

데이터 베이스 3장

SQL 개요

관계대수는 알겠다 하지만 SQL문으로?

└─ 컴퓨터에게 무엇을 가져와 줄 수 있니를 묻는거랑
어떠한 일을 시키는 것이랑의 차이!

▶ SQL은?

비절차적 언어(선언적 언어)이므로 사용자는 자신이 원하는 데이터(**what**)만
명시하며, 원하는 것을 처리하는 방법(**how**)을 명시할 수 없다.

▶ 내포된 SQL -> 기존 SQL과는 달리 C++, java, python등의 고급 프로그래밍 언어 (호스트언어)내에 SQL을 포함하여 사용하는 방식

	SQL	관계대수
언어	비절차적 언어 : 사용자는 자신이 무엇을 원하는지만 질의에 명시하고 , DBMS 는 이 질의를 효율적으로 처 리하는 방법을 결정한다.	절차적 언어 : 관계 대수식은 관계 연산자들이 수행되는 순 서를 명시한다.
연산	데이터베이스 구조 및 제약조건을 추가 하거 나 수정하는 명령(DDL) 투플을 검색 삭제, 수정하는 명령(DML)	검색 연산만 제공

데이터 조작어

예제로 다시 알아보시다.

- ▶ 예제를 보고 이 문장이 무엇을 뜻하는지 알아보시다.

UPDATE DEPT **SET** FLOOR = 10 **WHERE** DEPTNO = 1;

DELETE FROM DEPT **WHERE** DEPTNAME = 총무;

INSERT INTO DEPT **VALUES**(5, '연구', 9);

데이터 정의어

- ▶ 예제를 보고 맞춰봅시다.

```
DROP TABLE DEPT;
```

```
ALTER TABLE DEPT DROP COLUMN LOC;
```

```
ALTER TABLE DEPT ADD SNAME VARCHAR2(4) NULL, FLOOR CHAR(2) NOT NULL;
```

```
ALTER TABLE DEPT ALTER COLUMN FLOOR NUMBER(2) NULL;
```

데이터 정의어(2)

```
CREATE TABLE EMP(  
    EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,  
    ENAME VARCHAR2(10),  
    JOB VARCHAR2(9),  
    MGR NUMBER(4),  
    HIREDATE DATE,  
    SAL NUMBER(7,2),    (최대 자릿수, 최대 소수점 이하 자릿수)  
    COMM NUMBER(7,2),  
    DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT);
```

데이터 제어어와 정의어 종류

- ▶ 데이터 제어어 : GRANT, REVOKE
- ▶ 정의어의 종류들

CREATE	DOMAIN	도메인을 생성
	TABLE	테이블을 생성
	VIEW	뷰를 생성
	INDEX	인덱스를 생성
ALTER	TABLE	테이블의 구조를 변경
DROP	DOMAIN	도메인을 제거
	TABLE	테이블을 제거
	VIEW	뷰를 제거
	INDEX	인덱스를 제거

제약조건

▶ NOT NULL

애트리뷰트에 대해 디폴트 값을 지정하지 않았으면 **INSERT**문에서 이 애트리뷰트에 반드시 값을 입력해야 한다.

밑의 예시들은 **CREATE TABLE** 테이블명 (이 사이에 들어가는것) 입니다.

▶ UNIQUE

동일한 애트리뷰트 값을 갖는 튜플이 두 개 이상 존재하지 않도록 보장

ENAME VARCHAR2(4) **UNIQUE**,

▶ DEFAULT

애트리뷰트에 널값대신 특정값을 디폴트값으로 지정

TITLE CHAR(10) **DEFAULT** '사원',

▶ CHECK

애트리뷰트가 가질 수 있는 값들의 범위 지정

DNO NUMBER **CHECK** (DNO IN (10,20,30,40,50)) **DEFAULT** 1,

제약조건 (2)

- ▶ PRIMARY KEY (EMPNO),
- ▶ FOREIGN KEY (MANAGER) REFERENCES EMP(EMPNO),
- ▶ FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO)
ON DELETE CASCADE
(OR NO ACTION, SET NULL, SET DEFAULT

SELECT, FROM, WHERE

▶ SELECT

- ▶ 선택트는 프로젝션 (π) 연산!

▶ FROM

- ▶ 프롬은 어떤 테이블이 대상인지 명시

▶ WHERE

- ▶ 웨어절은 실렉션 (σ) 연산!
- ▶ 비교연산자 =, <>, <, <=, >, >=
- ▶ 부울연산자 AND, OR, NOT
- ▶ 집합연산자 IN, NOT IN, ANY, ALL

GROUP BY, HAVING, ORDER BY, 별칭

- ▶ **GROUP BY**는 명시된 애트리뷰트에 동일한 값을 갖는 튜플들을 그룹으로 묶는다.
SELECT문에는 **GROUP BY**에서 명시된 애트리뷰트 & 그룹함수만 사용이 가능하다.
- ▶ **HAVING**절은 튜플들의 그룹이 만족해야 하는 조건을 나타낸다.
- ▶ **ORDER BY**는 정렬 순서를 지정한다. 생략하면 순서는 기본키의 순서나 튜플들이 검색된 순서를 따른다.
- ▶ 별칭(**ALIAS**)
 - : 서로 다른 릴레이션에 이름이 같은 애트리뷰트가 있다면??
 - 1) EMP.DEPTNO, DEPT.DEPTNO
 - 2) FROM EMP AS E, DEPT AS D 혹은 EMP E, DEPT D그리고 사용법은 E.DEPTNO, D.DEPTNO

SELECT절 심화

DISTINCT절 : 중복된 튜플을 없앤다. (**SELECT DISTINCT TITLE ~**)

LIKE비교 연산자 : 문자열의 일부에 대하여 비교조건을 명시한다.

wild키워드 '%'는 0개 이상의 문자열과 대치

wild키워드 '_'는 임의의 1개의 문자와 대치 (**LIKE**의 반대 키워드는??)

IN 비교 연산자 : 리스트 내의 값과 비교 (**IN**의 반대 키워드도?)

Select절에서 산술연산자(+,-,*,/)사용해도 실제 **DB**의 값이 바뀌는것은 아님!

문제1. **NULL**은 어떻게 비교할까요?

문제2. **ORDER BY 1, 2**는 무엇을 의미?

문제3. 집단 함수(**COUNT,AVG,SUM,MAX,MIN**)는 **WHERE**에서 사용가능?

문제4. **HAVING**절에 나타나는 애트리뷰트는 반드시 **GROUP BY**절에 나타나거나

집단 함수에 포함되어야 사용 가능?

집합연산

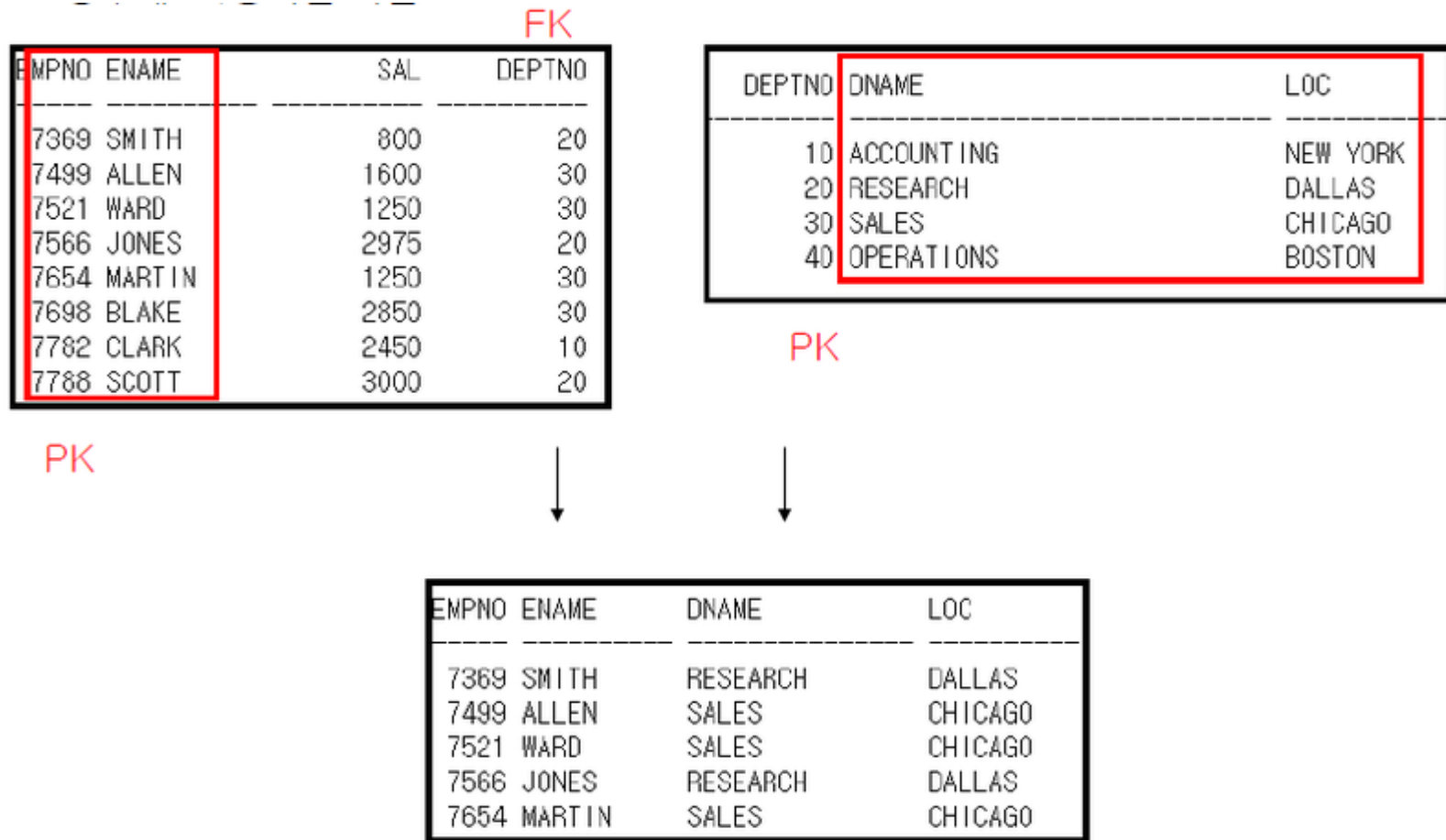
- ▶ 두 릴레이션이 합집합 호환성을 가져야 한다.
합집합 호환성은 무엇일까?
- ▶ 예시 :

```
SELECT DEPTNO  
FROM EMP  
WHERE ENAME = '김창섭'  
UNION  
SELECT DEPTNO  
FROM DEPT  
WHERE DNAME = '개발'
```

조인

- ▶ 아까 별명(alias)을 릴레이션에 쓸 수 있었죠? 그걸 기억해봅시다.
- ▶ 조인 질의 하는것을 맞춰볼까요?

이전에 우리가 관계대수에서 배웠던 조인을 떠올리면서 아래의 테이블을 유추하는 방법을 생각합시다



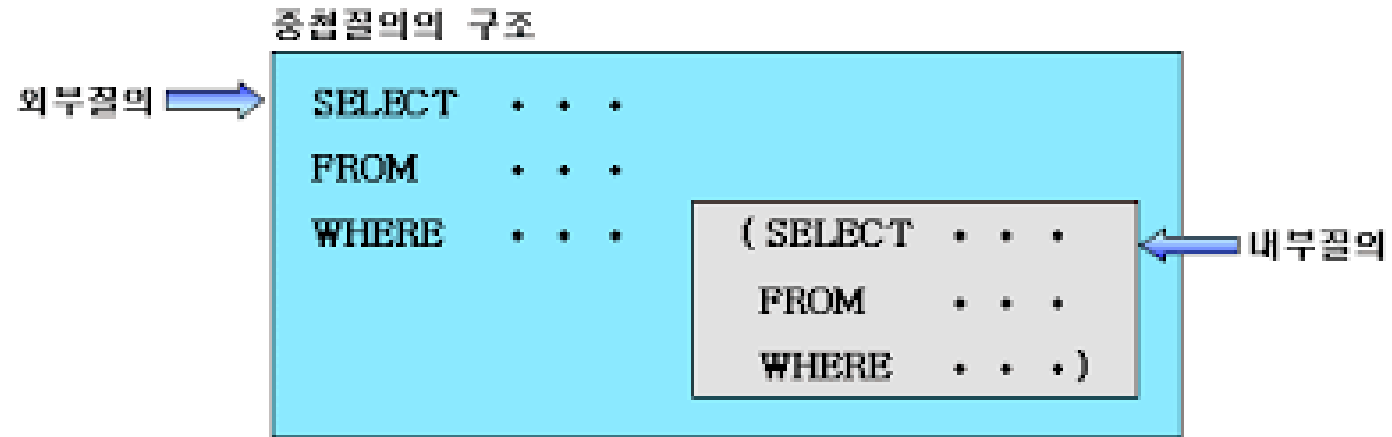
힌트와 정답

```
SELECT E.EMPNO, E.ENAME, D.DNAME, D.LOC  
FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO
```

힌트1.
EMP와 DEPT의 카티션의 곱은
FROM EMP, DEPT 로 합니다.

힌트2.
관계대수에서 동등조인 할때
어떻게 조건을 줘을까요?

중첩 질의



※ 내부질의의 **SELECT**절에 사용된 애트리뷰트의 데이터 타입은
외부질의의 **WHERE**절에 사용된 애트리뷰트의 데이터 타입과 같아야함.

중첩 질의(2)

예 : IN을 사용한 질의

질의: 영업부나 개발부에 근무하는 직원들의 이름을 검색하라.

```
SELECT EMPNAME  
FROM EMPLOYEE  
WHERE DNO IN
```

(1, 3)

```
(SELECT DEPTNO  
FROM DEPARTMENT  
WHERE DEPTNAME = '영업' OR DEPTNAME = '개발') ;
```

이 질의를 중첩 질의를 사용하지 않은 다음과 같은 조인 질의로 나타낼 수 있다. 실제로, 중첩 질의를 사용하여 표현된 대부분의 질의를 중첩 질의가 없는 조인 질의로 표현할 수 있다. 그러나 어떤 질의는 중첩 질의를 포함한 질의로만 표현할 수 있다. 일반적으로 중첩 질의를 포함한 질의가 좀더 읽기 쉽다.

```
SELECT EMPNAME  
FROM EMPLOYEE E, DEPARTMENT D  
WHERE E.DNO = D.DEPTNO  
AND (D.DEPTNAME = '영업' OR D.DEPTNAME = '개발') ;
```

EMPNAME
박영권
이수민
최종철
김상원

중첩질의(3)

- ▶ 아까의 사진에서 IN 키워드 말고도 ANY, ALL 같은 키워드가 있을 수 있습니다.

IN : 3426 IN (2106,3426,3011) IS TRUE

1365 IN (2106,3426,3011) IS FALSE

퀴즈1. 1365 NOT IN (2106,3426,3011) ??

ANY : 30 < ANY (25,30,40) IS TRUE

퀴즈2. 40 < ANY (25, 30, 40)??

ALL: 30 < ALL (25, 30, 40) IS FALSE

15 < ALL (25, 30, 40) IS TRUE

30 = ALL (25, 30, 40) IS FALSE

퀴즈3. 15 <> ALL (25, 30, 40) ??

저번 퀴즈에 나왔던 SELECT문의 실행순서

SELECT E1.ENAME, SUM(E2.SAL)	--5
FROM EMP E1, DEPT, EMP E2	--1
WHERE E1.DNO = DEPT.DNO AND E2.ENAME = DEPT.MGR	--2
GROUP BY E1.ENAME	--3
HAVING COUNT(*) > 1	--4
ORDER BY ENAME;	--6

1. 튜플들을 구하고 (카티션의 곱)
2. 조건들을 만족하는 튜플들을 식별하고
3. 그룹들을 구하고
4. **HAVING**을 적용하여 일부 그룹들을 제거하고
5. 집단 함수의 값을 구하고
6. 결과 튜플들을 정렬하여 사용자에게 제시한다.