

조인(Join)

<조인 유형>

- NATURAL JOIN, USING절 JOIN, ON절 JOIN
- LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN
- CROSS JOIN

• NATURAL JOIN

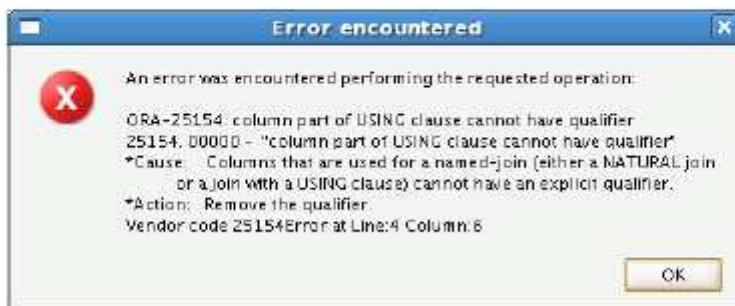
```
SELECT department_id, department_name,  
       location_id, city  
FROM   departments  
NATURAL JOIN locations ;
```

• USING절 JOIN

```
SELECT employee_id, last_name,  
       location_id, department_id  
FROM   employees JOIN departments  
USING (department_id) ;
```

(주의) Using절에서 참조하고 있는 컬럼명 앞에는 테이블명 및 테이블 Alias 사용 금지

```
SELECT l.city, d.department_name  
FROM   locations l JOIN departments d  
USING (location_id)  
WHERE d.location_id = 1400;
```



- ON절 JOIN

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM   employees e JOIN departments d
ON     (e.department_id = d.department_id);
```

```
SELECT employee_id, city, department_name
FROM   employees e
JOIN   departments d
ON     d.department_id = e.department_id
JOIN   locations l
ON     d.location_id = l.location_id;
```

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM   employees e JOIN departments d
ON     (e.department_id = d.department_id)
WHERE  e.manager_id = 149 ;
```

- ON절 JOIN – Self Join

EMPLOYEES (WORKER)

EMPLOYEE_ID	LAST_NAME	MANAGER_ID
200	Whalen	101
201	Hartstein	100
202	Fay	201
205	Higgins	101
206	Gietz	205
100	King	(null)
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
200	Whalen
201	Hartstein
202	Fay
205	Higgins
206	Gietz
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst

...

WORKER 테이블의 MANAGER_ID는 MANAGER
테이블의 EMPLOYEE_ID와 같습니다.

```
SELECT worker.last_name emp, manager.last_name mgr
FROM   employees worker JOIN employees manager
ON     (worker.manager_id = manager.employee_id);
```

- Outer Join : LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN

Inner Join	Outer Join
조인 조건을 만족하는 행만 반환 - Natural Join - Using절 Join - On절 Join	조인 조건을 만족하지 않는 행까지 반환 - Left Outer Join - Right Outer Join - Full Outer Join

- Left Outer Join

```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e LEFT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

- Right Outer Join

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e RIGHT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

- Full Outer Join

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

- CROSS JOIN

```
SELECT last_name, department_name
FROM   employees
CROSS JOIN departments ;
```

<Quiz>

1. employees 테이블로부터 성이 "n"으로 끝나는 사원의 수를 구하는 쿼리구문을 작성하시오.

[case1] like 비교연산자 사용

[case2] substr 함수 사용

	COUNT(*)
1	3

```
SELECT COUNT(*)
FROM employees
WHERE last_name LIKE '%n';
```

```
SELECT COUNT(*)
FROM employees
WHERE SUBSTR(last_name, -1) = 'n';
```

2. employees 테이블과 departments 테이블로부터 각 부서에 대한 부서번호, 부서이름, 위치 및 사원 수를 보여주는 쿼리구문을 작성하시오.
단, 사원이 없는 부서도 출력을 시키시오.

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	COUNT(E.EMPLOYEE_ID)
1	80	Sales	2500	3
2	110	Accounting	1700	2
3	10	Administration	1700	1
4	60	IT	1400	3
5	20	Marketing	1800	2
6	90	Executive	1700	3
7	50	Shipping	1500	5
8	190	Contracting	1700	0

```
SELECT d.department_id, d.department_name,
       d.location_id, COUNT(e.employee_id)
FROM employees e RIGHT OUTER JOIN departments d
ON e.department_id = d.department_id
GROUP BY d.department_id, d.department_name, d.location_id;
```

3. employees 테이블로부터 각 월의 16일 이전에 채용된 사원을 모두 출력하는 쿼리구문을 작성하시오.

	LAST_NAME	HIRE_DATE
1	De Haan	13-JAN-93
2	Hunold	03-JAN-90
3	Lorentz	07-FEB-99
4	Matos	15-MAR-98
5	Vargas	09-JUL-98
6	Abel	11-MAY-96
7	Higgins	07-JUN-94
8	Gietz	07-JUN-94

```
SELECT last_name, hire_date  
FROM employees  
WHERE TO_CHAR(hire_date, 'DD') < 16;
```