

데이터베이스 SQL 작성하기_다중테이블 검색



다중테이블 검색

employee 직원에 대한 정보

emp_no	name	dept	position	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

department 부서에 대한 정보

dept_no	dept_name	location
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

데이터 중복을 최소화하기 위해 데이터를 테이블로 분해하여 저장

(직원 정보, 부서 정보 구별하여 각 테이블에 저장)

필요시 두 테이블을 연계하여 필요한 정보를 검색

(직원의 부서에 대한 정보가 필요할 시 두 테이블 결합)

두 테이블을 결합하여 필요한 데이터를 찾는 것을 조인(JOIN)

두 테이블이 결합방식에 따라 다양한 조인 형태

내부 조인(INNER JOIN)

```
SELECT 컬럼명1, 컬럼명2..  
FROM 테이블명1 INNER JOIN 테이블명2  
ON 조인 조건;
```

```
SELECT emp_no, name, dept_name  
FROM employee as e INNER JOIN department as d  
ON e.department = d.dept_no;
```

- employee 테이블의 department 값과 department 테이블의 dept_no값이 같은 레코드들을 조인
- AS 는 별칭,
employee 를 e라는 별칭으로 사용

내부 조인(INNER JOIN)

employee 직원에 대한 정보

emp_no	name	depart	position	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

department 부서에 대한 정보

dept_no	dept_name	location
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

INNER JOIN

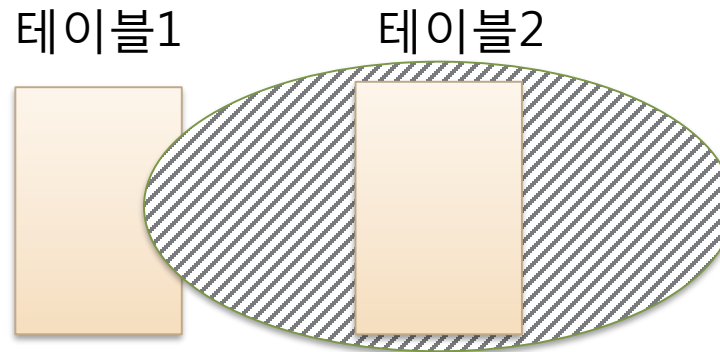
emp_no	name	dept_name
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부

```
SELECT emp_no, name, dept_name
FROM employee as e INNER JOIN department as d
ON e.depart = d.dept_no;
```

오른쪽 외부 조인(RIGHT JOIN)

```
SELECT 컬럼명1, 컬럼명2..  
FROM 테이블명1 RIGHT JOIN 테이블명2  
ON 조인 조건;
```

조인된 테이블1(왼쪽 테이블)의 내용을 오른쪽 테이블의 모든 내용과 출력



```
SELECT emp_no, name, dept_name FROM employee as e  
RIGHT JOIN department as d ON e.depart = d.dept_no;
```

오른쪽 외부 조인 (RIGHT JOIN)

employee 직원에 대한 정보

emp_no	name	depart	position	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부상	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

department 부서에 대한 정보

dept_no	dept_name	location
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

RIGHT JOIN

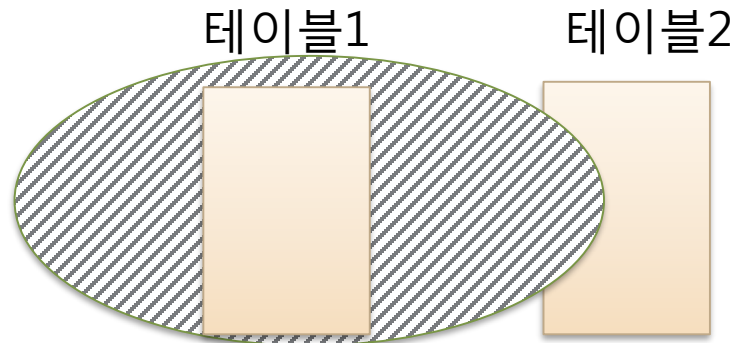
emp_no	name	dept_name
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부
NULL	NULL	총무부
NULL	NULL	기획부

```
SELECT emp_no, name, dept_name FROM  
employee as e RIGHT JOIN department as d ON  
e.depart = d.dept_no;
```

왼쪽 외부 조인(LEFT JOIN)

```
SELECT 컬럼명1, 컬럼명2..  
FROM 테이블명1 LEFT JOIN 테이블명2  
ON 조인 조건;
```

조인된 테이블2(오른쪽 테이블)의 내용을 왼쪽 테이블의 모든 내용과 출력



```
SELECT emp_no, name, dept_name FROM employee as e  
LEFT JOIN department as d ON e.dept = d.dept_no;
```

왼쪽 외부 조인(LEFT JOIN)

employee 직원에 대한 정보

emp_no	name	dept	position	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1996-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000
1005	김미나	5	사원	F	2018-03-01	1800000

department 부서에 대한 정보

dept_no	dept_name	location
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

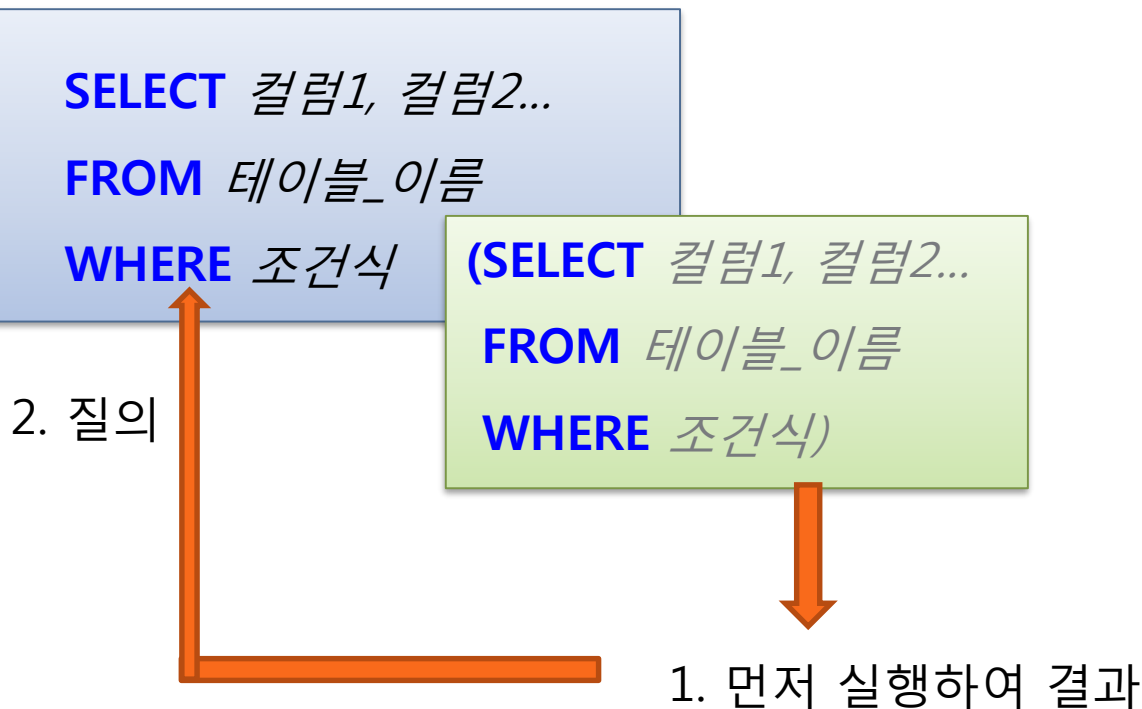
LEFT JOIN

```
SELECT emp_no, name, dept_name FROM  
employee as e LEFT JOIN department as d ON  
e.dept = d.dept_no;
```

emp_no	name	dept_name
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부
1005	김미나	NULL

서브 쿼리문

- SQL문 안에 SQL문 포함
- WHERE, FROM에 서브 쿼리문을 작성, 서브쿼리문을 실행한 후 그 결과를 대상으로 질의



서브 쿼리문

회사의 평균 연봉보다 더 많이 받는 직원의 이름, 직급 검색

```
SELECT name, position FROM employee
WHERE salary > (SELECT AVG(salary) FROM employee);
```

emp_no	name	department	position	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000
1005	김미나	5	사원	F	2018-03-01	1800000

```
SELECT AVG(salary) FROM employee
```

3960000

```
SELECT name, position FROM employee
```

```
WHERE salary > 3960000
```

집합연산(UNION)

```
SELECT 컬럼1, 컬럼2... FROM 테이블1  
UNION
```

```
SELECT 컬럼1, 컬럼2... FROM 테이블2
```

- 각 질의문 결과 합집합
- 자동으로 중복된 데이터를 배제하고, order by정렬

```
SELECT 컬럼1, 컬럼2... FROM 테이블1  
UNION ALL
```

```
SELECT 컬럼1, 컬럼2... FROM 테이블2
```

- 각 질의문 결과 합집합
- 중복된 데이터도 모두 보여주고 정렬하지 않음

GROUP BY

```
SELECT 컬럼1, 컬럼2...  
FROM 테이블_이름  
WHERE 조건식  
GROUP BY column_name(s);
```

- 그룹 지어 그룹별 값을 계산. 예) 인사부 직원들의 평균 급여
- 그룹별 값을 계산할 때는 집합 함수(sum, avg) 를 사용

```
SELECT COUNT(emp_no)  
FROM employee  
WHERE depart = 2  
GROUP BY department ;
```

부서 2번에 소속된 사람들의 인원

GROUP BY 와 HAVING

HAVING은 그룹에 대한 조건식

```
SELECT 컬럼1, 컬럼2...  
FROM 테이블_이름  
WHERE 조건식  
GROUP BY 컬럼명  
HAVING 조건식
```

- 2명 이상의 부서원을 가진 부서명 검색

```
SELECT dept_name  
FROM employee INNER JOIN department  
ON dept_no = depart  
GROUP BY dept_no  
HAVING count(emp_no) > 2
```