수』

주: 다른 매개변수들과는 달리, ASYNCJ 및 APYRMT에는 시스템 값이 없습니다. 기본값이 아닌 다른 값이 필요할 경우, 각 작업에서 그 값을 변경해야 합니다.

DB2 Multisystem에서의 ASYNCJ(비동기 작업 사용) 매개변수

임시 출력기의 사용을 제어하기 위해 ASYNCJ 명령을 사용할 수 있습니다. ASYNCJ 매개변수에는 다음 옵션들이 있습니다.

- *ANY - 임시 출력기 작업을 분산 파일을 포함하는 데이터베이스 조회에 사용할 수 있습니다.
- *DIST - 임시 출력기 작업을 분산 파일을 포함하는 데이터베이스 조회에 사용할 수 있습니다.
- *LOCAL - 임시 출력기 작업을 로컬 파일 조회에 대해서만 사용할 수 있습니다. 이 옵션이 허용되더라도, 현재 로컬 조회 처리를 위해 임시 출력기를 사용하기 위한 시스템 지원이 없습니다. 분산 조회에 대한 임시 출력기가 작동되지 못하도록 *LOCAL 이 추가되었고, 아직 통신이 비동기식으로 수행되도록 허용됩니다.
- *NONE - 임시 출력기를 사용하지 않습니다. 또한, 분산 처리가 수행될 때 통신이 비동기식으로 수행됩니다. 이것은 조회를 분석할 때 매우 유용합니다. 시스템으로부터의 조회 디버그 메시지가 로컬 시스템으로 리턴될 수 있도록 하기 때문입니다.

다음 예는 분산 파일 처리에 대해 비동기 작업 사용이 가능하지 않도록 하는 방법을 나타내고 있습니다.

```
CHGQRYA ASYNCJ(*LOCAL)
```

이 명령은 비동기 작업이 분산 파일을 포함하는 조회에 사용되지 못하게 합니다.

다음 예는 분산 비동기 작업 사용이 완전히 가능하지 않게 하는 방법을 나타내고 있습니다.

```
CHGQRYA ASYNCJ(*NONE)
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

이 명령은 비동기 작업이 어떤 조회에 대해서도 사용되지 못하게 합니다. 또한, 분산 파일을 포함하는 조회에 대해 리모트 시스템에 대한 통신이 동기 형태로 수행됩니다.

다음 예는 STRDBG(디버그 시작) 명령과 조합으로 CHGQRYA 명령을 사용하여 분산 조회를 분석하는 방법을 나타내고 있습니다.

```
STRDBG UPDPROD(*YES)
CHGQRYA ASYNCJ(*NONE)
STRSQL
      SELECT COUNT(*) FROM EMPLOYEE A
```

다음과 같은 디버그 메시지가 작업 기록부에 기록됩니다.

```
Current connection is to relational database SYSA.
DDM job started.
Optimizer debug messages for distributed query step 1 of 2 follow:
Temporary distributed result file *QTD0001 built for query.
Following messages created on target system SYSA.
Arrival sequence access was used for file EMPLOYEE.
Arrival sequence access was used for file EMPLOYEE.
Optimizer debug messages for distributed query step 2 of 2 follow:
Arrival sequence access was used for file EMPLOYEE.
ODP created.
Blocking used for query.
```

코드 예에 관한 정보는 v 페이지의 『코드 면책사항 관련 정보』를 참조하십시오.

APYRMT(리모트 적용) 매개변수

APYRMT 매개변수를 사용하여 다른 CHGQRYA 옵션이 분산 조회 요구 처리시 사용되는 연관되는 리모트 시스템 작업에 적용되는지 지정할 수 있습니다.

- *YES-CHGQRYA 옵션을 리모트 작업에 적용합니다. 이렇게 하면 리모트 작업은 CHGQRYA 명령을 사용할 수 있는 권한을 갖고 있어야 합니다. 그렇지 않으면, 오류 신호가 리모트 작업에 보내집니다.
- *NO-CHGQRYA 옵션을 로컬로만 적용합니다.

다음 예는 CHGQRYA 옵션이 리모트로 적용되지 못하게 합니다.

```
CHGQRYA DEGREE(*NONE) APYRMT(*NO)
```

이 경우, SMP 병렬 처리는 조정 담당자에서 금지되지만, 리모트 시스템은 자체의 고유한 병렬 정도를 선택할 수 있습니다.

이 매개변수들 외에도, QRYTIMLMT(시간 제한 조회)가 작동하는 방법을 알아야 합니다. 매개변수에 지정된 시간 제한은 분산 조회의 각 단계(로컬 및 리모트)에 적용됩니다. 이것은 전체 조회에 적용되지 않습니다. 그러므로, 하나의 조회 단계에서 시간 제한이 발생할 수 있습니다. 그래도, 다른 단계는 문제 없이 계속할 수 있습니다. 시간 제한 옵션이 아주 유용할 수 있는 반면, 분산 조회에서는 주의하여 사용해야 합니다.



성능 고려사항 요약

분산 파일을 사용하는 조회를 개발할 때 다음을 고려해야 합니다.



1. OPNQRYF 명령과 조회 API(QQQQRY)에 대해, ALWCOPYDTA(*OPTIMIZE)를 지정하면 각 노드가 지정된 순서화를 만족시키기 위해 색인이나 정렬을 선택할 수 있습니다.
2. OPNQRYF 명령 및 조회 API(QQQQRY)의 경우, ALWCOPYDTA(*YES) 또는 ALWCOPYDTA(*NO)를 지정하면 각 노드가 지정된 순서화 필드와 정확히 일치하는 색인을 사용하도록 합니다. 이것은 Optimizer가 비분산 파일에 대해 순서화를 처리하는 방법보다 더 제한적입니다.
3. ORDER BY절을 DISTINCT 선택에 추가하면 요구하는 시스템에서 최종 정렬이 요구되지 않으므로 더 빠르게 레코드를 리턴할 수 있습니다.
4. 분할 키의 모든 필드를 그룹화 필드에 포함시키면 일반적으로 2단계 그룹화보다 더 낮게 수행하는 1단계 그룹화가 수행됩니다.
5. 분할 키의 모든 필드를 결합 기준에 포함시키면 일반적으로 상호 위치된 결합이 수행됩니다.
6. 분할 키의 모든 필드를 분리가능한 동일한 레코드 선택사항에 포함시키면 일반적으로 한 노드에서만 조회가 처리됩니다.
7. 다음의 스칼라 함수를 분리가능한 동일한 레코드 선택사항에 포함시키면 일반적으로 한 노드에서만 조회가 처리됩니다.
 - NODENAME
 - NODENUMBER
 - PARTITION

참고 문헌

다음 참고 문헌에는 이 책에 언급된 정보가 나열되어 있습니다.

- 백업 및 회복 주제는 다음 사항에 대한 설정 및 관리 정보를 제공합니다.
 - 저널링, 액세스 경로 보호 및 확약 제어
 - 사용자 보조 기억장치 풀(ASP)
 - 디스크 보호(장치 패리티, 이중화 및 체크섬)백업 매체 및 저장/복원 조작에 대한 성능 정보를 제공합니다. 또한, 활동시 저장(save-while-active) 지원 사용, 다른 릴리스에 저장 및 복원, 그리고 프로그래밍 추가 정보 및 기법과 같은 향상된 백업 및 회복 주제들도 포함되어 있습니다.
- 제어 언어(CL) 주제에서는 iSeries 서버 제어 언어(CL)와 해당 OS/400 명령에 대한 설명을 제공합니다. (비 OS/400 명령은 관련된 사용권 프로그램 자료에 설명되어 있습니다.) iSeries 서버에 대한 모든 CL 명령의 개요를 제공하고 이 명령들을 코딩하는 데 필요한 구문 규칙에 대해서도 설명합니다.
- 데이터베이스 프로그래밍  주제에서는 시스템에서 데이터베이스 파일 작성, 서술 및 갱신 방법에 대한 정보를 포함하여 iSeries 데이터베이스 구성에 대한 자세한 설명을 제공합니다. 또한, OS/400 자료 서술 스펙(DDS) 키워드를 사용하여 시스템에 대해 파일을 정의하는 방법에 대해서도 설명합니다.
- SQL 프로그래밍 개념  주제에서는 AS/400용 DB2 조회 관리자와 SQL Development Kit 사용권 프로그램의 사용 방법에 대한 정보를 제공합니다. 데이터베이스 라이브러리의 자료에 액세스하고 삽입된 SQL문을 포함하는 어플리케이션 프로그램을 준비, 실행 및 테스트하는 방법을 나타내고 있습니다. SQL/400 명령문의 예와 대화식 SQL 함

수의 설명이 수록되어 있습니다. COBOL/400, ILE COBOL/400, PL/I, ILE C/400, FORTRAN/400, RPG/400, ILE RPG/400 및 REXX에서 SQL/400 문을 사용하기 위한 공통 개념 및 규칙에 대해 설명합니다.

- SQL 참조서  주제에서는 Structured Query Language/400 DB2/400문의 사용 방법과 그 명령문의 적절한 사용법에 대한 세부사항을 제공합니다. 명령문의 예에는 구문 다이어그램, 매개변수 및 정의가 포함됩니다. SQLCA(SQL communication area) 및 SQLDA(SQL descriptor area)의 설명과 SQL 한계 리스트도 제공됩니다.
- 분산 데이터베이스 프로그래밍  주제에서는 분산 관계형 데이터베이스 구조(DRDA)를 사용하는 분산 관계형 데이터베이스에 있어서 iSeries 서버 준비 및 관리 작업에 대한 정보를 제공합니다. 유사 시스템 환경의 여러 서버에서 분산 관계형 데이터베이스(RDB)를 계획, 설정, 프로그래밍, 관리 및 작동하는 것에 대해 설명합니다.
- *OptiConnect for OS/400*, SC41-5414-03 책에서는 OptiConnect 지원에 대해 설명합니다. 이 지원을 통해 광섬유 케이블을 사용하는 여러 서버를 연결시킬 수 있습니다. OptiConnect는 더 신속하게 시스템 사이의 데이터베이스에 액세스하고 또 다른 서버로 작업을 분담시킬 수 있습니다. 추가 주제로는 구성, 설치 및 작동 정보가 있습니다.
- *Software Installation*, SC41-5120-06 책에서는 오퍼레이팅 시스템과 사용권 프로그램의 설치를 위한 다음 절차의 관한 단계별 지침과 계획 정보가 수록되어 있습니다.
 - 초기 설치
 - 설치된 릴리스를 새로운 릴리스로 대체
 - 추가적인 사용권 프로그램 추가

- 2차 언어 추가
- 시스템의 1차 언어 변경
- OS/400 API 참조 주제에서는 경험있는 프로그래머를 위해 다음과 같은 OS/400 기능들에 대한 어플리케이션 프로그래밍 인터페이스(API)의 사용 방법에 대한 정보를 제공합니다.
 - 동적 화면 관리자
 - 파일(데이터베이스, 스펴, 계층)
 - 저널 및 예약 제어
 - 메세지 처리
 - 자국어 지원
 - 네트워크 관리
 - 오브젝트
 - 문제점 관리
 - 등록 기능
 - 보안
 - 소프트웨어 제품
 - 소스 디버그
 - UNIX**-유형
 - 사용자 정의 통신
 - 사용자 인터페이스
 - 작업 관리

최초 프로그램 모델(OPM), ILE(Integrated Language Environment) 및 UNIX형 API를 포함하고 있습니다.

DB2 UDB 대칭형 멀티프로세싱에 대한 정보에 대해서는 SQL 프로그래밍 개념 및 데이터베이스 프로그래밍 주제를 참조하십시오.

색인

[가]

개요
최적화 40

결합
경로가 지시된 45
예 45
구현 44
그룹화로 51
브로드캐스트 48
예 48
상호 위치된 44
예 44
재분할된 46
예 46
최적화 44, 48

결합 필드
분할 키를 통한 49
경로가 지시된 결합 45
예 45

계획, 액세스 52
공유 없음 클러스터 1
정의 1

권한 오류 20

그룹화
결합으로 51
구현 49
최적화 49

그룹화, 1단계
최적화 50

그룹화, 2단계
최적화 50

그림 9
노드 그룹 2
데이터베이스 대 노드 번호 관계 9
데이터베이스 파일의 분배 1
분할 파일 7
열린 자료 경로(ODP) 52
파티션 맵 2

[나]

널(null) 허용 필드
분할 키에서 23

네트워크에 시스템 추가
재분산 관련 사항 36

네트워크, 시스템을 추가 36

노드
비주얼
정의 7
자료 7
정의 2
표시 9

노드 그룹
권한 변경 비밀관성 20
그림 2
디폴트 분할 5
변경 5
자료 분할 속성 예 10
삭제 12
예 12
소개 5
작성 7
예 7
특정 분할로 7
정의 2
표시 9
예 9
정보 18
*NODGRP 시스템 오브젝트 5

노드 그룹 명령
사용하기 전에 완료할 태스크 6

노드 그룹 삭제(DLTNODGRP) 명령
사용 12

노드 그룹 속성 변경(CHGNODGRPA) 명령
5, 10

노드 그룹 오브젝트
정의 2

노드 그룹 작성(CRTNODGRP) 명령
사용 7
자료 분할 변경 5

노드 그룹 표시 화면
데이터베이스 대 노드 번호 관계 그림 9
파티션 번호 대 노드 번호 관계 9

노드 그룹 표시(DSPNODGRP) 명령
사용 9

노드 그룹에서의 권한 변경사항
비밀관성 20

노드 그룹(NODGRP) 매개변수 7, 13
논리 파일 멤버 변경(CHGLFM) 명령
분산 파일에 대해 SHARE 매개변수가 허
용되지 않음 18
논리 파일 멤버 추가(ADDLFM) 명령 20
논리 파일 변경(CHGLF) 명령 20
논리 파일 작성(CRTLFL) 명령 20

[다]

단일 파일 조회
구현 40
최적화 40

대화, 보호되는 33

데이터베이스 대 노드 번호 관계 그림 9

데이터베이스 시스템
DB2 Multisystem을 사용하여 시스템 확장
35

데이터베이스 파일
분산 파일로 작성 13
시스템 그림 전체에 분배 1

데이터베이스 파일 위치지정(POSDBF) 명령
분산 파일에 대해 허용되지 않음 18

디폴트 분할 5

[라]

라이브러리 재명명(RNMLIB) 명령
분산 파일에 대해 허용되지 않음 18

레코드 순서화
구현 42
병합 최적화 42
정렬 최적화 42
최적화 42

로컬 레코드
표시 18

로컬(*LOCAL) 시스템
분산 관계형 데이터베이스 네트워크 정의
6

리모트 위치명(RMTLOCNAME) 매개변수 6

리모트 자료
표시 18

리모트 적용(APYRMT) 매개변수 60

[마]

맵, 파티션

- 그림 2
- 정의 2

메세지

- Optimizer 56

멤버 재명명(RNMM) 명령

- 분산 파일에 대해 허용되지 않음 18

멤버 제거(RMVM) 명령

- 분산 파일에 대해 허용되지 않음 18

명령, CL

- 노드 그룹 삭제(DLTNODGRP) 명령 12

노드 그룹 속성 변경

- (CHGNODGRPA) 5, 10

- 노드 그룹 작성(CRTNODGRP) 7

- 자료 분할 변경 5

- 노드 그룹 표시(DSPNODGRP) 9

논리 파일 멤버 변경(CHGLFM)

- 분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

- 논리 파일 멤버 추가(ADDLFM) 20

- 논리 파일 변경(CHGLF) 20

- 논리 파일 작성(CRTLTF) 20

데이터베이스 파일 위치지정(POSDBF)

- 분산 파일에 대해 허용되지 않음 18

라이브러리 재명명(RNMLIB) 명령

- 분산 파일에 대해 허용되지 않음 18

리스트

- 분산 파일에 대해 실행이 허용되지 않는 명령 18

- 분산 파일의 로컬 부분에만 영향을 주는 명령 18

- 분산 파일의 모든 부분에 영향을 주는 명령 20

멤버 재명명(RNMM)

- 분산 파일에 대해 허용되지 않음 18

멤버 제거(RMVM)

- 분산 파일에 대해 허용되지 않음 18

복사(COPY) 명령

- 분산 파일에 대해 허용되지 않음 18

분산 파일 제한사항 18

실제 파일 멤버 변경(CHGPFM)

- 분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

- 실제 파일 멤버 재구성(RGZPFM) 20

- 실제 파일 멤버 지우기(CLRPFM) 20

명령, CL (계속)

실제 파일 멤버 초기화(INZPFM)

- 분산 파일에 대해 허용되지 않음 18

실제 파일 멤버 표시(DSPPFM) 18

실제 파일 변경(CHGPF) 13, 20

- 자료 분배, 사용자 정의 26

실제 파일 작성(CRTPF) 7, 13

실제 파일 제한조건 변경

- (CHGPFPCST) 20

실제 파일 제한조건 제거

- (RMVPCST) 20

실제 파일 제한조건 추가

- (ADDPFCST) 20

실제 파일 트리거 제거(RMVPFTRG) 20

실제 파일 트리거 추가(ADDPFTRG) 20

오브젝트 권한 부여(GRTOBJAUT) 20

오브젝트 권한 취소(RVKOBJAUT) 20

오브젝트 덤프(DMPOBJ)

- 로컬 파일 부분에만 영향을 미침 18

오브젝트 복원(RSTOBJ)

- 로컬 파일 부분에만 영향을 미침 18

오브젝트 설명 표시(DSPOBJD)

- 로컬 파일 부분에만 영향을 미침 18

오브젝트 소유자 변경(CHGOBJOWN) 20

오브젝트 이동(MOVOBJ)

- 분산 파일에 대해 허용되지 않음 18

오브젝트 재명명(RNMOBJ) 20

오브젝트 저장(SAVOBJ)

- 로컬 파일 부분에만 영향을 미침 18

오브젝트 할당해제(DLCOBJ) 20

오브젝트 할당(ALCOBJ) 20

저널 실제 파일 변경사항 종료

- (ENDJRNP) 20

저널 실제 파일 시작(STRJRNP) 20

저널 액세스 경로 시작(STRJRNAP)

- 로컬 파일 부분에만 영향을 미침 18

저널 액세스 경로 종료(ENDJRNAP)

- 로컬 파일 부분에만 영향을 미침 18

저널된 변경사항 적용(APYJRNCHG)

- 로컬 파일 부분에만 영향을 미침 18

저널된 변경사항 제거(RMVJRNCHG)

- 로컬 파일 부분에만 영향을 미침 18

조회 속성 변경(CHGQRYA)

- 변경사항 59

조회 파일 열기(OPNQRYF) 42

중복 오브젝트 작성(CRTDUPOBJ)

- 분산 파일에 대해 허용되지 않음 18

파일 복사(CPYF) 20, 22

명령, CL (계속)

파일 삭제(DLTF) 20

화면 파일 설명(DSPFD) 18

ADDLFM(논리 파일 멤버 추가) 20

ADDPFCST(실제 파일 제한조건 추가) 20

ADDPFM(실제 파일 멤버 추가) 20

ADDPFTRG(실제 파일 트리거 추가) 20

ADDRDBDIRE(RDB 디렉토리 항목 추가) 6

ALCOBJ(오브젝트 할당) 20

APYJRNCHG(저널된 변경사항 적용)

- 로컬 파일 부분에만 영향을 미침 18

CHGLFM(논리 파일 멤버 변경)

- 분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

CHGLF(논리 파일 변경) 20

CHGNODGRPA(노드 그룹 속성 변경) 5, 10

CHGOBJOWN(오브젝트 소유자 변경) 20

CHGPFPCST(실제 파일 제한조건 변경) 20

CHGPFM(실제 파일 멤버 변경)

- 분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

CHGPF(실제 파일 변경) 13, 20

- 자료 분배, 사용자 정의 26

CHGQRYA(조회 속성 변경)

- 변경사항 59

CLRPFM(실제 파일 멤버 지우기) 20

COPY(복사) 명령

- 분산 파일에 대해 허용되지 않음 18

CPYF(파일 복사) 20, 22

CRTDUPOBJ(중복 오브젝트 작성)

- 분산 파일에 대해 허용되지 않음 18

CRTLTF(논리 파일 작성) 20

CRTNODGRP(노드 그룹 작성) 7

- 자료 분할 변경 5

CRTPF(실제 파일 작성) 7, 13

DLCOBJ(오브젝트 할당해제) 20

DLTF(파일 삭제) 20

DLTNODGRP(노드 그룹 삭제) 명령 12

DMPOBJ(오브젝트 덤프)

- 로컬 파일 부분에만 영향을 미침 18

DSPFD(화면 파일 설명) 18

DSPNODGRP(노드 그룹 표시) 9

DSPOBJD(오브젝트 설명 표시)

- 로컬 파일 부분에만 영향을 미침 18

명령, CL (계속)

DSPPFM(실제 파일 멤버 표시) 18
 ENDJRNP(저널 액세스 경로 종료)
 로컬 파일 부분에만 영향을 미침 18
 ENDJRNP(저널 실제 파일 변경사항 종료) 20
 GRTOBJAUT(오브젝트 권한 부여) 20
 INZPFM(실제 파일 멤버 초기화)
 분산 파일에 대해 허용되지 않음 18
 MOVOBJ(오브젝트 이동)
 분산 파일에 대해 허용되지 않음 18
 OPNQRYF(조회 파일 열기) 42
 POSDBF(데이터베이스 파일 위치지정)
 분산 파일에 대해 허용되지 않음 18
 RDB 디렉토리 항목 추가
 (ADDRDBDIRE) 6
 RDB 디렉토리 항목에 대한 작업
 (WRKRDBDIRE) 명령 6
 RGZPFM(실제 파일 멤버 재구성) 20
 RMVJRNCHG(저널된 변경사항 제거)
 로컬 파일 부분에만 영향을 미침 18
 RMVM(멤버 제거)
 분산 파일에 대해 허용되지 않음 18
 RMVPCST(실제 파일 제한조건 제거) 20
 RMVPTRG(실제 파일 트리거 제거) 20
 RNMLIB(라이브러리 재명명) 명령
 분산 파일에 대해 허용되지 않음 18
 RNMM(멤버 재명명)
 분산 파일에 대해 허용되지 않음 18
 RNMOBJ(오브젝트 재명명) 20
 RSTOBJ(오브젝트 복원)
 로컬 파일 부분에만 영향을 미침 18
 RVKOBJAUT(오브젝트 권한 취소) 20
 SAVOBJ(오브젝트 저장)
 로컬 파일 부분에만 영향을 미침 18
 STRJRNAP(저널 액세스 경로 시작)
 로컬 파일 부분에만 영향을 미침 18
 STRJRNPF(저널 실제 파일 시작) 20
 WRKRDBDIRE(RDB 디렉토리 항목에 대한 작업) 명령 6

[바]

번호

파티션

정의 2

병합

레코드 순서화 최적화 42

보호되는 대화

정의 33

확약 제어로 33

복사(COPY) 명령

분산 파일에 대해 허용되지 않음 18

부속 조회(subquery)

설명 52

분산 관계형 데이터베이스(RDB) 네트워크

로컬(*LOCAL) 시스템 정의 6

설정 6

로컬(*LOCAL) 시스템 6

분산 실제 파일

작성

실제 파일 작성(CRTPF) 명령 사용

13

예 13

SQL CREATE TABLE문 사용 13

분산 조회

임시 결과 파일 40

최적화 개요 40

분산 파일

소개 13, 16

작성 13

정의 2

제한사항 15

파일 복사(CPYF) 명령 사용 22

표시

로컬 레코드 18

리모트 자료 18

분할 5

계획 24

디폴트 분할 5

맵 23

번호 23

설정 고려사항 24

정의 2

키 23

SMP의 장점 24

분할 속성, 변경

예 10

분할 키

결합 필드를 통한 49

널(null) 허용 필드 23

선택 25

정의 2

분할 키(PNKEY) 매개변수 13

분할 파일

정의 2

파일 정의 7

분할 파일 예

그림 7

분할 파일(PNFILE) 매개변수 5, 7

브로드캐스트 결합 48

예 48

비동기 작업 사용(ASYN CJ) 매개변수 59

비주일 노드

사용 방법 예 7

작성 방법 7

정의 7

[사]

상대 레코드 번호(RRN) 함수 31

상호 위치된 결합 44

예 44

성능 33

조회 39

조회 요약 60

조회 처리 개선 33

향상 33

예 33

SMP(Symmetric Multiprocessing) 개선점

33

소개

노드 그룹 5

순서화, 레코드

최적화 42

병합 42

정렬 42

스칼라 함수 27

OPNQRYF(조회 파일 열기) CL 명령

HASH 29

NODENAME 29

NODENUMBER 30

PARTITION 27

SQL

HASH 28

NODENAME 29

NODENUMBER 30

PARTITION 27

시스템 오브젝트

*NODGRP 5

시스템, 네트워크에 추가

재분산 관련 사항 36

실제 파일 멤버 변경(CHGPFM) 명령
 분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

실제 파일 멤버 재구성(RGZPFM) 명령 20

실제 파일 멤버 지우기(CLRPFM) 명령 20

실제 파일 멤버 초기화(INZPFM) 명령
 분산 파일에 대해 허용되지 않음 18

실제 파일 멤버 추가(ADDPFM) 명령 20

실제 파일 멤버 표시(DSPPFM) 명령 18

실제 파일 변경(CHGPF) 명령 13, 20
 자료 분배, 사용자 정의 26

실제 파일 작성(CRTPF) 명령 7, 13

실제 파일 제한조건 변경(CHGPF CST) 명령 20

실제 파일 제한조건 제거(RMVPFCST) 명령 20

실제 파일 제한조건 추가(ADDPFCST) 명령 20

실제 파일 트리거 제거(RMVPFTRG) 명령 20

실제 파일 트리거 추가(ADDPFTRG) 명령 20

[아]

액세스 계획 52

열린 자료 경로(ODP)
 그림 52
 재사용가능한 52

예

경로가 지시된 결합 45

경로가 지시된 결합 조회 45

노드 그룹
 디폴트 분할로 작성 7
 삭제 12
 작성 7
 표시 9

노드 그룹을 지정하기 위해 사용되는
 CREATE TABLE문 13

분산 실제 파일, 작성 13

브로드캐스트 결합 48

브로드캐스트 결합 조회 48

비주얼 노드, 사용 7

상호 위치된 결합 44

상호 위치된 결합 조회 44

성능 향상 33

자료 분할 속성, 변경 10

재분할된 결합 46

예 (계속)

재분할된 결합 조회 46

특정 분할로 노드 그룹 작성 7

오브젝트 권한 부여(GRTOBJAUT) 명령 20

오브젝트 권한 취소(RVKOBJAUT) 명령 20

오브젝트 덤프(DMPOBJ) 명령
 로컬 파일 부분에만 영향을 미침 18

오브젝트 복원(RSTOBJ) 명령
 로컬 파일 부분에만 영향을 미침 18

오브젝트 설명 표시(DSPOBJD) 명령
 로컬 파일 부분에만 영향을 미침 18

오브젝트 소유자 변경(CHGOBJOWN) 명령 20

오브젝트 이동(MOVOBJ) 명령
 분산 파일에 대해 허용되지 않음 18

오브젝트 재명명(RNMOBJ) 명령 20

오브젝트 저장(SAVOBJ) 명령
 로컬 파일 부분에만 영향을 미침 18

오브젝트 할당해제(DLCOBJ) 명령 20

오브젝트 할당(ALCOBJ) 명령 20

인쇄된 관련 정보 63

임시 결과 출력기 54

임시 결과 파일
 정의 40

임시 출력기 54
 사용 단점 55
 사용 장점 55
 제어 56

[자]

자료 노드 7

자료 무결성 33

자료 복사 허용(ALWCOPYDTA) 매개변수 42

처리 43

자료 분배
 사용자 정의 26

실제 파일 변경(CHGPF) 명령 26

자료 분할 속성, 변경
 예 10

자료 유형
 파티션 호환가능한 44

재분산 관련 사항
 시스템을 네트워크에 추가할 때 36

재분할된 결합 46
 예 46

재사용가능한 열린 자료 경로(ODP) 52

저널 실제 파일 변경사항 종료(ENDJRNPF) 명령 20

저널 실제 파일 시작(STRJRNPF) 명령 20

저널 액세스 경로 시작(STRJRNAP) 명령
 로컬 파일 부분에만 영향을 미침 18

저널 액세스 경로 종료(ENDJRNAP) 명령
 로컬 파일 부분에만 영향을 미침 18

저널된 변경사항 적용(APYJRNCHG) 명령
 로컬 파일 부분에만 영향을 미침 18

저널된 변경사항 제거(RMVJRNCHG) 명령
 로컬 파일 부분에만 영향을 미침 18

저널링
 DB2 Multisystem 고려사항 21

정의
 공유 없음 클러스터 1
 노드 2
 노드 그룹 2
 노드 그룹 오브젝트 2
 보호되는 대화 33
 분산 파일 2
 분할 2
 분할 키 2
 분할 파일 2
 비주얼 노드 7
 임시 결과 파일 40
 조정 담당자 노드 31
 파티션 맵 2
 파티션 번호 2
 파티션 호환가능 44
 해싱 2

제어
 임시 출력기 56

제한사항
 분산 파일 15
 분산 파일에 대해 실행이 허용되지 않는
 CL 명령 18
 분산 파일의 로컬 부분에만 영향을 주는
 CL 명령 18
 분산 파일의 모든 부분에 영향을 주는 CL
 명령 20

조정 담당자 노드
 정의 31

조회
 결합 구현 44
 결합 최적화 44
 경로가 지시된 결합 45
 예 45
 브로드캐스트 결합 48

조회 (계속)

예 48

상호 위치된 결합 44

예 44

성능 39

성능 고려사항 요약 60

성능 향상 33

임시 결과 파일 40

재분할된 결합 46

예 46

최적화 개요 40

조회 속성 변경(CHGQRYA) 명령

변경사항 59

조회 파일 열기(OPNQRYF) 명령 42

스칼라 함수 27

조회 Optimizer 33

조회, 단일 파일

구현 40

최적화 40

중복 오브젝트 작성(CRTDUPOBJ) 명령

분산 파일에 대해 허용되지 않음 18

[차]

참고 문헌 63

처리

자료 복사 허용(ALWCPYDTA) 매개변수

43

자료 분배(DSTDTA) 매개변수 43

ALWCPYDTA(자료 복사 허용) 매개변수

43

DSTDTA(자료 분배) 매개변수 43

최적화

개요 40

결합 44, 48

그룹화 49

단일 파일 조회 40

레코드 순서화 42

병합 42

정렬 42

분산 조회 40

1단계 그룹화 50

2단계 그룹화 50

DISTINCT절 43

UNION절 43

출력기, 임시 54

출력기, 임시 결과 54

[카]

키, 분할

정의 2

[타]

특수 레지스터 함수

SQL에 대한 31

[파]

파일

분할

정의 2

파일 복사(CPYF) 명령 20

분산 파일에 대해 사용 22

파일 삭제(DLTF) 명령 20

파티션 맵

그림 2

정의 2

파티션 번호 5

정의 2

파티션 번호 대 노드 번호 관계 그림 9

파티션 호환가능

자료 유형 44

정의 44

[하]

함수

SQL

상대 레코드 번호(RRN) 31

특수 레지스터 31

해시 알고리즘 5

해싱

정의 2

화면 파일 설명(DSPFD) 명령 18

확약 제어 33

확장성 33

효과

DB2 Multisystem 2

[숫자]

1단계 그룹화 50

2단계 그룹화 50

2단계 확약 프로토콜 33

A

ADDLFM(논리 파일 멤버 추가) 명령 20

ADDPFCST(실제 파일 제한조건 추가) 명령 20

ADDPFM(실제 파일 멤버 추가) 명령 20

ADDPFTRG(실제 파일 트리거 추가) 명령 20

ADDRDBDIRE(RDB 디렉토리 항목 추가) 명령 6

ALCOBJ(오브젝트 할당) 명령 20

ALWCPYDTA(자료 복사 허용) 매개변수 42
처리 43

APYJRNCHG(저널된 변경사항 적용) 명령
로컬 파일 부분에만 영향을 미침 18

C

CHGLFM(논리 파일 멤버 변경) 명령

분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

CHGLF(논리 파일 변경) 명령 20

CHGNODGRPA(노드 그룹 속성 변경) 명령 5, 10

CHGOBJOWN(오브젝트 소유자 변경) 명령 20

CHGPFCST(실제 파일 제한조건 변경) 명령 20

CHGPFM(실제 파일 멤버 변경) 명령

분산 파일에 대해 SHARE 매개변수가 허용되지 않음 18

CHGPF(실제 파일 변경) 명령 13, 20

자료 분배, 사용자 정의 26

CHGQRYA(조회 속성 변경) 명령
변경사항 59

CLRPFM(실제 파일 멤버 지우기) 명령 20

COPY(복사) 명령

분산 파일에 대해 허용되지 않음 18

CPYF(파일 복사) 명령 20

분산 파일에 대해 사용 22

CREATE TABLE

노드 그룹 예 13

CRTDUPOBJ(중복 오브젝트 작성) 명령

분산 파일에 대해 허용되지 않음 18

CRTL(논리 파일 작성) 명령 20

CRTNODGRP(노드 그룹 작성) 명령

사용 7

자료 분할 변경 5

CRTPF(실제 파일 작성) 명령 7, 13

D

DB2 Multisystem

고려사항 저널링 21

설치 1

소개 1

효과 2

DISTINCT절

구현 43

최적화 43

DLCOBJ(오브젝트 할당해제) 명령 20

DLTF(파일 삭제) 명령 20

DLTNODGRP(노드 그룹 삭제) 명령

사용 12

DMPOBJ(오브젝트 덤프) 명령

로컬 파일 부분에만 영향을 미침 18

DSPFD(화면 파일 설명) 명령 18

DSPNODGRP(노드 그룹 표시) 명령

사용 9

DSPOBJD(오브젝트 설명 표시) 명령

로컬 파일 부분에만 영향을 미침 18

DSPPFM(실제 파일 멤버 표시) 명령 18

E

ENDJRNAP(저널 액세스 경로 종료) 명령

로컬 파일 부분에만 영향을 미침 18

ENDJRNPF(저널 실제 파일 변경사항 종료)

명령 20

F

from 레코드(FROMRCD) 매개변수 18

FROMRCD(from 레코드) 매개변수 18

G

GRTOBJAUT(오브젝트 권한 부여) 명령 20

H

HASH 스칼라 함수

OPNQRYF(조회 파일 열기) CL 명령에 대

한 29

SQL에 대한 28

I

INZPFM(실제 파일 멤버 초기화) 명령

분산 파일에 대해 허용되지 않음 18

M

MOV OBJ(오브젝트 이동) 명령

분산 파일에 대해 허용되지 않음 18

N

NODENAME 스칼라 함수

OPNQRYF(조회 파일 열기) CL 명령에 대
한 29

SQL에 대한 29

NODENUMBER 스칼라 함수

OPNQRYF(조회 파일 열기) CL 명령에 대
한 30

SQL에 대한 30

NODGRP(노드 그룹) 매개변수 7, 13

O

ODP(열린 자료 경로)

그림 52

로컬 52

분산 52

제사용가능한 52

OPNQRYF(조회 파일 열기) 명령 42

스칼라 함수 27

OptiConnect

연결 방법 2

Optimizer

메세지 56

P

PARTITION 스칼라 함수

OPNQRYF(조회 파일 열기) CL 명령에 대
한 27

SQL에 대한 27

POSDBF(데이터베이스 파일 위치지정) 명령

분산 파일에 대해 허용되지 않음 18

PTNFILE(분할 파일) 매개변수 5, 7

PTNKEY(분할 키) 매개변수 13

R

RDB 디렉토리 항목 추가(ADDRDBDIRE) 명
령 6

RDB 디렉토리 항목에 대한 작업

(WRKRDBDIRE) 명령 6

RGZPFM(실제 파일 멤버 재구성) 명령 20

RMTLOCNAME(리모트 위치명) 매개변수 6

RMVJRNCHG(저널된 변경사항 제거) 명령

로컬 파일 부분에만 영향을 미침 18

RMV M(멤버 제거) 명령

분산 파일에 대해 허용되지 않음 18

RMV PFCST(실제 파일 제한조건 제거) 명령
20

RMV P FTRG(실제 파일 트리거 제거) 명령

20

RNMLIB(라이브러리 재명명) 명령

분산 파일에 대해 허용되지 않음 18

RNMM(멤버 재명명) 명령

분산 파일에 대해 허용되지 않음 18

RNMOBJ(오브젝트 재명명) 명령 20

RRN(상대 레코드 번호) 함수 31

RSTOBJ(오브젝트 복원) 명령

로컬 파일 부분에만 영향을 미침 18

RVKOBJAUT(오브젝트 권한 취소) 명령 20

S

SAVOBJ(오브젝트 저장) 명령

로컬 파일 부분에만 영향을 미침 18

SMP(Symmetric Multiprocessing) 33

SQL

스칼라 함수 27

SQL 시작(STRSQL) 명령 42

STRJRNAP(저널 액세스 경로 시작) 명령

로컬 파일 부분에만 영향을 미침 18

STRJRNPF(저널 실제 파일 시작) 명령 20

STRSQL(SQL 시작) 명령 42

U

UNION절

구현 43

최적화 43

W

WRKRDBDIRE(RDB 디렉토리 항목에 대한
작업) 명령 6

[특수 문자]

(APYRMT) 리모트 적용 매개변수 60

(ASYNCJ) 비동기 작업 사용 매개변수 59

*NODGRP 시스템 오브젝트 5

IBM