

IBM® DB2® Universal Database



SQL 시작하기

버전 7

IBM® DB2® Universal Database



SQL 시작하기

버전 7

이 책의 정보와 지원하는 제품을 사용하기 전에 반드시 129 페이지의 『부록C. 주의사항』을 읽으십시오.

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

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

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

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

목차

환영	v	테이블 함수	35
관련 책	v	그룹화	35
강조표시 규약	vi	GROUP BY 절에 WHERE 절 사용	36
제1장 관계형 데이터베이스와 SQL	1	GRUOUP BY 절 다음에 HAVING 절 사용	36
제2장 데이터 구성	3	제5장 표현식과 부속 조회	39
테이블	3	스칼라 fullselect	39
뷰	4	데이터 유형 변환	40
스키마	4	CASE 표현식	40
데이터 유형	5	테이블 표현식	42
제3장 테이블과 뷰의 작성	9	중첩 테이블 표현식	42
테이블 작성	9	공통 테이블 표현식	43
데이터 삽입	10	상관 이름	45
데이터 변경	13	상관 부속 조회	46
데이터 삭제	13	상관 부속 조회 구현	48
뷰 작성	14	제6장 조회에서 연산자와 술어 사용	53
뷰를 사용한 데이터 조작	16	집합 연산자로 조회 결합	53
제4장 SQL 문을 사용하여 데이터에 액세스	19	UNION 연산자	53
데이터베이스에 연결	20	EXCEPT 연산자	55
오류 조사	21	INTERSECT 연산자	55
컬럼 선택	21	술어	56
행 선택	22	IN 술어 사용	56
행 정렬	25	BETWEEN 술어 사용	57
중복 행 제거	27	LIKE 술어 사용	57
연산 순서	28	EXISTS 술어 사용	58
표현식을 사용하여 값 계산	28	규정화된 술어	59
표현식 이름 지정	29	제7장 고급 SQL	61
한 개 이상의 테이블에서 데이터 선택	30	제한조건 및 트리거가 있는 비즈니스 규칙 시 행	61
부속 조회 사용	31	키	62
함수 사용	32	고유한 제한조건	62
컬럼 함수	33	참조 무결성 제한조건	63
스칼라 함수	34		

테이블 점검 제한조건	64	STAFFG 테이블	98
트리거	65	BLOB 및 CLOB 데이터 유형으로 된 샘플	
조인	70	파일	99
복합 조회	75	Quintana 사진	100
ROLLUP 및 CUBE 조회	75	Quintana 이력서	100
순환 조회	75	Nicholls 사진	102
OLAP 함수	76	Nicholls 이력서	102
		Adamson 사진	104
제8장 데이터 조작의 조정 및 향상	77	Adamson 이력서	104
사용자 정의 유형	78	Walker 사진	106
사용자 정의 함수	79	Walker 이력서	106
대형 오브젝트(LOB)	80	부록B. DB2 라이브러리 사용.	109
대형 오브젝트(LOB) 조작	80	DB2 PDF 파일 및 인쇄된 책	109
특수 레지스터	81	DB2 정보	109
카탈로그 뷰 소개	82	PDF 책 인쇄	119
시스템 카탈로그에서 행 선택	82	인쇄된 책 주문	119
부록A. 샘플 데이터베이스 테이블.	85	DB2 온라인 문서	121
샘플 데이터베이스	86	온라인 도움말 액세스	121
샘플 데이터베이스 작성	86	정보 온라인 보기	123
샘플 데이터베이스 삭제	87	DB2 마법사 사용	125
CL_SCHED 테이블	87	문서 서버 설정	126
DEPARTMENT 테이블	87	정보 온라인 검색	127
EMPLOYEE 테이블	88	부록C. 주의사항	129
EMP_ACT 테이블	91	등록 상표	132
EMP_PHOTO 테이블	93	색인	135
EMP_RESUME 테이블	93	IBM에 문의	141
IN_TRAY 테이블	94	제품 정보	141
ORG 테이블	94		
PROJECT 테이블	95		
SALES 테이블	96		
STAFF 테이블	97		

환영

이 책은 사용자에게 SQL 및 관계형 데이터베이스를 소개하고자 하는 책입니다. 이 책에는 다음과 같은 내용이 들어 있습니다.

- DB2 제품에 사용되는 SQL의 기본 개념 설명
- 데이터베이스 조작 TASK 수행 방법 설명
- 간단한 예를 통한 TASK 설명

사용자가 관리자라면, 이 책에 있는 예를 시도해 보기 전에 먼저 다음과 같은 작업을 해야 합니다.

- 빠른 시작 책에 설명된 대로 사용자의 운영 체제에 서버를 설치하고 구성하십시오. "첫단계" 옵션을 사용하여 SAMPLE 데이터베이스를 작성하십시오. SAMPLE 데이터베이스는 명령행 프롬프트로부터 작성할 수도 있습니다. SQL 참조서에서 자세한 내용을 참조하십시오. 개인적인 데이터는 DB2 SAMPLE 데이터베이스에 넣지 않도록 권장합니다.
- 빠른 시작 책에 있는 지침에 따라 DB2 관리자 사용자 ID를 작성하십시오.

시스템 관리자가 아니라면, 유효한 사용자 ID 및 SAMPLE 데이터베이스에 액세스하기 위한 권한 및 특권이 있는지 확인하십시오.

관련 책

다음과 같은 책을 이용할 수 있습니다.

빠른 시작	데이터베이스 관리 프로그램을 설치하고 사용하는 데 필요한 정보가 들어 있습니다.
SQL 참조서	SQL 참조 정보가 들어 있습니다.
관리 안내서	지역 또는 클라이언트/서버 환경에서 액세스할 수 있는 데이터베이스를 설계, 구현 및 유지보수하는 데 필요한 정보가 들어 있습니다.
응용프로그램 개발 안내서	응용프로그램 개발 프로세스와 데이터베이스에 액세스하기 위해 Embedded SQL을 사용하는 응용프로그램을 코딩하고, 컴파일하며, 실행할 수 있는 방법과 SQL 프로시저어 언어(또는 다른 지원되는 프로그래밍 언어)를 사용하여 DB2 저장 프로시저어를 수행하는 방법에 대해 논의합니다.

강조표시 규약

이 책에서 사용되는 규약은 다음과 같습니다.

굵은체	예를 들면, 시스템이 미리 정의한 명령과 키워드를 가리킵니다.
이탤릭체	다음 중 하나를 나타냅니다. <ul style="list-style-type: none">• 새로운 용어 소개• 또다른 정보 소스에 대한 참조
대문자체	다음 중 하나를 나타냅니다. <ul style="list-style-type: none">• 시스템이 미리 정의한 명령과 키워드• 특정 데이터 값과 컬럼 이름의 예

제1장 관계형 데이터베이스와 SQL

관계형 데이터베이스에서, 데이터는 테이블에 저장됩니다. 테이블은 행과 컬럼의 집합입니다. 4 페이지의 그림1에서 테이블의 그래픽 예에 대한 자세한 내용을 참조하십시오. 그림에 컬럼(세로) 및 행(가로)이 표시되어 있습니다. SQL은 컬럼, 테이블 및 그들 간의 여러 관계를 지정하여 데이터를 검색 또는 갱신하는 데 사용됩니다.

SQL은 관계형 데이터베이스에서 데이터를 정의하고 조작하기 위한 표준화된 언어입니다. SQL 문은 데이터베이스 관리 프로그램에 의해 실행됩니다. 데이터베이스 관리 프로그램은 데이터를 관리하는 컴퓨터 프로그램입니다.

파티션된 관계형 데이터베이스는 복수 파티션(노드라고도 함)을 통해 데이터를 관리하는 관계형 데이터베이스입니다. 파티션을 이해하는 간단한 방법으로는 각 파티션을 물리적 컴퓨터로 생각하는 것입니다. 이 책에서는 단일 파티션 데이터베이스에 대해 주로 설명합니다.

명령행 처리기(CLP) 또는 명령 센터(CC)와 같은 인터페이스를 사용하여 대화식 SQL을 통해 샘플 데이터베이스에 액세스하고 이 책의 모든 예를 시험해 볼 수 있습니다.

제2장 데이터 구성

여기에는 테이블, 뷰 그리고 스키마에 대한 중요한 개념적 설명이 제시되어 있습니다. 이 설명은 관계형 데이터베이스의 다양한 구축 블록 간의 연결을 보여주는 일반 개요입니다. 마지막 절에서는 중요하고 자주 사용하는 데이터 유형 중 몇 가지에 대해 간략하게 설명합니다.

테이블

테이블은 정의된 수의 컬럼과 여러 행으로 구성된 논리 구조입니다. 컬럼은 동일한 데이터 유형을 가진 값 세트입니다. 행은 테이블 내의 단일 행을 구성하는 일련의 값입니다. 행은 테이블 안에서 반드시 정렬되어 있지는 않습니다. 결과 집합을 정렬하려면, 테이블에서 데이터를 선택하는 SQL 문에 명확히 순서를 지정하기만 하면 됩니다. 모든 컬럼과 행의 교차점에는 값이라는 특정 데이터 항목이 있습니다. 4 페이지의 그림1에서 'Sanders'는 테이블 값의 예입니다.

기본 테이블은 사용자 데이터를 보유하며 CREATE TABLE 문을 사용하여 작성됩니다. 결과 테이블은 조화를 만족시키기 위해서 데이터베이스 관리 프로그램이 하나 이상의 기본 테이블에서 선택하거나 생성하는 일련의 행입니다.

4 페이지의 그림1에서는 테이블의 한 예를 보여줍니다. 컬럼과 행이 표시되어 있습니다.

	컬럼			
행	ID	NAME	DEPT	J
	10	Sanders	20	Mg
	20	Pernal	20	Sa
	30	Marenghi	38	Mg
	40	O'Brien	38	Sa
	50	Hanes	15	Mg
	60	Quigley	38	Sa
			15	Sa

그림 1. 테이블

뷰

뷰는 하나 이상의 테이블에서 데이터를 보는 다른 방법을 제공합니다. 뷰는 테이블에 대한 동적인 창입니다.

뷰를 이용하면, 여러 사용자가 같은 데이터의 상이한 표현을 볼 수 있습니다. 예를 들면, 여러 사용자는 직원들에 대한 데이터가 있는 하나의 테이블에 액세스할 수 있습니다. 관리자는 자기 직원에 대한 데이터는 볼 수 있으나 다른 부서 직원의 데이터는 볼 수 없습니다. 신규 채용 부서 관리자는 모든 직원의 입사일을 볼 수 있으나 급여는 볼 수 없는 반면, 경리부 관리자는 급여를 볼 수 있으나 입사일은 볼 수 없습니다. 각각의 사용자는 실제 테이블에서 끌어낸 뷰로 작업합니다. 각각의 뷰는 테이블로 나타나며 각각의 이름이 있습니다.

뷰를 이용하면, 민감한 데이터에 대한 액세스를 제어할 수 있다는 이점이 있습니다. 그래서 여러 사람들이 같은 데이터의 다른 컬럼이나 행에 액세스할 수 있습니다.

스키마

스키마는 이름 지정된 오브젝트(예를 들어, 테이블 및 뷰)의 집합입니다. 스키마는 데이터베이스 내의 오브젝트에 대한 논리적 뷰를 제공합니다.

스키마는 테이블, 뷰 또는 기타 이름 지정된 오브젝트를 작성할 때 내재적으로 작성됩니다. 또는 'CREATE SCHEMA' 문을 사용하여 명시적으로 스키마를 작성할 수도 있습니다.

이름 지정된 오브젝트를 작성할 때, 특정 스키마 이름으로 오브젝트 이름을 규정할(연관지을) 수 있습니다. 이름 지정된 오브젝트의 이름은 두 부분으로 구성되는데, 첫번째 부분은 오브젝트가 할당되는 스키마 이름입니다. 스키마 이름을 지정하지 않으면, 오브젝트는 기본 스키마에 할당됩니다. 기본 스키마의 이름은 명령문을 실행하는 사용자의 권한 부여 ID입니다.

이 책의 예를 실행하는 데 사용되는 방법인 대화식 SQL에서, 권한 부여 ID는 CONNECT 문에 지정된 사용자 ID입니다. 예를 들어, 테이블 이름이 STAFF이고 지정된 ID가 USERXYZ이면, 규정된 테이블 이름은 USERXYZ.STAFF입니다. CONNECT에 대해서는 20 페이지의 『데이터베이스에 연결』에서 자세한 내용을 보십시오.

어떤 스키마 이름은 예약되어 있습니다. 예를 들어, 내장 함수(built-in function)는 SYSIBM 스키마에 있는 반면, 사전 설치된 사용자 정의 함수는 SYSFUN 스키마에 있습니다. SQL 참조서에서 CREATE SCHEMA 문에 대한 자세한 내용을 참조하십시오.

데이터 유형

데이터 유형은 상수, 컬럼, 호스트 변수, 함수, 표현식 그리고 특수 레지스터에 대한 허용값을 정의합니다. 여기에서는 예에서 언급된 데이터 유형에 대해 설명합니다. SQL 참조서에서 다른 데이터 유형에 대한 전체 목록과 자세한 설명을 참조하십시오.

문자열

문자열은 일련의 바이트입니다. 문자열의 길이는 문자열에 있는 바이트 수입니다. 길이가 0인 경우, 그 값은 빈 문자열이라고 합니다.

고정 길이 문자열

CHAR(x)은 고정 길이 문자열입니다. 길이 속성 x는 1 - 254 이 내여야 합니다.

가변 길이 문자열

가변 길이 문자열은 VARCHAR, LONG VARCHAR 및 CLOB 유형입니다.

VARCHAR(x) 유형은 가변 길이 문자열입니다. 그래서 길이 9의 문자열은 VARCHAR(15)에 삽입할 수 있지만 여전히 그 문자열의 길이는 9가 될 것입니다.

80 페이지의 『대형 오브젝트(LOB)』에서 CLOB에 대한 자세한 내용을 참조하십시오.

그래픽 문자열

그래픽 문자열은 일련의 2 바이트 문자 데이터입니다.

고정 길이 그래픽 문자열

GRAPHIC(x)은 고정 길이 문자열입니다. 길이 속성 x는 1 - 127 이내여야 합니다.

가변 길이 그래픽 문자열

가변 길이 그래픽 문자열의 유형은 VARGRAPHIC, LONG VARGRAPHIC 및 DBCLOB입니다. 80 페이지의 『대형 오브젝트(LOB)』에서 DBCLOB에 대한 자세한 내용을 참조하십시오.

2진 문자열

2진 문자열은 일련의 바이트입니다. 2진 문자열은 사진과 같은 데이터를 보관하는데 사용됩니다. 2진 대형 오브젝트(BLOB)는 2진 문자열입니다. 80 페이지의 『대형 오브젝트(LOB)』에서 자세한 내용을 참조하십시오.

숫자

모든 숫자에는 부호와 정밀도가 있습니다. 정밀도는 부호를 제외한 비트 또는 자리수입니다.

SMALLINT

SMALLINT(작은 정수)는 5자리의 정밀도를 가진 2 바이트 정수입니다.

INTEGER

INTEGER(큰 정수)는 10자리의 정밀도를 가진 4 바이트 정수입니다.

BIGINT

BIGINT(큰 정수)는 19 자리로 된 8 바이트 정수입니다.

REAL

REAL(단정밀도 부동 소수점)은 실수의 32 비트 근사치입니다.

DOUBLE

DOUBLE(배정밀도 부동 소수점)은 실수의 64 비트 근사치입니다. *DOUBLE*은 또한 *FLOAT*이라고도 합니다.

DECIMAL(p,s)

*DECIMAL*은 소수입니다. 소수점의 위치는 숫자의 정밀도(*p*) 및 스케일(*s*)에 의해 결정됩니다. 정밀도는 전체 자리수이고 32보다 작아야 합니다. 스케일은 소수부에 있는 자리수이고 항상 정밀도 값보다 작거나 같아야 합니다. 정밀도와 스케일을 지정하지 않으면 소수값은 정밀도 5 및 스케일 0의 기본값을 갖게 됩니다.

날짜 시간 값

날짜 시간 값은 날짜, 시간 및 시간소인(*yyyyxxddhhmmss* 형식으로 유효한 날짜 및 시간을 나타내는 14자리의 문자열)을 나타냅니다. 날짜 시간 값은 문자열도 아니고 숫자도 아니지만, 특정 연산 및 문자열 조작에서 사용할 수 있고 특정 문자열과 호환됩니다.¹

날짜 날짜는 연, 월, 일 세 부분으로 구성됩니다.

시간 시간은 시, 분, 초 세 부분으로 구성되며 24시간을 이용합니다.

시간소인

시간소인은 날짜와 시간을 나타내는 연, 월, 일, 시, 분, 초, 마이크로초 일곱 부분으로 구성되어 있습니다.

널(Null) 값

1. 이 책에 날짜 시간 값의 ISO 표현이 설명되어 있습니다.

널(*Null*) 값은 모든 널이 아닌 값과 다른 특별한 값입니다. 널 값은 행 안에 그 컬럼에 대한 어떤 다른 값도 없다는 것을 의미합니다. 널 값은 모든 데이터 유형에 존재합니다.

다음 표에는 예에서 사용된 데이터 유형의 특징이 설명되어 있습니다. 모든 숫자 데이터 유형은 특정 범위 내에서 정의됩니다. 이 표에는 또한 숫자 데이터 유형의 범위도 포함되어 있습니다. 이 표는 적절한 데이터 유형을 사용할 때 빠른 참조로 사용할 수 있습니다.

데이터 유형	유형	특징	예 또는 범위
CHAR(15)	고정 길이 문자 열	254의 최대 길이	'Sunny day '
VARCHAR(15)	가변 길이 문자 열	32672의 최대 길이	'Sunny day'
SMALLINT	숫자	2 바이트 5 자리의 정밀도	범위는 32768 - 32767입니다.
INTEGER	숫자	4 바이트 10 자리의 정밀도	범위는 2147483648 - 2147483647입니다.
BIGINT	숫자	8 바이트 19 자리의 정밀도	범위는 -9223372036854775808에서 9223372036854775807까지입니다.
REAL	숫자	단정밀도 부동 소숫점 약 32 비트	범위는 $-3.402E+38$ 에서 $-1.175E-37$ 또는 $1.175E-37$ 에서 $-3.402E+38$ 또는 0입니다.
DOUBLE	숫자	배정밀도 부동 소숫점 약 64 비트	범위는 $-1.79769E+308$ 에서 $-2.225E-307$ 또는 $2.225E-307$ 에서 $1.79769E+308$ 또는 0입니다.
DECIMAL(5, 2)	숫자	정밀도는 5이고 스케일은 2입니다.	범위는 $-10^{**3}1+1$ 에서 $10^{**3}1-1$ 입니다.
DATE	날짜 시간	세 부분 값	1991-10-27
TIME	날짜 시간	세 부분 값	13.30.05
TIMESTAMP	날짜 시간	일곱 부분 값	1991-10-27-13.30.05.000000

보다 자세한 정보는 *SQL 참조서*에 있는 데이터 유형 호환성 테이블을 참조하십시오.

제3장 테이블과 뷰의 작성

여기에서는 DB2 Universal Database에서 테이블과 뷰를 작성하고 조작하는 방법에 대해 설명합니다. 도표와 예를 통해 테이블과 뷰의 관계를 설명합니다.

여기에는 다음과 같은 내용이 들어 있습니다.

- 테이블 작성 및 뷰 작성
- 데이터 삽입
- 데이터 변경
- 데이터 삭제
- 뷰를 사용한 데이터 조작

테이블 작성

CREATE TABLE 문을 사용하여 컬럼 이름 및 유형 그리고 제한조건을 지정하여 자신의 테이블을 작성하십시오. 제한조건은 61 페이지의 『제한조건 및 트리거가 있는 비즈니스 규칙 시행』에 설명되어 있습니다.

다음은 PERS라는 이름을 갖는 테이블을 작성하는 명령문입니다. 이 테이블은 STAFF 테이블과 비슷하지만 생년월일에 대한 컬럼이 추가되어 있습니다.

```
CREATE TABLE PERS
  (ID          SMALLINT NOT NULL,
   NAME       VARCHAR(9),
   DEPT       SMALLINT WITH DEFAULT 10,
   JOB        CHAR(5),
   YEARS      SMALLINT,
   SALARY     DECIMAL(7,2),
   COMM       DECIMAL(7,2),
   BIRTH_DATE DATE)
```

이 명령문으로 작성된 테이블에는 데이터가 들어 있지 않습니다. 다음 절에서는 데이터를 새 테이블에 삽입하는 방법에 대해 설명합니다.

예에서 보는 것처럼, 각 컬럼에 대해 이름 및 데이터 유형을 지정하십시오. 데이터 유형은 5 페이지의 『데이터 유형』에 설명되어 있습니다. NOT NULL은 생략 가능하고 널(Null) 값이 컬럼에서 허용될 수 없다는 것을 나타내기 위해 지정할 수 있습니다. 기본값 또한 옵션입니다.

CREATE TABLE 문에서는 고유한 제한조건(*unique constraints*) 또는 참조 제한조건(*referential constraints*)과 같이 사용자는 많은 다른 옵션들을 지정할 수 있습니다. 옵션에 대한 자세한 정보는 SQL 참조서에 있는 CREATE TABLE 문을 보십시오.

데이터 삽입

새 테이블을 작성하면 그 안에는 어떠한 데이터도 없습니다. 테이블에 새 행을 입력하려면, INSERT 문을 사용해야 합니다. 이 명령문은 두 가지 일반적인 양식이 있습니다.

- 한 양식은 VALUES 절을 사용하여 하나 이상의 행의 컬럼에 값을 지정하는 것입니다. 다음 세 가지 예는 이 일반적인 양식을 사용하여 테이블에 데이터를 삽입하는 것입니다.
- 또다른 양식은 VALUES가 아니라 *fullselect*를 지정하여 다른 테이블 및/또는 뷰에 포함된 행의 컬럼을 식별하는 것입니다.

*fullselect*는 INSERT 또는 CREATE VIEW 문 또는 술어 다음에 사용하는 선택 문입니다. 괄호 안에 들어 있는 *fullselect*는 부속 조화라고 합니다.

테이블을 작성할 때 선택한 기본 옵션에 따라, 삽입한 모든 행에 대해서 각 컬럼에 값을 주거나 기본값을 수락해야 합니다. 여러 데이터 유형의 기본값은 SQL 참조서에 나와 있습니다.

다음은 VALUES 절을 사용하여 데이터의 한 행을 PERS 테이블에 삽입하는 것입니다.

```
INSERT INTO PERS
VALUES (12, 'Harris', 20, 'Sales', 5, 18000, 1000, '1950-1-1')
```

다음은 VALUES 절을 사용하여 세 행을 PERS 테이블에 삽입하라는 명령문입니다. 이 테이블에는 ID, NAME 그리고 JOB만이 나와 있습니다. 어떤 컬럼이 NOT NULL로 정의되어 있으면서도 기본값을 가지고 있지 않다면, 반드시 컬럼 값을 지정해야 합니다.

CREATE TABLE 문에 있는 컬럼 정의에 대한 NOT NULL 절은 WITH DEFAULT를 사용하여 확장할 수 있습니다. 어떤 컬럼이 NOT NULL WITH DEFAULT 또는 WITH DEFAULT 10과 같은 기본 상수 값으로 정의되어 있고 사용자가 컬럼 목록에서 컬럼을 지정하지 않았다면, 기본값이 삽입된 행에 있는 그 컬럼에 삽입됩니다. 예를 들어, CREATE TABLE 문에는 기본값이 DEPT 컬럼에만 지정되었고 그것은 10으로 정의되어 있습니다. 따라서, 부서 번호(DEPT)는 10으로 설정되고 값이 제공되지 않은 다른 컬럼은 널(Null)로 설정됩니다.

```
INSERT INTO PERS (NAME, JOB, ID)
VALUES ('Swagerman', 'Prgmr', 500),
       ('Limoges', 'Prgmr', 510),
       ('Li', 'Prgmr', 520)
```

다음은 삽입 결과를 리턴합니다.

```
SELECT *
FROM PERS
ID      NAME      DEPT  JOB  YEARS  SALARY  COMM  BIRTH_DATE
-----
12 Harris      20 Sales    5  18000.00  1000.00  01/01/1950
500 Swagerman  10 Prgmr    -      -      -
510 Limoges    10 Prgmr    -      -      -
520 Li         10 Prgmr    -      -      -
```

이 때, 값이 모든 컬럼에 지정되어 있지 않은지 살펴 보십시오. NULL 값은 대쉬 (-)로 표시됩니다. 이 작업을 위해서는 컬럼 이름의 목록이 순서와 데이터 유형에서 VALUES 절에 있는 값과 일치해야 합니다. (첫번째 예에서 처럼) 컬럼 이름의 목록이 생략되어 있으면, VALUES 다음의 데이터 값 목록은 테이블의 컬럼과 똑같은 순서가 되어야 하고 값의 수는 테이블에 있는 컬럼의 수와 같아야 합니다.

각각의 값은 삽입된 컬럼의 데이터 유형과 호환되어야 합니다. 컬럼이 널(Null) 입력 가능으로 정의되고 그 컬럼의 값이 지정되어 있지 않다면, 삽입된 행에 있는 그 컬럼에는 널 값이 주어집니다.

다음 예는 행에 있는 컬럼들에 값이 지정되어 있지 않기 때문에 YEARS, COMM 및 BIRTH_DATE에 널(Null) 값을 삽입합니다.

```
INSERT INTO PERS (ID, NAME, JOB, DEPT, SALARY)
VALUES (410, 'Perna', 'Sales', 20, 20000)
```

INSERT 문의 두번째 양식을 사용하면, 한 테이블에 있는 행 값을 다른 테이블에 쉽게 삽입할 수 있습니다. 위에서 언급한대로, VALUES를 지정하기보다는 fullselect를 지정하여 다른 테이블 및/또는 뷰에 들어 있는 행에서 컬럼을 식별할 수 있습니다.

다음 예는 STAFF 테이블에서 부서 번호(DEPT) 38에 대한 데이터를 선택하여 그 데이터를 PERS 테이블에 삽입합니다.

```
INSERT INTO PERS (ID, NAME, DEPT, JOB, YEARS, SALARY)
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM STAFF
WHERE DEPT = 38
```

이렇게 삽입한 후에는, 다음 SELECT 문은 INSERT 문에 있는 fullselect와 같은 결과를 산출합니다.

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM PERS
WHERE DEPT = 38
```

결과는 다음과 같습니다.

ID	NAME	DEPT	JOB	YEARS	SALARY
30	Marengi	38	Mgr	5	17506.75
40	O'Brien	38	Sales	6	18006.00
60	Quigley	38	Sales	-	16808.30
120	Naughton	38	Clerk	-	12954.75
180	Abrahams	38	Clerk	3	12009.75

데이터 변경

UPDATE 문을 사용하여 테이블에 있는 데이터를 변경하십시오. 이 명령문을 사용하면, WHERE 절의 검색 조건을 만족시키는 각 행에서 하나 이상의 컬럼 값을 변경할 수 있습니다.

다음 예는 ID가 410인 직원에 대한 정보를 갱신합니다.

```
UPDATE PERS
  SET JOB='Prgmr', SALARY = SALARY + 300
  WHERE ID = 410
```

SET 절은 갱신할 컬럼을 지정하고 그 값을 제공합니다.

WHERE 절은 생략 가능하고 갱신할 행을 지정합니다. WHERE 절이 생략되면, 데이터베이스 관리 프로그램이 테이블 또는 뷰에 있는 모든 행을 사용자가 지정한 값으로 갱신합니다.

이 예에는, 처음에는 테이블(PERS)이 이름 지정되어 있고, 그 다음에는 갱신할 행에 대한 조건이 지정되어 있습니다. ID가 410인 직원에 대한 정보는 다음과 같이 변경되었습니다. 직함은 Prgmr로 변경되었고 급여는 \$300 올랐습니다.

2개 이상의 행에 적용할 수 있는 WHERE 절을 포함하면 2개 이상의 행에서 데이터를 변경할 수 있습니다. 다음 예는 모든 영업사원의 봉급을 15%씩 인상합니다.

```
UPDATE PERS
  SET SALARY = SALARY * 1.15
  WHERE JOB = 'Sales'
```

데이터 삭제

DELETE 문을 사용하여 WHERE 절에 지정된 검색 조건에 기초한 테이블에서 데이터에 관한 행을 삭제하십시오. 다음 예는 직원 ID가 120인 행을 삭제합니다.

```
DELETE FROM PERS
  WHERE ID = 120
```

WHERE 절은 생략 가능하고 삭제할 행을 지정합니다. WHERE 절이 생략되면, 데이터베이스 관리 프로그램이 테이블 또는 뷰에 있는 모든 행을 삭제합니다.

DELETE 문을 사용하면 두 개 이상의 행을 삭제할 수 있습니다. 다음 예는 직원 DEPT가 20인 모든 행을 삭제합니다.

```
DELETE FROM PERS
WHERE DEPT = 20
```

행을 삭제하면 전체 행을 삭제하는 것이고 그 행에 있는 특정한 컬럼을 삭제하는 것이 아닙니다.

테이블의 정의뿐만 아니라 그 내용도 삭제하려면, SQL 참조서에서 설명되어 있는 DROP TABLE 문을 사용하십시오.

뷰 작성

4 페이지의 『뷰』에서 설명한 대로, 뷰는 하나 이상의 테이블에서 데이터를 보는 다른 방법을 제공합니다. 뷰를 작성하여 다양한 사용자가 보고자 하는 정보를 제한할 수 있습니다. 다음 도표는 뷰와 테이블 사이의 관계를 보여줍니다.

15 페이지의 그림2에서, View_A는 Table_A의 컬럼 AC1 및 AC2에만 액세스를 제한합니다.

View_AB는 Table_A의 AC3 및 Table_B의 BC2에만 액세스를 허용합니다.

View_A를 작성하면 TABLE_A에 액세스할 수 있는 사용자를 제한할 수 있고, VIEW_AB를 작성하면 두 테이블 내의 특정 컬럼에 대한 액세스를 제한할 수 있습니다.

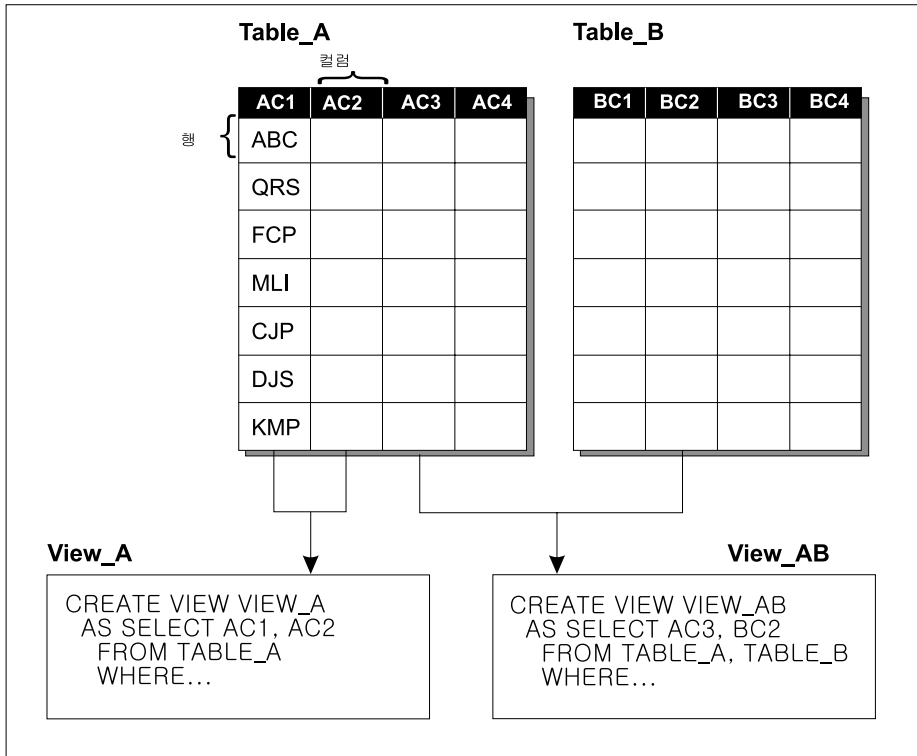


그림 2. 테이블과 뷰 사이의 관계

다음 예는 STAFF 테이블에 있는 부서(DEPT) 20의 비관리자들의 뷰를 작성합니다. 여기에는 급여 및 수당이 나타나지 않습니다.

```
CREATE VIEW STAFF_ONLY
AS SELECT ID, NAME, DEPT, JOB, YEARS
FROM STAFF
WHERE JOB <> 'Mgr' AND DEPT=20
```

다음은 작성한 후에 뷰의 내용을 표시합니다.

```
SELECT *
FROM STAFF_ONLY
```

이 명령문은 다음과 같은 결과를 산출합니다.

ID	NAME	DEPT	JOB	YEARS
20	Pernal	20	Sales	8
80	James	20	Clerk	-
190	Sneider	20	Clerk	8

좀더 복잡한 예로서, STAFF 및 ORG 테이블을 사용하여 각 부서의 이름과 부서장의 이름을 나열하는 뷰를 작성할 수 있습니다. 다음 명령문이 이 뷰를 작성합니다.

```
CREATE VIEW DEPARTMENT_MGRS
AS SELECT NAME, DEPTNAME
FROM STAFF, ORG
WHERE MANAGER = ID
```

뷰를 작성할 때 WITH CHECK OPTION 절을 사용하여 뷰를 통한 테이블의 삽입 및 갱신에 제한조건을 추가할 수 있습니다. 이 절을 이용하면, 데이터베이스 관리 프로그램이 뷰의 갱신 또는 뷰로의 삽입이 뷰의 정의와 일치하는지 검사할 수 있고 일치하지 않는 것들은 거부할 수 있습니다. 이 절을 생략하면, 뷰 정의에 위반되어서 뷰에 대한 삽입과 갱신을 검사할 수 없습니다. WITH CHECK OPTION 작동 방법에 대한 자세한 내용은 SQL 참조서에 있는 CREATE VIEW 문을 참조하십시오.

뷰를 사용한 데이터 조작

SELECT 문과 같이 뷰가 실제 테이블인 것처럼 INSERT, DELETE 및 UPDATE 문을 뷰에 적용합니다. 이들은 하위 기본 테이블에 있는 데이터를 조작합니다. 그래서 그 뷰에 다시 액세스할 때, 가장 최신의 기본 테이블을 사용해 그 뷰는 평가됩니다. WITH CHECK OPTION을 사용하지 않으면, 뷰를 사용해 수정한 데이터는 본래의 뷰 정의에 더 이상 적합하지 않기 때문에, 뷰에 반복적으로 액세스할 때 나타나지 않을 수도 있습니다.

다음은 FIXED_INCOME 뷰에 적용된 갱신의 예입니다.

```
CREATE VIEW FIXED_INCOME (LNAME, DEPART, JOBTITLE, NEWSALARY)
AS SELECT NAME, DEPT, JOB, SALARY
FROM PERS
WHERE JOB <> 'Sales' WITH CHECK OPTION
```

```
UPDATE FIXED_INCOME
SET NEWSALARY = SALARY * 1.10
WHERE LNAME = 'Li'
```


이전 뷰의 갱신은 다음과 같이 기본 테이블 PERS를 갱신하는 것(CHECK OPTION 제외)과 같습니다.

```
UPDATE PERS
  SET SALARY = SALARY * 1.10
  WHERE NAME = 'Li'
  AND JOB <> 'Sales'
```

뷰는 CREATE VIEW FIXED_INCOME에 있는 제한조건 JOB <> 'Sales'에 대한 WITH CHECK OPTION을 사용하여 작성되기 때문에 Limoges가 'Sales'로 이동할 때 다음의 갱신이 허용되지 않을 것입니다.

```
UPDATE FIXED_INCOME
  SET JOBTITLE = 'Sales'
  WHERE LNAME = 'Limoges'
```

SALARY + COMM 또는 SALARY * 1.25와 같은 표현식으로 정의된 컬럼은 갱신할 수 없습니다. 그러한 컬럼을 하나 이상 포함하여 뷰를 정의하면, 사용자는 이러한 컬럼에 대해 UPDATE 특권을 받을 수 없습니다. INSERT 문은 그러한 컬럼을 포함하는 뷰에서는 허용될 수 없지만 DELETE 문에서는 허용됩니다.

NOT NULL로 정의된 컬럼이 없는 PERS 테이블의 경우에는 FIXED_INCOME 뷰가 하위 테이블 PERS의 ID, YEARS, COMM 또는 BIRTHDATE를 포함하지 않더라도 그 뷰를 통해 PERS 테이블에 행을 삽입할 수 있습니다. 뷰를 통해 볼 수 없는 컬럼에는 NULL 또는 기본값이 적절하게 설정됩니다.

그러나, PERS 테이블은 NOT NULL로 정의된 컬럼 ID가 없습니다. FIXED_INCOME 뷰를 통해 행을 삽입하려 하면, 시스템은 뷰를 통해 『볼 수 없는』 모든 PERS 컬럼안에 널(Null) 값을 삽입하려고 시도합니다. ID 컬럼이 뷰에 포함되어 있지 않고 널(Null) 값을 허용하지 않기 때문에, 시스템은 뷰를 통한 삽입을 허용하지 않습니다.

뷰를 수정하는 규칙과 제한사항은 SQL 참조서에 있는 CREATE VIEW 문을 참조하십시오.

제4장 SQL 문을 사용하여 데이터에 액세스

여기에서는 SQL 문을 사용해 데이터베이스에 연결하여 데이터를 검색하는 방법에 대해 설명합니다.

예에서, 샘플 데이터베이스에 대해 명령문이 발행된 경우 표시될 결과 앞에 대부분 입력된 명령문이 표시됩니다. 명령문은 대문자로 되어 있지만, 사용자는 대문자와 소문자를 섞어서 명령문을 입력할 수 있습니다(텍스트가 따옴표(') 또는 (')로 묶여 있는 경우는 제외).

DB2 Universal Database에 포함된 SAMPLE 데이터베이스는 85 페이지의 『부록A. 샘플 데이터베이스 테이블』에 나열된 대로 다수의 테이블로 구성되어 있습니다. 이 데이터베이스는 "첫단계" 설치 런치패드를 사용하여 작성될 수 있습니다. 또한 명령행에서 SAMPLE 데이터베이스를 작성할 수 있습니다. *SQL 참조서*에서 자세한 내용을 참조하십시오.

DB2 Universal Database에 추가 샘플 데이터베이스가 들어 있으므로, 데이터 웨어하우스 센터 및 OLAP Starter 킷을 기능적으로 보여줄 수 있음을 주의하십시오. 이 책에 있는 예는 SAMPLE 데이터베이스를 사용한 것입니다.

데이터베이스를 설정한 방법에 따라, 테이블 이름 앞에 스키마 및 마침표를 붙여 사용된 테이블 이름을 규정해야 할 수 있습니다. 예를 들어, 이 책에서 기본 스키마는 USERID로 가정됩니다. 따라서 테이블 ORG는 USERID.ORG라고 할 수 있습니다. 이것이 필요한지 아닌지 관리자에게 문의하십시오.

여기에는 다음과 같은 내용이 들어 있습니다.

- 데이터베이스에 연결
- 오류 조사
- 컬럼 선택 및 행 선택
- 행 정렬 및 중복 행 제거
- 연산 순서

- 표현식을 사용하여 값 계산
- 표현식 이름 지정
- 한 개 이상의 테이블에서 데이터 선택
- 부속 조회 사용
- 함수 사용
- 그룹화

데이터베이스에 연결

데이터베이스를 조회하거나 조작하기 위해 SQL 문을 사용하려면, 먼저 데이터베이스에 연결해야 합니다. CONNECT 문을 사용하여 데이터베이스 연결과 사용자 이름을 연결할 수 있습니다.

예를 들어 SAMPLE 데이터베이스에 연결하려면, DB2 명령행 처리기에서 다음 명령을 입력하십시오.

```
CONNECT TO SAMPLE USER USERID USING PASSWORD
```

(서버 시스템에서 유효한 사용자 ID 및 암호를 선택했는지 확인하십시오)

이 예에서, USER는 USERID이고 USING은 PASSWORD입니다.

다음 메시지는 정상적인 연결이 수행되었다는 것을 보여줍니다.

데이터베이스 연결 정보

```
데이터베이스 제품      = DB2/NT 7.1.0
SQL 권한 부여 ID      = USERID
지역 데이터베이스 별명 = SAMPLE
```

일단 연결되면 데이터베이스 조작을 시작할 수 있습니다. 연결에 대한 자세한 내용은 SQL 참조서의 CONNECT 문을 참조하십시오.

오류 조사

예에 있는 내용 중 하나를 잘못 입력하거나 SQL 문을 실행하는 중에 오류가 발생하면, 데이터베이스 관리 프로그램이 오류 메시지를 리턴합니다. 오류 메시지는 메시지 식별자, 간략한 설명 그리고 SQLSTATE로 구성되어 있습니다.

SQLSTATE 오류는 DB2 제품군에 공통인 오류 코드입니다. SQLSTATE 오류는 ISO/ANSI SQL92 표준을 따릅니다.

예를 들어, CONNECT 문에서 사용자 ID 또는 암호가 틀리면 데이터베이스 관리 프로그램은 SQL1403N의 메시지 식별자 및 08004의 SQLSTATE를 리턴하게 됩니다. 메시지는 다음과 유사합니다.

```
SQL1403N 제공된 사용자 이름이나 암호가  
올바르지 않습니다. SQLSTATE=08004
```

물음표(?)와 메시지 식별자 또는 SQLSTATE를 입력하면 오류 메시지에 대한 더 많은 정보를 볼 수 있습니다.

```
? SQL1403N  
또는  
? SQL1403  
또는  
? 08004
```

오류 SQL1403N을 설명하는 끝에서 두 번째 라인에 SQLCODE가 -1403이라고 되어 있는지 살펴 보십시오. SQLCODE는 제품에 특정한 오류 코드입니다. N(통지) 또는 C(중요)로 끝나는 메시지 식별자는 오류를 나타내고 음수의 SQLCODE를 가집니다. W(경고)로 끝나는 메시지 식별자는 경고를 나타내고 양수의 SQLCODE를 가집니다.

컬럼 선택

SELECT 문을 사용하여 테이블에서 특정 컬럼을 선택하십시오. 명령문에서 쉽표로 분리된 컬럼 이름 목록을 지정하십시오. 이 목록을 선택 목록이라고 합니다.

다음 명령문은 SAMPLE 데이터베이스의 ORG 테이블에서 부서 이름(DEPTNAME) 및 부서 번호(DEPTNUMB)를 선택합니다.

```
SELECT DEPTNAME, DEPTNUMB
FROM ORG
```

위의 명령문은 다음과 같은 결과를 산출합니다.

DEPTNAME	DEPTNUMB
Head Office	10
New England	15
Mid Atlantic	20
South Atlantic	38
Great Lakes	42
Plains	51
Pacific	66
Mountain	84

별표(*)를 사용하여 테이블에서 모든 컬럼을 선택할 수 있습니다. 다음 예는 ORG 테이블의 모든 컬럼 및 행을 나열합니다.

```
SELECT *
FROM ORG
```

이 명령문은 다음과 같은 결과를 산출합니다.

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lakes	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

행 선택

테이블에서 특정 행을 선택하려면, SELECT 문 다음에 WHERE 절을 사용하여 행을 선택하기 위해 충족시켜야 할 조건을 지정하십시오. 테이블에서 행을 선택하는 기준은 검색 조건입니다.

검색 조건은 하나 이상의 술어로 구성됩니다. 술어는 행에 대한 참 또는 거짓(또는 알려지지 않은)인 조건을 지정합니다. 다음과 같은 기본적인 술어를 사용하여 WHERE 절의 조건을 지정할 수 있습니다.

술어	기능
$x = y$	x는 y와 같습니다.
$x \neq y$	x는 y와 같지 않습니다.
$x < y$	x는 y보다 작습니다.
$x > y$	x는 y보다 큼니다.
$x \leq y$	x는 y보다 작거나 같습니다.
$x \geq y$	x는 y보다 크거나 같습니다.
IS NULL/IS NOT NULL	널(Null) 값인지 테스트

검색 조건을 구성할 때는 숫자 데이터 유형에 대해서만 산술 연산을 하고 호환되는 데이터 유형에서만 비교를 해야 합니다. 예를 들어, 문자열을 숫자와 비교할 수 없습니다.

문자값을 기초로 한 행을 선택하면, 그 값은 따옴표로 묶어야 하고(예를 들면, WHERE JOB = 'Clerk') 각 문자값은 데이터베이스에 있는대로 정확히 입력해야 합니다. 데이터 값이 데이터베이스에 대문자로 되어 있는데 소문자로 입력하면 어떤 행도 선택되지 않습니다. 숫자값을 기초로 한 행을 선택하면 그 값은 따옴표로 묶어서는 안됩니다(예를 들면, WHERE DEPT = 20).

다음 예는 STAFF 테이블에서 부서(DEPT) 20의 행만을 선택합니다.

```
SELECT DEPT, NAME, JOB
      FROM STAFF
     WHERE DEPT = 20
```

이 명령문은 다음과 같은 결과를 산출합니다.

```
DEPT  NAME      JOB
-----
    20 Sanders   Mgr
    20 Pernal   Sales
    20 James    Clerk
    20 Sneider  Clerk
```

다음은 AND를 사용하여 하나 이상의 조건을 지정하는 명령문입니다. 원하는 만큼 조건을 지정할 수 있습니다. 이 예는 STAFF 테이블에서 부서(DEPT) 20에 있는 직원(STAFF)을 선택합니다.

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE JOB = 'Clerk'
AND DEPT = 20
```

이 명령문은 다음과 같은 결과를 산출합니다.

DEPT	NAME	JOB
20	James	Clerk
20	Sneider	Clerk

어떠한 값도 입력되어 있지 않고 컬럼이 기본값을 지원하지 않을 때 널(Null) 값이 발생합니다. 또한 값이 특수하게 널로 설정되어 있어도 널(Null) 값이 발생합니다. 널 값은 널 값을 지원하도록 정의된 컬럼에만 발생합니다. 테이블의 널 값 정의 및 지원은 9 페이지의 『테이블 작성』에서 논의됩니다.

술어 IS NULL 및 IS NOT NULL을 사용하여 널 값 여부를 점검하십시오.

다음 명령문은 수당(COMM)이 알려지지 않은 직원을 나열합니다.

```
SELECT ID, NAME
FROM STAFF
WHERE COMM IS NULL
```

이 명령문은 다음과 같은 결과를 산출합니다.

ID	NAME
10	Sanders
30	Marenghi
50	Hanes
100	Plotz
140	Fraye
160	Molinare
210	Lu
240	Daniels
260	Jones
270	Lea
290	Quill

값 0은 널 값과 같지 않습니다. 다음은 수당(COMM)이 0인 모든 직원을 테이블에서 선택하라는 명령문입니다.

```
SELECT ID, NAME
FROM STAFF
WHERE COMM = 0
```

샘플 테이블에 있는 COMM 컬럼에는 0값이 없기 때문에, 리턴된 결과 집합은 비어 있습니다.

다음은 STAFF 테이블에서 YEARS 값이 9 보다 큰 모든 행을 선택하라는 명령문입니다.

```
SELECT NAME, SALARY, YEARS
FROM STAFF
WHERE YEARS > 9
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	SALARY	YEARS
Hanes	20659.80	10
Lu	20010.00	10
Jones	21234.00	12
Quill	19818.00	10
Graham	21000.00	13

행 정렬

지정된 순서로 정보를 리턴할 수 있습니다. ORDER BY 절을 사용하여 하나 이상의 컬럼값을 기준으로 정보를 정렬하십시오.

다음 명령문은 부서(DEPT) 84에 있는 직원을 고용된 연수에 따라 순서대로 표시합니다.

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	JOB	YEARS
Davis	Sales	5
Gafney	Clerk	5
Edwards	Sales	7
Quill	Mgr	10

전체 SELECT 문에서는 ORDER BY를 마지막 절로 지정하십시오. 이 절에서 이름 지정된 컬럼은 표현식 또는 그 테이블의 어떠한 컬럼도 될 수 있습니다. ORDER BY 절에 있는 컬럼 이름은 선택 목록에서 지정할 필요가 없습니다.

ORDER BY 절에서는 ASC 또는 DESC를 명시적으로 지정하여 행을 내림차순 또는 올림차순으로 정렬할 수 있습니다. 어떤 것도 지정하지 않으면, 행은 올림차순(DESC)으로 자동적으로 정렬됩니다. 다음 명령문은 부서(DEPT) 84에 있는 직원을 고용된 연수(YEAR)에 따라 내림차순으로 표시합니다.

```

SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS DESC

```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	JOB	YEARS
Quill	Mgr	10
Edwards	Sales	7
Davis	Sales	5
Gafney	Clerk	5

문자값 및 숫자값에 따라 행을 정렬할 수 있습니다. 다음 명령문은 부서(DEPT) 84에 있는 직원을 이름(NAME)에 따라 영문자순으로 표시합니다.

```

SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY NAME

```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	JOB	YEARS
Davis	Sales	5
Edwards	Sales	7
Gafney	Clerk	5
Quill	Mgr	10

중복 행 제거

SELECT 문을 사용할 때 중복된 정보는 리턴하고 싶지 않을 때가 있습니다. 예를 들어, STAFF에는 다수의 부서 번호가 한번 이상 나열된 DEPT 컬럼 및 다수의 직함 설명이 한번 이상 나열된 JOB 컬럼이 있습니다.

중복 행을 제거하려면, SELECT 절에서 DISTINCT 옵션을 사용하십시오. 예를 들어, 다음과 같이 명령문에 DISTINCT를 삽입하면 부서 내의 각 직함을 한번만 나열합니다.

```
SELECT DISTINCT DEPT, JOB
FROM STAFF
WHERE DEPT < 30
ORDER BY DEPT, JOB
```

이 명령문은 다음과 같은 결과를 산출합니다.

DEPT	JOB
10	Mgr
15	Clerk
15	Mgr
15	Sales
20	Clerk
20	Mgr
20	Sales

DISTINCT가 SELECT 문에 지정한 컬럼 집합에서 중복 데이터를 포함하고 있는 모든 행을 제거했습니다.

연산 순서

연산의 순서를 고려하는 것이 중요합니다. 한 절의 출력값은 아래 목록에 나열된 대로 다음 절의 입력값이 됩니다. 연산 순서가 고려되어야 하는 예는 29 페이지의 『표현식 이름 지정』에 나와 있습니다.

다음 조작 순서는 반드시 DB2 코드 내에서 수행되는 방법은 아닙니다. 이 간단한 설명은 단지 조화를 좀더 잘 이해할 수 있도록 하기 위한 것입니다. 연산의 순서는 다음과 같습니다.

1. FROM 절
2. WHERE 절
3. GROUP BY 절
4. HAVING 절
5. SELECT 절
6. ORDER BY 절

표현식을 사용하여 값 계산

표현식은 명령문에 포함시킨 계산 또는 함수입니다. 다음 명령문은 부서(DEPT) 38에 있는 각 직원이 보너스를 \$500 받은 경우 각 직원의 급여가 얼마가 될지 계산합니다.

```
SELECT DEPT, NAME, SALARY + 500
FROM STAFF
WHERE DEPT = 38
ORDER BY 3
```

결과는 다음과 같습니다.

DEPT	NAME	3
38	Abrahams	12509.75
38	Naughton	13454.75
38	Quigley	17308.30
38	Marenghi	18006.75
38	O'Brien	18506.00

세번째 컬럼의 컬럼 이름이 숫자로 되어 있음을 유의하십시오. SALARY+500의 컬럼 이름을 지정하지 않았기 때문에 이것은 시스템이 생성한 번호입니다. 앞으로는 이 번호가 세번째 컬럼을 지칭하기 위해 ORDER BY 절에서 사용됩니다. 『표현식 이름 지정』에서는 표현식에 의미있는 이름을 부여하는 방법에 대해 설명합니다.

덧셈(+), 뺄셈(-), 곱셈(*) 및 나눗셈(/)과 같은 기본적인 산술 연산자를 사용하여 산술식을 만들 수 있습니다.

연산자는 여러 다른 유형의 피연산자의 값을 조작할 수 있습니다. 그것들 중 몇몇은 다음과 같습니다.

- 컬럼 이름 (RATE * HOURS 형식)
- 상수 값 (RATE * 1.07 형식)
- 스칼라 함수 (LENGTH(NAME) + 1 형식).

표현식 이름 지정

옵션인 AS 절을 이용하여 표현식에 의미있는 이름을 지정할 수 있고, 그 이름을 사용하여 나중에 표현식을 쉽게 찾을 수 있습니다. AS 절을 이용하면 선택 목록에 있는 어떠한 항목에도 이름을 부여할 수 있습니다.

다음 명령문은 급여(SALARY) + 수당(COMM)이 \$13,000보다 적은 모든 직원을 표시합니다. SALARY + COMM 표현식은 PAY로 이름 지정되어 있습니다.

```
SELECT NAME, JOB, SALARY + COMM AS PAY
      FROM STAFF
     WHERE (SALARY + COMM) < 13000
     ORDER BY PAY
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	JOB	PAY
Yamaguchi	Clerk	10581.50
Burke	Clerk	11043.50
Scoutten	Clerk	11592.80
Abrahams	Clerk	12246.25
Kermisch	Clerk	12368.60
Ngan	Clerk	12714.80

AS 절을 사용하여 ORDER BY 절에서 시스템 생성 번호보다는 특별한 컬럼 이름을 참조할 수 있습니다. 이 예에서는 PAY 이름을 사용하는 대신에 WHERE 절에서 (SALARY + COMM)을 13000과 비교하고 있습니다. 이것은 연산 순서의 결과입니다. WHERE 절은 (SALARY + COMM)이 PAY 이름에 주어지기 전에 평가되는데, SELECT 절은 WHERE 절 다음에 실행되기 때문입니다. 따라서 PAY는 술어에서 사용할 수 없습니다.

한 개 이상의 테이블에서 데이터 선택

SELECT 문을 사용하여 둘 이상의 테이블의 정보를 포함하는 보고서를 산출할 수 있습니다. 이것은 일반적으로 조인이라고 합니다. 예를 들어 STAFF와 ORG 테이블의 데이터를 조인하여 새 테이블을 만들 수 있습니다. 테이블을 조인하려면 SELECT 절에 표시하려는 컬럼 이름, FROM 절에 테이블 이름 그리고 WHERE 절에 검색 조건을 지정하십시오. WHERE 절은 생략 가능합니다.

다음은 각 관리자의 이름을 부서 명칭과 연결하는 예입니다. 직원 정보(STAFF 테이블) 및 부서 정보(ORG 테이블)는 개별적으로 저장되어 있기 때문에 두 테이블에서 정보를 선택해야 합니다. 다음 조회는 STAFF와 ORG 테이블에서 NAME과 DEPTNAME 컬럼을 각각 선택하라는 명령문입니다. 검색 조건은 MANAGER 컬럼에 있는 값이 ID 컬럼에 있는 값과 같은 행으로 선택 범위를 좁힙니다.

```
SELECT DEPTNAME, NAME
FROM ORG, STAFF
WHERE MANAGER = ID
```

31 페이지의 그림3에서는 두 개의 서로 다른 테이블내 컬럼이 어떻게 비교되는지를 보여줍니다. 상자에 있는 값은 검색 조건이 만족되었다는 것을 나타냅니다.

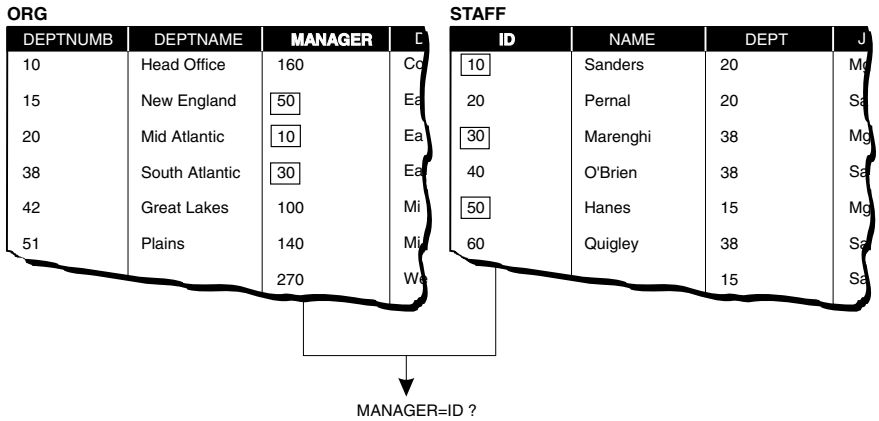


그림 3. STAFF와 ORG 테이블에서 선택

SELECT 문은 다음과 같은 결과를 산출합니다.

```

DEPTNAME      NAME
-----
Mid Atlantic   Sanders
South Atlantic Marenghi
New England   Hanes
Great Lakes   Plotz
Plains        Fraye
Head Office    Molinare
Pacific        Lea
Mountain      Quill

```

그 결과에는 각 관리자의 이름 및 부서가 나열되어 있습니다.

부속 조회 사용

SQL SELECT 문을 작성할 때, WHERE 절 안에 추가적인 SELECT 문을 둘 수 있습니다. 각각의 추가적인 SELECT는 부속 조회를 시작합니다.

다음에 부속 조회는 다른 부속 조회를 포함할 수 있고, 그 부속 조회의 값은 WHERE 절에 대체되어 들어갑니다. 그 외에도 WHERE 절은 둘 이상의 검색 조건에 있는 부속 조회를 포함할 수 있습니다. 부속 조회는 주 조회에서 사용된 테이블이나 컬럼과는 다른 테이블이나 컬럼을 참조할 수 있습니다.

다음 명령문은 STAFF 테이블에서 ID가 280인 직원의 국(DIVISION)과 위치(LOCATION)를 ORG 테이블에서 선택합니다.

```
SELECT DIVISION, LOCATION
FROM ORG
WHERE DEPTNUMB = (SELECT DEPT
                  FROM STAFF
                  WHERE ID = 280)
```

명령문을 처리할 때, DB2는 먼저 부속 조회의 결과를 판별합니다. 이 예에서 ID가 280인 직원이 부서 66에 있기 때문에 부속 조회의 결과는 66입니다. 최종 결과는 DEPTNUMB 컬럼 값이 66인 ORG 테이블로부터 얻어집니다. 최종 결과는 다음과 같습니다.

```
DIVISION  LOCATION
-----  -
Western   San Francisco
```

부속 조회를 사용할 때 데이터베이스 관리 프로그램은 부속 조회를 평가하고 결과 값을 WHERE 절에 대체합니다.

부속 조회는 46 페이지의 『상관 부속 조회』에서 더 자세히 설명됩니다.

함수 사용

이 절에서는 이 책의 예에서 사용할 함수에 대해 간략하게 설명합니다. 데이터베이스 함수는 입력 데이터 값 및 결과 값 집합 사이의 관계입니다.

함수는 내장 또는 사용자 정의 함수일 수 있습니다. DB2 Universal Database는 많은 내장된 함수 및 사전 설치된 사용자 정의 함수를 전달합니다.

내장된 함수는 SYSIBM에 있고 사전 설치된 사용자 정의 함수는 SYSFUN 스키마에 들어 있습니다. SYSIBM 및 SYSFUN은 예약된 스키마입니다.

내장 및 사전 설치된 사용자 정의 함수는 모든 사용자의 요구사항을 만족시킬 수 없습니다. 그러므로, 응용프로그램 개발자들은 해당 응용프로그램에만 적용되는 고유의 함수를 작성해야 할 수 있습니다. 사용자 정의 함수는 사용자가 정의한 비

지니스나 과학적인 함수를 포함하도록 DB2 Universal Database의 범위를 확장시켜 이를 가능하게 만듭니다. 이것은 79 페이지의 『사용자 정의 함수』에서 더 자세히 설명됩니다.

컬럼 함수

컬럼 함수는 컬럼의 값 집합에 대해 작업하여 하나의 결과 값을 얻어 냅니다. 다음이 바로 컬럼 함수의 몇 가지 예입니다. *SQL* 참조서에서 전체 목록에 대한 자세한 내용을 참조하십시오.

AVG 집합 내 값의 합계를 해당 집합 내 값의 수로 나눈 값을 리턴합니다.

COUNT 행 또는 값 집합의 행 또는 값의 수를 리턴합니다.

MAX 집합 내 값 중 가장 큰 값을 리턴합니다.

MIN 집합 내 값 중 가장 작은 값을 리턴합니다.

다음 명령문은 STAFF 테이블에서 최대 급여를 선택합니다.

```
SELECT MAX(SALARY)
FROM STAFF
```

이 명령문은 STAFF 샘플 테이블로부터 값 22959.20을 리턴합니다.

다음 예에서는 연봉이 평균 연봉보다 많지만 회사 근무 연수가 평균 연수보다 짧은 직원의 이름(NAME)과 급여(SALARY)를 선택합니다.

```
SELECT NAME, SALARY
FROM STAFF
WHERE SALARY > (SELECT AVG(SALARY) FROM STAFF)
AND YEARS < (SELECT AVG(YEARS) FROM STAFF)
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	SALARY
Marengi	17506.75
Daniels	19260.25
Gonzales	16858.20

위의 예의 WHERE 절에서, 컬럼 함수는 직접 구현되는 것과는 대조적으로 부속 조화에 나타나 있습니다(예: WHERE SALARY > AVG(SALARY)). 컬럼 함수

는 WHERE 절에 나타날 수 없습니다. 이것은 연산의 순서 때문입니다. WHERE 절은 SELECT 절이 이전에 평가된다고 생각할 수 있습니다. 결과적으로, WHERE 절이 평가되고 있으면, 컬럼 함수는 값 집합에 액세스하지 못합니다. 이 값 세트는 SELECT 절에 의해 나중에 선택됩니다.

함수가 적용되기 전에 중복된 값을 제거하기 위해 컬럼 함수의 인수로서 DISTINCT 요소를 사용할 수 있습니다. 따라서, COUNT(DISTINCT WORKDEPT)는 서로 다른 부서의 수를 계산합니다.

스칼라 함수

스칼라 함수는 하나의 값에 연산을 수행하여 또 다른 값을 리턴합니다. 다음이 바로 DB2 Universal Database에서 제공하는 스칼라 함수의 몇 가지 예입니다.

ABS 수의 절대값을 리턴합니다.

HEX 값의 16진 표현을 리턴합니다.

LENGTH 인수의 바이트 수를 리턴합니다(2바이트 문자 수를 리턴하는 그래픽 문자열의 경우).

YEAR 날짜 시간 값의 연도 부분을 가져옵니다.

SQL 참조서에서 스칼라 함수의 자세한 목록과 설명을 참조하십시오.

다음 명령문은 ORG 테이블에서 부서 이름(DEPTNAME)과 이들 이름 각각의 길이(LENGTH)를 리턴합니다.

```
SELECT DEPTNAME, LENGTH(DEPTNAME)
FROM ORG
```

이 명령문은 다음과 같은 결과를 산출합니다.

DEPTNAME	2
-----	-----
Head Office	11
New England	11
Mid Atlantic	12
South Atlantic	14
Great Lakes	11
Plains	6
Pacific	7
Mountain	8

주: AS 절이 LENGTH(DEPTNAME)에 의미있는 이름을 제공하는 데 사용되지 않았으므로 시스템에서 생성한 번호가 두번째 컬럼에 표시됩니다.

테이블 함수

테이블 함수는 간단한 CREATE TABLE 문으로부터 작성된 테이블과 유사한 테이블 컬럼을 리턴합니다.

테이블 함수는 SQL 문의 FROM 절에 사용될 수 있습니다.

DB2 Universal Database에서 현재 지원되는 유일한 테이블 함수는 SQLCACHE_SNAPSHOT입니다.

SQLCACHE_SNAPSHOT

DB2 동적 SQL 문 캐시의 스냅샷 결과를 테이블로서 리턴합니다.

그룹화

DB2 Universal Database에는 테이블의 특정 컬럼에 따라 데이터를 분석하는 기능이 있습니다.

GROUP BY 절에 정의된 그룹 구조에 따라 행을 구성할 수 있습니다. 간단한 양식으로, 그룹은 행 세트로서 각각은 "GROUP BY" 컬럼에 있는 값과 동일합니다. SELECT 절의 컬럼 이름은 그룹화 컬럼이나 컬럼 함수여야 합니다. 컬럼 함수는 GROUP BY 절에 의해 정의된 각 그룹에 대한 값을 리턴합니다. 각 그룹은 결과 집합에 한 행으로 표시됩니다. 다음 예는 각 부서 번호에 대한 최대 급여를 나열하는 결과를 작성합니다.

```
SELECT DEPT, MAX(SALARY) AS MAXIMUM
      FROM STAFF
GROUP BY DEPT
```

이 명령문은 다음과 같은 결과를 산출합니다.

```
DEPT  MAXIMUM
-----
10    22959.20
15    20659.80
20    18357.50
38    18006.00
```

```

42  18352.80
51  21150.00
66  21000.00
84  19818.00

```

MAX(SALARY)가 전체 회사가 아닌 GROUP BY 절에서 정의된 그룹인 각 부서별로 계산된다는 점에 유의하십시오.

GROUP BY 절에 WHERE 절 사용

그룹화 조회에는 그룹이 작성되고 컬럼 함수가 계산되기 전에 규정되지 않은 행을 제거하는 표준 WHERE 절이 있을 수 있습니다. WHERE 절을 GROUP BY 절 앞에 지정해야 합니다. 예를 들면, 다음과 같습니다.

```

SELECT WORKDEPT, EDLEVEL, MAX(SALARY) AS MAXIMUM
      FROM EMPLOYEE
 WHERE HIREDATE > '1979-01-01'
 GROUP BY WORKDEPT, EDLEVEL
 ORDER BY WORKDEPT, EDLEVEL

```

결과는 다음과 같습니다.

WORKDEPT	EDLEVEL	MAXIMUM
D11	17	18270.00
D21	15	27380.00
D21	16	36170.00
D21	17	28760.00
E11	12	15340.00
E21	14	26150.00

SELECT 문에 지정된 모든 컬럼 이름이 GROUP BY 절에도 있는지 살펴 보십시오. 그 두 곳 모두에 컬럼 이름이 없으면 오류가 발생하게 됩니다. GROUP BY 절은 WORKDEPT 및 EDLEVEL의 고유한 결합에 대한 행을 리턴합니다.

GROUP BY 절 다음에 HAVING 절 사용

규정 조건을 그룹에 적용하면 DB2는 그 조건을 만족시키는 그룹에만 결과를 리턴합니다. 이를 수행하려면, HAVING 절을 GROUP BY 절 다음에 포함시키십시오. HAVING 절은 AND 및 OR로 연결된 하나 이상의 술어를 포함할 수 있습니다. 각 술어는 그룹(AVG(SALARY)와 같은)의 등록 정보를 다음 둘 중 하나와 비교합니다.

- 그룹의 다른 등록 정보

예를 들면, 다음과 같습니다.

```
HAVING AVG(SALARY) > 2 * MIN(SALARY)
```

- 상수

예를 들면, 다음과 같습니다.

```
HAVING AVG(SALARY) > 20000
```

예를 들어, 다음 조회는 직원이 5명 이상인 부서에서 최대 및 최소 급여를 찾습니다.

```
SELECT WORKDEPT, MAX(SALARY) AS MAXIMUM, MIN(SALARY) AS MINIMUM
FROM EMPLOYEE
GROUP BY WORKDEPT
HAVING COUNT(*) > 4
ORDER BY WORKDEPT
```

이 명령문은 다음과 같은 결과를 산출합니다.

WORKDEPT	MAXIMUM	MINIMUM
D11	32250.00	18270.00
D21	36170.00	17250.00
E11	29750.00	15340.00

조회에서는 GROUP BY 절이 아니라 HAVING 절을 사용할 수 있습니다(흔하지는 않지만). 이 경우, DB2는 전체 테이블을 하나의 그룹으로 취급합니다. 테이블이 하나의 그룹으로 취급되기 때문에 사용자는 많아야 하나의 결과 행만을 갖게 됩니다. HAVING 조건이 전체 테이블의 내용과 일치하면, 선택된 결과(전체가 컬럼 함수로 구성되어야 합니다)가 리턴됩니다. 일치하지 않으면 어떠한 행도 리턴되지 않습니다.

제5장 표현식과 부속 조회

DB2는 다양하게 조회를 표현할 수 있습니다. 여기에서는 보다 복잡한 조회를 표현하는 데 이용할 수 있는 몇 가지 중요한 방법을 설명합니다.

여기에서는 다음에 대해 포괄적인 설명을 제시합니다.

- 스칼라 fullselect
- 데이터 유형 변환
- CASE 표현식
- 테이블 표현식
- 상관 이름

스칼라 fullselect

fullselect SQL 문에 사용될 수 있는 조회 양식입니다. 스칼라 fullselect는 단지 하나의 값만을 포함하고 있는 한 행을 리턴하는 fullselect입니다. 스칼라 fullselect를 사용하면 데이터베이스에서 표현식에 사용할 데이터 값을 쉽게 검색할 수 있습니다.

- 다음 예는 모든 직원의 평균 급여보다 많은 급여를 받는 직원의 이름을 나열합니다. 조회 내의 스칼라 fullselect는 괄호 안의 select 문입니다.

```
SELECT LASTNAME, FIRSTNAME
      FROM EMPLOYEE
 WHERE SALARY > (SELECT AVG(SALARY)
                  FROM EMPLOYEE)
```

- 다음 예는 STAFF 테이블에 있는 직원과 EMPLOYEE 테이블에 있는 직원의 평균 급여를 찾습니다.

```
SELECT AVG(SALARY) AS "Average_Employee",
       (SELECT AVG(SALARY) AS "Average_Staff" FROM STAFF)
      FROM EMPLOYEE
```

데이터 유형 변환

하나의 데이터 유형을 다른 데이터 유형으로 변환해야 할 때가 있습니다. 예를 들면, 숫자값에서 문자열로 변환해야 하는 경우입니다. 어떤 값을 다른 유형으로 변환하려면, CAST 스펙을 사용하십시오.

CAST 스펙을 사용하면 아주 긴 문자열을 잘라낼 수 있습니다. EMP_RESUME 테이블에 있는 컬럼 RESUME은 CLOB(5K)입니다. 최초의 370개의 문자만을 사용하여 지원자의 개인적 정보를 포함하는 정보를 표시하고자 할 때가 있습니다. 테이블 EMP_RESUME에서 ASCII 형식 이력서 최초의 370개의 문자를 표시하려면, 다음과 같은 조회를 발행하십시오.

```
SELECT EMPNO, CAST(RESUME AS VARCHAR(370))
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

370 문자보다 긴 문자를 갖는 값은 절단된다는 경고가 발행됩니다.

조회 조작에 더 편리한 다른 데이터 유형으로 NULL 값을 유형 변환할 수 있습니다. 43 페이지의 『공통 테이블 표현식』에 이러한 목적을 위해 유형 변환을 사용하는 예가 나와 있습니다.

CASE 표현식

SQL 문에 있는 CASE 표현식을 사용하여 테이블의 데이터 표현을 쉽게 다룰 수 있습니다. 이것을 이용하면, 프로그래밍 언어에 있는 CASE 문과 개념상 비슷한 강력한 조건 표현식을 제공할 수 있습니다.

- ORG 테이블에 있는 DEPTNAME 컬럼에서 부서 번호를 의미있는 단어로 변경하려면, 다음과 같이 조회를 입력하십시오.

```
SELECT DEPTNAME,
CASE DEPTNUMB
WHEN 10 THEN 'Marketing'
WHEN 15 THEN 'Research'
WHEN 20 THEN 'Development'
WHEN 38 THEN 'Accounting'
ELSE 'Sales'
END AS FUNCTION
FROM ORG
```


결과는 다음과 같습니다.

DEPTNAME	FUNCTION
Head Office	Marketing
New England	Research
Mid Atlantic	Development
South Atlantic	Accounting
Great Lakes	Sales
Plains	Sales
Pacific	Sales
Mountain	Sales

- CASE 표현식을 사용하면, 0으로 나누는 경우와 같은 예외로부터 보호할 수 있습니다. 다음 예에서는 보너스 또는 수당이 없는직원의 경우, 명령문 조건이 나눗셈 연산을 피함으로써 오류를 방지합니다.

```
SELECT LASTNAME, WORKDEPT FROM EMPLOYEE
WHERE (CASE
      WHEN BONUS+COMM=0 THEN NULL
      ELSE SALARY/(BONUS+COMM)
      END ) > 10
```

- CASE 표현식을 사용하면 컬럼의 모든 값의 합계에 대한 한 컬럼에 있는 값의 부속 집합의 합계 비율을 계산할 수 있습니다. 이 비율은 CASE 표현식을 사용하는 한 명령문에 포함될 수 있습니다. 이 표현식에는 단 한 번의 데이터 패스만이 필요합니다. CASE 표현식이 없으면, 같은 계산을 수행하는 데 적어도 두 개의 패스가 필요합니다.

다음 예는 모든 급여 총계에 대한 부서(DEPT) 20의 급여 합계 비율을 CASE 표현식을 사용하여 계산합니다.

```
SELECT CAST(CAST (SUM(CASE
      WHEN DEPT = 20 THEN SALARY
      ELSE 0
      END) AS DECIMAL(7,2))/
      SUM(SALARY) AS DECIMAL (3,2))
FROM STAFF
```

결과값은 0.11입니다. 결과값의 정밀도가 CAST 함수와 일치함을 유의하십시오.

- CASE 표현식을 사용하면, 추가적인 오버헤드가 필요한 함수 자체를 호출하지 않고 단순 함수를 평가할 수 있습니다. 예를 들면, 다음과 같습니다.

```

CASE
WHEN X<0 THEN -1
WHEN X=0 THEN 0
WHEN X>0 THEN 1
END

```

이 표현식은 SYSFUN 스키마에 있는 SIGN 사용자 정의 함수와 같은 결과를 가집니다.

테이블 표현식

단일 조회에 대한 뷰 정의가 필요하다면 테이블 표현식을 사용할 수 있습니다.

테이블 표현식은 일시적이며 SQL 문이 수행되는 동안에만 유효합니다. 테이블 표현식은 뷰처럼 공유할 수는 없으나 뷰보다는 좀더 유연성이 있습니다.

이 절에서는 조회에서 공통 테이블 표현식과 중첩 테이블 표현식을 사용하는 방법을 설명합니다.

중첩 테이블 표현식

중첩 테이블 표현식은 임시적 뷰입니다. 중첩 테이블 표현식에서는 정의가 주 조회의 FROM 절에 중첩되어 있습니다(직접 정의된 상태).

다음은 중첩 테이블 표현식을 사용하여 교육 수준(EDLEVEL)이 16보다 큰 직원의 평균 급여, 교육 수준 그리고 입사년도를 찾으라는 조회입니다.

```

SELECT EDLEVEL, HIREYEAR, DECIMAL(AVG(TOTAL_PAY),7,2)
FROM (SELECT EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
SALARY+BONUS+COMM AS TOTAL_PAY
FROM EMPLOYEE
WHERE EDLEVEL > 16) AS PAY_LEVEL
GROUP BY EDLEVEL, HIREYEAR
ORDER BY EDLEVEL, HIREYEAR

```

결과는 다음과 같습니다.

EDLEVEL	HIREYEAR	3
17	1967	28850.00
17	1973	23547.00
17	1977	24430.00
17	1979	25896.50

18	1965	57970.00
18	1968	32827.00
18	1973	45350.00
18	1976	31294.00
19	1958	51120.00
20	1975	42110.00

이 조회는 중첩 테이블 표현식을 사용하여 HIREDATE 컬럼에서 입사년도를 끌어내어 GROUP BY 절에 사용할 수 있습니다. EDLEVEL의 다른 값을 사용하여 비슷한 조회를 수행하려 한다면, 이것을 뷰로 작성하지 않으려 할 수도 있습니다.

이 예에서는 스칼라 내장 함수 DECIMAL을 사용하고 있습니다. DECIMAL은 숫자 또는 문자열을 소수로 나타낸 것을 리턴합니다. SQL 참조서에서 함수에 관한 자세한 내용을 참조하십시오.

공통 테이블 표현식

공통 테이블 표현식은 복잡한 조회에 사용하기 위해 작성하는 테이블 표현식입니다. WITH 절을 사용하여 조회를 시작할 때 공통 테이블 표현식을 정의하고 이름 지정하십시오. 공통 테이블 표현식을 반복해서 참조하면 같은 결과 집합을 사용합니다. 반면에, 중첩 테이블 표현식 또는 뷰를 사용하면, 결과 집합이 매번 다시 생성되어 다른 결과를 가질 수 있습니다.

다음 예는 교육 수준이 16보다 크지만 같은 교육 수준을 가진 같은 해에 입사한 직원들보다 평균 급여가 적은 직원 목록을 표시합니다. 조회의 부분에 대해서는 조회 다음에 상세하게 설명합니다.

1

WITH

PAYLEVEL AS

(SELECT EMPNO, EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,

SALARY+BONUS+COMM AS TOTAL_PAY

FROM EMPLOYEE

WHERE EDLEVEL > 16),

2

```
PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS
(SELECT EDLEVEL, HIREYEAR, AVG(TOTAL_PAY)
FROM PAYLEVEL
GROUP BY EDLEVEL, HIREYEAR)
```

3

```
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, DECIMAL(AVG_TOTAL_PAY,7,2)
FROM PAYLEVEL, PAYBYED
WHERE EDLEVEL = EDUC_LEVEL
AND HIREYEAR = YEAR_OF_HIRE
AND TOTAL_PAY < AVG_TOTAL_PAY
```

1

이것은 이름이 PAYLEVEL인 공통 테이블 표현식입니다. 결과 테이블에는 직원 번호, 입사년도, 총급여 그리고 교육 수준이 포함되어 있습니다. 교육 수준이 16보다 큰 직원의 행만이 포함됩니다.

2

이것은 이름이 PAYBYED(또는 PAY BY Education)인 공통 테이블 표현식입니다. 이전의 공통 테이블 표현식에서 작성된 PAYLEVEL 테이블을 사용하여 같은 연도에 입사한 직원의 교육 수준, 입사년도 그리고 각 교육 수준 내에서의 평균 급여를 판별할 수 있습니다. 이 테이블이 리턴한 컬럼에는 선택 목록에서 사용되는 컬럼 이름과는 다른 이름(예를 들어, EDUC_LEVEL)이 부여됩니다. 이것은 중첩 테이블 표현식 예에서 산출된 결과와 같은 PAYBYED라는 이름의 결과 집합을 산출합니다.

3

이것은 원하는 결과를 산출하는 실제 조회입니다. 두 테이블(PAYLEVEL, PAYBYED)을 조인하여 같은 연도에 입사한 사람들의 평균 급여보다 작은 급여를 받는 직원들을 판별합니다. PAYBYED가 PAYLEVEL에 기초하고 있음을 유의하십시오. 따라서 완전한 명령문에서 PAYLEVEL에 두 번 효과적으로 액세스합니다. 조회를 평가할 때는 두 번 다 같은 행 집합을 사용합니다.

최종 결과는 다음과 같습니다.

EMPNO	EDLEVEL	YEAR_OF_HIRE	TOTAL_PAY	5
000210	17	1979	20132.00	25896.50

상관 이름

상관 이름은 한 오브젝트를 여러 사용자가 사용할 때 구별하기 위한 식별자입니다. 상관 이름은 조회의 FROM 절, UPDATE 또는 DELETE 문의 처음 절에서 정의할 수 있습니다. 상관 이름은 테이블, 뷰 또는 중첩 테이블 표현식과 연관될 수 있지만 정의된 문맥 내에서만 연관됩니다.

다음은 절 FROM STAFF S 및 ORG O에서 S 및 O를 STAFF 및 ORG의 상관 이름으로 각각 설정하라는 예입니다.

```
SELECT NAME, DEPTNAME
FROM STAFF S, ORG O
WHERE O.MANAGER = S.ID
```

일단 상관 이름을 정의하면, 오브젝트를 규정하기 위해 상관 이름만을 사용할 수 있습니다. 예를 들어, 위의 예에서 ORG.MANAGER=STAFF.ID라고 기술했다면 그 명령문은 실패했을 것입니다.

또한 상관 이름을 데이터베이스 오브젝트를 가리키는 단축 이름으로 사용할 수도 있습니다. 예를 들어, STAFF는 S로 정의할 수 있습니다.

상관 이름을 사용하면 오브젝트도 복제할 수 있습니다. 이것은 테이블 항목과 테이블 자체를 비교할 때 유용하게 쓸 수 있습니다. 다음 예는 테이블 EMPLOYEE를 그 자신의 다른 인스턴스와 비교하여 모든 직원의 관리자를 찾는 내용입니다. 여기에는 디자이너가 아닌 직원의 이름, 그 관리자의 이름 및 부서 번호가 표시됩니다.

```
SELECT E2.FIRSTNAME, E2.LASTNAME, E2.JOB, E1.FIRSTNAME AS MGR_FIRSTNAME,
       E1.LASTNAME AS MGR_LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1, EMPLOYEE E2
WHERE E1.WORKDEPT = E2.WORKDEPT
      AND E1.JOB = 'MANAGER'
      AND E2.JOB <> 'MANAGER'
      AND E2.JOB <> 'DESIGNER'
```

이 명령문은 다음과 같은 결과를 산출합니다.

FIRSTNAME	LASTNAME	JOB	MGR_FIRSTNAME	MGR_LASTNAME	WORKDEPT
DOLORES	QUINTANA	ANALYST	SALLY	KWAN	C01
HEATHER	NICHOLLS	ANALYST	SALLY	KWAN	C01
JAMES	JEFFERSON	CLERK	EVA	PULASKI	D21
SALVATORE	MARINO	CLERK	EVA	PULASKI	D21
DANIEL	SMITH	CLERK	EVA	PULASKI	D21

SYBIL	JOHNSON	CLERK	EVA	PULASKI	D21
MARIA	PEREZ	CLERK	EVA	PULASKI	D21
ETHEL	SCHNEIDER	OPERATOR	EILEEN	HENDERSON	E11
JOHN	PARKER	OPERATOR	EILEEN	HENDERSON	E11
PHILIP	SMITH	OPERATOR	EILEEN	HENDERSON	E11
MAUDE	SETRIGHT	OPERATOR	EILEEN	HENDERSON	E11
RAMLAL	MEHTA	FIELDREP	THEODORE	SPENSER	E21
WING	LEE	FIELDREP	THEODORE	SPENSER	E21
JASON	GOUNOT	FIELDREP	THEODORE	SPENSER	E21

상관 부속 조회

앞에서 언급한 테이블 중 어느 하나에 참조가 허용된 부속 조회를 상관 부속 조회라고 합니다. 또한 부속 조회는 주 조회에 있는 테이블의 상관 참조를 가지고 있다고도 합니다.

다음 예는 급여가 부서의 평균 급여보다 많은 'A00' 부서 내의 직원의 이름 및 직번을 나열하는 상관없는 부속 조회입니다.

```

SELECT EMPNO, LASTNAME
      FROM EMPLOYEE
 WHERE WORKDEPT = 'A00'
    AND SALARY > (SELECT AVG(SALARY)
      FROM EMPLOYEE
      WHERE WORKDEPT = 'A00')
```

이 명령문은 다음과 같은 결과를 산출합니다.

```

EMPNO  LASTNAME
-----
000010 HAAS
000110 LUCCHESI
```

모든 부서의 평균 급여에 대해 알려면, 모든 부서에 대해 부속 조회를 한번은 평가해야 합니다. 평가를 수행하려면 상관 기능을 사용하면 됩니다. 상관 기능을 사용하면 일단 외부 기능에서 식별된 테이블의 각 행에 대해 반복적으로 실행되는 부속 조회를 작성할 수 있습니다.

다음 예는 급여가 부서의 평균 급여보다 높은 모든 직원을 나열하기 위해 상관 부속 조회를 사용합니다.

```

SELECT E1.EMPNO, E1.LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1
WHERE SALARY > (SELECT AVG(SALARY)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
ORDER BY E1.WORKDEPT

```

이 조회에서는 모든 부서의 부속 조회를 한번 평가합니다. 결과는 다음과 같습니다.

EMPNO	LASTNAME	WORKDEPT
000010	HAAS	A00
000110	LUCCHESI	A00
000030	KWAN	C01
000060	STERN	D11
000150	ADAMSON	D11
000170	YOSHIMURA	D11
000200	BROWN	D11
000220	LUTZ	D11
000070	PULASKI	D21
000240	MARINO	D21
000270	PEREZ	D21
000090	HENDERSON	E11
000280	SCHNEIDER	E11
000100	SPENSER	E21
000330	LEE	E21
000340	GOUNOT	E21

상관 부속 조회로 조회를 작성하려면, 부속 조회가 있는 보통의 외부 조회문과 같은 기본 형식을 사용하십시오. 그러나, 외부 조회의 FROM 절에서는, 테이블 이름 다음에 상관 이름을 두십시오. 그러면 부속 조회에 상관 이름으로 규정된 컬럼 참조가 포함되게 됩니다. 예를 들어, E1이 상관 이름이면, E1.WORKDEPT는 외부 조회에 있는 테이블의 현재 행의 WORKDEPT 값을 의미합니다. 외부 조회에서는 테이블의 각 행의 부속 조회를 (개념적으로) 재평가합니다.

상관 부속 조회를 사용하면, 시스템이 사용자를 대신하여 그 작업을 하여 사용자의 응용프로그램 내에서 작성할 필요가 있는 코드의 양을 줄여줍니다.

DB2는 규정되지 않은 상관 참조를 허용합니다. 예를 들어, EMPLOYEE 테이블은 LASTNAME이라고 컬럼을 갖고 있으며, SALES 테이블은 SALES_PERSON이라는 테이블은 갖고 있으나 LASTNAME이라는 컬럼은 갖고 있지 않습니다.

```

SELECT LASTNAME, FIRSTNAME, COMM
FROM EMPLOYEE
WHERE 3 > (SELECT AVG(SALES)
FROM SALES
WHERE LASTNAME = SALES_PERSON)

```

위 예에서, 시스템은 LASTNAME 컬럼의 가장 안쪽의 FROM 절을 검사합니다. 시스템이 그 절을 찾지 못하면 그 다음으로 안쪽에 있는 FROM 절을 검사합니다 (이 경우에 그 다음 절은 외부 FROM 절입니다.). 반드시 필요한 것은 아니지만, 상관 참조를 규정하면, 조회의 읽기 능력을 향상시키고 사용자가 원하는 결과를 확실히 얻을 수 있습니다.

상관 부속 조회 구현

상관 부속 조회를 사용하려 할 때는 언제입니까? 컬럼 함수를 사용하면 때때로 유용합니다.

교육 수준이 그들 부서의 평균보다 더 높은 직원의 목록을 나열하고 싶다고 가정합니다.

먼저, 선택 목록 항목을 결정해야 합니다. 『직원을 나열하십시오』라는 문제가 있습니다. 이는 EMPLOYEE 테이블에 LASTNAME만 있으면 충분히 직원을 고유하게 식별할 수 있음을 의미합니다. 문제에는 또한 조건으로서 교육 수준(EDLEVEL) 및 직원의 부서(WORKDEPT)도 기술되어 있습니다. 문제는 명시적으로 컬럼이 표시되어야 한다고 요구하고 있지는 않지만, 선택 목록에 컬럼을 포함하면 솔루션을 설명하는데 도움이 됩니다. 조회의 한 부분은 이제 다음과 같이 구성할 수 있습니다.

```

SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE

```

다음에는, 검색 조건(WHERE 절)이 필요합니다. 문제의 명령문에는 『...의 교육 수준은 그 직원 부서의 평균보다 높습니다』라고 기술되어 있습니다. 이것은 그 테이블에 있는 모든 직원에 대해서 그 직원 부서의 평균 교육 수준이 계산되어야 한다는 것을 의미합니다. 이 명령문은 상관 부속 조회의 설명과 부합합니다. 몇 가지 알려져 있지 않은 등록 정보(현재 직원 부서의 평균 교육 수준)가 각 행에 대해 계산됩니다. EMPLOYEE 테이블에는 상관 이름이 필요합니다.


```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
```

필요한 부속 조회는 단순합니다. 부속 조회는 각 부서의 평균 교육 수준을 계산합니다. 완성된 SQL 문은 다음과 같습니다.

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
WHERE EDLEVEL > (SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
```

결과는 다음과 같습니다.

LASTNAME	WORKDEPT	EDLEVEL
HAAS	A00	18
KWAN	C01	20
PULASKI	D21	16
HENDERSON	E11	16
LUCCHESSI	A00	19
PIANKA	D11	17
SCOUTTEN	D11	17
JONES	D11	17
LUTZ	D11	18
MARINO	D21	17
JOHNSON	D21	16
SCHNEIDER	E11	17
MEHTA	E21	16
GOUNOT	E21	16

직원 부서의 번호가 아니라 부서 이름을 나열한다고 가정해 봅시다. 필요한 정보 (DEPTNAME)는 개별적인 테이블(DEPARTMENT)에 있습니다. 상관 변수를 정의하는 외부 레벨 조회는 조인 조회가 될 수도 있습니다(30 페이지의 『한 개 이상의 테이블에서 데이터 선택』에 자세한 내용이 있습니다).

외부 레벨 조회에서 조인을 사용할 때, FROM 절에 조인될 테이블을 나열하고 해당 테이블 이름 다음에 상관 이름을 두십시오.

부서의 번호 대신에 부서 이름을 나열하기 위해 조회를 수정하려면, 선택 목록에서 WORKDEPT를 DEPTNAME으로 대체하십시오. FROM 절은 또한 DEPARTMENT 테이블을 포함해야 하고 WHERE 절은 적절한 조인 조건을 표현해야 합니다.

다음은 수정된 조회입니다.

```
SELECT LASTNAME, DEPTNAME, EDLEVEL
FROM EMPLOYEE E1, DEPARTMENT
WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
AND EDLEVEL > (SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
```

이 명령문은 다음과 같은 결과를 산출합니다.

LASTNAME	DEPTNAME	EDLEVEL
HAAS	SPIFFY COMPUTER SERVICE DIV.	18
LUCCHESI	SPIFFY COMPUTER SERVICE DIV.	19
KWAN	INFORMATION CENTER	20
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17
MEHTA	SOFTWARE SUPPORT	16
GOUNOT	SOFTWARE SUPPORT	16

위의 예는 부속 조회에서 사용된 상관 이름이 상관 부속 조회를 포함하는 조회의 FROM 절에 정의되어야 한다는 것을 보여줍니다. 그러나, 이러한 포함에는 몇 개의 중첩 레벨을 포함할 수 있습니다.

어떤 부서가 단지 몇명의 직원만이 있어서 그들의 평균 교육 수준이 잘못될 수도 있다고 가정합니다. 직원을 비교하기 위해 평균 교육 수준이 의미있는 수치가 되게 하려면, 최소한 한 부서에 5명의 직원이 있어야 합니다. 그래서 교육 수준이 그 직원이 속한 부서의 평균보다 높은 직원을 나열하고 최소한 5명의 직원을 가진 부서만을 고려 대상으로 해야 합니다.

문제는 또 다른 부속 조회를 암시하는데, 이는 외부 레벨 조회에 있는 각 직원에 대해 그 직원이 속한 부서의 총 직원 수를 계산해야 하기 때문입니다.

```
SELECT COUNT(*)
FROM EMPLOYEE E3
WHERE E3.WORKDEPT = E1.WORKDEPT
```

다음은 계수가 5보다 크거나 같을 경우에 평균을 계산하는 명령문입니다.

```
SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT
AND 5 <= (SELECT COUNT(*)
          FROM EMPLOYEE E3
          WHERE E3.WORKDEPT = E1.WORKDEPT)
```

마지막으로, 교육 레벨이 해당 부서의 평균보다 큰 직원만이 포함되어야 합니다. 다음은 이에 대한 명령문입니다.

```
SELECT LASTNAME, DEPTNAME, EDLEVEL
FROM EMPLOYEE E1, DEPARTMENT
WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
AND EDLEVEL >
(SELECT AVG(EDLEVEL)
 FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <=
 (SELECT COUNT(*)
  FROM EMPLOYEE E3
  WHERE E3.WORKDEPT = E1.WORKDEPT))
```

이 명령문은 다음과 같은 결과를 산출합니다.

LASTNAME	DEPTNAME	EDLEVEL
PIANKA	MANUFACTURING SYSTEMS	17
SCOTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17

제6장 조회에서 연산자와 술어 사용

DB2 Universal Database에서는 조회와 여러 집합 연산자를 결합하여 규정화된 술어가 있는 복합 조건문을 구성할 수 있습니다.

여기에서는 다음과 같은 방법을 설명합니다.

- UNION, EXCEPT 그리고 INTERSECT 집합 연산자와 여러 테이블을 결합하는 방법
- 규정화된 술어가 있는 조회문의 복합 조건을 구성하는 방법. 기본적인 술어는 2 2 페이지의 『행 선택』에서 간략히 설명되었습니다.

집합 연산자로 조회 결합

UNION, EXCEPT 그리고 INTERSECT 집합 연산자를 이용하면, 두 개 이상의 외부 레벨 조회를 단일 조회에 결합해 넣을 수 있습니다. 이러한 집합 연산자에 의해 연결된 각 조회가 실행되면 각각의 결과가 결합됩니다. 각 연산자는 다른 결과를 만듭니다.

UNION 연산자

UNION 연산자는 두 개의 다른 결과 테이블(예를 들어, TABLE1 및 TABLE2)을 결합하고 그 테이블에 있는 모든 중복 행을 제거하여 결과 테이블을 끌어냅니다. ALL을 UNION과 함께 사용할 때(즉, UNION ALL), 중복 행은 제거되지 않습니다. 어떤 경우라도, 끌어낸 테이블의 각 행은 TABLE1 또는 TABLE2에서 끌어낸 행입니다.

UNION 연산자의 다음 예에서 조회는 \$21,000 이상의 급여를 받거나 경영 책임이 있고 8년 미만 근무한 모든 직원의 이름을 리턴합니다.

1

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000  
UNION
```

2

```
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8  
ORDER BY ID
```

각 조회의 결과는 다음과 같습니다.

1

```
ID      NAME  
-----  
140 Fraye  
160 Molinare  
260 Jones
```

2

```
ID      NAME  
-----  
10 Sanders  
30 Marengchi  
100 Plotz  
140 Fraye  
160 Molinare  
240 Daniels
```

데이터베이스 관리 프로그램은 두 조회의 결과를 결합해서 중복된 것을 제거하고 최종 결과를 오름차순으로 리턴합니다.

```
ID      NAME  
-----  
10 Sanders  
30 Marengchi  
100 Plotz  
140 Fraye  
160 Molinare  
240 Daniels  
260 Jones
```

집합 연산자가 있는 조회에서 ORDER BY 절을 사용하려면, 마지막 조회문 다음에 써야 합니다. 시스템은 결합된 응답 집합을 순서대로 나열합니다.

두 테이블에 있는 컬럼 이름이 다르면, 결합된 결과 테이블은 그에 대응하는 컬럼의 이름을 가질 수 없습니다. 대신에 그 컬럼들은 나타나는 순서대로 번호가 매겨 집니다. 그래서, 결과 테이블을 순서대로 놓고 싶으면, ORDER BY 절에서 컬럼 이름을 지정해야 합니다.

EXCEPT 연산자

EXCEPT 연산자는 TABLE2가 아니라 TABLE1에 있는 모든 행을 포함하고 모든 중복 행을 제거함으로써 결과 테이블을 끌어냅니다. EXCEPT와 ALL을 같이 사용하면(EXCEPT ALL), 중복 행은 제거되지 않습니다.

EXCEPT 연산자에 대한 다음 예에서, 조회는 \$21,000 이상의 급여를 받으나 관리자가 아니고 8년 이상 근무한 모든 직원의 이름을 리턴합니다.

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
EXCEPT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

각 조회의 결과는 UNION에 관한 절에 나열되어 있습니다. 위의 명령문은 다음과 같은 결과를 산출합니다.

```
ID      NAME
-----
260 Jones
```

INTERSECT 연산자

INTERSECT 연산자는 TABLE1과 TABLE2 양쪽 모두에 있는 행만을 포함하고 중복된 모든 행을 제거하여 결과 테이블을 끌어냅니다. INTERSECT와 ALL을 같이 사용하면(INTERSECT ALL) 중복된 행은 제거되지 않습니다.

INTERSECT 연산자에 대한 다음 예에서, 조회는 \$21,000 이상의 급여를 받으며 관리자이고 8년 이상 근무한 직원의 ID와 이름을 리턴합니다.

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
INTERSECT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

각 조회의 결과는 UNION에 관한 절에 나열되어 있습니다. INTERSECT를 사용한 두 조회의 결과는 다음과 같습니다.

ID	NAME
140	Fraye
160	Molinare

UNION, EXCEPT 및 INTERSECT 연산자를 사용할 때는 다음을 명심하십시오.

- 조회의 선택항목에 있는 모든 해당 항목은 연산자에 대해서 호환되어야 합니다. 더 많은 정보를 원하면 *SQL* 참조서에 있는 데이터 유형 호환 표를 보십시오.
- ORDER BY 절을 사용하려면, 집합 연산자가 있는 마지막 조회 다음에 두어야 합니다. 컬럼 이름이 모든 연산자에 대한 조회의 선택 목록에 있는 해당 항목과 동일한 경우, ORDER BY 절에 컬럼 이름을 사용할 수 있습니다.
- 같은 데이터 유형 및 같은 길이 사이의 연산은 해당 유형 및 길이를 가진 컬럼을 산출합니다. UNION, EXCEPT 그리고 INTERSECT 집합 연산자의 결과에 대해서는 *SQL* 참조서에 있는 결과 데이터 유형의 규칙을 보십시오.

술어

술어를 이용하면 이러한 조건을 충족시키는 행만을 처리할 수 있도록 조건을 구성할 수 있습니다. 기본 술어는 22 페이지의 『행 선택』에 설명되어 있습니다. 이 절에서는 IN, BETWEEN, LIKE, EXISTS 그리고 규정화된 술어를 설명합니다.

IN 술어 사용

어떤 값을 몇 가지 다른 값과 비교하려면 IN 술어를 사용하십시오. 예를 들면, 다음과 같습니다.

```
SELECT NAME
      FROM STAFF
 WHERE DEPT IN (20, 15)
```

이 명령문은 다음 명령문과 같은 의미입니다.

```
SELECT NAME
      FROM STAFF
 WHERE DEPT = 20 OR DEPT = 15
```

부속 조회를 했을 때 일련의 값을 리턴하면 IN 및 NOT IN 연산자를 사용할 수 있습니다. 예를 들어, 다음 조회는 프로젝트 MA2100 및 OP2012를 담당하는 직원의 성을 나열합니다.


```

SELECT LASTNAME
      FROM EMPLOYEE
WHERE EMPNO IN
      (SELECT RESPEMP
       FROM PROJECT
        WHERE PROJNO = 'MA2100'
         OR PROJNO = 'OP2012')

```

조회는 한 번 평가되고 그 결과 목록은 외부 레벨 조회로 대체되어 들어갑니다. 예를 들어, 위의 부속 조회는 10에서 330 사이의 직원 번호를 선택하고 외부 레벨 조회는 자신의 WHERE 절이 아래와 같은 것처럼 평가됩니다.

```

WHERE EMPNO IN (10, 330)

```

부속 조회가 리턴한 값의 목록은 0개, 1개 또는 그 이상의 값을 포함할 수 있습니다.

BETWEEN 술어 사용

BETWEEN 술어는 하나의 값을 (BETWEEN 술어에 이름이 지정된) 값의 범위와 비교합니다.

다음 예는 급여가 \$10,000 및 \$20,000 사이인 직원의 이름을 찾습니다.

```

SELECT LASTNAME
      FROM EMPLOYEE
WHERE SALARY BETWEEN 10000 AND 20000

```

이것은 다음과 같은 의미입니다.

```

SELECT LASTNAME
      FROM EMPLOYEE
WHERE SALARY >= 10000 AND SALARY <= 20000

```

다음 예는 급여가 \$10,000 미만 또는 \$20,000 이상인 직원의 이름을 찾습니다.

```

SELECT LASTNAME
      FROM EMPLOYEE
WHERE SALARY NOT BETWEEN 10000 AND 20000

```

LIKE 술어 사용

특정 패턴을 가진 문자열을 검색하려면 LIKE 술어를 사용하십시오. 이 패턴은 비유 기호 및 밑줄로 지정됩니다.

- 밑줄 문자(_)는 하나의 문자를 나타냅니다.
- 비율 기호(%)는 0개 또는 그 이상의 문자를 가진 문자열을 나타냅니다.
- 그 밖의 문자는 그 자체를 나타냅니다.

다음 예는 문자 'S'로 시작되는 길이가 7자인 직원의 이름을 선택합니다.

```
SELECT NAME
  FROM STAFF
 WHERE NAME LIKE 'S _ _ _ _ _ _ _'
```

다음은 'S'로 시작하지 않는 직원의 이름을 선택하는 명령문입니다.

```
SELECT NAME
  FROM STAFF
 WHERE NAME NOT LIKE 'S%'
```

EXISTS 술어 사용

부속 조회를 사용하면, 어떤 조건을 만족시키는 행의 존재 여부를 테스트할 수 있습니다. 이 경우, 부속 조회는 EXISTS 또는 NOT EXISTS 술어를 통해 외부 레벨 조회에 링크됩니다.

EXISTS 술어를 사용하여 부속 조회를 외부 조회에 링크하면, 부속 조회는 값을 리턴하지 않습니다. 정확히 말하면, EXISTS 술어는 부속 조회의 응답 집합이 하나 이상의 행을 포함하면 참이 되고, 행을 포함하지 않으면 거짓이 됩니다.

EXISTS 술어는 종종 상관 부속 조회와 함께 사용됩니다. 다음 예는 현재 PROJECT 테이블에 항목이 없는 부서를 나열합니다.

```
SELECT DEPTNO, DEPTNAME
  FROM DEPARTMENT X
 WHERE NOT EXISTS
       (SELECT *
        FROM PROJECT
         WHERE DEPTNO = X.DEPTNO)
 ORDER BY DEPTNO
```

외부 레벨 조회의 WHERE 절에서 AND 및 OR을 사용하면, EXISTS 및 NOT EXISTS 술어를 다른 술어에 연결할 수 있습니다.

규정화된 술어

규정화된 술어는 한 값의 집합과 비교합니다. `fullselect`가 하나 이상의 값을 리턴하면, 접미사 `ALL`, `ANY` 그리고 `SOME`를 붙여서 술어에 있는 비교 연산자를 수정해야 합니다. 이 접미사는 리턴된 값의 집합이 외부 레벨 술어에서 처리되는 방법을 결정합니다. 비교 연산자 `>`는 예로서 사용됩니다(아래의 비교는 다른 연산자에도 적용됩니다).

표현식 > **ALL (fullselect)**

표현식이 `fullselect`가 리턴한 각각의 값보다 클 때 술어는 참입니다. `fullselect`가 아무 값도 리턴하지 않으면 그 술어는 참입니다. 지정된 관계가 최소한 한 값에 대해 거짓일 때 그 결과는 거짓입니다. `<>ALL` 규정화된 술어가 `NOT IN` 술어와 같음을 유의하십시오.

다음 예는 부속 조회와 `> ALL` 비교를 사용하여 모든 관리자보다 많은 급여를 받는 모든 직원의 이름과 직업을 찾습니다.

```
SELECT LASTNAME, JOB
FROM EMPLOYEE
WHERE SALARY > ALL
(SELECT SALARY
FROM EMPLOYEE
WHERE JOB='MANAGER')
```

표현식 > **ANY (fullselect)**

표현식이 `fullselect`가 리턴한 값 중 최소한 하나보다 크면 그 술어는 참입니다. `fullselect`가 아무 값도 리턴하지 않으면 그 술어는 거짓입니다. `=ANY` 규정화된 연산자가 `IN` 술어와 같음을 유의하십시오.

표현식 > **SOME (fullselect)**

`SOME`은 `ANY`와 동의어입니다.

술어와 연산자에 대해 더 많은 정보를 원하면, *SQL 참조서*에서 자세한 내용을 참조하십시오.

제7장 고급 SQL

여기에서는 조화를 사용자의 요구에 맞게, 보다 효과적으로 설계할 수 있도록 해 주는 DB2 Universal Database의 몇몇 특징을 설명합니다. 여기에서 논의되는 내용을 이해하려면 먼저 지금까지 설명한 내용을 잘 이해해야 합니다.

여기에는 다음과 같은 내용이 들어 있습니다.

- 제한조건 및 트리거가 있는 비즈니스 규칙 시행
- 조인
- ROLLUP 및 CUBE 조회 및 순환 조회
- OLAP 함수

제한조건 및 트리거가 있는 비즈니스 규칙 시행

비즈니스상에서는 특정 규칙이 항상 시행되도록 해야 합니다. 예를 들어, 프로젝트에 참여하는 직원은 급여 명부에 기재되어 있어야 합니다. 아니면, 특정 이벤트가 자동적으로 발생하기를 원할 때도 있습니다. 예를 들어, 영업 사원이 판매를 하면 수당이 증가해야 합니다.

DB2 Universal Database는 이러한 목적에 유용한 방법을 제공합니다.

- **고유한 제한조건**은 테이블에 있는 하나 이상의 컬럼에서 값이 중복되는 것을 방지합니다.
- **참조 무결성 제한조건**은 지정된 테이블에서 데이터의 일관성을 유지하도록 해 줍니다.
- **테이블 점검 제한조건**은 컬럼이 가질 수 있는 값에 대한 제한사항을 규정하는 규칙입니다. 컬럼에 지정된 값이 그 컬럼에 대한 점검 제한조건을 만족하지 못하면 삽입 및 갱신은 실패합니다.
- **트리거**는 지정된 테이블에 대한 삭제, 삽입 또는 갱신 조작에 의해 실행 또는 트리거된 조치 세트를 정의합니다. 다른 테이블에 기록하거나 입력 값을 수정하고 경고 메시지를 발행하기 위해서는 트리거를 사용할 수 없습니다.

첫번째 절에서는 키에 대한 개념을 설명합니다. 참조 무결성, 제한조건 및 트리거는 나중에 예 및 도표를 통해 설명합니다.

키

키는 특별한 행들을 식별하거나 액세스하기 위해 사용할 수 있는 일련의 컬럼입니다.

하나 이상의 컬럼으로 구성된 키를 복합 키라고 합니다. 복합 키가 있는 테이블에서, 복합 키에서의 컬럼 순서는 테이블에서의 순서에 제한을 받지 않습니다.

고유 키

고유 키는 두 값이 동일하지 않은 컬럼(또는 컬럼 세트)으로 정의됩니다. 고유 키의 컬럼은 널(Null) 값을 포함할 수 없습니다. INSERT나 UPDATE 문을 수행하는 동안 데이터베이스 관리 프로그램에 의해 이 제한조건이 적용됩니다. 테이블은 여러 개의 고유한 키를 가질 수 있습니다. 고유 키는 생략 가능하고 CREATE TABLE 또는 ALTER TABLE 문에서 정의할 수 있습니다.

기본 키

기본 키는 테이블 정의의 한 부분인 고유한 키입니다. 테이블은 하나 이상의 기본 키를 가질 수 없으며, 기본 키의 컬럼에는 널(Null) 값을 포함할 수 없습니다. 기본 키는 생략 가능하고 CREATE TABLE 또는 ALTER TABLE 문에서 정의할 수 있습니다.

외부 키

외부 키는 참조 제한조건에서 지정할 수 있습니다. 테이블은 외부 키가 없을 수도 있고 여러 개를 가질 수도 있습니다. 복합 외부 키값은 그 값의 구성요소 중 어떤 것이 널(Null)인 경우 널이 됩니다. 외부 키는 생략 가능하고 CREATE TABLE 또는 ALTER TABLE 문에서 정의할 수 있습니다.

고유한 제한조건

고유한 제한조건은 키 값이 테이블 내에서 고유한 값을 갖도록 해줍니다. 고유한 제한조건은 생략 가능하고, CREATE TABLE 또는 ALTER TABLE 문을 사용

하여 PRIMARY KEY 또는 UNIQUE 절을 지정함으로써 정의할 수 있습니다. 예를 들어, 테이블의 직원 번호 컬럼에 고유한 제한조건을 정의하여 모든 직원이 고유 번호를 갖도록 할 수 있습니다.

참조 무결성 제한조건

고유한 제한조건 및 외부 키를 정의하여 테이블 사이의 관계를 정의하고 그렇게 함으로써 특정한 비즈니스 규칙을 시행할 수 있습니다. 고유 키 및 외부 키 제한조건의 결합은 참조 무결성 제한조건으로 공통적으로 참조할 수 있습니다. 외부 키가 참조하는 고유한 제한조건을 상위 키라고 합니다. 외부 키는 특정 상위 키를 참조하거나 그것과 관련되어 있습니다. 예를 들어, 모든 직원(EMPLOYEE 테이블)은 기존의 부서(DEPARTMENT 테이블)에 속해야 한다는 규칙이 있을 수 있습니다. 그리고 EMPLOYEE 테이블에 있는 부서 번호는 외부 키로 정의되어 있고, DEPARTMENT 테이블에 있는 부서 번호는 기본 키로 정의되어 있습니다. 다음 도표는 참조 무결성 제한조건을 보여줍니다.

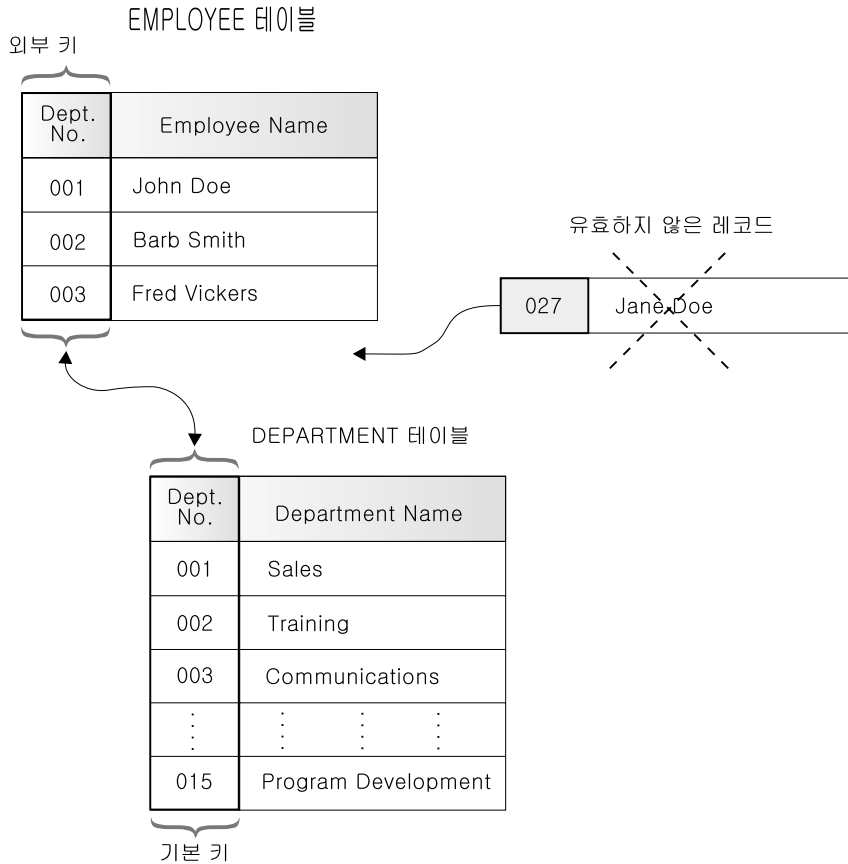


그림 4. 외부 및 기본 제한조건은 관계를 정의하고 데이터를 보호합니다.

테이블 점검 제한조건

테이블 점검 제한조건은 테이블의 각 행을 점검하는 조건을 지정합니다. 각 컬럼에 대해 점검 제한조건을 지정할 수 있습니다. CREATE 또는 ALTER TABLE 문을 사용하여 점검 제한조건을 추가할 수 있습니다.

다음의 명령문은 아래의 제한조건을 가지고 테이블을 작성한 것입니다.

- 부서 번호의 값은 10에서 100의 범위에 있어야 합니다.
- 직원의 직함은 “Sales”, “Mgr” 또는 “Clerk” 중에 하나만 될 수 있습니다.
- 1986년 이전에 입사한 모든 직원의 급여는 \$40,500 이상이어야 합니다.


```

CREATE TABLE EMP
  (ID          SMALLINT NOT NULL,
   NAME       VARCHAR(9),
   DEPT       SMALLINT CHECK (DEPT BETWEEN 10 AND 100),
   JOB        CHAR(5)  CHECK (JOB IN ('Sales', 'Mgr', 'Clerk')),
   HIREDATE   DATE,
   SALARY     DECIMAL(7,2),
   COMM       DECIMAL(7,2),
   PRIMARY KEY (ID),
   CONSTRAINT YEARSAL CHECK
     (YEAR(HIREDATE) >= 1986 OR SALARY > 40500) )

```

조건을 평가해서 거짓으로 판명될 때만 제한조건이 위반됩니다. 예를 들어, DEPT 가 삽입된 행에 대해 NULL이면, DEPT의 값이 제한조건에서 정의한 대로 10에서 100 사이에 있어야 한다고 하더라도, 그 삽입은 오류없이 진행됩니다.

다음 명령문은 직원의 총 보상금이 \$15,000을 초과해야 한다는 제한조건을 COMP 라고 이름 지정된 EMPLOY 테이블에 추가합니다.

```

ALTER TABLE EMP
  ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)

```

테이블에 있는 기존의 행이 새로운 제한조건을 위반하지 않는지 확인하기 위해 점검이 있을 것입니다. 다음과 같이 SET CONSTRAINTS 문을 사용하여 이 점검을 지연할 수 있습니다.

```

SET CONSTRAINTS FOR EMP OFF
ALTER TABLE EMP ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)
SET CONSTRAINTS FOR EMP IMMEDIATE CHECKED

```

먼저, SET CONSTRAINTS 문을 사용하여 테이블을 점검하는 제한조건을 지연시키십시오. 제한조건을 점검하지 않고 하나 이상의 제한조건을 테이블에 추가할 수 있습니다. 그런 다음에 SET CONSTRAINTS 문을 다시 발행하여 제한조건 점검을 되돌려 지연된 제한조건 점검을 수행하십시오.

트리거

트리거는 조치 세트를 정의합니다. 트리거는 지정된 기본 테이블에서 데이터를 수정하는 조치에 의해 활성화됩니다.

일부 조치에서는 다음과 같은 목적으로 트리거를 사용합니다.

- 입력 데이터에 대한 검증을 수행할 때

- 새로 삽입된 행에 대한 값을 자동으로 생성할 때
- 상호 참조 목적으로 다른 테이블을 읽을 때
- 감사 추적 목적으로 다른 테이블에 쓸 때
- 전자 메일 메시지를 통해 경고를 지원할 때

트리거를 사용하면, 응용프로그램을 더 신속하게 개발할 수 있고, 비즈니스 규칙을 전역적으로 시행할 수 있으며, 응용프로그램과 데이터의 유지보수를 더 쉽게 할 수 있습니다.

DB2 Universal Database는 몇 가지 유형의 트리거를 지원합니다. 트리거가 DELETE, INSERT 또는 UPDATE 연산 직후에 시작할 수 있도록 정의할 수 있습니다. 각 트리거는 생략 가능한 검색 조건을 포함할 수 있는 트리거 조차라고 부르는 일련의 SQL 문을 포함합니다.

사후 트리거는 각 행 또는 명령문에 대해 한번 트리거된 조치를 수행하도록 정의를 할 수 있습니다. 사전 트리거는 항상 각 행에 대해 트리거된 조치를 수행합니다.

트리거 연산을 수행하기 전에 특정 조건을 점검하거나 입력값이 테이블에 저장되기 전에 입력값을 변경하려면 INSERT, UPDATE, DELETE 문 앞에 트리거를 사용하십시오.

메시지 송신과 같이 트리거 연산의 일부로 필요할 수 있는 다른 작업을 수행하거나 필요한만큼 값을 전달하려면, 사후 트리거를 사용하십시오.

다음 예는 사전 및 사후 트리거를 사용한 예입니다. 레코드와 트랙이 재고 가격으로 변경되는 응용프로그램을 생각해 봅시다. 데이터베이스에는 다음과 같이 정의된 CURRENTQUOTE 및 QUOTEHISTORY 테이블이 있습니다.

```
CREATE TABLE CURRENTQUOTE
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
STATUS VARCHAR(9))
```

```
CREATE TABLE QUOTEHISTORY
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
TIMESTAMP TIMESTAMP)
```

다음은 CURRENTQUOTE의 QUOTE 컬럼을 갱신할 때 사용하는 명령문입니다.

```
UPDATE CURRENTQUOTE
  SET QUOTE = 68.5
  WHERE SYMBOL = 'IBM'
```

CURRENTQUOTE의 STATUS 컬럼은 다음과 같은 재고 상태를 반영하도록 갱신되어야 합니다.

- 값 증가
- 금년에 새로 증가한 재고
- 값 하락
- 금년에 새로 하락한 재고
- 값 안정

이 작업은 다음 사전 트리거를 사용하여 수행됩니다.

1

```
CREATE TRIGGER STOCK_STATUS
  NO CASCADE BEFORE UPDATE OF QUOTE ON CURRENTQUOTE
  REFERENCING NEW AS NEWQUOTE OLD AS OLDQUOTE
  FOR EACH ROW MODE DB2SQL
```

2

```
SET NEWQUOTE.STATUS =
```

3

```
CASE
```

4

```
  WHEN NEWQUOTE.QUOTE >=
    (SELECT MAX(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
  THEN 'High'
```

5

```

WHEN NEWQUOTE.QUOTE <=
    (SELECT MIN(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
THEN 'Low'

```

6

```

WHEN NEWQUOTE.QUOTE > OLDQUOTE.QUOTE
    THEN 'Rising'
WHEN NEWQUOTE.QUOTE < OLDQUOTE.QUOTE
    THEN 'Dropping'
WHEN NEWQUOTE.QUOTE = OLDQUOTE.QUOTE
    THEN 'Steady'

```

END

1

이 코드 블록은 STOCK_STATUS를 CURRENTQUOTE 테이블의 QUOTE 컬럼을 갱신하기 전에 활성화되어야 하는 트리거로서 정의합니다. 두번째 라인은 CURRENTQUOTE 테이블의 실제 갱신에 의해 일어나는 모든 변경이 데이터베이스에 적용되기 전에 트리거 조치가 적용되어야 한다는 것을 지정합니다. NO CASCADE 절은 트리거된 조치가 다른 트리거를 활성화해서는 안된다는 것을 의미합니다. 세번째 라인은 새 값(NEWQUOTE) 및 이전 값(OLDQUOTE)에 대한 컬럼 이름의 규정자로 사용되어야 하는 이름을 지정합니다. 이러한 상관 이름(NEWQUOTE 및 OLDQUOTE)으로 규정된 컬럼 이름을 전이 변수라고 합니다. 네번째 행은 트리거 조치가 각 행에 실행되어야 한다는 것을 나타냅니다.

2

이것은 이 트리거의 트리거 조치에서 첫번째이자 유일한 SQL 문의 시작을 나타냅니다. SET 전이 변수 명령문은 트리거를 시작한 명령문에 의해 갱신중인 테이블의 행에 있는 컬럼에 값을 지정하기 위해 트리거에서 사용됩니다. 이 명령문은 CURRENTQUOTE 테이블의 STATUS 컬럼에 값을 지정합니다.

3

지정 오른쪽에 사용된 표현식은 CASE 표현식입니다. CASE 표현식은 END 키워드까지 확장할 수 있습니다.

4

첫번째 경우에는 새 할당량(NEWQUOTE.QUOTE)이 당년에 재고 기호의 최대 값을 초과하는지를 알기 위해 점점합니다. 부속 조치는 따라오는 사후 트리거에 의해 갱신된 QUOTEHISTORY 테이블을 사용합니다.

- 5 두번째 경우에는 새 할당량(NEWQUOTE.QUOTE)이 당년 재고 기호의 최소 값보다 작은지를 알기 위해 점검합니다. 부속 조치는 따라오는 사후 트리거에 의해 갱신된 QUOTEHISTORY 테이블을 사용합니다.
- 6 마지막 세 경우에는 새 할당량(NEWQUOTE.QUOTE)을 테이블에 있었던 할당량(OLDQUOTE.QUOTE)과 비교하여 그것이 더 큰지, 작는지 또는 같은지 판별합니다. SET 전이 변수 명령문은 여기에서 끝납니다.

CURRENTQUOTE 테이블의 항목을 갱신하는 것 외에, 새 할당량을 시간소인과 함께 QUOTEHISTORY 테이블에 복사하여 감사 레코드를 작성해야 합니다. 이 작업은 다음 사후 트리거를 사용하여 수행됩니다.

1

```
CREATE TRIGGER RECORD_HISTORY
AFTER UPDATE OF QUOTE ON CURRENTQUOTE
REFERENCING NEW AS NEWQUOTE
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
```

2

```
INSERT INTO QUOTEHISTORY
VALUES (NEWQUOTE.SYMBOL, NEWQUOTE.QUOTE, CURRENT TIMESTAMP);
END
```

1

이 코드 블록은 RECORD_HISTORY로 이름 지정된 트리거를 CURRENTQUOTE 테이블의 QUOTE 컬럼의 갱신 후에 시작되어야 하는 트리거로서 정의합니다. 세번째 라인은 새 값(NEWQUOTE)에 대한 컬럼 이름의 규정자로 사용되어야 하는 이름을 지정합니다. 네번째 행은 트리거 조치가 각 행에 실행되어야 한다는 것을 나타냅니다.

2

이 트리거의 트리거 조치에는 갱신된 행(NEWQUOTE.SYMBOL 및 NEWQUOTE.QUOTE) 및 현 시간소인의 데이터를 사용하여 QUOTEHISTORY 테이블에 행을 삽입하는 단일 SQL 문이 포함되어 있습니다.

CURRENT TIMESTAMP는 시간소인을 포함하고 있는 특수 레지스터입니다. 목록 및 표현식은 81 페이지의 『특수 레지스터』에 나와 있습니다.

조인

두 개 이상의 테이블 데이터를 결합하는 프로세스를 테이블 조인이라고 합니다. 데이터베이스 관리 프로그램은 지정된 테이블의 행을 모두 결합할 수 있습니다. 각 결합에 대해서 데이터베이스 관리 프로그램은 조인 조건을 조사합니다. 조인 조건은 몇 가지 제한사항이 있는 검색 조건입니다. *SQL* 참조서에서 제한사항에 대한 자세한 목록을 참조하십시오.

조인 조건에 포함된 컬럼의 데이터 유형이 같을 필요는 없으나, 호환되어야 합니다. 조인 조건은 다른 검색 조건과 같은 방식으로 평가되며 비교에도 같은 규칙이 적용됩니다.

조인 조건을 지정하지 않으면, *FROM* 절에 나열된 테이블의 행이 서로 전혀 관련이 없다고 해도 모든 행들의 결합이 리턴됩니다. 그 결과를 두 테이블의 교차 곱이라고 합니다.

이 절의 예는 다음 두 테이블에 기초한 것입니다. 그것들은 샘플 데이터베이스 테이블을 단순화한 것으로서 샘플 데이터베이스에는 존재하지 않습니다. 그것들은 일반적으로 조인에 대한 흥미있는 점을 요약하는 데 사용됩니다. *SAMP_STAFF*에는 계약자로서 채용되지 않은 직원의 이름 및 그들의 직업에 대한 설명이 나열되어 있고, *SAMP_PROJECT*에는 직원(계약직 및 정규직)의 이름 및 그들이 참여하는 프로젝트의 이름이 나열되어 있습니다.

테이블은 다음과 같습니다.

NAME	PROJ
Haas	AD3100
Thompson	PL2100
Walker	MA2112
Lutz	MA2111

그림 5. *SAMP_PROJECT* 테이블

NAME	JOB
Haas	PRES
Thompson	MANAGER
Lucchessi	SALESREP
Nicholls	ANALYST

그림 6. SAMP_STAFF 테이블

다음 예는 두 테이블의 교차 곱을 작성합니다. 조인 조건이 지정되지 않았으므로 모든 행의 결합이 표시됩니다.

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT, SAMP_STAFF
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Thompson	PL2100	Haas	PRES
Walker	MA2112	Haas	PRES
Lutz	MA2111	Haas	PRES
Haas	AD3100	Thompson	MANAGER
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	Thompson	MANAGER
Lutz	MA2111	Thompson	MANAGER
Haas	AD3100	Lucchessi	SALESREP
Thompson	PL2100	Lucchessi	SALESREP
Walker	MA2112	Lucchessi	SALESREP
Lutz	MA2111	Lucchessi	SALESREP
Haas	AD3100	Nicholls	ANALYST
Thompson	PL2100	Nicholls	ANALYST
Walker	MA2112	Nicholls	ANALYST
Lutz	MA2111	Nicholls	ANALYST

두 가지 주요 조인의 유형은 내부 조인 및 외부 조인입니다. 지금까지 모든 예에서는 내부 조인만을 사용했습니다. 내부 조인은 조인 조건을 충족시키는 공통 곱의 행만을 유지합니다. 행이 한 테이블에만 존재하고 다른 테이블에는 존재하지 않는다면, 그 정보는 결과 테이블에 포함되지 않습니다.

다음은 두 테이블의 내부 조인을 산출하는 예입니다. 내부 조인은 프로젝트에 할당된 정규 직원을 나열합니다.

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT, SAMP_STAFF
WHERE SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

또는 아래와 같이 내부 조인을 지정할 수 있습니다.

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT INNER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

결과는 다음과 같습니다.

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Thompson	PL2100	Thompson	MANAGER

내부 조인의 결과는 좌우 테이블에 있는 NAME 컬럼의 값이 일치하는 행으로 구성됨을 유의하십시오. 'Haas' 및 'Thompson'은 정규 직원을 나열하는 SAMP_STAFF 테이블 및 프로젝트에 포함된 정규 직원 및 계약 직원을 나열하는 SAMP_PROJECT 테이블 모두에 나열되어 있습니다.

다른 조인은 내부 조인과 내부 조인에서 누락된 왼쪽 테이블, 오른쪽 테이블 및 두 테이블에 있는 행을 결합한 것입니다. 두 테이블에서 외부 조인을 수행하면, 한 테이블은 왼쪽 테이블로, 다른 테이블은 오른쪽 테이블로 임의적으로 지정할 수 있습니다. 외부 조인에는 세 가지 유형이 있습니다.

1. **왼쪽 외부 조인**에는 내부 조인 및 내부 조인에 포함되지 않은 왼쪽 테이블 행이 포함됩니다.
2. **오른쪽 외부 조인**에는 내부 조인 및 내부 조인에 포함되지 않은 오른쪽 테이블 행이 포함됩니다.
3. **완전 외부 조인**에는 내부 조인 및 내부 조인에 포함되지 않은 좌우 테이블의 행이 포함됩니다.

표시될 컬럼을 지정하려면 SELECT 문을 사용하십시오. FROM 절에는, 키워드가 LEFT OUTER JOIN, RIGHT OUTER JOIN 및 FULL OUTER JOIN이 뒤에 오는 첫번째

테이블의 이름이 나열됩니다. 다음에는 키워드 ON이 다음에 오게 하여 두번째 테이블을 지정해야 합니다. 키워드 ON 다음에, 조인 조건을 지정하여 조인될 테이블 사이의 관계를 표현하십시오.

다음 예에서, SAMP_STAFF는 오른쪽 테이블로, SAMP_PROJECT는 왼쪽 테이블로 지정됩니다. LEFT OUTER JOIN을 사용하여 (SAMP_PROJECT에 나열된) 모든 정규 및 계약 직원의 이름과 프로젝트 번호 그리고 (SAMP_STAFF에 나열된) 정규 직원이라면 직함 이름까지 나열합니다.

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT LEFT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Lutz	MA2111	-	-
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	-	-

모든 컬럼에 값이 있는 행은 내부 조인의 결과입니다. 이들은 조인 조건을 만족시키는 행입니다. 'Haas' 및 'Thompson'은 SAMP_PROJECT(왼쪽 테이블) 및 SAMP_STAFF(오른쪽 테이블) 둘다에 나열되어 있습니다. 조인 조건이 만족되지 않는 행은 널(Null) 값이 오른쪽 테이블 컬럼에 나타납니다. 'Lutz' 및 'Walker'는 SAMP_PROJECT 테이블에는 나열되어 있지만 SAMP_STAFF 테이블에는 나열되어 있지 않습니다. 왼쪽 테이블의 모든 행이 결과 집합에 포함되어 있음을 유의하십시오.

다음 예에서, SAMP_STAFF는 오른쪽 테이블로, SAMP_PROJECT는 왼쪽 테이블로 지정됩니다. RIGHT OUTER JOIN을 사용하여, (SAMP_STAFF에 나열된) 모든 정규 직원의 이름 및 직함 그리고 그 직원이 (SAMP_PROJECT에 나열된) 프로젝트에 할당되었다면 프로젝트 번호까지 나열합니다.

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT RIGHT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

결과는 다음과 같습니다.

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER

왼쪽 외부 조인에서 처럼, 모든 컬럼에 값이 있는 행은 내부 조인의 결과입니다. 이들은 조인 조건을 만족시키는 행입니다. 'Haas' 및 'Thompson'은 SAMP_PROJECT(왼쪽 테이블) 및 SAMP_STAFF(오른쪽 테이블) 둘다에 나열되어 있습니다. 조인 조건을 만족시키지 못한 행의 경우, 널(Null) 값이 오른쪽 테이블의 컬럼에 나타납니다. 'Lucchessi' 및 'Nicholls'은 프로젝트에 지정된 정규 직원입니다. 이 직원들은 SAMP_STAFF에는 나열되어 있지만, SAMP_PROJECT에는 나열되어 있지 않습니다. 오른쪽 테이블의 모든 행이 결과 집합에 포함되어 있음을 유의하십시오.

다음 예에서, SAMP_PROJECT 및 SAMP_STAFF 테이블에 FULL OUTER JOIN을 사용합니다. 프로젝트에 할당되지 않은 직원을 비롯하여 모든 정규 직원 및 계약 직원의 이름이 나열됩니다.

```

SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT FULL OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

결과는 다음과 같습니다.

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER
Lutz	MA2111	-	-
Walker	MA2112	-	-

이 결과에는 왼쪽 외부 조인, 오른쪽 외부 조인 그리고 내부 조인이 포함되어 있습니다. 모든 정규 및 계약 직원이 나열되어 있습니다. 좌우 외부 조인처럼 결합 조건이 만족되지 못한 값은 해당 컬럼에 널(Null) 값이 나타납니다. SAMP_STAFF 및 SAMP_PROJECT에서의 내 행은 결과 세트에 포함됩니다.

복합 조회

DB2 Universal Database를 이용하면 ROLLUP 및 CUBE를 사용하여 단일 결과 집합에 있는 복수의 컬럼을 그룹화하거나 통합하거나 볼 수 있습니다. 이 새롭고 강력한 기능을 이용하면, SQL 기반 데이터 분석 능력을 향상시키면서도 단순화할 수 있습니다.

데이터베이스에서 유용한 데이터를 끌어내는 여러 가지 방법이 있습니다. 사용자는 순환 조회를 사용하여 기존 데이터 집합에서 결과 테이블을 산출할 수 있습니다.

ROLLUP 및 CUBE 조회

조회에서 GROUP BY 절에서 ROLLUP 및 CUBE 연산을 지정하십시오.

ROLLUP 그룹화는 정규 그룹화 행 및 부분 합 행을 포함하는 결과 집합을 산출합니다. CUBE 그룹화는 ROLLUP 행 및 공통 테이블 행을 포함하는 결과 집합을 산출합니다.

따라서, ROLLUP의 경우 1인당 월별 매상 및 총 매상을 확보할 수 있습니다. CUBE의 경우 1인당 총매상에 행이 추가로 포함됩니다.

SQL 참조서에서 자세한 내용을 참조하십시오.

순환 조회

순환 조회는 더 많은 결과를 알아내기 위해 결과 데이터를 반복적으로 사용하는 조회입니다. 사용자는 순환 조회가 트리나 그래프를 거치면서 자세히 검토합니다.

실제 예로는 예약 시스템, 네트워크 플랜 및 스케줄등이 있습니다.

순환 조회는 자신의 이름에 대한 참조를 포함하는 공통 테이블 표현식을 사용하여 작성합니다.

순환 조회의 예를 보려면 *SQL 참조서* 부분을 보십시오.

OLAP 함수

OnLine Analytical Processing (OLAP) 기능은 데이터 창을 통해 컬럼 기능 조작을 수행합니다. 이 창에서는 행 파티션, 파티션 내의 행 정렬 또는 총계 그룹을 지정할 수 있습니다. 사용자는 총계 그룹을 통해 현재 행에 관련하여 어느 행을 계산에 사용할 것인지를 지정할 수 있습니다. 이러한 창을 사용하여 누적 합계 또는 평균 이동과 같은 조작을 수행할 수 있습니다.

기존의 컬럼 기능(SUM 및 AVG)에 대해 창을 지정할 수 있는 것 외에도 OLAP 함수를 사용하여 서열화(RANK 및 DENSE_RANK) 조작을 수행하고 행 번호(ROW_NUMBER)를 제공하여 특정 파티션 및 행 순서를 부여할 수 있습니다.

다음 조회 예는 급여를 기본으로 한 부서 내에서의 직원의 서열을 제공하며, 부서 내에서의 급여의 누계를 보여줍니다(부서 15 및 38에 대해).

```
SELECT NAME, DEPT,  
       RANK () OVER (PARTITION BY DEPT ORDER BY SALARY DESC) AS RANK,  
       SUM (SALARY) OVER (PARTITION BY DEPT  
                          ORDER BY SALARY DESC  
                          ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)  
       AS CUMULATIVE_SUM  
FROM STAFF  
WHERE DEPT IN (15,38)  
ORDER BY DEPT, RANK
```

이 명령문은 다음과 같은 결과를 산출합니다.

NAME	DEPT	RANK	CUMULATIVE_SUM
Hanes	15	1	20659.80
Rothman	15	2	37162.63
Ngan	15	3	49670.83
Kermisch	15	4	61929.33
O'Brien	38	1	18006.00
Marenghi	38	2	35512.75
Quigley	38	3	52321.05
Naughton	38	4	65275.80
Abrahams	38	5	77285.55

제8장 데이터 조작의 조정 및 향상

여기에서는 DB2 Universal Database에 있는 오브젝트 지향 확장에 대해 간략하게 소개합니다. 오브젝트 지향 확장을 사용하면 많은 이점이 있습니다. 사용자 정의 유형(UDT)이 사용자의 응용프로그램에 이용할 수 있는 데이터 유형의 집합을 증가시키는 반면, 사용자 정의 함수(UDF)는 응용프로그램에 고유한 함수의 작성을 허용합니다. UDF는 일관 작동 및 유형의 캡슐화를 제공함으로써 UDT의 메소드로 작용합니다.

특수 레지스터 및 시스템 카탈로그는 다음에 논의됩니다. 특수 레지스터는 데이터베이스 관리 프로그램에 의해 정의된 저장 영역으로서, SQL 문이 참조할 수 있는 정보를 저장하는 데 사용됩니다. 특수 레지스터는 연결시 설정되며 해당 응용프로그램에 대해 고유합니다. 시스템 카탈로그는 데이터베이스 오브젝트의 논리적 및 물리적 구조에 대한 정보를 제공합니다.

여기에는 다음과 같은 내용이 들어 있습니다.

- 사용자 정의 유형
- 사용자 정의 함수
- 대형 오브젝트(LOB)
- 특수 레지스터
- 카탈로그 뷰 소개

위 주제에 대한 세부적인 설명은 이 책의 범위를 넘으며 이에 대한 내용은 *SQL 참조서 및 관리 안내서*에서 자세한 내용을 참조하십시오.

사용자 정의 유형

구별 유형(*distinct type*)은 내적 표현을 기존의 유형(『소스』 유형)과 공유하는 사용자 정의 데이터 유형이지만 대부분의 연산과 분리되어 있고 호환이 되지 않습니다. 예를 들어, 나이 유형, 체중 유형 그리고 신장 유형을 정의한다면 이 모든 것은 다른 의미를 가져야 하고 그들의 내적인 표현을 위해 내장 데이터 유형 INTEGER를 사용합니다.

다음은 PAY라는 구별 유형을 작성하는 예입니다.

CREATE DISTINCT TYPE PAY AS DECIMAL(9,2) WITH COMPARISONS

PAY는 내장 데이터 유형 DECIMAL(9,2)와 같은 표현을 가지고 있지만, 그것은 DECIMAL(9,2) 또는 다른 유형과 호환되지 않는 분리된 유형입니다. 그것은 같은 구별 유형과만 호환됩니다. 또한, DECIMAL을 처리할 수 있는 연산자 및 함수가 여기에는 적용되지 않습니다. 예를 들어, PAY 데이터 유형의 값은 INTEGER 데이터 유형의 값과 곱할 수 없습니다. 따라서, PAY 데이터 유형에만 적용할 수 있는 함수를 작성해야 합니다.

구별 데이터 유형을 사용하면 우발적인 실수를 줄일 수 있습니다. 예를 들어, 테이블 EMPLOYEE의 SALARY 컬럼을 PAY 데이터 유형으로 정의한다면, 그들의 소스 유형이 같아도, SALARY 컬럼은 COMM에 추가되지 않습니다.

구별 데이터 유형은 유형 변환(*casting*)을 지원합니다. 소스 유형은 구별 데이터 유형으로 유형 변환할 수 있고, 구별 데이터 유형은 소스 유형으로 유형 변환할 수 있습니다. 예를 들어, 테이블 EMPLOYEE의 SALARY 컬럼 이름을 PAY 데이터 유형으로 정의한다면, 다음 예는 비교 연산자에서 실패할 것입니다.

```
SELECT * FROM EMPLOYEE
WHERE DECIMAL(SALARY) = 41250
```

DECIMAL(SALARY)은 소수 데이터 유형을 리턴합니다. 반대로, 숫자 데이터 유형은 PAY 유형으로 유형 변환할 수 있습니다. 예를 들어, PAY(41250)을 사용하여 숫자 41250을 유형 변환할 수 있습니다.

사용자 정의 함수

32 페이지의 『함수 사용』에서 설명했듯이, DB2 Universal Database는 내장 및 사용자 정의 함수(UDF)를 제공합니다. 그러나, 이 함수 집합이 모든 요구조건을 만족시키지는 못할 것입니다. 종종 특별한 타스크를 위해 사용자 정의 함수를 작성해야 할 때가 있습니다. 사용자 정의 함수를 이용하면, 사용자에게 의해 정의된 함수를 작성할 수 있습니다.

사용자 정의 기능에는 전래(또는 템플릿), 외부 스칼라, 외부 테이블, OLE DB 외부 테이블 등 네 가지 유형이 있습니다.

이 절에서는 전래 및 외부 스칼라 유형에 대해 설명합니다. SQL 참조서에서 외부 테이블 및 OLE DB 테이블 유형에 대한 자세한 내용을 참조하십시오.

전래 사용자 정의 함수는 데이터베이스가 이미 알고 있는 또 다른 내장 및 사용자 정의 함수를 선택적으로 참조하는 사용자 정의 유형을 허용합니다. 스칼라 및 컬럼 함수를 모두 사용할 수 있습니다.

다음 예에서는 입력으로서 DECIMAL 데이터 유형을 취하는 내장 MAX 컬럼 함수에 기초하여 사용자 정의 함수(MAX라고 함)가 작성됩니다. MAX UDF는 PAY 유형을 입력으로 취하고 PAY 유형을 출력으로 리턴합니다.

```
CREATE FUNCTION MAX(PAY) RETURNS PAY
SOURCE MAX(DECIMAL)
```

외부 사용자 정의 함수는 프로그래밍 언어로 사용자가 작성합니다. 사용자 정의 함수로는 외부 스칼라 함수 및 외부 테이블 함수가 있고 그 내용은 SQL 참조서에 설명되어 있습니다.

다른 예로서, 문자열에 있는 단어를 세는 함수를 이미 작성했다면 CREATE FUNCTION 문을 사용하여 WORDCOUNT라는 이름으로 데이터베이스에 이를 등록할 수 있습니다. 그러면 이 함수는 SQL 문에서 사용할 수 있습니다.

다음 명령문은 직원 번호와 ASCII 양식의 직원 이름의 단어 수를 리턴합니다. WORDCOUNT는 사용자가 데이터베이스로 등록했으며 지금 명령문에서 사용되는 외부 스칼라 함수입니다.

```

SELECT EMPNO, WORDCOUNT(RESUME)
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'

```

사용자 정의 함수 작성에 대해서는 응용프로그램 개발 안내서에서 자세한 내용을 참조하십시오.

대형 오브젝트(LOB)

대형 오브젝트 및 그 약어인 *LOB*는 세 가지 데이터 유형(BLOB, CLOB, DBCLOB)을 참조하는 데 사용됩니다. 이러한 유형에는 오디오, 사진, 문서와 같은 오브젝트처럼 많은 양의 데이터를 포함할 수 있습니다.

2진 대형 오브젝트(*BLOB*)는 가변 길이 문자열로서 바이트 단위로 측정되고 2기가바이트의 길이까지 될 수 있습니다. 2진 대형 오브젝트(*BLOB*)는 기본적으로 그림, 음성 그리고 혼합 미디어 같은 비전통적인 데이터를 보관하기 위한 것입니다.

문자 대형 오브젝트(*CLOB*)는 가변 길이 문자열로서 바이트 단위로 측정되고 2기가바이트 길이까지 될 수 있습니다. *CLOB*는 문서와 같은 대형 1바이트 문자 집합 데이터를 저장하는 데 사용됩니다. *CLOB*는 문자열로 간주됩니다.

2바이트 문자 대형 오브젝트(*DBCLOB*)는 2바이트 문자로 된 가변 길이 문자열로서 2기가바이트 길이(1 073 741 823 바이트 문자)까지 될 수 있습니다. *DBCLOB*는 문서와 같은 대형 2바이트 문자 집합 데이터를 저장하는 데 사용됩니다. 문자 대형 오브젝트(*CLOB*)는 그래픽 문자열로 간주됩니다.

대형 오브젝트(LOB) 조작

대형 오브젝트(*LOB*) 값은 매우 클 수 있으므로, 이러한 값을 데이터베이스 서버에서 클라이언트 응용프로그램으로 전송하는 작업에는 시간이 소요될 수 있습니다. 그러나 일반적으로 대형 오브젝트(*LOB*) 값은 전체가 아닌 한 조각씩 처리됩니다. 응용프로그램 메모리에 전체 대형 오브젝트(*LOB*) 값을 저장할 필요가 없는 경우(원치 않는 경우), 응용프로그램은 대형 오브젝트 위치 지정자 변수를 통해 이 값을 참조할 수 있습니다.

그러면 그 다음의 명령문에서 대형 오브젝트(LOB) 전체를 꼭 검색하지 않고서도 데이터에 대한 조작을 수행하기 위해서 위치 지정자를 사용할 수 있습니다. 위치 지정자 변수는 응용프로그램의 저장 요구량을 줄이고 클라이언트와 서버 사이에 데이터의 흐름을 줄임으로써 성능을 향상시키기 위해서 사용됩니다.

또 다른 메카니즘은 파일 참조 변수입니다. 파일 참조 변수는 대형 오브젝트(LOB)를 직접 파일로 검색하고 테이블에 있는 대형 오브젝트(LOB)를 파일로부터 직접 갱신하는 데 사용됩니다. 파일 참조 변수는 대형 오브젝트(LOB) 데이터를 저장할 필요가 없기 때문에 응용프로그램의 저장 요구량을 줄이는 데 사용됩니다. 응용프로그램 개발 안내서 및 SQL 참조서에서 더 많은 내용을 참조하십시오.

특수 레지스터

특수 레지스터는 데이터베이스 관리 프로그램에 의한 연결을 위해 정의된 저장 영역으로서 SQL 문에서 참조할 수 있는 정보를 저장하기 위해 사용됩니다. 흔하게 사용되는 특수 레지스터의 몇 가지 예가 다음에 있습니다. SQL 참조서에서 모든 특수 레지스터의 목록 및 자세한 내용을 참조하십시오.

- CURRENT DATE: SQL 문 실행시에 날짜 시계에 따라 데이터를 보관합니다.
- CURRENT FUNCTION PATH: 함수 및 데이터 유형 참조를 해결하기 위해 사용되는 함수 경로를 지정하는 값을 보관합니다.
- CURRENT SERVER: 현재 응용프로그램 서버를 지정합니다.
- CURRENT TIME: SQL 문 실행시에 날짜 시계에 따라 시간을 보관합니다.
- CURRENT TIMESTAMP: SQL 문 실행시에 날짜 시계에 따라 시간소인을 지정합니다.
- CURRENT TIMEZONE: 응용프로그램 서버에서의 세계 표준시 및 지역시간의 차이를 지정합니다.
- USER: 수행시 권한 부여 ID를 지정합니다.

VALUES 문을 사용하면 특수 레지스터의 내용을 표시할 수 있습니다. 예를 들면, 다음과 같습니다.

```
VALUES (CURRENT TIMESTAMP)
```

또한 다음과 같은 명령문을 사용할 수도 있습니다.

```
SELECT CURRENT_TIMESTAMP FROM ORG
```

이 명령문은 테이블에 있는 모든 행 항목에 대한 `TIMESTAMP`를 리턴합니다.

카탈로그 뷰 소개

DB2는 각 데이터베이스에 대해 확장 시스템 카탈로그 테이블 세트를 작성하고 유지보수합니다. 이 테이블에는 테이블, 뷰, 패키지, 참조 무결성 관계, 함수, 구별 유형 그리고 트리거와 같은 데이터베이스 오브젝트의 논리적 및 물리적 구조에 대한 정보가 들어 있습니다. 그 정보들은 데이터베이스가 작성될 때 작성되고 정상적인 조작중에 갱신됩니다. 사용자는 명시적으로 그것들을 작성하거나 없앨 수는 없지만, 그 내용을 조회하거나 볼 수는 있습니다.

SQL 참조서에서 자세한 내용을 참조하십시오.

시스템 카탈로그에서 행 선택

카탈로그 뷰는 다른 데이터베이스 뷰와 비슷합니다. 사용자는 *SQL* 문을 사용하여 시스템에 있는 다른 뷰를 보는 것과 같은 식으로 데이터를 볼 수 있습니다.

`SYSCAT.TABLES` 카탈로그에서 테이블에 대한 유용한 정보를 찾을 수 있습니다. 사용자가 작성한 기존 테이블 이름을 찾으려면, 다음과 비슷한 명령문을 실행하십시오.

```
SELECT TABNAME, TYPE, CREATE_TIME  
FROM SYSCAT.TABLES  
WHERE DEFINER = USER
```

이 명령문은 다음과 같은 결과를 산출합니다.

TABNAME	TYPE	CREATE_TIME
ORG	T	1999-07-21-13.42.55.128005
STAFF	T	1999-07-21-13.42.55.609001
DEPARTMENT	T	1999-07-21-13.42.56.069001
EMPLOYEE	T	1999-07-21-13.42.56.310001
EMP_ACT	T	1999-07-21-13.42.56.710001
PROJECT	T	1999-07-21-13.42.57.051001
EMP_PHOTO	T	1999-07-21-13.42.57.361001

EMP_RESUME	T	1999-07-21-13.42.59.154001
SALES	T	1999-07-21-13.42.59.855001
CL_SCHED	T	1999-07-21-13.43.00.025002
IN_TRAY	T	1999-07-21-13.43.00.055001

다음 목록에는 이 책에서 설명한 주제와 관련된 카탈로그 뷰가 포함되어 있습니다. 여러 카탈로그 뷰가 있으며 이들은 *SQL* 참조서 및 관리 안내서 매뉴얼에 자세히 나와 있습니다.

설명	카탈로그 뷰
점검 제한조건	SYSCAT.CHECKS
컬럼	SYSCAT.COLUMNS
점검 제한조건이 참조하는 컬럼	SYSCAT.COLCHECKS
키에서 사용되는 컬럼	SYSCAT.KEYCOLUSE
데이터 유형	SYSCAT.DATATYPES
함수 매개변수 또는 함수의 결과	SYSCAT.FUNCPARMS
참조 제한조건	SYSCAT.REFERENCES
스키마	SYSCAT.SCHEMATA
테이블 제한조건	SYSCAT.TABCONST
테이블	SYSCAT.TABLES
트리거	SYSCAT.TRIGGERS
사용자 정의 함수	SYSCAT.FUNCTIONS
뷰	SYSCAT.VIEWS

부록A. 샘플 데이터베이스 테이블

이 부록에는 샘플 데이터베이스 SAMPLE의 샘플 테이블에 포함된 정보와 그들을 작성하고 삭제하는 방법이 나와 있습니다.

추가적인 샘플 데이터베이스가 비즈니스 인텔리전스 기능을 데모하기 위해 DB2 Universal Database와 함께 제공됩니다. 또한 이 추가적인 샘플 데이터베이스는 비즈니스 인텔리전스 자습서에서 사용됩니다. 하지만, 샘플 데이터베이스 SAMPLE의 내용만 이 부록에 설명됩니다. 비즈니스 인텔리전스 데이터베이스에 대해서는 *Data Warehouse Center 관리 안내서*에서 자세한 내용을 참조하십시오.

샘플 테이블은 이 매뉴얼 및 이 라이브러리에 있는 다른 매뉴얼에서 사용됩니다. 그 외에, BLOB 및 CLOB 데이터 유형을 갖는 샘플 파일에 포함된 데이터가 나와 있습니다.

다음 절이 이 부록에 포함되어 있습니다.

- 86 페이지의 『샘플 데이터베이스』
- 86 페이지의 『샘플 데이터베이스 작성』
- 87 페이지의 『샘플 데이터베이스 삭제』
- 87 페이지의 『CL_SCHED 테이블』
- 87 페이지의 『DEPARTMENT 테이블』
- 88 페이지의 『EMPLOYEE 테이블』
- 91 페이지의 『EMP_ACT 테이블』
- 93 페이지의 『EMP_PHOTO 테이블』
- 93 페이지의 『EMP_RESUME 테이블』
- 94 페이지의 『IN_TRAY 테이블』
- 94 페이지의 『ORG 테이블』
- 95 페이지의 『PROJECT 테이블』
- 96 페이지의 『SALES 테이블』
- 97 페이지의 『STAFF 테이블』
- 98 페이지의 『STAFFG 테이블』

샘플 데이터베이스 테이블

- 99 페이지의 『BLOB 및 CLOB 데이터 유형으로 된 샘플 파일』
- 100 페이지의 『Quintana 사진』
- 100 페이지의 『Quintana 이력서』
- 102 페이지의 『Nicholls 사진』
- 102 페이지의 『Nicholls 이력서』
- 104 페이지의 『Adamson 사진』
- 104 페이지의 『Adamson 이력서』
- 106 페이지의 『Walker 사진』
- 106 페이지의 『Walker 이력서』.

샘플 테이블에서 대쉬(-)는 널(Null) 값을 의미합니다.

샘플 데이터베이스

이 책에 있는 예들은 샘플 데이터베이스를 사용한 것입니다. 이 예들을 사용하려면, SAMPLE 데이터베이스를 작성해야 합니다. 샘플 데이터베이스를 사용하려면, 데이터베이스 관리 프로그램을 설치해야 합니다.

샘플 데이터베이스 작성

실행가능한 파일이 샘플 데이터베이스를 작성합니다.² 데이터베이스를 작성하려면 사용자는 SYSADM 권한이 있어야 합니다.

- **UNIX** 플랫폼을 사용할 때

운영 체제 명령 프롬프트를 사용하고 있다면, 다음과 같이 입력하십시오.

```
sqllib/bin/db2samp1 <path>
```

여기서 *path*는 샘플 데이터베이스가 작성될 경로를 지정하는 생략 가능한 매개 변수입니다. Enter 키를 누르십시오.³ DB2SAMPL용 스키마는 CURRENT SCHEMA 특수 레지스터 값입니다.

- **OS/2 또는 Windows** 플랫폼을 사용할 때

2. 이 명령에 관련된 정보는 *Command Reference*에 있는 DB2SAMPL 명령을 보십시오.

3. 경로 매개변수를 지정하지 않으면, 샘플 테이블은 데이터베이스 관리 프로그램 구성 파일에 있는 DFTDBPATH에 의해 지정된 기본 경로 값에 작성됩니다.

운영 체제 명령 프롬프트를 사용하고 있다면, 다음과 같이 입력하십시오.

```
db2sample
```

여기서, *e*는 데이터베이스를 작성할 드라이브를 지정하는 생략 가능한 매개변수입니다. Enter 키를 누르십시오.⁴

사용자 프로파일 관리를 통해 워크스테이션에 로그인되지 않았다면, 시스템이 그렇게 하도록 프롬프트할 것입니다.

샘플 데이터베이스 삭제

샘플 데이터베이스에 액세스할 필요가 없다면, 다음과 같이 DROP DATABASE 명령을 사용하여 이를 삭제할 수 있습니다.

```
db2 drop database sample
```

CL_SCHED 테이블

이름:	CLASS_CODE	DAY	STARTING	ENDING
유형:	char(7)	smallint	time	time
설명:	클래스 코드(room:teacher)	4일 일정의 요일 번호	강의 시작 시간	강의 종료 시간

DEPARTMENT 테이블

이름:	DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
유형:	char(3) not null	varchar(29) not null	char(6)	char(3) not null	char(16)
설명:	부서 번호	일반적인 부서의 활동을 설명하는 이름	부서 관리자의 직원 번호 (EMPNO)	이 부서의 상위 부서	원격 위치 이름
값:	A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
	B01	PLANNING	000020	A00	-
	C01	INFORMATION CENTER	000030	A00	-
	D01	DEVELOPMENT CENTER	-	A00	-
	D11	MANUFACTURING SYSTEMS	000060	D01	-

4. 드라이브 매개변수가 지정되지 않으면, 샘플 데이터베이스는 DB2와 같은 드라이브에 작성됩니다.

샘플 데이터베이스 테이블

이름:	DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
	D21	ADMINISTRATION SYSTEMS	000070	D01	-
	E01	SUPPORT SERVICES	000050	A00	-
	E11	OPERATIONS	000090	E01	-
	E21	SOFTWARE SUPPORT	000100	E01	-

EMPLOYEE 테이블

이름:	EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE
유형:	char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date
설명:	직원 번호	이름	가운데 이름	성	소속 부서	전화 번호	입사일

JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
직함	공적 학력(연수)	성 (M 남, F 여)	생년월일	연봉	연간 보너스	연간 수당

EMPLOYEE 테이블의 값은 다음 페이지를 보십시오.

EMPNO	FIRSTNAME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date	char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18	F	1933-08-24	52750	1000	4220
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M	1948-02-02	41250	800	3300
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F	1941-05-11	38250	800	3060
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M	1925-09-15	40175	800	3214
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M	1945-07-07	32250	500	2580
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36170	700	2893
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29750	600	2380
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26150	500	2092
000110	VINCENZO	G	LUCCHESSI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46500	900	3720
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M	1942-10-18	29250	600	2340
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1925-09-15	23800	500	1904
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28420	600	2274
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M	1947-05-17	25280	500	2022
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22250	400	1780
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M	1951-01-05	24680	500	1974
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21340	500	1707
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20450	400	1636
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M	1941-05-29	27740	600	2217
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18270	400	1462
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29840	600	2387
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK	14	M	1935-05-30	22180	400	1774
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M	1954-03-31	28760	600	2301
000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M	1939-11-12	19180	400	1534
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F	1936-10-05	17250	300	1380
000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F	1953-05-26	27380	500	2190
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F	1936-03-28	26250	500	2100
000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M	1946-07-09	15340	300	1227
000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M	1936-10-27	17750	400	1420
000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F	1931-04-21	15900	300	1272

EMPNO	FIRSTNME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M	1932-08-11	19950	400	1596
000330	WING		LEE	E21	2103	1976-02-23	FIELDREP	14	M	1941-07-18	25370	500	2030
000340	JASON	R	GOUNOT	E21	5698	1947-05-05	FIELDREP	16	M	1926-05-17	23840	500	1907

EMP_ACT 테이블

이름:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
유형:	char(6) not null	char(6) not null	smallint not null	dec(5,2)	date	date
설명:	직원 번호	프로젝트 번호	작업 번호	프로젝트에 직원의 투입된 시간 비율	작업 시작일	작업 종료일
값:	000010	AD3100	10	.50	1982-01-01	1982-07-01
	000070	AD3110	10	1.00	1982-01-01	1983-02-01
	000230	AD3111	60	1.00	1982-01-01	1982-03-15
	000230	AD3111	60	.50	1982-03-15	1982-04-15
	000230	AD3111	70	.50	1982-03-15	1982-10-15
	000230	AD3111	80	.50	1982-04-15	1982-10-15
	000230	AD3111	180	1.00	1982-10-15	1983-01-01
	000240	AD3111	70	1.00	1982-02-15	1982-09-15
	000240	AD3111	80	1.00	1982-09-15	1983-01-01
	000250	AD3112	60	1.00	1982-01-01	1982-02-01
	000250	AD3112	60	.50	1982-02-01	1982-03-15
	000250	AD3112	60	.50	1982-12-01	1983-01-01
	000250	AD3112	60	1.00	1983-01-01	1983-02-01
	000250	AD3112	70	.50	1982-02-01	1982-03-15
	000250	AD3112	70	1.00	1982-03-15	1982-08-15
	000250	AD3112	70	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.50	1982-10-15	1982-12-01
	000250	AD3112	180	.50	1982-08-15	1983-01-01
	000260	AD3113	70	.50	1982-06-15	1982-07-01
	000260	AD3113	70	1.00	1982-07-01	1983-02-01
	000260	AD3113	80	1.00	1982-01-01	1982-03-01
	000260	AD3113	80	.50	1982-03-01	1982-04-15
	000260	AD3113	180	.50	1982-03-01	1982-04-15
	000260	AD3113	180	1.00	1982-04-15	1982-06-01
	000260	AD3113	180	.50	1982-06-01	1982-07-01
	000270	AD3113	60	.50	1982-03-01	1982-04-01
	000270	AD3113	60	1.00	1982-04-01	1982-09-01
	000270	AD3113	60	.25	1982-09-01	1982-10-15
	000270	AD3113	70	.75	1982-09-01	1982-10-15
	000270	AD3113	70	1.00	1982-10-15	1983-02-01

샘플 데이터베이스 테이블

이름:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000270	AD3113	80	1.00	1982-01-01	1982-03-01
	000270	AD3113	80	.50	1982-03-01	1982-04-01
	000030	IF1000	10	.50	1982-06-01	1983-01-01
	000130	IF1000	90	1.00	1982-01-01	1982-10-01
	000130	IF1000	100	.50	1982-10-01	1983-01-01
	000140	IF1000	90	.50	1982-10-01	1983-01-01
	000030	IF2000	10	.50	1982-01-01	1983-01-01
	000140	IF2000	100	1.00	1982-01-01	1982-03-01
	000140	IF2000	100	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-10-01	1983-01-01
	000010	MA2100	10	.50	1982-01-01	1982-11-01
	000110	MA2100	20	1.00	1982-01-01	1982-03-01
	000010	MA2110	10	1.00	1982-01-01	1983-02-01
	000200	MA2111	50	1.00	1982-01-01	1982-06-15
	000200	MA2111	60	1.00	1982-06-15	1983-02-01
	000220	MA2111	40	1.00	1982-01-01	1983-02-01
	000150	MA2112	60	1.00	1982-01-01	1982-07-15
	000150	MA2112	180	1.00	1982-07-15	1983-02-01
	000170	MA2112	60	1.00	1982-01-01	1983-06-01
	000170	MA2112	70	1.00	1982-06-01	1983-02-01
	000190	MA2112	70	1.00	1982-02-01	1982-10-01
	000190	MA2112	80	1.00	1982-10-01	1983-10-01
	000160	MA2113	60	1.00	1982-07-15	1983-02-01
	000170	MA2113	80	1.00	1982-01-01	1983-02-01
	000180	MA2113	70	1.00	1982-04-01	1982-06-15
	000210	MA2113	80	.50	1982-10-01	1983-02-01
	000210	MA2113	180	.50	1982-10-01	1983-02-01
	000050	OP1000	10	.25	1982-01-01	1983-02-01
	000090	OP1010	10	1.00	1982-01-01	1983-02-01
	000280	OP1010	130	1.00	1982-01-01	1983-02-01
	000290	OP1010	130	1.00	1982-01-01	1983-02-01
	000300	OP1010	130	1.00	1982-01-01	1983-02-01
	000310	OP1010	130	1.00	1982-01-01	1983-02-01
	000050	OP2010	10	.75	1982-01-01	1983-02-01
	000100	OP2010	10	1.00	1982-01-01	1983-02-01
	000320	OP2011	140	.75	1982-01-01	1983-02-01

이름:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000320	OP2011	150	.25	1982-01-01	1983-02-01
	000330	OP2012	140	.25	1982-01-01	1983-02-01
	000330	OP2012	160	.75	1982-01-01	1983-02-01
	000340	OP2013	140	.50	1982-01-01	1983-02-01
	000340	OP2013	170	.50	1982-01-01	1983-02-01
	000020	PL2100	30	1.00	1982-01-01	1982-09-15

EMP_PHOTO 테이블

이름:	EMPNO	PHOTO_FORMAT	PICTURE
유형:	char(6) not null	varchar(10) not null	blob(100k)
설명:	직원 번호	사진 형식	직원 사진
값:	000130	bitmap	db200130.bmp
	000130	gif	db200130.gif
	000130	xwd	db200130.xwd
	000140	bitmap	db200140.bmp
	000140	gif	db200140.gif
	000140	xwd	db200140.xwd
	000150	bitmap	db200150.bmp
	000150	gif	db200150.gif
	000150	xwd	db200150.xwd
	000190	bitmap	db200190.bmp
	000190	gif	db200190.gif
	000190	xwd	db200190.xwd

- 100 페이지의 『Quintana 사진』은 직원 Delores Quintana의 사진입니다.
- 102 페이지의 『Nicholls 사진』은 직원 Heather Nicholls의 사진입니다.
- 104 페이지의 『Adamson 사진』은 직원 Bruce Adamson의 사진입니다.
- 106 페이지의 『Walker 사진』은 직원 James Walker의 사진입니다.

EMP_RESUME 테이블

이름:	EMPNO	RESUME_FORMAT	RESUME
유형:	char(6) not null	varchar(10) not null	clob(5k)
설명:	직원 번호	이력서 형식	직원 이력서
값:	000130	ascii	db200130.asc

샘플 데이터베이스 테이블

이름:	EMPNO	RESUME_FORMAT	RESUME
	000130	script	db200130.scr
	000140	ascii	db200140.asc
	000140	script	db200140.scr
	000150	ascii	db200150.asc
	000150	script	db200150.scr
	000190	ascii	db200190.asc
	000190	script	db200190.scr

- 100 페이지의 『Quintana 이력서』는 직원 Delores Quintana의 이력서입니다.
- 102 페이지의 『Nicholls 이력서』는 직원 Heather Nicholls의 이력서입니다.
- 104 페이지의 『Adamson 이력서』는 직원 Bruce Adamson의 이력서입니다.
- 106 페이지의 『Walker 이력서』는 직원 James Walker의 이력서입니다.

IN_TRAY 테이블

이름:	RECEIVED	SOURCE	SUBJECT	NOTE_TEXT
유형:	timestamp	char(8)	char(64)	varchar(3000)
설명:	접수일 및 시간	메모 송신 직원의 사용자 ID	간략한 설명	메모

ORG 테이블

이름:	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
유형:	smallint not null	varchar(14)	smallint	varchar(10)	varchar(13)
설명:	부서 번호	부서 이름	관리자 번호	부서	도시
값:	10	Head Office	160	Corporate	New York
	15	New England	50	Eastern	Boston
	20	Mid Atlantic	10	Eastern	Washington
	38	South Atlantic	30	Eastern	Atlanta
	42	Great Lakes	100	Midwest	Chicago
	51	Plains	140	Midwest	Dallas
	66	Pacific	270	Western	San Francisco
	84	Mountain	290	Western	Denver

PROJECT 테이블

이름:	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
유형:	char(6) not null	varchar(24) not null	char(3) not null	char(6) not null	dec(5,2)	date	date	char(6)
설명:	프로젝트 번호	프로젝트 이름	담당 부서	담당 직원	예상되는 평균 소요 인원	예상 시작일	예상 종료일	프로젝트인 경 우 주 프로젝트
값:	AD3100	ADMIN SERVICES	D01	000010	6.5	1982-01-01	1983-02-01	-
	AD3110	GENERAL ADMIN SYSTEMS	D21	000070	6	1982-01-01	1983-02-01	AD3100
	AD3111	PAYROLL PROGRAMMING	D21	000230	2	1982-01-01	1983-02-01	AD3110
	AD3112	PERSONNEL PROGRAMMING	D21	000250	1	1982-01-01	1983-02-01	AD3110
	AD3113	ACCOUNT PROGRAMMING	D21	000270	2	1982-01-01	1983-02-01	AD3110
	IF1000	QUERY SERVICES	C01	000030	2	1982-01-01	1983-02-01	-
	IF2000	USER EDUCATION	C01	000030	1	1982-01-01	1983-02-01	-
	MA2100	WELD LINE AUTOMATION	D01	000010	12	1982-01-01	1983-02-01	-
	MA2110	W L PROGRAMMING	D11	000060	9	1982-01-01	1983-02-01	MA2100
	MA2111	W L PROGRAM DESIGN	D11	000220	2	1982-01-01	1982-12-01	MA2110
	MA2112	W L ROBOT DESIGN	D11	000150	3	1982-01-01	1982-12-01	MA2110
	MA2113	W L PROD CONT PROGS	D11	000160	3	1982-02-15	1982-12-01	MA2110
	OP1000	OPERATION SUPPORT	E01	000050	6	1982-01-01	1983-02-01	-
	OP1010	OPERATION	E11	000090	5	1982-01-01	1983-02-01	OP1000
	OP2000	GEN SYSTEMS SERVICES	E01	000050	5	1982-01-01	1983-02-01	-
	OP2010	SYSTEMS SUPPORT	E21	000100	4	1982-01-01	1983-02-01	OP2000
	OP2011	SCP SYSTEMS SUPPORT	E21	000320	1	1982-01-01	1983-02-01	OP2010
	OP2012	APPLICATIONSE21 SUPPORT	E21	000330	1	1982-01-01	1983-02-01	OP2010

샘플 데이터베이스 테이블

이름:	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
	OP2013	DB/DC SUPPORT	E21	000340	1	1982-01-01	1983-02-01	OP2010
	PL2100	WELD LINE PLANNING	B01	000020	1	1982-01-01	1982-09-15	MA2100

SALES 테이블

이름:	SALES_DATE	SALES_PERSON	REGION	SALES
유형:	date	varchar(15)	varchar(15)	int
설명:	판매일	판매 직원의 성	판매 지역	판매 수량
값:	12/31/1995	LUCCHESI	Ontario-South	1
	12/31/1995	LEE	Ontario-South	3
	12/31/1995	LEE	Quebec	1
	12/31/1995	LEE	Manitoba	2
	12/31/1995	GOUNOT	Quebec	1
	03/29/1996	LUCCHESI	Ontario-South	3
	03/29/1996	LUCCHESI	Quebec	1
	03/29/1996	LEE	Ontario-South	2
	03/29/1996	LEE	Ontario-North	2
	03/29/1996	LEE	Quebec	3
	03/29/1996	LEE	Manitoba	5
	03/29/1996	GOUNOT	Ontario-South	3
	03/29/1996	GOUNOT	Quebec	1
	03/29/1996	GOUNOT	Manitoba	7
	03/30/1996	LUCCHESI	Ontario-South	1
	03/30/1996	LUCCHESI	Quebec	2
	03/30/1996	LUCCHESI	Manitoba	1
	03/30/1996	LEE	Ontario-South	7
	03/30/1996	LEE	Ontario-North	3
	03/30/1996	LEE	Quebec	7
	03/30/1996	LEE	Manitoba	4
	03/30/1996	GOUNOT	Ontario-South	2
	03/30/1996	GOUNOT	Quebec	18
	03/30/1996	GOUNOT	Manitoba	1
	03/31/1996	LUCCHESI	Manitoba	1
	03/31/1996	LEE	Ontario-South	14
	03/31/1996	LEE	Ontario-North	3
	03/31/1996	LEE	Quebec	7
	03/31/1996	LEE	Manitoba	3
	03/31/1996	GOUNOT	Ontario-South	2
	03/31/1996	GOUNOT	Quebec	1

이름:	SALES_DATE	SALES_PERSON	REGION	SALES
	04/01/1996	LUCCHESSE	Ontario-South	3
	04/01/1996	LUCCHESSE	Manitoba	1
	04/01/1996	LEE	Ontario-South	8
	04/01/1996	LEE	Ontario-North	-
	04/01/1996	LEE	Quebec	8
	04/01/1996	LEE	Manitoba	9
	04/01/1996	GOUNOT	Ontario-South	3
	04/01/1996	GOUNOT	Ontario-North	1
	04/01/1996	GOUNOT	Quebec	3
	04/01/1996	GOUNOT	Manitoba	7

STAFF 테이블

이름:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
유형:	smallint not null	varchar(9)	smallint	char(5)	smallint	dec(7,2)	dec(7,2)
설명:	직원 번호	직원 이름	부서 번호	직함	근무 연수	현재 봉급	수당
값:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marengchi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-

샘플 데이터베이스 테이블

이름:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	220	Smith	51	Sales	7	17654.50	992.80
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

STAFFG 테이블

주: STAFFG는 2바이트 코드 페이지에서만 작성됩니다.

이름:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
유형:	smallint not null	vargraphic(9)	smallint	graphic(5)	smallint	dec(9,0)	dec(9,0)
설명:	직원 번호	직원 이름	부서 번호	직함	근무 연수	현재 봉급	수당
값:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marengi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-

이름:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-
	220	Smith	51	Sales	7	17654.50	992.80
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

BLOB 및 CLOB 데이터 유형으로 된 샘플 파일

다음 절에는 EM_PHOTO 파일(직원 사진) 및 EMP_RESUME 파일(직원 이력서)의 데이터가 나와 있습니다.

Quintana 사진



그림 7. *Delores M. Quintana*

Quintana 이력서

다음 텍스트는 db200130.asc 및 db200130.scr 파일에 있습니다.

Resume: Delores M. Quintana

Personal Information

Address:	1150 Eglinton Ave Mellonville, Idaho 83725
Phone:	(208) 555-9933
Birthdate:	September 15, 1925
Sex:	Female
Marital Status:	Married
Height:	5'2"
Weight:	120 lbs.

Department Information

Employee Number:	000130
Dept Number:	C01
Manager:	Sally Kwan
Position:	Analyst
Phone:	(208) 555-4578

Hire Date: 1971-07-28

Education

1965 Math and English, B.A. Adelphi University

1960 Dental Technician Florida Institute of Technology

Work History

10/91 - present Advisory Systems Analyst Producing documentation tools for engineering department.

12/85 - 9/91 Technical Writer Writer, text programmer, and planner.

1/79 - 11/85 COBOL Payroll Programmer Writing payroll programs for a diesel fuel company.

Interests

- Cooking
- Reading
- Sewing
- Remodeling

Nicholls 사진



그림 8. Heather A. Nicholls

Nicholls 이력서

다음 텍스트는 db200140.asc 및 db200140.scr 파일에 있습니다.

Resume: Heather A. Nicholls

Personal Information

Address:	844 Don Mills Ave Mellonville, Idaho 83734
Phone:	(208) 555-2310
Birthdate:	January 19, 1946
Sex:	Female
Marital Status:	Single
Height:	5'8"
Weight:	130 lbs.

Department Information

Employee Number:	000140
Dept Number:	C01
Manager:	Sally Kwan

Position: Analyst
Phone: (208) 555-1793
Hire Date: 1976-12-15

Education

1972 Computer Engineering, Ph.D. University of Washington

1969 Music and Physics, M.A. Vassar College

Work History

2/83 - present Architect, OCR Development Designing the architecture of OCR products.

12/76 - 1/83 Text Programmer Optical character recognition (OCR) programming in PL/I.

9/72 - 11/76 Punch Card Quality Analyst Checking punch cards met quality specifications.

Interests

- Model railroading
- Interior decorating
- Embroidery
- Knitting

Adamson 사진

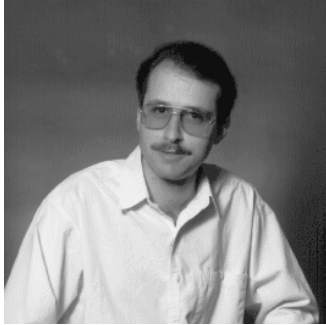


그림 9. Bruce Adamson

Adamson 이력서

다음 텍스트는 db200150.asc 및 db200150.scr 파일에 있습니다.

Resume: Bruce Adamson

Personal Information

Address:	3600 Steeles Ave Mellonville, Idaho 83757
Phone:	(208) 555-4489
Birthdate:	May 17, 1947
Sex:	Male
Marital Status:	Married
Height:	6'0"
Weight:	175 lbs.

Department Information

Employee Number:	000150
Dept Number:	D11
Manager:	Irving Stern

Position: Designer
Phone: (208) 555-4510
Hire Date: 1972-02-12

Education

1971 Environmental Engineering, M.Sc. Johns Hopkins University
1968 American History, B.A. Northwestern University

Work History

8/79 - present Neural Network Design Developing neural networks for machine intelligence products.
2/72 - 7/79 Robot Vision Development Developing rule-based systems to emulate sight.
9/71 - 1/72 Numerical Integration Specialist Helping bank systems communicate with each other.

Interests

- Racing motorcycles
- Building loudspeakers
- Assembling personal computers
- Sketching

Walker 사진

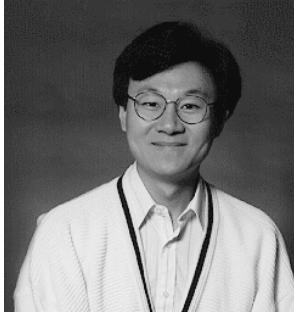


그림 10. James H. Walker

Walker 이력서

다음 텍스트는 db200190.asc 및 db200190.scr 파일에 있습니다.

Resume: James H. Walker

Personal Information

Address:	3500 Steeles Ave Mellonville, Idaho 83757
Phone:	(208) 555-7325
Birthdate:	June 25, 1952
Sex:	Male
Marital Status:	Single
Height:	5'11"
Weight:	166 lbs.

Department Information

Employee Number:	000190
Dept Number:	D11
Manager:	Irving Stern
Position:	Designer

Phone: (208) 555-2986

Hire Date: 1974-07-26

Education

1974 Computer Studies, B.Sc. University of Massachusetts

1972 Linguistic Anthropology, B.A. University of Toronto

Work History

6/87 - present Microcode Design Optimizing algorithms for mathematical functions.

4/77 - 5/87 Printer Technical Support Installing and supporting laser printers.

9/74 - 3/77 Maintenance Programming Patching assembly language compiler for mainframes.

Interests

- Wine tasting
- Skiing
- Swimming
- Dancing

샘플 데이터베이스 테이블

부록B. DB2 라이브러리 사용

DB2 Universal Database 라이브러리는 온라인 도움말, 책(PDF 및 HTML) 및 샘플 프로그램이 HTML 형식으로 구성됩니다. 이 절에서는 제공되는 정보 및 액세스하는 방법을 설명합니다.

제품 정보에 온라인으로 액세스하려면, 정보 센터를 이용할 수 있습니다. 124 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오. 웹에서 타스크 정보, DB2 책, 문제점 해결 정보, 샘플 프로그램 및 DB2 정보를 열람할 수 있습니다.

DB2 PDF 파일 및 인쇄된 책

DB2 정보

다음의 테이블은 DB2 책을 4개의 범주로 나눕니다.

DB2 안내 및 참조 정보

이 책에는 모든 플랫폼에 공통적인 DB2 정보가 들어 있습니다.

DB2 설치 및 구성 정보

이 책에는 특정 플랫폼의 DB2를 위한 것입니다. 예를 들어, OS/2, Windows 및 UNIX 플랫폼에서의 DB2용으로 각각 다른 빠른 시작 책이 있습니다.

플랫폼간 샘플 프로그램(HTML)

이 샘플들은 응용프로그램 개발 클라이언트와 함께 설치된 샘플 프로그램의 HTML 버전입니다. 이들은 단지 정보용으로서 실제 프로그램을 대체하지는 않습니다.

릴리스 정보

이러한 파일에는 DB2 책에 포함될 수 없었던 최신 정보가 포함되어 있습니다.

설치 매뉴얼, 릴리스 정보 및 지습서는 제품 CD-ROM의 HTML 디렉토리에서 볼 수 있습니다. 대부분의 책은 단지 보기용으로 제품 CD-ROM에서 HTML 형식으로 제공되고 보기와 인쇄용으로 제품 CD-ROM에서 PDF 형식으로 제공됩니다. 또한 IBM에서 인쇄된 책을 주문하려면 119 페이지의 『인쇄된 책 주문』에서 자세한 내용을 참조하십시오. 다음 테이블에는 주문할 수 있는 책을 보여줍니다.

OS/2 및 Windows 플랫폼에서는 `sqllib\doc\html` 디렉토리에 HTML 파일을 설치할 수 있습니다. DB2 정보는 여러 나라 언어로 번역되었습니다. 하지만, 모든 정보가 모든 나라의 언어로 번역된 것은 아닙니다. 정보가 특정 나라의 언어로 사용할 수 없을 경우에는 영문으로 제공됩니다.

UNIX 플랫폼에서는 `doc/%L/html` 디렉토리에 여러 나라 언어 버전의 HTML 파일을 설치할 수 있습니다. 여기서 `%L`은 해당 언어의 로케일을 나타냅니다. 빠른 시작 책에서 보다 자세한 내용을 참조하십시오.

다음의 여러 가지 방법으로 DB2 책을 구하고 정보를 액세스할 수 있습니다.

- 123 페이지의 『정보 온라인 보기』
- 127 페이지의 『정보 온라인 검색』
- 119 페이지의 『인쇄된 책 주문』
- 119 페이지의 『PDF 책 인쇄』

표 1. DB2 정보

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
DB2 안내 및 참조 정보			
관리 안내서	<p>관리 안내서: 계획에서는 데이터베이스의 개념에 대한 개요, 논리적 또는 물리적인 데이터베이스 설계와 같은 설계에 대한 정보 그리고 고가용성에 대한 정보를 제공합니다.</p> <p>관리 안내서: 구현에서는 사용자의 설계, 데이터베이스 액세스, 감사, 백업 및 복구와 같은 구현에 대한 정보를 제공합니다.</p> <p>관리 안내서: 성능에서는 데이터베이스의 환경, 응용프로그램 성능 평가 및 성능 조정에 대한 정보를 제공합니다.</p> <p>사용자는 문서 번호 SBOF-8934를 사용하여 세 권으로 된 관리 안내서 책을 주문할 수 있습니다.</p>	SA30-0990 db2d1x70 SA30-0988 db2d2x70 SA30-0989 db2d3x70	db2d0
Administrative API Reference	데이터베이스를 관리하는 데 사용할 수 있는 DB2 API와 데이터 구조에 대해 설명합니다. 또한 응용프로그램에서 API를 호출하는 방법을 설명합니다.	SC09-2947 db2b0x70	db2b0
응용프로그램 빌드 안내서	환경 설정 정보와 Windows에서 DB2 응용프로그램을 컴파일, 링크 및 수행하기 위한 지침이 단계별로 제공되어 있습니다.	SA30-0991 db2axx70	db2ax
APPC, CPI-C, and SNA Sense Codes	DB2 Universal Database 제품을 사용할 때 발생할 수 있는 APPC, CPI-C 및 SNA 센스 코드에 관한 일반 정보를 제공합니다. HTML 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2apx70	db2ap
응용프로그램 개발 안내서	Embedded SQL 또는 Java(JDBC 및 SQLJ)를 사용하여 DB2 데이터베이스를 액세스하는 응용프로그램을 개발하는 방법을 설명합니다. 저장 프로시저어 작성, 사용자 정의 함수 작성, 사용자 정의 유형 작성, 트리거 사용, 파티션된 환경 또는 연합 시스템에서 응용프로그램을 개발하는 등의 다양한 주제가 다루어집니다.	SA30-0992 db2a0x70	db2a0

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>CLI Guide and Reference</i>	DB2 콜 레벨 인터페이스와 Microsoft ODBC 스펙과 호환 가능한 호출 가능 SQL 인터페이스를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램의 개발 방법에 대해 설명합니다.	SC09-2950 db210x70	db210
<i>Command Reference</i>	명령행 프로세서를 사용하는 방법을 설명하고 데이터베이스를 관리하기 위해 사용할 수 있는 DB2 명령을 설명합니다.	SC09-2951 db2n0x70	db2n0
연결성 보충 설명서	AS/400용 DB2, OS/390용 DB2, MVS용 DB2 또는 VM용 DB2를 DB2 Universal Database 서버와의 DRDA 응용프로그램 리퀘스터로 사용하는 방법에 대한 참조 정보 및 설치 정보를 제공합니다. 또한 DB2 Connect AR(응용프로그램 리퀘스터)과 함께 DRDA AS(응용프로그램 서버)를 사용하는 방법에 대해서도 상세히 설명합니다. HTML 및 PDF 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2h1x70	db2h1
데이터 이동 유틸리티 안내 및 참조서	Import, Export, Load, AutoLoader 및 DPROP와 같이 데이터 이동을 용이하게 해 주는 DB2 UDB 유틸리티의 사용 방법에 대해 설명합니다.	SA30-0994 db2dmx70	db2dm
<i>Data Warehouse Center</i> 관리 안내서	Data Warehouse Center를 사용하여 데이터 웨어하우스를 구축 및 유지보수하는 방법을 제공합니다.	SA30-1000 db2ddx70	db2dd
<i>Data Warehouse Center</i> 응용프로그램 통합 안내서	프로그래머들이 Data Warehouse Center 및 Information Catalog Manager를 응용프로그램과 통합하는 데 도움을 주는 정보를 제공합니다.	SA30-1001 db2adx70	db2ad
<i>DB2 Connect</i> 사용자 안내서	DB2 Connect 제품에 대한 개념, 프로그래밍 및 일반 사용 정보를 제공합니다.	SA30-0993 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	DB2 Query Patroller 시스템의 조작 개요, 특정 조작 및 관리 정보, 관리 그래픽 사용자 인터페이스 유틸리티에 대한 타스크 정보를 제공합니다.	SC09-2958 db2dwx70	db2dw

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Query Patroller User's Guide</i>	DB2 Query Patroller의 도구 및 함수를 사용하는 방법을 설명합니다.	SC09-2960	db2ww
		db2wwx70	
용어집	DB2에서 사용되는 용어와 그 구성요소에 대한 정의를 제공합니다.	문서 번호가 없습니다.	db2t0
	HTML 형식과 SQL 참조서에서 사용할 수 있습니다.	db2t0x70	
<i>Image, Audio 및 Video Extenders</i> 관리 및 프로그래밍	DB2 Extender에 대한 일반적인 정보와 이미지, 오디오 및 비디오(IAV)의 관리 및 구성에 대한 정보 그리고 IAV extenders를 사용한 프로그램에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1043	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	정보 카탈로그 관리에 대한 지침을 제공합니다.	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	Information Catalog Manager에 대한 아키텍처 인터페이스에 대한 정의를 제공합니다.	SC26-9997	db2bi
		db2bix70	
<i>Information Catalog Manager 사용자 안내서</i>	Information Catalog Manager 사용자 인터페이스 사용에 대한 정보를 제공합니다.	SA30-1002	db2ai
		db2aix70	
설치 및 구성 보충 설명서	플랫폼 특정 DB2 클라이언트의 플랜, 설치 및 설정에 대해 설명합니다. 또한 바인딩, 클라이언트 및 서버 통신의 설정, DB2 GUI 도구, DRDA AS, 분산 설치 및 이중 데이터 소스에 대한 분산 요구와 액세스 방식의 구성에 대한 정보가 들어 있습니다.	GA30-0975	db2iy
		db2iyx70	
메시지 참조서	DB2, Information Catalog Manager 및 Data Warehouse Center에서 발행하는 메시지와 코드를 나열하고 수행해야 할 조치에 대해 설명합니다.	볼륨 1 GA30-0986	db2m0
		db2m1x70	
	문서 번호(SBOF-8932)를 사용하여 두 권으로 된 메시지 참조서 책을 모두 주문할 수 있습니다.	볼륨 2 GA30-0987	
		db2m2x70	

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>OLAP Integration Server Administration Guide</i>	OLAP 통합 서버의 관리 프로그램 구성요소를 사용하는 방법을 설명합니다.	SC27-0787	n/a
		db2dpx70	
<i>OLAP Integration Server Metaoutline User's Guide</i>	표준 OLAP Metaoutline 인터페이스 (Metaoutline Assistant가 아닌)를 사용하여 OLAP Metaoutlines을 작성하고 사용하는 방법을 설명합니다.	SC27-0784	n/a
		db2upx70	
<i>OLAP Integration Server Model User's Guide</i>	표준 OLAP 모델 인터페이스(Model Assistant가 아닌)를 사용하여 OLAP 모델을 작성하는 방법을 설명합니다.	SC27-0783	n/a
		db2lpx70	
<i>OLAP 설치 및 사용자 안내서</i>	OLAP Starter 키트에 대한 구성 및 설치 정보를 제공합니다.	SA30-1074	db2ip
		db2ipx70	
<i>Excel용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Excel 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0564	db2ep
		db2epx70	
<i>Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Lotus 1-2-3 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0565	db2tp
		db2tpx70	
<i>복제 안내 및 참조서</i>	DB2와 함께 제공된 IBM 복제 도구에 관한 플랜, 구성, 관리 및 사용 정보를 제공합니다.	SA30-1003	db2e0
		db2e0x70	
<i>Spatial Extender 사용자 안내 및 참조서</i>	Spatial Extender 설치, 구성, 관리, 프로그래밍 및 문제 해결에 대한 정보를 제공합니다. 또한 공간 데이터 개념에 대한 설명을 제공하고 Spatial Extender에만 고유하게 적용되는 참조 정보(메시지 및 SQL)를 제공합니다.	SA30-1045	db2sb
		db2sbx70	
<i>SQL 시작하기</i>	SQL 개념을 소개하고, 많은 구조와 타스크에 관한 예를 보여줍니다.	SA30-0996	db2y0
		db2y0x70	
<i>SQL 참조서, 볼륨 1 및 볼륨 2</i>	SQL 구문, 의미 그리고 언어 규칙에 대해 설명합니다. 또한 릴리스 간 비호환성, 제품 제한 사항 및 카탈로그 뷰에 대한 정보도 들어 있습니다.	볼륨 1 SA30-0997	db2s0
		db2s1x70	
	SBOF-8933 문서 번호를 사용하여 SQL 참조서를 주문할 수 있습니다.	볼륨 2 SA30-0998	
		db2s2x70	

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
시스템 모니터 안내 및 참조 서	데이터베이스와 데이터베이스 관리 프로그램에 관한 여러 종류의 정보를 수집하는 방법에 대해 설명합니다. 이 책은 데이터베이스 활동을 이해하고, 성능을 향상시키고, 문제점의 원인을 판별하기 위한 정보를 사용하는 방법을 설명합니다.	SA30-0995	db2f0
		db2f0x70	
Text Extender 관리 및 프로그래밍	DB2 extenders에 관한 일반적인 정보와 Text extenders 관리 및 구성에 관한 정보, Text extenders를 사용한 프로그래밍에 관한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1044	desu9
		desu9x70	
문제점 해결 안내서	오류의 출처를 판별하고 문제점으로부터 회복하고, DB2 고객 서비스와 상담하여 진단 도구를 사용하는 것을 도와줍니다.	GA30-0704	db2p0
		db2p0x70	
새로운 기능	DB2 Universal Database, 버전 7의 새로운 특성, 기능 및 향상된 내용을 설명합니다.	SA30-0999	db2q0
		db2q0x70	
DB2 설치 및 구성 정보			
OS/2 및 Windows용 DB2 Connect Enterprise Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Enterprise Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0974	db2c6
		db2c6x70	
UNIX용 DB2 Connect Enterprise Edition 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Connect Enterprise Edition에 대한 플랜, 이주, 설치, 구성 및 타스크 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0973	db2cy
		db2cyx70	
DB2 Connect Personal Edition 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 모든 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0981	db2c1
		db2c1x70	
DB2 Connect Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2962	db2c4
		db2c4x70	

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Data Links Manager</i> 빠른 시작	AIX 및 Windows 32 비트 운영 체제용 DB2 Data Links Manager에 대한 플랜, 설치, 구성 및 타스크 정보를 제공합니다.	GA30-0980 db2z6x70	db2z6
<i>UNIX용 DB2 Enterprise - Extended Edition</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Enterprise - Extended Edition 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0978 db2v3x70	db2v3
<i>Windows용 DB2 Enterprise - Extended Edition</i> 빠른 시작	Windows 32 비트 운영 체제용 DB2 Enterprise - Extended Edition에 관한 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0977 db2v6x70	db2v6
<i>OS/2용 DB2</i> 빠른 시작	OS/2 운영 체제에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0982 db2i2x70	db2i2
<i>UNIX용 DB2</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0984 db2ixx70	db2ix
<i>Windows용 DB2</i> 빠른 시작	Windows 32 비트 운영 체제에서 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0985 db2i6x70	db2i6
<i>DB2 Personal Edition</i> 빠른 시작	OS/2 및 Windows 32 비트 운영 체제에서의 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GA30-0983 db2i1x70	db2i1
DB2 Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux에서 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller</i> 설치 안내서	DB2 Query Patroller에 관한 설치 정보를 제공합니다.	GA30-0976 db2iw70	db2iw

표 1. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
DB2 Warehouse Manager 설치 안내서	웨어하우스 에이전트, 웨어하우스 변환기 및 Information Catalog Manager에 관한 설치 정보를 제공합니다.	GA30-1027 db2idx70	db2id
플랫폼간 샘플 프로그램(HTML)			
샘플 프로그램(HTML)	DB2가 지원하는 모든 플랫폼에서 프로그래밍 언어에 대한 샘플 프로그램이 HTML 형식으로 제공됩니다. 이 샘플 프로그램은 정보용으로만 제공됩니다. 모든 샘플을 모든 프로그래밍 언어로 사용할 수 있는 것은 아닙니다. HTML 샘플은 DB2 응용프로그램 개발 클라이언트가 설치될 때에 사용할 수 있습니다.	문서 번호가 없습니다.	db2hs
릴리스 정보			
<i>DB2 Connect</i> 릴리스 정보	DB2 Connect 책에는 포함될 수 없었던 최신 정보를 제공합니다.	#2를 참조하십시오.	db2cr
<i>DB2</i> 설치 정보	DB2 책에는 포함될 수 없었던 최신 설치 정보를 제공합니다.	제품 CD-ROM에서만 사용할 수 있습니다.	
<i>DB2</i> 릴리스 정보	DB2 책에는 포함될 수 없었던 모든 DB2 제품 및 기능에 대한 최신 정보를 제공합니다.	#2를 참조하십시오.	db2ir

주:

1. 파일 이름의 6번째 자리에 있는 문자 *x*는 책의 언어 버전을 나타냅니다. 예를 들면, 파일 이름 db2d0e70은 관리 안내서 책의 영문 버전을 나타내며 db2d0k70은 같은 책의 한글 버전을 나타냅니다. 다음 문자는 언어 버전을 나타내기 위해 파일 이름의 6번째 자리에 사용됩니다.

언어	식별자
브라질 포르투갈어	b
불가리아어	u
체코어	x
덴마크어	d
네덜란드어	q
영어	e

핀란드어	y
프랑스어	f
독일어	g
그리스어	a
헝가리어	h
이탈리아어	i
일본어	j
한글	k
노르웨이어	n
폴란드어	p
포르투갈어	v
러시아어	r
중국어	c
슬로베니아어	l
스페인어	z
스웨덴어	s
대만어	t
터키어	m

2. DB2 책에 포함되어 있지 않을 수 있는 최신 정보는 릴리스 정보에서 HTML 형식과 ASCII 파일로 사용할 수 있습니다. HTML 버전은 정보 센터와 제품 CD-ROM에서 사용할 수 있습니다. ASCII 파일을 보려면,

- UNIX 기반 플랫폼의 경우에는 Release.Notes 파일을 참조하십시오. 이 파일은 DB2DIR/Readme/%L 디렉토리에 있으며 여기서, %L은 로케일 이름이고 DB2DIR은 다음과 같습니다.
 - AIX에서는 /usr/lpp/db2_07_01
 - HP-UX, PTX, Solaris 및 Silicon Graphics IRIX에서는 /opt/IBMdb2/V7.1
 - Linux에서는 /usr/IBMdb2/V7.1
- 다른 플랫폼의 경우에는 RELEASE.TXT 파일을 참조하십시오. 이 파일은 제품이 설치된 디렉토리에 있습니다. OS/2 플랫폼에서는 IBM DB2 폴더를 더블 클릭하고 릴리스 정보 아이콘을 더블 클릭할 수 있습니다.

PDF 책 인쇄

책의 사본을 원하는 경우 DB2 책 CD-ROM에 있는 PDF 파일을 인쇄할 수 있습니다. Adobe Acrobat Reader를 사용하여 책 전체나 특정 페이지를 인쇄할 수 있습니다. 라이브러리에 있는 각 책의 파일 이름에 대해서는 111 페이지의 표1에서 자세한 내용을 참조하십시오.

Adobe 웹 사이트인 <http://www.adobe.com>에서 Adobe Acrobat Reader의 최신 버전을 얻을 수 있습니다.

PDF 파일은 파일 확장자가 PDF로서 DB2 책 CD-ROM에 들어 있습니다. PDF 파일을 액세스하려면,

1. DB2 책 CD-ROM을 삽입하십시오. UNIX 기반의 플랫폼에서는 DB2 책 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 *빠른 시작* 책에서 자세한 내용을 참조하십시오.
2. Acrobat Reader를 시작하십시오.
3. 다음 위치에서 원하는 PDF 파일을 여십시오.
 - OS/2 및 Windows 플랫폼에서:
x:\doc\language 디렉토리. 여기서 *x*는 CD-ROM 드라이브를 나타내며 *language*는 사용자 언어를 나타내는 2문자 국가 코드를 나타냅니다. 예를 들면 영문인 경우에는 EN입니다.
 - UNIX 기반 플랫폼에서:
/cdrom/doc/%L 디렉토리. 여기서 */cdrom*은 CD-ROM의 마운트 위치이고 *%L*은 원하는 로케일의 이름입니다.

또한 PDF 파일을 CD-ROM에서 지역이나 네트워크로 파일을 복사하고 거기서 읽을 수도 있습니다.

인쇄된 책 주문

인쇄된 DB2 책은 책 주문 번호(SBOF)를 사용하여 세트나 날권으로 주문할 수 있습니다. 인쇄본을 주문하려면, IBM 협력업체 또는 영업 대표에게 문의하십시오. 또한 웹 사이트 <http://www.elink.ibm.com/pbl/pbl>에서도 책을 주문할 수 있습니다.

두 종류의 책 세트를 사용할 수 있습니다. SBOF-8935는 DB2 Warehouse Manager에 대한 참조 및 사용에 관한 정보를 제공합니다. SBOF-8931은 다른 모든 DB2 Universal Database 제품과 특징에 대한 참조 및 사용 정보를 제공합니다. 각 SBOF의 내용은 다음 테이블에 나열되어 있습니다.

표 2. 인쇄된 책 주문

SBOF 번호	포함된 책
SBOF-8931	<ul style="list-style-type: none"> • 관리 안내서: 계획 • 관리 안내서: 구현 • 관리 안내서: 성능 • Administrative API Reference • 응용프로그램 빌드 안내서 • 응용프로그램 개발 안내서 • CLI Guide and Reference • Command Reference • 데이터 이동 유틸리티 안내 및 참조서 • Data Warehouse Center 관리 안내서 • Data Warehouse Center 응용프로그램 통합 안내서 • DB2 Connect 사용자 안내서 • 설치 및 구성 보충 설명서 • Image, Audio 및 Video Extenders 관리 및 프로그래밍 • 메시지 참조서, 볼륨 1 및 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP 설정 및 사용자 안내서 • Excel용 OLAP Spreadsheet Add-in 사용자 안내서 • Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서 • 복제 안내 및 참조서 • Spatial Extender 관리 및 프로그래밍 안내서 • SQL 시작하기 • SQL 참조서, 볼륨 1 및 2 • 시스템 모니터 안내 및 참조서 • Text Extender 관리 및 프로그래밍 • 문제점 해결 안내서 • 새로운 기능
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager 사용자 안내서 • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 온라인 문서

온라인 도움말 액세스

온라인 도움말은 모든 DB2 구성요소에서 사용할 수 있습니다. 다음의 테이블에서는 다양한 도움말 유형을 설명합니다.

도움말의 유형	내용	액세스하는 방법
명령 도움말	명령행 처리기 내의 명령 구문을 설명합니다.	대화식 모드인 명령행 처리기에서, 다음을 입력하십시오. <code>? command</code> 여기서, <i>command</i> 는 키워드이거나 전체 명령입니다. 예를 들어, <code>? catalog</code> 는 모든 CATALOG 명령에 대한 도움말을 표시하고, <code>? catalog database</code> 는 CATALOG DATABASE 명령에 대한 도움말을 표시합니다.
클라이언트 구성 지원 프로그램 도움말	창 또는 노트북에서 수행할 수 있는 작업을 설명합니다. 도움말은 알아야 할 개요와 전체조건 정보를 포함하고, 창 또는 노트북 제어를 사용하는 방법을 설명합니다.	창이나 노트북에서, 도움말 버튼을 누르거나 F1 키를 누르십시오.
명령 센터 도움말		
제어 센터 도움말		
Data Warehouse Center 도움말		
이벤트 분석기 도움말		
Information Catalog Manager 도움말		
위성 관리 센터 도움말		
스크립트 센터 도움말		

도움말의 유형	내용	액세스하는 방법
메시지 도움말	메시지의 원인과 사용자가 취해야 할 조치를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn</code></p> <p>여기서, <code>XXXnnnnn</code>은 유효한 메시지 식별자입니다.</p> <p>예를 들어, <code>? SQL30081</code>은 <code>SQL30081</code> 메시지에 대한 도움말을 표시합니다.</p> <p>한 번에 한 화면씩 메시지 도움말을 보려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn more</code></p> <p>파일에 메시지 도움말을 저장하려면, 다음을 입력하십시오.</p> <p><code>? XXXnnnnn > filename.ext</code></p> <p>여기서, <code>filename.ext</code>는 메시지 도움말을 저장하려는 파일입니다.</p>
SQL 도움말	SQL문의 구문을 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>help statement</code></p> <p>여기서, <code>statement</code>는 SQL문입니다.</p> <p>예를 들어, <code>help SELECT</code>는 <code>SELECT</code>문에 대한 도움말을 표시합니다.</p> <p>주: SQL 도움말은 UNIX 기반 플랫폼에서 사용할 수 없습니다.</p>
SQLSTATE 도움말	SQL 상태 및 클래스 코드를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <p><code>? sqlstate</code> 또는 <code>? class code</code></p> <p>여기서, <code>sqlstate</code>는 유효한 5자리 숫자로 된 SQL 상태이고 <code>class code</code>는 SQL 상태의 처음 2자리 숫자입니다.</p> <p>예를 들어, <code>? 08003</code>은 <code>08003</code> SQL 상태에 대한 도움말을 표시하고, <code>? 08</code>은 <code>08</code> 클래스 코드에 대한 도움말을 표시합니다.</p>

정보 온라인 보기

이 제품에 들어 있는 책은 HTML(Hypertext Markup Language) 소프트웨어 형식으로 제공됩니다. 소프트웨어는 정보를 검색할 수 있게 하고 관련된 정보로 링크하는 하이퍼텍스트를 제공합니다. 또한, 사이트에서 라이브러리를 공유하는 것도 더 쉬워집니다.

HTML 버전 3.2 스펙을 따르는 브라우저로 온라인 책 또는 샘플 프로그램을 볼 수 있습니다.

온라인 책 또는 샘플 프로그램을 보려면:

- DB2 관리 도구를 수행할 경우, 정보 센터를 사용하십시오.
- 브라우저에서, 파일 → 페이지 열기를 클릭하십시오. 열린 페이지에 DB2 정보에 대한 설명과 링크가 들어 있습니다.
 - UNIX 기반 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

여기서 %L은 로케일 이름입니다.

- 다른 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
sql1lib\doc\html\index.htm
```

이 경로는 DB2가 설치된 드라이브에 있습니다.

정보 센터를 설치하지 않은 경우, **DB2 정보** 아이콘을 더블 클릭하여 페이지를 열 수 있습니다. 사용하는 시스템에 따라, 주 제품 폴더나 Windows 시작 메뉴에 아이콘이 있습니다.

Netscape 브라우저 설치

웹 브라우저를 설치하지 않은 경우, 제품 상자에 있는 Netscape CD-ROM에서 Netscape를 설치할 수 있습니다. 설치하는 방법에 대해서는 다음을 수행하십시오.

1. Netscape CD-ROM을 삽입하십시오.
2. UNIX 기반의 플랫폼에서는 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 빠른 시작 책에서 자세한 내용을 참조하십시오.

3. 설치 지침서는 CDNAVnn.txt 파일을 참조하십시오. 여기서, nn은 2문자로 된 언어 식별자입니다. 파일은 CD-ROM의 루트 디렉토리에 있습니다.

정보 센터로 정보에 액세스

정보 센터는 DB2 제품 정보로의 빠른 액세스를 제공합니다. 정보 센터는 DB2 관리 도구를 사용할 수 있는 모든 플랫폼에서 사용할 수 있습니다.

정보 센터 아이콘을 더블 클릭하여 정보 센터를 열 수 있습니다. 사용하는 시스템에 따라 아이콘은 주 제품 폴더나 Windows 시작 메뉴의 정보 폴더에 있습니다.

또한 DB2 Windows 플랫폼에서 도구 모음이나 도움말 메뉴를 사용하여 정보 센터를 액세스할 수 있습니다.

정보 센터는 6개 유형의 정보를 제공합니다. 적당한 탭을 클릭하여 그 유형에서 지원하는 주제를 보십시오.

타스크 DB2를 사용하여 수행할 수 있는 키 타스크.

참조 키워드, 명령 및 API와 같은 DB2 참조 정보.

책 DB2 책.

문제점 해결 오류 메시지의 종류와 복구 조치.

샘플 프로그램 DB2 응용프로그램 개발 클라이언트와 함께 제공되는 샘플 프로그램. DB2 응용프로그램 개발 클라이언트를 설치하지 않은 경우, 이 탭은 표시되지 않습니다.

웹 월드 와이드 웹에서의 DB2 정보. 이 정보에 액세스하려면, 사용자의 시스템으로부터 웹으로의 연결이 있어야 합니다.

목록 중 하나에서 항목을 선택할 때, 정보 센터는 정보를 표시하기 위해 표시기를 시작합니다. 표시기는 사용자가 선택하는 정보의 종류에 따라, 시스템 도움말 표시기, 편집기 또는 웹브라우저가 될 수 있습니다.

정보 센터는 찾기 기능을 제공하므로 목록을 찾지 않고도 특정 주제를 찾을 수 있습니다.

전체 텍스트 검색을 위해서는 **DB2 온라인 정보 검색** 검색 양식으로 연결된 정보 센터의 하이퍼텍스트 링크를 따라 검색하십시오.

HTML 검색 서버는 보통 자동으로 시작됩니다. HTML 정보에서 검색 기능이 작동하지 않으면, 다음 방법 중 하나를 사용하여 검색 서버를 시작할 수 있습니다.

Windows의 경우:

시작을 클릭하고 프로그램 → IBM DB2 → 정보 → HTML 검색 서버 시작을 선택하십시오.

OS/2 경우:

OS/2용 DB2 폴더를 더블 클릭하고 HTML 검색 서버 시작 아이콘을 더블 클릭하십시오.

HTML 정보를 검색하면서 다른 문제가 생길 경우, 릴리스 정보를 참조하십시오.

주: 검색 기능은 Linux, PTX 및 Silicon Graphics IRIX 환경에서는 작동하지 않습니다.

DB2 마법사 사용

마법사는 한 번에 한 단계씩 각 작업을 수행하게 함으로써 특정 관리 작업을 완료하는 데 도움을 줍니다. 마법사는 제어 센터 및 클라이언트 구성 지원 프로그램을 통해 사용할 수 있습니다. 다음 테이블에서는 마법사를 나열하고 그 기능을 설명합니다.

주: 데이터베이스 작성, 색인 작성, 다중 사이트 갱신 구성 및 성능 구성 마법사는 파티션된 데이터베이스 환경에서 사용할 수 있습니다.

마법사	도움 대상	액세스하는 방법
데이터베이스 추가	클라이언트 워크스테이션의 데이터베이스를 카탈로그화합니다.	클라이언트 구성 지원 프로그램에서 추가를 클릭하십시오.
데이터베이스 백업	백업 계획을 결정하고, 작성하고, 일정을 세웁니다.	제어 센터에서 백업하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 백업 → 마법사를 사용한 데이터베이스 백업을 선택하십시오.
다중 사이트 갱신 구성	다중 사이트 갱신, 분선 트랜잭션 또는 2 단계 확약을 구성합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭하고 다중 사이트 갱신을 선택하십시오.

마법사	도움 대상	액세스하는 방법
데이터베이스 작성	데이터베이스를 작성한 다음, 몇 가지 기본적인 구성 작업을 수행합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭한 다음 작성 → 마법사를 사용한 데이터베이스 작성을 선택하십시오.
테이블 작성	기본 데이터 유형을 선택한 다음, 테이블에 대한 기본 키를 작성합니다.	제어 센터에서 테이블 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 테이블을 선택하십시오.
테이블 공간 작성	새로운 테이블 공간을 작성합니다.	제어 센터에서 테이블 공간 아이콘을 마우스의 오른쪽 버튼으로 선택하고 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
색인 작성	사용자의 모든 조회를 작성하고 삭제하기 위해 색인화합니다.	제어 센터에서 색인 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 색인을 선택하십시오.
성능 구성	업무 요구조건에 맞게 구성 매개변수를 갱신하여 데이터베이스의 성능을 조정합니다.	제어 센터에서 성능을 조정하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오. 파티션된 데이터베이스에 대해 데이터베이스 파티션 뷰로부터 성능을 조정하려는 첫번째 파티션을 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오.
데이터베이스 복원	실패 후에 데이터베이스를 복구합니다. 사용할 백업 위치 및 재작동할 로그 기록을 이해하는 데 도움을 줍니다.	제어 센터에서 복원하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.

문서 서버 설정

기본 값으로 DB2 정보는 지역 시스템에 설치됩니다. 이는 DB2 정보에 액세스해야 하는 모든 사람이 동일한 파일을 설치해야 함을 의미합니다. DB2 정보를 한 위치에 저장하려면, 다음과 같이 하십시오.

1. 지역 시스템의 `\sqllib\doc\html`에 있는 모든 파일과 서브디렉토리를 웹 서버로 복사하십시오. 각 책은 책을 구성하는 데 필요한 모든 HTML 및 GIF 파일이 들어 있는 서브디렉토리를 가집니다. 디렉토리 구조가 변경되지 않게 하십시오.

2. 새로운 위치에 있는 파일을 찾으려면 웹 서버를 구성하십시오. 보다 자세한 정보는 **설치 및 구성 보충 설명서**의 부록 NetQuestion을 참조하십시오.
3. Java 버전의 정보 센터를 이용하는 경우, 모든 HTML 파일에 대한 기본 URL을 지정할 수 있습니다. 책 목록에 대해서는 URL을 사용해야 합니다.
4. 책 파일을 열람할 수 있게 되면, 다음과 같이 자주 열람하는 주제 항목에 대해서는 북마크를 설정할 수 있습니다. 다음의 페이지들을 북마크로 설정해 두면 도움이 될 것입니다.
 - 책 목록
 - 자주 이용하는 책의 목차
 - ALTER TABLE 주제와 같은 자주 참조하는 항목
 - 검색 양식

DB2 Universal Database 온라인 문서 파일을 중앙 시스템에서 제공하는 방법에 대한 정보를 보려면 **설치 및 구성 보충 설명서**의 부록 NetQuestion을 참조하십시오.

정보 온라인 검색

HTML 파일에서 정보를 찾으려면, 다음 방법 중 하나를 사용하십시오.

- 맨 위 프레임에서 검색을 클릭하십시오. 특정 주제를 찾으려면 검색 양식을 사용하십시오. 이 기능은 Linux, PTX 또는 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.
- 맨 위 프레임에서 색인을 클릭하십시오. 책에서 특정 주제를 찾으려면 색인을 사용하십시오.
- 책에서 특정 주제를 찾으려면 목차나 도움말의 색인 또는 HTML 책을 표시하고 웹 브라우저의 찾기 기능을 사용하십시오.
- 특정 주제로 빨리 리턴하려면 웹 브라우저의 북마크 기능을 사용하십시오.
- 특정 주제를 찾으려면 정보 센터의 검색 기능을 사용하십시오. 124 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오.

부록C. 주의사항

IBM은 이 책에서 논의된 제품, 서비스 또는 기능을 다른 나라에서는 제공하지 않을 수 있습니다. 현재 사용자가 사용할 수 있는 제품 및 서비스에 대한 정보는 해당 지역의 IBM 영업대표에게 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스만을 사용해야 함을 의미하지는 않습니다. IBM의 지적 소유권을 침해하지 않는 기능상으로 동등한 타사의 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나, 타사 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대한 특허를 보유하고 있거나 출원중일 수 있습니다. 이 책을 제공한다고 해서 그러한 특허에 대한 사용권까지를 부여하는 것은 아닙니다. 특허 사용권에 대한 문의는 다음 주소로 하십시오.

135-270

서울특별시 강남구 도곡동 462-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
지적 재산권부

2바이트(DBCS) 정보에 관한 사용권 문의는 사용자 국가의 IBM 지적 재산권부나 다음 주소로 서면 문의하십시오.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 사항은 영국이나 이 조항이 현지법과 상충되는 나라에는 적용되지 않습니다. IBM에서는 이 책을 명시적 또는 암시적인 어떠한 종류의 보증없이 『있는 그대로』 제공하므로, 판매 가능성을 보장하거나 특정 목적에 적합한지 여부에 대해서는 책임질 수 없습니다. 일부 국가에서는 특정 거래의 명시적 또는 암시적인 보증을 부인하는 문장을 허용하지 않으므로, 이 사항이 사용자에게 적용되지 않을 수도 있습니다.

이 책에는 기술상 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 개정판에 통합됩니다. IBM은 사전 통지없이 언제든지 이 책에 설명된 제품과 프로그램을 개선 및 변경할 수 있습니다.

이 책에서 타사의 웹 사이트를 언급한 것은 단지 편의를 위해서일 뿐이며 이런 웹 사이트를 추천하려는 의도는 아닙니다. 이런 웹 사이트의 데이터가 이 IBM 제품에 대한 데이터의 일부는 아니므로 이런 웹 사이트 사용에 대한 책임은 사용자가 져야 합니다.

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

이 프로그램의 사용권자가 (i) 독립적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함한) 사이의 정보 교환과 (2) 교환된 정보의 공동 사용을 목적으로 그 프로그램에 대한 정보를 원하는 경우, 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

소프트웨어 사업부

이러한 정보는 특정한 기간 및 조건하에 사용가능하며 어떤 경우에는 사용료를 지불해야 합니다.

이 책에 기술된 사용권 프로그램 및 이 프로그램에 사용가능한 모든 사용권 데이터는 IBM 고객 협약, IBM 국제 프로그래밍 사용권 협약 또는 이와 동등한 모든 협약 조건하에 IBM에서 제공됩니다.

여기에 제시된 어떠한 성능 데이터는 주위 환경에 따라 결정될 수 있습니다. 따라서, 다른 운영 체제에서 제시된 결과 값과 다를 수 있습니다. 몇몇 측정값은 개발 단계에서 얻은 값일 수 있습니다. 따라서 일반적인 사용자 시스템에서 얻은 값과 다를 수 있습니다. 또한 몇몇 측정값은 보외법을 통해 측정된 값입니다. 실제 값과는 다를 수 있습니다. 이 책의 사용자는 사용자의 특정 환경에 맞게 적용가능한 데이터를 변경해야 합니다.

타사 제품과 관련된 정보는 해당 제품의 공급자, 공개 발표 또는 기타 공개적으로 사용가능한 소스에서 확보한 것입니다. IBM은 이들 제품을 검사하지 않았고 성능 상의 정확성, 호환성 또는 타사 제품과 관련된 기타 주장을 확인할 수 없습니다. 타사 제품의 성능에 관한 문제는 해당 제품의 공급자에게 제기되어야 합니다.

IBM이 제시하는 방향 또는 의도에 관한 어떠한 언급도 특별한 통지없이 변경될 수 있습니다.

이 정보는 일상적인 비즈니스 처리에 사용되는 데이터와 보고서의 예가 들어 있을 수 있습니다. 보다 구체적으로 예를 나타내기 위해 특정 개인, 회사, 상표 또는 제품 이름이 언급되는 경우가 있습니다. 여기서 언급된 이름은 가상의 이름이며 실제 비즈니스 업체가 사용하는 이름 및 주소와 유사하다면 우연인 것입니다.

사용권:

이 정보에는 여러 운영 체제에서 프로그래밍 소스 언어로 예제 응용프로그램이 들어 있을 수 있습니다. 사용자는 이들 예제 프로그램을 IBM에게 비용을 지급하지 않고 복사, 수정 및 분배할 수 있습니다. 이들 예제 프로그램은 모든 조건에서 철저히 검사되지 않았습니다. 따라서, IBM은 이들 프로그램에 대해 어떠한 보증도 할 수 없습니다.

이들 예제 프로그램의 각각의 복사본이나 특정 부분은 다음과 같은 사용권 주의 사항을 포함해야 합니다.

© (사용자 회사 이름) (년도). 이 코드의 일부는 IBM Corp. 예제 프로그램에서 발췌된 것입니다. © Copyright IBM Corp. (년도 입력). All rights reserved.

등록 상표

별표(*)로 표시된 다음의 용어는 전세계에서 IBM의 상표입니다.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

다음 용어는 해당 회사의 상표 또는 등록 상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록 상표입니다.

Java 또는 모든 Java 관련 상표 및 로고 그리고 Solaris는 전세계에서 Sun Microsystems, Inc.의 상표입니다.

Tivoli 및 NetView는 전세계에서 Tivoli Systems Inc.의 상표입니다.

UNIX는 전세계에서 X/Open Company Limited가 독점권을 갖는 등록 상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스 표시입니다.

색인

[가]

값
정의 3

검색
온라인 정보 124, 127

검색 조건 22

결과 테이블 3

결합, 조희 53

고유 키 62
고유한 제한조건 62

고유한 제한조건 62
정의 61

공통 곱 70

공통 테이블 표현식
설명 43

공통 테이블(cross-tabulation) 행 75

관계형 데이터베이스, 정의 1

관련 문서 v

관리 안내서 v

구별 데이터 유형 78

권한 부여 ID 4

그래픽 문자열
가변 길이 5
고정 길이 5

그룹화 칼럼, 정의 35

기본 키 62

기본 테이블 3

[나]

날짜 시간 값, 설명 5

내부 조인 70

널(Null) 값 53
칼럼 값 삭제 13

널(null) 값, 설명 5

[다]

다중 사이트 갱신 구성 마법사 125

대형 오브젝트(LOB)
문자열, 정의 80
위치 지정자, 정의 80

대화식 SQL, 정의 1

데이터 검색 21

데이터 구조
값 3
컬럼 3
행 3

데이터 변환
조인 조건 71
집합 연산자 56

데이터 유형
구별 78
BIGINT 5
CHAR 5
DATE 5
DATETIME 5
DECIMAL 5
DOUBLE 5
FLOAT 5
INTEGER 5
REAL 5
SMALLINT 5
TIME 5
TIMESTAMP 5
VARCHAR 5

데이터 유형 변환
설명 40

데이터베이스 관리 프로그램 1

데이터베이스 백업 마법사 125

데이터베이스 작성 마법사 125

데이터베이스 추가 마법사 125, 126

[라]

랄리스 정보 118

[마]

마법사
다중 사이트 갱신 구성 125
데이터베이스 백업 125
데이터베이스 복원 126
데이터베이스 작성 125
데이터베이스 추가 125, 126
색인 126
성능 구성 126
타스크 완료 125
테이블 공간 작성 126
테이블 작성 126

명령행 처리기 1

문서 서버 설정 126

문자 대형 오브젝트(CLOB) 데이터 유형 80

문자 대형 오브젝트(CLOB) 문자열 80

문자열
가변 길이 5
고정 길이 5
대형 오브젝트(LOB) 80
데이터 유형 5

[바]

보기
온라인 정보 123
복수 노드 관계형 데이터베이스, 정의 1
복원 마법사 126
복합 키 62

부분 합(sub-total) 행 75
부속 조회
 정의 31
부속 조회에 사용된 비교 연산자 59
뷰
 설명 4
 이점 4
 컬럼 이름 규정 45
뷰를 통하여 테이블 수정 16
 WITH CHECK OPTION 16
빠른 시작 v

[사]

사용자 정의 함수 79
 외부 스칼라 함수 79
 외부 테이블 함수 79
 전래 함수 79
 정의 79
 OLE DB 외부 테이블 함수 79
산술 연산자 28
상관
 부속 조회 46
 설명 45
 이름 47
 조인을 사용한 부속 조회 49
상관 부속 조회
 사용시 48
 설명 46
상관 부속 조회 중첩 50
상관 이름
 규칙 45
 컬럼 이름의 규정된 참조 45
상관 참조, 설명 46
상위 키, 정의 63
색인 마법사 126
샘플 데이터베이스 85
 삭제 87
 작성 86
샘플 데이터베이스 삭제 87
샘플 데이터베이스 작성 86

샘플 테이블 85, 109
샘플 프로그램
 상호 플랫폼 117
 HTML 117
선택 목록 21
설치
 Netscape 브라우저 123
성능 구성 마법사 126
소수, 설명 5
순환 조회, 설명 75
술어
 IS NOT NULL 22
 IS NULL 22
숫자 속성으로서의 부호 5
숫자 속성으로서의 정밀도 5
숫자, 설명 5
스칼라 함수 32
 ABS 34
 DECIMAL 42
 HEX 34
 LENGTH 34
 SIGN 34
 YEAR 34
스칼라 fullselect
 설명 39
스키마
 정의 4
스키마 예약 4
시스템 카탈로그 82

[아]

언어 식별자
 책 117
연산 순서 28, 33
오류 메시지
 메시지 식별자 21
 SQLSCODE 21
 SQLSTATE 21
오브젝트 규정 4, 19
온라인 도움말 121

온라인 정보
 검색 127
 보기 123
외부 레벨 술어 59
외부 레벨 조회, 상관 48
외부 스칼라 함수 79
외부 조인
 설명 70
 FULL OUTER 조인 70
 LEFT OUTER 조인 70
 RIGHT OUTER 조인 70
외부 키 62
외부 테이블 함수 79
위치 지정자 80
응용프로그램 개발 안내서 v

[자]

전래 함수 79
정보 센터 124
제한사항
 집합 연산자용 56
제한조건
 고유한 제한조건 9
 참조 제한조건 9
조인
 공동 곱 70
 테이터 변환 71
 상관 부속 조회 49
 정의 30
 조인 조건 70
 조인 조건 없음 70
조인 조건 70
조회 결과 병합 53
조회 연결 56
조회, 연결 56
존재 여부 테스트 58
중복 행 제거 27
중첩 테이블 표현식, 설명 42

[차]

참조 무결성 제한조건

상위 키 63

설명 63

외부 키 63

정의 61

책 109, 119

최신 정보 118

[카]

컬럼

정의 3

ASC, 오름차순 정렬 26

DESC, 내림차순 정렬 26

컬럼 함수 32, 33

AVG 33

COUNT 33

MAX 33

MIN 33

키

고유 62

기본 62

복합 62

외부(foreign) 62

정의 62

[타]

테스트, 존재 여부 58

테이블

결과 테이블 3

고유 키 62

고유한 제한조건 62

기본 키 62

기본 테이블 3

데이터 결합(조인) 30

샘플 데이터베이스 85

외부 키 62

정의 3

테이블 (계속)

컬럼 이름 규정 45

함수 35

테이블 공간 작성 마법사 126

테이블 작성 마법사 126

테이블 점검 제한조건

설명 64

정의 61

제한조건 점검 지연 64

테이블 표현식

설명 42

테이블 함수

SQLCACHE_SNAPSHOT 35

테이블과 뷰 사이의 관계 14

트리거

사전 트리거(before trigger) 65

사후 트리거(after trigger) 65

설명 65

전이 변수 68

정의 61

CREATE TRIGGER 65

특수 레지스터 81

CURRENT DATE 81

CURRENT DEGREE 81

CURRENT FUNCTION PATH 81

CURRENT PATH 81

CURRENT SERVER 81

CURRENT TIME 81

CURRENT TIMESTAMP 81

CURRENT TIMEZONE 81

USER 81

[파]

파티션된 관계형 데이터베이스, 정의 1

표현식 28

표현식, 이름 지정 29

[하]

함수

내장 32

사용자 정의 32

설명 32

스칼라 32

컬럼 32

테이블 35

OLAP(Online Analytical Processing) 76

행

선택 22

정의 3

행 정렬 25

[숫자]

2바이트 문자 대형 오브젝트(DBLOB) 데이터 유형 80

2바이트 문자 대형 오브젝트(DBLOB) 문자열 80

2진 대형 오브젝트(BLOB) 데이터 유형 80

2진 대형 오브젝트(BLOB) 문자열 80

2진 정수, 설명 5

A

ADD CONSTRAINT 문 64

ALL, 조회에서 사용 59

ALTER TABLE 문 64

ANY 키워드 59

AS 절 29

B

BETWEEN 술어 57

BIGINT, 데이터 유형 5

C

CASE 표현식

설명 40

CASE 표현식 (계속)
 SIGN 함수 40
 CHAR, 데이터 유형 5
 CL_SCHED 샘플 테이블 87
 CONNECT 문 20
 내재적 20
 명시적 20
 CREATE DISTINCT TYPE 78
 CREATE FUNCTION 79
 CREATE TABLE 문 9
 컬럼의 NOT NULL/NOT NULL
 WITH DEFAULT 값 9
 CREATE TRIGGER 65
 CREATE VIEW 문 14
 WITH CHECK OPTION 14
 CUBE 75
 공동 테이블(cross-tabulation) 행 75
 부분 합(sub-total) 행 75
 CURRENT DATE 특수 레지스터 81
 CURRENT FUNCTION PATH 특수
 레지스터 81
 CURRENT SERVER 특수 레지스터
 81
 CURRENT TIME 특수 레지스터 81
 CURRENT TIMESTAMP 특수 레지스
 터 81
 CURRENT TIMEZONE 특수 레지스터
 81

D

DATETIME, 데이터 유형 5
 DATE, 데이터 유형 5
 DB2 라이브러리
 구성 방법 109
 마법사 125
 문서 서버 설정 126
 온라인 도움말 121
 온라인 정보 검색 127
 온라인 정보 보기 123
 인쇄된 책 주문 119

DB2 라이브러리 (계속)
 정보 센터 124
 책 109
 책에 대한 언어 식별자 117
 최신 정보 118
 PDF 책 인쇄 119
 DECIMAL, 데이터 유형 5
 DELETE 문 13
 DEPARTMENT 샘플 테이블 87
 DISTINCT 키워드 27, 34
 DOUBLE, 데이터 유형 5

E

EMPLOYEE 샘플 테이블 88
 EMP_ACT 샘플 테이블 91
 EMP_PHOTO 샘플 테이블 93
 EMP_RESUME 샘플 테이블 93
 EXCEPT ALL 55
 EXCEPT 연산자 55
 결과 순서화 56
 사용 제한 포함 56
 데이터 유형 56
 EXISTS 술어 58

F

FLOAT, 데이터 유형 5
 FROM 절 21
 FULL OUTER 조인 70
 fullselect 39
 부속 조회 10, 59
 ALL 키워드 59
 ANY 키워드 59
 INSERT 문 사용 10
 fullselect, 정의 10

G

GROUP BY 28
 GROUP BY 절
 그룹화 컬럼 35

GROUP BY 절 (계속)
 HAVING 절 사용 36

H

HAVING 28
 HAVING 절
 설명 36
 HTML
 샘플 프로그램 117

I

IN 술어 56
 INSERT 문 10
 컬럼의 NOT NULL/NOT NULL
 WITH DEFAULT 값 10
 INTEGER, 데이터 유형 5
 INTERSECT ALL 55
 INTERSECT 연산자 55
 결과 순서화 56
 사용 제한 포함 56
 데이터 유형 56
 IN_TRAY 샘플 테이블 94

L

LEFT OUTER 조인 70
 LIKE 술어 57

N

Netscape 브라우저
 설치 123
 NOT BETWEEN 술어 57
 NOT EXISTS 술어 58
 NOT IN 술어 56
 NOT LIKE 술어 57

O

OLAP 함수 76
 총계 그룹 76

OLAP 함수 76 (계속)
 행 정렬 76
 행 파티션 76
OLAP(Online Analytical
 Processing) 76
OLE DB 외부 테이블 함수 79
ORDER BY 절 25
 집합 연산자 56
ORG 샘플 테이블 94

P

PDF 119
PDF 책 인쇄 119
PROJECT 샘플 테이블 95

R

REAL, 데이터 유형 5
RIGHT OUTER 조인 70
ROLLUP 75
ROLL-UP
 부분 합(sub-total) 행 75

S

SALES 샘플 테이블 96
SELECT 문 21
SET CONSTRAINTS 문 64
SET 절
 UPDATE 문 사용 13
SMALLINT, 데이터 유형 5
SmartGuides
 마법사 125
SOME 키워드 59
SQL 값 5
SQL 참조서 v
SQL 프로시저어 언어 v
SQL, 정의 1
STAFF 샘플 테이블 97
STAFFG 샘플 테이블 98

T

TIMESTAMP, 데이터 유형 5
TIME, 데이터 유형 5

U

UNION ALL 53
UNION 연산자 53, 55
 결과 순서화 54
 사용 제한 포함 56
 데이터 유형 56
 설명 53
UPDATE 문 13
USER 특수 레지스터 81

V

VALUES 절
 INSERT 문 사용 10
VARCHAR, 데이터 유형 5

W

WHERE 절 22
 그룹화시 고려사항 36
 SELECT 절에서 테이블 데이터 결합
 (조인) 30
WITH CHECK OPTION 16
WITH 절 43

IBM에 문의

기술적인 문제가 발생한 경우에는 DB2 고객 지원 센터에 문의하기 전에 문제점 해결 안내서에서 제안한 조치를 검토하고 실행해 보십시오. 이것은 DB2 고객 지원 부서로 하여금 사용자를 보다 더 잘 지원할 수 있도록 사용자가 모을 수 있는 정보를 제공합니다.

DB2 Universal Database 제품에 대한 정보나 주문은 그 지역의 IBM 영업 대표나 공인 IBM 소프트웨어 재판매업자에게 문의하십시오.

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 고객 지원을 받으려면, 1-800-237-5511.
- 사용가능한 서비스 옵션을 알려면, 1-888-426-4343.

제품 정보

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 제품 주문이나 일반 정보를 얻으려면, 1-800-IBM-CALL (1-800-426-2255)이나 1-800-3IBM-OS2 (1-800-342-6672).
- 책에 대한 주문은 1-800-879-2755.

<http://www.ibm.com/software/data/>

DB2 월드 와이드 웹 페이지에는 새로운 소식, 제품 설명, 교육 일정 등에 관한 현재의 DB2 정보를 제공합니다.

<http://www.ibm.com/software/data/db2/library/>

DB2 제품 및 서비스 기술 라이브러리는 빈도 높은 질문(FAQ), 수정사항(fixes), 책 및 최신 DB2 기술 정보에 대한 액세스를 제공합니다.

주: 이러한 정보는 영어로만 제공됩니다.

<http://www.elink.ibm.com/pbl/pbl/>

여기에서는 책을 웹 사이트에서 주문할 수 있는 방법을 제공합니다.

<http://www.ibm.com/education/certify/>

IBM 웹 사이트에서 기술 전문 인증 프로그램은 DB2를 포함하여 다른 IBM 제품의 기술 전문 인증 테스트 정보를 제공합니다.

<ftp.software.ibm.com>

anonymous로 로그인하십시오. /ps/products/db2 디렉토리에서, DB2와 많은 관련 제품에 관한 데이터, 수정사항, 도구 등을 찾을 수 있습니다.

<comp.databases.ibm-db2>, <bit.listserv.db2-l>

이러한 인터넷 뉴스 그룹으로 사용자는 DB2 제품에 대한 자신의 사용 경험을 토론할 수 있습니다.

Compuserve에서, GO IBMDB2

이 명령을 입력하여 IBM DB2 계열 포럼을 액세스하십시오. 모든 DB2 제품이 이러한 포럼을 통해 지원됩니다.

미국 외 지역에서 IBM에 연락하는 방법에 관한 정보는 **IBM Software Support Handbook**의 Appendix A를 참조하십시오. 이 문서에 액세스하려면, 웹 사이트 <http://www.ibm.com/support/>로 가서 페이지 맨 밑에 있는 IBM Software Support Handbook 링크를 클릭하십시오.

주: 일부 국가에서는 IBM-authorized 공인 딜러는 IBM 지원 센터 대신 해당 딜러 지원 부서에 연락해야 합니다.

IBM 한글 지원에 관한 설문



FAX : (02) 781-7778

보내 주시는 의견은 더 나은 고객 지원 체제를 위한 귀중한 자료가 됩니다.
독자 여러분의 좋은 의견을 기다립니다.

책 제목: IBM® DB2® Universal Database
SQL 시작하기
버전 7

책 번호: SA30-0996-00

성명		직위/담당업무	
회사명		부서명	
주소			
전화번호		팩스번호	
전자우편 주소			
사용중인 시스템	<input type="checkbox"/> 중대형 서버 <input type="checkbox"/> UNIX 서버 <input type="checkbox"/> PC 및 PC 서버		

- IBM에서 제공하는 한글 책자와 영문 책자 중 어느 것을 더 좋아하십니까? 그 이유는 무엇입니까?
 한글 책자 영문 책자
 (이유: _____)
- 본 책자와 해당 소프트웨어에서 사용된 한글 용어에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자와 해당 소프트웨어에서 번역 품질에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 본 책자의 인쇄 상태에 대한 귀하의 평가 점수는?
 수 우 미 양 가
- 한글 소프트웨어 및 책자가 지원되는 분야에 대해 귀하는 어떻게 생각하십니까?
 한글 책자를 늘려야 함 현재 수준으로 만족
 그다지 필요성을 느끼지 않음
- IBM은 인쇄물 형식(hardcopy)과 화면 형식(softcopy)의 두 종류로 책자를 제공합니다. 어느 형식을 더 좋아하십니까?
 인쇄물 형식(hardcopy) 화면 형식(softcopy) 둘 다

㉾ IBM 한글 지원 서비스에 대해 기타 제안사항이 있으시면 적어주십시오.

☺ 설문에 대해 주셔서 감사합니다.
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해 드리기 위해 최선을 다하겠습니다.



부품 번호: CT7YHKO

Printed in Singapore

SA30-0996-00



CT7YHKO

