

# 데이터베이스관리

11주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# SQL: 그룹화/ 관계대수

11주차 11-01

담당교수: 김희숙  
(jasmin11@hanmail.net)

## [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

- 12) 2번 부서에 근무하는 직원들의 급여, 직급, 이름을 검색하여 급여의 오름차순으로 정렬하라
- 13) 직원들의 평균 급여와 최대급여를 검색하라
- 14) 직원들이 속한 부서번호별로 그룹화하고 부서번호, 평균급여, 최대급여를 검색하라
- 15) 직원들이 속한 부서번호별로 그룹화하고, 평균급여가 2500000원 이상인
- 부서에 대해서 부서번호, 평균급여, 최대급여를 검색하라

사원

직원코드	성명	직책	연봉
161353	김미나	대리	2300
181323	최영락	사원	1900
151453	홍진호	과장	2800
135485	구준표	과장	3000
104895	강나래	팀장	3600
165484	김하늘	대리	2400

- 1) 직책별 연봉의 평균을 구하시오
- 2) 직책별 사원수를 구하시오
- 3) 연봉이 2500 이상인 사원이 2명 이상인 직책의 사원수를 구하시오

〈사원〉

사원	직급
김형석	대리
홍영선	과장
류기선	부장
김현천	이사

〈직원〉

사원	직급
신원섭	이사
이성호	대리
홍영선	과장
류기선	부장

사원 명단이 사원, 직원 테이블에 저장되어 있다  
두 테이블을 통합하는 SQL문법을 작성하시오  
같은 레코드가 두 번 나오지 않도록 한다

```
CREATE TABLE 성적 (  
    이름 varchar(20) ,  
    국어 int ,  
    영어 int ,  
    수학 int ,  
    과학 int ,  
    PRIMARY KEY(이름)  
);  
  
CREATE TABLE 학적부 (  
    이름 varchar(20) ,  
    주소 varchar(50) ,  
    나이 int ,  
    PRIMARY KEY(이름) ,  
    FOREIGN KEY(이름) REFERENCES 성적(이름)  
);
```

```
insert into 성적 values('고길동',70,90,78,88);  
insert into 성적 values('이순신',90,80,88,98);  
insert into 성적 values('둘리',80,68,98,78);  
insert into 성적 values('아무개',79,79,70,57);  
insert into 성적 values('임꺽정',67,98,80,84);  
  
insert into 학적부 values('고길동','모진동',25);  
insert into 학적부 values('이순신','화양동',20);  
insert into 학적부 values('둘리','자양동',24);  
insert into 학적부 values('아무개','화양동',19);  
insert into 학적부 values('임꺽정','구의동',18);
```

## <성적>

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
아무개	79	79	70	57
임꺽정	67	98	80	84

## <학적부>

이름	주소	나이
고길동	모진동	25
이순신	화양동	20
둘리	자양동	24
아무개	화양동	19
임꺽정	구의동	18

**예제 1**  $\sigma_{\text{국어} \geq 80}(\text{성적})$  : <성적> 릴레이션에서 국어 점수가 80점 이상인 튜플들을 추출한다.

이름	국어	영어	수학	과학
이순신	90	80	88	98
둘리	80	68	98	78

```
SELECT *
FROM 성적
WHERE 국어 >= 80;
```

**예제 2**  $\sigma_{\text{국어} \geq 80 \vee \text{영어} \geq 80}(\text{성적})$  : <성적> 릴레이션에서 국어 점수 또는 영어 점수가 80점 이상인 튜플들을 추출한다.

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
임꺽정	67	98	80	84

```
SELECT *
FROM 성적
WHERE 국어 >= 80 OR 영어 >= 90;
```

<성적>

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
아무개	79	79	70	57
임꺽정	67	98	80	84

<학적부>

이름	주소	나이
고길동	모진동	25
이순신	화양동	20
둘리	자양동	24
아무개	화양동	19
임꺽정	구의동	18

**예제**  $\pi_{\text{이름, 국어}}(\text{성적})$  : <성적> 릴레이션에서 이름과 국어 속성을 추출한다,

이름	국어
고길동	70
이순신	90
둘리	80
아무개	79
임꺽정	67

```
SELECT 이름, 국어
FROM 성적;
```



**예제** 성적×이름=이름학적부 : <성적> 릴레이션과 <학적부> 릴레이션으로부터 이름 속성이 같은 튜플들을 결합하여 새로운 릴레이션을 생성한다.

<성적>

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
아무개	79	79	70	57
임꺽정	67	98	80	84

<학적부>

이름	주소	나이
고길동	모진동	25
이순신	화양동	20
둘리	자양동	24
아무개	화양동	19
임꺽정	구의동	18



성적,이름	국어	영어	수학	과학	학적부,이름	주소	나이
고길동	70	90	78	88	고길동	모진동	25
이순신	90	80	88	98	이순신	화양동	20
둘리	80	68	98	78	둘리	자양동	24
아무개	79	79	70	57	아무개	화양동	19
임꺽정	67	98	80	84	임꺽정	구의동	18

## [실습] 관계대수(Oracle)

### <사원>

사번	이름
1	고길동
2	이순신

### <직원>

사번	이름
2	이순신
3	둘리

**예제1**  $\pi_{\text{이름}}(\text{사원}) \cup \pi_{\text{이름}}(\text{직원}) :$

이름
고길동
이순신
둘리

**예제2**  $\pi_{\text{이름}}(\text{사원}) \cap \pi_{\text{이름}}(\text{직원}) :$

이름
이순신

**예제3**  $\pi_{\text{이름}}(\text{사원}) - \pi_{\text{이름}}(\text{직원}) :$

이름
고길동

```
select * from 사원
UNION
select * from 직원;
```

```
select * from 사원
INTERSECT
select * from 직원;
```

```
select * from 사원
MINUS
select * from 직원;
```

[실습] <http://livesql.oracle.com>

```
CREATE TABLE 사원 (
  사번 int ,
  이름 varchar(20) ,
  PRIMARY KEY(사번)
);
```

```
CREATE TABLE 직원 (
  사번 int ,
  이름 varchar(20) ,
  PRIMARY KEY(사번)
);
```

```
insert into 사원 values(1, '고길동');
insert into 사원 values(2, '이순신');
```

```
insert into 직원 values(2, '이순신');
insert into 직원 values(3, '둘리');
```

**[문제 1]** 관계대수는 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어로서 두 그룹의 연산이 있다. 첫 번째 그룹은 수학적 집합이론으로부터 나온 일반 집합의 연산이다. 릴레이션은 튜플의 집합이기 때문에 집합 연산은 기본적으로 릴레이션에 그대로 적용할 수 있다. 일반 집합 연산에는 합집합, 차집합, 교집합, 카디션 프로덕트가 있다. 두 번째 그룹은 관계 데이터베이스에 적용할 수 있도록 특별히 개발한 순수 관계 연산들이 있다.

순수 관계 연산자들 중 (① )연산은 릴레이션에서 주어진 조건을 만족하는 튜플들을 선택하는 연산이다.

(① )연산의 결과 릴레이션은 주어진 릴레이션을 수평적으로 절단하여 그 일부를 가지고 구성한 것과 같기 때문에 수평연산이라고도 한다. 또한 (① )연산에서는 부울 연산자 AND, OR, NOT 등으로 결합시켜 여러 조건을 부여하여 이용할 수도 있다.

### 답항보기

1	JOIN	2	UNION	3	PROJECT	4	$\pi$ 학년,학과(학생)	5	DIFFERENCE
6	CARTESIAN	7	$\sigma$ 학년=4(학생)	8	SELECT	9	제약(Constraint)	10	DIVISION

## [예제] 관계대수: [문제 1]

### <학생> 릴레이션

학번	이름	학년	학과
1001	나기태	4	컴퓨터
1002	송찬영	3	전기
1003	정연	1	컴퓨터
1004	박병화	4	건축
1005	김미나	1	컴퓨터

<학생> 릴레이션을 대상으로 <예1>과 같은 릴레이션을 생성하는 (① ) 연산의 표기형식은 (② )와 같이 나타낼 수 있다.

### <예1>

학번	이름	학년	학과
1001	나기태	4	컴퓨터
1004	박병화	4	건축

### 답항보기

1	JOIN	2	UNION	3	PROJECT	4	$\pi$ 학년, 학과(학생)	5	DIFFERENCE
6	CARTESIAN	7	$\sigma$ 학년=4(학생)	8	SELECT	9	제약(Constraint)	10	DIVISION

## [예제] 관계대수: [문제 1]

순수 관계 연산들 중 (③ )연산은 릴레이션의 애트리뷰트 들을 대상으로 하는 연산으로, 연산에 명시된 애트리뷰트 값들만 선택한다. (③) )연산의 결과로 만들어진 릴레이션은 주어진 릴레이션의 몇몇 열들로 구성된 것으로 릴레이션의 수직적 부분 집합과 같으므로 수직연산이라고도 한다. (③ )연산의 결과 릴레이션은 중복된 튜플을 배제하고 생성된다. 예를들어, <학생>릴레이션을 대상으로 <예2>와 같은 결과 릴레이션을 생성하는 (③ )연산의 표기 형식은 (④ )(으)로 나타낼 수 있다.

<예2>

학번	학과
4	컴퓨터
3	전기
1	컴퓨터
4	건축

관계대수 연산을 이용하여 <예3>과 같이 <학생>릴레이션에 대해 학과가 “컴퓨터”이고, 학년이 1인 학생의 학번과 이름으로 구성된 릴레이션 표기형식은  $\pi_{\text{학번}, \text{이름}}(\sigma_{\text{학과}=\text{“컴퓨터”} \wedge \text{학년}=1}(\text{학생}))$ 으로 나타낼 수 있다.

<예3>

학번	이름
1003	정연
1005	김미나

답항보기

1	JOIN	2	UNION	3	PROJECT	4	$\pi_{\text{학번}, \text{학과}}(\text{학생})$	5	DIFFERENCE
6	CARTESIAN	7	$\sigma_{\text{학번}=4}(\text{학생})$	8	SELECT	9	제약(Constraint)	10	DIVISION



또한, 순수 관계 연산에는 공통 속성을 중심으로 2개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 (⑤ )연산 등이 있고, (⑤ )연산의 연산자 기호는 “ $\bowtie$ ”를 사용한다.

### 답항보기

1	JOIN	2	UNION	3	PROJECT	4	$\pi$ 학년,학과(학생)	5	DIFFERENCE
6	CARTESIAN	7	$\sigma$ 학년=4(학생)	8	SELECT	9	제약(Constraint)	10	DIVISION

관계 대수는 질의에 대한 결과를 생성하기 위해 연산의 순서를 어떻게 해야 하는지 명시하는 절차 언어이다. 관계 대수의 연산은 하나의 릴레이션이나 여러 개의 릴레이션으로 부터 일정한 조건에 맞는 튜플들을 찾아 새로운 릴레이션을 만들거나 여러 개의 릴레이션들을 통합하여 새로운 하나의 릴레이션을 만드는 연산을 수행하기 때문에 연산의 피연산자가 모두 릴레이션이고 그 결과도 릴레이션이다.

관계 대수의 연산 중 조인 연산에 대하여 살펴 보자.

조인(Join)연산은 두 개의 릴레이션으로 부터 연관된 튜플(Tuple)들을 결합하는 연산이다. 결과만 비교 한다면 조인 연산은 (① ) 연산을 수행한 후, (② )연산을 적용한 것과 같다. (② )연산은 조건을 만족하는 릴레이션의 수평적 부분 집합으로 구성되며, 연산자의 기호는 그리스 문자 시그마를 사용한다.

조인 연산은 조인을 효율적으로 시행하는 여러 가지 알고리즘이 개발되었다.

조인을 비교 연산자 'θ'로 일반화하여, 'θ'로 표현될 수 있는 조인을 (③ )조인이라 한다. 두 릴레이션  $R(A_1, A_2, \dots, A_n)$ 과  $S(B_1, B_2, \dots, B_m)$ 의 (③ )조인의 결과는 차수(Degree)가  $(n+m)$ 이고, 애트리뷰트가  $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ 이며, 조인 조건을 만족하는 튜플들로 이루어진 집합이다.

### 답항보기

1	JOIN	2	UNION	3	PROJECT	4	⋈	5	DIVISION
6	Cartesian Product	7	선택(Select)	8	세타(Theta)	9	관계연산자	10	자연(Natural)

(③ )는(은) 비교 연산자 중의 하나이다. (③ )조인을 수행하기 위해서 양쪽 릴레이션의 애트리뷰트(일반적으로 기본 키와 외래키 관계)들이 (③ )조인 조건을 만족하는 튜플들만 골라낸다.

이것이 (① )연산과 조인 연산의 주요 차이점이다. (① )연산은 릴레이션 R에 존재하는 튜플과 릴레이션 S에 존재하는 튜플들을 모두 접속시킨 새로운 결과 릴레이션을 만드는 연산이다. 따라서, (① )연산의 결과로 만들어지는 릴레이션의 차수는 피연산자 R과 S릴레이션의 차수의 합과 같고, 카디널리티는 R의 카디널리티와 S 카디널리티의 곱과 같다.

동등조인은 (③ )조인 중에서 비교연산자가 '='인 조인이다.

동등조인의 형식은 다음과 같이 표현한다.

$R(\text{④})_{R.attribute=S.attribute}S$

조인의 기호로는 (④ )를(을) 사용한다.

### 답항보기

1	JOIN	2	UNION	3	PROJECT	4	⋈	5	DIVISION
6	Cartesian Product	7	선택(Select)	8	세타(Theta)	9	관계연산자	10	자연(Natural)



## [예제] 관계대수: [문제 2]

다음과 같은 두 릴레이션이 있다고 가정하자.

답항보기

1	JOIN	2	UNION	3	PROJECT	4	⋈	5	DIVISION
6	Cartesian Product	7	선택(Select)	8	세타(Theta)	9	관계연산자	10	자연(Natural)

EMPLOYEE

EMPNO	EMPNAME	DNO
1234	김원호	2
1235	박영주	1
1236	이수미	3
1237	조민수	2
1238	최종인	3

DEPARTMENT

DEPTNO	DEPTNAME
1	영업
2	기획
3	개발
4	총무

'EMPLOYEE(④ ) DNO=DEPTNO DEPARTMENT' 로 동등 조인한 결과는 다음과 같다.

EMPNO	EMPNAME	DNO	DEPTNO	DEPTNAME
1234	김원호	2	2	영업
1235	박영주	1	1	기획
1236	이수미	3	3	개발
1237	조민수	2	2	총무
1238	최종인	3	3	개발

동등 조인에서는 두 릴레이션에서 조인 조건에 사용된 두 애트리뷰트가 결과 릴레이션에 포함된다. 그러나, 이 두 애트리뷰트는 이름이 다를 수 있어도 결과 릴레이션의 각 튜플에서 두 애트리뷰트 값이 같으므로 둘 중 하나의 애트리뷰트만 포함시켜도 무방하다. 즉 속성의 중복은 결과에 아무런 의미를 주지 못하기 때문에 제거되어 연산을 수행하는 것이 바람직하다. 동등조인의 결과 릴레이션에서 중복된 속성을 제거하여 수행하는 연산을 (⑤ )조인이라고 한다.

즉 동등조인에서 중복 속성 중 하나가 제거된 것이 (⑤ )조인이다.

(⑤ )조인의 핵심은 두 릴레이션의 공통된 속성을 매개체로 하여 두 릴레이션의 정보를 '관계'로 묶어 내는 것이다. 일반 테이블에서는 보통 외래키가 매개체 역할을 담당한다.

```
CREATE TABLE department (
  deptno int ,
  deptname varchar(20) ,
  PRIMARY KEY(deptno)
);

CREATE TABLE employee (
  empno char(4) ,
  empname varchar(20) ,
  dno int
  PRIMARY KEY(empno) ,
  FOREIGN KEY(dno) REFERENCES department(deptno)
);
```

EMPLOYEE

EMPNO	EMPNAME	DNO
1234	김원호	2
1235	박영주	1
1236	이수미	3
1237	조민수	2
1238	최종인	3

```
insert into department values(1,'영업');
insert into department values(2,'기획');
insert into department values(3,'개발');
insert into department values(4,'총무');

insert into employee values('1234','김원호',2);
insert into employee values('1235','박영주',1);
insert into employee values('1236','이수미',3);
insert into employee values('1237','조민수',2);
insert into employee values('1238','최종인',3);
```

DEPARTMENT

DEPTNO	DEPTNAME
1	영업
2	기획
3	개발
4	총무

'EMPLOYEE(④ ) DNO=DEPTNO DEPARTMENT' 로 동등 조인한 결과는 다음과 같다.

EMPNO	EMPNAME	DNO	DEPTNO	DEPTNAME
1234	김원호	2	2	영업
1235	박영주	1	1	기획
1236	이수미	3	3	개발
1237	조민수	2	2	총무
1238	최종인	3	3	개발

# SQL: 조인(join)

11주차 11-02

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 조인(join)



## □조인

- ✓ 2개 이상의 테이블로 부터 원하는 결과를 나타내는 것
- ✓ 2개 이상의 테이블로부터 필요한 열을 선택하여 하나의 테이블로 논리적인 결합하는 것

개인신상

이름	주민번호	생년월일	관계	핸드폰번호	우편번호	주소
박찬호	730201-1034343	1973-02-01	후배	(017) 234-2342	(333-333)	서울시 강남구 역삼동 234-2
선동열	680709-1078656	1968-07-09	동생	(018) 234-2342	(444-444)	서울시 서초구 서초동 234-2
임꺽정	700101-1027362	1970-01-01	고객	(016)2342-2342	(222-222)	서울시 영등포구 여의도동 234-2
차범근	600809-1987766	1960-08-09	친구	(019) 234-2342	(555-555)	서울시 종로구 관철동 234-2
홍길동	651214-1078767	1965-12-14	친구	(011) 234-2342	(111-111)	서울시 구로구 구로본동 234-24

자동차

이름	차종	차명	년식	구입일
임꺽정	트럭	포토	95	97-5
차범근	승용차	에쿠스	01	01-8
홍길동	승용차	소나타	98	98-8



이름	생년월일	차종	차명
임꺽정	1970-01-01	트럭	포토
차범근	1960-08-09	승용차	에쿠스
홍길동	1965-12-14	승용차	소나타

## (테이블 1개) 고객관리

## ❖ 테이블 1개인 경우

	이름	주민번호	생년월일	관계	핸드폰번호	우편번호	주소	차종	차명	년식	구입일
▶	박찬호	720201-1034343	1973-02-01	후배	(017) 234-2342	(333-333)	서울시 강남구 역삼동 234-2	NULL	NULL	NULL	NULL
	선동열	680709-1078656	1968-07-09	동생	(018) 234-2342	(444-444)	서울시 서초구 서초동 234-2	NULL	NULL	NULL	NULL
	임꺽정	700101-1027362	1970-01-01	고객	(016)2342-2342	(222-222)	서울시 영등포구 여의도동 234-2	트럭	포토	1995	1997-05
	차범근	600809-1987766	1960-08-09	친구	(017) 234-2342	(555-555)	서울시 종로구 관철동 234-2	승용차	메쿠스	2001	2001-08
	홍길동	651214-1078767	1965-12-14	친구	(011) 234-2342	(111-111)	서울시 구로구 구로본동 234-24	승용차	소나타	1998	1998-08

## (테이블 2개) 개인신상, 자동차

## ❖ 테이블 2개인 경우

	이름	주민번호	생년월일	관계	핸드폰번호	우편번호	주소
▶	박찬호	720201-1034343	1973-02-01	후배	(017) 234-2342	(333-333)	서울시 강남구 역삼동 234-2
	선동열	680709-1078656	1968-07-09	동생	(018) 234-2342	(444-444)	서울시 서초구 서초동 234-2
	임꺽정	700101-1027362	1970-01-01	고객	(016)2342-2342	(222-222)	서울시 영등포구 여의도동 234-2
	차범근	600809-1987766	1960-08-09	친구	(017) 234-2342	(555-555)	서울시 종로구 관철동 234-2
	홍길동	651214-1078767	1965-12-14	친구	(011) 234-2342	(111-111)	서울시 구로구 구로본동 234-24

	이름	생년월일	차종	차명
▶	임꺽정	1970-01-01	트럭	포토
	차범근	1960-08-09	승용차	메쿠스
	홍길동	1965-12-14	승용차	소나타



# [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

# [실습] 조인(테이블 2개)

[내부조인] (테이블 2개) freshman, dmember

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

# [실습] 조인(테이블 2개)

[외부조인] (테이블 2개) freshman, dmember

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과



# [실습] 외부조인(테이블 2개)

[외부조인] (테이블 2개) emp, project

emp

ename	pno
홍길동	101
임꺽정	102
박찬호	101
박찬호	103
신동엽	NULL

project

pno	pname
101	작전중
102	특공대
103	유레카
104	다모여

## [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

--2-1) 수강한 학생의 학생명, 과목명, 성적을 검색하라(내부조인)

--2-2) 이홍근의 학생명, 과목명, 성적을 검색하라

--2-3) 전체 학생의 학생명, 과목명, 성적을 검색하라(외부조인)