



Visual Explain 자습서



Visual Explain 자습서

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 105 페이지의 부록 B 『주의사항』의 일반 정보를 읽으십시오.

개정판 주의사항

이 문서에는 IBM에서 소유하고 있는 정보가 있습니다. 이는 라이선스 계약에 따라 제공한 것이며 저작권의 보호를 받습니다. 이 책의 정보에는 제품 보증이 포함되지 않으며, 이 매뉴얼에서 제공된 어떠한 문장도 이와 같이 해석할 수 없습니다.

온라인으로 IBM 서적을 주문하거나 로컬 IBM 담당자를 통해 서적을 주문할 수 있습니다.

- 온라인으로 서적을 주문하려면 IBM Publications Center(www.ibm.com/shop/publications/order)로 이동하십시오.
- 로컬 IBM 담당자를 찾으려면 IBM Directory of Worldwide Contacts(www.ibm.com/planetwide)로 이동하십시오.

미국 또는 캐나다의 DB2 Marketing and Sales에서 DB2 서적을 주문하려면 1-800-IBM-4YOU(426-4968)로 전화하십시오.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

목차

이 책에 대한 정보 v

제 1 부 Visual Explain 자습서 1

제 1 장 레슨 1. Explain 스냅샷 작성	3
Explain 테이블 작성	3
Explain 스냅샷 사용	4
동적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성	5
정적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성	6
다음 레슨 내용	7
제 2 장 레슨 2. 액세스 플랜 그래프 표시 및 사용	9
이전에 Explain된 SQL 또는 XQuery문 목록에서 선 택하여 액세스 플랜 그래프 표시	9
액세스 플랜 그래프에서 기호 읽기	10
확대/축소 슬라이더를 사용하여 그래프의 일부 확대 그래프에서 오브젝트에 대한 세부사항 가져오기	10
테이블, 인덱스 또는 테이블 함수에 대한 통계 가 져오기	11
그래프에서 연산자에 대한 세부사항 가져오기	12
함수에 대한 통계 가져오기	12
테이블 스페이스에 대한 통계 가져오기	12
SQL 또는 XQuery문의 컬럼에 대한 통계 가져오 기	13
구성 매개변수 및 바인드 옵션에 대한 정보 가져 오기	13
그래프 모양 변경	13
다음 레슨 내용	13
제 3 장 레슨 3. 단일 파티션 데이터베이스 환경에서 액세스 플랜 향상	15
액세스 플랜 그래프로 작업	15
단일 파티션 데이터베이스 환경에서 인덱스와 통계 없이 쿼리 실행	16
다음 레슨 내용	19
단일 파티션 데이터베이스 환경에서 runstats를 사용 하여 테이블 및 인덱스에 대한 현재 통계 수집	19
다음 레슨 내용	23
단일 파티션 데이터베이스 환경의 쿼리에서 테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성	23
다음 레슨 내용	28

단일 파티션 데이터베이스 환경에서 테이블 컬럼에
추가 인덱스 작성 28
 다음 레슨 내용 31

제 4 장 레슨 4. 파티션된 데이터베이스 환경에서 액 세스 플랜 향상	33
액세스 플랜 그래프로 작업	33
파티션된 데이터베이스 환경에서 인덱스와 통계 없이 쿼리 실행	34
다음 레슨 내용	36
파티션된 데이터베이스 환경에서 runstats를 사용하여 테이블 및 인덱스에 대한 현재 통계 수집	36
다음 레슨 내용	40
파티션된 데이터베이스 환경의 쿼리에서 테이블을 조 인하는 데 사용되는 컬럼에 인덱스 작성	40
다음 레슨 내용	44
파티션된 데이터베이스 환경에서 테이블 컬럼에 추가 인덱스 작성	44
다음 레슨 내용	47

제 2 부 참조 49

제 5 장 Visual Explain 연산자	51
CMPEXP 연산자	51
DELETE 연산자	51
EISCAN 연산자	51
FETCH 연산자	52
FILTER 연산자	52
GENROW 연산자	53
GRPBY 연산자	53
HSJOIN 연산자	53
INSERT 연산자	54
IXAND 연산자	54
IXSCAN 연산자	55
MSJOIN 연산자	56
NLJOIN 연산자	56
PIPE 연산자	57
RETURN 연산자	57
RIDSCN 연산자	57
RPD 연산자	58
SHIP 연산자	58
SORT 연산자	58

TBSCAN 연산자	59
TEMP 연산자	60
TQ 연산자	60
UNION 연산자	62
UNIQUE 연산자	62
UPDATE 연산자	62
XANDOR 연산자	62
XISCAN 연산자	64
XSCAN 연산자	66
제 6 장 Visual Explain 개념	67
액세스 플랜	67
액세스 플랜 그래프	67
액세스 플랜 그래프 노트	68
클러스터링	68
컨테이너	69
비용	69
커서 블로킹	69
데이터베이스 관리 테이블 스페이스	70
동적 SQL 또는 XQuery	70
Explain 스냅샷	70
Explain 가능한 명령문	71
Explain된 명령문	71
피연산자	72
연산자	72
옵티마이저	73
패키지	74
술어	74
쿼리 최적화 클래스	75
Explain XML 연산자에 대한 샘플 XML 문서 조각	76
술어의 선택성	76
스타 조인	77
정적 SQL 또는 XQuery	77
시스템 관리 테이블 스페이스	78

테이블 스페이스	78
Visual Explain	78
제 7 장 Visual Explain 태스크	81
명령 편집기를 사용하여 액세스 플랜 작성	81
액세스 플랜의 그래픽 표현 보기	82
이미 Explain된 쿼리 명령문 보기	83
패키지에 대해 Explain 가능한 명령문 보기	86
인덱스 작성 지침	87
이전 액세스 플랜	87
LONGDATACOMPAT 사용시 액세스 플랜 검색	88
RUNSTATS 사용	88
이전 및 추후 릴리스에 대해서 Visual Explain 지원	89

제 3 부 부록 91

부록 A. DB2 기술 정보 개요	93
DB2 기술 라이브러리(하드카피 또는 PDF 형식)	94
인쇄된 DB2 서적 주문	96
명령행 처리기에서 SQL 상태 도움말 표시	97
DB2 정보 센터의 다른 버전에 액세스	98
DB2 정보 센터에서 원하는 언어로 항목 표시	98
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신	99
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신	100
DB2 지습서	102
DB2 문제점 해결 정보	103
이용약관	103
부록 B. 주의사항	105
색인	109

이 책에 대한 정보

다음 지습서는 DB2® Visual Explain의 기능에 대한 안내를 제공합니다. 이 지습서의 레슨을 완료함으로써 Visual Explain을 사용하여 Explain된 SQL 또는 XQuery문에 대한 액세스 플랜을 그래프로 볼 수 있는 방법을 학습하게 됩니다.

그래프와 같은 정보를 사용하여 SQL 쿼리를 조정함으로써 성능을 향상시키는 방법도 학습하게 됩니다.

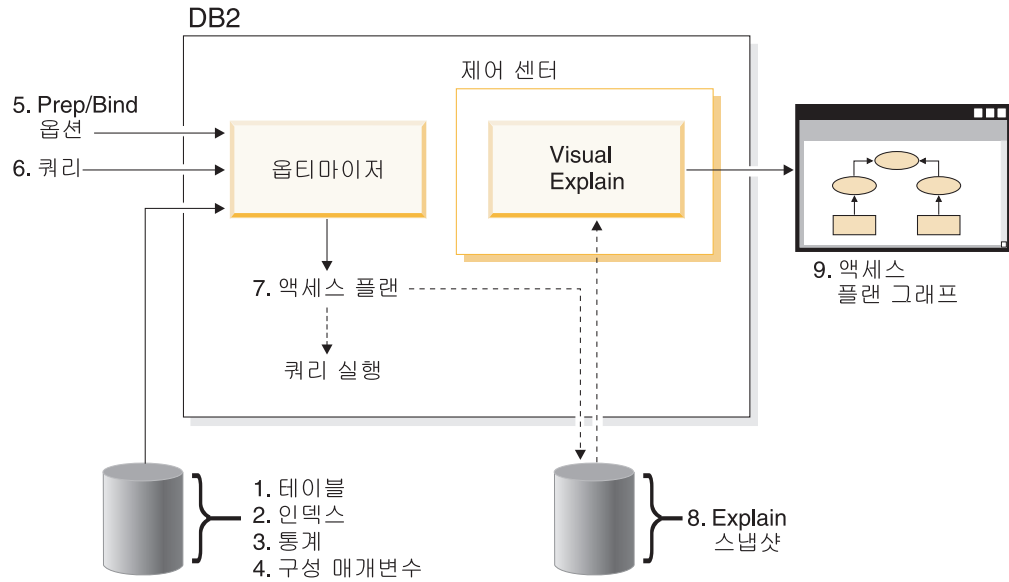
중요사항: 제어 센터 도구로 Visual Explain에 액세스하는 것은 버전 9.7에서 지원되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오. IBM® Data Studio 도구 세트를 사용하여 Visual Explain 기능에 액세스하는 것은 지원되지 않습니다.

옵티마이저를 사용하여 데이터베이스 관리 프로그램이 SQL 쿼리를 살펴보고 최적의 데이터 액세스 방법을 판별합니다. 이러한 데이터 경로를 액세스 플랜이라고 합니다. Visual Explain에서는 특정 쿼리를 수행하기 위해 선택한 액세스 플랜을 볼 수 있으므로 옵티마이저가 수행한 내용을 확인할 수 있습니다. Visual Explain을 사용하여 액세스 플랜을 그래프로 표시할 수 있습니다. 그래프는 쿼리에 관련된 데이터베이스 오브젝트(예: 테이블 및 인덱스)를 시각적으로 표현한 것입니다. 그래프에는 해당 오브젝트에 대해 수행되는 조작(예: 스캔 및 정렬)도 포함되어 있으며 데이터의 흐름을 보여줍니다.

다음 성능 조정 활동 중 하나 또는 모두를 수행하여 데이터에 대한 쿼리 액세스를 향상시킬 수 있습니다.

1. 테이블 설계 조정 및 테이블 데이터 재구성
2. 적절한 인덱스 작성
3. **RUNSTATS** 명령을 사용하여 옵티마이저에 현재 통계 제공
4. 적절한 구성 매개변수 선택
5. 적절한 바인드 옵션 선택
6. 필수 데이터만 검색하도록 쿼리 설계
7. 액세스 플랜에 대한 작업
8. Explain 스냅샷 작성
9. 액세스 플랜을 향상하기 위해 액세스 플랜 그래프 사용

다음 그림은 성능 관련 활동을 보여줍니다. (점선은 Visual Explain에 필요한 조치를 나타냅니다.)



학습 목표

학습서의 내용은 다음과 같습니다.

- Explain 스냅샷 작성. 이 작업은 액세스 플랜 그래프 표시를 위해 필수적입니다.
- 액세스 플랜 그래프 표시 및 조작
- 성능 조정 활동 수행 및 액세스 플랜 향상 방법 설명

주: 성능 조정 관련 레슨은 단일 파티션 데이터베이스 환경에 대한 내용과 파티션된 데이터베이스 환경에 대한 내용으로 나뉘어져 있습니다.

레슨 내내 제공된 샘플 데이터베이스를 사용합니다. 샘플 데이터베이스가 작성되지 않은 경우에는 DB2 정보 센터에서 샘플 데이터베이스 설치에 대한 절을 참조하십시오.

소요 시간

이 자습서를 완료하는 데 소요되는 시간은 대략 60분입니다. 이 학습서와 관련된 다른 개념을 탐색하면 시간이 더 소요될 수 있습니다.

기술 레벨


고급

대상

데이터베이스 관리자 또는 SQL 쿼리 조정을 담당하는 응용프로그램 개발자

환경별 정보

 이 아이콘이 표시된 정보는 단일 파티션 데이터베이스 환경에만 해당됩니다.

 이 아이콘이 표시된 정보는 파티션된 데이터베이스 환경에만 해당됩니다.

제 1 부 Visual Explain 자습서

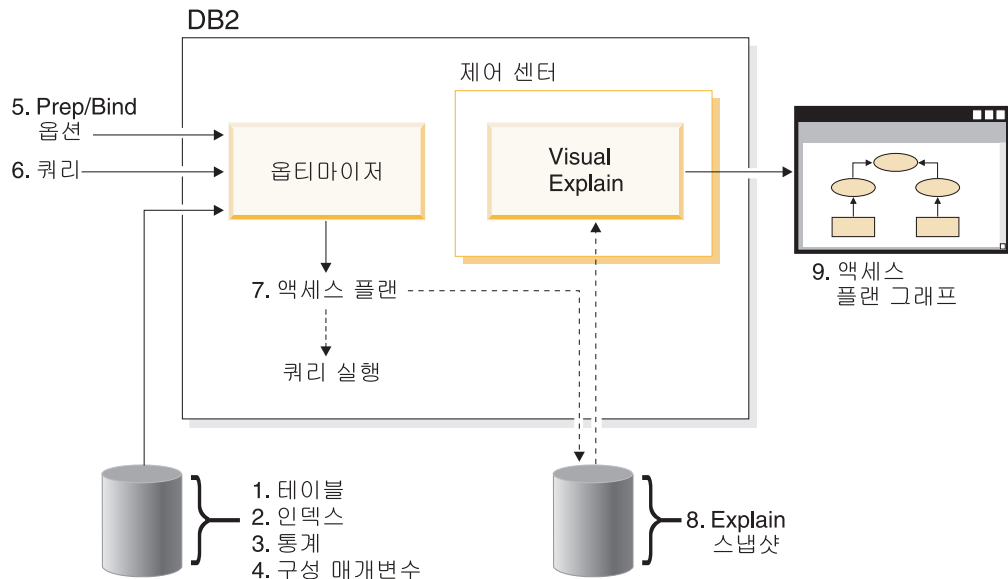
Visual Explain을 사용하면 Explain된 SQL 또는 XQuery문에 대한 액세스 플랜을 그래프로 볼 수 있습니다.

중요사항: 제어 센터 도구로 Visual Explain에 액세스하는 것은 버전 9.7에서 지원되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오. IBM Data Studio 도구 세트를 사용하여 Visual Explain 기능에 액세스하는 것은 지원되지 않습니다.

그래프 정보를 사용하여 쿼리를 조정할 때 수행할 수 있는 태스크는 다음과 같습니다.

- 최적화 시 사용된 통계 보기: 이 통계와 현재 카탈로그 통계를 비교하여, 패키지 리바인딩이 성능 개선에 도움이 되는지 판별할 수 있습니다.
- 인덱스를 사용하여 테이블에 액세스했는지 판별: 인덱스를 사용하지 않은 경우, Visual Explain을 통해 어떠한 컬럼을 인덱스해야 유리한지 판별할 수 있습니다.
- 쿼리에 대한 액세스 플랜 그래프의 이전 버전과 이후 버전을 비교하는 등 여러 가지 유형으로 성능 조정을 수행한 영향 보기
- 계산된 총 비용 및 검색 행 수(카디널리티)를 비롯하여 67 페이지의 『액세스 플랜』의 각 조작 정보 얻기

다음 그림은 DB2 옵티마이저와 제어 센터에서 호출된 Visual Explain 사이의 상호작용을 보여줍니다. (점선은 Visual Explain에 필요한 조치를 나타냅니다.)



Visual Explain의 사용법은 Visual Explain 자습서의 시나리오에 따라 학습할 수 있습니다.

전제조건

- SQL 또는 XQuery문을 동적으로 Explain하려면 Explain 테이블에 INSERT 권한이 있어야 합니다. Explain 테이블이 없는 경우에는 SQL 또는 XQuery문을 Explain 할 때 해당 테이블이 작성됩니다.
- 통계를 포함하여 Explain된 명령문을 자세히 보려면 Explain 테이블과 시스템 카탈로그 테이블에 SELECT 권한이 있어야 합니다.
- Explain된 명령문을 변경하려면 Explain 테이블에 UPDATE 권한이 있어야 합니다.
- Explain된 명령문을 제거하려면 Explain 테이블에 DELETE 권한이 있어야 합니다.

Visual Explain을 시작하려면 다음을 수행하십시오.

- 제어 센터에서 데이터베이스 이름을 마우스 오른쪽 단추로 누르고 **Explain된 명령문 실행기록 표시** 또는 **Explain 쿼리**를 선택하십시오.
- 명령 편집기의 대화식 페이지 또는 스크립트 페이지에서 Explain 가능한 명령문을 실행하십시오.
- Query Patroller의 관리 쿼리 등록 정보 노트북 또는 실행기록 쿼리 등록 정보 노트북에서 액세스 플랜 표시를 누르십시오.

문제점 해결 추가 정보

- 88 페이지의 『LONGDATACOMPAT 사용시 액세스 플랜 검색』
- 89 페이지의 『이전 및 추후 릴리스에 대해서 Visual Explain 지원』

제 1 장 레슨 1. Explain 스냅샷 작성

Explain 스냅샷을 작성하면 SQL 또는 XQuery문의 구조와 잠재적인 실행 성능을 이해할 수 있습니다.

SQL Explain 기능은 정적 또는 동적 SQL 또는 XQuery문이 컴파일되는 환경 정보를 캡처하는 데 사용됩니다. 캡처된 정보를 사용하여 SQL 또는 XQuery문의 구조와 잠재적인 실행 성능을 이해할 수 있습니다. Explain 스냅샷은 SQL 또는 XQuery문이 Explain될 때 수집되는 압축 정보입니다. 이 정보는 EXPLAIN_STATEMENT 테이블에 BLOB(Binary Large Object)로 저장되며 다음과 같은 정보를 포함합니다.

- 해당 연산자와 액세스된 테이블 및 인덱스를 포함하는 액세스 플랜의 내부 표시
- 데이터베이스 오브젝트에 대한 통계와 각 조작에 대한 누적 비용을 포함하는, 옵티마이저가 사용하는 결정 기준

액세스 플랜 그래프를 표시하려면 Visual Explain에서는 Explain 스냅샷에 포함된 정보가 필요합니다.

Explain 테이블 작성

이 절에서는 Explain 테이블의 작성 방법을 보여줍니다.

Explain 스냅샷을 작성하려면, 사용자 ID에 다음 Explain 테이블이 존재하는지 확인해야 합니다.

- EXPLAIN_INSTANCE
- EXPLAIN_STATEMENT

존재 여부를 확인하려면 list tables 명령을 사용하십시오. 이러한 테이블이 없으면, 다음 지시사항을 따라 테이블을 작성해야 합니다.

1. DB2 데이터베이스 관리 프로그램이 아직 시작되지 않았으면 **db2start** 명령을 실행하십시오.
2. CLP 프롬프트에서 사용하려는 데이터베이스로 연결하십시오.
이 자습서에서는 **connect to sample** 명령을 사용하여 샘플 데이터베이스에 연결하십시오.
3. EXPLAIN.DDL 파일에 제공된 샘플 명령 파일을 사용하여 Explain 테이블을 작성하십시오. 이 파일은 sqllib#misc 디렉토리에 있습니다.
명령 파일을 실행하려면, sqllib#misc 디렉토리로 가서 **db2 -tf EXPLAIN.DDL**

명령을 발행하십시오. 이 명령 파일은 연결된 사용자 ID가 접두어로 붙은 Explain 테이블을 작성합니다. 이 사용자 ID는 데이터베이스에 대한 CREATETAB 특권 또는 DBADM 권한을 가져야 합니다.

주: 버전 9에서 Explain문 실행기록 창은 SYSTOOLS 스키마 및 현재 권한 부여 ID의 스키마 둘 모두에서 Explain 레코드를 표시합니다. Visual Explain이 SYSTOOLS 레코드를 검색하여 Explain문 실행기록 창에 표시하려면 SYSTOOLS Explain 테이블에 대한 읽기 특권이 있어야 합니다. 읽기 액세스 권한이 없으면, 이러한 레코드는 표시되지 않습니다. 또한 DB2 데이터베이스 관리 프로그램의 이전 버전에서 이주된 경우에는 db2exmig를 실행해야 Explain 테이블을 이주할 수 있습니다.

Explain 스냅샷 사용

이 절에서는 Explain 스냅샷의 사용 방법을 보여줍니다.

Visual Explain에 대해 학습하는 데 유용한 네 가지 샘플 스냅샷이 제공됩니다. 사용자 자신의 스냅샷 작성에 대한 정보는 다음 절에 설명되어 있지만 이 자습서에서 사용할 사용자 자신의 스냅샷을 작성할 필요는 없습니다.

- 동적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성
- 정적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성

다음 샘플 스냅샷에 사용된 쿼리는 가장 많은 급여를 받는 관리자 급여의 90% 이상을 받는 모든 비관리자급 사원의 이름, 부서 및 급여를 나열합니다.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
O.DEPTNUM = S.DEPT AND
S.JOB <> 'Mgr' AND
S.SALARY+S.COMM > ALL (SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr')
```

쿼리는 다음 두 부분으로 구성됩니다.

1. 서브쿼리(괄호 안의 부분)는 각 관리자 급여의 90%를 구성하는 데이터 행을 생성합니다. 서브쿼리는 ALL로 규정되므로 이 테이블에서 가장 큰 값만이 검색됩니다.
2. 기본 쿼리는 부서 번호가 동일하고 JOB이 'Mgr'이 아니며 (급여+수당)이 서브쿼리에서 리턴된 값보다 큰 모든 행을 ORG 및 STAFF 테이블에서 조인합니다.

기본 쿼리에는 다음과 같이 세 가지 술어가 있습니다.



1. O.DEPTNUMB = S.DEPT
2. S.JOB <> 'Mgr'
3. S.SALARY+S.COMM > ALL (SELECT ST.SALARY*.9
 FROM STAFF ST
 WHERE ST.JOB='Mgr')

이러한 술어는 각각 다음을 나타냅니다.

1. 부서 번호가 동일한 ORG 및 STAFF 테이블을 조인하는 Join 술어
2. STAFF 테이블의 JOB 컬럼의 로컬 술어
3. 서브쿼리 결과를 사용하는 STAFF 테이블의 SALARY 및 COMM 컬럼에 있는 로컬 술어

샘플 스냅샷을 로드하려면 다음을 수행하십시오.

1. 데이터베이스 관리 프로그램이 아직 시작되지 않았으면 **db2start** 명령을 발행하십시오.
2. Explain 테이블이 데이터베이스에 존재하는지 확인하십시오.
확인 방법은 Explain 테이블 작성의 지시사항을 따르십시오.
3. 사용하려는 데이터베이스에 연결하십시오.
이 지습서의 경우, 샘플 데이터베이스에 연결하게 됩니다. 샘플 데이터베이스에 연결하려면, CLP 프롬프트에서 **connect to sample** 명령을 발행하십시오. 샘플 데이터베이스가 작성되지 않은 경우에는 DB2 정보 센터에서 샘플 데이터베이스 설치에 대한 절을 참조하십시오.
4. 미리 정의된 스냅샷을 импорт하려면 DB2 명령 파일 VESAMPL.DDL을 실행하십시오.

-  이 파일은 sqllib\samples\ve 디렉토리에 있습니다.
-  이 파일은 sqllib\samples\ve\winter 디렉토리에 있습니다.

명령 파일을 실행하려면, 이 디렉토리로 가서 **db2 -tf vesAMPL.ddl** 명령을 발행하십시오.

- 이 명령 파일은 Explain 테이블을 작성하는 데 사용된 것과 동일한 사용자 ID를 사용하여 실행되어야 합니다.
- 이 명령 파일은 미리 정의된 스냅샷만 импорт합니다. 테이블이나 데이터를 작성하지 않습니다. 나중에 설명되는 성능 조정 활동(예: CREATE INDEX 및 runstats)은 샘플 데이터베이스의 테이블 및 데이터에서 실행됩니다.

이제 액세스 플랜 그래프를 표시하고 사용할 준비가 되었습니다.

동적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성

이 레슨에서는 동적 SQL 또는 XQuery문에 대한 Explain 스냅샷의 작성 방법을 대략적으로 설명합니다.

주: 이 절에서는 사용자 참조를 위해 Explain 스냅샷 작성 정보가 제공됩니다. 샘플 Explain 스냅샷이 함께 제공되므로 자습서를 사용하여 학습하려면 이 태스크를 완료하지 않아도 됩니다.

동적 SQL 또는 XQuery문에 대한 Explain 스냅샷을 작성하려면 다음 단계를 수행하십시오.

1. 데이터베이스 관리 프로그램이 아직 시작되지 않았으면 **db2start** 명령을 발행하십시오.
2. Explain 테이블이 데이터베이스에 존재하는지 확인하십시오.
이렇게 하려면, 3 페이지의 『Explain 테이블 작성』의 지시사항을 따르십시오.
3. CLP 프롬프트에서 사용하려는 데이터베이스로 연결하십시오.
예를 들어 샘플 데이터베이스에 연결하려면, **connect to sample** 명령을 발행하십시오. 샘플 데이터베이스를 작성하려면 DB2 정보 센터에서 샘플 데이터베이스 설치에 대한 절을 참조하십시오.
4. CLP 프롬프트에서 다음 명령 중 하나를 사용하여 동적 SQL 또는 XQuery문에 대한 Explain 스냅샷을 작성하십시오.
 - SQL 또는 XQuery문을 실행하지 않고 Explain 스냅샷을 작성하려면, **set current explain snapshot=explain** 명령을 발행하십시오.
 - Explain 스냅샷을 작성하고 SQL 또는 XQuery문을 실행하려면, **set current explain snapshot=yes** 명령을 발행하십시오.

이 명령은 Explain 특수 레지스터를 설정합니다. 일단 설정되면, 모든 후속 SQL 또는 XQuery문에 영향을 줍니다. 자세한 정보는 CURRENT EXPLAIN SNAPSHOT 특수 레지스터 및 SET CURRENT EXPLAIN SNAPSHOT문을 참조하십시오.

5. CLP 프롬프트에서 SQL 또는 XQuery문을 제출하십시오.
6. 스냅샷의 액세스 플랜 그래프를 보려면 Explain된 명령문 실행기록 창(제어 센터에서도 가능)을 새로 고친 후 스냅샷을 더블 클릭하십시오.
7. 선택사항: 스냅샷 기능 작동을 해제하려면, SQL 또는 XQuery문을 제출한 후에 **set current explain snapshot=no** 명령을 발행하십시오.

정적 SQL 또는 XQuery문에 대한 Explain 스냅샷 작성

이 레슨에서는 정적 SQL 또는 XQuery문에 대한 Explain 스냅샷의 작성 방법을 대략적으로 설명합니다.

주: 이 절에서는 사용자 참조를 위해 Explain 스냅샷 작성 정보가 제공됩니다. 샘플 Explain 스냅샷이 함께 제공되므로 자습서를 사용하여 학습하려면 이 태스크를 완료하지 않아도 됩니다.

정적 SQL 또는 XQuery문에 대한 *Explain* 스냅샷을 작성하려면 다음 단계를 수행하십시오.

1. 데이터베이스 관리 프로그램이 아직 시작되지 않았으면 **db2start** 명령을 발행하십시오.
2. *Explain* 테이블이 데이터베이스에 존재하는지 확인하십시오.
이렇게 하려면, 3 페이지의 『*Explain* 테이블 작성』의 지시사항을 따르십시오.
3. CLP 프롬프트에서 사용하려는 데이터베이스로 연결하십시오.
예를 들어 샘플 데이터베이스에 연결하려면, **connect to sample** 명령을 발행하십시오.
4. 응용프로그램을 바인딩하거나 준비할 때 **EXPLSNAP** 옵션을 사용하여 정적 SQL 또는 XQuery문에 대한 *Explain* 스냅샷을 작성하십시오.
예를 들면, *bind your file explsnap yes* 명령을 발행하십시오.
5. 선택사항: 스냅샷의 액세스 플랜 그래프를 보려면 *Explain*된 명령문 실행기록 창(제어 센터에서도 가능)을 새로 고친 후 스냅샷을 더블 클릭하십시오.

EXPLSNAP 옵션 사용에 대한 정보는 『바인드 고려사항』, **CURRENT EXPLAIN SNAPSHOT** 특수 레지스터, **BIND** 및 **REBIND** 명령, *EXPLAIN*문을 참조하십시오.

다음 레슨 내용

레슨 2로 이동합니다.

"레슨 2. 액세스 플랜 그래프 표시 및 사용"에서는 액세스 플랜 그래프를 보고 그 내용을 이해하는 방법에 대해 학습합니다.

제 2 장 레슨 2. 액세스 플랜 그래프 표시 및 사용

이 레슨에서는 액세스 플랜 그래프를 표시하고 사용하기 위해 액세스 플랜 그래프 창을 사용하게 됩니다.

액세스 플랜 그래프는 액세스 플랜의 그래픽 표현입니다. 여기에서 다음에 대한 세부사항을 볼 수 있습니다.

- 테이블(및 그와 관련된 컬럼)과 인덱스
- 연산자(예: 테이블 스캔, 정렬 및 조인)
- 테이블 스페이스 및 함수

다음을 수행하여 액세스 플랜 그래프를 표시할 수 있습니다.

- 이전에 Explain된 명령문 목록에서 선택
- 패키지의 Explain 가능한 명령문 목록에서 선택
- SQL 또는 XQuery문을 동적으로 Explain

레슨 1에서 로드한 샘플 Explain 스냅샷의 액세스 플랜 그래프로 작업할 것이므로, 이전에 Explain된 명령문 목록에서 선택하게 됩니다. 액세스 플랜 그래프를 표시하는 다른 메소드에 대한 정보는 Visual Explain 도움말을 참조하십시오.

이전에 Explain된 SQL 또는 XQuery문 목록에서 선택하여 액세스 플랜 그래프 표시

그래프는 아래쪽에서 위쪽으로 읽혀집니다. 쿼리의 첫 단계는 그래프의 맨 아래에 나열되며 마지막 단계는 맨 위에 나열됩니다.

이전에 Explain된 명령문 목록에서 선택하여 액세스 플랜 그래프를 표시하려면 다음을 수행하십시오.

1. 제어 센터에서 샘플 데이터베이스를 찾을 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain**된 명령문 실행기록 표시를 선택하십시오.
Explain된 명령문 실행기록 창이 열립니다.
3. Explain 스냅샷이 있는 명령문에 대한 액세스 플랜 그래프는 표시만 할 수 있습니다. 규정된 명령문에는 **Explain** 스냅샷 컬럼에 YES 항목이 있습니다. 쿼리 번호 1로 식별되는 항목을 더블 클릭하십시오(쿼리 번호 컬럼을 찾기 위해 오른쪽으로 스크롤해야 할 수도 있습니다).

명령문에 대한 액세스 플랜 그래프 창이 열립니다.

액세스 플랜 그래프에서 기호 읽기

액세스 플랜 그래프는 액세스 플랜 구조를 트리로 보여줍니다.

트리의 노드는 다음을 나타냅니다.

- 테이블: 직사각형
- 인덱스: 다이아몬드
- 연산자: 팔각형. TQ 연산자(평행사변형)
- 테이블 함수(육각형)

연산자의 경우, 연산자 유형 오른쪽 대괄호 안의 수는 각 노드에 대해 고유한 ID입니다. 연산자 유형 아래의 숫자는 누적 비용입니다.

확대/축소 슬라이더를 사용하여 그래프의 일부 확대

이 절에서는 확대/축소 슬라이더를 사용하여 그래프의 일부를 확대하는 방법에 대해 대략적으로 설명합니다.

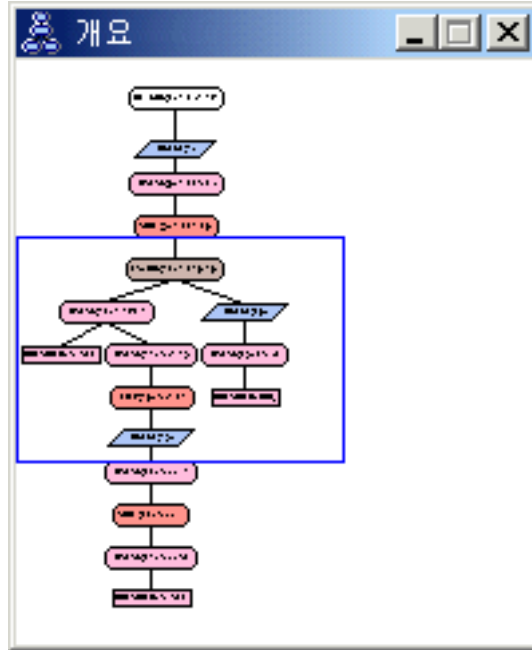
액세스 플랜 그래프를 표시하면, 그래프 전체가 표시되므로 각 노드를 구별하는 세부사항이 표시되지 않을 수도 있습니다.

액세스 플랜 그래프 창에서 확대/축소 슬라이더를 사용하여 그래프의 일부를 확대하십시오.

1. 그래프 왼쪽에 있는 확대/축소 슬라이더 막대에서 작은 스크롤 상자 위에 마우스 포인터를 놓으십시오.
2. 그래프가 원하는 확대 레벨에 올 때까지 마우스 오른쪽 단추를 누른 상태에서 슬라이더를 끄십시오.

그래프의 다른 부분을 보려면, 스크롤 막대를 사용하십시오.

크고 복잡한 액세스 플랜 그래프를 보려면 그래프 개요 창을 사용하십시오. 이 창을 사용하여 보고 있는 그래프의 일부를 표시하거나 그래프를 확대하거나 스크롤할 수 있습니다. 확대/축소 상자의 섹션은 액세스 플랜에 표시됩니다.



그래프를 스크롤하려면, 그래프 개요 창에서 강조표시된 영역 위에 마우스 포인터를 놓고 왼쪽 마우스 단추를 누른 상태에서 원하는 액세스 그래프 부분이 표시될 때까지 마우스를 이동시키십시오.

그래프에서 오브젝트에 대한 세부사항 가져오기

이 절에서는 액세스 플랜 그래프 오브젝트에 대한 자세한 정보에 액세스하는 방법을 설명합니다.

액세스 플랜 그래프에서 오브젝트에 대한 자세한 정보에 액세스할 수 있습니다. 다음을 표시할 수 있습니다.

- 다음과 같은 오브젝트의 시스템 카탈로그 통계
 - 테이블, 인덱스 및 테이블 함수
 - 비용, 등록 정보 및 입력 인수와 같은 연산자에 대한 정보
 - 내장 함수 또는 사용자 정의 함수(UDF)
 - 테이블 스페이스
 - SQL 또는 XQuery문에서 참조된 컬럼
- 구성 매개변수 및 바인드 옵션에 대한 정보(최적화 매개변수)

테이블, 인덱스 또는 테이블 함수에 대한 통계 가져오기

그래프에서 단일 테이블(직사각형), 인덱스(다이아몬드) 또는 테이블 함수(육각형)에 대한 카탈로그 통계를 보려면, 해당 노드를 더블 클릭하십시오. 시스템 카탈로그 테이블

에 현재 존재하는 정보뿐 아니라 스냅샷 작성 시 적용된 통계에 대한 정보까지 표시된 통계 창이 선택된 오브젝트에 대해 열립니다.

그래프에서 여러 테이블, 인덱스 또는 테이블 함수에 대한 카탈로그 통계를 보려면, 해당 항목(강조표시)을 눌러 각각 선택한 후 **노드->통계** 표시를 선택하십시오. 선택된 오브젝트마다 통계 창이 열립니다. (창은 겹쳐 배열될 수 있으므로 모든 창을 보려면 해당 창의 위치를 조정해야 할 수도 있습니다.)

Explain 컬럼의 *STATS_TIME* 항목에 통계 갱신 안됨 항목이 포함되면, 옵티마이저가 액세스 플랜을 작성할 때 통계가 없었던 것입니다. 따라서 옵티마이저가 액세스 플랜을 작성하는 데 특정 통계를 필요로 하는 경우, 디폴트가 사용됩니다. 디폴트 통계가 옵티마이저에서 사용되면 **Explain** 컬럼에서 (디폴트)로 식별됩니다.

그래프에서 연산자에 대한 세부사항 가져오기

단일 연산자(팔각형)에 대한 카탈로그 통계를 보려면, 해당 노드를 더블 클릭하십시오. 선택한 연산자에 대해 다음과 같은 정보를 표시하는 연산자 세부사항 창이 열립니다.

- 계산된 누적 비용(I/O, CPU 지시사항 및 전체 비용)
- 지금까지의 카디널리티(즉, 검색된 예상 행 수)
- 플랜에서 지금까지 액세스되고 조인된 테이블
- 지금까지 액세스된 테이블의 컬럼
- 예상된 선택성을 포함하여 지금까지 적용된 술어
- 각 연산자에 대한 입력 인수

여러 연산자의 세부사항을 보려면, 해당 항목(강조표시)을 눌러 각각 선택한 후 **노드->세부사항** 표시를 선택하십시오. 선택된 오브젝트마다 통계 창이 열립니다. (창은 겹쳐 배열될 수 있으므로 모든 창을 보려면 해당 창의 위치를 조정해야 할 수도 있습니다.)

함수에 대한 통계 가져오기

내장 함수 및 사용자 정의 함수에 대한 카탈로그 통계를 보려면, **명령문->통계 표시->함수**를 선택하십시오. 함수 창에 표시된 목록에서 하나 이상의 항목을 선택하십시오. 선택된 함수마다 함수 통계 창이 열립니다.

테이블 스페이스에 대한 통계 가져오기

테이블 스페이스에 대한 카탈로그 통계를 보려면, **명령문->통계 표시->테이블 스페이스**를 선택하십시오. 테이블 스페이스 창에 표시된 목록에서 하나 이상의 항목을 선택하십시오. 선택한 테이블 스페이스마다 테이블 스페이스 통계 창이 열립니다.

SQL 또는 XQuery문의 컬럼에 대한 통계 가져오기

SQL 또는 XQuery문에서 참조된 컬럼의 통계를 가져오려면 다음을 수행하십시오.

1. 액세스 플랜 그래프에서 테이블을 더블 클릭하십시오. 테이블 통계 창이 열립니다.
2. 참조된 컬럼 누름 단추를 누르십시오. 참조된 컬럼 창이 열리고 테이블에 컬럼이 나열됩니다.
3. 목록에서 하나 이상의 컬럼을 선택하십시오. 선택한 컬럼마다 참조된 컬럼 통계 창이 열립니다.

구성 매개변수 및 바인드 옵션에 대한 정보 가져오기

구성 매개변수 및 바인드 옵션에 대한 정보(최적화 매개변수)를 보려면, 액세스 플랜 그래프 창에서 명령문->최적화 매개변수 표시를 선택하십시오. 현재 값뿐만 아니라 스냅샷 작성 시 적용된 매개변수 값에 대한 정보도 표시하는 최적화 매개변수 창이 열립니다.

그래프 모양 변경

이 절에서는 그래프 모양을 변경하는 데 필요한 단계에 대해 설명합니다.

그래프가 표시되는 방법에 대한 다양한 특성을 변경하려면 다음을 수행하십시오.

1. 액세스 플랜 그래프 창에서 보기->설정을 선택하십시오. 액세스 플랜 그래프 설정 노트북이 열립니다.
2. 배경색을 변경하려면 그래프 탭을 선택하십시오.
3. 다양한 연산자 색상을 변경하려면, 기본, 확장, 갱신 및 기타 탭을 사용하십시오.
4. 테이블, 인덱스 또는 테이블 함수 노드의 색상을 변경하려면, 피연산자 탭을 선택하십시오.
5. 연산자 노드에 표시된 정보(비용 또는 카디널리티의 유형으로서 지금까지 리턴된 예상 행 수)를 지정하려면 연산자 탭을 선택하십시오.
6. 테이블 노드에 스키마 이름을 표시할지 또는 사용자 ID를 표시할지 지정하려면 연산자 탭을 선택하십시오.
7. 노드를 2차원으로 표시할지 또는 3차원으로 표시할지를 지정하려면, 노드 탭을 선택하십시오.
8. 선택한 옵션으로 그래프를 갱신한 후 설정값을 저장하려면 적용을 누르십시오.

다음 레슨 내용

레슨 3 또는 4로 이동합니다.

단일 파티션 데이터베이스 환경에서 작업 중인 경우, 15 페이지의 제 3 장 『레슨 3. 단일 파티션 데이터베이스 환경에서 액세스 플랜 향상』으로 이동하여 여러 가지 성능 조정 활동이 액세스 플랜을 어떻게 변경하고 향상시킬 수 있는지 학습하십시오.

파티션된 데이터베이스 환경에서 작업 중인 경우, 33 페이지의 제 4 장 『레슨 4. 파티션된 데이터베이스 환경에서 액세스 플랜 향상』으로 이동하여 여러 가지 성능 조정 활동이 액세스 플랜을 어떻게 변경하고 향상시킬 수 있는지 학습하십시오.

제 3 장 레슨 3. 단일 파티션 데이터베이스 환경에서 액세스 플랜 향상

이 레슨에서는 다양한 성능 조정 활동을 수행할 때 기본 쿼리에 대한 액세스 플랜 및 관련 창이 어떻게 변경되는지 학습합니다.

그림과 함께 제공되는 일련의 예를 사용하여, **RUNSTATS** 명령을 사용하고 적절한 인덱스를 추가함으로써 단순 쿼리의 액세스 플랜에 대해 계산된 전체 비용이 향상될 수 있는 방법에 대해 배우게 됩니다.

Visual Explain을 이해할수록, 쿼리 성능을 조정하는 또 다른 방법을 발견하게 됩니다.

액세스 플랜 그래프로 작업

네 개의 샘플 Explain 스냅샷을 예로 들어 성능 조정이 어떻게 데이터베이스 성능의 중요한 부분이 되는지 학습합니다.

Explain 스냅샷과 연관된 쿼리는 1에서 4까지의 번호가 지정되어 있습니다. 각 쿼리는 동일한 SQL 또는 XQuery문(레슨 1에서 설명)을 사용합니다.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
O.DEPTNUMB = S.DEPT AND
S.JOB <> 'Mgr' AND
S.SALARY+S.COMM > ALL ( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
```

그러나 반복되는 각 쿼리는 이전 실행보다 더 많은 성능 조정 기술을 사용합니다.

예를 들면, 쿼리 1에는 성능 조정이 없는 반면, 쿼리 4에는 가장 많습니다. 쿼리의 차이점은 다음에 설명되어 있습니다.

쿼리 1

인덱스 및 통계가 없는 쿼리 실행

쿼리 2

쿼리에 테이블 및 인덱스에 대한 현재 통계 수집

쿼리 3

쿼리에서 테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성

쿼리 4

테이블 컬럼에 추가 인덱스 작성

단일 파티션 데이터베이스 환경에서 인덱스와 통계 없이 쿼리 실행

이 예에서 액세스 플랜은 인덱스와 통계 없이 SQL 쿼리에 대해 작성되었습니다.

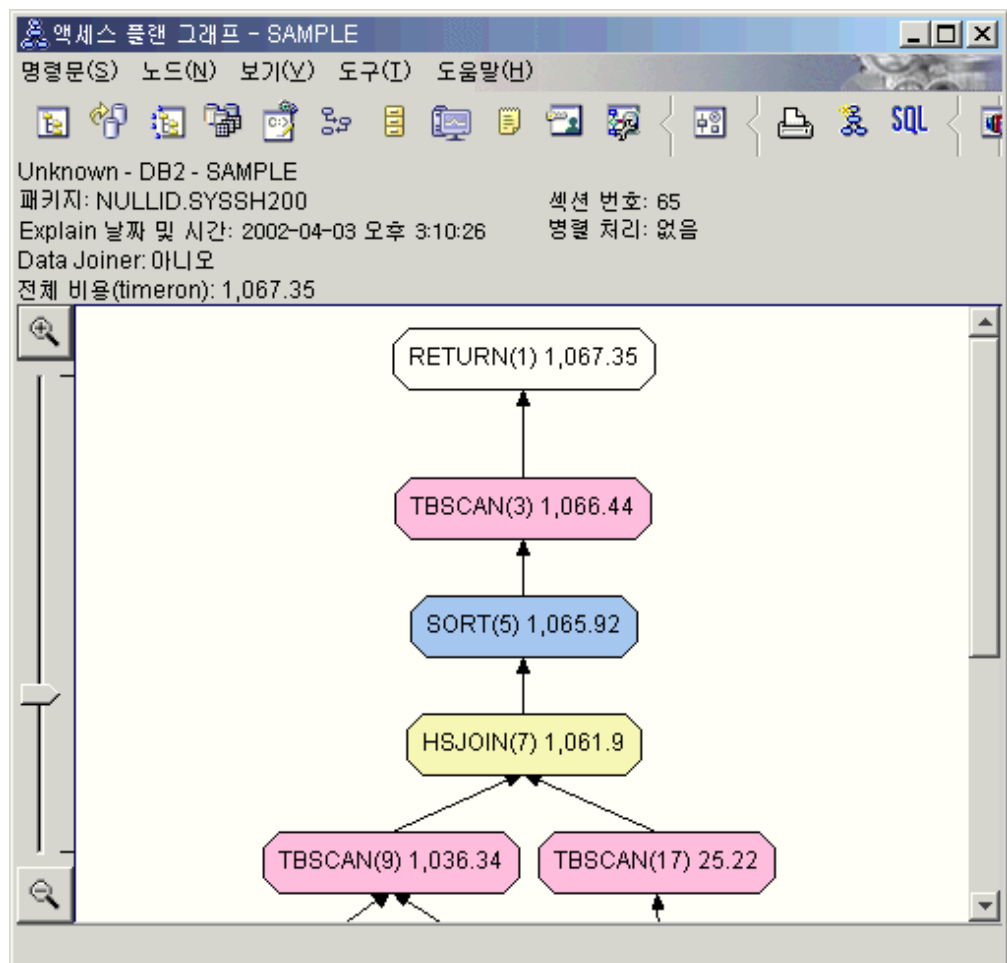
이 쿼리(쿼리 1)에 대한 액세스 플랜 그래프를 보려면 다음을 수행하십시오.

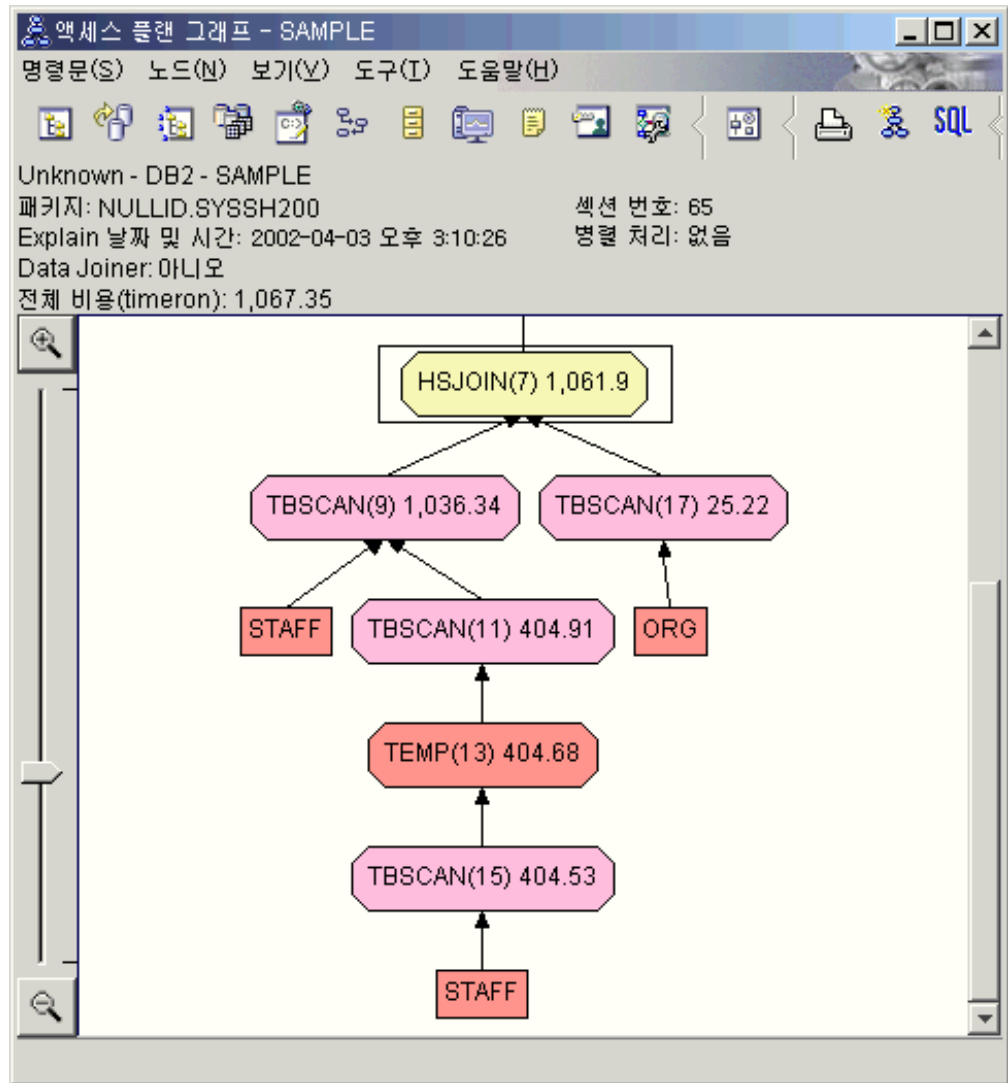
1. 제어 센터에서 샘플 데이터베이스를 찾을 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain**된 명령문 실행기록 표시를 선택하십시오.

Explain된 명령문 실행기록 창이 열립니다.

3. 쿼리 번호 1로 식별되는 항목을 더블 클릭하십시오. (오른쪽으로 스크롤하여 쿼리 번호 컬럼을 찾아야 할 수도 있습니다.)

명령문에 대한 액세스 플랜 그래프 창이 열립니다.





다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 쿼리에 각 테이블에 대한 현재 통계가 존재합니까?

쿼리에 각 테이블에 대한 현재 통계가 존재하는지 점검하려면, 액세스 플랜 그래프에서 각 테이블 노드를 더블 클릭하십시오. 열린 테이블 통계 창에는 **Explain** 컬럼 아래의 **STATS_TIME** 행에 스냅샷 작성 시 통계가 수집되지 않았음을 알리는 "통계 갱신 안됨"이 표시됩니다.

현재 통계가 없으면 옵티마이저는 실제 통계와 다를 수도 있는 디폴트 통계를 사용합니다. 디폴트 통계는 테이블 통계 창의 **Explain** 컬럼 아래에 "디폴트"라는 단어로 식별됩니다.

ORG 테이블에 대한 테이블 통계 창의 정보에 따르면, 옵티마이저는 디폴트 통계 (Explain 값 옆에 표시)를 사용했습니다. 스냅샷 작성 시 실제 통계를 사용할 수 없으므로 디폴트 통계가 사용되었습니다(STATS_TIME 행에 표시).

통계	Explain	현재
CREATE_TIME	2002-04-03 오후 3:05:03	2002-04-03 오후 3:05:03
STATS_TIME	통계 갱신 안됨	2002-04-03 오후 4:25:19
CARD	55 (디폴트)	8
NPAGES	1 (디폴트)	1
FPAGES	1 (디폴트)	1
COLCOUNT	5 (디폴트)	5
OVERFLOW	0 (디폴트)	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	아니오 (디폴트)	아니오

2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

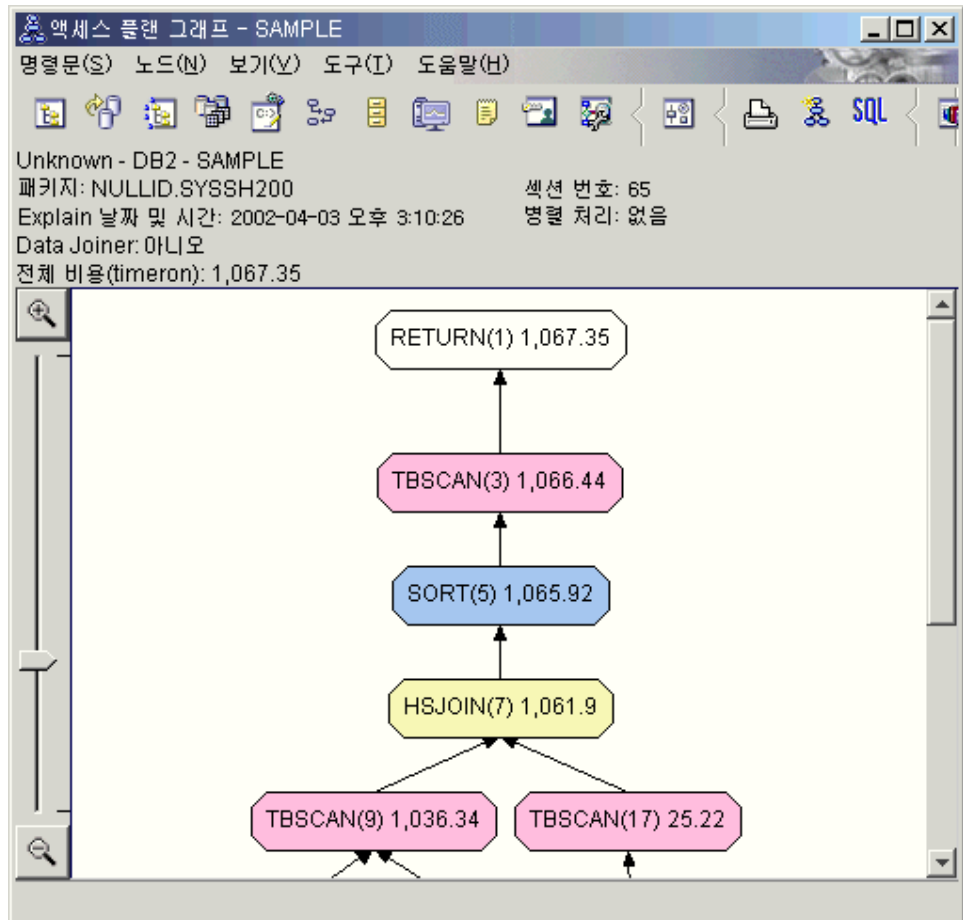
이 액세스 플랜에는 인덱스 스캔이 아니라 테이블 스캔이 포함되어 있습니다. 테이블 스캔은 팔각형으로 표시되며 59 페이지의 『TBSCAN 연산자』라는 레이블이 붙어 있습니다. 인덱스 스캔을 사용한 경우에는 다이아몬드로 표시되며 55 페이지의 『IXSCAN 연산자』이라는 레이블이 붙어 있습니다. 적은 데이터가 추출되고 있는 경우 테이블에 대해 작성된 인덱스를 사용하는 것이 테이블 스캔보다 비용면에서 효율적입니다.

3. 이 액세스 플랜이 얼마나 효과적입니까?

실제 통계를 사용하는 경우에만 액세스 플랜의 효과를 판별할 수 있습니다. 옵티마이저가 액세스 플랜의 디폴트 통계를 사용했으므로, 플랜이 얼마나 효과적인지 판별할 수 없습니다.

일반적으로 액세스 플랜에 대해 계산된 전체 69 페이지의 『비용』을 기록해 두면 나중에 개정된 액세스 플랜과 비교할 수 있습니다. 각 노드에 나열된 비용은 쿼리의 첫 단계에서 노드까지(경계 포함) 누적됩니다.

액세스 플랜 그래프 창에서 그래프 맨 위에 있는 **RETURN(1)**에 표시된 전체 비용은 대략 1,067timeron입니다. 계산된 전체 비용은 창의 맨 위 영역에도 표시됩니다.



다음 레슨 내용

쿼리 2로 이동합니다.

쿼리 2는 runstats가 실행된 후에 기본 쿼리에 대한 액세스 플랜을 조회합니다. runstats 명령을 사용하면 쿼리에서 액세스되는 모든 테이블의 현재 통계가 옵티마이저에 제공됩니다.

단일 파티션 데이터베이스 환경에서 runstats를 사용하여 테이블 및 인덱스에 대한 현재 통계 수집

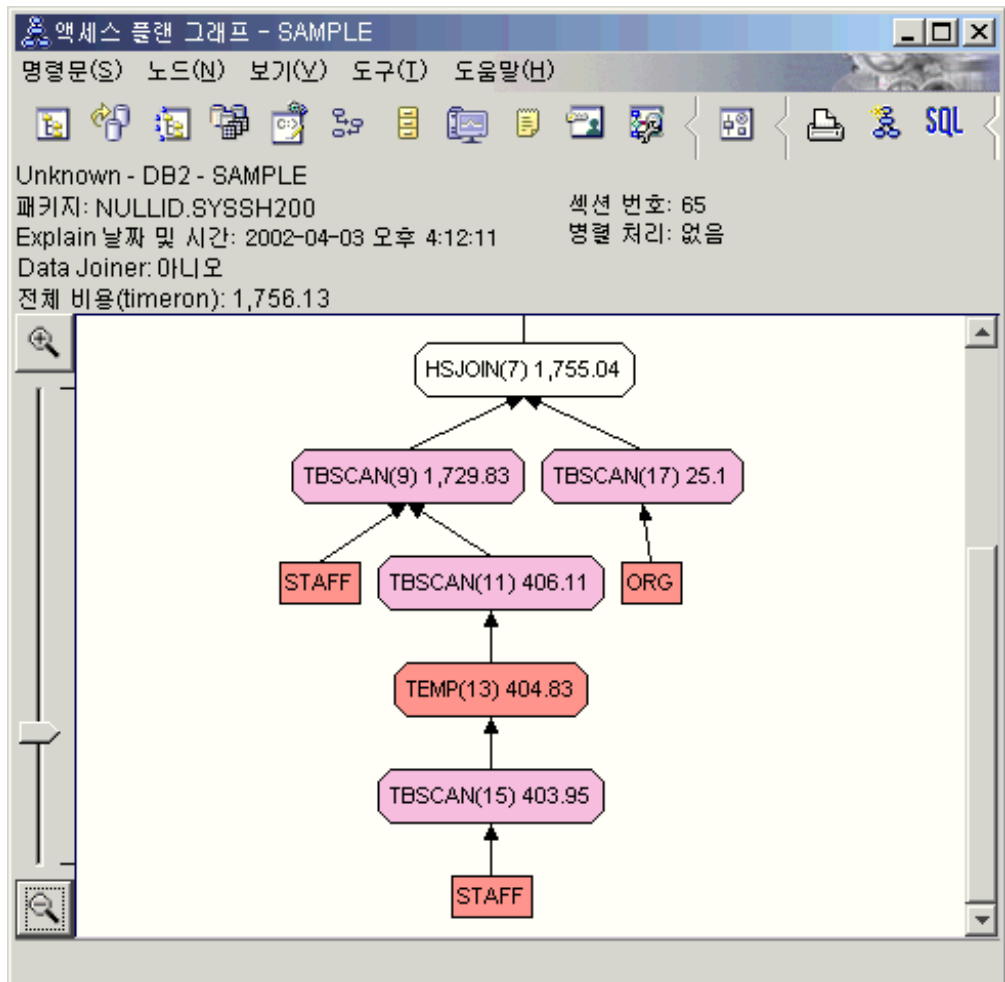
이 예는 runstats 명령으로 현재 통계를 수집하여 쿼리 1에 설명된 액세스 플랜에 빌드합니다.

runstats 명령을 사용하여 테이블 및 인덱스에 대한 현재 통계를 수집하는 것이 바람직 한데, 이는 마지막으로 runstats 명령을 실행한 이후 새 인덱스가 작성되었거나 중요한 갱신 활동이 발생한 경우에 특히 권장됩니다. 이는 옵티마이저에 최상의 액세스 플랜을

판별할 수 있는 가장 정확한 정보를 제공합니다. 현재 통계를 사용할 수 없으면, 옵티마이저는 정확하지 않은 디폴트 통계에 따라 비효율적인 액세스 플랜을 선택할 수 있습니다.

테이블을 갱신하지 않고 runstats를 사용하면 빈 테이블이 옵티마이저에 표시될 수 있습니다. 이러한 문제점은 연산자 세부사항 창에 대한 카디널리티가 0인 경우 입증됩니다. 이 경우, 테이블 갱신을 완료하고 runstats 명령을 재실행한 후 영향을 받는 테이블에 대한 Explain 스냅샷을 재작성하십시오.

Explain된 명령문 실행기록 창에서 이 쿼리(쿼리 2)에 대한 액세스 플랜 그래프를 보려면, 쿼리 번호 2로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.



다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 쿼리에 각 테이블에 대한 현재 통계가 존재합니까?

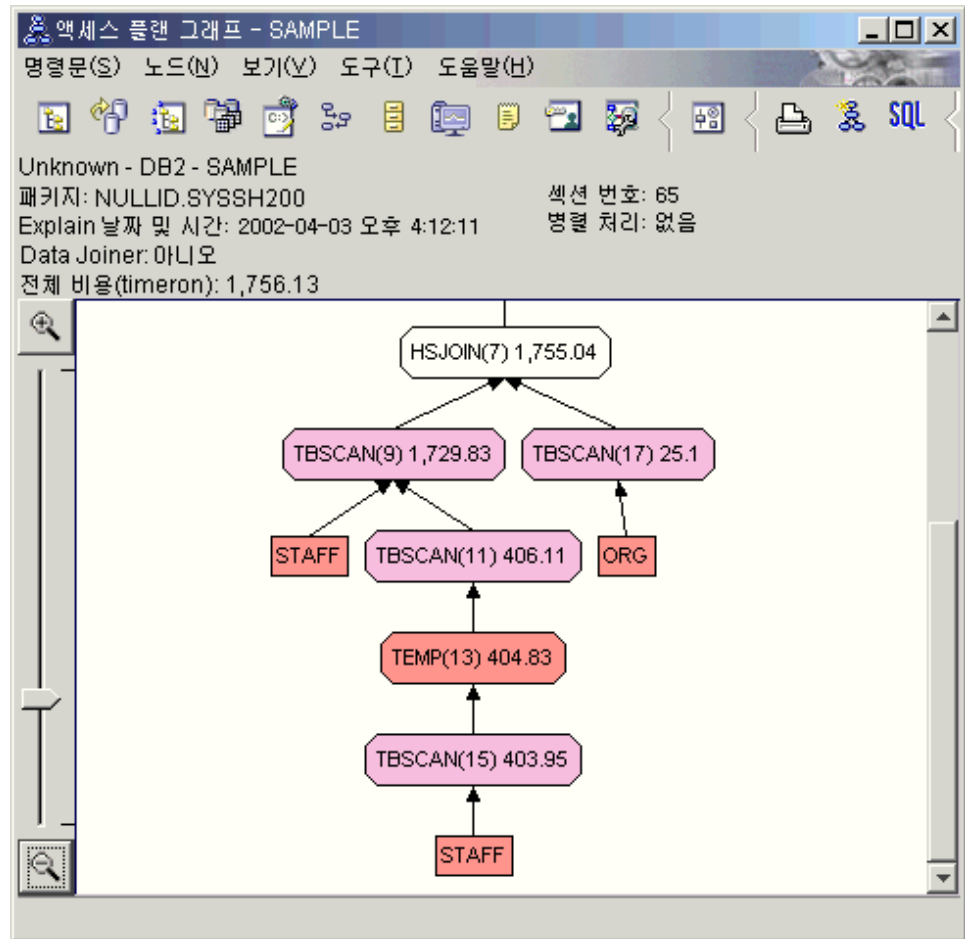
ORG 테이블에 대한 테이블 통계 창은 옵티마이저에 실제 통계가 사용되었음을 보여줍니다(STATS_TIME 값은 통계가 수집된 실제 시간입니다). 통계의 정확성은

runstats 명령이 실행된 이후에 테이블의 내용에 현저한 변경사항이 있는지 여부에 따라 다릅니다.

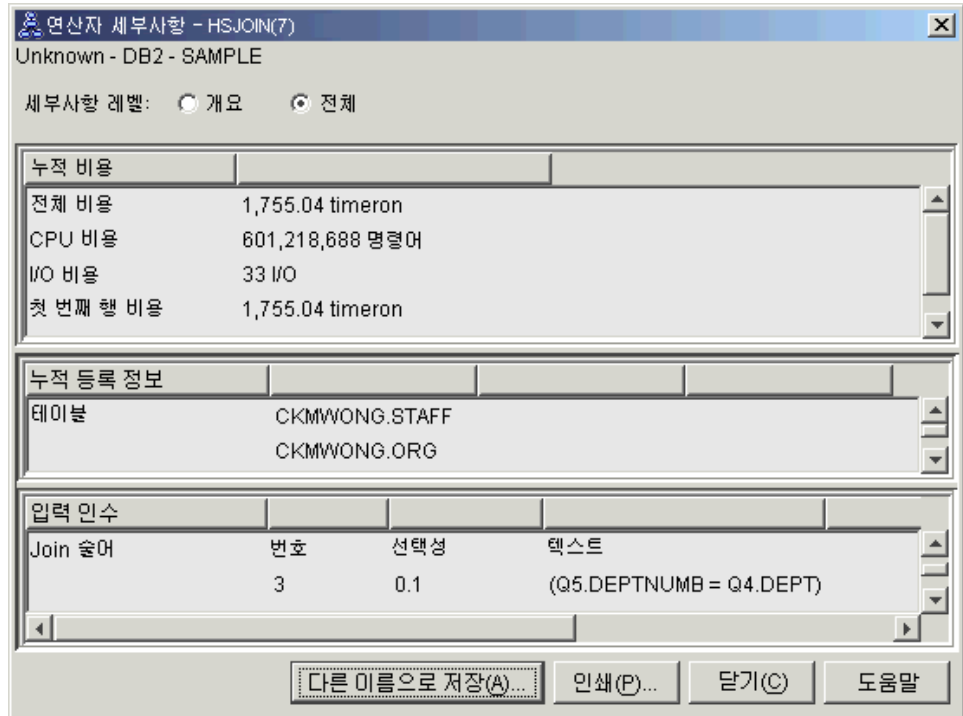
통계	Explain	현재
CREATE_TIME	2002-04-03 오후 3:05:03	2002-04-03 오후 3:05:03
STATS_TIME	2002-04-03 오후 4:12:05	2002-04-03 오후 4:12:05
CARD	8	8
NPAGES	1	1
FPAGES	1	1
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	아니오	아니오

2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

쿼리 1과 같이 쿼리 2의 액세스 플랜은 인덱스 스캔(55 페이지의 『IXSCAN 연산자』)이 아닌 테이블 스캔(59 페이지의 『TBSCAN 연산자』)을 사용합니다. 현재 통계가 존재하는 경우에도 인덱스 스캔은 쿼리에서 사용된 컬럼에 인덱스가 없으므로 수행되지 않았습니다. 테이블을 조인하는 데 사용되는 컬럼(즉, Join 74 페이지의 『술어』에서 사용되는 컬럼)의 인덱스를 옵티마이저에게 제공하면 쿼리를 향상시킬 수 있습니다. 이 예에서는 처음 병합 스캔 조인 HSJOIN(7)입니다.



HSJOIN(7) 연산자에 대한 연산자 세부사항 창에서 입력 인수 아래에 있는 **Join** 술어 섹션을 참조하십시오. 이 조인 조작에서 사용되는 컬럼은 텍스트 컬럼 아래에 나열되어 있습니다. 이 예에서, 이러한 컬럼은 DEPTNUMB 및 DEPT입니다.



3. 이 액세스 플랜이 얼마나 효과적입니까?

최신 통계에 기초한 액세스 플랜은 항상 실제로 계산된 비용(timeron 단위)을 생성합니다. 쿼리 1에서 계산된 비용이 디폴트 통계를 기본으로 하므로, 어느 것이 더 효과적인지 판별하기 위해 두 액세스 플랜 그래프의 비용을 비교할 수 없습니다. 비용이 높거나 낮은지 여부는 관련되지 않습니다. 유효한 효과 측정을 얻기 위해서는 실제 통계에 따라 액세스 플랜 비용을 비교해야 합니다.

다음 레슨 내용

쿼리 3으로 이동합니다.

쿼리 3에서는 DEPTNUMB 및 DEPT 컬럼에서 인덱스 추가 효과를 검색합니다. Join 술어에 사용되는 컬럼에 인덱스 추가는 성능을 향상시킬 수 있습니다.

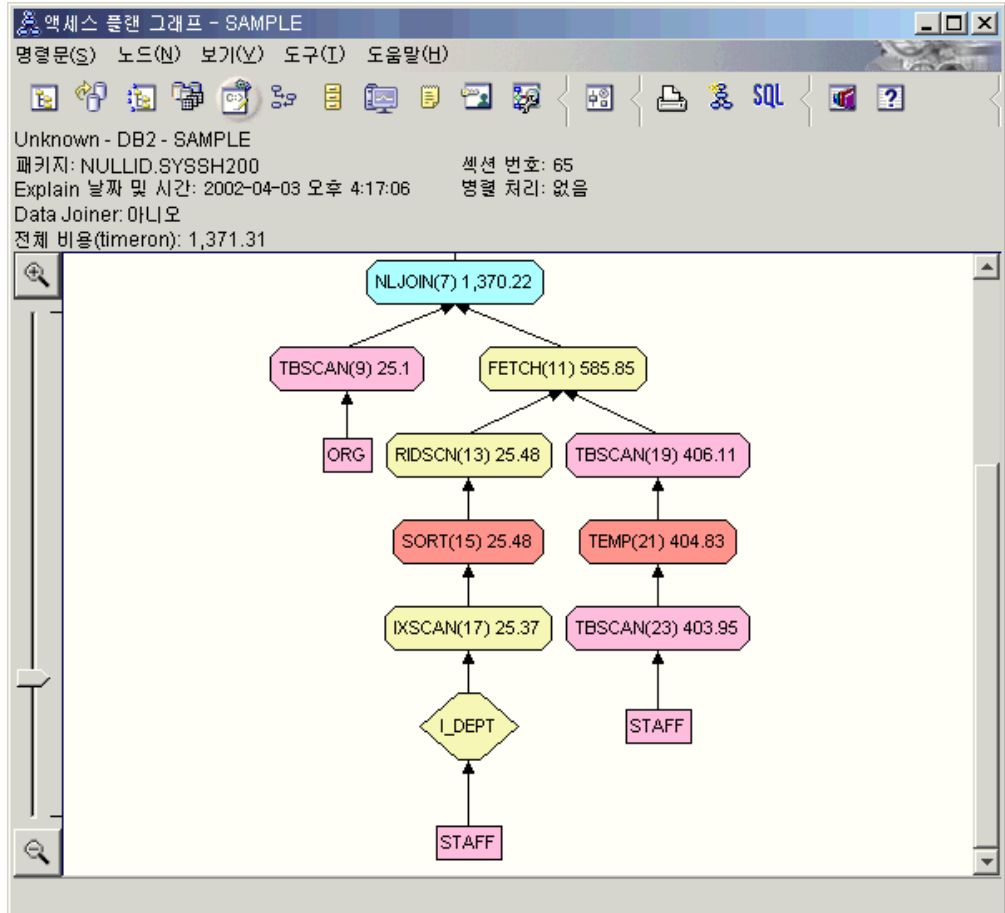
단일 파티션 데이터베이스 환경의 쿼리에서 테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성

이 예는 STAFF 테이블의 DEPT 컬럼과 ORG 테이블의 DEPTNUMB 컬럼에 인덱스를 작성하여 쿼리 2에 설명된 액세스 플랜에서 빌드합니다.

주: 권장 인덱스는 디자인 어드바이저를 사용하여 작성할 수 있습니다.

이 쿼리(쿼리 3)에 대한 액세스 플랜 그래프를 보려면 Explain된 명령문 실행기록 창에서 쿼리 번호 3으로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.

주: DEPTNUM에 대한 인덱스가 작성되었더라도 옵티마이저가 이를 사용하지 않았습니다.

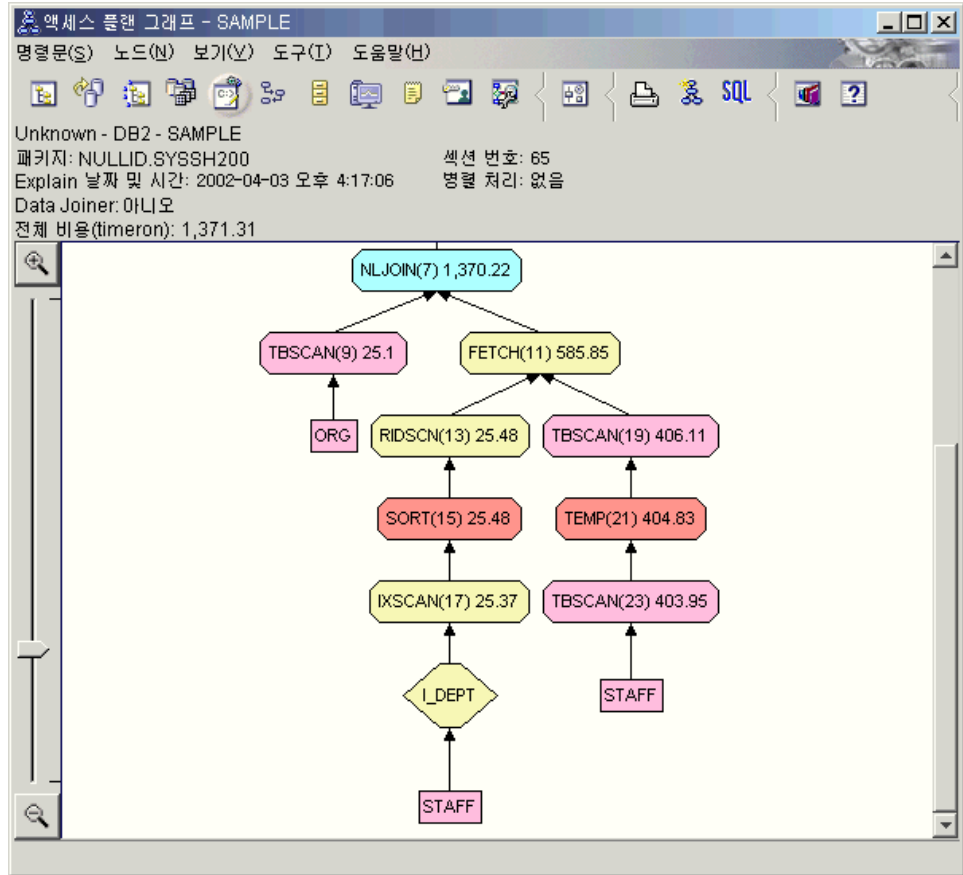


다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 인덱스가 있는 액세스 플랜에서 무엇이 변경되었습니까?

56 페이지의 『NLJOIN 연산자』 NLJOIN(7)은 쿼리 2에서 사용된 병합 스캔 조인 HSJOIN(7)을 대체했습니다. 중첩 루프 조인을 사용하면 이 유형의 조인에 정렬 또는 임시 테이블이 필요하지 않으므로 병합 스캔 조인보다 비용이 더 낮게 계산됩니다.

새 다이아몬드 모양의 노드인 I_DEPT가 STAFF 테이블 바로 위에 추가되었습니다. 이 노드는 DEPT에 작성된 인덱스를 나타내며, 검색할 행을 판별하기 위해 옵티마이저가 테이블 스캔 대신에 인덱스 스캔을 사용한다는 것을 보여줍니다.



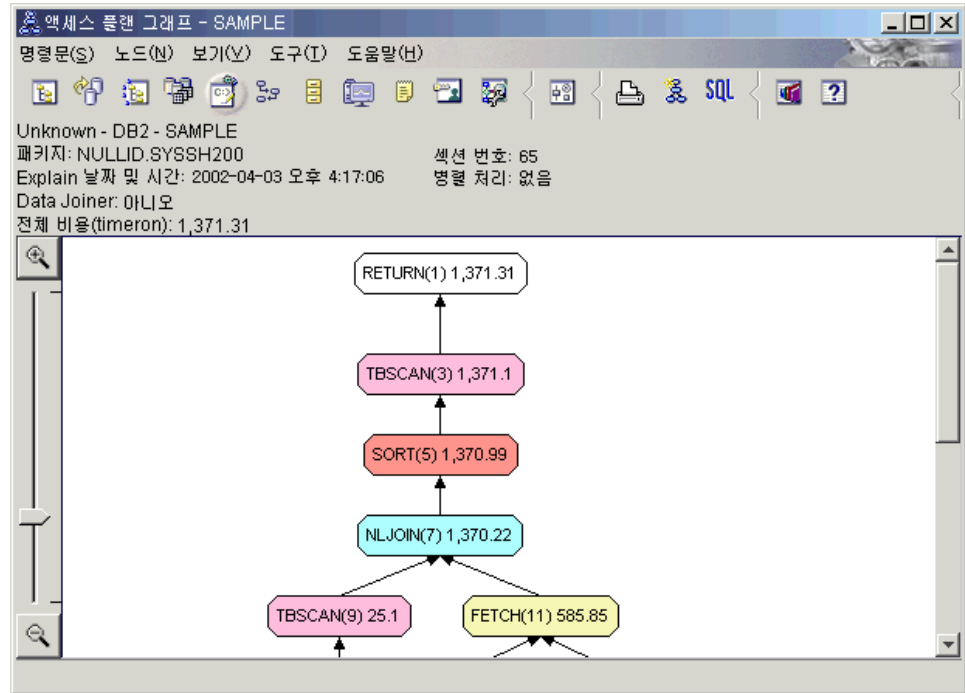
이 부분의 액세스 플랜 그래프에서는 새 인덱스(I_DEPT)가 DEPT 컬럼에 작성되었고 IXSCAN(17)이 STAFF 테이블을 액세스하는 데 사용되었다는 점에 유의하십시오. 쿼리 2에서는 테이블 스캔이 STAFF 테이블을 액세스하는 데 사용되었습니다.

2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

인덱스를 추가한 결과, 55 페이지의 『IXSCAN 연산자』 노드인 IXSCAN(17)이 STAFF 테이블에 액세스하는 데 사용되었습니다. 쿼리 2에는 인덱스가 없으므로 이 예에서 테이블 스캔이 사용되었습니다.

52 페이지의 『FETCH 연산자』 노드인 FETCH(11)를 보면 옵티마이저가 인덱스 스캔을 사용하여 컬럼 DEPT를 검색할 뿐만 아니라 인덱스를 포인터로 사용하여 STAFF 테이블에서 추가 컬럼을 검색했음을 알 수 있습니다. 이 경우, 인덱스 스캔과 페치의 조합은 이전 액세스 플랜에서 사용된 전체 테이블 스캔보다 비용이 덜 드는 것으로 계산되었습니다.

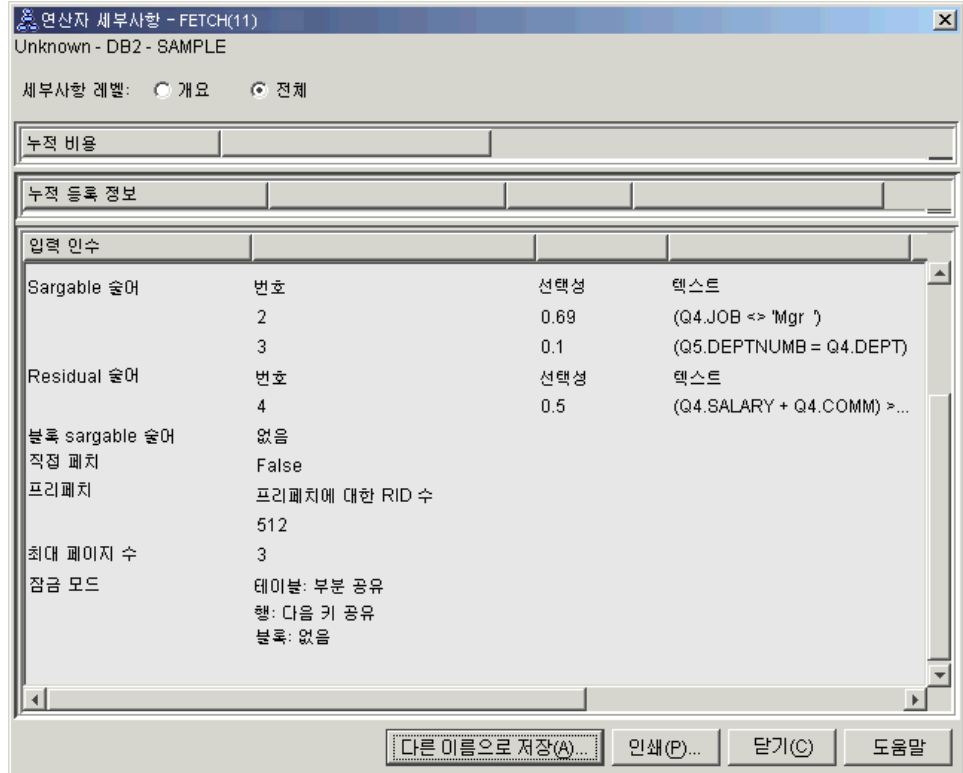
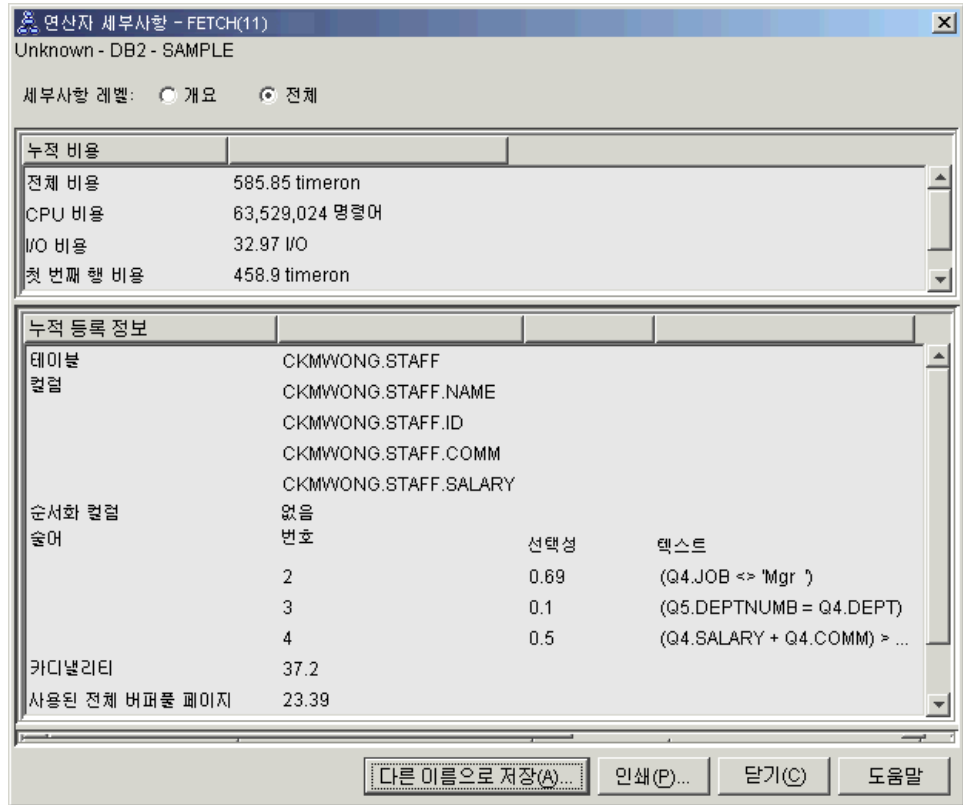
주: STAFF 테이블의 노드는 두 번 표시되어 DEPT에 대한 인덱스 및 FETCH 조작 사이의 관계를 보여줍니다.



이 조회에 대한 액세스 플랜은 Join 술어에 관련된 컬럼에서의 인덱스 작성 효과를 보여줍니다. 인덱스는 로컬 술어의 응용프로그램 속도를 높일 수도 있습니다. 로컬 술어에서 참조된 컬럼에 인덱스를 추가하는 것이 액세스 플랜에 어떤 영향을 줄 수 있는지를 알기 위해 이 조회의 각 테이블마다 로컬 술어를 살펴보겠습니다.

FETCH(11) 연산자에 대한 연산자 세부사항 창에서 누적 등록 정보 아래에 있는 컬럼을 참조하십시오. 이 페치 조작에 대해 술어에 사용된 컬럼은 술어 섹션에 표시된 대로 JOB입니다.

주: 이 술어의 선택성은 .69입니다. 이는 이 술어를 사용하면 행의 69%가 다음 처리에 선택될 것임을 의미합니다.



FETCH(11) 연산자에 대한 연산자 세부사항 창에는 이 조작에 사용되고 있는 컬럼이 표시됩니다. DEPTNAME이 입력 인수 아래 검색된 컬럼 옆에 있는 첫 행에 나열되어 있습니다.

3. 이 액세스 플랜이 얼마나 효과적입니까?

이 액세스 플랜은 이전 예의 액세스 플랜보다 비용면에서 더 효율적입니다. 누적 비용은 쿼리 2의 대략 1,755timeron에서 쿼리 3의 대략 959timeron으로 줄었습니다.

그러나 쿼리 3의 액세스 플랜은 STAFF 테이블의 인덱스 스캔 IXSCAN(17)과 FETCH(11)을 보여줍니다. 페치 조작과 조합된 인덱스 스캔은 전체 테이블 스캔보다 비용이 덜 들지만, 이는 검색되는 행마다 테이블이 한 번 액세스되고 인덱스가 한 번 액세스된다는 것을 의미합니다. STAFF 테이블에서의 이러한 이중 액세스를 줄여봅시다.

다음 레슨 내용

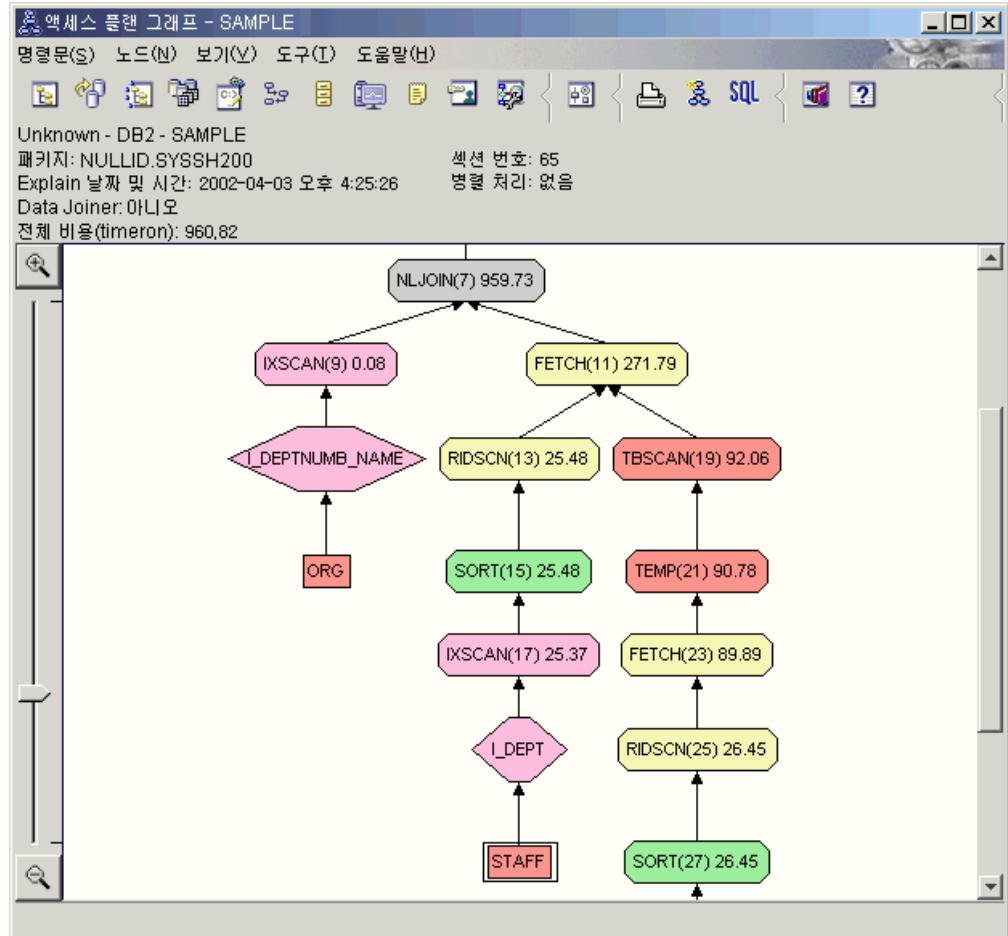
쿼리 4로 이동합니다.

쿼리 4는 페치 및 인덱스 스캔을 페치가 없는 단일 인덱스 스캔으로 줄입니다. 추가 인덱스를 작성하면 액세스 플랜에 대해 계산된 비용이 줄어들 수 있습니다.

단일 파티션 데이터베이스 환경에서 테이블 컬럼에 추가 인덱스 작성

이 예는 STAFF 테이블에서 JOB 컬럼에 인덱스를 작성하고 ORG 테이블에서 기존 인덱스에 DEPTNAME을 추가하여 쿼리 3에서 설명한 액세스 플랜에서 빌드합니다. (별도의 인덱스를 추가하면, 추가 액세스가 발생합니다.)

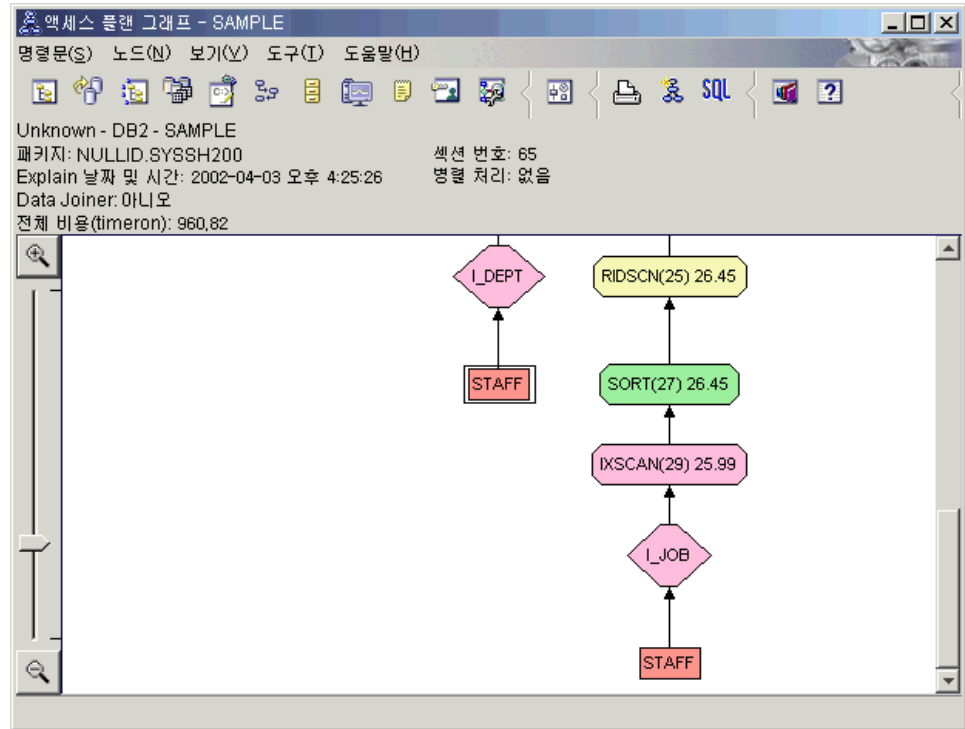
이 쿼리(쿼리 4)에 대한 액세스 플랜 그래프를 보려면 Explain된 명령문 실행기록 창에서 쿼리 번호 4로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.



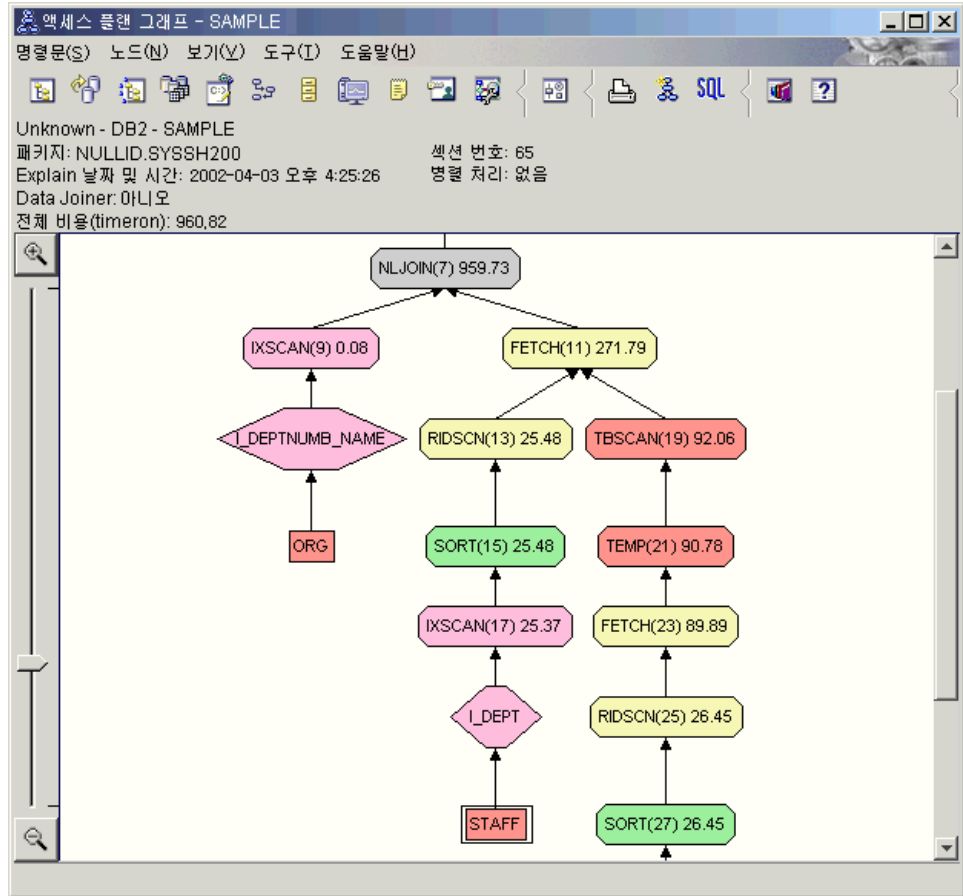
다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 추가 인덱스 작성 결과로 이 액세스 플랜을 어떻게 변경했습니까?

옵티마이저가 나중에 이 액세스 플랜을 더 세분화하기 위해 STAFF 테이블(I_JOB 레이블이 붙은 다이어몬드로 표시)에서 JOB 컬럼에 작성된 인덱스를 이용했습니다.



액세스 플랜 그래프 중간 부분에서, ORG 테이블의 경우 이전 인덱스 스캔 및 페치가 인덱스 스캔인 IXSCAN(9)로 변경되었다는 점에 유의하십시오. ORG 테이블의 인덱스에 DEPTNAME 컬럼을 추가하면 옵티마이저가 페치에 관련된 여분의 액세스를 제거하게 할 수 있습니다.



2. 이 액세스 플랜이 얼마나 효과적입니까?

이 액세스 플랜은 이전 예의 액세스 플랜보다 비용면에서 더 효율적입니다. 누적 비용은 쿼리 3의 대략 1,370timeron에서 쿼리 4의 대략 959timeron으로 줄었습니다.

다음 레슨 내용

SQL 또는 XQuery문의 성능이 향상됩니다.

성능 향상을 위한 추가 단계의 세부 정보를 찾으려면 DB2 정보 센터를 참조하십시오. Visual Explain으로 돌아가면 조치 효과에 액세스할 수 있습니다.

제 4 장 레슨 4. 파티션된 데이터베이스 환경에서 액세스 플랜 향상

다양한 성능 조정 활동을 수행할 때 기본 쿼리에 대한 액세스 플랜 및 관련 창이 어떻게 변경되는지 학습합니다.

그림과 함께 제공되는 일련의 예를 사용하여, **runstats** 명령을 사용하고 해당 인덱스를 추가함으로써 단순 쿼리의 액세스 플랜에 대해 계산된 전체 비용이 향상될 수 있는 방법에 대해 배우게 됩니다.

Visual Explain을 이해할수록, 쿼리 성능을 조정하는 또 다른 방법을 발견하게 됩니다.

액세스 플랜 그래프로 작업

네 개의 샘플 Explain 스냅샷을 예로 들어 성능 조정이 어떻게 데이터베이스 성능의 중요한 부분이 되는지 학습합니다.

Explain 스냅샷과 연관된 쿼리는 1 - 4로 번호가 지정되어 있습니다. 각 쿼리는 동일한 SQL 또는 XQuery문(레슨 1에서 설명)을 사용합니다.

```
SELECT S.ID,SNAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL ( SELECT ST.SALARY*.9
                          FROM STAFF ST
                          WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

그러나 반복되는 각 쿼리는 이전 실행보다 더 많은 성능 조정 기술을 사용합니다. 예를 들면, 쿼리 1에는 성능 조정이 없는 반면, 쿼리 4에는 가장 많습니다. 쿼리의 차이점은 다음에 설명되어 있습니다.

쿼리 1

인덱스 및 통계가 없는 쿼리 실행

쿼리 2

쿼리에 테이블 및 인덱스에 대한 현재 통계 수집

쿼리 3

쿼리에서 테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성

쿼리 4

테이블 컬럼에 추가 인덱스 작성

이 예는 파티션간 병렬 처리를 사용하여 실제 노드가 7개인 RS/6000® SP™ 머신에서 생성되었습니다.

파티션된 데이터베이스 환경에서 인덱스와 통계 없이 쿼리 실행

이 예에서 액세스 플랜은 인덱스와 통계 없이 SQL 쿼리에 대해 작성되었습니다.

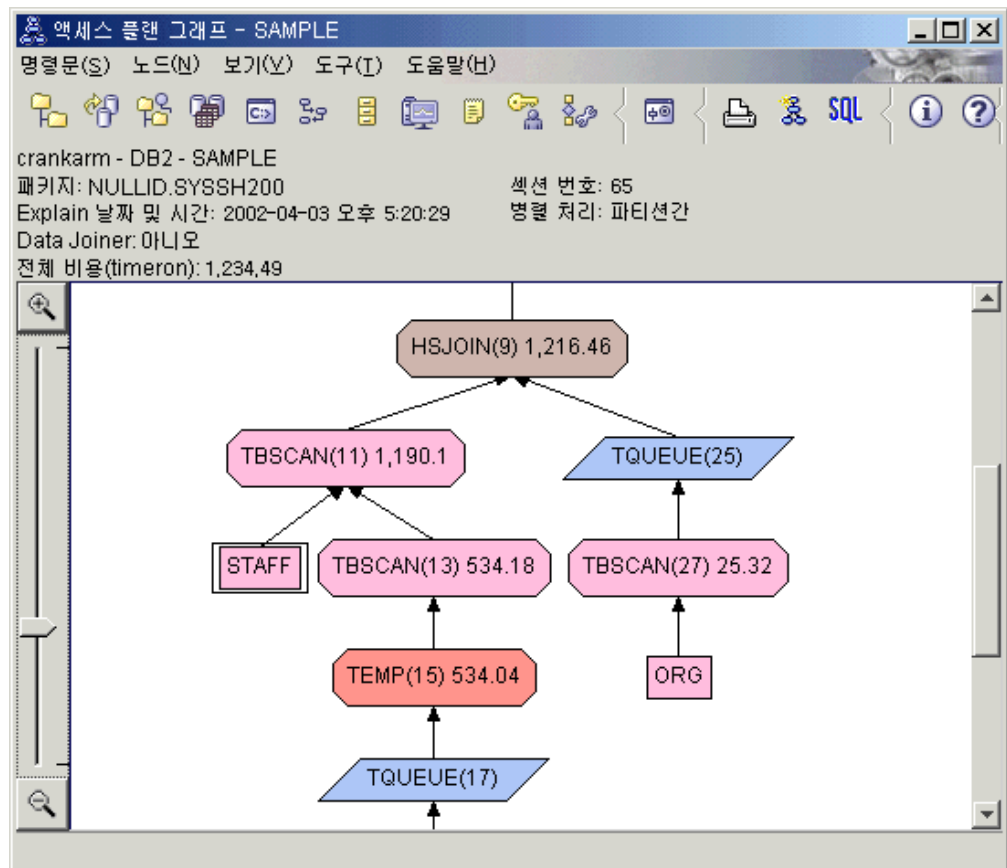
이 쿼리(쿼리 1)에 대한 액세스 플랜 그래프를 보려면 다음을 수행하십시오.

1. 제어 센터에서 샘플 데이터베이스를 찾을 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain**된 명령문 실행기록 표시를 선택하십시오.

Explain된 명령문 실행기록 창이 열립니다.

3. 쿼리 번호 1로 식별되는 항목을 더블 클릭하십시오.(쿼리 번호 컬럼을 찾기 위해 오른쪽으로 스크롤해야 할 수 있습니다.)

명령문에 대한 액세스 플랜 그래프 창이 열립니다.



다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 쿼리에 각 테이블에 대한 현재 통계가 존재합니까?

쿼리에 각 테이블에 대한 현재 통계가 존재하는지 점검하려면, 액세스 플랜 그래프에서 각 테이블 노드를 더블 클릭하십시오. 열린 테이블 통계 창에는 **Explain** 컬럼 아래의 **STATS_TIME** 행에 스냅샷 작성 시 통계가 수집되지 않았음을 알리는 "통계 갱신 안됨"이 표시됩니다.

현재 통계가 없으면 옵티마이저는 실제 통계와 다를 수도 있는 디폴트 통계를 사용합니다. 디폴트 통계는 테이블 통계 창의 **Explain** 컬럼 아래에 "디폴트"라는 단어로 식별됩니다.

ORG 테이블에 대한 테이블 통계 창의 정보에 따르면, 옵티마이저는 디폴트 통계 (Explain 값 옆에 표시)를 사용했습니다. 디폴트 통계는 스냅샷 작성 시 실제 통계를 사용할 수 없으므로 사용되었습니다(STATS_TIME 행에 표시된 대로).

통계	Explain	현재
CREATE_TIME	2002-03-26 오후 1:35:42	2002-03-26 오후 1:35:42
STATS_TIME	통계 갱신 안됨	통계 갱신 안됨
CARD	55(디폴트)	-1
NPAGES	1(디폴트)	-1
FPAGES	1(디폴트)	-1
COLCOUNT	5(디폴트)	5
OVERFLOW	0(디폴트)	-1
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	아니오(디폴트)	아니오

2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

이 액세스 플랜에는 인덱스 스캔이 아니라 테이블 스캔이 포함되어 있습니다. 테이블 스캔은 팔각형으로 표시되며 여기에 TBSCAN 연산자라고 레이블됩니다. 인덱스 스캔을 사용한 경우에는 다이아몬드로 표시되며 여기에 IXSCAN이라고 레이블됩니다. 적은 데이터가 추출되고 있는 경우 테이블에 대해 작성된 인덱스를 사용하는 것이 테이블 스캔보다 비용면에서 효율적입니다.

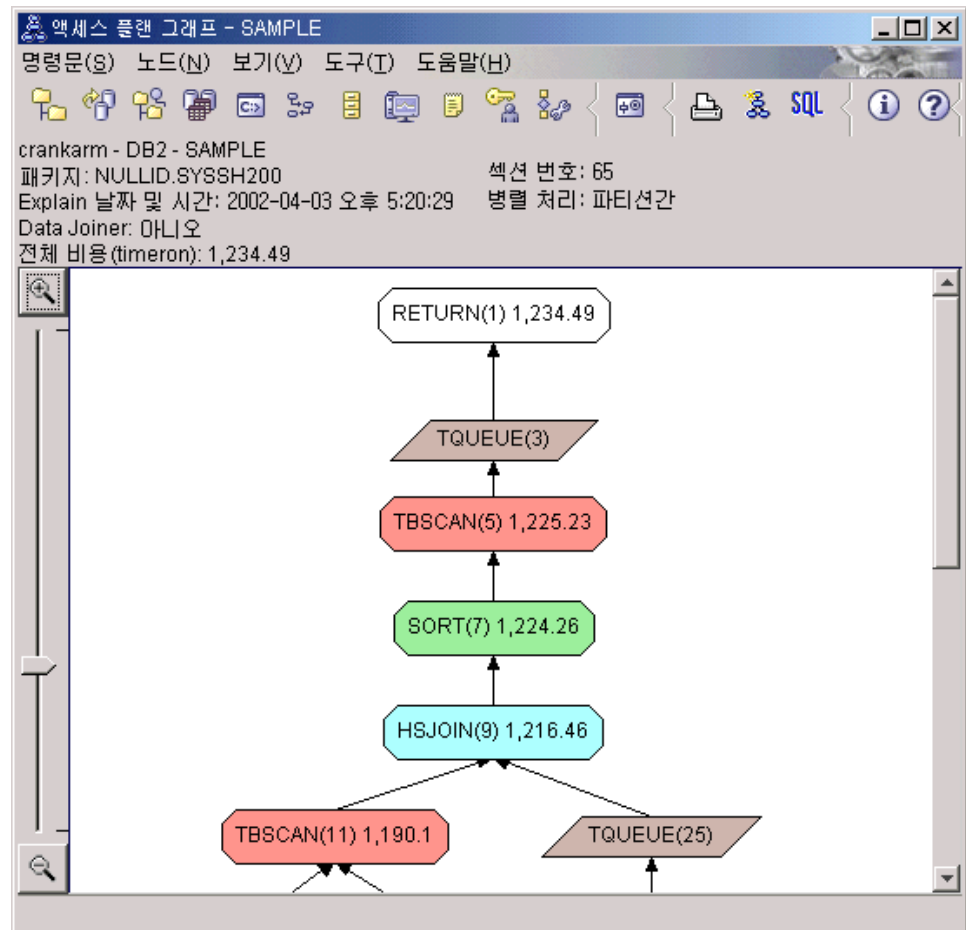
3. 이 플랜이 얼마나 효과적입니까?

실제 통계를 사용하는 경우에만 액세스 플랜의 효과를 판별할 수 있습니다. 옵티마이저가 액세스 플랜의 디폴트 통계를 사용했으므로, 플랜이 얼마나 효과적인지 판별할 수 없습니다.

일반적으로 액세스 플랜에 대해 계산된 전체 비용을 기록해 두면 나중에 개정된 액세스 플랜과 비교할 수 있습니다. 각 노드에 나열된 비용은 쿼리의 첫 단계에서 노드까지(경계 포함) 누적됩니다.

주: 파티션된 데이터베이스의 경우, 이는 대부분의 자원을 사용하는 노드의 누적 비용입니다.

엑세스 플랜 그래프 창에서 그래프 맨 위에 있는 **RETURN(1)**에 표시된 전체 비용은 대략 1,234timeron입니다. 계산된 전체 비용은 창의 맨 위에도 표시됩니다.



다음 레슨 내용

쿼리 2로 이동합니다.

쿼리 2는 runstats가 실행된 후에 기본 쿼리에 대한 액세스 플랜을 조회합니다. runstats 명령을 사용하면 쿼리에서 액세스되는 모든 테이블의 현재 통계가 옵티마이저에 제공됩니다.

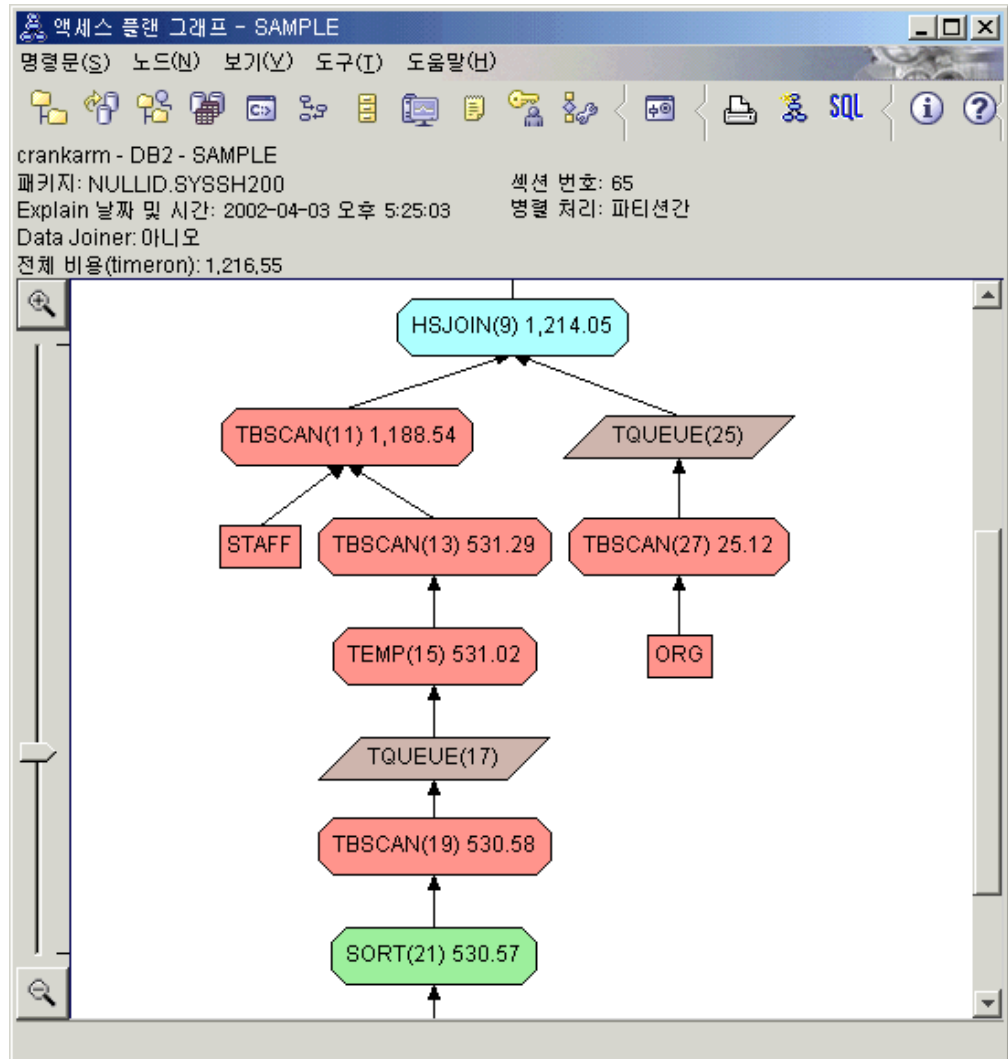
파티션된 데이터베이스 환경에서 runstats를 사용하여 테이블 및 인덱스에 대한 현재 통계 수집

이 예는 runstats 명령으로 현재 통계를 수집하여 쿼리 1에 설명된 액세스 플랜에 빌드합니다.

runstats 명령을 사용하여 테이블 및 인덱스에 대한 현재 통계를 수집하는 것이 바람직 한데, 이는 마지막으로 runstats 명령을 실행한 이후 새 인덱스가 작성되었거나 중요한 갱신 활동이 발생한 경우에 특히 권장됩니다. 이는 옵티마이저에 최상의 액세스 플랜을 판별할 수 있는 가장 정확한 정보를 제공합니다. 현재 통계를 사용할 수 없으면, 옵티 마이저는 정확하지 않은 디폴트 통계에 따라 비효율적인 액세스 플랜을 선택할 수 있 습니다.

테이블을 갱신하지 않고 runstats를 사용하면 빈 테이블이 옵티마이저에 표시될 수 있 습니다. 이러한 문제점은 연산자 세부사항 창에 대한 카디널리티가 0인 경우 입증됩니 다. 이 경우, 테이블 갱신을 완료하고 runstats 명령을 재실행한 후 영향을 받는 테이블 에 대한 Explain 스냅샷을 재작성하십시오.

이 쿼리(쿼리 2)에 대한 액세스 플랜 그래프를 보려면 Explain문 실행기록 창에서 쿼 리 번호 2로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.



다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

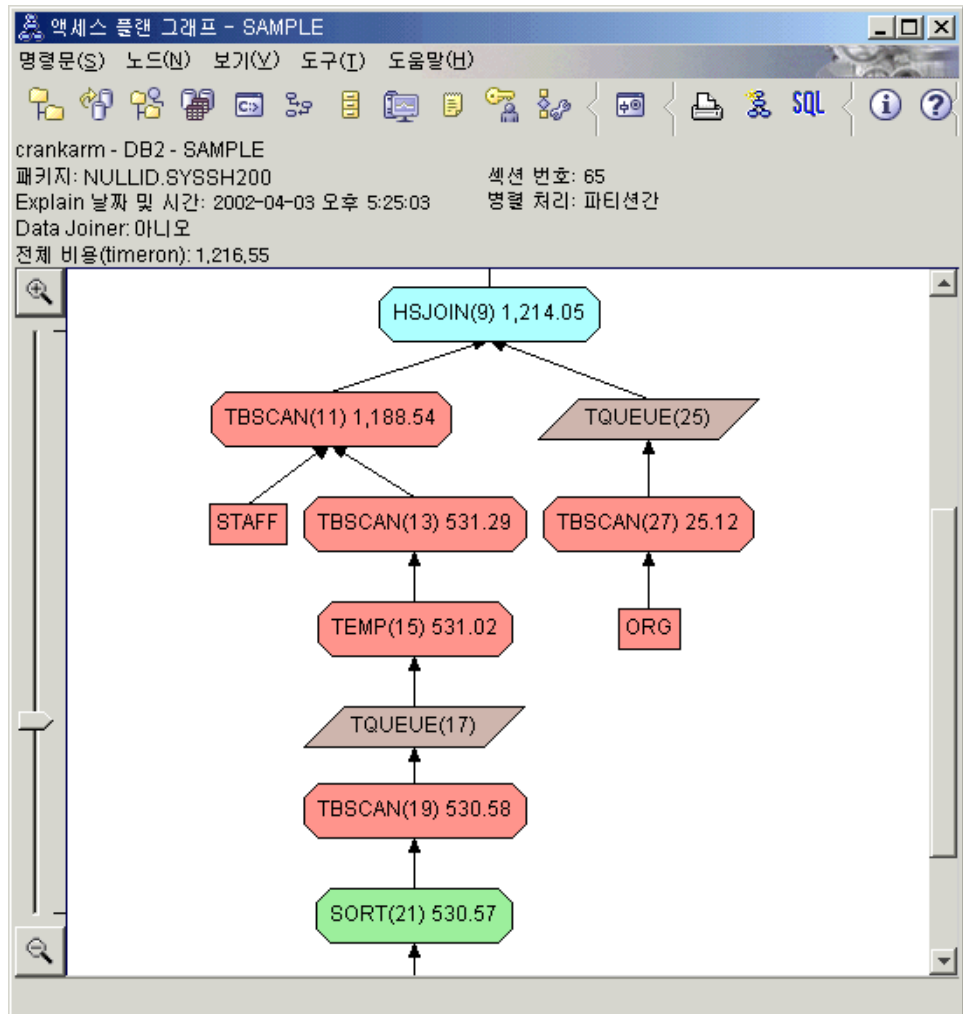
1. 쿼리에 각 테이블에 대한 현재 통계가 존재합니까?

ORG 테이블에 대한 테이블 통계 창은 옵티마이저에 실제 통계가 사용되었음을 보여줍니다(STATS_TIME 값은 통계가 수집된 실제 시간입니다). 통계의 정확성은 runstats 명령이 실행된 이후에 테이블 내용이 현저하게 변경되었는지에 따라 다릅니다.

통계	Explain	현재
CREATE_TIME	2002-03-26 오후 1:36:42	2002-03-26 오후 1:36:42
STATS_TIME	2002-04-03 오후 5:24:55	2002-04-03 오후 5:24:55
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	아니오	아니오

2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

쿼리 1과 같이 쿼리 2의 액세스 플랜은 테이블 스캔(TBSCAN 연산자)을 사용하고 인덱스 스캔을 사용하지 않습니다(IXSCAN). 현재 통계가 존재하는 경우에도 인덱스 스캔은 쿼리에서 사용된 컬럼에 인덱스가 없으므로 수행되지 않았습니다. 쿼리를 향상시키는 방법은 테이블을 조인하는 데 사용되는 컬럼(즉, Join 술어에서 사용되는 컬럼)에서 인덱스가 있는 옵티마이저를 제공하는 것입니다. 이 예에서는 처음 병합 스캔 조인 HSJOIN(9)입니다.



HSJOIN(9) 연산자에 대한 연산자 세부사항 창에서 입력 인수 아래에 있는 **Join** 술어 섹션을 참조하십시오. 이 조인 조작에서 사용되는 컬럼은 텍스트 컬럼 아래에 나열되어 있습니다. 이 예에서, 이러한 컬럼은 DEPTNUMB 및 DEPT입니다.



3. 이 액세스 플랜이 얼마나 효과적입니까?

최신 통계에 기초한 액세스 플랜은 항상 실제로 계산된 비용(timeron 단위)을 생성합니다. 쿼리 1에서 계산된 비용이 디폴트 통계를 기본으로 하므로, 어느 것이 더 효과적인지 판별하기 위해 두 액세스 플랜 그래프의 비용을 비교할 수 없습니다. 비용이 높거나 낮은지 여부는 관련되지 않습니다. 유효한 효과 측정을 얻기 위해서는 실제 통계에 따라 액세스 플랜 비용을 비교해야 합니다.

다음 레슨 내용

쿼리 3으로 이동합니다.

쿼리 3에서는 DEPTNUMB 및 DEPT 컬럼에서 인덱스 추가 효과를 검색합니다. Join 술어에 사용되는 컬럼에 인덱스 추가는 성능을 향상시킬 수 있습니다.

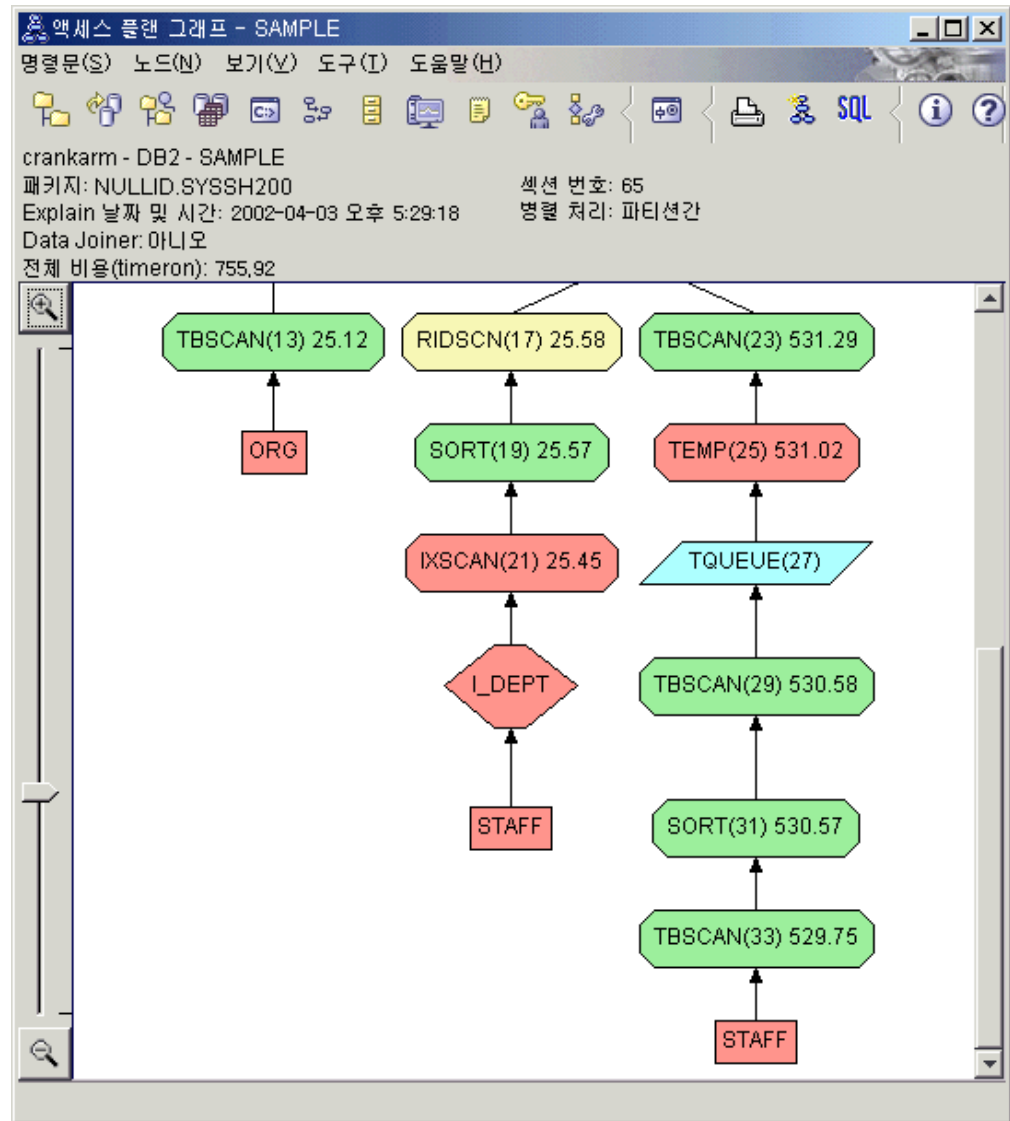
파티션된 데이터베이스 환경의 쿼리에서 테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성

이 예는 STAFF 테이블의 DEPT 컬럼과 ORG 테이블의 DEPTNUMB 컬럼에 인덱스를 작성하여 쿼리 2에 설명된 액세스 플랜에서 빌드합니다.

주: 권장 인덱스는 디자인 어드바이저를 사용하여 작성할 수 있습니다.

이 쿼리(쿼리 3)에 대한 액세스 플랜 그래프를 보려면 Explain된 명령문 실행기록 창에서 쿼리 번호 3으로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.

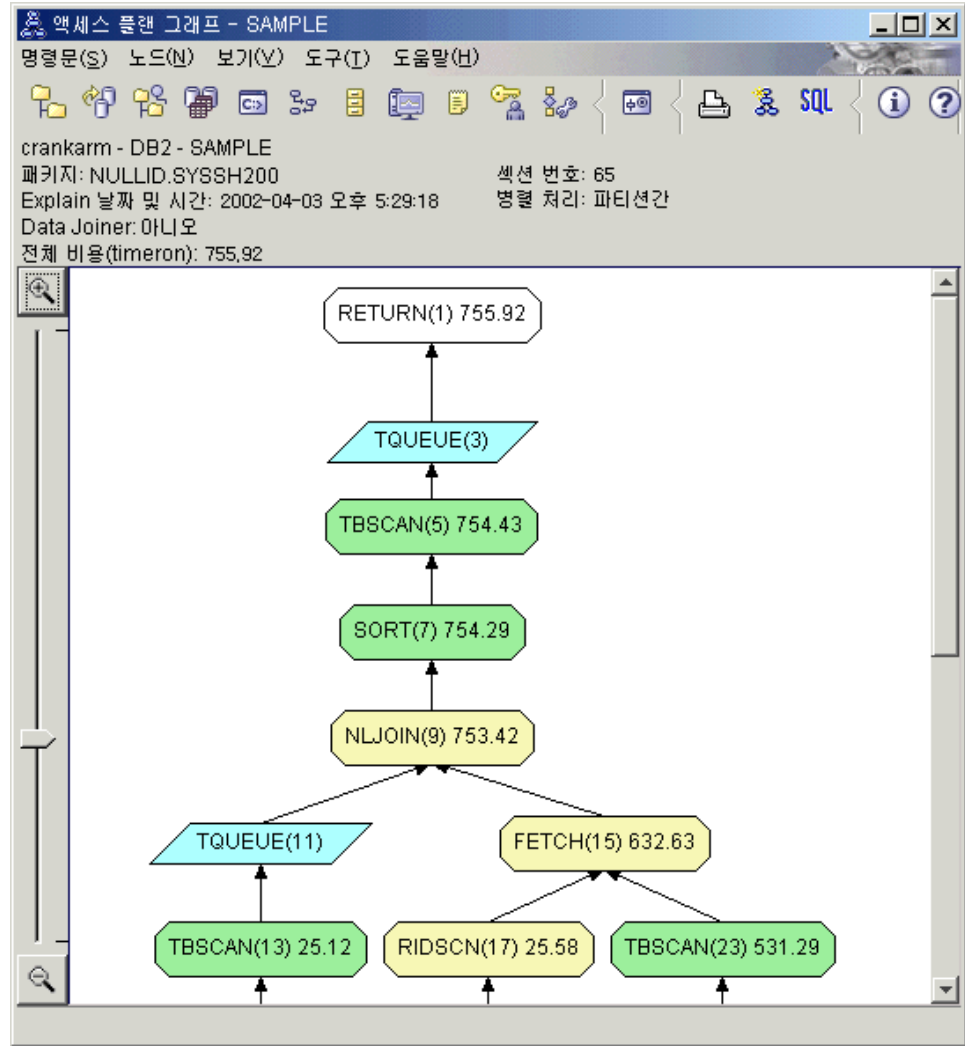
주: DEPTNUM에 대한 인덱스가 작성되었더라도 옵티마이저가 이를 사용하지 않았습니다.



다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

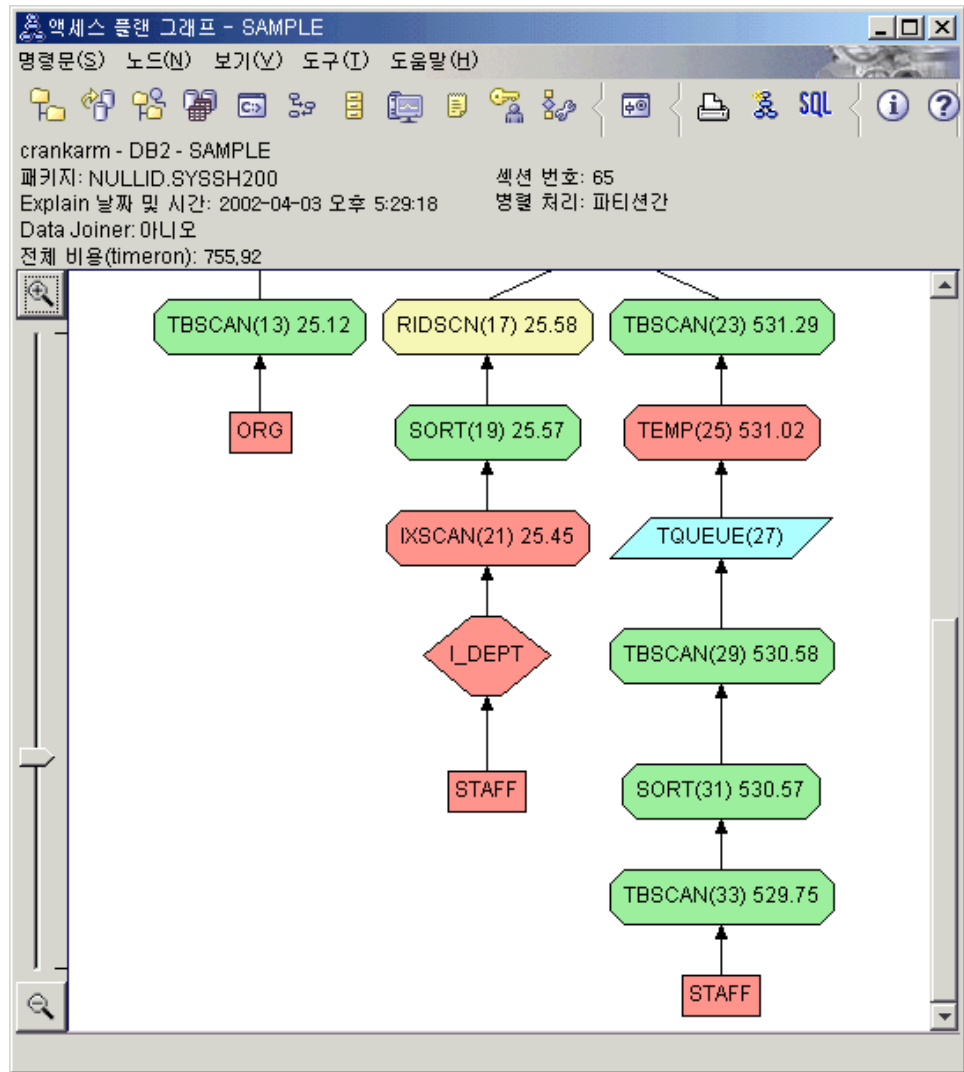
1. 인덱스가 있는 액세스 플랜에서 무엇이 변경되었습니까?

새 다이아몬드 모양의 노드인 **I_DEPT**가 **STAFF** 테이블 바로 위에 추가되었습니다. 이 노드는 **DEPT**에 작성된 인덱스를 나타내며, 옵티마이저는 테이블 스캔 대신에 인덱스 스캔을 사용하여 검색할 행을 판별하고 있습니다.

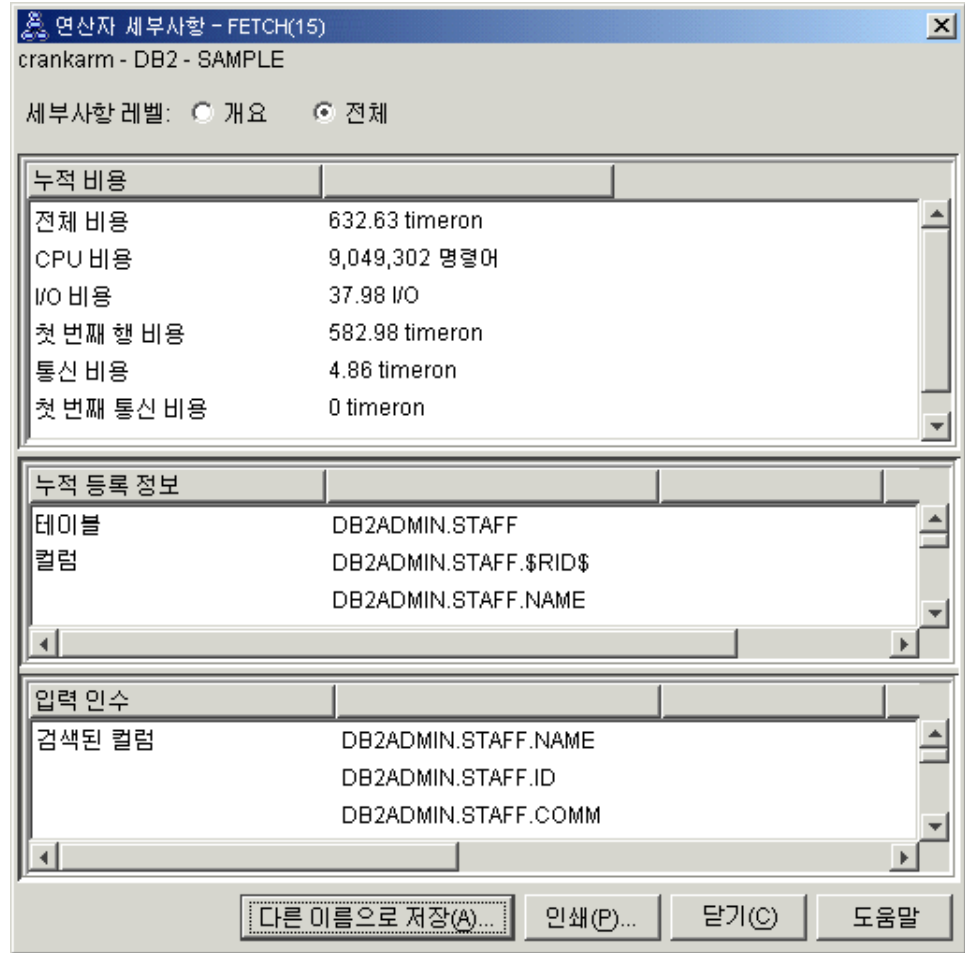


2. 이 액세스 플랜이 데이터 액세스에 가장 효과적인 메소드를 사용합니까?

이 쿼리에 대한 액세스 플랜은 결과가 FETCH(15) 및 IXSCAN(21)인 ORG 테이블의 DEPTNUMB 컬럼과 STAFF 테이블의 DEPT 컬럼에 대한 인덱스 작성 효과를 보여줍니다. 쿼리 2에 이 인덱스가 없으므로 예에서는 테이블 스캔이 사용되었습니다.



FETCH(15) 연산자에 대한 연산자 세부사항 창에는 이 조작에 사용되고 있는 컬럼이 표시됩니다.



인덱스와 페치의 조합은 이전 액세스 플랜에서 사용된 전체 테이블 스캔보다 비용이 덜 드는 것으로 계산되었습니다.

3. 이 액세스 플랜이 얼마나 효과적입니까?

이 액세스 플랜은 이전 예의 액세스 플랜보다 비용면에서 더 효율적입니다. 누적 비용은 쿼리 2의 대략 1,214timeron에서 쿼리 3의 대략 755timeron으로 줄었습니다.

다음 레슨 내용

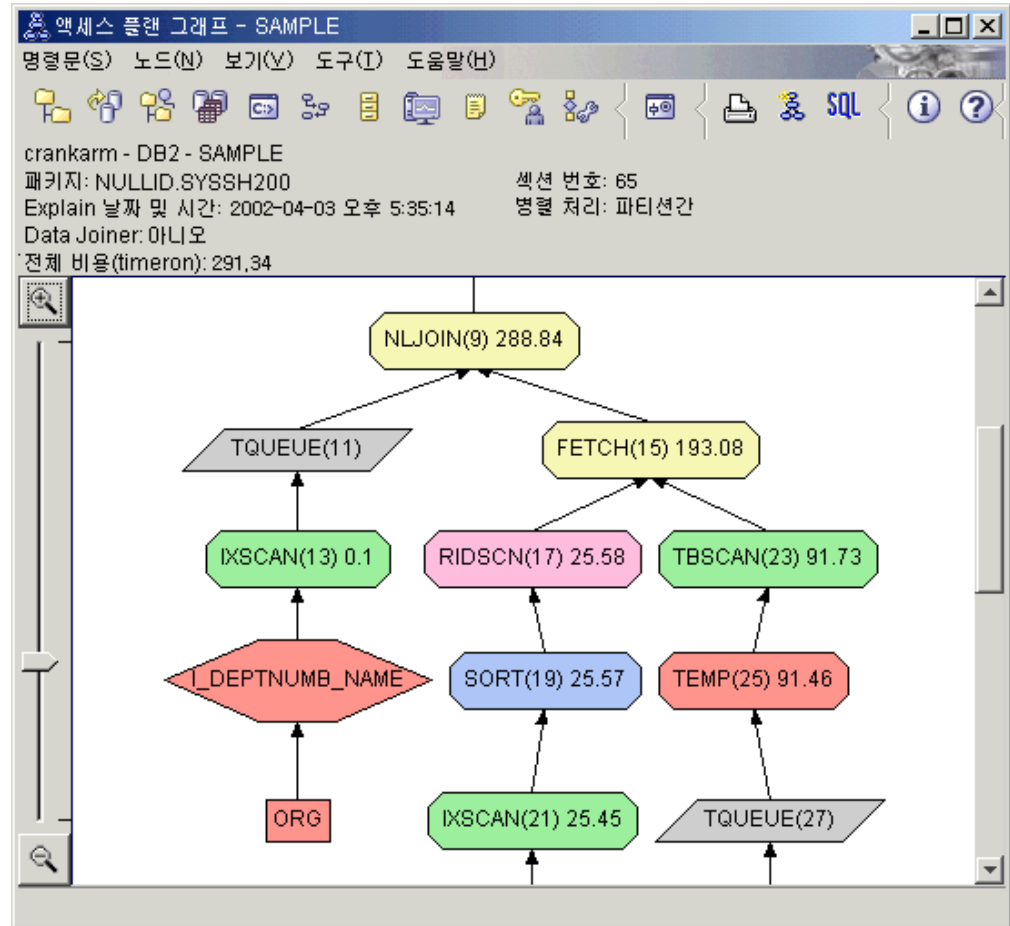
쿼리 4로 이동합니다.

쿼리 4는 페치 및 인덱스 스캔을 페치가 없는 단일 인덱스 스캔으로 줄입니다. 인덱스를 추가로 작성하면 액세스 플랜에 대해 계산된 비용이 줄어들 수 있습니다.

파티션된 데이터베이스 환경에서 테이블 컬럼에 추가 인덱스 작성

이 예는 STAFF 테이블에서 JOB 컬럼에 인덱스를 작성하고 ORG 테이블에서 기존 인덱스에 DEPTNAME을 추가하여 쿼리 3에서 설명한 액세스 플랜에서 빌드합니다. (별도의 인덱스를 추가하면, 추가 액세스가 발생합니다.)

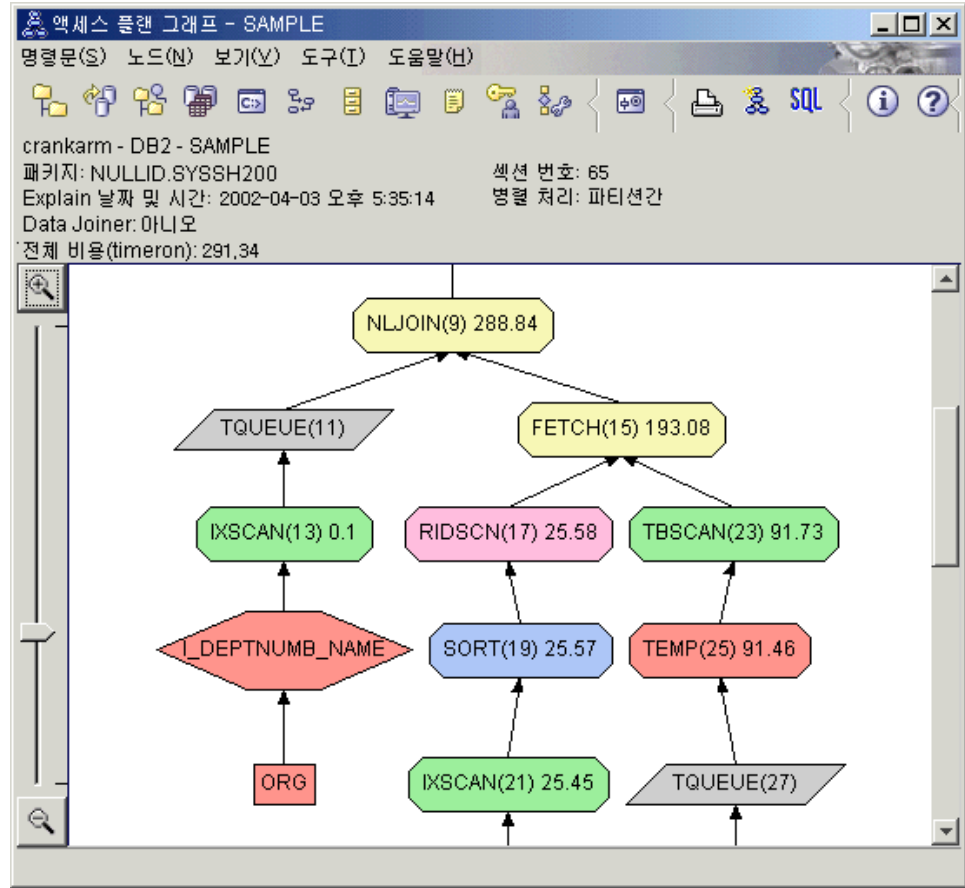
이 쿼리(쿼리 4)에 대한 액세스 플랜 그래프를 보려면 Explain된 명령문 실행기록 창에서 쿼리 번호 4로 식별되는 항목을 더블 클릭하십시오. 명령문을 실행할 액세스 플랜 그래프 창이 열립니다.



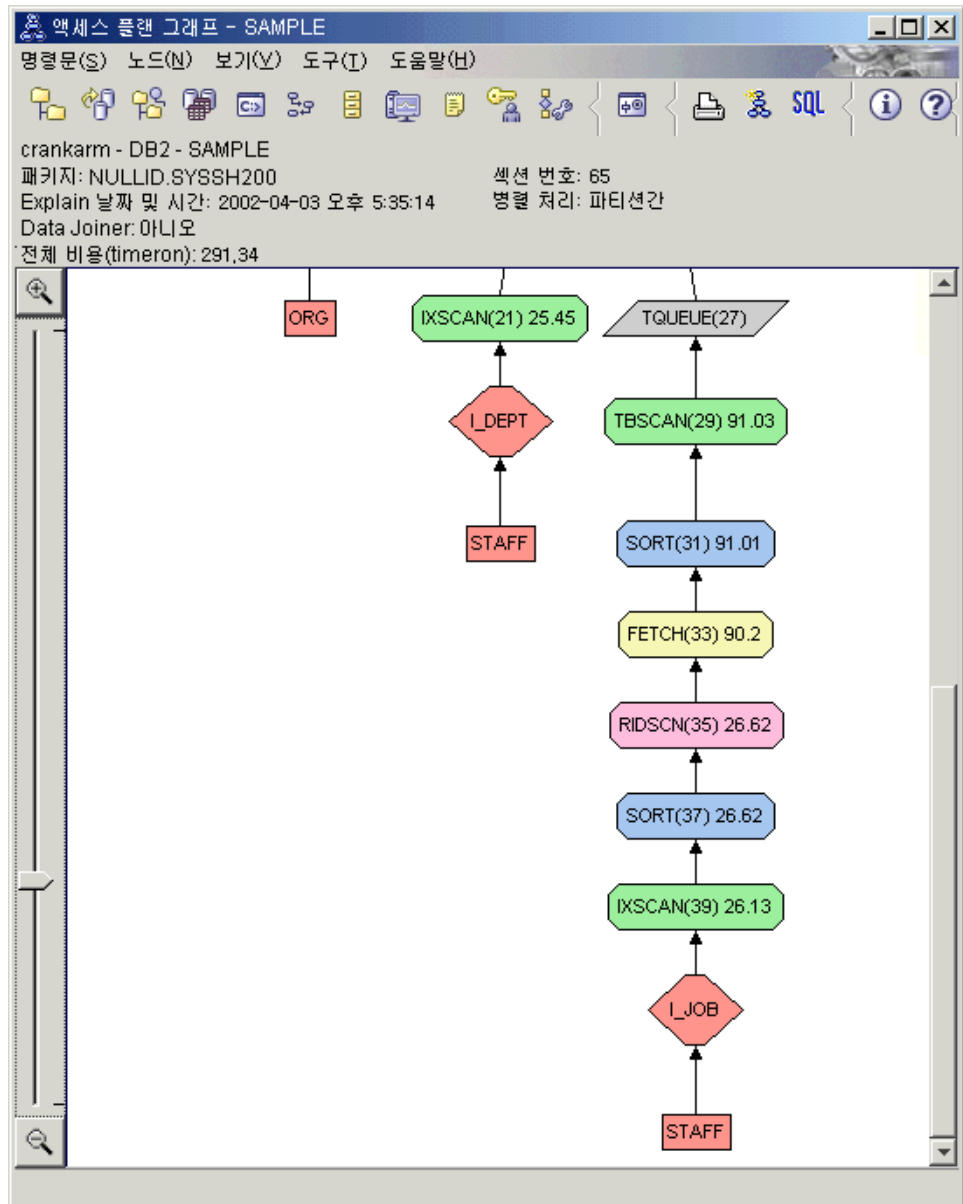
다음 질문에 응답하면 쿼리를 향상시키는 방법을 이해하는 데 도움이 됩니다.

1. 인덱스를 추가로 작성한 결과 이 프로세스 플랜이 어떻게 변경되었습니까?

액세스 플랜 그래프의 중간 부분에서, **ORG** 테이블의 경우 이전 테이블 스캔이 인덱스 스캔 **IXSCAN(9)**로 변경되었다는 점에 유의하십시오. **ORG** 테이블의 인덱스에 **DEPTNAME** 컬럼을 추가하면 옵티마이저가 테이블 스캔에 관련된 액세스를 세분화할 수 있습니다.



액세스 플랜 그래프 맨 아래 부분에서, STAFF 테이블의 경우 이전 인덱스 스캔 및 페치가 인덱스 스캔 IXSCAN(39)만으로 변경되었다는 점에 유의하십시오. STAFF 테이블에 JOB 인덱스를 작성하면 옵티마이저가 페치에 관련된 여분의 액세스를 제거할 수 있습니다.



2. 이 액세스 플랜이 얼마나 효과적입니까?

이 액세스 플랜은 이전 예의 액세스 플랜보다 비용면에서 더 효율적입니다. 누적 비용은 쿼리 3의 대략 753timeron에서 쿼리 4의 대략 288timeron으로 줄었습니다.

다음 레슨 내용

SQL 또는 XQuery문의 성능이 향상됩니다.

성능 향상을 위한 추가 단계의 세부 정보를 찾으려면 DB2 정보 센터를 참조하십시오. Visual Explain으로 돌아가면 조치 효과에 액세스할 수 있습니다.

제 2 부 참조

제 5 장 Visual Explain 연산자

연산자는 SQL 또는 XQuery문에 대한 액세스 플랜을 실행할 때 데이터에 수행해야 할 조치이거나 테이블 또는 인덱스의 출력입니다. 이 절에는 액세스 플랜 그래프에 표시될 수 있는 연산자 목록이 포함되어 있습니다.

CMPEXP 연산자

이 연산자는 디버그 모드에만 사용됩니다.

연산자 이름: CMPEXP

설명: 중간 또는 최종 결과에 필요한 표현식을 계산합니다.

DELETE 연산자

이 연산자는 테이블에서의 행 삭제를 나타냅니다.

연산자 이름: DELETE

설명: 테이블에서 행을 삭제합니다.

이 연산자는 필수 조작입니다. 액세스 플랜 비용을 개선하려면 삭제할 행 세트를 정의하는 다른 연산자(예: scans 및 joins)를 사용하십시오.

성능 제안사항:

- 테이블에서 모든 행을 삭제하려면 DROP TABLE문 또는 LOAD REPLACE 명령 사용을 고려해 보십시오.

EISCAN 연산자

이 연산자는 사용자 정의 인덱스를 스캔하여 행의 스트림을 줄입니다.

연산자 이름: EISCAN

설명: 스캔은 사용자가 제공한 탐색 범위 생성 함수에서 시작/중지 조건을 다중으로 사용합니다.

이 조작은 기본 테이블(술어 기준)에 액세스하기 전에 자격이 있는 행 세트를 줄이기 위해 수행됩니다.

성능 제안사항:

- 시간이 지남에 따라 데이터베이스를 갱신하면 인덱스가 조각화되어 인덱스 페이지가 필요 이상으로 늘어날 수 있습니다. 이 문제는 인덱스를 삭제한 후 다시 작성하거나 인덱스를 재구성하여 해결할 수 있습니다.
- 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.

FETCH 연산자

이 연산자는 테이블에서의 컬럼의 페치를 나타냅니다.

연산자 이름: `FETCH`

설명: 특정 행 ID(RID)를 사용하여 테이블에서 컬럼을 페치합니다.

성능 제안사항:

- 데이터 페이지가 액세스되지 않도록 페치된 컬럼을 포함하는 인덱스 키를 펼치십시오.
- 페치와 관련된 인덱스를 찾은 후 인덱스의 노드를 더블 클릭하여 통계 창을 표시하십시오. 인덱스에 대한 클러스터링 등급이 높은지 확인하십시오.
- 페치한 후의 입/출력(I/O)이 테이블의 페이지 수보다 크면 버퍼 크기를 늘리십시오.
- 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.

Quantile 및 자주 사용되는 값 통계는 술어 선택성에 대한 정보를 제공하여 테이블 스캔을 대비되는 인덱스 스캔의 선택 시기를 판별합니다. 이와 같은 통계를 갱신하려면 `WITH DISTRIBUTION`절이 있는 테이블에서 `runstats` 명령을 사용하십시오.

FILTER 연산자

이 연산자는 데이터가 필터되는 방법을 나타냅니다.

연산자 이름: `FILTER`

설명: 술어에서 제공된 기준에 따라 데이터를 필터하도록 하는 레지듀얼 술어의 응용프로그램입니다.

성능 제안사항:

- 필요한 데이터만 검색하는 술어를 사용했는지 확인하십시오. 예를 들면, 술어의 선택성 값이 리턴하려는 테이블의 일부를 나타내는지 확인하십시오.
- 옵티마이저가 서브쿼리 대신에 조인을 사용할 수 있도록 최적화 클래스가 3 이상인지 확인하십시오. 확인할 수 없으면 SQL 쿼리를 수동으로 다시 작성하여 서브쿼리를 제거하십시오.

GENROW 연산자

이 연산자는 데이터 행을 생성하는 옵티마이저에 의해 사용됩니다.

연산자 이름: GENROW

설명: 테이블, 인덱스 및 연산자에서 입력을 사용하지 않고 행 테이블을 생성하는 내장 함수입니다.

GENROW는 옵티마이저가 데이터 행을 생성할 때 사용할 수 있습니다(예를 들면 INSERT문 또는 조인으로 변환되는 일부 IN 목록의 경우).

GENROW 함수로 생성된 테이블에 대해 제거된 통계를 보려면 노드를 더블 클릭하십시오.

GRPBY 연산자

이 연산자는 행 그룹화를 나타냅니다.

연산자 이름: GRPBY

설명: 지정된 컬럼 또는 함수의 공통 값에 따라 행을 그룹화합니다. 값 그룹을 생성하거나 세트 함수를 평가하는 데 필요한 조작입니다.

GROUP BY 컬럼이 지정되지 않은 경우에도 SELECT 목록에 집계 함수(집계시 전체 테이블을 단일 그룹으로 처리함)가 있으면 GRPBY 연산자를 사용할 수 있습니다.

성능 제안사항:

- 이 연산자는 필수 조작입니다. 액세스 플랜 비용을 개선하려면 그룹화할 행 세트를 정의하는 다른 연산자(예: scans 및 joins)를 사용하십시오.
- 단일 집계 함수는 있지만 GROUP BY절이 없는 SELECT문의 성능을 향상시키려면 다음을 수행하십시오.
 - MIN(C) 집계 함수에는 C에 오름차순 인덱스를 작성하십시오.
 - MAX(C) 집계 함수에는 C에 내림차순 인덱스를 작성하십시오.

HSJOIN 연산자

이 연산자는 테이블의 규정된 행을 해시하는 해시 조인을 나타냅니다.

연산자 이름: HSJOIN

설명: 테이블의 규정된 행을 해시하는 해시 조인으로서 테이블 콘텐츠의 순서를 미리 지정하지 않고 바로 조인할 수 있습니다.

FROM절에서 참조한 테이블이 둘 이상 있을 때 반드시 필요한 조인입니다. 해시 조인은 서로 다른 두 개의 테이블에서 컬럼을 동등하게 하는 조인 술어가 있을 때는 언제든지 사용할 수 있습니다. 조인 술어는 반드시 동일한 데이터 유형이어야 합니다. 또한 해시 조인은 NLJOIN 의 경우와 같이 재작성된 서브쿼리에서도 발생할 수 있습니다.

해시 조인에서는 입력 테이블의 순서를 지정할 필요가 없습니다. 해시 조인의 내부 테이블을 스캔하고 조인 컬럼 값 해시를 통해 찾아보기 테이블을 생성하여 조인을 수행합니다. 그런 다음 조인 컬럼 값을 해시하고 내부 테이블에 대해 생성된 찾아보기 테이블을 체크인하면서 외부 테이블을 읽습니다.

성능 제안사항:

- 조인할 행 수를 줄이려면 로컬 술어(즉, 하나의 테이블을 참조하는 술어)를 사용하십시오.
- 메모리에 해시 찾아보기 테이블을 보관하는 데 충분하도록 정렬 힙(heap)의 크기를 늘리십시오.
- 현재 통계가 아니면 runstats 명령을 사용하여 통계를 갱신하십시오.

INSERT 연산자

이 연산자는 테이블로의 행 삽입을 나타냅니다.

연산자 이름: INSERT

설명: 이 연산자는 필수 조작입니다. 액세스 플랜 비용을 개선하려면 삽입할 행 세트를 정의하는 다른 연산자(예: scans 및 joins)를 사용하십시오.

IXAND 연산자

이 연산자는 여러 개의 인덱스를 스캔한 결과의 ANDing을 나타냅니다.

연산자 이름: IXAND

설명: Dynamic Bitmap 기술을 사용하여 여러 개의 인덱스를 스캔한 결과의 ANDing입니다. 이 연산자는 ANDed 술어를 여러 개의 인덱스에 적용하여 기본적인 테이블 액세스를 최소화할 수 있습니다.

이 연산자가 수행되는 경우는 다음과 같습니다.

- 기본 테이블에 액세스하기 전에 행 세트를 줄이는 경우
- AND와 술어가 여러 개의 인덱스에 적용되는 경우
- AND와 세미조인 결과가 스타 조인에 사용되는 경우

성능 제안사항:

- 시간이 지남에 따라 데이터베이스를 갱신하면 인덱스가 조각화되어 인덱스 페이지가 필요 이상으로 늘어날 수 있습니다. 이 문제는 인덱스를 삭제한 후 다시 작성하거나 인덱스를 재구성하여 해결할 수 있습니다.
- 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.
- 일반적으로 인덱스 스캔은 규정되는 행이 적을 때 가장 효율적입니다. 옵티마이저는 술어에서 참조된 컬럼에 사용할 수 있는 통계를 통해 규정 행의 수를 계산합니다. 다른 값보다 더 자주 발생하는 값이 있으면 `runstats` 명령에 `WITH DISTRIBUTION` 절을 사용하여 분산 통계를 요청해야 합니다. 불규칙한 분산 통계를 사용하여 옵티마이저는 자주 발생하는 값과 자주 발생하지 않는 값을 구분합니다.
- `IXAND`를 사용할 때 시작 및 중지 키가 중요하기 때문에 `IXAND`는 가장 잘 활용되는 단일 컬럼 인덱스라고 할 수 있습니다.
- 스타 조인의 경우에는 사실 테이블 및 관련 차원 테이블의 선택성 컬럼마다 단일 컬럼 인덱스를 작성하십시오.

IXSCAN 연산자

이 연산자는 인덱스의 스캔을 나타냅니다.

연산자 이름: IXSCAN

설명: 인덱스를 스캔하여 행 ID의 스트림을 줄입니다. 스캐닝은 선택적 시작/중지 조건을 사용하거나 인덱스 컬럼을 참조하는 인덱스 기능 술어에 적용될 수 있습니다.

이 조작은 기본 테이블(술어 기준)에 액세스하기 전에 규정 행 ID를 줄일 때 수행됩니다.

성능 제안사항:

- 시간이 지남에 따라 데이터베이스를 갱신하면 인덱스가 조각화되어 인덱스 페이지가 필요 이상으로 늘어날 수 있습니다. 이 문제는 인덱스를 삭제한 후 다시 작성하거나 인덱스를 재구성하여 해결할 수 있습니다.
- 둘 이상의 테이블에 액세스하는 경우에는 외부 테이블의 조인 컬럼에 인덱스를 제공하므로 인덱스를 통해 내부 테이블로 액세스하는 것이 더 효과적입니다.

인덱스에 대한 안내를 더 보려면 Visual Explain의 온라인 도움말을 참조하십시오.

- 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.
- 일반적으로 인덱스 스캔은 규정되는 행 ID가 적을 때 가장 효율적입니다. 옵티마이저는 술어에서 참조된 컬럼에 사용할 수 있는 통계를 통해 규정 행 ID의 수를 계산합니다. 다른 값보다 더 자주 발생하는 값이 있으면 `runstats` 명령에 `WITH DISTRIBUTION` 절을 사용하여 분산 통계를 요청해야 합니다. 불규칙한 분산 통계를 사용하여 옵티마이저는 자주 발생하는 값과 자주 발생하지 않는 값을 구분합니다.

MSJOIN 연산자

이 연산자는 병합 조인을 나타냅니다.

연산자 이름: MSJOIN

설명: 조인 술어 순서에 외부 및 내부 테이블의 규정 행이 있어야 하는 병합 조인을 나타냅니다. 병합 조인은 또한 병합 스캔 조인 또는 정렬된 병합 조인이라고도 합니다.

FROM절에서 참조한 테이블이 둘 이상 있을 때 반드시 필요한 조인입니다. 병합 조인은 서로 다른 두 개의 테이블에서 컬럼을 동등하게 하는 조인 술어가 있을 때는 언제든지 사용할 수 있습니다. 또한 병합 조인은 재작성된 서브쿼리에서도 발생할 수 있습니다.

일반적으로 테이블은 한 번만 스캔되기 때문에 병합 조인은 컬럼을 조인할 때 순서가 지정된 입력을 필요로 합니다. 순서가 지정된 입력은 인덱스 또는 정렬된 테이블에 액세스하여 가져옵니다.

성능 제안사항:

- 조인할 행 수를 줄이려면 로컬 술어(즉, 하나의 테이블을 참조하는 술어)를 사용하십시오.
- 현재 통계가 아니면 runstats 명령을 사용하여 통계를 갱신하십시오.

NLJOIN 연산자

이 연산자는 중첩 루프 조인을 나타냅니다.

연산자 이름: NLJOIN

설명: 외부 테이블의 각 행에 대해 내부 테이블을 한 번만 스캔(대개 인덱스 스캔)하는 중첩 루프 조인입니다.

FROM절에서 참조한 테이블이 둘 이상 있을 때 반드시 필요한 조인입니다. 중첩 루프 조인은 조인 술어를 필요로 하지 않지만 조인 술어가 있으면 더 효율적으로 수행됩니다.

중첩 루프 조인은 다음과 같은 두 가지 방법으로 수행됩니다.

- 외부 테이블에 액세스된 행 별로 내부 테이블을 통해 스캔하는 방법
- 외부 테이블에 액세스된 행 별로 내부 테이블에서 인덱스 찾아보기를 수행하는 방법

성능 제안사항:

- 중첩 루프 조인은 내부 테이블(NLJOIN 연산자 오른쪽에 표시된 테이블)의 조인 술어 컬럼에 인덱스가 있는 경우에 더 효율적입니다. 내부 테이블이 IXSCAN이 아닌 TBSCAN인지 확인하십시오. 확인 후 조인 컬럼에 인덱스를 추가하십시오.

외부 테이블의 조인 컬럼에 인덱스를 작성하여 외부 테이블의 순서를 지정하면 조인을 보다 효율적으로 사용할 수 있으나 중요한 방법은 아닙니다.

- 현재 통계가 아니면 runstats 명령을 사용하여 통계를 갱신하십시오.

PIPE 연산자

이 연산자는 디버그 모드에만 사용됩니다.

연산자 이름: PIPE

설명: 행을 전혀 변경하지 않고 다른 연산자로 행을 전송합니다.

RETURN 연산자

이 연산자는 쿼리로부터의 데이터 리턴을 나타냅니다.

연산자 이름: RETURN

설명: 쿼리에서 사용자에게 데이터를 리턴합니다. 액세스 플랜 그래프의 마지막 연산자로서 액세스 플랜의 누적된 전체 값과 비용을 표시합니다.

이 연산자는 필수 조작입니다.

성능 제안사항:

- 필요한 데이터만 검색하는 술어를 사용했는지 확인하십시오. 예를 들면, 술어의 선택성 값이 리턴하려는 테이블의 일부를 나타내는지 확인하십시오.

RIDSCN 연산자

이 연산자는 행 ID 목록의 스캔을 나타냅니다.

연산자 이름: RIDSCN

설명: 하나 이상의 인덱스에서 가져온 행 ID 목록을 스캔합니다.

옵티마이저가 이 연산자를 수행하는 경우는 다음과 같습니다.

- 술어가 OR 키워드로 연결되거나 IN 술어가 있습니다. 인덱스 ORing이라는 기술을 사용하여 동일한 테이블에서 여러 개의 인덱스 액세스 결과를 결합합니다.
- 기본 행에 액세스하기 전에 행 ID를 정렬하면 I/O의 효율이 높아지므로 단일 인덱스 액세스에는 리스트 프리페치를 사용하는 것이 유리합니다.

RPD 연산자

이 연산자는 리모트 데이터 소스에서 데이터를 검색합니다.

연산자 이름: RPD

설명: 비관계형 랩퍼를 통해 리모트 데이터 소스에서 데이터를 검색할 때 페더레이티드 시스템에서 사용되는 연산자입니다.

옵티마이저가 점검하지 않는 리모트 플랜이 포함될 때 이 연산자가 수행됩니다. RPD 연산자는 리모트 비관계형 데이터 소스에 요청을 보내어 쿼리 결과를 검색합니다. 비관계형 랩퍼는 데이터 소스에서 지원되는 API를 사용하여 요청을 생성합니다.

SHIP 연산자

이 연산자는 리모트 데이터 소스에서 데이터를 검색합니다.

연산자 이름: SHIP

설명: 리모트 데이터 소스에서 데이터를 검색할 때 페더레이티드 시스템에서 사용되는 연산자입니다. 옵티마이저가 점검하지 않는 리모트 플랜이 포함될 때 이 연산자가 수행됩니다. SHIP 연산자는 SQL 또는 XQuery SELECT문을 리모트 데이터 소스에 보내어 쿼리 결과를 검색합니다. SELECT문은 데이터 소스에서 지원하는 SQL 또는 XQuery 형식을 사용하여 생성되므로 데이터 소스에서 허용하는 유효한 쿼리가 포함됩니다.

SORT 연산자

이 연산자는 테이블에서의 행 정렬을 나타냅니다.

연산자 이름: SORT

설명: 테이블에 있는 행을 하나 이상의 컬럼에 정렬하거나 중복된 항목을 제거합니다.

정렬은 요청한 순서를 만족하는 인덱스가 없거나 인덱스 스캔보다 비용이 적게 드는 경우에 필요합니다. 일반적으로 정렬은 필수 행을 폐치한 후 최종 조작으로 수행되거나, 조인 또는 그룹화하기 전에 데이터를 정렬할 때 수행됩니다.

행 수가 많거나 정렬된 데이터를 파이프할 수 없는 경우 조작하려면 임시 테이블을 생성해야 되는데 비용이 많이 듭니다.

성능 제안사항:

- 정렬 컬럼에 인덱스를 추가해 보십시오.
- 필요한 데이터만 검색하는 술어를 사용했는지 확인하십시오. 예를 들면, 술어의 선택성 값이 리턴하려는 테이블의 일부를 나타내는지 확인하십시오.

- 시스템 임시 테이블 스페이스의 프리페치 크기가 충분한지, 즉 I/O 바운드가 아닌지 점검하십시오. (명령문->통계 표시->테이블 스페이스를 선택하면 점검할 수 있습니다.)
- 큰 정렬이 필요한 경우가 자주 발생하면 다음과 같은 구성 매개변수 값을 늘리십시오.
 - 정렬 힙 크기(sortheap). 이 매개변수를 변경하려면 제어 센터에서 데이터베이스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 구성을 선택하십시오. 노트북이 열리면 성능 탭을 선택하십시오.
 - 정렬 힙 임계값(sheaphthres). 이 매개변수를 변경하려면 제어 센터에서 데이터베이스 인스턴스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 구성을 선택하십시오. 노트북이 열리면 성능 탭을 선택하십시오.
- 현재 통계가 아니면 runstats 명령을 사용하여 통계를 갱신하십시오.

TBSCAN 연산자

이 연산자는 테이블 스캔을 나타냅니다.

연산자 이름: TBSCAN

설명: 데이터 페이지로부터 모든 필수 데이터를 직접 읽어서 행을 검색하는 테이블 스캔(관계 스캔)입니다.

인덱스 스캔과 비교하여 옵티마이저가 이러한 스캔 유형을 선택하는 경우는 다음과 같습니다.

- 스캔한 값 범위가 자주 발생합니다(즉, 대부분의 테이블에 액세스해야 합니다).
- 테이블이 작습니다.
- 인덱스 클러스터링이 낮습니다.
- 인덱스가 없습니다.

성능 제안사항:

- 테이블이 커서 대부분의 테이블 행에 액세스하지 못하는 경우에는 테이블 스캔보다 인덱스 스캔이 더 효율적입니다. 이러한 경우 옵티마이저가 인덱스 스캔을 쉽게 사용할 수 있도록 하려면 선택적 술어가 있는 컬럼에 인덱스를 추가하십시오.
- 인덱스가 이미 있으나 사용되지 않으면 각각의 선행 컬럼에 선택적 술어가 있는지 점검하십시오. 선택적 술어가 없으면 인덱스에 대해 클러스터링 등급이 높은지 점검하십시오. (이러한 통계를 보려면 정렬 아래에 있는 테이블의 테이블 통계 창을 연 후 인덱스 누름 단추를 선택하여 인덱스 통계 창을 가져오십시오.)
- 테이블 스페이스의 프리페치 크기가 충분한지, 즉 I/O 바운드가 아닌지 점검하십시오. (명령문->통계 표시->테이블 스페이스를 선택하면 점검할 수 있습니다.)

- 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.

Quantile 및 자주 사용되는 값 통계는 숨어 선택성에 대한 정보를 제공합니다. 예를 들어 이 통계는 테이블 스캔과 대비되는 인덱스 스캔의 선택 시기를 판별하는 데 사용됩니다. 이 값을 갱신하려면 `WITH DISTRIBUTION` 절이 있는 테이블에서 `runstats` 명령을 사용하십시오.

TEMP 연산자

이 연산자는 임시 테이블에 있는 데이터 스토리지를 나타냅니다.

연산자 이름: `TEMP`

설명: 임시 테이블에 데이터를 저장하는 조치로서 다른 연산자가 나중에 읽을 수 있습니다(여러 번 읽기 가능). 테이블은 SQL 또는 XQuery문이 처리된 후에 제거됩니다.

이 연산자는 서브쿼리를 평가하거나 중간 결과를 저장하는 데 필요합니다. 명령문을 갱신하는 경우와 같이 필수로 사용되기도 합니다.

TQ 연산자

이 연산자는 테이블 큐를 나타냅니다.

연산자 이름: `TQ`

설명: 쿼리를 처리하는 데이터베이스 에이전트가 여러 개인 경우 하나의 데이터베이스 에이전트에서 다른 데이터베이스 에이전트로 테이블 데이터를 전달하는 데 사용되는 테이블 큐입니다. 다중 데이터베이스 에이전트는 병렬 처리가 포함된 쿼리를 처리하는 데 사용됩니다. 테이블 큐 유형은 다음과 같습니다.

- 로컬: 단일 노드에서 데이터베이스 에이전트 간 데이터를 전달하는 데 사용하는 테이블 큐입니다. 로컬 테이블 큐는 파티션 내 병렬 처리에 사용됩니다.
- 로컬이 아님: 서로 다른 노드에서 데이터베이스 에이전트 간 데이터를 전달하는 데 사용하는 테이블 큐입니다.

다음은 `TQ` 연산자의 두 가지 유형입니다.

- `ATQ` - 비동기 `TQ` 연산자
- `XTQ` - XML 집계 `TQ` 연산자

`ATQ` 연산자는 하위 계획의 비동기 실행을 사용 가능하게 합니다.

`XTQ` 연산자는 데이터베이스 파티션에 저장된 XML 문서의 XML 시퀀스를 구성하는 테이블 큐입니다.

다음 예에서, US_ORDERS은 여러 데이터 파티션에 걸쳐 있는 파티션된 테이블이며, US_ORDERS.DETAILS은 XML 컬럼입니다. 다음 XQuery문은 판매한 총 제품 수가 100을 초과할 때 모든 주문을 리턴합니다.

```
Xquery let $all_orders := db2-fn:xmlcolumn('US_ORDERS.DETAILS')
```

```
Xquery let $all_orders := db2-fn:xmlcolumn('US_ORDERS.DETAILS')
```

여기서

```
sum($all_orders//product/qty) > 100
```

은 다음을 리턴합니다.

```
$all_orders
```

XQuery문의 경우에 db2exfmt 명령을 사용하면 XTQ 연산자가 있는 다음 액세스 플랜 출력이 표시됩니다. 플랜 출력에서, US_ORDERS.DETAILS 컬럼에 저장된 모든 XML 문서에 대한 참조는 코디네이터 파티션으로 경로지정되고 전역 시퀀스로 집계된 후 나중에 전역 시퀀스의 각 항목은 탐색을 위해 다시 원래의 파티션으로 경로지정되고 결과를 새 출력 전역 시퀀스로 함께 집계합니다.

```

      Rows
      RETURN
      ( 1)
      Cost
      I/O
      |
      1
      NLJOIN
      ( 2)
      98.171
      8
      /-+--W
      1          2000
      FILTER      XTQ
      ( 3)      ( 7)
      37.4289      60.7421
      1          7
      |          |
      1          0.5
      GRPBY      XSCAN
      ( 4)      ( 8)
      37.3755      57.2
      1          7
      |          |
      360
      DTQ
      ( 5)
      33.7655
      1
      |
      180
      TBSCAN      ( 6)
      22.051
      1
  
```

UNION 연산자

이 연산자는 여러 테이블에서의 행 스트림 병합을 나타냅니다.

연산자 이름: UNION

설명: 이 연산자는 필수 조작입니다. 액세스 플랜 비용을 개선하려면 병합할 행 세트를 정의하는 다른 연산자(예: scans 및 joins)를 사용하십시오.

UNIQUE 연산자

이 연산자는 중복 값이 포함된 행을 나타냅니다.

연산자 이름: UNIQUE

설명: 지정된 컬럼에 대해 값이 중복된 행을 제거합니다.

성능 제안사항:

- 해당 컬럼에 고유 인덱스가 있으면 필요하지 않은 연산자입니다.

인덱스에 대한 안내는 Visual Explain의 온라인 도움말에서 [Creating appropriate indexes](#) 항목을 참조하십시오.

UPDATE 연산자

이 연산자는 테이블 행에서의 데이터 갱신을 나타냅니다.

연산자 이름: UPDATE

설명: 이 연산자는 필수 조작입니다. 액세스 플랜 비용을 개선하려면 갱신할 행 세트를 정의하는 다른 연산자(예: scans 및 joins)를 사용하십시오.

XANDOR 연산자

이 연산자는 ANDed 술어를 여러 개의 인덱스에 적용하여 기본적인 테이블 액세스를 최소화할 수 있습니다.

연산자 이름: XANDOR

설명: 여러 개의 인덱스를 스캔한 결과의 XML 데이터에 있는 인덱스 ANDing은 하나의 쿼리에서 복잡한 술어를 평가하는 데 사용됩니다.

XANDOR 연산자를 사용하려면 다음 조건을 만족해야 합니다.

- 등호 술어만 사용됩니다.
- 인덱스 찾아보기 경로에 와일드카드가 없습니다.
- 동일한 XML 컬럼에 모든 술어가 사용됩니다.

이 조건을 모두 만족하지 않으면 IXAND 연산자가 대신 사용됩니다.

db2exfmt 도구로 표시된 다중 XANDORed XML 데이터에 있는 인덱스 스캔의 액세스 플랜은 다음과 같습니다.

```

          Rows
          RETURN
          ( 1)
          Cost
          I/O
          |
          0.00915933
          NLJOIN
          ( 2)
          985.789
          98.9779
          /---+---#
          2.96215 0.00309213
          FETCH      XSCAN
          ( 3)      ( 11)
          340.113  217.976
          19      27
          /---+---#
          2.96215      210000
          RIDSCN  TABLE: DB2XML
          ( 4)      TPCHX
          332.008
          18
          |
          2.96215
          SORT
          ( 5)
          331.957
          18
          |
          2.96215
          XANDOR
          ( 6)
          331.784
          18
          +-----+-----+-----+-----+
          355.62      6996.81      105000      105000
          XISCAN      XISCAN      XISCAN      XISCAN
          ( 7)      ( 8)      ( 9)      ( 10)
          165.892      3017.54      1.6473e+06      851554
          9      81      27768      14898
          |      |      |      |
          210000      210000      210000      210000
          XMLIN: DB2XML  XMLIN: DB2XML  XMLIN: DB2XML  XMLIN: DB2XML
          TPCHX_IDX      TPCHX_IDX      TPCHX_IDX      TPCHX_IDX
    
```

각각의 XISCAN 연산자는 인덱스 스캔을 수행하고 규정된 XML 노드에 XANDOR 연산자를 제공합니다. XANDOR 연산자는 AND 및 OR 술어를 적용하며, 쿼리에 대한 XML 패턴을 만족하는 XML 노드를 리턴합니다.

성능 제안사항:

- 시간이 지남에 따라 데이터베이스를 갱신하면 인덱스가 조각화되어 인덱스 페이지가 필요 이상으로 늘어날 수 있습니다. 이 문제는 인덱스를 삭제한 후 다시 작성하거나 인덱스를 재구성하여 해결할 수 있습니다.
- 현재 통계가 아니면 RUNSTATS 명령을 사용하여 통계를 갱신하십시오.
- 일반적으로 인덱스 스캔은 규정되는 행이 적을 때 가장 효율적입니다. 옵티마이저는 술어에서 참조된 컬럼에 사용할 수 있는 통계를 통해 규정 행의 수를 계산합니다. 다른 값보다 더 자주 발생하는 값이 있으면 runstats 명령에 WITH DISTRIBUTION 절을 사용하여 분산 통계를 요청해야 합니다. 불규칙한 분산 통계를 사용하여 옵티마이저는 자주 발생하는 값과 자주 발생하지 않는 값을 구분합니다.

XISCAN 연산자

단일 쿼리 술어에 수행되는 조작입니다.

연산자 이름: XISCAN

설명: 이 조작을 평가하면 기본 테이블에 액세스하기 전에 연관된 XML 데이터에 있는 인덱스를 범위 스캔하여 행 ID 및 XML 노드 ID의 규정 리턴 세트를 줄입니다. 컴파일러가 쿼리 완료에 대한 인덱스 정보 사용 여부와 사용법을 판별하므로 인덱스를 사용하면 쿼리의 성능이 향상됩니다. 일반적인 결과는 다음과 같습니다.

- 문서별로 노드가 정렬되고 중복된 노드는 제거됩니다.
- 규정 문서가 포함된 테이블의 행을 각각 폐치합니다.
- XML 문서에서 XSCAN이 조작됩니다.

예를 들어 샘플 XML 문서 조각에 나열된 모든 사람들 중에 성이 "Murphy"인 사람의 이름을 찾는 경우, 올바른 결과를 폐치할 수 있는 XQuery문은 다음과 같습니다.

```
db2-fn:column("EMPLOYEE.XMLCOL")/emp//name[last="Murphy"]/first
```

다음과 같은 명령문을 사용하여 모든 성에 대한 XML 데이터의 인덱스를 이미 작성했다고 가정합니다.

```
CREATE INDEX empname on EMPLOYEE(XMLCOL)
GENERATE KEY USING XMLPATTERN '//name/last'
AS SQL VARCHAR(50)
```

쿼리 컴파일러는 XISCAN 연산자를 선택하여 쿼리를 평가할 수 있습니다. 따라서 패턴은 /emp//name/last, 연산자는 "=", 값은 "Murphy"가 수신됩니다. 인덱스를 사용하면 "Murphy"라는 성과 연관된 노드를 신속하게 찾을 수 있습니다. db2exfmt로 출력되는 플랜 조각의 결과는 다음과 같습니다.

```

      Rows
      RETURN
      ( 1)
      Cost
      I/O
      |
      6454.4
      NLJOIN
      ( 2)
      1.53351e+06
      189180
      /--+--#
      6996.81  0.922477
      FETCH    XSCAN
      ( 3)     ( 7)
      4091.76  218.587
      266      27
      /---+---#
      6996.81      210000
      RIDSCN  TABLE: DB2XML
      ( 4)     TX
      3609.39
      81
      |
      6996.81
      SORT
      ( 5)
      3609.34
      81
      |
      6996.81
      XISCAN
      ( 6)
      3017.54
      81
      |
      210000
      XMLIN: DB2XML
      TX_IDX
  
```

성능 제안사항:

- 시간이 지남에 따라 데이터베이스를 갱신하면 인덱스가 조각화되어 인덱스 페이지가 필요 이상으로 늘어날 수 있습니다. 이 문제는 인덱스를 삭제한 후 다시 작성하거나 인덱스를 재구성하여 해결할 수 있습니다.
- 둘 이상의 테이블에 액세스하는 경우에는 외부 테이블의 조인 컬럼에 인덱스를 제공하므로 인덱스를 통해 내부 테이블로 액세스하는 것이 더 효과적입니다.
- 현재 통계가 아니면 RUNSTATS 명령을 사용하여 통계를 갱신하십시오.

XSCAN 연산자

이 연산자는 XPath 표현식을 평가하고 필요에 따라 문서 조각을 추출하기 위해 XML 조각을 탐색하는 데 사용됩니다.

연산자 이름: XSCAN

설명: 이 연산자는 중첩 루프 조인 연산자(NLJOIN)가 전달한 노드 참조를 처리합니다. 액세스 플랜에 직접 입력해서 표현하지 않습니다.

예를 들어, 다음 XQuery문을 참조하십시오.

```
XQUERY for $i in db2-fn:xmlcolumn("MOVIES.XMLCOL")//actor return $i
```

db2exfmt 유틸리티에서 제공된 이 명령문의 액세스 플랜에는 테이블 TELIAZ.MOVIES에서 테이블 스캔으로 검색된 문서 노드 참조를 처리하는 XSCAN 연산자가 표시됩니다. 이러한 경우의 XSCAN 연산자는 TELIAZ.MOVIES.XMLCOL 수집에 있는 조차자 노드 참조를 리턴합니다.

```
      Rows
      RETURN
      ( 1)
      Cost
      I/O
      |
      180
      NLJOIN
      ( 2)
      10137.9
      1261
      /-+W
      180      1
      TBSCAN  XSCAN
      ( 3)    ( 4)
      21.931  56.2
      1      7
      |
      180
      TABLE: TELIAZ
      MOVIES
```

제 6 장 Visual Explain 개념

이 절에는 Visual Explain 개념 정보가 포함되어 있습니다.

액세스 플랜

Explain 가능한 명령문을 해석하려면 특정 데이터가 필요합니다. 액세스 플랜은 이러한 데이터를 액세스할 조작 순서를 지정합니다.

액세스 플랜을 사용하면 선택된 테이블, 인덱스 또는 컬럼(연산자 등록 정보, 테이블 스페이스 및 함수 통계와 같은 전역 정보 및 최적화와 관련된 구성 매개변수)에 대한 통계를 볼 수 있습니다. Visual Explain에서는 SQL 또는 XQuery문의 액세스 플랜을 그래프로 볼 수 있습니다.

옵티마이저는 Explain 가능한 SQL 또는 XQuery문을 컴파일할 때마다 액세스 플랜을 작성합니다. 이는 정적 명령문의 경우 prep/바인드 시, 동적 명령문의 경우 런타임 시 발생합니다.

액세스 플랜은 사용 가능한 정보에 따른 예측임을 이해해야 합니다. 옵티마이저의 예측은 다음과 같은 정보를 기준으로 합니다.

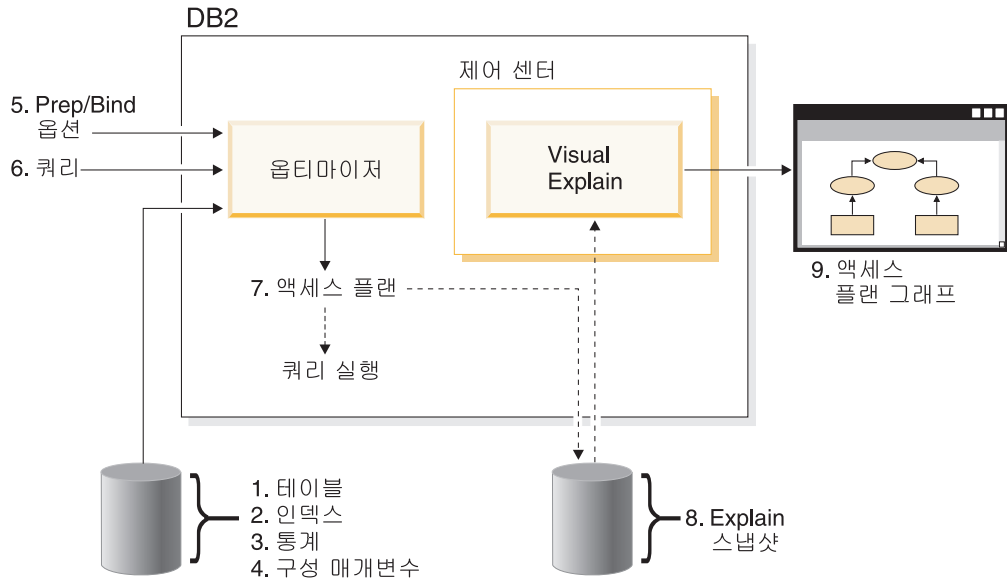
- 시스템 카탈로그 테이블의 통계(현재 통계가 아니면 RUNSTATS 명령을 사용하여 통계를 갱신해야 함)
- 구성 매개변수
- 바인드 옵션
- 쿼리 최적화 클래스

액세스 플랜과 연관된 비용 정보는 쿼리 자원 사용에 대한 옵티마이저의 최상의 예측입니다. 쿼리의 실제 경과 시간은 데이터베이스 관리 프로그램 범위 밖의 요소(예: 동시에 실행 중인 기타 응용프로그램 수)에 따라 달라질 수 있습니다. 실제 경과 시간은 쿼리가 실행되는 동안 성능 모니터를 사용하여 측정할 수 있습니다.

액세스 플랜 그래프

Visual Explain은 수많은 소스 정보를 사용하여 액세스 플랜 그래프를 작성합니다.

아래 그림과 같이 옵티마이저가 여러 가지 입력 사항에 따라 액세스 플랜을 선택하면, Visual Explain이 액세스 플랜 그래프에 선택된 액세스 플랜을 표시합니다. 그래프의 노드에는 테이블 및 인덱스와 각각의 조작이 표시됩니다. 노드 사이의 링크는 데이터 플로우를 보여줍니다.



액세스 플랜 그래프 노드

액세스 플랜 그래프는 노드를 표시하는 트리로 구성됩니다.

노드는 다음과 같이 표시됩니다.

- 테이블: 직사각형
- 인덱스: 다이아몬드
- 연산자(팔각형) TQ 연산자(평행사변형)
- 테이블 함수(육각형)

클러스터링

시간을 초과하여 갱신하면 데이터 페이지의 행은 위치를 변경하여 인덱스와 데이터 사이에 있는 클러스터링 등급을 낮출 수 있습니다.

선택한 인덱스와 관련된 테이블을 재구성하면 데이터가 다시 클러스터됩니다. 클러스터된 인덱스는 기본 테이블에서 데이터에 더 적절하게 순차적으로 액세스할 수 있으므로 범위 술어가 있는 컬럼에 가장 유용합니다. 따라서 동일한 데이터 페이지에 같은 값이 있게 되므로 폐치하는 페이지 수가 줄어듭니다.

일반적으로 높은 등급의 클러스터링은 테이블에서 하나의 인덱스에만 허용됩니다.

노드를 더블 클릭하여 인덱스 통계 창이 표시되면 인덱스의 클러스터링 등급을 점검할 수 있습니다. 이 창에는 클러스터 비율 또는 클러스터 인수 값이 표시됩니다. 값이 낮으면 테이블 데이터를 재구성하십시오.

컨테이너

컨테이너는 데이터의 실제 스토리지 위치입니다.

테이블 스페이스와 관련되므로 파일 또는 디렉토리 또는 디바이스가 될 수 있습니다. 컨테이너의 번호는 0부터 순차적으로 시작됩니다.

비용

액세스 플랜에서의 비용은 명령문(또는 명령문 요소)의 액세스 플랜을 실행하는 데 필요로 하는 계산된 총 자원 사용량입니다.

비용은 CPU 비용(지시사항 수)과 I/O(찾기 및 페이지 전송 수)를 조합하여 산출됩니다.

비용 단위는 *timeron*입니다. *timeron*은 실제 경과 시간과 직접적으로 같지는 않지만 데이터베이스 관리 프로그램에서 동일한 쿼리에 두 개의 플랜을 실행하는 데 필요한 자원의 대략적 예측입니다.

액세스 플랜 그래프의 연산자 노드별로 표시된 비용은 액세스 플랜 실행의 시작부터 특정 연산자 실행까지의 누적 비용입니다. 시스템 워크로드 또는 데이터 행 리턴 비용과 같은 인수는 반영되지 않습니다.

커서 블로킹

커서 블로킹이란 데이터베이스 관리 프로그램이 단일 조작에서 행 블록을 검색하여 오버헤드를 줄이는 기술입니다.

이러한 행은 처리되는 동안 DB2 클라이언트의 캐시에 저장됩니다. 응용프로그램이 OPEN CURSOR 요청을 발행하면 캐시가 할당되고, 커서를 닫으면 할당 해제됩니다. 모든 행이 처리되면 행의 다른 블록이 검색됩니다.

커서 블로킹의 유형을 지정하려면 PREP 또는 BIND 명령의 BLOCKING 옵션을 다음 매개변수와 함께 사용하십시오.

UNAMBIG

FOR READ ONLY절로 지정된 커서의 커서 블로킹

앰비규어스가 아니면서 읽기 전용인 FOR READ ONLY 또는 FOR UPDATE 절로 선언되지 않은 커서가 블록화됩니다. 앰비규어스 커서는 블록화되지 않습니다.

ALL FOR READ ONLY절로 지정되거나 FOR UPDATE로 지정되지 않은 커서의 커서 블로킹

NO 커서에 블로킹이 발생하지 않습니다.

읽기 전용 커서 및 앰비규어스 커서의 정의는 DECLARE CURSOR문을 참조하십시오.

데이터베이스 관리 테이블 스페이스

데이터베이스에 존재할 수 있는 테이블 스페이스 유형은 데이터베이스 관리 스페이스(DMS)와 시스템 관리 스페이스(SMS)입니다.

DMS 테이블 스페이스는 데이터베이스 관리 프로그램이 관리합니다. 또한 데이터베이스 관리 프로그램의 요구사항에 맞게 디자인되고 조정됩니다.

DMS 테이블 스페이스 정의에는 데이터베이스 데이터가 DMS 테이블 스페이스 형식으로 저장된 파일(또는 디바이스) 목록이 포함됩니다.

사전 할당된 파일(또는 디바이스)을 기존 DMS 테이블 스페이스에 추가하면 스토리지 용량을 늘릴 수 있습니다. 데이터베이스 관리 프로그램은 해당 테이블 스페이스에 속하는 모든 컨테이너에서 기존 데이터를 자동으로 다시 조절합니다.

DMS와 SMS 테이블 스페이스는 동일한 데이터베이스에 함께 존재할 수 있습니다.

동적 SQL 또는 XQuery

동적 SQL 또는 XQuery문은 프로그램이 실행되는 동안 응용프로그램에서 준비 및 실행되는 명령문입니다.

동적 SQL 또는 XQuery에서는 다음과 같은 경우가 모두 가능합니다.

- CLI 또는 CLP를 사용하여 SQL 또는 XQuery문을 대화식으로 발행합니다.
- SQL 또는 XQuery 소스는 응용프로그램에 임베디드(embedded)된 호스트 언어 변수에 포함됩니다.

데이터베이스 관리 프로그램이 동적 SQL 또는 XQuery문을 실행하면, 현재 카탈로그 통계 및 구성 매개변수를 기준으로 하는 액세스 플랜이 작성됩니다. 이 액세스 플랜은 명령문 응용프로그램 실행에서 다음으로 변경될 수 있습니다.

동적 SQL 또는 XQuery의 대안으로 정적 SQL 또는 XQuery가 있습니다.

Explain 스냅샷

Visual Explain을 통해 Explain 스냅샷의 내용을 살펴볼 수 있습니다. Explain 스냅샷은 SQL문이 Explain될 때 수집되는 압축 정보입니다.

이 정보는 EXPLAIN_STATEMENT 테이블에 BLOB(Binary Large Object)로 저장되며 다음과 같은 정보를 포함합니다.

- 해당 연산자와 액세스된 테이블 및 인덱스를 포함하는 액세스 플랜의 내부 표시
- 데이터베이스 오브젝트에 대한 통계와 각 조작에 대한 누적 비용을 포함하는, 옵티마이저가 사용하는 결정 기준

Explain 스냅샷은 SQL문의 액세스 플랜을 그래픽으로 표시하려는 경우에 필요합니다. Explain 스냅샷이 작성되었는지 확인하려면 다음을 수행하십시오.

1. Explain 테이블이 데이터베이스 관리 프로그램에 있어야 Explain 스냅샷을 저장할 수 있습니다. Explain 테이블을 작성하는 방법은 온라인 도움말에서 Explain 테이블 작성을 참조하십시오.
2. 정적 SQL 또는 XQuery문이 포함된 패키지의 경우 패키지를 바인드 또는 prep할 때 EXPLSNAP 옵션을 ALL 또는 YES로 설정하십시오. Explain 가능한 SQL문에 따라 Explain 스냅샷을 패키지에 가져옵니다. 자세한 정보는 BIND 및 PREP 명령을 참조하십시오.
3. 동적 SQL문의 경우, EXPLSNAP 옵션을 ALL로 설정하여 해당 SQL문을 발행한 응용프로그램을 바인드하거나 CURRENT EXPLAIN SNAPSHOT 특수 레지스터를 YES 또는 EXPLAIN을 설정한 후 대화식으로 SQL문을 발행하십시오. 자세한 정보는 CURRENT EXPLAIN SNAPSHOT 특수 레지스터 및 SET CURRENT EXPLAIN SNAPSHOT문을 참조하십시오.

Explain 가능한 명령문

Explain 가능한 명령문은 Explain 조작을 수행할 수 있는 SQL 또는 XQuery문입니다.

Explain 가능한 SQL 또는 XQuery문은 다음과 같습니다.

- DELETE
- INSERT
- MERGE
- REFRESH TABLE
- SELECT
- SET INTEGRITY
- UPDATE
- VALUES

Explain된 명령문

*Explain*된 명령문은 Explain 조작이 수행된 SQL 또는 XQuery문입니다.

Explain된 명령문은 Explain된 명령문 실행기록 창에 표시됩니다.

피연산자

피연산자는 조작이 수행되는 엔티티입니다.

예를 들어, 테이블 또는 인덱스는 여러 가지 연산자(예: TBSCAN 및 IXSCAN0)의 피연산자입니다.

연산자

연산자는 SQL 또는 XQuery문에 대한 액세스 플랜을 실행할 때 데이터에 수행해야 할 조치이거나 테이블 또는 인덱스의 출력입니다.

액세스 플랜 그래프에 표시되는 연산자는 다음과 같습니다.

DELETE

테이블에서 행을 삭제합니다.

EISCAN

사용자 정의 인덱스를 스캔하여 행의 스트림을 줄입니다.

FETCH

특정 레코드 ID를 사용하여 테이블에서 컬럼을 페치합니다.

FILTER

하나 이상의 술어를 적용하여 데이터를 필터합니다.

GENROW

행 테이블을 생성합니다.

GRPBY

지정된 컬럼 또는 함수의 공통 값으로 행을 그룹화하고, 설정 함수를 평가합니다.

HSJOIN

둘 이상의 테이블이 조인 컬럼에 해시된 해시 조인을 나타냅니다.

INSERT

테이블에 행을 삽입합니다.

IXAND

두 개 이상 인덱스 스캔에서 행 ID(RID)를 묶는 AND입니다.

IXSCAN

선택적 시작/중지 조건으로 테이블 인덱스를 스캔하여 순서가 지정된 행 스트림을 생성합니다.

MSJOIN

조인 술어 순서에 외부 및 내부 테이블이 있어야 하는 병합 조인을 나타냅니다.

NLJOIN

외부 테이블의 각 행에 대해 내부 테이블에 한 번만 액세스하는 중첩 루프 조인을 나타냅니다.

RETURN

쿼리에서 사용자로의 데이터 리턴을 나타냅니다.

RIDSCAN

하나 이상의 인덱스에서 가져온 행 ID 목록을 스캔합니다.

RPD(Remote PushDown)

리모트 플랜에 대한 연산자입니다. 버전 8의 SHIP 연산자(이전 버전에서는 RQUERY 연산자)와 매우 유사하지만 SQL 또는 XQuery문이 포함되지 않습니다.

SHIP 리모트 데이터베이스 소스에서 데이터를 검색합니다. 페더레이티드 시스템에서 사용됩니다.

SORT 지정된 컬럼의 순서대로 행을 정렬합니다. 중복된 항목을 제거하기도 합니다.

TBSCAN

데이터 페이지로부터 모든 필수 데이터를 직접 읽어서 행을 검색합니다.

TEMP

다시(여러 번) 읽을 수 있도록 임시 테이블에 데이터를 저장합니다.

TQ 데이터베이스 에이전트 사이에서 테이블 데이터를 전송합니다.

UNION

여러 개의 테이블에서 행 스트림을 병합합니다.

UNIQUE

지정된 컬럼에 대해 값이 중복된 행을 제거합니다.

UPDATE

테이블에서 행을 갱신합니다.

XISCAN

XML 테이블 인덱스를 스캔합니다.

XSCAN

XML 문서 노드 서브트리를 탐색합니다.

XANDOR

ANDed 및 ORed 술어를 여러 개의 XML 인덱스에 적용할 수 있습니다.

옵티마이저

옵티마이저는 DML(Data Manipulation Language) SQL문에 대한 액세스 플랜을 선택하는 SQL 컴파일러 구성요소입니다.

수많은 대안 액세스 플랜의 실행 비용을 모델링하고 계산된 비용이 가장 적은 것을 선택합니다.

패키지

패키지는 응용프로그램의 소스 파일과 연관된 SQL문을 처리하는 데 필요한 정보가 있는 데이터베이스에 저장된 오브젝트입니다.

패키지는 다음과 같은 두 가지 방법으로 생성됩니다.

- PREP 명령으로 소스 파일 프리컴파일링
- BIND 명령으로 프리컴파일러가 생성한 바인드 파일 바인딩

술어

술어는 비교 조작을 표현하거나 암시하는 검색 조건의 요소입니다. 술어는 WHERE 또는 HAVING으로 시작하는 절에 포함됩니다.

SQL문 예제:

```
SELECT * FROM SAMPLE
WHERE NAME = 'SMITH' AND
DEPT = 895 AND YEARS > 5
```

술어: NAME = 'SMITH', DEPT = 895 및 YEARS > 5

술어는 다음 범주 중 하나에 해당됩니다(유용한 순서대로 정렬).

1. 시작 및 중지 조건은 인덱스 스캔을 대괄호(축소)로 묶습니다. (이러한 조건을 범위 지정 술어라고도 합니다.)
2. 술어와 관련된 컬럼이 인덱스 키의 일부이므로 인덱스 페이지(일명 인덱스 Sargable) 술어를 인덱스에서 평가할 수 있습니다.
3. 데이터 페이지(일명 데이터 Sargable) 술어는 인덱스에서 평가할 수 없으나 버퍼에 행에 남아 있는 경우에는 평가할 수 있습니다.
4. 레지듀얼(Residual) 술어는 일반적으로 기본 테이블의 단순한 액세스 외에 I/O를 필요로 하므로 버퍼 페이지에서 데이터를 복사한 후 적용해야 합니다. 여기에 포함된 술어는 서브쿼리를 포함하거나 파일(테이블과 분리)에 저장된 LONG VARCHAR 또는 LOB 데이터를 읽습니다.

술어를 디자인하려면 가능한 선택성을 높여야 하므로 최소한의 행이 리턴됩니다.

제일 효율적이면서 가장 일반적으로 사용되는 술어의 유형은 다음과 같습니다.

- 단순한 *Equality Join* 술어는 병합 조인에 필요합니다. `table1.column = table2.column`과 같은 형식으로 두 개의 서로 다른 테이블의 컬럼을 동등하게 하므로 테이블을 조인할 수 있습니다.

- 로컬 술어는 하나의 테이블에만 적용됩니다.

쿼리 최적화 클래스

쿼리 최적화 클래스는 쿼리를 컴파일하는 쿼리 제작성 규칙 및 최적화 기술 세트입니다.

기본 쿼리 최적화 클래스는 다음과 같습니다.

- 1 제한된 최적화. 메모리 및 처리 자원이 엄격하게 제한될 때 유용합니다. 대략적으로 버전 1에서 제공하는 최적화와 같습니다.
- 2 경미한 최적화. 버전 1보다는 높지만 매우 복잡한 쿼리의 경우 최적화 비용이 3 이상의 레벨보다 적은 최적화 레벨을 지정합니다.
- 3 중간 최적화. z/OS[®]용 DB2의 쿼리 최적화 특성과 가장 근접하게 일치합니다.
- 5 일반 최적화. 단순한 트랜잭션과 복잡한 쿼리를 둘 다 사용하는 혼합 환경에서 사용하도록 권장합니다.
- 7 일반 최적화. 쿼리 최적화 5와 같지만 복잡한 동적 쿼리에 대한 쿼리 최적화량이 줄어들지 않습니다.

특수 환경에서만 사용되는 기타 쿼리 최적화 클래스는 다음과 같습니다.

- 0 최소 최적화. 필요한 최적화가 적거나 없는 경우에만 사용합니다(즉, 인덱스화된 테이블의 쿼리가 매우 단순한 경우).
- 9 최대 최적화. 대량의 메모리 및 처리 자원을 사용합니다. 클래스 5가 충분하지 않은 경우에만 사용합니다(즉, 쿼리가 너무 복잡하고 실행 시간이 길어서 클래스 5에서는 제대로 수행되지 않는 경우).

일반적으로 정적 쿼리 및 실행 시간이 긴 쿼리에는 상위 최적화 클래스를 사용하고, 동적으로 제출되거나 실행 시간이 짧은 단순한 쿼리에는 하위 최적화 클래스를 사용하십시오.

동적 SQL 또는 XQuery문에 쿼리 최적화를 설정하려면 명령행 처리기에 다음 명령을 입력하십시오.

```
SET CURRENT QUERY OPTIMIZATION = n;
```

여기서 'n'은 사용하려는 쿼리 최적화 클래스입니다.

정적 SQL 또는 XQuery문에 쿼리 최적화를 설정하려면 BIND 또는 PREP 명령에 QUERYOPT 옵션을 사용하십시오.

Explain XML 연산자에 대한 샘플 XML 문서 조각

이 샘플 XML 문서 조각은 Explain XML 연산자 XSCAN, XISCAN 및 XANDOR 을 검토하는 데 사용됩니다.

```
<emp id='12345' salary='60000'>
  <name>
    <first>William</first>
    <last>Murphy</last>
  </name>
  <spouse>
    <name>
      <first>Cecilia</first>
      <last>Murphy</last>
    </name>
  </spouse>
  <dept id='K55'>
    Finance
  </dept>
</emp>
<emp id='12345' salary='40000'>
  <name>
    <first>Patricio</first>
    <last>Murphy</last>
  </name>
  <dept id='A15'>
    Sales
  </dept>
</emp>
<emp id='12346' salary='70000'>
  <name>
    <first>Victoria</first>
    <last>Zubiri</last>
  </name>
  <dept id='B11'>
    Marketing
  </dept>
</emp>
```

술어의 선택성

선택성은 행이 술어를 만족시킬(즉, 참이 되는) 가능성입니다.

예를 들어 행이 1,000,000개인 테이블에서 조작되는 술어의 선택성이 0.01(1%)이면 술어는 계산된 10,000개(1,000,000개의 1%)의 행을 리턴하고 990,000개의 행은 버립니다.

선택성이 높은 술어(선택성 0.10 이하)가 좋습니다. 이러한 술어는 나중에 연산자가 조작할 행을 더 조금 리턴하므로 쿼리를 만족시킬 CPU 및 I/O가 더 조금 필요합니다.

예: 테이블에 1,000,000개의 행이 있고, 원래의 쿼리에 'ORDER BY'절이 있어서 정렬 단계가 추가로 필요하다고 가정합니다. 술어의 선택성이 0.01이므로 계산된 10,000

개의 행이 정렬되어야 합니다. 그러나 선택적 술어가 0.50으로 줄어들면 계산된 500,000 개의 행이 정렬되어야 하지만 CPU 및 I/O 시간이 추가로 필요합니다.

스타 조인

사실 테이블(대형 중앙 테이블)이 둘 이상의 차원 테이블(사실 테이블의 컬럼 값 설명을 포함하는 소형 테이블)에 조인되는 경우 조인 세트를 스타 조인이라고 합니다.

스타 조인을 구성하는 3개의 주요 파트는 다음과 같습니다.

- 세미조인
- 세미조인 결과의 인덱스 ANDing
- 세미조인 완료

54 페이지의 『IXAND 연산자』를 제공하는 둘 이상의 조인으로 표시됩니다.

세미조인은 내부 및 외부 테이블 컬럼을 조인하는 대신에 내부 테이블의 행 ID(RID)만 조인 결과가 될 수 있는 특별한 형식의 조인입니다.

스타 조인은 세미조인을 사용하여 행 ID(RID)를 인덱스 ANDing 연산자에게 제공합니다. 인덱스 ANDing 연산자는 여러 가지 조인의 필터링 작용을 모아둡니다. 인덱스 ANDing 연산자의 출력이 인덱스 ORing 연산자로 제공됩니다. 이때 행 ID(RID) 순서가 지정되고 인덱스 ANDing 연산자를 제공하는 조인에서 발생할 수 있는 중복 행이 제거됩니다. 그런 다음 사실 테이블의 행이 Fetch 연산자를 사용하여 폐치됩니다. 마지막으로 줄어든 사실 테이블이 모든 차원 테이블에 조인되면 조인을 완료합니다.

성능 제안사항:

- 각각의 차원 테이블 조인에 따라 사실 테이블에 인덱스를 작성하십시오.
- 인덱스 ANDing 연산자의 비트 필터를 할당할 수 있을 만큼 정렬 힙 임계값이 높은지 확인하십시오. 스타 조인에서는 12MB 또는 3000 4K 페이지가 필요합니다. 파티션 내 병렬 처리를 위해 공유 정렬 힙과 동일한 공유 메모리 세그먼트에서 비트 필터가 할당되고, *sorthcap* 데이터베이스 구성 매개변수 및 *sheapthres_shr* 데이터베이스 구성 매개변수에 의해 바운드됩니다.
- 차원 테이블에 대한 필터링 술어를 적용하십시오. 현재 통계가 아니면 `runstats` 명령을 사용하여 통계를 갱신하십시오.

정적 SQL 또는 XQuery

정적 SQL 또는 XQuery문은 응용프로그램에 임베디드(embedded)됩니다. 모든 임베디드(embedded) 명령문은 응용프로그램이 실행되기 전에 미리 컴파일하여 패키지로 바운드해야 합니다.

정적 SQL에서 XQuery 표현식을 실행하려면 XMLQUERY 함수를 사용하십시오.

데이터베이스 관리 프로그램이 이러한 명령문을 컴파일하면, 명령문을 사전 컴파일하고 바운드할 때 카탈로그 통계 및 구성 매개변수를 기준으로 각각에 액세스 플랜이 작성됩니다.

이 액세스 플랜은 응용프로그램을 실행할 때 항상 사용됩니다. 따라서 이 플랜은 패키지를 다시 바운드해야 변경할 수 있습니다.

정적 SQL 또는 XQuery의 대안으로 동적 SQL 또는 XQuery가 있습니다.

시스템 관리 테이블 스페이스

데이터베이스에 존재할 수 있는 테이블 스페이스 유형은 시스템 관리 스페이스(SMS)와 데이터베이스 관리 스페이스(DMS)입니다.

SMS 테이블 스페이스는 운영 체제를 통해 관리되고, 테이블 스페이스를 작성할 때 지정한 스페이스에 데이터베이스 데이터를 저장합니다. 테이블 스페이스 정의에는 이 데이터가 저장된 하나 이상의 디렉토리 경로 목록이 포함됩니다.

파일 시스템은 미디어 스토리지 할당과 관리를 관리합니다.

SMS와 DMS 테이블 스페이스는 동일한 데이터베이스에 함께 존재할 수 있습니다.

테이블 스페이스

대형 데이터베이스를 테이블 스페이스라는 관리 파트로 파티션하면 더욱 쉽게 관리할 수 있습니다. 테이블 스페이스를 사용하면 특정 논리적 디바이스나 부분으로 데이터 위치를 지정할 수 있습니다.

예를 들면 테이블을 작성할 때 긴 오브젝트 또는 대형 오브젝트(LOB) 데이터의 인덱스 또는 긴 컬럼이 나머지 테이블 데이터가 되지 않도록 지정할 수 있습니다.

테이블 스페이스를 하나 이상의 실제 스토리지 디바이스(컨테이너)에 분산시키면 성능을 향상시킬 수 있습니다. 그러나 테이블 스페이스에 있는 모든 디바이스 또는 컨테이너가 유사한 성능 특성을 갖는 것이 좋습니다.

테이블 스페이스는 시스템 관리 스페이스(SMS) 또는 데이터베이스 관리 스페이스(DMS)와 같이 두 가지 방법으로 관리할 수 있습니다.

Visual Explain

Visual Explain을 사용하면 Explain된 SQL 또는 XQuery문에 대한 액세스 플랜을 그래프로 볼 수 있습니다. 그래프에 있는 정보를 사용하여 쿼리를 조정하면 성능을 향상시킬 수 있습니다.

중요사항: 제어 센터 도구로 Visual Explain에 액세스하는 것은 버전 9.7에서 지원되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오. IBM Data Studio 도구 세트를 사용하여 Visual Explain 기능에 액세스하는 것은 지원되지 않습니다.

Visual Explain을 사용하면 다음을 수행할 수 있습니다.

- 최적화 시 사용된 통계를 볼 수 있습니다. 패키지를 리바인드하면 성능이 개선될 수 있는지 판별할 때 이 통계와 현재 카탈로그 통계를 비교할 수 있습니다.
- 인덱스를 사용하여 테이블에 액세스했는지 판별할 수 있습니다. 인덱스를 사용하지 않은 경우에는 Visual Explain을 통해 어떠한 컬럼을 인덱스해야 유리한지 판별합니다.
- 쿼리에 대한 액세스 플랜 그래프의 이전 버전과 이후 버전을 비교하는 등 다양한 성능 조정 기술을 수행한 효과를 볼 수 있습니다.
- 계산된 총 비용 및 검색 행 수(카디널리티)를 비롯하여 액세스 플랜의 각 조작 정보를 가져올 수 있습니다.

액세스 플랜 그래프에 표시되는 세부사항은 다음과 같습니다.

- 테이블(및 그와 관련된 컬럼)과 인덱스
- 연산자(예: 테이블 스캔, 정렬 및 조인)
- 테이블 스페이스 및 함수

주: Visual Explain은 명령행에서 호출할 수 없으나 제어 센터의 여러 가지 데이터베이스 오브젝트에서만 호출할 수 있습니다.

Visual Explain을 시작하려면 다음을 수행하십시오.

- 제어 센터에서 데이터베이스 이름을 마우스 오른쪽 단추로 누르고 **Explain**된 명령문 실행기록 표시 또는 **Explain** 쿼리를 선택하십시오.
- 명령 편집기의 대화식 페이지 또는 스크립트 페이지에서 Explain 가능한 명령문을 실행하십시오.
- Query Patroller의 관리 쿼리 등록 정보 노트북 또는 실행기록 쿼리 등록 정보 노트북에서 액세스 플랜 표시를 누르십시오.

제 7 장 Visual Explain 태스크



이 절에는 Visual Explain을 사용하여 수행할 수 있는 관련 태스크 목록이 포함되어 있습니다.


명령 편집기를 사용하여 액세스 플랜 작성

명령 편집기를 사용하면 SQL 및 XQuery문, IMS™ 명령 및 DB2 명령을 생성, 편집, 실행 및 조작할 수 있습니다.

또한 결과를 출력하여 작업하고, Explain된 SQL문의 액세스 플랜을 그래픽 표현으로 볼 때도 명령 편집기가 사용됩니다. Linux® 및 Windows®, z/OS 및 OS/390® 시스템과 서브시스템 및 IMSplexes용 DB2 데이터베이스에서 명령과 SQL문을 실행할 수 있습니다.

명령 편집기를 사용하여 액세스 플랜을 작성하려면 다음을 수행하십시오.

1. 명령 편집기를 여십시오. 독립형 명령 편집기를 열려면 시작 -> 프로그램 -> **IBM DB2** -> 명령행 도구 -> 명령 편집기를 선택하십시오.
2. 대화식 또는 스크립트 탭을 선택한 후 다음을 수행하십시오.
 - a. 데이터베이스에 연결하십시오. (명령을 실행하려면 텍스트 영역에 연결 명령을 입력한 후, 2단계에서 선택한 페이지에 따라 대화식 또는 스크립트 메뉴에서 실행을 선택하십시오. 또는  아이콘을 누르거나 **Ctrl+Enter** 키를 누르십시오.
 - b. 명령문을 실행하지 않고 액세스 플랜을 작성하려면, 텍스트 영역에 Explain 가능한 명령문을 입력한 후 대화식 또는 스크립트 메뉴에서 액세스 플랜 작성을 선택하거나  아이콘을 누르십시오. 액세스 플랜 페이지에 액세스 플랜 그래프가 표시됩니다.

기존 스크립트에서 Explain 가능한 명령문을 선택할 수도 있습니다.
3. 액세스 플랜을 작성하고 명령문도 실행하려면 다음을 수행하십시오.
 - a. 대화식 또는 스크립트 메뉴에서 옵션을 선택하십시오. 명령 센터 옵션 노트북이 열립니다. 액세스 플랜 탭을 누르십시오. 자동으로 액세스 플랜 생성 선택란을 선택하십시오.
 - b. 텍스트 영역에 Explain 가능한 명령문을 입력하거나 기존 명령문을 선택하십시오. 대화식 또는 스크립트 메뉴에서 실행을 선택하거나  아이콘을 누르십시오. 결과가 결과 페이지에 표시됩니다. 생성된 액세스 플랜을 보려면 액세스 플랜 탭을 누르십시오.

액세스 플랜의 그래픽 표현 보기

액세스 플랜 그래프 창을 사용하면 Explain된 SQL 또는 XQuery문의 액세스 플랜을 그래픽 표현으로 볼 수 있습니다.

그래프의 노드에는 테이블 및 인덱스와 각각의 조작이 표시됩니다. 노드 사이의 링크는 데이터 플로우를 보여줍니다.

태스크

- 그래프를 인쇄하고, SQL 또는 XQuery문을 동적으로 Explain하며, 텍스트 또는 최적화 텍스트를 보거나 최적화 매개변수 또는 통계를 보려면 명령문 메뉴를 사용하십시오.
- 노드에서 세부사항 또는 통계를 보거나 각각의 연산자에 대한 추가 도움말을 가져오려면 노드 메뉴를 사용하십시오.
- 그래프 설정을 변경하거나 그래프 개요를 보려면 보기 메뉴를 사용하십시오. 이 태스크는 특히 대형 그래프에 유용합니다.

이 창에서는 다음 오브젝트에 대한 세부사항을 볼 수 있습니다.

- 테이블 스페이스 및 테이블 스페이스 통계
- 함수 및 함수 통계
- 연산자
- 파티션된 데이터베이스
- 피연산자
 - 컬럼 분포 통계
 - 인덱스 및 인덱스 통계
 - 페이지 페치 쌍 통계
 - 컬럼 그룹
 - 참조된 컬럼, 참조된 컬럼 그룹 및 참조된 컬럼 통계
 - 테이블 함수 통계 및 테이블 통계

액세스 플랜 그래프 창을 열려면 다음 방법 중 하나를 사용하십시오.

1. Explain 가능한 명령문 또는 Explain된 명령문 실행기록 창을 여십시오. 명령문-> 액세스 플랜 표시를 선택하십시오. 액세스 플랜 그래프 창이 열립니다.
2. Explain 가능한 명령문 또는 Explain된 명령문 실행기록 창에서 **Explain** 쿼리를 호출하십시오. 동적으로 Explain하고 나면 Explain 쿼리 명령문 창이 열립니다.

액세스 플랜 그래프 창 내용 읽기

창의 맨 위 영역

액세스 플랜 그래프 창의 맨 위 영역은 그래프에 표시된 액세스 플랜의 명령문을 식별합니다.

여기에 표시되는 다른 항목은 다음과 같습니다.

- 명령문의 Explain 날짜, 시간, 패키지 이름 및 버전
- 명령문 작성 시 페더레이티드 함수가 사용 가능한 경우
- 계산된 총 비용
- 이 명령문이 Explain된 시스템의 병렬 처리 유형. 이와 같은 유형은 다음과 같습니다.
 - 없음
 - 파티션 내 병렬 처리
 - 파티션 간 병렬 처리
 - 전체 병렬 처리(파티션 내 및 파티션 간)

그래프

그래프의 노드는 피연산자(테이블, 인덱스 또는 테이블 함수) 및 수행되는 연산자를 나타냅니다. 노드에 대한 세부 통계 정보를 보려면 노드를 더블 클릭하십시오.

그래프에 표시된 정보를 더 자세히 보려면 확대/축소 슬라이더를 위 아래로 끌어보십시오.

부동 소수점 값이 과학적 기수법으로 표시됩니다.

문제점 해결 추가 정보

- LONGDATACOMPAT 사용시 액세스 플랜 검색
- 이전 및 추후 릴리스에 대한 Visual Explain 지원

이미 Explain된 쿼리 명령문 보기

Explain된 명령문 실행기록 창을 사용하면 선택한 데이터베이스에 대해 이미 Explain된 SQL 또는 XQuery문의 실행기록을 볼 수 있습니다.

각각의 항목은 다음과 연관된 Explain된 명령문입니다.

- 패키지의 정적 SQL 또는 XQuery문
- 동적 SQL 또는 XQuery문

태스크

- 액세스 플랜의 그래픽 표현을 보고 쿼리 명령문을 동적으로 Explain하며 쿼리 명령문 텍스트를 보거나 쿼리 명령문을 변경 또는 제거하려면 명령문 메뉴를 사용하십시오.

- Explain 가능한 명령문을 정렬, 필터 또는 사용자 정의하려면 보기 메뉴 또는 보조 도구모음의 아이콘을 사용하십시오. 또한 이 메뉴에 있는 옵션을 사용하여 창 내용을 저장할 수 있습니다.

Explain된 명령문 실행기록 창을 열려면 다음 중 하나를 수행하십시오.

- 제어 센터에서 오브젝트 트리를 펼쳐서 데이터베이스 폴더를 찾고, 원하는 데이터베이스를 찾을 때까지 폴더를 펼친 후 다음 중 하나를 수행하십시오.
 - 데이터베이스를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain**된 명령문 실행기록 표시를 선택하거나 선택->Explain된 명령문 실행기록을 선택하십시오.
 - 데이터베이스를 강조표시한 후 선택->Explain된 명령문 실행기록을 선택하십시오.
- 제어 센터에서 오브젝트 트리를 펼쳐서 패키지 폴더(응용프로그램 오브젝트 폴더 아래 있음)를 찾으십시오. 그런 다음 다음을 수행하십시오.
 - 패키지 폴더를 누르십시오. 창 오른쪽에 기존 패키지 오브젝트가 표시됩니다.
 - 원하는 패키지를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain**된 명령문 실행기록을 선택하거나, 패키지를 강조표시한 후 선택->Explain된 명령문 실행기록을 선택하십시오. 아니면 패키지를 더블 클릭하십시오.
- Explain 가능한 명령문 창에서 명령문->Explain된 명령문 실행기록 표시를 선택하십시오.

Explain 가능한 명령문 창에 명령문이 선택된 경우에는 Explain된 명령문 실행기록 창에 선택한 SQL문과 관련된 Explain된 명령문이 모두 표시됩니다.

명령문이 선택되지 않은 경우에는 Explain된 명령문 실행기록 창에 Explain 가능한 명령문이 있는 패키지와 관련된 Explain된 명령문이 모두 표시됩니다.

Explain된 명령문 실행기록 창에는 Explain 테이블의 존재 여부에 따라 Explain된 명령문이 포함되거나 포함되지 않을 수도 있습니다.

Explain된 명령문 실행기록 창 내용 읽기

창에 있는 컬럼은 Explain된 쿼리 명령문에 대해 다음과 같은 정보를 제공합니다.

패키지 이름

패키지 이름은 다음 중 하나를 수행합니다.

- SQL 또는 XQuery문을 포함합니다(정적 쿼리의 경우).
- SQL 또는 XQuery문을 발행합니다(동적 쿼리의 경우).

패키지 작성자

패키지를 작성한 사용자의 사용자 ID입니다.

패키지 버전

패키지의 버전 번호입니다.

Explain 스냅샷

SQL 또는 XQuery문에 Explain 스냅샷을 선택했는지 설명합니다. (선택하지 않은 경우에는 명령문에 대한 액세스 플랜 그래프를 볼 수 없습니다.)

최종 바인드

명령문이 패키지에 포함되어 있으면 이 필드는 명령문과 바인드된 최종 패키지와의 연관 여부를 나타냅니다.

동적 Explain

Explain된 쿼리 명령문이 동적인지 설명합니다. (동적이 아니면 해당 명령문은 패키지의 정적 또는 SQL 또는 XQuery문입니다.)

Explain 날짜

명령문이 Explain 조사를 수행한 날짜입니다.

Explain 시간

명령문이 Explain 조사를 수행한 시간입니다.

전체 비용

명령문의 계산된 전체 비용(timeron)입니다.

명령문 번호

응용프로그램 소스 모듈의 SQL 또는 XQuery문 행 번호입니다.

섹션 번호

SQL 또는 XQuery문과 연관된 패키지의 섹션 번호입니다.

쿼리 번호

명령문과 연관된 쿼리 번호입니다.

쿼리 태그

명령문과 연관된 쿼리 태그입니다.

쿼리 텍스트

원래의 SQL 또는 XQuery문의 처음 100자입니다. (창 맨 아래에 있는 화면 이동 막대를 사용하여 스크롤하십시오.) 전체 SQL 또는 XQuery문을 보려면 명령문->쿼리 텍스트 표시를 선택하십시오.

설명

명령문과 연관된 설명입니다. (예를 들어 정적 쿼리 명령문에서의 설명은 명령문이 포함된 패키지와 연관되어 있습니다.)

문제점 해결 추가 정보

- 88 페이지의 『LONGDATACOMPAT 사용시 액세스 플랜 검색』
- 89 페이지의 『이전 및 추후 릴리스에 대해서 Visual Explain 지원』

패키지에 대해 Explain 가능한 명령문 보기

Explain 가능한 명령문 창을 사용하면 선택한 패키지에 대해 Explain 가능한 쿼리 명령문을 볼 수 있습니다.

명령문에 Explain 스냅샷이 선택된 경우에는 이 목록을 사용하여 명령문에 대한 추가 정보(예: 전체 비용 및 액세스 플랜의 그래픽 보기)를 볼 수 있습니다.

태스크

- 이미 Explain된 SQL 또는 XQuery문의 실행기록 및 액세스 플랜의 그래픽 표현을 보고 쿼리 명령문을 동적으로 Explain하며 쿼리 명령문 텍스트를 보려면 명령문 메뉴를 사용하십시오.
- Explain 가능한 명령문을 정렬, 필터 또는 사용자 정의하려면 보기 메뉴 또는 보조 도구모음의 아이콘을 사용하십시오. 또한 이 메뉴에 있는 옵션을 사용하여 창 내용을 저장할 수 있습니다.

Explain 가능한 명령문 창을 열려면 다음을 수행하십시오.

- 제어 센터에서 오브젝트 트리를 펼쳐서 패키지 폴더(응용프로그램 오브젝트 폴더 아래에 있음)를 찾으십시오.
- 패키지 폴더를 누르십시오. 창 오른쪽의 분할창에 기존 패키지 오브젝트가 표시됩니다.
- 다음 중 하나를 수행하십시오.
 - 원하는 패키지를 마우스 오른쪽 단추로 누른 후 팝업 메뉴에서 **Explain 가능한 명령문** 표시를 선택하십시오.
 - 패키지를 강조표시한 후 선택->Explain 가능한 명령문 표시를 선택하십시오.
 - 패키지를 더블 클릭하십시오.

Explain 가능한 명령문 창 내용 읽기

창에 있는 컬럼은 SQL 또는 XQuery문에 대해 다음과 같은 정보를 제공합니다.

명령문 번호

응용프로그램 소스 모듈의 SQL 또는 XQuery문 행 번호입니다. 정적 쿼리의 경우 이 번호는 SYSCAT.STATEMENTS 테이블의 STMTNO 컬럼에 해당합니다.

섹션 번호

SQL 또는 XQuery문과 연관된 패키지의 섹션 번호입니다.

Explain 스냅샷

SQL 또는 XQuery문에 Explain 스냅샷을 선택했는지 설명합니다. (선택하지 않은 경우에는 명령문에 대한 액세스 플랜 그래프를 볼 수 없습니다.)

전체 비용

선택한 SQL 또는 XQuery문에 대해 쿼리 결과를 리턴하는 계산된 전체 비용 (timeron)입니다. (명령문이 포함된 패키지를 이미 Explain한 경우에만 사용할 수 있습니다.)

쿼리 텍스트

쿼리 명령문의 처음 100자입니다. (창 맨 아래에 있는 화면이동 막대를 사용하여 스크롤하십시오.) 전체 SQL 또는 XQuery문을 보려면 명령문->쿼리 텍스트 표시를 선택하십시오.

문제점 해결 추가 정보

- 88 페이지의 『LONGDATACOMPAT 사용시 액세스 플랜 검색』
- 89 페이지의 『이전 및 추후 릴리스에 대해서 Visual Explain 지원』

인덱스 작성 지침

적절한 인덱스를 작성하면 옵티마이저는 인덱스 스캔을 선택하여 테이블 스캔보다 더 효율적으로 사용할 수 있습니다.

인덱스 작성에 대한 안내서에서 다루는 내용은 다음과 같습니다.

- 적용할 기본 키와 고유 인덱스를 정의합니다.
- 테이블을 조인(*Join* 술어)할 때 쿼리가 사용할 컬럼에 인덱스를 작성합니다.
- 일반적인 기준으로 특정 값을 검색할 컬럼에 인덱스를 작성합니다.
- ORDER BY절에 공통으로 사용되는 컬럼에 인덱스를 작성합니다.
- 필요한 데이터만 검색하는 술어를 사용했는지 확인합니다. 예를 들면, 술어의 술어 선택성 값이 리턴하려는 테이블의 일부를 나타내는지 확인하십시오.
- 다중 컬럼 인덱스를 작성하면 인덱스의 첫 번째 컬럼은 쿼리에서 술어로 가장 자주 사용되는 컬럼이 됩니다.
- 디스크 및 갱신 유지보수 오버헤드 인덱스 소개가 너무 높아지지 않는지 확인하십시오.

이전 액세스 플랜

이전 액세스 플랜인 경우 통계를 갱신한 후 패키지를 리바인드해야 합니다.

증상 STATS_TIME 행은 갱신되지 않은 통계를 나타냅니다.

가능한 원인

옵티마이저는 디폴트값을 사용했습니다. (이 디폴트값은 "디폴트"라는 키워드로 표시됩니다.) 이로 인해 액세스 플랜의 유효 기간이 만료됩니다.

조치 Runstats 명령 사용을 통해 통계를 갱신한 후 패키지를 리마인드하는 것이 좋습니다.

LONGDATACOMPAT 사용시 액세스 플랜 검색

LONGDATACOMPAT 사용시 액세스 플랜을 검색할 수 없는 경우 데이터베이스 별명을 작성하고 다시 시도하십시오.

증상

Visual Explain을 사용하여 표시할 수 있는 Explain된 명령문 실행기록 또는 액세스 플랜이 없습니다.

가능한 원인

db2cli.ini 파일에 LONGDATACOMPAT 값이 1로 설정되어 있으면 Visual Explain 액세스 플랜을 생성할 수는 있으나 검색할 수 없습니다.

조치

일시적인 해결책으로 LONGDATACOMPAT가 0으로 설정된 데이터베이스의 데이터베이스 별명을 작성할 수 있습니다.

예:

```
DB2 UPDATE CLI CFG FOR SECTION db-alias-name USING LONGDATACOMPAT 0
```

다음 명령을 사용하면 CLI 구성 값을 점검할 수 있습니다.

```
GET CLI CONFIGURATION [AT GLOBAL LEVEL] [FOR SECTION section-name]
```

예를 들어 데이터베이스 별명이 sample인 경우:

```
GET CLI CONFIGURATION FOR SECTION sample
```

RUNSTATS 사용

옵티마이저는 데이터베이스의 카탈로그 테이블을 사용해 데이터베이스 정보, 상당한 양의 데이터 및 기타 특성을 가져오고, 이 정보를 토대로 하여 데이터에 액세스하는 가장 좋은 방법을 선택합니다.

현재 통계를 사용할 수 없으면, 옵티마이저는 정확하지 않은 디폴트 통계에 따라 비효율적인 액세스 플랜을 선택할 수 있습니다.

runstats 명령을 사용하여 테이블 및 인덱스에 대한 현재 통계를 수집하는 것이 바람직 한데, 이는 마지막으로 runstats 명령을 실행한 이후 새 인덱스가 작성되었거나 중요한 갱신 활동이 발생한 경우에 특히 권장됩니다. 이는 옵티마이저에 최상의 액세스 플랜을 판별할 수 있는 가장 정확한 정보를 제공합니다.

테이블을 갱신하지 않고 runstats를 사용하면 빈 테이블이 옵티마이저에 표시될 수 있습니다. 이러한 문제점은 연산자 세부사항 창에 대한 카디널리티가 0인 경우 입증됩니다. 이 경우, 테이블 갱신을 완료하고 runstats 명령을 재실행한 후 영향을 받는 테이블에 대한 Explain 스냅샷을 재작성하십시오.

참고:

- 쿼리로 액세스할 수 있는 모든 테이블과 인덱스에 runstats를 사용하십시오.
- Quantile 및 자주 사용되는 값 통계는 데이터가 균등하지 않게 분산되는 시기를 판별합니다. 이 값을 갱신하려면 WITH DISTRIBUTION절이 있는 테이블에서 runstats를 사용하십시오.
- 통계뿐만 아니라 기타 요인(예: 규정 행 순서 지정, 테이블 크기 및 버퍼 풀 크기)도 액세스 플랜 선택 방법에 영향을 줄 수 있습니다.
- Runstats 명령을 실행하거나 구성 매개변수를 변경하고 나면 응용프로그램이 리바인드됩니다(명령문이 다시 Explain되는 경우도 있습니다).

Runstats 명령(CLP 프롬프트에서 입력할 수 있음)은 다음 구문에 표시된 대로 여러 가지 레벨의 통계를 제공할 수 있습니다.

기본 통계

테이블:

```
RUNSTATS ON TABLE tablename
```

인덱스:

```
RUNSTATS ON TABLE tablename FOR INDEXES ALL
```

테이블 및 인덱스:

```
RUNSTATS ON TABLE tablename AND INDEXES ALL
```

확장 통계

테이블:

```
RUNSTATS ON TABLE tablename WITH DISTRIBUTION
```

인덱스:

```
RUNSTATS ON TABLE tablename FOR DETAILED INDEXES ALL
```

테이블 및 인덱스:

```
RUNSTATS ON TABLE tablename WITH DISTRIBUTION AND  
DETAILED INDEXES ALL
```

주: 위 명령 각각의 tablename은 스키마 이름이 있는 전체 이름이어야 합니다.

이전 및 추후 릴리스에 대해서 Visual Explain 지원

버전 9에서 생성된 스냅샷은 버전 8에서 생성된 스냅샷과 다릅니다.

버전 8 데이터베이스에 액세스하는 버전 9 클라이언트에서 Visual Explain을 실행하는 경우 Visual Explain은 버전 8 스냅샷을 핸들하지 않습니다. Visual Explain은 이전 릴리스 호환성을 지원합니다.

그러나 버전 9 데이터베이스에 액세스하는 버전 8 클라이언트에서 Visual Explain을 실행하는 경우 Visual Explain은 버전 9 데이터를 구문 분석할 때 오류를 리턴합니다. 버전 9에서 생성된 스냅샷이 버전 8에서 생성된 스냅샷과 다르기 때문에 Visual Explain은 이러한 상위 레벨 호환성을 지원하지 않습니다.

제 3 부 부록

부록 A. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
 - 주제 항목(태스크, 개념 및 참조 항목)
 - DB2 도구에 대한 도움말
 - 샘플 프로그램
 - 자습서
- DB2 서적
 - PDF 파일(다운로드)
 - PDF 파일(DB2 PDF DVD)
 - 인쇄된 서적
- 명령행 도움말
 - 명령 도움말
 - 메시지 도움말

주: DB2 정보 센터의 주제는 PDF 또는 하드카피 서적보다 더 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 ibm.com에서 DB2 정보 센터를 참조하십시오.

[ibm.com](http://www.ibm.com)에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks® 서적)를 온라인으로 액세스할 수 있습니다. 다음은 DB2 정보 관리 라이브러리 소프트웨어 사이트의 주소입니다. <http://www.ibm.com/software/data/sw-library/>

문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안 사항이 있는 경우 db2docs@ca.ibm.com으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 한 구체적인 예를 제공하십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때는 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center(www.ibm.com/shop/publications/order)에서 사용할 수 있는 DB2 라이브러리에 대해 설명합니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책이 나와 있는 경우에도, 사용 국가 또는 지역에서 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: DB2 정보 센터는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 1. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	예	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
<i>DB2 Text Search Guide</i>	SC27-2459-00	예	2009년 8월
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	예	2009년 8월
<i>Developing Embedded SQL Applications</i>	SC27-2445-00	예	2009년 8월
<i>Developing Java Applications</i>	SC27-2446-00	예	2009년 8월

표 1. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	아니오	2009년 8월
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-00	예	2009년 8월
<i>Getting Started with Database Application Development</i>	GI11-9410-00	예	2009년 8월
<i>Linux 및 Windows에서 DB2 설치 및 관리 시작하기</i>	GA30-3960-00	예	2009년 8월
<i>자국어 안내서</i>	SA30-3972-00	예	2009년 8월
<i>DB2 Servers 설치</i>	GA30-3962-00	예	2009년 8월
<i>IBM Data Server Clients 설치</i>	GA30-3963-00	아니오	2009년 8월
<i>Message Reference Volume 1</i>	SC27-2450-00	아니오	2009년 8월
<i>Message Reference Volume 2</i>	SC27-2451-00	아니오	2009년 8월
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-00	아니오	2009년 8월
<i>파티셔닝 및 클러스터링 안내서</i>	SA30-3973-00	예	2009년 8월
<i>pureXML Guide</i>	SC27-2465-00	예	2009년 8월
<i>Query Patroller 관리 및 사용자 안내서</i>	SA30-3974-00	아니오	2009년 8월
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	아니오	2009년 8월
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	예	2009년 8월
<i>SQL 참조서, 볼륨 1</i>	SA30-3956-00	예	2009년 8월
<i>SQL 참조서, 볼륨 2</i>	SA30-3957-00	예	2009년 8월
<i>문제점 해결 및 데이터베이스 성능 조정</i>	SA30-3952-00	예	2009년 8월
<i>DB2 버전 9.7로 업그레이드</i>	SA30-3961-00	예	2009년 8월
<i>Visual Explain 자습서</i>	SA30-3968-00	아니오	2009년 8월

표 1. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
DB2 버전 9.7의 새로운 내용	SA30-3967-00	예	2009년 8월
Workload Manager Guide and Reference	SC27-2464-00	예	2009년 8월
XQuery Reference	SC27-2466-00	아니오	2009년 8월

표 2. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
DB2 Connect Personal Edition 설치 및 구성	SA30-3965-00	예	2009년 8월
DB2 Connect Servers 설치 및 구성	SA30-3966-00	예	2009년 8월
DB2 Connect 사용자 안내서	SA30-3964-00	예	2009년 8월

표 3. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
Information Integration: Administration Guide for Federated Systems	SC19-1020-02	예	2009년 8월
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-04	예	2009년 8월
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	아니오	2009년 8월
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	예	2009년 8월
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	예	2009년 8월

인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에서 가능한 것은 아닙니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 일부 소프트웨어 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들면, DB2 메시지 참조서의 어떤 볼륨도 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이 불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신의 완전한 DB2 문서는 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 <http://www.ibm.com/shop/publications/order>의 IBM Publications Center를 확인하십시오. 서적 주문 정보를 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
 1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
 - www.ibm.com/planetwide에 있는 IBM 전세계 문의처 디렉토리
 - <http://www.ibm.com/shop/publications/order>의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
 2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
 3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 94 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.

예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.

DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>입니다.

DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 해당 주제 항목을 영어로 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.

1. Internet Explorer에서 도구 → 인터넷 옵션 → 언어 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 추가... 단추를 누르십시오.

주: 언어를 추가하더라도 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 컴퓨터에 설치되지 않습니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.

1. 도구 → 설정 → 내용 대화 상자의 언어 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 언어 선택 창에서 원하는 언어를 선택한 다음 추가... 단추를 누르십시오.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

일부 브라우저 및 운영 체제 조합에서는 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

시작하기 전에

DB2 버전 9.7 정보 센터는 미리 설치된 상태여야 합니다. 자세한 내용은 *DB2 Servers* 설치의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 사용할 수 없는 시간이 매우 짧다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정할 수도 있습니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스로 설치되어 있으며, 수동 갱신을 통해 기존 정보 센터의 기능 및 언어 갱신뿐만 아니라 독일어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제의 경우
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.

- b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
- c. 다음과 같이 ic-update 스크립트를 실행하십시오.

```
ic-update
```

2. Windows 운영 체제의 경우

- a. 명령 창을 여십시오.
- b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
- c. 설치 디렉토리에서 doc\bin 디렉토리로 이동하십시오.
- d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

```
ic-update.bat
```

결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc\weclipse\configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다. 예: 1239053440785.log

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다.
2. 사용 가능한 갱신사항을 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결되고 DB2 정보 센터가 설치된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템으로 미리하십시오. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미리링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.

갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.

3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.

주: Windows 2008, Windows Vista 이상의 경우 이 절 다음에 나오는 명령은 관리자 권한으로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축 아이콘을 마우스 오른쪽 단추로 누른 후 관리자로 실행을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.

1. DB2 정보 센터를 중지하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 중지를 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.
`/etc/init.d/db2icdv97 stop`

2. 독립형 모드에서 정보 센터를 시작하십시오.

- Windows의 경우:
 - a. 명령 창을 여십시오.
 - b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>#IBM#DB2 Information Center#Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
 - c. 설치 디렉토리에서 doc#bin 디렉토리로 이동하십시오.
 - d. 다음과 같이 help_start.bat 파일을 실행하십시오.
`help_start.bat`
- Linux의 경우:
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.
 - b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
 - c. 다음과 같이 help_start 스크립트를 실행하십시오.
`help_start`

시스템의 기본 웹 브라우저가 열리고 독립형 정보 센터가 표시됩니다.

3. 갱신 단추(🔄)를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 갱신사항 찾기를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.

4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 갱신사항 설치를 누르십시오.

5. 설치 프로세스가 완료되면 완료를 누르십시오.

6. 독립형 정보 센터를 중지하십시오.

- Windows의 경우, 설치 디렉토리의 doc\win 디렉토리로 이동한 후 다음과 같이 help_end.bat 파일을 실행하십시오.

```
help_end.bat
```

주: help_end 일괄처리 파일에는 help_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start.bat 를 중지할 때 Ctrl+C 또는 다른 메소드를 사용하지 마십시오.

- Linux의 경우, 설치 디렉토리의 doc/bin 디렉토리로 이동한 후 다음과 같이 help_end 스크립트를 실행하십시오.

```
help_end
```

주: help_end 스크립트에는 help_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start 스크립트를 중지할 때 다른 메소드를 사용하지 마십시오.

7. DB2 정보 센터를 재시작하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 시작을 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.

```
/etc/init.d/db2icdv97 start
```

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

DB2 자습서

DB2 자습서는 DB2 제품의 여러가지 측면을 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

시작하기 전에

정보 센터(<http://publib.boulder.ibm.com/infocenter/db2help/>)에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 필요한 전제조건 설명은 자습서를 참조하십시오.

DB2 자습서

자습서를 보려면 제목을 누르십시오.

『pureXML[®]』(pureXML Guide)

DB2 데이터베이스를 설정하여 XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행할 수 있습니다.

Visual Explain 자습서의 『Visual Explain』

더 나은 성능을 위해 Visual Explain을 사용하여 SQL문을 분석, 최적화 및 조정할 수 있습니다.

DB2 문제점 해결 정보

DB2 데이터베이스 제품 사용 시 발생하는 광범위한 문제점을 판별하고 해결하는 데 도움이 되는 정보를 사용할 수 있습니다.

DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 절을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등에 관한 정보가 있습니다.

DB2 기술 지원 웹 사이트

문제점이 발생한 경우 해당 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대해 사용 가능한 솔루션을 찾을 수 있습니다.

다음은 DB2 기술 지원 웹 사이트의 주소입니다. http://www.ibm.com/software/data/db2/support/db2_9/

이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

개인적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 귀하 사업장 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서의 2차적 저작물을 만들거나 본 문서 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 문서의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 본 문서의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

부록 B. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

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 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 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 『현상 태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 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 사이트 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) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 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, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 "있는 그대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. `_enter 연도_`. All rights reserved.

상표

IBM, IBM 로고 및 `ibm.com`[®]은 여러 국가에 등록된 IBM Corp.의 상표 또는 등록 상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 “저작권 및 상표 정보”(www.ibm.com/legal/copytrade.shtml)에 있습니다.

다음 용어는 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java[™] 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX[®]는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel[®], Intel 로고, Intel Inside[®], Intel Inside 로고, Intel[®] Centrino[®], Intel Centrino 로고, Celeron[®], Intel[®] Xeon[®], Intel SpeedStep[®], Itanium[®] 및 Pentium[®]은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft[®], Windows, Windows NT[®] 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

색인

[가]

갱신사항
DB2 정보 센터 99, 100

[다]

데이터베이스 관리 스페이스(DMS)
정의 70
도움말
언어 구성 98
SQL문 97
동적 SQL 또는 XQuery문
정의 70

[마]

명령
EXPLAIN.DDL 3
RUNSTATS, 사용 88
vesampl.ddl 4
명령 편집기
액세스 플랜 추가 81
문서
개요 93
이용약관 103
인쇄됨 94
PDF 94
문제점 판별
사용 가능 정보 103
지습서 103
문제점 해결
온라인 정보 103
지습서 103

[바]

비용
정의 69

[사]

서적
인쇄됨
주문 96
술어
정의 74
술어의 선택성
정의 76
스타 조인
정의 77
시스템 관리 테이블 스페이스
정의 78

[아]

액세스 플랜
개요 67
그래픽 표현 보기 82
명령 편집기에서 작성 81
이전 87
인덱스와 통계 없는 쿼리
단일 파티션 데이터베이스 환경에서 16
파티션된 데이터베이스 환경에서 34
추가 인덱스 작성
단일 파티션 데이터베이스 환경에서 28
파티션된 데이터베이스 환경에서 45
테이블을 조인하는 데 사용되는 컬럼에 인덱스 작성
단일 파티션 데이터베이스 환경에서 23
파티션된 데이터베이스 환경에서 40
항상
단일 파티션 데이터베이스 환경에서 15
파티션된 데이터베이스 환경에서 33
현재 통계 수집
단일 파티션 데이터베이스 환경에서 19
파티션된 데이터베이스 환경에서 37
LONGDATACOMPAT 사용시 검색 88
액세스 플랜 그래프 67
기호 읽기 10
노드 68
모양 변경 13
이전에 Explain된 SQL 또는 XQuery문 목록의 9
자세한 내용 가져오기 11
표시 및 사용 9

엑세스 플랜 그래프 (계속)

확대 10

연산자

정의 72

DELETE 51

EISCAN 51

FETCH 연산자 52

FILTER 52

GENROW 53

GRPBY 53

HSJOIN 53

INSERT 54

IXAND 54

IXSCAN 55

MSJOIN 56

NLJOIN 56

RETURN 57

RIDSCN 57

RPD 58

SHIP 58

SORT 58

TBSCAN 59

TEMP 60

TQ 60

UNION 62

UNIQUE 62

UPDATE 62

XANDOR 62

XISCAN 64

XSCAN 66

옵티마이저

정의 74

이용약관

서적 사용 103

인덱스

작성 87

파티션된 데이터베이스 환경의 테이블 컬럼 45

[자]

자습서

문제점 판별 103

문제점 해결 103

Visual Explain v, 102

작성

인덱스 87

정적 SQL 또는 XQuery문

정의 77

주의사항 105

[카]

커서

블로킹

정의 69

컨테이너

정의 69

쿼리 최적화 클래스

정의 75

클러스터링

정의 68

[타]

테이블 스페이스

정의 78

[파]

패키지

정의 74

Explain 가능한 명령문 보기 86

피연산자

정의 72

[하]

행 블로킹

커서 블로킹 확인 69

C

CREATE INDEX문

개요 87

D

DB2 서적 주문 96

DB2 정보 센터

갱신 99, 100

다른 언어로 보기 98

버전 98

언어 98

DELETE 연산자

정의 51

E

EISCAN 연산자

정의 51

Explain 가능한 명령문

보기 86

정의 71

Explain 기능

샘플 XML 문서 조각 76

Explain 스냅샷 3

동적 SQL 또는 XQuery문의 6

정의 70

정적 SQL 또는 XQuery문의 6

Visual Explain 자습서의 4

Explain 테이블

작성 3

Explain된 SQL문

실행기록 보기 83

정의 71

Explain된 XQuery문

실행기록 보기 83

정의 71

EXPLAIN.DDL

명령 3

explsnap 옵션 6

F

FETCH 연산자

정의 52

FILTER 연산자

정의 52

G

GENROW 연산자

정의 53

GRPBY 연산자

정의 53

H

HSJOIN 연산자

정의 53

I

INSERT 연산자

정의 54

IXAND 연산자

정의 54

IXSCAN 연산자

정의 55

L

LONGDATACOMPAT

액세스 플랜 검색 88

M

MSJOIN 연산자

정의 56

N

NLJOIN 연산자

정의 56

R

RETURN 연산자

정의 57

RIDSCN 연산자

정의 57

RPD 연산자

정의 58

RUNSTATS 명령

사용 88

S

SHIP 연산자

정의 58

SORT 연산자

정의 58

SQL문

도움말 표시 97

explain됨

실행기록 보기 83

T

TBSCAN 연산자

정의 59

TEMP 연산자

정의 60

TQ 연산자

정의 60

U

UNION 연산자

정의 62

UNIQUE 연산자

정의 62

UPDATE 연산자

정의 62

V

vesampl.ddl 명령 4

Visual Explain

액세스 플랜 67

액세스 플랜 그래프 67

이전 및 추후 릴리스에 대한 지원 90

자습서 v, 1, 102

정의 79

Visual Explain 자습서

개념 67

태스크 51, 81

X

XANDOR 연산자

샘플 문서 76

정의 62

XISCAN 연산자

샘플 문서 76

정의 64

XQuery문

explain됨

실행기록 보기 83

XSCAN 연산자

샘플 문서 76

정의 66



SA30-3968-00



Spine information:

Linux, UNIX 및 Windows용 IBM DB2 9.7

Visual Explain 지습서

