

# 제 4 장

## 조인(JOIN)과 부질의(SubQuery)



# 목차

4.1 조인(Join)

4.2 부질의(SubQuery)

4.3 집합(SET)연산자

## 4.1 조인(Join)

### ● 조인(Join)

- 상호 특정 관련성을 갖는 두 개 이상의 테이블로부터 새로운 테이블을 생성하는데 사용되는 연산
- 두 개 이상의 테이블을 “조인(join)” 할 수 있는 기능은 관계시스템이 갖는 가장 강력한 특징.
- 정규화를 거친 데이터베이스의 각 테이블은 분리되어 관련된 다른 테이블과의 결합이 필요하며 이런 경우 관련 열의 관계성을 유도하여 결합됨

### ● 오라클에서 조인의 종류

- Cross 조인
- Equi 조인
- Non-Equi 조인
- Self 조인
- Outer 조인

## 4.1 조인(Join)

### ●CROSS 조인

- 크로스 조인(cross join)은 관계시스템의 관계 대수 8가지 연산 중 카티션 프로덕트(Cartesian Product)를 구현함
- 2개 이상의 테이블을 조건 없이 실행되는 조인 연산

SQL> select \*

2 from student cross join enrol;

Student X enroll  
10      12 → 120

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT	SUB_NO	STU_NO_1	ENR_GRADE
20153075	육한빛	기계	1	C	M	177	80	101	20131001	80
20153075	육한빛	기계	1	C	M	177	80	104	20131001	56
20153075	육한빛	기계	1	C	M	177	80	106	20132003	72
20153075	육한빛	기계	1	C	M	177	80	103	20152088	45
20153075	육한빛	기계	1	C	M	177	80	101	20131025	65
20153075	육한빛	기계	1	C	M	177	80	104	20131025	65
20153075	육한빛	기계	1	C	M	177	80	108	20151062	81
20153075	육한빛	기계	1	C	M	177	80	107	20143054	41
20153075	육한빛	기계	1	C	M	177	80	102	20153075	66
20153075	육한빛	기계	1	C	M	177	80	105	20153075	56
20153075	육한빛	기계	1	C	M	177	80	102	20153088	61
20153075	육한빛	기계	1	C	M	177	80	105	20153088	78
20153088	이태연	기계	1	C	F	162	50	101	20131001	80
20153088	이태연	기계	1	C	F	162	50	104	20131001	56
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....

## 4.1 조인(Join)

●EQUI 조인    공통 열 때문에 중복이 발생한다. 중복을 최소화 시킨다.(공통 열이 존재하면 중복이 존재할 수밖에 없음.)

➢ 2개 이상의 테이블에 **관련 있는 공통 열의 값을 이용하여 논리적으로 결합하는 연산**이 수행되는 조인

관계형(RDB) -> join 기능이 있기에 사용한다.

```
SQL> select student.stu_no, stu_name, stu_dept, enr_grade  
2   from student, enrol  
3   where student.stu_no = enrol.stu_no;
```

관계형 데이터베이스에서는 조인을 사용하기 위해 공통열을 만들어 설계한다.

실무에서 제일 많이 사용되는 조인 방식  
= 로 공통 열을 연결한다.(전통적 방식)

STU_NO	STU_NAME	STU_DEPT	ENR_GRADE
20131001	김종헌	컴퓨터정보	80
20131001	김종헌	컴퓨터정보	56
20132003	박희철	전기전자	72
20152088	조민우	전기전자	45
20131025	육성우	컴퓨터정보	65
20131025	육성우	컴퓨터정보	65
20151062	김인중	컴퓨터정보	81
20143054	유가인	기계	41
.....	.....	.....	.....

## 4.1 조인(Join)

Equi 조인의 또 다른 형식이다.

### ● NATURAL JOIN 같은 열은 한 개만 존재해야 한다.

```
SQL> select stu_no, stu_name, stu_dept, enr_grade  
2 from student natural join enrol;
```

➤ 두 테이블에 같은 열의 이름이 2쌍 이상 존재하면 사용 못함.

natural join - 이름의 칼럼을 자동으로 기준 사용 하는 특징이 있다.  
그렇기에 해당 칼럼들을 자동으로 기준 삼아 모든 동일한 값을 가진  
행들이 결합되기 때문에 아 과정에서 예상치 못한 칼럼이 연결되거나,  
값이 중복되는 상황이 발생할 수 있다. (모든 가능한 조합 생성)

### ● JOIN ~ USING 열이 같은 것이 여러개일 땐 using, on 을 통해 조인 할 열을 명시해줘야 한다.

```
SQL> select stu_no, stu_name, stu_dept, enr_grade  
2 from student join enrol using(stu_no);
```

칼럼 이름이 같을 때 사용 가능

### ● JOIN ~ ON SQL-92

```
SQL> select student.stu_no, stu_name, stu_dept, enr_grade  
2 from student join enrol on student.stu_no = enrol.stu_no ;
```

칼럼 이름이 다를 때 사용 가능,  
더 복잡한 조건을 정의할 수 있음

가독성이 높은 것으로 사용하는 것이 좋음,  
join ~ on 형식을 선호하는 경우가 많음

## 4.1 조인(Join)

- 101번 과목을 수강하는 학생들의 학번과 이름 검색

```
SQL> select student.stu_no, stu_name  
2   from student, enrol  
3   where student.stu_no = enrol.stu_no and  
4         sub_no = 101;
```

STU_NO	STU_NAME
20131001	김종현
20131025	육성우



sub\_no = 101 **or** sub\_no = 102

## 4.1 조인(Join)

### ●과목번호 101 또는 102를 수강하는 학생의 학번과 이름 검색

```
SQL> select student.stu_no, stu_name  
2   from student, enrol  
3   where student.stu_no = enrol.stu_no  
4     and sub_no = 101 or sub_no = 102;
```

STU_NO	STU_NAME
20153075	옥한빛
20153088	이태연
20143054	유가인
20152088	조민우
20142021	심수정
20132003	박희철
.....	.....
20131025	옥성우
20131001	김종현
20131025	옥성우

Equi조인의 또다른 3가지 형식은 where에서 조인 하지 않고  
from절에서 끝낸다.

그렇기에 where 절에서 원하는 것들만  
표현 가능.

and, or 혼합 연산은 되도록 피해야 함

그렇기에 where 절에서 자유로운 세가지 기법의 조인을 사용하는 것이 좋음

and 조건이 우선순위가기 때문에 조인 해주고  
연결 할 때 조심해야 한다. 혼합 할 일이 생긴다면  
우선적으로 연산해야 하는 부분에 괄호 씌워야 한다.



## 4.1 조인(Join)

### ●논리 연산자의 우선순위

```
SQL> select student.stu_no, stu_name
2   from student, enrol
3   where student.stu_no = enrol.stu_no
4     and ( sub_no = 101 or sub_no = 102 );
```

STU_NO	STU_NAME
20153075	육한빛
20153088	이태연
20131001	김종현
20131025	육성우

in, between 등 비연속족, 연속적  
값을 처리하는 함수를 사용하는 것이 가독성에 좋음

```
SQL> select student.stu_no, stu_name
2   from student, enrol
3   where student.stu_no = enrol.stu_no
4     and sub_no in (101, 102);
```

## 4.1 조인(Join)

```
SQL> select stu_no, stu_name  
2   from student natural join enrol  
3   where sub_no = 101 or sub_no = 102;
```

3가지 형식을 사용하여  
where 절을 자유롭게 사용한다.

```
SQL> select stu_no, stu_name  
2   from student join enrol using(stu_no)  
3   where sub_no = 101 or sub_no = 102;
```

```
SQL> select student.stu_no, stu_name  
2   from student join enrol on student.stu_no = enrol.stu_no  
3   where sub_no = 101 or sub_no = 102;
```

## 4.1 조인(Join)

### ●3 테이블 조인

➤ ‘컴퓨터개론’ 과목을 수강하는 학생들의 학번, 이름, 과목이름 검색

```
SQL> select student.stu_no, stu_name, sub_name  
2   from student, enrol, subject  
3   where student.stu_no = enrol.stu_no  
4         and enrol.sub_no = subject.sub_no  
5         and enrol.sub_name = ‘컴퓨터개론’;
```

STU_NO	STU_NAME	SUB_NAME
20131001	김종현	컴퓨터개론
20131025	육성우	컴퓨터개론

## 4.1 조인(Join)

### ●Non-Equi 조인

➤ WHERE 절에서 사용하는 '='이 아닌 연산자를 사용

```
SQL> select empno, ename, sal, grade
2   from emp, salgrade
3  where sal between losal and hisal;
```

EMPNO	ENAME	SAL	GRADE
7900	JAMES	950	1
7369	SMITH	800	1
7876	ADAMS	1100	1
7521	WARD	1250	2
7654	MARTIN	1250	2
7934	MILLER	1300	2
7499	ALLEN	1600	3
7844	TURNER	1500	3
.....	.....	.....	.....

$sal \geq losal$  and  $sal \leq hisal$

# 4.1 조인(Join)

EQUI 조인 영역

## ●SELF 조인

➤ 같은 테이블 간의 조인, 테이블의 별칭을 사용함

➤ 다음은 자신의 상급자를 구하는 질의문

```
SQL> select a.empno as 사원번호, a.ename as 사원이름,  
2         b.empno as 상급자사원번호, b.ename as 상급자이름  
3 from emp a, emp b  
4 where a.mgr = b.empno;
```

사원번호	사원이름	상급자사원번호	상급자이름
7369	SMITH	7902	FORD
7499	ALLEN	7698	BLAKE
7521	WARD	7698	BLAKE
7566	JONES	7839	KING
.....	.....	.....	.....

➔ JOIN ~ ON

```
SQL> select a.empno as 사원번호, a.ename as 사원이름,  
2         b.empno as 상급자사원번호, b.ename as 상급자이름  
3 from emp a, join emp b on a.mgr = b.empno;
```

## 4.1 조인(Join)

### ●Outer 조인

➤ OUTER JOIN은 **조인 조건을 만족하지 않는 행들도 질의 결과에 포함함**

➤ 수강(enrol) 테이블을 기준으로 과목이름 검색

```
SQL> select a.*, sub_name
2   from enrol a, subject b
3   where a.sub_no = b.sub_no
4   order by 1;
```

**\*과목번호 109 ~ 111은 수강생이 없어  
조인에서 제외됨**

SUB_NO	SUB_NAME
109	자동화설계
110	자동제어
111	데이터베이스

SUB_NO	STU_NO	ENR_GRADE	SUB_NAME
101	20131001	80	컴퓨터개론
101	20131025	65	컴퓨터개론
102	20153075	66	기계공학법
102	20153088	61	기계공학법
103	20152088	45	기초전자실험
104	20131001	56	시스템분석설계
104	20131025	65	시스템분석설계
105	20153088	78	기계요소설계
105	20153075	56	기계요소설계
106	20132003	72	전자회로실험
107	20143054	41	CAD응용실습
108	20151062	81	소프트웨어공학

## 4.1 조인(Join)

```
SQL> select a.*, b.sub_no, sub_name
2   from enrol a right outer join subject b
3     on a.sub_no = b.sub_no
4   order by 1;
```

SUB_NO	STU_NO	ENR_GRADE	SUB_NO	SUB_NAME
101	20131001	80	101	컴퓨터개론
101	20131025	65	101	컴퓨터개론
102	20153088	61	102	기계공학법
102	20153075	66	102	기계공학법
103	20152088	45	103	기초전자실험
104	20131025	65	104	시스템분석설계
104	20131001	56	104	시스템분석설계
105	20153075	56	105	기계요소설계
105	20153088	78	105	기계요소설계
106	20132003	72	106	전자회로실험
107	20143054	41	107	CAD응용실습
108	20151062	81	108	소프트웨어공학
			109	데이터베이스
			110	자동화설계
			111	자동제어

- ✓RIGHT OUTER JOIN
- ✓LEFT OUTER JOIN
- ✓FULL OUTER JOIN

## 4.1 조인(Join)

### ●SELF 조인과 OUTER 조인의 결합

SQL>select a.empno as **사원번호**, a.ename as **사원이름**,  
2           b.empno as **상급자사원번호**, b.ename as **상급자이름**  
3   from emp a left outer join emp b on a.mgr = b.empno;

사원번호	사원이름	상급자사원번호	상급자이름
7782	CLARK	7839	KING
7698	BLAKE	7839	KING
7566	JONES	7839	KING
.....	.....	.....	.....
7499	ALLEN	7698	BLAKE
7934	MILLER	7782	CLARK
7876	ADAMS	7788	SCOTT
7369	SMITH	7902	FORD
7839	KING		



## 4.2 부질의(SubQuery)

### ●부질의(SubQuery)

- SELECT문내에 또 다른 SELECT문을 포함함(부질의)
- ‘옥성우’ 학생보다 신장이 큰 학생들의 학번, 이름 신장 검색

SQL>select stu\_height

2 from student

3 where stu\_name = ‘옥성우’;

STU_HEIGHT
172

SQL>select stu\_height

2 from student

3 where stu\_height > 172;

STU_HEIGHT
177
188
174

## 4.2 부질의(SubQuery)

부질의 <-> 조인 변환가능

### ●하나의 질의문으로 처리함

```
SQL> select stu_no, stu_name, stu_height
2   from student
3   where stu_height >
4         (select stu_height
5          from student
6          where stu_name = '옥성우');
```

STU_NO	STU_NAME	STU_HEIGHT
20153075	옥한빛	177
20152088	조민우	188
20141007	진현무	174

### ●WHERE절, HAVING절, FROM절

## 4.2 부질의(SubQuery)

### ●SELF 조인 이용하여 재 작성함

```
SQL> select a.stu_no, a.stu_name, a.stu_height  
2   from student a, student b  
3   where a.stu_height > b.stu_height  
4     and b.stu_name = '옥성우';
```

STU_NO	STU_NAME	STU_HEIGHT
20153075	옥한빛	177
20152088	조민우	188
20141007	진현무	174

## 4.2 부질의(SubQuery)

### ● 단일열 부질의

➤ 열의 값 한 개를 반환하는 부질의

➤ 박희철 학생과 같은 체중을 가지고 있는 학생의 정보 검색

SQL> select \*

2 from student

3 where stu\_weight =

4 (select stu\_weight

5 from student

6 where stu\_name = '박희철');



STU_WEIGHT
63

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20132003	박희철	전기전자	3	B	M		63
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.2 부질의(SubQuery)

```
SQL> select *  
2   from student  
3   where stu_weight =  
4       (select stu_weight  
5         from student  
6         where stu_name = '박희철')  
7   and stu_name <> '박희철';
```

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.2 부질의(SubQuery)

비교연산자 사용 불가

- 부질의 결과가 다중일 경우(IN 연산자 사용)
- '컴퓨터정보' 과 학생과 같은 반을 가진 학생의 정보 검색

in 불연속적인 값  
between d연속적

SQL> select \*

2 from student

3 where stu\_class in

4 (select distinct stu\_class

5 from student

6 where stu\_dept = '컴퓨터정보')

7 and stu\_dept <> '컴퓨터정보';

STU_CLASS
B
A
C

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20132003	박희철	전기전자	3	B	M		63
20142021	심수정	전기전자	2	A	F	168	45
20152088	조민우	전기전자	1	C	M	188	90
20143054	유가인	기계	2	C	F	154	47
20153088	이태연	기계	1	C	F	162	50
20153075	옥한빛	기계	1	C	M	177	80

## 4.2 부질의(SubQuery)

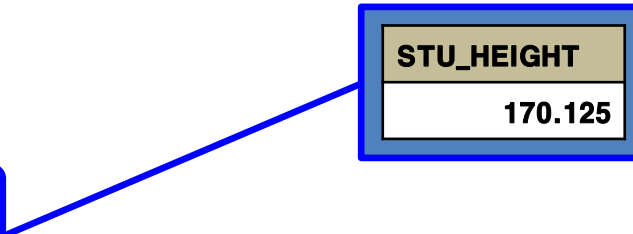
➤ 전체 학생들의 평균신장 보다 큰 학생의 정보 검색

SQL> select \*

2 from student

3 where stu\_height >

4 (select avg(stu\_height)  
5 from student);



STU_HEIGHT
170.125

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20152088	조민우	전기전자	1	C	M	188	90
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.2 부질의(SubQuery)

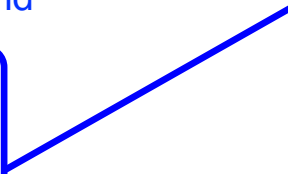
➤ 신장이 모든 학과들의 평균 신장보다 큰 학생의 정보 검색

SQL> select \*

2 from student

3 where stu\_height > all and

4 (select avg(stu\_height)  
5 from student  
6 group by stu\_dept);



STU_HEIGHT
178
164.3333333
170.6666667

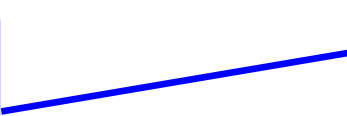
STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20152088	조민우	전기전자	1	C	M	188	90



## 4.2 부질의(SubQuery)

➤ 컴퓨터정보과의 최소 신장보다 최소 신장이 더 큰 학과의 학과명과 최소 신장 검색

```
SQL> select stu_dept, min(stu_height)
2  from student
3  group by stu_dept having min(stu_height) >
4  (select min(stu_height)
5   from student
6   where stu_dept = '컴퓨터정보');
```



MIN(STU_HEIGHT)
166

STU_DEPT	MIN(STU_HEIGHT)
전기전자	168

## 4.2 부질의(SubQuery)

- 일반적으로 **부질의**의 질의문을 **조인** 질의로 표현 가능

- 101번 과목을 수강한 학생들의 정보 검색

SQL> select \*

그룹 함수가 들어가면 대부분 부질의로 푼다 ( 대전제 )

2 from student

3 where stu\_no in

4 (select stu\_no

5 from enrol

6 where sub\_no = 101);

||

SQL> select \*

2 from student a, enrol b

3 where a.stu\_no = b.stu\_no and b.sub\_no = 101;

## 4.2 부질의(SubQuery)

- 101번 과목을 수강한 학생들의 학번, 이름, 점수를 검색

```
SQL> select a.stu_no, a.stu_name, b.enr_grade  
2   from student a, enrol b  
3   where a.stu_no = b.stu_no and b.sub_no = 101;
```

STU_NO	STU_NAME	ENR_GRADE
20131001	김종현	80
20131025	옥성우	65

- 부질의로 표현 불가

## 4.2 부질의(SubQuery)

- 복수열 부질의

- 복수열 값을 반환하는 부질의

- 실습을 위한 사원(emp) 테이블과 구조가 같은 test테이블 생성

```
SQL> create table test(empno, ename, sal, comm, deptno)
```

```
2 as
```

```
3 select empno, ename, sal, comm, deptno
```

```
4 from emp
```

```
5 where deptno = 1;
```

```
SQL> insert into test values(11, 'apple', 1000, null, 30);
```

```
SQL> insert into test values(12, 'banana', 2000, 100, 30);
```

```
SQL> insert into test values(13, 'cheese', 1000, 0, 10);
```

```
SQL> insert into test values(14, 'dog', 2000, null, 20);
```

```
SQL> insert into test values(15, 'egg', 1000, 100, 20);
```

## 4.2 부질의(SubQuery)

### ●test 테이블 데이터 검색

SQL> select \*

2 from test;

EMPNO	ENAME	SAL	COMM	DEPTNO
11	apple	1000		30
12	banana	2000	100	30
13	cheese	1000	0	10
14	dog	2000		20
15	egg	1000	100	20

## 4.2 부질의(SubQuery)

➤ 부질의 결과값이 복수열      컬럼 여러개면 비교 대상도 같아야 한다.

SQL> select \*

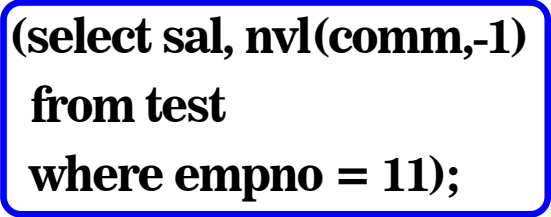
2    from test

3    where (sal, nvl(comm, -1)) =

4        (select sal, nvl(comm, -1)

5        from test

6        where empno = 11);



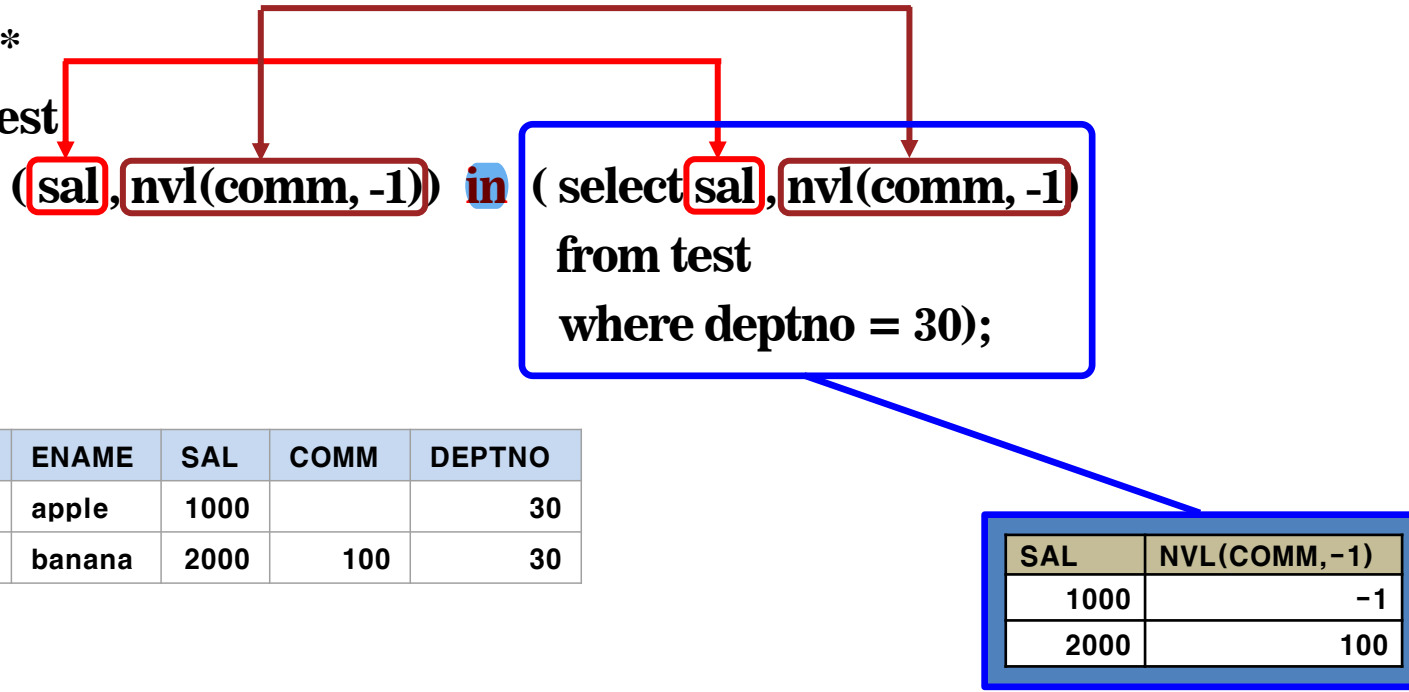
SAL	NVL(COMM, -1)
1000	-1

EMPNO	ENAME	SAL	COMM	DEPTNO
11	apple	1000		30

## 4.2 부질의(SubQuery)

➤ 부질의 결과값이 복수열, 복수행

```
SQL> select *  
2  from test  
3  where (sal, nvl(comm, -1)) in (select sal, nvl(comm, -1)  
4  
5  from test  
   where deptno = 30);
```



EMPNO	ENAME	SAL	COMM	DEPTNO
11	apple	1000		30
12	banana	2000	100	30

SAL	NVL(COMM, -1)
1000	-1
2000	100

## 4.2 부질의(SubQuery)

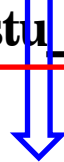
### ●FROM 절의 부질의(In-Line 뷰)

하나의 쿼리에서 해결하기 까다로울 때 사용

➤ 학생들의 학과별 평균 신장보다 큰 신장의 학생들 정보 검색

```
SQL> select stu_dept, round(avg(stu_height),2) as avg_height  
2   from student  
3   group by stu_dept;
```

```
SQL> select stu_no, stu_name, a.stu_dept, stu_height, avg_height  
2   from student a,  
3  
4  
5   where a.stu_dept = b.stu_dept and stu_height > avg_height;
```

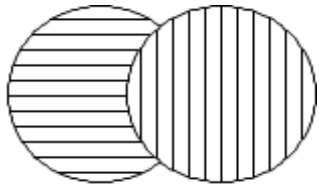


중간 시험범위 - 부질의까지  
20문제 객관식  
3지선다 - 복수정답  
1문제당 2점  
sql 문제 6개

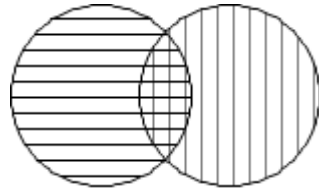


## 4.3 집합(SET)

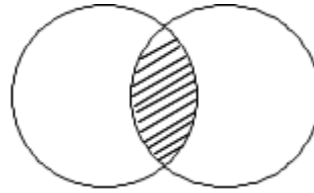
연산자	설 명
UNION	두 질의 결과값의 합으로 중복을 제거됨
UNION ALL	두 질의 결과값의 합으로 중복을 포함됨
INTERSECT	두 질의 결과값의 공통되는 값
MINUS	첫 번째 질의 결과에서 두 번째 질의 결과에 있는 행을 제거한 값



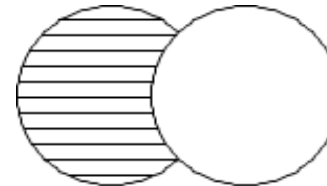
UNION



UNION ALL



INTERSECT



MINUS

## 4.3 집합(SET)

- 집합 연산자 사용 규칙은 다음과 같다.

- 두 개 이상의 SELECT문장의 **열의 수와 데이터 타입 일치**
- 열의 이름 일치하지 않을 경우 첫 번째 SELECT문의 열의 이름
- 정렬을 위해서는 **마지막 SELECT문에 ORDER BY절 표현**
- 집합 연산자 부질문에 사용가능

## 4.3 집합(SET)

- 실습을 위해 a\_student, b\_student 테이블 생성

```
SQL> create table a_student  
2   as select *  
3   from student  
4   where stu_dept in ('기계', '전기전자');
```

```
SQL> create table b_student  
2   as select *  
3   from student  
4   where stu_dept in ('전기전자', '컴퓨터정보');
```

## 4.3 집합(SET)

➤ 생성된 a\_student, b\_student의 내용 확인

**SQL> select \* from a\_student;**

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20153088	이태연	기계	1	C	F	162	50
20143054	유가인	기계	2	C	F	154	47
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63

**SQL> select \* from b\_student;**

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63
20151062	김인중	컴퓨터정보	1	B	M	166	67
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131001	김종현	컴퓨터정보	3	C	M		72
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.3 집합(SET)

### ●UNION

SQL> select \* from a\_student

2    **union**

3    select \* from b\_student;

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20153088	이태연	기계	1	C	F	162	50
20143054	유가인	기계	2	C	F	154	47
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63
20151062	김인중	컴퓨터정보	1	B	M	166	67
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131001	김종현	컴퓨터정보	3	C	M		72
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.3 집합(SET)

### ●UNION

SQL> select \* from a\_student

2 union all

3 select \* from b\_student;

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20153088	이태연	기계	1	C	F	162	50
20143054	유가인	기계	2	C	F	154	47
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63
20151062	김인중	컴퓨터정보	1	B	M	166	67
20141007	진현무	컴퓨터정보	2	A	M	174	64
20131001	김종헌	컴퓨터정보	3	C	M		72
20131025	옥성우	컴퓨터정보	3	A	F	172	63

## 4.3 집합(SET)

### ●INTERSECT

SQL> select \* from a\_student

2 intersect

3 select \* from b\_student;

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20152088	조민우	전기전자	1	C	M	188	90
20142021	심수정	전기전자	2	A	F	168	45
20132003	박희철	전기전자	3	B	M		63

## 4.3 집합(SET)

### ●MINUS

SQL> select \* from a\_student

2    **minus**

3    select \* from b\_student;

STU_NO	STU_NAME	STU_DEPT	STU_GRADE	STU_CLASS	STU_GENDER	STU_HEIGHT	STU_WEIGHT
20153075	옥한빛	기계	1	C	M	177	80
20153088	이태연	기계	1	C	F	162	50
20143054	유가인	기계	2	C	F	154	47



**4장을 마치며.....**

**Q & A**