

데이터베이스관리

13주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] SELECT

(fruit)

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습] 기본 예제(quiz_fruit.sql)

-- fruit(과일번호,과일이름,가격,색상, 원산지)

[실습예제]

데이터베이스이름: fruit

테이블이름: 과일

테이블 이름	열 이름	데이터 형식	제약조건	기본키	외래키	FK테이블	FK열
과일	과일번호	int	NOT NULL	PK			
	과일이름	varchar(30)					
	가격	int					
	색상	char(12)					
	원산지	varchar(30)					

과일

과일번호	과일이름	가격	색상	원산지
1	사과	2000	빨간색	한국
2	배	5000	노란색	한국
3	바나나	1500	노란색	인도네시아
4	키위	4000	초록색	호주
5	수박	8000	초록색	한국

-- [기본예제 1]

-- fruit(과일번호,과일이름,가격,색상, 원산지)

-- 1-1) 노란색 과일의 이름, 색상을 검색하라.

-- 1-2) 과일이름을 오름차순으로 정렬하여 모든 필드를 검색하라

-- 1-3) 가격이 5000 이하인 빨간색 과일의 이름을 검색하라

-- 1-4) 가격이 2000 ~ 5000 사이의 과일 이름과 가격, 원산지를 검색하라(BETWEEN AND)

-- 1-5) 색상이 노란색이거나 초록색인 과일이름, 색상을 검색하라(IN 사용)

-- 1-6) 색상이 노란색이거나 초록색인 과일이름, 색상을 검색하라(논리연산자 OR 사용)

-- 1-7) 원산지가 한국이 아닌 과일이름, 원산지를 검색하라

[실습] 기본 예제(quiz_fruit.sql)

-- fruit(과일번호,과일이름,가격,색상, 원산지)

[실습예제]

데이터베이스이름: fruit

테이블이름: 과일

테이블 이름	열 이름	데이터 형식	제약조건	기본키	외래키	FK테이블	FK열
과일	과일번호	int	NOT NULL	PK			
	과일이름	varchar(30)					
	가격	int					
	색상	char(12)					
	원산지	varchar(30)					

과일

과일번호	과일이름	가격	색상	원산지
1	사과	2000	빨간색	한국
2	배	5000	노란색	한국
3	바나나	1500	노란색	인도네시아
4	키위	4000	초록색	호주
5	수박	8000	초록색	한국

-- [기본예제 2]

-- fruit(과일번호,과일이름,가격,색상, 원산지)

-- 2-1) UPDATE 문을 이용하여 사과의 가격을 2배로 하여 수정하라

-- 2-2) DELETE 문을 이용하여 원산지가 호주인 과일의 목록을 삭제하라

-- 3-1) ALTER TABLE 문을 이용하여 과일정보 필드를 추가하라(데이터형식 varchar(20))



[Quiz] 추가예제

fruit

과일번호	과일이름	가격	색상	원산지
1	사과	2000	빨간색	한국
2	배	5000	노란색	한국
3	바나나	1500	노란색	NULL
4	키위	4000	초록색	호주
5	수박	8000	초록색	한국
6	오렌지	1000	주황색	한국
7	망고	5000	노란색	페루
8	체리	4800	빨간색	중국
9	포도	2500	보라색	한국
10	아보카도	2000	초록색	남아프리카
11	모과	5000	노란색	NULL
12	레몬	6000	노란색	미국

-- 4-1) 원산지를 알 수 없는 과일의 과일이름, 원산지를 검색하라

-- 4-2) 과일 가격의 내림차순으로 정렬하는데,
-- 가격이 같은 경우, 원산지의 오름차순 정렬로 검색하라

-- 4-3) 원산지가 국으로 끝나는 과일이름, 원산지를 검색하라

-- 4-4) 원산지가 인으로 시작하는 과일이름, 원산지를 출력하라

-- 4-5) 원산지별로 원산지와 개수를 검색하라

-- 4-6) 색상별로 색상이 3개 이상인 과일만 색상, 과일개수를 검색하라

-- 단, 과일개수가 많은 것부터 나타나도록 검색하라

-- 5-1) UPDATE 문을 이용하여 빨간색 과일 가격을 10% 인상하라

-- 5-2) DELETE 문을 이용하여

-- 원산지가 호주인 과일의 목록을 삭제하라

SQL: 중첩 질의(sub query)

13주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습] 하위질의



[실습]

```
-- department(deptno, deptname, floor)
-- employee(empno, empname, title, manager, salary, dno)
```

-- **노트**에 SQL문법을 작성하시오

-- 20) 하위질의(= 사용)

-- 21) 하위질의(IN 사용)

deptno	deptname	floor
1	영업	8
2	기획	10
3	개발	9
4	총무	7

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

-- 20) 박영권과 같은 직급을 갖는 모든 직원들의 이름과 직급을 검색하라

-- 21) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라(IN 사용)

[요약] 조인

- ❖ 내부조인(theta join, equi join, natural join)
- ❖ 외부조인(left outer join, right outer join, full outer join)

1. 내부조인(동등조인)

- 1) SELECT .. FROM .. WHERE 방법
- 2) SELECT .. FROM .. ON 방법

2. 내부조인(자연조인)

- 1) SELECT .. FROM .. WHERE 방법
- 2) SELECT .. FROM .. ON 방법

[요약] 중첩질의

-- 1. 스칼라 질의

```
SELECT ( SELECT ... FROM ... WHERE ... )  
FROM  
WHERE
```

-- 2. 인 라인 뷰(inline view)

```
SELECT  
FROM ( SELECT ...  
      FROM ...  
      WHERE ...  
      ) as T  
WHERE
```

- ❖ 하위질의(Sub query)
- ❖ 서브쿼리 (Sub query, 부질의)
- ❖ 중첩 질의
- ❖ 상관중첩질의

-- 3. 하위 질의

```
SELECT  
FROM  
WHERE ... ( SELECT ...  
            FROM ...  
            WHERE ... )
```

[요약] 중첩질의(서브쿼리, 부질의)

- ❖ **ALL**: 하위질의의 모든 결과에 대해 조건 수식이 참인 경우 결과를 반환
- ❖ **SOME**: 하위질의의 한 개 이상의 결과에 대해 조건 수식이 참인 경우 결과를 반환
- ❖ **ANY**: **SOME** 과 동일

- **ANY** 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- **ALL** 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다

IN 는 =ANY 와 동일,
NOT IN 은 <>ALL 과 동일

SOME 은 ANY 와 동일한 의미
=ANY 는 IN 과 동일한 의미

[실습] 조인

[실습]

```
-- department(deptno, deptname, floor)
```

```
-- employee(empno, empname, title, manager, salary, dno)
```

deptno	deptname	floor
1	영업	8
2	기획	10
3	개발	9
4	총무	7

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

- 17) 사원의 이름과 이 사원이 속한 부서이름을 검색하라 (조인)
 - 19) 사원에 대해서 부서이름, 사원이름, 직급, 급여를 검색하라.
 - 부서이름에 대해서 오름차순, 부서이름이 같을 경우에는 salary에 대해서 내림차순으로 정렬하라
-
- 18) 사원에 대해서 사원이름과 직속상사의 이름을 검색하라(Self-Join)

[실습] 하위질의

[실습]

-- department(deptno, deptname, floor)

-- employee(empno, empname, title, manager, salary, dno)

deptno	deptname	floor
1	영업	8
2	기획	10
3	개발	9
4	총무	7

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

-- 20) 박영권과 같은 직급을 갖는 모든 사원들의 이름과 직급을 검색하라

-- 21) 영업부나 개발부에 근무하는 사원들의 이름을 검색하라(IN 사용)

-- 21-02) 영업부나 개발부에 근무하지 않는 사원들의 이름을 검색하라(NOT IN 사용)

-- 24) 사원들이 한 명도 소속되지 않은 부서명을 검색하라 (NOT EXISTS 사용)

[요약] 하위질의

- ❖ 하위질의(Sub query)
- ❖ 서브쿼리 (Sub query, 부질의)
- ❖ 중첩 질의
- ❖ 상관중첩질의

```
-- (하위질의)
-- 20) 박영권과 같은 직급을 갖는
-- 모든 사원들의 이름과 직급을 검색하라
```

[실습]

```
-- department(deptno, deptname, floor)
```

```
-- employee(empno, empname, title, manager, salary, dno)
```

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

```
SELECT empname, title
FROM employee
WHERE title = (SELECT title
                FROM employee
                WHERE empname = '박영권');
```

[요약] 하위질의

-- 21) 영업부나 개발부에 근무하는 직원들의 이름을 검색하라(IN 사용)

[실습]

-- department(deptno, deptname, floor)

-- employee(empno, empname, title, manager, salary, dno)

deptno	deptname	floor
1	영업	8
2	기획	10
3	개발	9
4	총무	7

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

```
SELECT empname
FROM employee
WHERE dno IN (SELECT deptno
              FROM department
              WHERE deptname IN ('영업','개발'));
```

```
SELECT empname
FROM employee, department
WHERE employee.dno = department.deptno
      AND (deptname = '영업' OR deptname = '개발');
```

[요약] 하위질의

-- 24) 사원들이 **한 명도 소속되지 않은**
부서명을 검색하라 (NOT EXISTS 사용)

[실습]

-- department(deptno, deptname, floor)

-- employee(empno, empname, title, manager, salary, dno)

deptno	deptname	floor
1	영업	8
2	기획	10
3	개발	9
4	총무	7

empno	empname	title	manager	salary	dno
1003	조민희	과장	4377	3000000	2
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
3011	이수민	부장	4377	4000000	3
3426	박영권	과장	4377	3000000	1
3427	최종철	사원	3011	1500000	3
4377	이성래	이사	NULL	5000000	2

```
select deptname
from department D
where NOT EXISTS (select *
                  from employee E
                  where D.deptno = E.dno);
```

[실습] 하위질의

```
-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)
```

```
CREATE TABLE pro (
  pno char(2),
  pname varchar(20),
  dept varchar(20),
  page int,
  primary key(pno)
);
```

```
CREATE TABLE stu (
  sno char(2),
  sname varchar(20),
  dept varchar(20),
  sage int,
  primary key(sno)
);
```

stu

sno	sname	dept	sage
s1	유준호	컴퓨터	23
s2	오정민	컴퓨터	34
s3	이태현	건축	22
s4	신현주	건축	21

[실습] 하위질의(subquery.sql)

```
insert into pro values('p1','이정무','컴퓨터',36);
insert into pro values('p2','우태하','컴퓨터',32);
insert into pro values('p3','이성민','건축',45);
```

```
insert into stu values('s1','유준호','컴퓨터',23);
insert into stu values('s2','오정민','컴퓨터',34);
insert into stu values('s3','이태현','건축',22);
insert into stu values('s4','신현주','건축',21);
```

```
select * from pro;
select * from stu;
```

prof

pno	pname	dept	page
p1	이정무	컴퓨터	36
p2	우태하	컴퓨터	32
p3	이성민	건축	45

[요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

-- (하위질의)

-- 3-1) 교수 테이블에서 이정무 의 학과와 같은 학생 이름, 학과, 나이를 검색하라

-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)

stu	sno	sname	dept	sage
	s1	유준호	컴퓨터	23
	s2	오정민	컴퓨터	34
	s3	이태현	건축	22
	s4	신현주	건축	21

prof	pno	pname	dept	page
	p1	이정무	컴퓨터	36
	p2	우태하	컴퓨터	32
	p3	이성민	건축	45

[요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

-- (하위질의)

-- 3-1) 교수 테이블에서 이정무 의 학과와 같은 학생 이름, 학과, 나이를 검색하라

-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)

stu	sno	sname	dept	sage
	s1	유준호	컴퓨터	23
	s2	오정민	컴퓨터	34
	s3	이태현	건축	22
	s4	신현주	건축	21

prof	pno	pname	dept	page
	p1	이정무	컴퓨터	36
	p2	우태하	컴퓨터	32
	p3	이성민	건축	45

[요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

```
-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)
```

stu	sno	sname	dept	sage
	s1	유준호	컴퓨터	23
	s2	오정민	컴퓨터	34
	s3	이태현	건축	22
	s4	신현주	건축	21

prof	pno	pname	dept	page
	p1	이정무	컴퓨터	36
	p2	우태하	컴퓨터	32
	p3	이성민	건축	45

- (하위질의)
- 3-2) 교수 테이블에서 pno 가 p1 이거나 p2 인 학과와 같은 학생 이름, 학과, 나이를 검색하라

[요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

```
-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)
```

stu	sno	sname	dept	sage
	s1	유준호	컴퓨터	23
	s2	오정민	컴퓨터	34
	s3	이태현	건축	22
	s4	신현주	건축	21

prof	pno	pname	dept	page
	p1	이정무	컴퓨터	36
	p2	우태하	컴퓨터	32
	p3	이성민	건축	45

- ```
-- (하위질의)
-- 3-3) 학생 테이블과 교수 테이블에서 모든 학생들보다 나이가 많은 교수의
-- 교번, 이름, 나이를 ALL 구문을 이용하여 검색하라
```

```
select pno, pname, page
from pro
where page > ALL (select sage
 from stu);
```

# [요약] 하위질의

-- stu(sno, sname, dept, sage)  
-- pro(pno, pname, dept, page)

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

| stu | sno | sname | dept | sage |
|-----|-----|-------|------|------|
|     | s1  | 유준호   | 컴퓨터  | 23   |
|     | s2  | 오정민   | 컴퓨터  | 34   |
|     | s3  | 이태현   | 건축   | 22   |
|     | s4  | 신현주   | 건축   | 21   |

| prof | pno | pname | dept | page |
|------|-----|-------|------|------|
|      | p1  | 이정무   | 컴퓨터  | 36   |
|      | p2  | 우태하   | 컴퓨터  | 32   |
|      | p3  | 이성민   | 건축   | 45   |

- (하위질의)
- 3-4) 학생 테이블과 교수 테이블에서 한 명 이상 교수보다
- 나이가 많은 학생이 있을 경우 학생의 학번, 이름, 나이를 SOME 구문으로 검색하라

```
select sno, sname, sage
from stu
where sage > SOME (select page
 from pro);
```

# [요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

-- stu(sno, sname, dept, sage)  
-- pro(pno, pname, dept, page)

| stu | sno | sname | dept | sage |
|-----|-----|-------|------|------|
|     | s1  | 유준호   | 컴퓨터  | 23   |
|     | s2  | 오정민   | 컴퓨터  | 34   |
|     | s3  | 이태현   | 건축   | 22   |
|     | s4  | 신현주   | 건축   | 21   |

| prof | pno | pname | dept | page |
|------|-----|-------|------|------|
|      | p1  | 이정무   | 컴퓨터  | 36   |
|      | p2  | 우태하   | 컴퓨터  | 32   |
|      | p3  | 이성민   | 건축   | 45   |

-- (하위질의)  
-- 3-5) 가장 나이가 많은 학생 이름, 나이를 검색하라

```
select sname, dept, sage
from stu
where sage = (select MAX(sage)
 from stu);
```

# [요약] 하위질의

- ❖ ANY 서브쿼리의 여러 개의 결과 중 한가지만 만족해도 되며
- ❖ ALL 서브쿼리의 여러 개의 결과를 모두 만족시켜야 된다
- ❖ SOME 은 ANY 와 동일한 의미
- ❖ IN 는 =ANY 와 동일, NOT IN 은 <>ALL 과 동일

```
-- stu(sno, sname, dept, sage)
-- pro(pno, pname, dept, page)
```

| stu | sno | sname | dept | sage |
|-----|-----|-------|------|------|
|     | s1  | 유준호   | 컴퓨터  | 23   |
|     | s2  | 오정민   | 컴퓨터  | 34   |
|     | s3  | 이태현   | 건축   | 22   |
|     | s4  | 신현주   | 건축   | 21   |

| prof | pno | pname | dept | page |
|------|-----|-------|------|------|
|      | p1  | 이정무   | 컴퓨터  | 36   |
|      | p2  | 우태하   | 컴퓨터  | 32   |
|      | p3  | 이성민   | 건축   | 45   |

```
-- (하위질의)
-- 3-6) 각 학과마다 가장 나이가 많은 학생 이름, 학과, 나이를 검색하라
select sname, dept, sage
from stu S1
where sage IN (select MAX(sage)
 from stu S2
 where S1.dept = S2.dept);
```

# SQL: 조인(테이블 3개)

13주차

담당교수: 김희숙  
(jasmin11@hanmail.net)



## [실습] (실습 2) 내부조인

### [실습]

```
-- department(deptno, deptname, floor)
-- employee(empno, empname, title, manager, salary, dno)
```

demartment

| deptno | deptname | floor |
|--------|----------|-------|
| 1      | 영업       | 8     |
| 2      | 기획       | 10    |
| 3      | 개발       | 9     |
| 4      | 총무       | 7     |

employeee

| empno | empname | title | manager | salary  | dno |
|-------|---------|-------|---------|---------|-----|
| 1003  | 조민희     | 과장    | 4377    | 3000000 | 2   |
| 1365  | 김상원     | 사원    | 3426    | 1500000 | 1   |
| 2106  | 김창섭     | 대리    | 1003    | 2500000 | 2   |
| 3011  | 이수민     | 부장    | 4377    | 4000000 | 3   |
| 3426  | 바경ㅇ권    | 과장    | 4377    | 3000000 | 1   |
| 3427  | 최종철     | 사원    | 3011    | 1500000 | 3   |
| 4377  | 이성래     | 이사    | NULL    | 5000000 | 2   |

- 16-1) 김창섭이 속한 부서번호와 부서의 모든 부서번호를 합집합으로 검색하라 (UNION)
- 16-2) 김창섭이 속한 부서번호와 부서의 모든 부서번호를 합집합으로 검색하라 (UNION ALL)
- 17) 사원의 이름과 이 사원이 속한 부서이름을 검색하라 (조인)
- 19) 사원에 대해서 부서이름, 사원이름, 직급, 급여를 검색하라.
- 부서이름에 대해서 오름차순, 부서이름이 같을 경우에는 salary에 대해서 내림차순으로 정렬하라

# [실습] 조인(테이블 3개)

[내부조인] (테이블 3개) 학생, 수강, 과목

(3개 테이블 조인)

학생

| 학번    | 학생명 |
|-------|-----|
| 11002 | 이홍근 |
| 24036 | 김순미 |
| 30419 | 박상웅 |
| 30555 | 홍서범 |
| 31890 | 유동희 |
| 31892 | 이정무 |
| 32000 | 김윤식 |

수강

| 학번    | 과목번호  | 성적 |
|-------|-------|----|
| 11002 | CS310 | 98 |
| 11002 | CS313 | 88 |
| 24036 | CS345 | 90 |
| 30419 | CS326 | 78 |
| 31892 | CS388 | 86 |
| 32000 | CS378 | 94 |
| 31890 | CS310 | 92 |

과목

| 과목번호  | 과목명     |
|-------|---------|
| CS310 | 데이터베이스  |
| CS313 | 운영체제    |
| CS345 | 자료구조    |
| CS326 | 자바      |
| CS546 | DB프로그래밍 |
| CS378 | 멀티미디어   |
| CS388 | 웹프로그래밍  |

- 3-1) 수강한 학생의 학생명, 과목명, 성적을 검색하라(내부조인)
- 3-2) 이홍근의 학생명, 과목명, 성적을 검색하라
- 3-3) 전체 학생의 학생명, 과목명, 성적을 검색하라(외부조인)

## [실습] 조인(테이블 3개)

### [내부조인] (테이블 3개) 학생, 수강, 과목

(3개 테이블 조인)

학생

| 학번    | 학생명 |
|-------|-----|
| 11002 | 이홍근 |
| 24036 | 김순미 |
| 30419 | 박상웅 |
| 30555 | 홍서범 |
| 31890 | 유동희 |
| 31892 | 이정무 |
| 32000 | 김윤식 |

수강

| 학번    | 과목번호  | 성적 |
|-------|-------|----|
| 11002 | CS310 | 98 |
| 11002 | CS313 | 88 |
| 24036 | CS345 | 90 |
| 30419 | CS326 | 78 |
| 31892 | CS388 | 86 |
| 32000 | CS378 | 94 |
| 31890 | CS310 | 92 |

과목

| 과목번호  | 과목명     |
|-------|---------|
| CS310 | 데이터베이스  |
| CS313 | 운영체제    |
| CS345 | 자료구조    |
| CS326 | 자바      |
| CS546 | DB프로그래밍 |
| CS378 | 멀티미디어   |
| CS388 | 웹프로그래밍  |

-- 3-1) 수강한 학생의 학생명, 과목명, 성적을 검색하라(내부조인)

```
SELECT 학생명, 과목명, 성적
FROM 학생, 수강, 과목
WHERE 학생.학번 = 수강.학번
 AND 수강.과목번호 = 과목.과목번호;
```

```
SELECT 학생명, 과목명, 성적
FROM 학생 INNER JOIN 수강
ON 학생.학번 = 수강.학번
 INNER JOIN 과목
ON 수강.과목번호 = 과목.과목번호;
```

## [실습] 조인(테이블 3개)

### [내부조인] (테이블 3개) 학생, 수강, 과목

(3개 테이블 조인)

학생

| 학번    | 학생명 |
|-------|-----|
| 11002 | 이홍근 |
| 24036 | 김순미 |
| 30419 | 박상웅 |
| 30555 | 홍서범 |
| 31890 | 유동희 |
| 31892 | 이정무 |
| 32000 | 김윤식 |

수강

| 학번    | 과목번호  | 성적 |
|-------|-------|----|
| 11002 | CS310 | 98 |
| 11002 | CS313 | 88 |
| 24036 | CS345 | 90 |
| 30419 | CS326 | 78 |
| 31892 | CS388 | 86 |
| 32000 | CS378 | 94 |
| 31890 | CS310 | 92 |

과목

| 과목번호  | 과목명     |
|-------|---------|
| CS310 | 데이터베이스  |
| CS313 | 운영체제    |
| CS345 | 자료구조    |
| CS326 | 자바      |
| CS546 | DB프로그래밍 |
| CS378 | 멀티미디어   |
| CS388 | 웹프로그래밍  |

```
-- 3-2) 이홍근의 학생명, 과목명, 성적을 검색하라
SELECT 학생명, 과목명, 성적
FROM 학생, 수강, 과목
WHERE 학생.학번 = 수강.학번
 AND 수강.과목번호 = 과목.과목번호
 AND 학생명 = '이홍근';
```

```
SELECT 학생명, 과목명, 성적
FROM 학생 INNER JOIN 수강
ON 학생.학번 = 수강.학번
 INNER JOIN 과목
ON 수강.과목번호 = 과목.과목번호
WHERE 학생명 = '이홍근';
```

## [실습] 조인(테이블 3개)

### [내부조인] (테이블 3개) 학생, 수강, 과목

(3개 테이블 조인)

학생

| 학번    | 학생명 |
|-------|-----|
| 11002 | 이홍근 |
| 24036 | 김순미 |
| 30419 | 박상웅 |
| 30555 | 홍서범 |
| 31890 | 유동희 |
| 31892 | 이정무 |
| 32000 | 김윤식 |

수강

| 학번    | 과목번호  | 성적 |
|-------|-------|----|
| 11002 | CS310 | 98 |
| 11002 | CS313 | 88 |
| 24036 | CS345 | 90 |
| 30419 | CS326 | 78 |
| 31892 | CS388 | 86 |
| 32000 | CS378 | 94 |
| 31890 | CS310 | 92 |

과목

| 과목번호  | 과목명     |
|-------|---------|
| CS310 | 데이터베이스  |
| CS313 | 운영체제    |
| CS345 | 자료구조    |
| CS326 | 자바      |
| CS546 | DB프로그래밍 |
| CS378 | 멀티미디어   |
| CS388 | 웹프로그래밍  |

-- 3-3) 전체 학생의 학생명, 과목명, 성적을 검색하라(외부조인)

```
SELECT 학생명, 과목명, 성적
FROM 학생 LEFT OUTER JOIN 수강
 ON 학생.학번 = 수강.학번
 LEFT OUTER JOIN 과목
 ON 수강.과목번호 = 과목.과목번호;
```

# SQL: 실습 예제(학사)

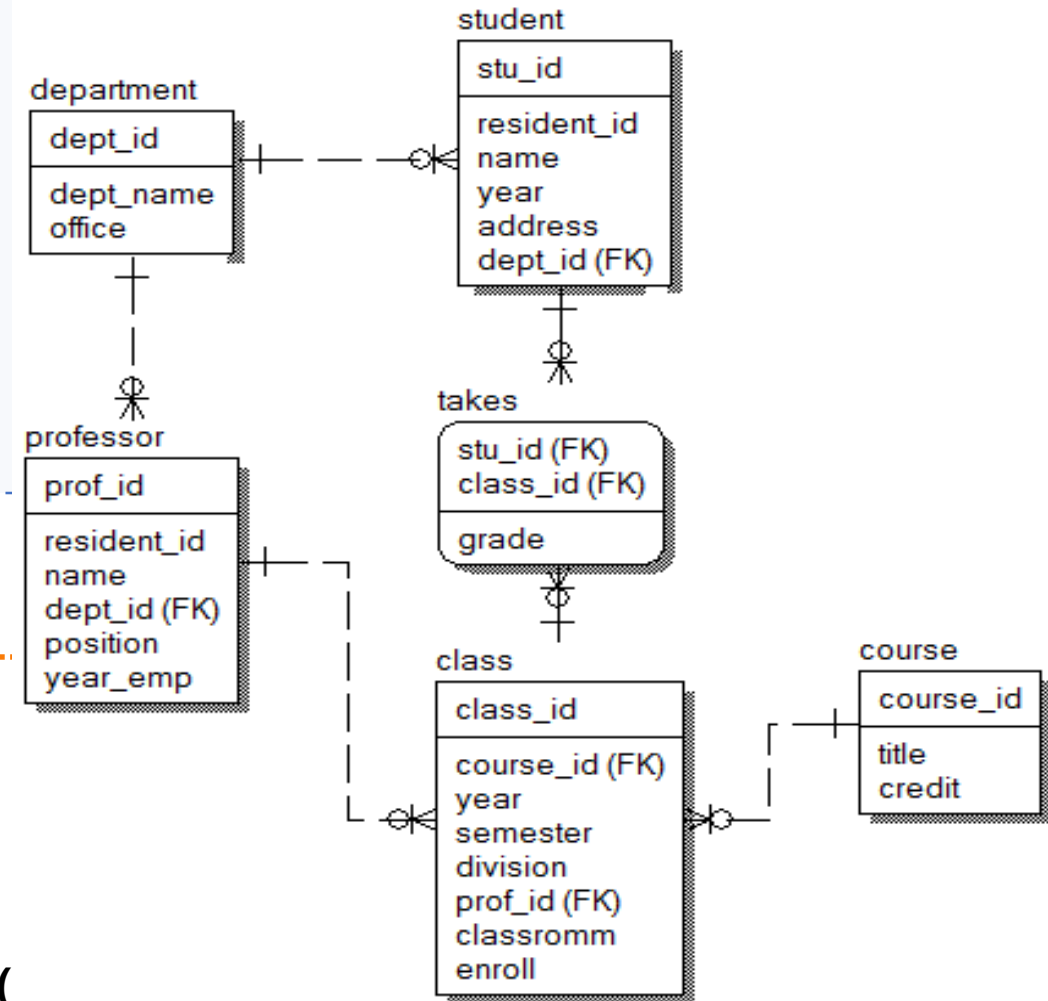
13주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```



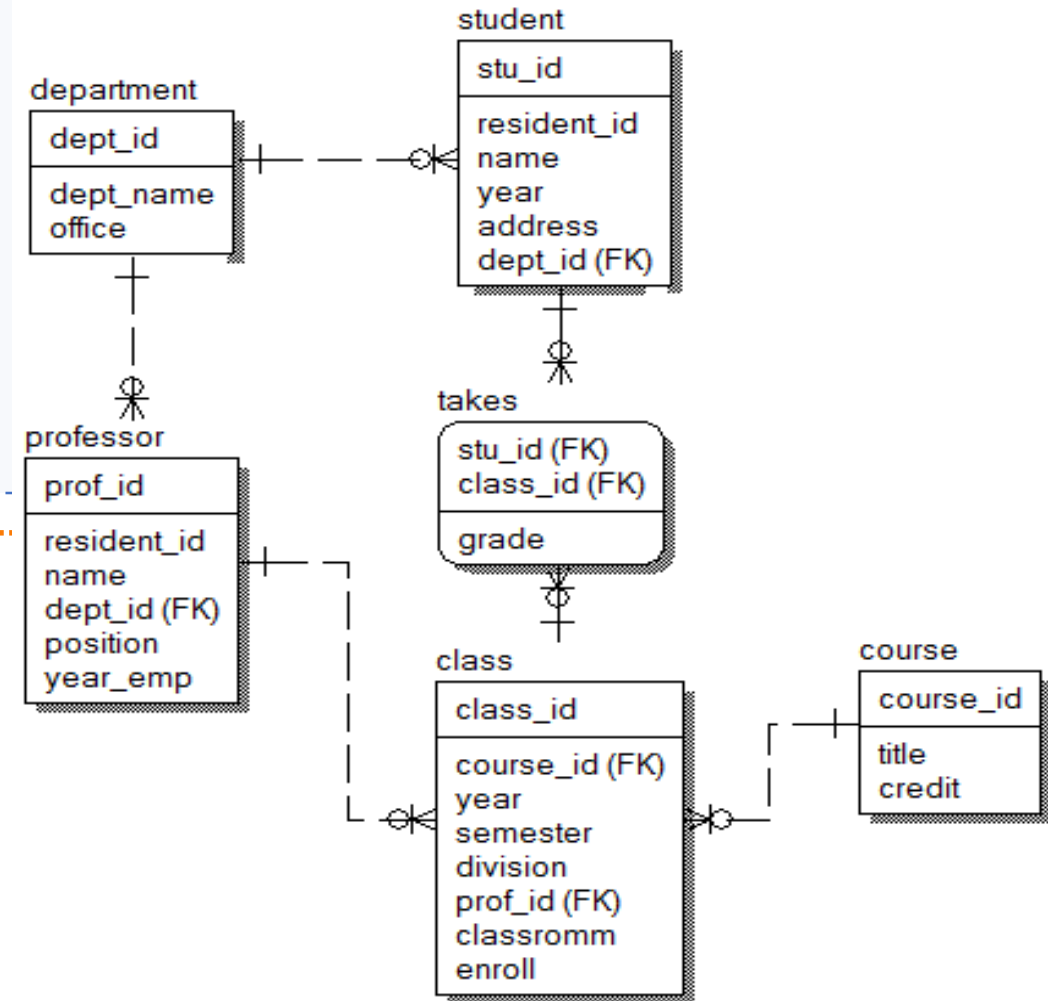
-- [예제1] 다음에 대하여 SQL문법과 실행결과를 작성하라

- 질의21) student 테이블에서 주소를 검색하라
- 질의22) student 테이블에서 주소를 검색하라(중복 제거)
- 질의23) student 테이블에서 모든 필드를 검색하라
- 질의24) professor 테이블에서 교수이름, 재직연수를 검색하라(
- 질의27) student 테이블에서 1, 2학년 학생들의 이름과 학번을 검색하라
- 단, 학생이름의 오름차순 정렬하고, 같은 이름은 학번의 오름차순 정렬하라
- 질의28) student 테이블에서 1, 2학년 학생들의 이름과 학번을 검색하라
- 단, 학생이름의 내림차순 정렬하고, 같은 이름은 학번의 오름차순 정렬하라

## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```



### -- [예제2] 조인, 집합연산

-- 질의20) 학생이름, 학과명을 검색하라

-- 방법1) SELECT ... FROM ... WHERE

-- 방법2) SELECT ... FROM ... ON

-- 질의25) 학생이름, 학번, 소속학과명을 검색하라

-- 질의26) 컴퓨터공학과 2학년 학생들의 학번을 검색하라

-- 질의30) student 테이블과 department 테이블을 조인하여 학생이름, 소속학과이름을 검색

-- 단, student 테이블은 s로, department 테이블은 d 로 재명명하라

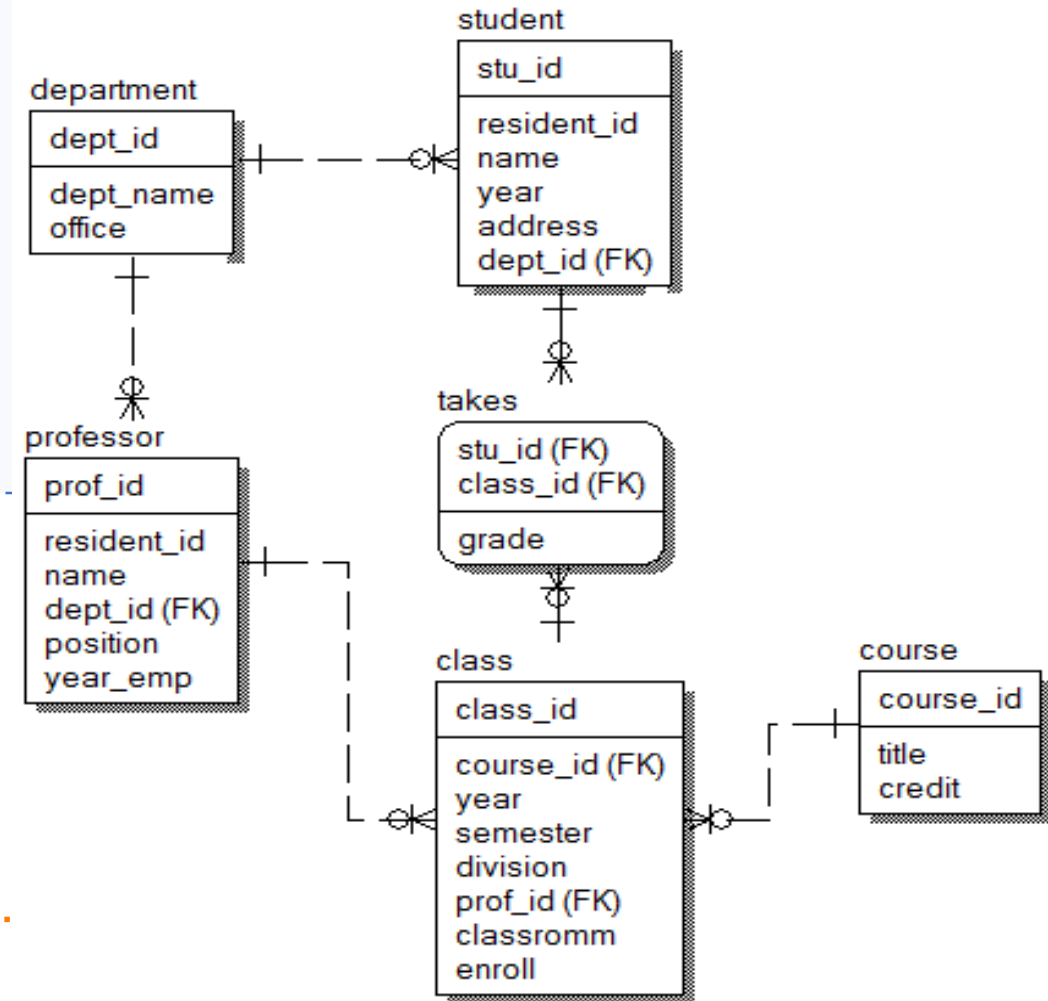
-- 질의31) student 테이블에서 김광식 학생과 주소가 같은 학생이름을 검색하라



## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```



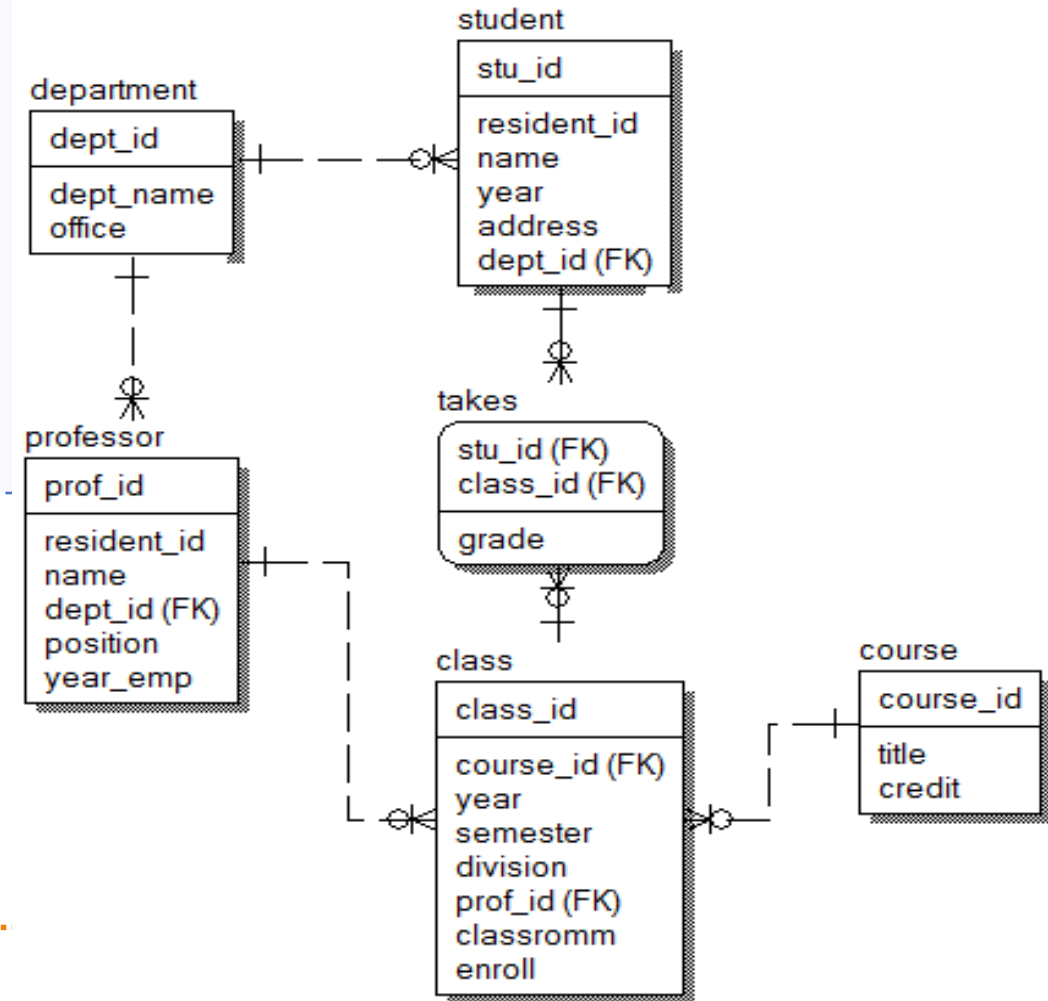
-- [예제3] 조인, 집합연산

- 질의36) student 테이블의 학생이름과 professor 테이블의 교수이름을 합집합하라
- 질의40) 컴퓨터공학과 학생들 중에서 교과목에 상관없이 학점을 A+ 받은 학생의 학번검색

## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```



-- [예제4] 외부조인

-- 질의42) 모든 교과목들에 대해 교과목명, 학점수, 개설연도, 개설학기를 검색하라

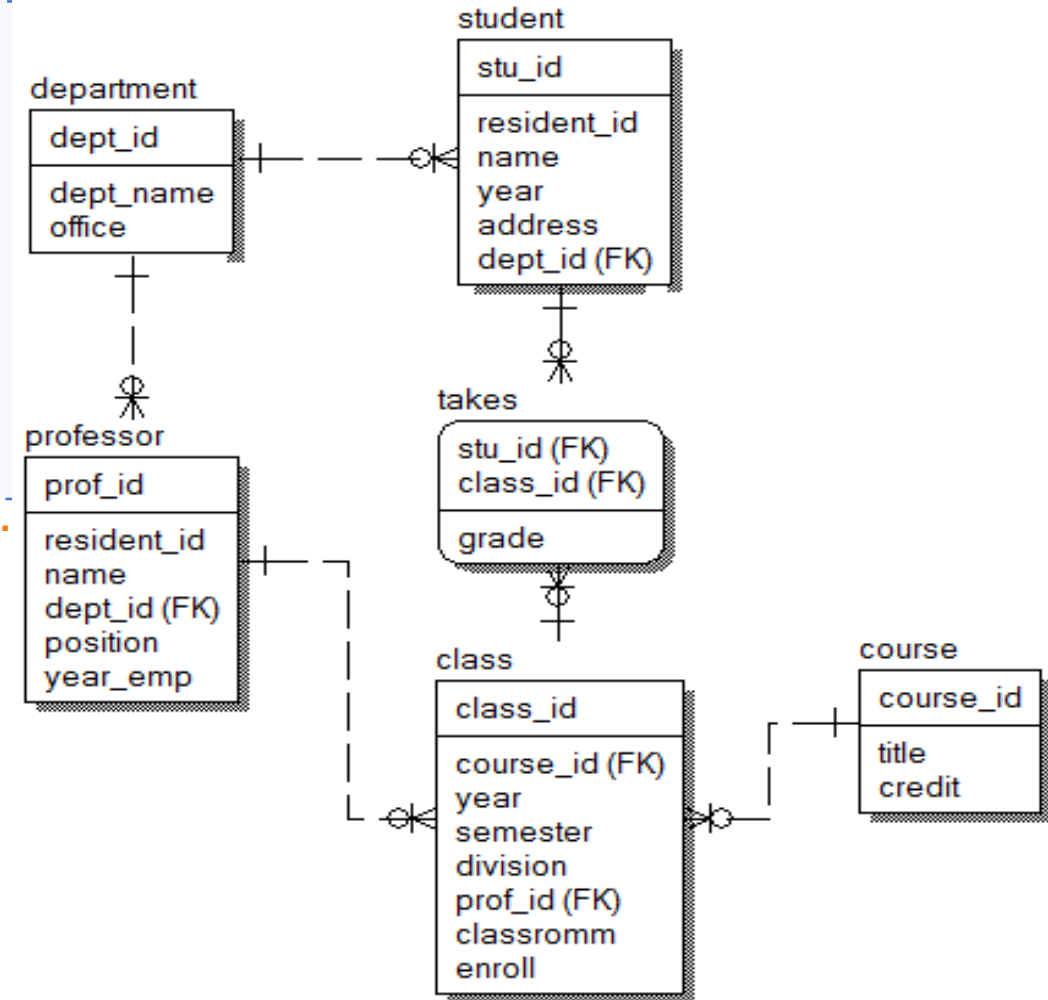
## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```

### -- [예제5] 집계함수, group by

```
-- 질의48) student 테이블에서 2학년 학생이 몇 명인지 검색하라
-- 질의49) student 테이블에서 dept_id 필드에 값이 몇 개인지 검색하라
-- 질의49) student 테이블에서 dept_id 필드에 값이 몇 개인지 검색하라
(중복 제거)
-- 질의50) 컴퓨터공학과와 학생 수를 검색하라
-- 질의51) 전체교수들의 재직연수 합을 구하라
-- 질의55) 전체 교수의 평균 재직연수를 출력하라
```



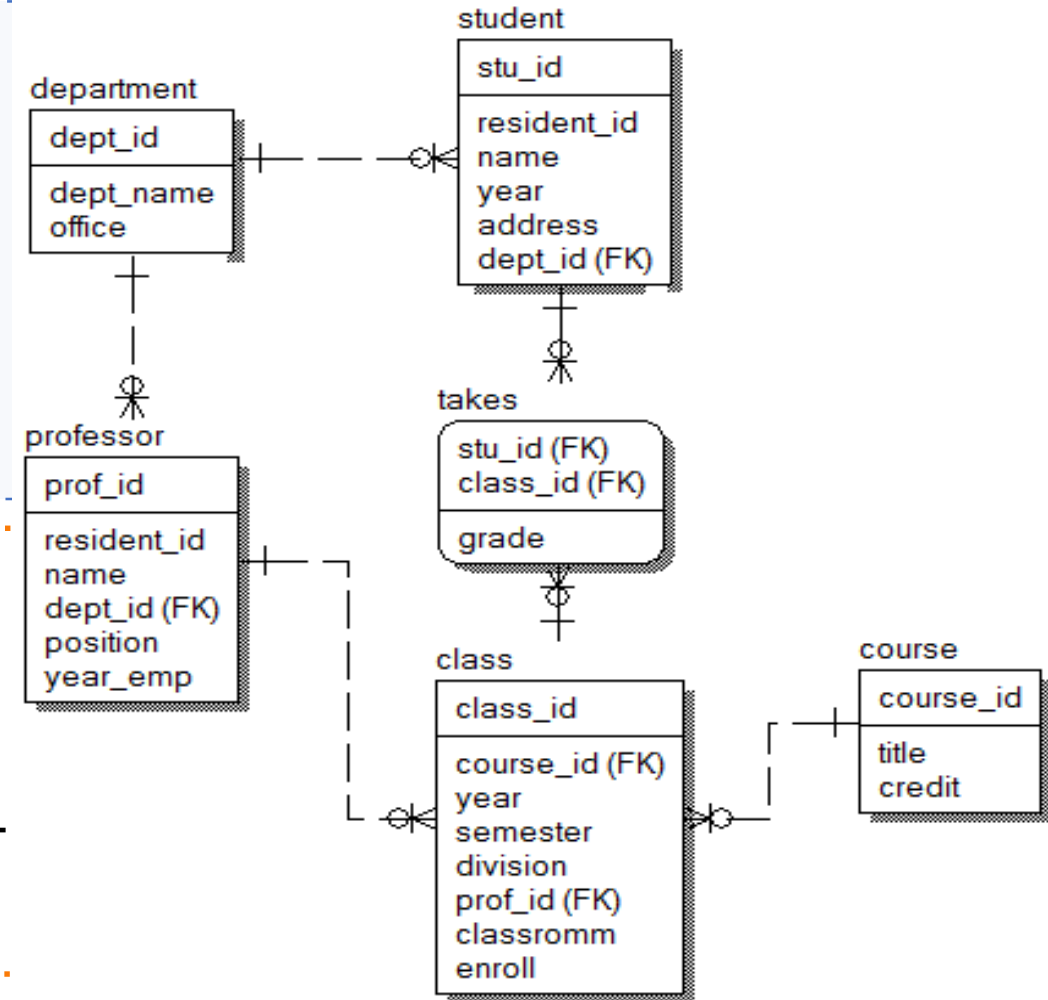
## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classroom, enroll)
-- takes(stu_id, class_id, grade)
```

### -- [예제5] 집계함수, group by

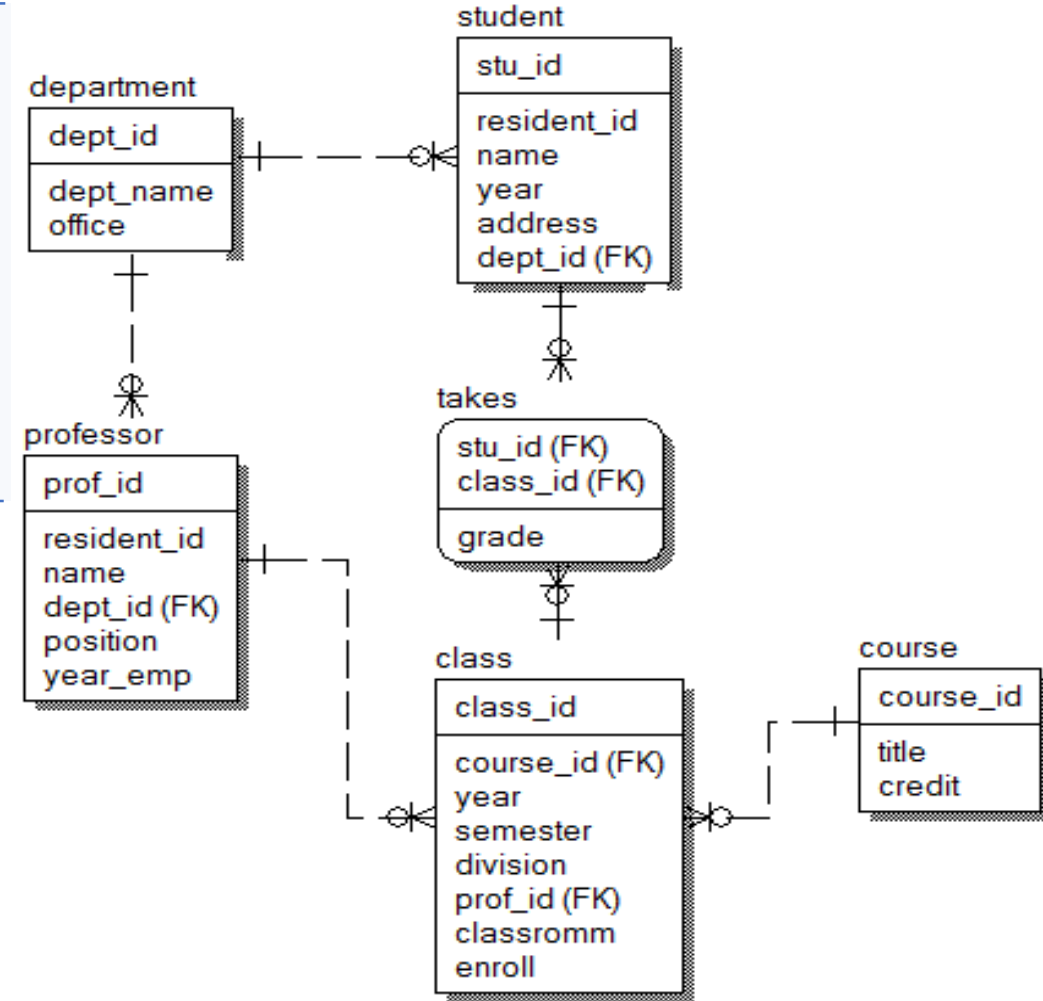
```
-- 질의57) student 테이블에서 학과번호(dept_id) 별로 레코드의 개수를
검색하라
-- 질의58) 학과명 별로 레코드의 개수를 검색하라
-- 질의60) 학과별 교수 숫자와 평균 재직연수, 최대 재직연수를 검색하라
-- 질의62) 평균 재직연수가 10년 이상인 학과에 대해서만
-- 교수 수, 평균재직연수, 최대재직연수를 검색하라(having)
```



## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classromm, enroll)
-- takes(stu_id, class_id, grade)
```



```
-- [예제6] 하위질의
질의66) 301호 강의실에서 개설된 강좌의 과목명을 검색하라(하위질의)
질의67) 301호 강의실에서 개설된 강좌의 과목명을 검색하라(조인)
```

## [실습] 실습 예제(ehan-mysql-stu.sql)

### [실습]

```
-- department(dept_id, dept_name, office)
-- student(stu_id, resident_id, name, year, address, dept_id)
-- professor(prof_id, resident_id, name, dept_id, position, year_emp)
-- course(course_id, title, credit)
-- class(class_id, course_id, year, semester, division, prof_id, classromm, enroll)
-- takes(stu_id, class_id, grade)
```

-- [예제6] 하위질의

질의69) 가장 많은 수강인원을 가진 강좌를 검색하라

