[SET operator] - 집합연산자

두 개 이상의 쿼리 결과를 하나로 결합시키는 연산자

1. UNION : 양쪽 쿼리를 모두 포함(중복 결과는 1번만 포함) → 합집합
2. UNION ALL : 양쪽 쿼리를 모두 포함(중복 결과도 모두 포함)
3. INTERSECT : 양쪽 쿼리 결과에 모두 포함되는 행만 표현 → 교집합
4. MINUS : 쿼리1 결과에 포함되고 쿼리2 결과에는 포함되지 않는 행만 표현 → 차집합

오라클의 집합연산자(SET operator) UNION, INTERSECT, MINUS 는 order by 한다.

→ UNION을 사용할 경우 합쳐진 결과에서 중복을 제거한 결과를 반환한다.
중복을 제거하기 위해서 order by 하기 때문에 컬럼이 많으면 느려진다.
수가 작은 튜플로 가공 후 사용 하는게 좋다

→ UNION ALL 는 order by 하지 않고 무조건 합해준다
Order by를 하려면 두 번째 쿼리에 작성해야 한다.

s e l e c t . .

```
create table employees_role as select * from employees where 1=0;
```
← 테이블 구조만 복사

```
select * from employees;
select * from employees_role;

insert into employees_role values(101, 'Neena', 'Kochhar', 'NKOCHHAR', '515.123.4568',
'1989-09-21', 'AD_VP', 17000.00, NULL, 100, 90);
insert into employees_role values(101, 'Neena', 'Kochhar', 'NKOCHHAR', '515.123.4568',
'1989-09-21', 'AD_VP', 17000.00, NULL, 100, 90);
insert into employees_role values(101, 'Nee', 'Ko', 'NKOCHHAR', '515.123.4568',
'1989-09-21', 'AD_VP', 17000.00, NULL, 100, 90);
insert into employees_role values(200, 'Neena', 'Kochhar', 'NKOCHHAR', '515.123.4568',
'1989-09-21', 'AD_VP', 17000.00, NULL, 100, 90);
insert into employees_role values(200, 'Nee', 'Kochhar', 'NKOCHHAR', '515.123.4568',
'1989-09-21', 'AD_VP', 17000.00, NULL, 100, 90);
insert into employees_role values(300, 'GilDong', 'Hong', 'NKOCHHAR', '010-123-4567',
'2009-03-01', 'IT_PROG', 23000.00, NULL, 100, 90);

commit;
```

ex1) union
employee_id, last_name이 같을 경우 중복제거 하시오 → 110 레코드

```
select employee_id, last_name from employees
union
select employee_id, last_name from employees_role;
```

ex2) union all
employee_id, last_name이 같을 경우 중복을 허용 하시오 → 113 레코드

```
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
employees_role과 중복되는 레코드는 제거하고 employees에만 있는 사원명단을 구하시오
(단, employee_id, last_name만 표시) → 106 레코드

```
select employee_id, last_name from employees
minus
select employee_id, last_name from employees_role;
```

ex4) intersect
employees와 employees_role에서 중복되는 레코드의 사원명단을 구하시오
(단, employee_id, last_name만 표시) → 1 레코드

```
select employee_id, last_name from employees
intersect
select employee_id, last_name from employees_role;
```

[문제1] employees와 employees_role에서 레코드의 사원명단을 구하시오
조건1) 사원이름, 업무ID, 부서ID을 표시하시오
조건2) employees 에서는 부서ID가 10인 사원만 검색
employees_role에서는 업무ID가 IT_PROG만 검색
조건3) 중복되는 레코드는 제거

ex5) SET operator과 IN operator관계
job_title이 'Stock Manager' 또는 'Programmer'인 사원들의 사원명과 job_title을 표시하시오

```
last_name         job_title
-------------------------------
Kaufling          StockManager
Hunlod            Programmer
            :
```

방법1 (join, in연산자 이용)

```
select last_name, job_title
from employees
join jobs using(job_id)
where job_title in('Stock Manager', 'Programmer');
```

방법2 (join, union 이용)

```
select last_name, job_title
from employees
join jobs using(job_id)
where job_title='Stock Manager'
union
select last_name, job_title
from employees
join jobs using(job_id)
where job_title='Programmer'
order by 2;
```

ex9) 컬럼명이 다른 경우의 SET operator
쿼리1과 쿼리2의 select 목록은 반드시동일(컬럼 개수, 데이터 타입)해야 하므로 이를 위해 Dummy Column을 사용할 수 있다.

```
select last_name, employee_id, hire_date
from employees
where department_id=20
union
select department_name, department_id, NULL
from departments
where department_id=20;
```