

--ex11) self 조인 : 자기 자신의 테이블과 조인하는 경우 사원과 관리자를 연결하시오

```
select EMPLOYEE_ID,MANAGER_ID,LAST_NAME from EMPLOYEES e;  
select EMPLOYEE_ID, LAST_NAME from EMPLOYEES m;
```

--방법 1

```
select e.EMPLOYEE_ID as 사원번호,  
       e.last_name as 사원이름,  
       m.LAST_NAME as 관리자  
from EMPLOYEES e, EMPLOYEES m  
where m.EMPLOYEE_ID = e.MANAGER_ID;
```

--방법 2

```
select e.EMPLOYEE_ID as 사원번호,  
       e.last_name as 사원이름,  
       m.LAST_NAME as 관리자  
from EMPLOYEES e  
join EMPLOYEES m on(m.EMPLOYEE_ID = e.MANAGER_ID);
```

--ex13) Non Equijoin 컬럼값이 같은 경우가 아닌 범위에 속하는지 여부를 확인 할 때

```
on (컬럼명 between 컬럼명1 and 컬럼명2);
```

```
create table salgrade(  
salvel varchar2(2),  
lowst number,  
highst number);
```

```
insert into salgrade values('A',20000,29999);  
insert into salgrade values('B',10000,19999);  
insert into salgrade values('C',0,9999);  
commit;
```

```
select * from salgrade;
```

```
select LAST_NAME, salary, salvel  
from EMPLOYEES  
join salgrade on(salary between lowst and highst)  
order by salary desc;
```

/*[SET operator]-집합연산자

두 개 이상의 쿼리 결과를 하나로 결합시키는 연산자

1.UNION : 양쪽쿼리를 모두 포함(중복 결과는 1번만 포함)

2.UNION ALL : 양쪽 쿼리를 모두 포함(중복 결과도 모두 포함)

3.INTERSECT : 양쪽 쿼리 결과에 모두 포함되는 행만 표현

4.MINUS : 쿼리1 결과에 포함되고 쿼리2 결과에는 포함되지 않는 행만 표현
*/

```
create table employees_role as select * from employees
where 1=0;
select * from employees;
select * from employees_role;
insert into employees_role values(101, 'Neena', 'Kochhar',
'515.123.4568',
                                '1989-09-21', '1',
17000.00,100, 90);
insert into employees_role values(101, 'Neena', 'Kochhar',
'515.123.4568',
                                '1989-09-21', '3',
17000.00,100, 90);
insert into employees_role values(101, 'Nee', 'Ko',
'515.123.4568',
                                '1989-09-21', '2',
17000.00,100, 90);
insert into employees_role values(200, 'Neena', 'Kochhar',
'515.123.4568',
                                '1989-09-21', '2',
17000.00,100, 90);
insert into employees_role values(200, 'Nee', 'Kochhar',
'515.123.4568',
                                '1989-09-21', '3',
17000.00,100, 90);
insert into employees_role values(300, 'GilDong', 'Hong',
'010-123-4567',
                                '2009-03-01', '4',
23000.00 ,100, 90);
commit;
```

--ex1) union 합집합인데 중복된 값은 1번만

--employee_id, last_name이 같을 경우 중복제거 하시오

```
select EMPLOYEE_ID, LAST_NAME from EMPLOYEES
union
select EMPLOYEE_ID, LAST_NAME from employees_role;
```

--ex2) union all 합집합인데 중복된 값 전부

```
select EMPLOYEE_ID, LAST_NAME from EMPLOYEES
union all
select EMPLOYEE_ID, LAST_NAME from employees_role;
```

```
select salary from EMPLOYEES where DEPARTMENT_ID=10
union all
select salary from EMPLOYEES where DEPARTMENT_ID=30
order by 1;
```

```

--ex3) minus 차집합 중복포함 쿼리2를 전부 뺀 쿼리1
select EMPLOYEE_ID, LAST_NAME from EMPLOYEES
minus
select EMPLOYEE_ID, LAST_NAME from employees_role;

--ex4) intersect 교집합 중복된 값만 추출
select EMPLOYEE_ID, last_name from EMPLOYEES
intersect
select EMPLOYEE_ID, LAST_NAME from employees_role;
/*[문제1] employees 와 employees_role에서 레코드의 사원명단을 구하시오
조건 1) 사원이름, Job_id, Department_ID를 표시하시오
조건 2) employees 에서는 부서ID가 10인 사원만 검색
employees_role에서는 job_id 3만 검색
조건 3) 중복되는 레코드는 제거
*/
select LAST_NAME, JOB_ID, DEPARTMENT_ID from EMPLOYEES where
DEPARTMENT_ID =10
union
select LAST_NAME, JOB_ID, DEPARTMENT_ID from employees_role
where JOB_ID = 3;

--ex5) SET operator와 IN operator관계
--Job title이 'Stock Manager'또는 'Programmer'인 사원들의 사원명과
Job_title을 표시하시오
select LAST_NAME, JOB_TITLE
from EMPLOYEES
join JOBS using (job_id)
where JOB_TITLE in('Stock Manger', 'Programmer');

--ex6) 컬럼명이 다른 경우의 SET operator 쿼리 1과 쿼리2의 select 목록
은 반드시 동일(컬럼 수, 데이터 타입)해야 하므로 이를 위해 Dummy Column을 사
용할 수 있다.
select LAST_NAME, EMPLOYEE_ID, HIRE_DATE
from EMPLOYEES
where DEPARTMENT_ID=10
union
select DEPARTMENT_NAME, DEPARTMENT_ID, null
from DEPARTMENTS
where DEPARTMENT_ID=10;

```