6. 고급 SQL 작성하기
다중테이블 검색

다중테이블 검색

employee 직원에 대한 정보

emp_no	emp_nm	dept_no	pos	hire_date	Salary	gender
1001	구창민	1	과장	1995-05-01	5000000	M
1002	김민서	1	사원	2017-09-01	2500000	M
1003	이은영	2	부장	1990-09-01	5500000	F
1004	한성일	2	과장	1993-04-01	5000000	M
1005	홍길동	8	과장	2020-09-22	4000	M

department부서에 대한 정보

dept_no	dept_nm	loc
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

데이터 중복을 최소화하기 위해 테이터를 테이블로 분해하여 저장

(직원 정보, 부서 정보 구별하여 각 테이블에 저장)

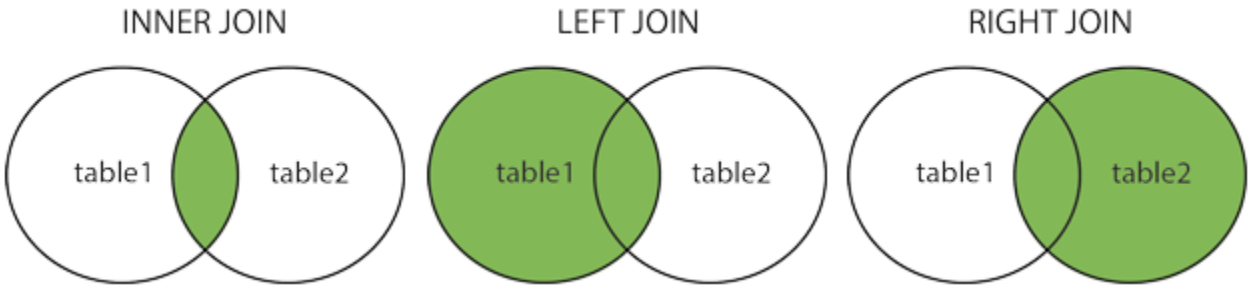
필요시 **두 테이블을 연계**하여 필요한 정보를 검색

(직원의 부서에 대한 정보가 필요할 시 두 테이블 결합)

두 테이블을 결합하여 필요한 데이터를 찾는 것을 **조인(JOIN)** 이라고 함.

두 테이블 결합 방식에 따라 다양한 조인 형태가 있음.

다중테이블 검색



(INNER) JOIN

2개 테이블에 모두 일치하는 데이터가 있는 튜플(행)을 검색

LEFT (OUTER) JOIN

왼쪽 테이블은 모든 행, 오른쪽 테이블은 일치하는 데이터가 있는 행만 검색

RIGHT (OUTER) JOIN

오른쪽 테이블은 모든 행, 왼쪽 테이블은 일치하는 데이터가 있는 행만 검색

내부 조인(INNER JOIN)

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

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

INNER JOIN 또는 JOIN만 써도 됨.

- employee 테이블의 dept_no 값과 department 테이블의 dept_no **값이 같은 레코드들을 조인**

- **as 는 별칭**,
  employee 테이블을 별칭 e로, department 테이블을 별칭 d로 사용

내부 조인(INNER JOIN)

employee 직원에 대한 정보

emp_no	emp_nm	dept_no	pos	hire_date	Salary	gender
1001	구창민	1	과장	1995-05-01	5000000	M
1002	김민서	1	사원	2017-09-01	2500000	M
1003	이은영	2	부장	1990-09-01	5500000	F
1004	한성일	2	과장	1993-04-01	5000000	M
1005	홍길동	8	과장	2020-09-22	4000	M

department 부서에 대한 정보

dept_no	dept_nm	loc
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

INNER JOIN

홍길동은 결과에 없음.

emp_no	emp_nm	dept_nm
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부

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

오른쪽 외부 조인(RIGHT OUTER JOIN)

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

```
SELECT e.emp_no, e.emp_nm, d.dept_nm
FROM employee as e RIGHT JOIN department as d
ON e.dept_no = d.dept_no;
```

오른쪽에 있는 department 테이블을 기준으로 이 테이블의 모든 데이터 출력.
department 테이블의 dept_no가 employee에 없으면 employee테이블의
값은 null로 출력

오른쪽 외부 조인 (RIGHT JOIN)

employee 직원에 대한 정보

emp_no	emp_nm	dept_no	pos	hire_date	Salary	gender
1001	구창민	1	과장	1995-05-01	5000000	M
1002	김민서	1	사원	2017-09-01	2500000	M
1003	이은영	2	부장	1990-09-01	5500000	F
1004	한성일	2	과장	1993-04-01	5000000	M
1005	홍길동	8	과장	2020-09-22	4000	M

department 부서에 대한 정보

dept_no	dept_nm	loc
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

RIGHT JOIN

emp_no	emp_nm	dept_nm
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부
NULL	NULL	총무부
NULL	NULL	기획부

```
SELECT e.emp_no, e.emp_nm, d.dept_nm
FROM employee as e RIGHT JOIN department as d
ON e.dept_no = d.dept_no;
```

왼쪽 외부 조인(LEFT JOIN)

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

```
SELECT e.emp_no, e.emp_nm, d.dept_nm
FROM employee as e LEFT JOIN department as d
ON e.dept_no = d.dept_no;
```

왼쪽에 있는 employee 테이블을 기준으로 이 테이블의 모든 데이터 출력.
employee 테이블의 dept_no가 departmen에 없으면 department테이블의 값은 null로 출력

왼쪽 외부 조인(LEFT JOIN)

employee 직원에 대한 정보

emp_no	emp_nm	dept_no	pos	hire_date	Salary	gender
1001	구창민	1	과장	1995-05-01	5000000	M
1002	김민서	1	사원	2017-09-01	2500000	M
1003	이은영	2	부장	1990-09-01	5500000	F
1004	한성일	2	과장	1993-04-01	5000000	M
1005	홍길동	8	과장	2020-09-22	4000	M

department 부서에 대한 정보

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

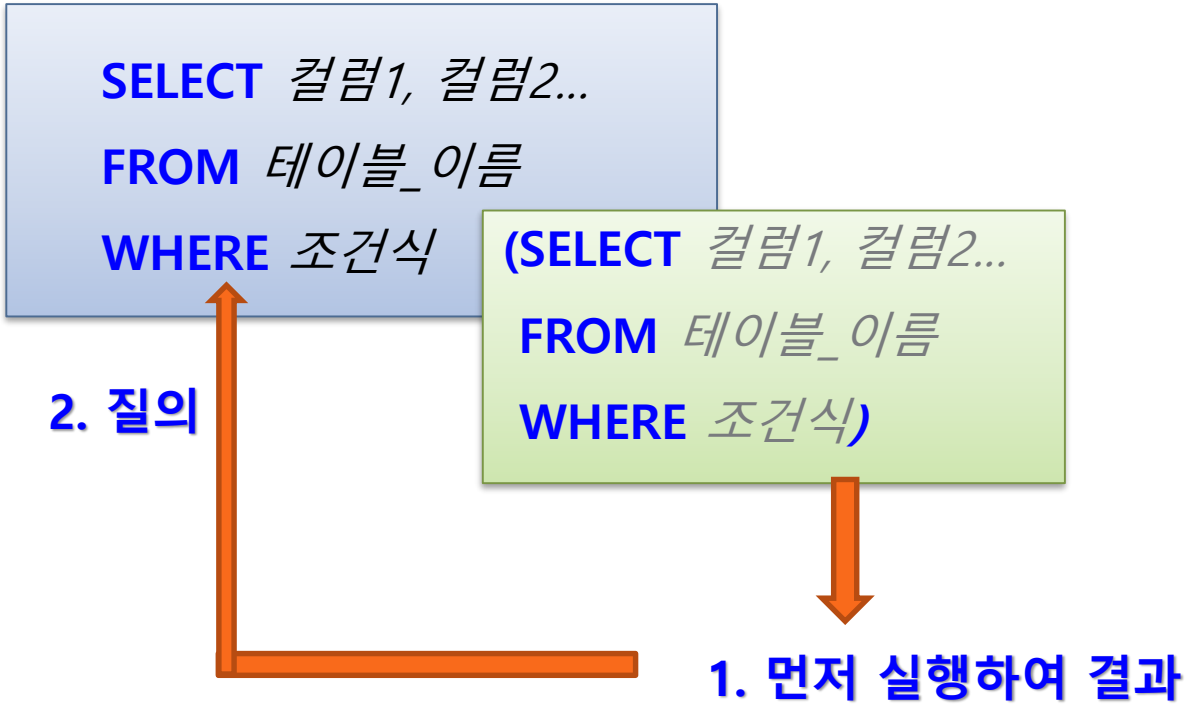
LEFT JOIN

emp_no	emp_nm	dept_nm
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	인사부
1004	한성일	인사부
1005	홍길동	NULL

```
SELECT e.emp_no, e.emp_nm, d.dept_nm
FROM employee as e LEFT JOIN department as d
ON e.dept_no = d.dept_no;
```

서브 쿼리문

- SQL문 안에 SQL문 포함
- WHERE, FROM에 서브 쿼리문을 작성, **서브쿼리문을 먼저 실행**한 후 그 결과를 대상으로 질의



서브 쿼리문

회사의 평균 연봉보다 더 많이 받는 직원의 이름, 직급 검색

```
SELECT emp_nm, pos FROM employee
WHERE salary > ( SELECT avg(salary) FROM employee );
```

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

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 by 정렬**

```
SELECT 컬럼1, 컬럼2... FROM 테이블1
UNION ALL
SELECT 컬럼1, 컬럼2... FROM 테이블2
```

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

- 2개 select문의 검색 **컬럼 개수**가 같아야 한다.
- 대응되는 컬럼은 **유사한 데이터형**이어야 한다.

집합연산(UNION)

employee

emp_no	emp_nm
1001	구창민
1002	김민서
1003	이은영
1004	한성일
1005	홍길동

employee1

emp_no	emp_nm
1002	김민서
1003	이은영
1004	한성일
1006	이여은

UNION

```
MariaDB [com_1]> select emp_no, emp_nm from employee union select emp_no, emp_nm from employee1;
+--------+-----------+
| emp_no | emp_nm    |
+--------+-----------+
| 1001   | 구창민    |
| 1002   | 김민서    |
| 1003   | 이은영    |
| 1004   | 한성일    |
| 1005   | 홍길동    |
| 1006   | 이여은    |
+--------+-----------+
6 rows in set (0.003 sec)
```

1002 김민서, 1003 이은영 **중복데이터로 1번만** 나옴.

emp_no를 기준으로 정렬됨.

집합연산(UNION)

employee

emp_no	emp_nm
1001	구창민
1002	김민서
1003	이은영
1004	한성일
1005	홍길동

employee1

emp_no	emp_nm
1002	김민서
1003	이은영
1004	한성일은
1006	이여은

UNION ALL

```
MariaDB [com_1]> select emp_no, emp_nm from employee union all select emp_no, emp_nm from employee1;
+--------+----------+
| emp_no | emp_nm   |
+--------+----------+
| 1001   | 구창민   |
| 1002   | 김민서   |
| 1003   | 이은영   |
| 1004   | 한성일   |
| 1005   | 홍길동   |
| 1002   | 김민서   |
| 1003   | 이은영   |
| 1004   | 한성일은 |
| 1006   | 이여은   |
+--------+----------+
9 rows in set (0.000 sec)
```

1002 김민서, 1003 이은영 **중복데이터 2번** 나옴.

정렬 안됨.

집합연산(UNION)

부서별 평균급여

```
MariaDB [com_4]> select dept_no, avg(salary) from employee group by dept_no;
+---------+--------------+
| dept_no | avg(salary)  |
+---------+--------------+
|       1 | 3750000.0000 |
|       2 | 5250000.0000 |
|       8 |    4000.0000 |
+---------+--------------+
```

회사 평균급여

```
MariaDB [com_3]> select avg(salary) from employee;
+--------------+
| avg(salary)  |
+--------------+
| 3600800.0000 |
+--------------+
```

UNION으로 한번에 보기

```
MariaDB [com_4]> select dept_no, avg(salary) from employee group by dept_no union select null, avg(salary) from employee;
+---------+--------------+
| dept_no | avg(salary)  |
+---------+--------------+
|       1 | 3750000.0000 |
|       2 | 5250000.0000 |
|       8 |    4000.0000 |
|    NULL | 3600800.0000 |
+---------+--------------+
```

```
MariaDB [com_4]> select dept_no, avg(salary) from employee group by dept_no union select "총 평균급여", avg(salary) from employee;
+------------+--------------+
| dept_no    | avg(salary)  |
+------------+--------------+
| 1          | 3750000.0000 |
| 2          | 5250000.0000 |
| 8          |    4000.0000 |
| 총 평균급여 | 3600800.0000 |
+------------+--------------+
```