

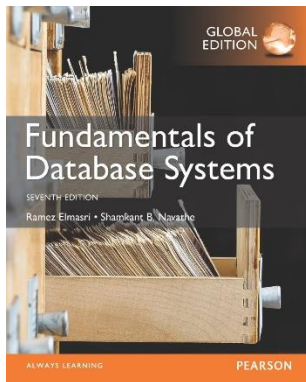
# 데이터베이스 시스템



충북대학교 컴퓨터과학과

이종윤 교수님

# 제6장: 기본 SQL



# 6장 개요

- SQL 데이터 정의 및 데이터 유형
- SQL에서 제약 조건 지정
- SQL의 기본 검색 쿼리
- 삽입, 삭제, 그리고 업데이트 SQL의 문장
- SQL의 추가 기능

# 개요

- SQL 언어

- 관계형 데이터베이스의 상업적 성공을 위한 주요 이유 중 하나로 간주됨

- SQL

- SQL의 기원은 관계적 술어 계산 ~라고 불리는 튜플 미적분학 (8장 참조) 원래는 SQUARE라는 언어로 제안되었습니다.
- SQL은 실제로 "**순서**L"은 논문에서 원래 사용된 용어인 "SEQUEL TO SQUARE"의 약자입니다. **챔벌린과 보이스** IBM은 해당 용어에 대한 저작권을 보호할 수 없었기 때문에 SQL로 줄여서 SQL이라는 용어에 대한 저작권을 보호했습니다.
- 이제 대중적으로 "로 알려져 있습니다 **구조화된 쿼리 언어**".
- SQL은 구문을 사용하여 관계형 데이터 모델을 비공식적으로 또는 실용적으로 렌더링한 것입니다.

# 관계형 작업

- 관계 대수학
- 관계적 계산
  - 튜플 관계 계산
  - 도메인 관계 계산

# 개요

SQL은 원래 IBM에서 개발되었습니다. **도날드 D. 챔버린** 그리고 **레이몬드 F. 보이스** 1970년대 초에 Ted Codd로부터 관계형 모델에 대해 배운 후, 이 버전은 원래 SEQUEL(Structured English Query Language)이라고 불렸으며, IBM의 원래 준관계형 데이터베이스 관리 시스템인 System R에 저장된 데이터를 조작하고 검색하도록 설계되었습니다. 이 시스템은 IBM San Jose Research Laboratory의 한 그룹이 1970년대에 개발했습니다.



**도날드 D. 챔버린**((1944년 12월 21일 출생)은 레이몬드 보이스와 함께 원래 SQL 언어 사양의 주요 설계자 중 한 명으로 가장 잘 알려진 미국의 컴퓨터 과학자입니다. 그는 또한 XQuery 개발에 상당한 기여를 했습니다.



**레이몬드 F. 보이스**(1947-1974)는 관계형 데이터베이스 연구로 유명한 미국의 컴퓨터 과학자였습니다. 그는 SQL 데이터베이스를 공동 개발한 작업으로 가장 잘 알려져 있습니다.



언어와 보이스-코드 정규형.

# SQL 데이터 정의 및 데이터 유형

# SQL 데이터 정의, 데이터 유형, 표준

- 술어:

- 테이블, 열, 그리고 열관계형 모델 용어에 사용됨 관계, 튜플, 그리고 기인하다

- 만들다성명

- 데이터 정의를 위한 주요 SQL 명령

- 이 언어에는 다음과 같은 특징이 있습니다. 데이터 정의, 데이터 조작, 트랜잭션 제어, 인덱싱, 보안 사양(승인하다그리고취소),활성 데이터베이스(방아쇠), 멀티미디어, 분산형 데이터베이스등.



# SQL 표준

- SQL은 많은 표준을 거쳤습니다. **SQL-86** 또는 SQL 1.A. **SQL92(한국어)** ~로 지칭된다 **SQL-2**.
- 이후의 표준 (**한국어: SQL-1999**) 이다 핵심사양과 전문사양으로 구분 이자형d 확장자.
  - 확장 기능은 데이터 마이닝, 데이터 웨어하우징, 멀티미디어 등과 같은 다양한 애플리케이션에 구현됩니다.
- **한국어: SQL-2006** XML 기능 추가(13장) **2008년에** 그들은 객체 지향적 기능을 추가했습니다(12장).
- **SQL-3** SQL-1999로 시작된 현재 표준입니다. 완전히는 아니다 피모든 RDBMS에 포함됨.

# SQL의 스키마 및 카탈로그 개념

- 우리는 기본적인 표준 SQL 구문을 다룹니다.있다 다섯아리에이션 기존 RDBMS 시스템에서

- SQL 스키마

일부 시스템에서는 스키마를 다음과 같이 부릅니다. *데이터 베이스*

- 에 의해 식별됨 스키마 이름
- 포함 사항 권한 식별자 그리고 설명자 각 요소에 대해

- 스키마 요소 포함하다

- 테이블, 제약 조건, 뷰, 도메인 및 기타 구성 요소

- 각 의원들은 티 SQL에서는 세미콜론으로 끝납니다.

# SQL의 스키마와 카탈로그 개념(계속)

- 스키마 생성성명

- 회사 권한 부여 스키마 'Jsmith'를 생성합니다.

- 목록

- 명명됨스키마 모음 SQL 환경에서

- SQL에는 카탈로그 클러스터라는 개념도 있습니다.

# 그만큼테이블 생성SQL의 명령

- 새로운 관계 지정

- 제공하다테이블의 이름
- 지정하다속성, 유형 및 초기 제약 조건

- 선택적으로 스키마를 지정할 수 있습니다:

- 회사.직원 테이블 생성 ... 또는
- 직원 테이블 생성 ...

# 그만큼테이블 생성SQL 명령(계속)

- 기본 테이블(기본 관계)

- 관계와 그 튜플은 실제로 파일로 생성되어 저장됨 DBMS에 의해

- 가상 관계 (조희수)

- 다음을 통해 생성됨뷰 생성성명.
  - 어떤 물리적 파일과도 일치하지 않습니다.

# 회사 관계형 데이터베이스 스키마(그림 5.7)

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

## PROJECT

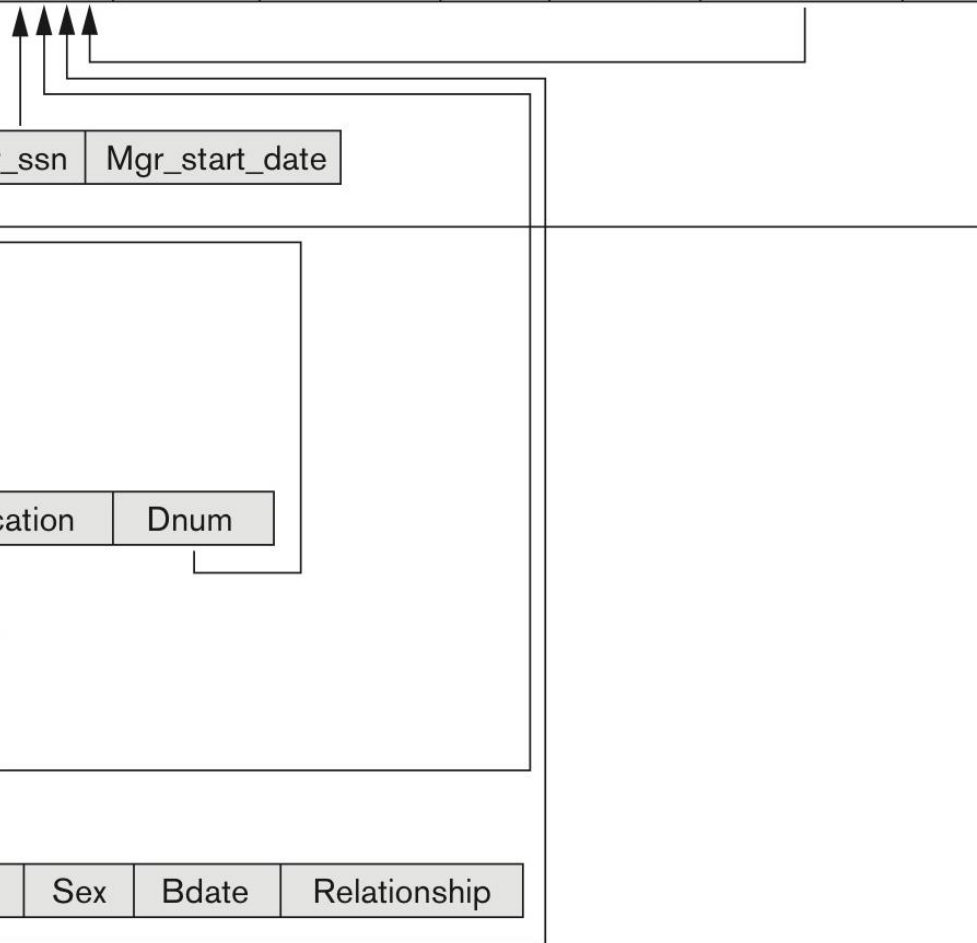
Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------



# COMPANY 관계형 데이터베이스 스키마에 대한 가능한 데이터베이스 상태 중 하나(그림 5.6)

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

# COMPANY 관계형 데이터베이스 스키마에 대한 가능한 데이터베이스 상태 중 하나 - 계속(그림 5.6)

WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

<u>Pname</u>	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse



## 그림 5.7의 COMPANY 스키마를 정의하기 위한 SQL CREATE TABLE 데이터 정의 문(그림 6.1)

```
CREATE TABLE EMPLOYEE
( Fname          VARCHAR(15)          NOT NULL,
  Minit          CHAR,
  Lname          VARCHAR(15)          NOT NULL,
  Ssn            CHAR(9)              NOT NULL,
  Bdate          DATE,
  Address        VARCHAR(30),
  Sex            CHAR,
  Salary         DECIMAL(10,2),
  Super_ssn      CHAR(9),
  Dno            INT                  NOT NULL,
  PRIMARY KEY (Ssn),
CREATE TABLE DEPARTMENT
( Dname          VARCHAR(15)          NOT NULL,
  Dnumber        INT                  NOT NULL,
  Mgr_ssn        CHAR(9)              NOT NULL,
  Mgr_start_date DATE,
  PRIMARY KEY (Dnumber),
  UNIQUE (Dname),
  FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
CREATE TABLE DEPT_LOCATIONS
( Dnumber        INT                  NOT NULL,
  Dlocation      VARCHAR(15)          NOT NULL,
  PRIMARY KEY (Dnumber, Dlocation),
  FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
```

# SQL테이블 생성데이터 정의를 위한 데이터 정의 문장회사그림

## 5.7의 스키마(그림 6.1)

**CREATE TABLE PROJECT**

( Pname	VARCHAR(15)	NOT NULL,
Pnumber	INT	NOT NULL,
Plocation	VARCHAR(15),	
Dnum	INT	NOT NULL,

**PRIMARY KEY** (Pnumber),

**UNIQUE** (Pname),

**FOREIGN KEY** (Dnum) **REFERENCES** DEPARTMENT(Dnumber) );

**CREATE TABLE WORKS\_ON**

( Essn	CHAR(9)	NOT NULL,
Pno	INT	NOT NULL,
Hours	DECIMAL(3,1)	NOT NULL,

**PRIMARY KEY** (Essn, Pno),

**FOREIGN KEY** (Essn) **REFERENCES** EMPLOYEE(Ssn),

**FOREIGN KEY** (Pno) **REFERENCES** PROJECT(Pnumber) );

**CREATE TABLE DEPENDENT**

( Essn	CHAR(9)	NOT NULL,
Dependent_name	VARCHAR(15)	NOT NULL,
Sex	CHAR,	
Bdate	DATE,	
Relationship	VARCHAR(8),	

**PRIMARY KEY** (Essn, Dependent\_name),

**FOREIGN KEY** (Essn) **REFERENCES** EMPLOYEE(Ssn) );

MySQL>데이터베이스 회사 생성; 쿼리 확인, 1개  
행이 영향을 받음(0.01초)

MySQL>

MySQL>사용 회사;

데이터베이스가 변경되었습니다

MySQL>

mysql> CREATE TABLE EMPLOYEE (

-> Fname VARCHAR(15) NULL이 아닙니다.

-> 미니트 CHAR,

-> Lname VARCHAR(15) NULL이 아닙니다.

-> Ssn CHAR(9) NULL이 아닙니다.

-> Bdata 날짜,

-> 주소 VARCHAR(30),

-> 섹스 CHAR,

-> 급여 DECIMAL(10,2),

-> Super\_ssn CHAR(9),

-> INT를 모르니,

-> 기본 키(Ssn));

mysql> 테이블 표시;

```
+-----+
- + | 회사의 테이블 |
+-----+
-- + | 직원          |
+-----+
```

나의질문>설명 직원;

		+-----+-----+-----+-----+ 필드			
	유형	Null	키	기본값	추가
여	varchar(15)	아니요			널(NULL)
엠 초기화	문자(1)	예			널(NULL)
엘 이름	varchar(15)	아니요			널(NULL)
씨 스엔	문자(9)	아니요		기본	널
비 데이터	날짜	예			널(NULL)
아 주소	varchar(30)	예			널(NULL)
씨 전-	문자(1)	예			널(NULL)
씨 알람	소수(10,2)	예			널(NULL)
씨 상위_ssn	char(9)	예			널(NULL)
디노	정수(11)	예			널(NULL)

+-----+-----+-----+-----+ 세트당 10개 행  
(0.00초)

# SQL의 속성 데이터 유형 및 도메인

- 기본 데이터 유형

- 숫자데이터 유형

- 정수:정수, 정수,그리고스몰린트
    - 부동 소수점(실수) 숫자:뜨다또는진짜,그리고이중 정밀도

- 문자열데이터 유형

- 고정 길이:숫(M), 성격(M)
    - 다양한 길이:VARCHAR(M), CHAR VARYING(M), 성격  
다양한(M)

# SQL의 속성 데이터 유형과 도메인(계속)

## – 비트 문자열 데이터 유형

- 고정 길이: **조금(M)**
- 다양한 길이: **비트 가변(M)**

## – 부울 데이터 유형

- 값 **진실** 또는 **거짓** 또는 **널(NULL)**

## – 날짜 데이터 유형

- 10개 포지션
- 구성 요소는 **년도**, **월**, 그리고 **낮** 형태로 **년-월-일**
- RDBMS에서 날짜 형식을 변경하기 위한 다양한 매핑 기능 제공

# SQL의 속성 데이터 유형과 도메인(계속)

- 추가 데이터 유형

- 타임스탬프데이터 유형

포함 내용날짜그리고시간전지

- 초의 소수점 이하 자릿수에 대해 최소 6개 위치 추가
    - 선택 사항시간대 포함자격자

- 간격데이터 유형

- 증가 또는 감소하는 데 사용할 수 있는 상대 값을 지정합니다.

날짜, 시간 또는 타임스탬프의 절대값

- 날짜, 시간, 타임스탬프, 간격데이터 유형은 다음과 같습니다.킵스또는  
비교를 위해 문자열 형식으로 변환합니다.

# SQL의 속성 데이터 유형과 도메인(계속)

- **도메인**

- 속성 사양과 함께 사용되는 이름
- 도메인에서 사용하는 데이터 유형을 변경하는 것을 더 쉽게 만듭니다.  
다양한 속성
- 스키마 가독성 향상
- 예:
  - 도메인 SSN\_TYPE을 CHAR(9)로 생성합니다.

- **유형**

- 객체 지향형의 경우 사용자 정의 유형(UDT)이 지원됩니다.  
응용 프로그램. (Ch.12 참조) 다음 명령을 사용합니다. **유형 생성**

# SQL에서 제약 조건 지정



# SQL에서 제약 조건 지정

기본 제약 사항:

- 관계 모델에는 SQL에서 지원되는 3가지 기본 제약 유형이 있습니다.
  - **열쇠** 제약 조건: 기본 키 값은 복제될 수 없습니다.
  - **엔티티 무결성** 제약 조건: 기본 키 값은 null일 수 없습니다.
  - **참조 무결성** 제약 조건 : "외래 키"에는 값이 있어야 합니다.  
이미 존재하는 것으로 기본 키 , 또는 될 수도 있습니다.

# 속성 제약 조건 지정

속성 도메인에 대한 기타 제한 사항:

- 속성의 기본값
  - 기본<가치>
  - 널(NULL)특정 속성에는 허용되지 않습니다.NULL이 아닙니다)
- **확인하다**절
  - Dnumber INT NOT NULL CHECK(Dnumber > 0 AND Dnumber < 21);

# 키 및 참조 무결성 제약 조건 지정

- **기본 키절**

- 관계의 기본 키를 구성하는 하나 이상의 속성을 지정합니다.
- Dnumber INT 기본 키;

- **고유한절**

- 대체(보조) 키(라고 함)를 지정합니다.후보자(관계형 모델의 키).
- Dname VARCHAR(15) UNIQUE;

# 키 및 참조 무결성 제약 조건 지정(계속)

- 외래키절

- 기본 작업:위반 시 업데이트 거부

- 불이다참조 트리거 동작절

- 옵션에는 다음이 포함됩니다.NULL 설정, CASCADE,그리고기본값으로 설정
    - DBMS에서 수행한 작업NULL 설정또는기본값으로 설정에 대해서도 동일하다  
둘 다삭제시그리고업데이트 중
    - 종속적합한 옵션"관계" 관계, 다중 값 속성,  
약한 엔티티 유형

# 제약조건에 이름 지정

- 키워드 사용강제
  - 제약조건의 이름 지정
  - 나중에 변경하는 데 유용합니다.

# 기본 속성 값 및 참조 무결성 트리거 작업 사양(그림 6.2)

```
CREATE TABLE EMPLOYEE
(
    ...,
    Dno          INT          NOT NULL      DEFAULT 1,
    CONSTRAINT EMPPK
        PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
            ON DELETE SET NULL      ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
            ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
(
    ...,
    Mgr_ssn CHAR(9)          NOT NULL      DEFAULT '888665555',
    ...,
    CONSTRAINT DEPTPK
        PRIMARY KEY (Dnumber),
    CONSTRAINT DEPTSK
        UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
            ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
(
    ...,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
        ON DELETE CASCADE          ON UPDATE CASCADE);
```

# 튜플에 대한 제약 조건 지정확인하다

- 관계 내 개별 튜플에 대한 추가 제약 조건도 다음을 사용하여 가능합니다.확인하다
  - **확인하다**조항의 끝에서테이블 생성성명
    - 각 튜플에 개별적으로 적용
    - CHECK(부서생성날짜 <= 관리자시작날짜);

# SQL의 기본 검색 쿼리



# SQL의 기본 검색 쿼리

- 선택하다성명

- 데이터베이스에서 정보를 검색하기 위한 기본 문장 하나

- SQL에서는 테이블에 두 개 이상의 튜플이 있을 수 있습니다. 모든 속성 값이 동일한

- 관계형 모델과 달리(관계형 모델은 엄격하게 집합 이론 기반)

- 멀티셋 또는 가방 동작

- Tuple-id는 키로 사용될 수 있습니다.

# 그만큼선택-위치-기본 SQL 쿼리의 구조

- 기본 형태**선택하다**성명:

```
SELECT    <attribute list>  
FROM      <table list>  
WHERE     <condition>;
```

where

- <attribute list> is a list of attribute names whose values are to be retrieved by the query.
- <table list> is a list of the relation names required to process the query.
- <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.

# 그만큼선택-위치-기본 SQL 쿼리의 구조

- 논리 비교 연산자

- =, <, <=, >, >=, 그리고 <>

- 투영 속성

- 값을 검색할 속성

- 선택 조건

- 검색된 튜플에 대해 참이어야 하는 부울 조건. 선택 조건에는 여러 관계가 관련될 때  
조인 조건(8장 참조)이 포함됩니다.

# 데이터베이스 상태회사

**Figure 5.6**

One possible database state for the COMPANY relational database schema.

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

## DEPARTMENT

Dname	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

# 기본 검색 쿼리

**Query 0.** Retrieve the birth date and address of the employee(s) whose name is 'John B. Smith'.

**Q0:**      **SELECT**      Bdate, Address  
             **FROM**        EMPLOYEE  
             **WHERE**      Fname = 'John' **AND** Minit = 'B' **AND** Lname = 'Smith';

<u>Bdate</u>	<u>Address</u>
1965-01-09	731 Fondren, Houston, TX

<u>Fname</u>	<u>Lname</u>	<u>Address</u>
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

**Query 1.** Retrieve the name and address of all employees who work for the 'Research' department.

**Q1:**      **SELECT**      Fname, Lname, Address  
             **FROM**        EMPLOYEE, DEPARTMENT  
             **WHERE**      Dname = 'Research' **AND** Dnumber = Dno;

# 기본 검색 쿼리(계속)

**Query 2.** For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

**Q2:**       **SELECT**       Pnumber, Dnum, Lname, Address, Bdate  
             **FROM**       PROJECT, DEPARTMENT, EMPLOYEE  
             **WHERE**       Dnum = Dnumber **AND** Mgr\_ssn = Ssn **AND**  
                          Plocation = 'Stafford'

(c)

<u>Pnumber</u>	<u>Dnum</u>	<u>Lname</u>	<u>Address</u>	<u>Bdate</u>
10	4	Wallace	291Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291Berry, Bellaire, TX	1941-06-20

# 모호한 속성 이름

- 같은 이름 기음~에 사용되다 영형r 두 개(또는 더 많은 아르 자형e) 속성 다른 관계에서

- 속성이 서로 다른 관계에 있는 한
- 해야 하다**자격을 얻다**관계 이름을 속성 이름으로 변경하여 방지

모호

**Q1A:**     **SELECT**     Fname, EMPLOYEE.Name, Address  
             **FROM**         EMPLOYEE, DEPARTMENT  
             **WHERE**        DEPARTMENT.Name = 'Research' **AND**  
                             DEPARTMENT.Dnumber = EMPLOYEE.Dnumber;

**Q1':**       **SELECT**        EMPLOYEE.Fname, EMPLOYEE.LName,  
                             EMPLOYEE.Address  
             **FROM**        EMPLOYEE, DEPARTMENT  
             **WHERE**        DEPARTMENT.DName = 'Research' **AND**  
                             DEPARTMENT.Dnumber = EMPLOYEE.Dno;

# 별칭 지정 및 이름 변경

- **별칭**또는 튜플 변수

- 대체 관계 이름을 선언합니다. 이자형 그리고 에스~을 참조하다 직원 쿼리에서 관계가 두 번:

질문 8:

각 직원에 대해 **직원의 성과 이름 및 직속 상사의 성과 이름을 검색합니다..**

선택하다 E.Fname, E.Lname, S.Fname, S.Lname  
에서 직원 AS E, 직원 AS S  
여기서 E.Super\_ssn=S.Ssn;

- 여러 테이블에서 동일하거나 유사한 속성에 이름을 약어로 표시하고 접두사를 붙이는 것이 권장되는 관행입니다.



# 별칭 지정 및 이름 변경

선택하다 E.Fname, E.Lname, S.Fname, S.Lname  
에서 직원 AS E, 직원 AS S  
여기서 E.Super\_ssn=S.Ssn;

<u>E.Fname</u>	<u>E.Lname</u>	<u>S.Fname</u>	<u>S.Lname</u>
John	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Alicia	Zelaya	Jennifer	Wallace
Jennifer	Wallace	James	Borg
Ramesh	Narayan	Franklin	Wong
Joyce	English	Franklin	Wong
Ahmad	Jabbar	Jennifer	Wallace

# 별칭 지정, 이름 변경 및 튜플 변수(계속)

- 속성 이름도 바꿀 수 있습니다.

직원 AS E(Fn, Mi, Ln, Ssn, Bd, Address, Sex, Sal, Sssn, Dno)

- 다음 관계에 주목하세요. 직원이제 변수 이름이 있습니다. 이자형 튜플 변수에 대응함
- “처럼” 삭제될 수 있습니다. 대부분의 SQL 구현에서

# 지정되지 않음어디별표의 조항 및 사용

- **없어진어디절**

- 튜플 선택에 조건이 없음을 나타냅니다.

- **효과는교차곱**

- 결과는 모든 가능한 튜플 조합입니다. (또는 데카르트 곱의 대수 연산 - 8장 참조) 결과

**Queries 9 and 10.** Select all EMPLOYEE Ssns (Q9) and all combinations of EMPLOYEE Ssn and DEPARTMENT Dname (Q10) in the database.

**Q9:**      **SELECT**      Ssn  
             **FROM**        EMPLOYEE;

→ **Q10:**    **SELECT**      Ssn, Dname  
             **FROM**        EMPLOYEE, DEPARTMENT;

# 지정되지 않음 어디별표의 조항 및 사용

**Queries 9 and 10.** Select all EMPLOYEE Ssns (Q9) and all combinations of EMPLOYEE Ssn and DEPARTMENT Dname (Q10) in the database.

**Q9:**        **SELECT**        Ssn  
              **FROM**        EMPLOYEE;

**Q10:**       **SELECT**        Ssn, Dname  
              **FROM**        EMPLOYEE, DEPARTMENT;

(f)

Ssn	Dname
123456789	Research
333445555	Research
999887777	Research
987654321	Research
666884444	Research
453453453	Research
987987987	Research
888665555	Research
123456789	Administration
333445555	Administration
999887777	Administration
987654321	Administration
666884444	Administration
453453453	Administration
987987987	Administration
888665555	Administration
123456789	Headquarters
333445555	Headquarters
999887777	Headquarters
987654321	Headquarters
666884444	Headquarters
453453453	Headquarters
987987987	Headquarters
888665555	Headquarters

사회보장번호
123456789
333445555
999887777
987654321
666884444
453453453
987987987
888665555

**Figure 5.6**  
One possible database state for the COMPANY relational database schema.

EMPLOYEE									
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS	
Dnumber	Location
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON		
Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT				
Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

# 지정되지 않은 WHERE 절과 별표의 사용(계속)

- 별표(\*)를 지정하세요

- 선택된 튜플의 모든 속성 값을 검색합니다.
- \*는 관계 이름 앞에 붙을 수 있습니다. 예: EMPLOYEE \*



Q1C:    SELECT    \*  
         FROM    EMPLOYEE  
         WHERE    Dno = 5;

직원.\*

Q1D:    SELECT    \*  
         FROM    EMPLOYEE, DEPARTMENT  
         WHERE    Dname = 'Research' AND Dno = Dnumber;

Q10A:   SELECT    \*  
         FROM    EMPLOYEE, DEPARTMENT;

# 지정되지 않은 WHERE 절과 별표의 사용(계속)

Q1C:    SELECT     \*  
          FROM     EMPLOYEE  
          WHERE    Dno = 5;

Q1D:    SELECT     \*  
          FROM     EMPLOYEE, DEPARTMENT  
          WHERE    Dname = 'Research' AND Dno = Dnumber;

Q10A:   SELECT     \*  
          FROM     EMPLOYEE, DEPARTMENT;

<u>Fname</u>	<u>Minit</u>	<u>Lname</u>	Ssn	<u>Bdate</u>	<u>Address</u>	<u>Sex</u>	<u>Salary</u>	<u>Super_ssn</u>	<u>Dno</u>
John	B	Smith	123456789	1965-09-01	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

# SQL에서 집합으로서의 테이블

- SQL은 자동으로 와이드를 제거하다 피복잡한 튜플 쿼리 결과에서
- 키워드를 사용하세요 **별개의**에서 선택하다 절
  - 결과에는 고유한 튜플만 남아야 합니다.

**Query 11.** Retrieve the salary of every employee (Q11) and all distinct salary values (Q11A).

Q11:     **SELECT**     **ALL** Salary  
          **FROM**     EMPLOYEE;

→ Q11A:   **SELECT**     **DISTINCT** Salary  
          **FROM**     EMPLOYEE;

(a)

Salary
30000
40000
25000
43000
38000
25000
25000
55000

(b)

Salary
30000
40000
25000
43000
38000
55000

# SQL의 집합으로서의 테이블(계속)

- **작업 설정**
  - 노동 조합,제외하고(차이점),교차하다
  - 이러한 작업이 유효하려면 유형 호환성이 필요합니다.

**Query 4.** Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

**Q4A:**

```
( SELECT DISTINCT Pnumber
  FROM PROJECT, DEPARTMENT, EMPLOYEE
 WHERE Dnum = Dnumber AND Mgr_ssn = Ssn
   AND Lname = 'Smith' )

→ UNION

( SELECT DISTINCT Pnumber
  FROM PROJECT, WORKS_ON, EMPLOYEE
 WHERE Pnumber = Pno AND Essn = Ssn
   AND Lname = 'Smith' );
```



# SQL의 집합으로서의 테이블(계속)

## • 작업 설정

– 해당멀티셋 연산 :모두 합치기, 모두 제외, 모두 교차)

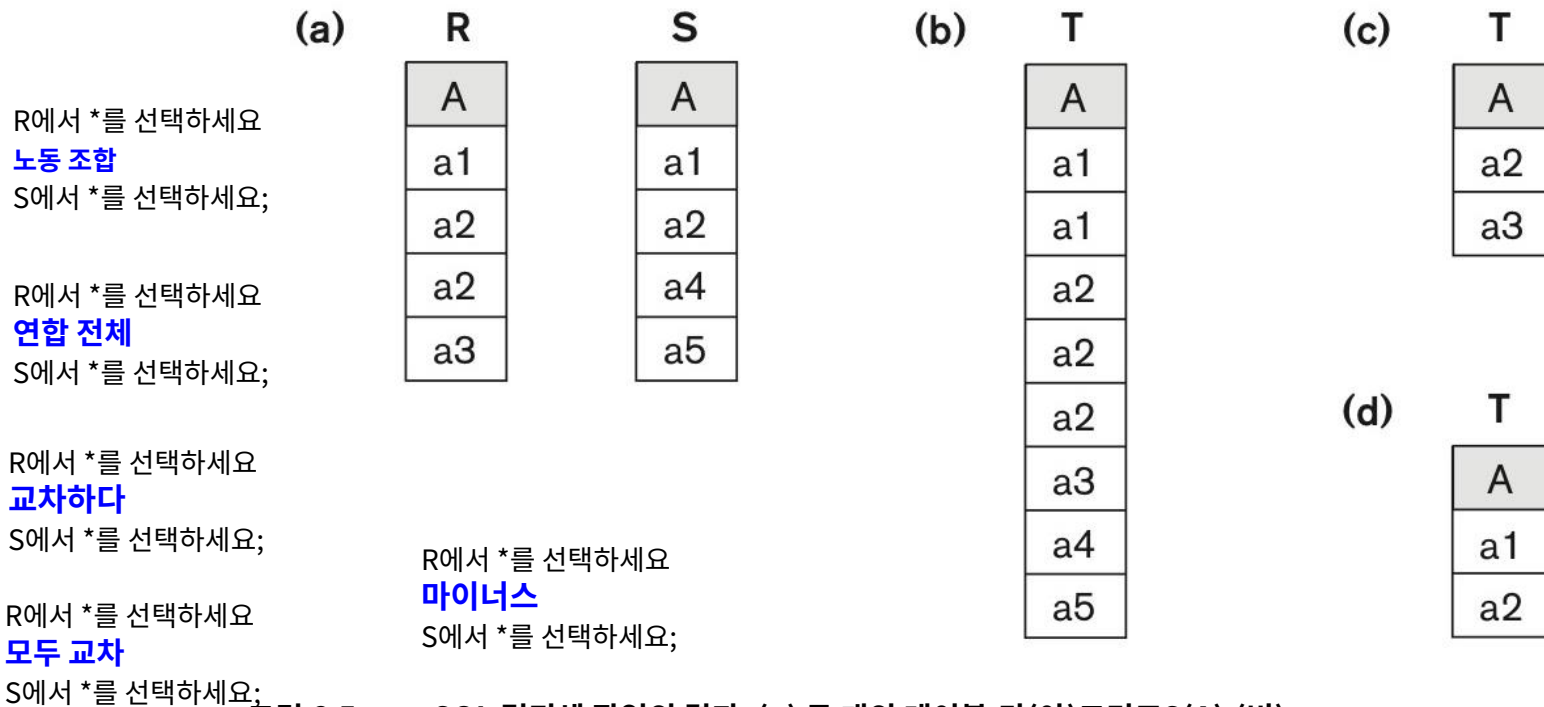


그림 6.5 SQL 멀티셋 작업의 결과. (a) 두 개의 테이블,라(아)그리고S(A).(비)  
R(A) UNION ALL S(A). (기음)모든 S(A)를 제외한 R(A) (디)R(A)는 모든 S(A)와 교집합합니다.

# 부분 문자열 패턴 매칭 및 산술 연산자

- **좋다비교 연산자**

- 문자열에 사용됨 **패턴 매칭**
- %는 임의의 수의 0개 이상의 문자를 대체합니다.
- **밑줄(\_)은 단일 문자를 대체합니다.**
- 예: **어디**주소**좋다**'%휴스턴,TX%';
- **어디**사회보장번호**좋다**'\_\_1\_\_ 8901';

**Query 12.** Retrieve all employees whose address is in Houston, Texas.

```
Q12:  SELECT    Fname, Lname
      FROM      EMPLOYEE
      WHERE     Address LIKE '%Houston,TX%';
```

**Query 12A.** Find all employees who were born during the 1950s.

```
Q12:  SELECT    Fname, Lname
      FROM      EMPLOYEE
      WHERE     Bdate LIKE '__7 _ _ _ _ _';
```

# 부분 문자열 패턴 매칭 및 산술 연산자

```
mysql> select employees.* from employees where last_name like '%mm%' limit 5;
```

```
+-----+-----+-----+-----+-----+-----+ | 사  
원 번호 | 생년월일 | 이름 | 성 | 성별 | 입사일 |  
+-----+-----+-----+-----+-----+-----+ |  
10002 | 1964-06-02 | 브잘렐 10301 | | 짐멜 | 여 | 1985-11-21 |  
| 1962-08-26 | 루시앙 10353 | | 슈타우드해머 | 엠 | 1988-05-23 | |
| 1953-01-15 | 풀레 10436 | 1963-06-17 | 해머슈미트 | M | 램멜 | 1989-08-24 |  
| 야히코 10777 | 1955-08-11 | 미즈히토 | | 엠 | 1988-01-30 |  
| | 케머러 | | 여 | 1988-07-05 |  
+-----+-----+-----+-----+-----+-----+ | 세트당  
5개 행(0.00초)
```

```
mysql> select employees.* from employees where last_name like '_a%' limit 5;
```

```
+-----+-----+-----+-----+-----+-----+ | 사원  
번호 | 생년월일 | 이름 | 성 | 성별 | 입사일 |  
+-----+-----+-----+-----+-----+-----+ |  
10001 | 1953-09-02 | 조지 10003 | | 파셀로 | 엠 | 1986-06-26 |  
| 1959-12-03 | 파르토 10005 | | 뱀포드 | 엠 | 1986-08-28 |  
| 1955-01-21 | 교이치 10008 | | 말리니악 | 엠 | 1989-09-12 |  
| 1958-02-19 | 사니야 10016 | | 칼루피 | 엠 | 1994-09-15 |  
| 1961-05-02 | 카즈히토 | | 카펠레티 | M | 1995-01-27 |  
+-----+-----+-----+-----+-----+-----+ | 세트당  
5개 행(0.00초)
```

# 부분 문자열 패턴 매칭 및 산술 연산자

- 사이비교 연산자

**Query 14.** Retrieve all employees in department 5 whose salary is between \$30,000 and \$40,000.

**Q14:**     **SELECT**        \*  
             **FROM**        EMPLOYEE  
             **WHERE**        (Salary **BETWEEN** 30000 **AND** 40000) **AND** Dno = 5;

```
mysql> SELECT E.first_name, S.salary FROM employees E, salaries S
WHERE E.emp_no = S.emp_no AND S.salary BETWEEN 50000 AND 60000
LIMIT 5;
```

```
+-----+-----+
+ | 이름 | 급여 |
+-----+-----+
+ | 크리스찬 | 50594 |
| 크리스찬 | 52119 |
| 크리스찬 | 54693 |
| 크리스찬 | 58326 |
| 안네케 | 52255 |
+-----+-----+
+ 5개 행 세트(0.00초)
```

# 산술 연산

- 표준 산술 연산자:

- 덧셈(+), 뺄셈(-), 곱셈(\*), 나눗셈(/)이 다음의 일부로 포함될 수 있습니다. **선택하다**

**Query 13.** Show the resulting salaries if every employee working on the 'ProductX' project is given a 10% raise.

→ **Q13:**     **SELECT**     E.Fname, E.Lname, 1.1 \* E.Salary **AS** Increased\_sal  
              **FROM**       EMPLOYEE **AS** E, WORKS\_ON **AS** W, PROJECT **AS** P  
              **WHERE**     E.Ssn = W.Essn **AND** W.Pno = P.Pnumber **AND**  
                         P.Pname = 'ProductX';

# 산술 연산

**Query 13.** Show the resulting salaries if every employee working on the 'ProductX' project is given a 10% raise.

**Q13:**     **SELECT**     E.Fname, E.Lname, 1.1 \* E.Salary **AS** Increased\_sal  
          **FROM**     EMPLOYEE **AS** E, WORKS\_ON **AS** W, PROJECT **AS** P  
          **WHERE**    E.Ssn = W.Essn **AND** W.Pno = P.Pnumber **AND**  
                    P.Pname = 'ProductX';

MySQL>E.이름, S.급여를 선택하세요.1.1\*S.salary를 increased\_salary로 변경  
원 E의 급여 S(E.emp\_no = S.emp\_no 제한 5)

이름	급여 인상	급여
조지	60117	66128.7
조지	62102	68312.2
조지	66074	72681.4
조지	66596	73255.6
조지	66961	73657.1

-----+ 5개 행 세트(0.00초)

# 쿼리 결과의 정렬

- **사용주문하기절**

- 키워드**설명**값을 내림차순으로 결과를 보려면
- 키워드**ASC**오름차순을 명시적으로 지정하려면
- 일반적으로 쿼리의 끝에 배치됩니다.

D.Dname DESC, E.Lname ASC, E.Fname ASC 순으로 정렬

```
mysql> SELECT E.first_name, S.salary FROM employees E,  
salaries S WHERE E.emp_no = S.emp_no AND S.salary주문하  
기S. 급여 한도 5;
```

```
+-----+-----+  
+ | 이름 | 급여 |  
+-----+-----+  
| 올리베라 | 38623 |  
| 후미야 | 38735 |  
| 추안이 | 38786 |  
| 유리 | 38812 |  
| 메히틸드 | 38836 |  
+-----+-----+
```

```
mysql> SELECT E.first_name, S.salary FROM employees E,  
salaries S WHERE E.emp_no = S.emp_no AND S.salary주문하  
기S.급여설명제한 5;
```

```
+-----+-----+  
+ | 이름 | 급여 |  
+-----+-----+  
+ | 도쿠야스 | 158220 |  
| 토쿠야스 | 157821 |  
| 정직함 | 156286 |  
| 샤화 | 155709 |  
| 산자이 | 155513 |  
+-----+-----+
```

# 기본 SQL 검색 쿼리 블록

<b>SELECT</b>	<attribute list>
<b>FROM</b>	<table list>
<b>[ WHERE</b>	<condition> ]
<b>[ ORDER BY</b>	<attribute list> ];



# **삽입, 삭제,그리고업데이트 SQL의 문장**

# 삽입, 삭제, 업데이트SQL의 문장

- 데이터베이스를 수정하는 데 사용되는 세 가지 명령은 다음과 같습니다.
  - 삽입, 삭제, 그리고 업데이트
- 끼워 넣다 일반적으로 관계(테이블)에 튜플(행)을 삽입합니다.
- 업데이트조건을 만족하는 관계(테이블)의 여러 튜플(행)을 업데이트할 수 있습니다.
- 삭제또한 조건을 만족하는 관계(테이블)의 여러 튜플(행)을 업데이트할 수도 있습니다.

# 끼워 넣다

- 가장 단순한 형태로는 다음과 같이 사용됩니다.하나 이상의 튜플을 추가하려면 관계에
- 속성 값은 속성이 지정된 순서와 동일한 순서로 나열되어야 합니다.테이블 생성명령
- 데이터 유형에 대한 제약 조건이 자동으로 준수됩니다.
- DDL 사양의 일부로서 모든 무결성 제약 조건이 적용됩니다.

# 그만큼끼워 넣다명령

- 튜플에 대한 관계 이름과 값 목록을 지정합니다. null을 포함한 모든 값이 제공됩니다.

```
U1:  INSERT INTO      EMPLOYEE
      VALUES          ( 'Richard', 'K', 'Marini', '653298653', '1962-12-30', '98
                        Oak Forest, Katy, TX', 'M', 37000, '653298653', 4 );
```

MySQL>직원에 삽입

```
      값('Richard', 'K', 'Marini', '653298653', '1962-12-30', '98 Oak Forest, Katy, TX', 'M', 37000,
'653298653', 4);
```

쿼리 확인, 1개 행이 영향을 받음(0.01초)

```
mysql> SELECT * from EMPLOYEE;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+ | 이름      | 이름 | 성 | 주민등록번호 | 수퍼          | 비데이터          | 주소                  | 섹스
+ | 급여      | _SSN | DNO |              |               |                   |                      |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+ | 리차드 | 케이          | 마리니 | 653298653 | 1962-12-30 | 98 오크 포레스트, 케이티, 텍사스 | M
+ | 37000.00 | 653298653 | 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
```

# 그만큼끼워 넣다명령

**U1A: INSERT INTO EMPLOYEE (Fname, Lname, Dno, Ssn)**  
**VALUES ('Richard', 'Marini', 4, '653298653');**

mysql> INSERT INTO EMPLOYEE (성명, 성, 일련번호, 주민등록번호)  
-> 값('에드거', '코드', 1, '111111111');  
쿼리 확인, 1개 행이 영향을 받음(0.01초)

MySQL>

mysql> SELECT \* from EMPLOYEE;

```
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
+ | 이름      | 이름 | 성 | 주민등록번호 | 수퍼      | 비데이터      | 주소      | 섹스
+ | 급여      | _SSN | DNO |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
-- + | 에드거 | 널      | 코드      | 111111111 | 널 1 |      | 널(NULL)      | 널(NULL)
+ |      | 널 | 널      |
+ | 리차드 | 케이      | 마리니 | 653298653 | 1962-12-30 | 98 오크 포레스트, 케이티, 텍사스 | M
+ | 37000.00 | 653298653 |      | 4 |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+
-- + 2개 행 세트(0.00초)
```

# 그만큼끼워 넣다명령

- 아래 변형은 쿼리 결과에서 값을 로드하여 새 테이블에 여러 튜플을 삽입합니다.

```
U3A:  CREATE TABLE      WORKS_ON_INFO
      ( Emp_name          VARCHAR(15),
        Proj_name         VARCHAR(15),
        Hours_per_week    DECIMAL(3,1) );
```

```
U3B:  INSERT INTO        WORKS_ON_INFO ( Emp_name, Proj_name,
                                         Hours_per_week )
      SELECT              E.Lname, P.Pname, W.Hours
      FROM                 PROJECT P, WORKS_ON W, EMPLOYEE E
      WHERE                P.Pnumber = W.Pno AND W.Essn = E.Ssn;
```

# 테이블의 대량 로딩

- 또 다른 변형끼워 넣다에 사용됩니다**대량 적재** 여러 튜플을 테이블로
- 새로운 테이블**새로운T**와 동일한 속성을 사용하여 생성할 수 있습니다.**좋다** 그리고**데이터구문**에서는 전체 데이터를 로드할 수 있습니다.

- 예:

**테이블 생성**디5엠피에스  
(**선택하다**  
**에서**  
**어디**

**데이터와 함께;**

**좋다** 직원

이자형.\*

직원**처럼**이자형  
(마.디노=5)

# 테이블의 대량 로딩

```
mysql> 직원을 사용합니다.
```

```
mysql> 테이블 표시;
```

```
+-----+
-+ | 직원의 테이블 |
+-----+
-+ | 현재 부서 직원 |
| 부서 |
| 부서_직원 |
| 부서_직원_최신_날짜 || 부서_
관리자 |
| 직원 |
| 급여 |
| 제목 |
+-----+
```

```
mysql> CREATE TABLE D001EMP LIKE dept_emp; 쿼
리 성공, 영향을 받은 행 0개(0.06초) mysql> show tables;
```

```
+-----+
-+ | 직원의 테이블 |
+-----+
+-----+
| D001EMP |
| 현재_부서_직원 |
| 부서 |
| 부서_직원 |
| 부서_직원_최신_날짜 || 부서_
관리자 |
| 직원 |
| 급여 |
| 제목 |
+-----+
```

```
mysql> SELECT * FROM D001EMP;
빈 세팅 (0.00초)
```

```
mysql> INSERT INTO D001EMP (SELECT * from dept_emp);
쿼리 확인 , 영향을 받은 행 331603개(2.52초) 331603개
기록: 중복: 0개 경고: 0개
```

```
mysql> SELECT * FROM D001EMP LIMIT 5;
```

```
+-----+-----+-----+-----+
| 직원 번호 | 부서번호 | 시작일 | 종료일 |
+-----+-----+-----+-----+
| 10001 | d005 | 1986-06-26 | 9999-01-01 |
| 10002 | d007 | 1996-08-03 | 9999-01-01 |
| 10003 | d004 | 1995-12-03 | 9999-01-01 |
| 10004 | d004 | 1986-12-01 | 9999-01-01 |
| 10005 | d003 | 1989-09-12 | 9999-01-01 |
+-----+-----+-----+-----+
```

```
5줄 i N -+ 설정(0.00초)
```

```
MySQL>
```



# 삭제

- 관계에서 튜플을 제거합니다

- 포함 사항어디-삭제할 튜플을 선택하는 절
- 참조 무결성을 강제로 적용해야 합니다.
- 튜플은 다음에서만 삭제됩니다. 한 테이블한 번에 (~하지 않는 한종속~이다 참조 무결성 제약 조건에 지정됨 )
- 누락된어디-절은 다음을 지정합니다. 모든 튜플관계에 있어서는 다음과 같다 삭제됨; 그러면 테이블이 빈 테이블이 됩니다.
- 삭제되는 튜플의 수는 관계에서 조건을 만족하는 튜플의 수에 따라 달라집니다. 어디-절

# 그만큼삭제명령

- 관계에서 튜플을 제거합니다

- 포함 사항어디삭제할 튜플을 선택하는 절. 삭제되는 튜플의 수는 다양합니다.

U4A: DELETE FROM  
WHERE

EMPLOYEE  
Lname = 'Brown';

→ U4B: DELETE FROM  
WHERE

EMPLOYEE  
Ssn = '123456789';

U4C: DELETE FROM  
WHERE

EMPLOYEE  
Dno = 5;

→ U4D: DELETE FROM

EMPLOYEE;

# 그만큼삭제명령

```
mysql> select * from employees limit 5;
```

```
+-----+-----+-----+-----+-----+-----+ | 사  
원 번호 | 생년월일 | 이름 | 성 | 성별 | 입사일 |  
+-----+-----+-----+-----+-----+-----+ |  
10001 | 1953-09-02 | 조지 10002 | | | |  
1964-06-02 | 베잘렐 10003 | 1959-12-03 | 짐멜 | 여 | 1985-11-21 |  
| 파르토 10004 | 1954-05-01 | 크리스티안 | 뱀포드 | 엠 | 1986-08-28 |  
10005 | 1955-01-21 | 교이치 | 코블릭 | 엠 | 1986-12-01 |  
| | | | 말리니악 | 엠 | 1989-09-12 |  
+-----+-----+-----+-----+-----+-----+ 세트  
당 5개 행(0.01초)
```

```
mysql> delete from employees where first_name = 'Georgi'; 쿼리 성공,  
253개 행이 영향을 받음(0.25초)
```

```
mysql> select * from employees limit 5;
```

```
+-----+-----+-----+-----+-----+-----+ | 사  
원 번호 | 생년월일 | 이름 | 성 | 성별 | 입사일 |  
+-----+-----+-----+-----+-----+-----+ |  
10002 | 1964-06-02 | 베잘렐 10003 | | | |  
1959-12-03 | 파르토 10004 | 1954-05-01 | 짐멜 | 여 | 1985-11-21 |  
| 크리스티안 10005 | 1955-01-21 | 교이치 | 뱀포드 | 엠 | 1986-08-28 |  
10006 | 1953-04-20 | 아네케 | 코블릭 | 엠 | 1986-12-01 |  
| | | | 말리니악 | 엠 | 1989-09-12 |  
| | | | 가격 | 여 | 1989-06-02 |  
+-----+-----+-----+-----+-----+-----+ |
```

# 업데이트

- 하나 이상의 선택된 튜플의 속성 값을 수정하는 데 사용됩니다.
- 에이더디-절은 수정할 튜플을 선택합니다.
- 추가세트-절은 수정할 속성과 해당 속성의 새 값을 지정합니다.
- 각 명령은 튜플을 수정합니다. *같은 관계에서*
- DDL 사양의 일부로 지정된 참조 무결성이 적용됩니다.

# 업데이트 (계속)

- 예: 프로젝트 번호 10의 위치 및 제어 부서 번호를 각각 'Bellaire' 및 5로 변경합니다.

U5: 업데이트    프로젝트  
                  세트

위치 = '벨레어', DNUM = 5

어디

번호=10

# 업데이트 (계속)

- 예: '연구' 부서의 모든 직원에게 급여를 10% 인상합니다.

U6: 업데이트 직원  
세트 급여 = 급여 \* 1.1 DNO IN  
어디 ( SELECT 디번호  
에서 부서  
어디 DNAME='연구')

- 이 요청에서는 수정된샐러리가치는 원본에 따라 달라집니다샐러리각 튜플의 값
  - 참조샐러리= 오른쪽의 속성은 이전을 참조합니다. 샐러리수정 전 값
  - 참조샐러리= 왼쪽의 속성은 새로운 것을 참조합니다. 샐러리수정 후의 값

# 업데이트 (계속)

```
mysql> select * from salaries limit 5;
```

```
+-----+-----+-----+-----+
----+ | 직원 번호 | 급여 | 시작일 | 종료일 |
+-----+-----+-----+-----+
      10002 |    65828 | 1996-08-03 | 1997-08-03 | 65909 |
| 10002 |    1997-08-03 | 1998-08-03 | 67534 |
| 10002 |    1998-08-03 | 1999-08-03 | 69366 |
| 10002 |    1999-08-03 | 2000-08-02 | 71963 |
| 10002 |    2000-08-02 | 2001-08-02 |
+-----+-----+-----+ 세트당 5개
행(0.00초)
```

```
mysql> UPDATE 급여 SET 급여 = 급여 * 1.1 WHERE emp_no =
10002;
```

쿼리 확인, 영향을 받은 행 6개(0.00초) 일치하는 행: 6 번  
경됨: 6 경고: 0

```
mysql> select * from salaries limit 5;
```

```
+-----+-----+-----+-----+
----+ | 직원 번호 | 급여 | 시작일 | 종료일 |
+-----+-----+-----+-----+
      10002 |    72411 | 1996-08-03 | 1997-08-03 | 72500 |
| 10002 |    1997-08-03 | 1998-08-03 | 74287 |
| 10002 |    1998-08-03 | 1999-08-03 | 76303 |
| 10002 |    1999-08-03 | 2000-08-02 | 79159 |
| 10002 |    2000-08-02 | 2001-08-02 |
+-----+-----+-----+-----+
```

# SQL의 추가 기능

- 지정을 위한 기술 **복잡한 검색 쿼리** (7장 참조)
- SQL 문을 포함하는 다양한 프로그래밍 언어로 프로그램 작성: **임베디드 및 동적 SQL, SQL/CLI** (Call Level Interface) 및 이전 버전 **오에스디씨(ODBC)**, **SQL/PSM** (영구 저장 모듈) (Ch.10 참조)
- 예를 들어, 물리적 데이터베이스 설계 매개변수, 관계에 대한 파일 구조 및 액세스 경로를 지정하기 위한 명령 세트입니다. **인덱스 생성**



# SQL의 추가 기능(계속)

- 거래 제어 명령(Ch.20)
- 허가 지정 (**승인하다**) 및 권한 박탈 (취소)사용자에게 (Ch.30)
- 트리거 생성을 위한 구성 요소(Ch.26)
- 객체 관계형으로 알려진 향상된 관계형 시스템은 관계를 클래스로 정의합니다.  
추상 데이터 유형(사용자 정의 유형-UDT라고 함)은 다음과 함께 지원됩니다.  
유형 생성

# 요약

- **SQL**

- 관계형 데이터베이스 관리를 위한 포괄적인 언어
- 데이터 정의, 쿼리, 업데이트, 제약 조건 지정 및 뷰 정의
- <https://dev.mysql.com/doc/refman/8.0/en/sql-syntax.html>

- **포함 :**

- 테이블 생성을 위한 데이터 정의 명령
- 제약 조건 지정을 위한 명령
- 간단한 검색 쿼리
- 데이터베이스 업데이트 명령