Database Part2

데이터 조회하기
조회 조건 추가하기
BETWEEN과 IN 연산자
조회데이터 정렬하기
LIKE 연산자와 와일드카드
DISTINCT로 중복 조회 데이터 제외하기

Database | 데이터 조회하기 - 1

데이터베이스에 저장된 데이터를 조회할 때는 SELECT 쿼리를 사용한다.

모든 쿼리문의 마지막에는 세미콜론을 작성한다.

SELECT 쿼리의 기본 문법

SELECT 조회할 컬럼명들 FROM 조회할 테이블명 WHERE 조회 조건;

SELECT EMPNO FROM EMP;

-> EMP 테이블에서 모든 사번 컬럼을 조회한다.

SELECT EMPNO, ENAME, SAL FROM EMP;

-> EMP 테이블에서 모든 사번, 사원명, 급여 컬럼을 조회한다.

SELECT * FROM EMP;

-> EMP 테이블에서 모든 컬럼을 조회한다. (* -> 에스테리스크라 부르며, 모든, all의 의미를 지님)

Database | 데이터 조회하기 - 2

조회 시 반드시 컬럼명만 작성할 수 있는 건 아니며, 숫자나 문자도 조회문에서 사용할 수 있다.

다만, 조회되는 데이터의 수는 조회할 테이블에 저장된 데이터 수와 동일하게 조회된다.

SELECT 3, 'DB' FROM EMP;

-> EMP 테이블의 데이터 수만큼 3과 'DB' 문자를 조회

SELECT 3+3, 'DB' FROM EMP;

-> EMP 테이블의 데이터 수만큼 6과 ' DB' 문자를 조회

SELECT 3, 'DB';

-> 3과 'DB'를 조회한다. FROM 절이 없기 때문에 테이블에 저장된 데이터수만큼 조회되지 않고 하나의 결과만 나온다.

FROM절을 사용하지 않는 이러한 쿼리문은 주로 문법 확인용으로 사용한다.

Database | 데이터 조회하기 - 3

컬럼 조회 시 각 컬럼의 조회 결과 이름을 별칭을 사용하여 변경 할 수 있다.

별칭은 사용하는 쿼리문에 한해 적용되는 것으로, 기존 데이터의 컬럼명이 바뀌는 것은 아니다.

별칭은 컬럼뿐만 아니라 테이블명에도 사용할 수 있다.

테이블명에 별칭을 사용하는 것은 JOIN과 같은 복잡한 쿼리문을 작성할 때 사용한다.

SELECT EMPNO AS 사번, ENAME AS 사원명 FROM EMP;

-> EMP 테이블에서 모든 EMPNO, ENAME 컬럼을 조회하되, 각각 사번, 사원명의 별칭으로 조회된다.

SELECT EMPNO 사번, ENAME 사원명 EMP;

-> 위와 같은 쿼리문이지만 AS 키워드를 생략해도 별칭을 사용할 수 있다.

SELECT EMPNO 사번, ENAME 사원명 FROM EMP 사원정보;

-> 테이블명에도 별칭을 사용할 수 있다.

테이블명에서 별칭 사용 시 AS 키워드는 DBMS 종류에 따라 허용 여부가 다르다.

MariaDB에서는 AS 사용과 생략 모두를 허용하지만, ORACLE에서는 AS 키워드를 생략해야 한다.

Database | 조회 조건 추가하기 - 1

WHERE절을 사용하면 특정 조건에 맞는 데이터만 조회 할 수 있다.

WHERE절은 FROM절 다음에 작성한다.

조건은 여러개 작성할 수 있으며, 여러 조건을 동시에 만족해야 할 때는 AND, 여러 조건 중 하나만 만족해도 될 때는 OR 키워드로 연결한다.

비교 연산자는 >, <, >=, <=, =, !=, <> 를 사용한다. 데이터베이스에서 '='은 같다는 의미이다.

문자는 반드시 홀따옴표에 감싸서 표현한다.

```
SELECT ENAME, JOB, SAL FROM EMP
WHERE SAL >= 300;
```

-> EMP 테이블에서 급여가 300이상인 사원의 사원명, 직급, 급여를 조회

```
SELECT EMPNO, ENAME, JOB FROM EMP
WHERE JOB = '대리';
```

-> EMP테이블에서 직급이 대리인 사원의 사번, 사원명, 직급을 조회

Database | 조회 조건 추가하기 - 2

SELECT ENAME, JOB, HIREDATE FROM EMP

WHERE SAL >= 300 AND JOB = '과장';

-> EMP 테이블에서 급여가 3000이상이면서 직급이 과정인 사원의 사원명, 직급, 입사일을 조회

SELECT * FROM EMP

WHERE SAL <= 400 OR SAL >= 900;

-> EMP테이블에서 급여가 400이하이거나 900이상인 사원들의 모든 컬럼 조회

SELECT * FROM EMP

WHERE JOB != '사원'

-> EMP테이블에서 직급이 사원이 아닌 사원들의 모든 컬럼 조회

Database | 조회 조건 추가하기 - 3

WHERE 조건절에서 NULL 데이터를 검사할 때는 주의가 필요하다.

NULL 데이터는 '=', '!=' 로 검사하지 않고 IS, IS NOT 키워드를 사용한다.

```
SELECT * FROM EMP
WHERE COMM IS NULL;
```

-> EMP 테이블에서 커미션이 NULL인 사원의 모든 컬럼 조회

```
SELECT * FROM EMP
WHERE COMM IS NOT NULL
AND SAL >= 300;
```

-> EMP테이블에서 커미션이 NULL이 아니고 급여가 300이상인 사원들의 모든 컬럼 조회

Database | 중간 점검

1. EMP 테이블에서 모든 사원의 사번, 사원명, 급여, 커미션을 조회하시오.

2. EMP 테이블에서 사번이 1005번 이상인 사원들의 사번, 사원명, 직급을 조회하되, 사원명은 'NAME'이라는 별칭으로 조회하시오.

3. EMP 테이블에서 직급이 대리이거나 과장인 사원들의 사원명, 직급, 급여를 조회하시오.

4. EMP 테이블에서 급여가 300 이상이면서 커미션이 300 이상인 사원들의 모든 컬럼을 조회하시오.

5. EMP 테이블에서 급여가 900 이하이고 커미션은 NULL이 아니며 직급은 대리이거나 과장인 사원들의 사원명, 직급, 급여, 커미션을 조회하시오.

Database | between 연산자와 in 연산자 - 1

다음은 급여가 3000이상이면서 8000이하인 사원의 모든 컬럼을 조회하는 쿼리문이다.

```
SELECT * FROM EMP
WHERE SAL >= 300 AND SAL <= 800;
```

의 쿼리문은 아래와 같이 BETWEEN A AND B 문법을 사용할 수 있다.

```
SELECT * FROM EMP
WHERE SAL BETWEEN 300 AND 800;
```

이처럼, BETWEEN A AND B 문법을 사용하면 컬럼의 값이 특정 범위에 해당하는지 파악할 수 있다.

위의 예제처럼 A와 B 값을 포함한 데이터를 조회한다는 것에 주의하자.

다음은 사번이 1003번에서 1007번 사이의 사원들의 사번, 사원명, 급여를 조회한 쿼리문이다.

```
SELECT EMPNO, ENAME, SAL FROM EMP
WHERE EMPNO BETWEEN 1003 AND 1007;
```

Database | between 연산자와 in 연산자 - 2

다음은 급여가 300이거나 400인 사원의 모든 컬럼을 조회하는 쿼리문이다.

```
SELECT * FROM EMP
WHERE SAL = 300 OR SAL = 400;
```

의 쿼리문은 아래와 같이 IN (A, B ...)문법을 사용할 수 있다.

```
SELECT * FROM EMP
WHERE SAL IN (300, 400);
```

이처럼, IN 연산자를 사용하면 OR 연산을 대체할 수 있다.

IN 연산자의 소괄호 안에는 필요한 만큼 데이터를 넣을 수 있다.

다음은 사번이 1003이거나 1005이거나 1007인 사원들의 사번, 사원명, 급여를 조회한 쿼리문이다.

```
SELECT EMPNO, ENAME, SAL FROM EMP
WHERE EMPNO IN (1003, 1005, 1007);
```

Database | 조회 데이터 정렬하기 - /

조회 쿼리의 마지막에 ORDER BY절을 작성하면 조회 데이터를 특정 컬럼 기준으로 정렬하여 조회할 수 있다.

```
SELECT * FROM EMP
ORDER BY SAL ASC;
```

-> EMP 테이블에서 사원들의 모든 컬럼 정보를 조회하되, 급여 기준 오름차순으로 정렬하여 조회함.

정렬하여 조회할 때는 다음과 같은 문법을 따른다.

- 정렬을 위한 ORDER BY 절은 무조건 조회 쿼리의 마지막에 작성한다.
- ORDER BY 키워드 다음에서 정렬의 기준이 되는 컬럼명을 작성한다.
- 정렬의 기준이 되는 컬럼명을 작성한 후에는 해당 컬럼을 기준으로 오름차순으로 정렬할지, 내림차순으로 정렬할지 작성한다.
- 오름차순으로 정렬하려면 ASC를, 내림차순으로 정렬하려면 DESC 키워드를 작성한다. 오름차순, 내림차순을 작성하지 않으면 오름차순으로 정렬된다.

다음은 EMP 테이블에서 20번 부서번호에 소속된 사원들의 사번, 사원명, 부서번호를 조회하되, 사원명 기준 내림차순으로 정렬한 것이다.

정렬기준이 문자라면 사전편찬 순으로 정렬된다.

```
SELECT EMPNO, ENAME, DEPTNO FROM EMP
WHERE DEPTNO = 20
ORDER BY ENAME DESC;
```

Database | 조회 데이터 정렬하기 - 2

다음은 EMP 테이블에 존재하는 사원들의 모든 정보를 조회하되, 직급 기준 오름차순 정렬한 쿼리이다.

```
SELECT * FROM EMP
ORDER BY JOB;
```

조회해보면 직급 기준 오름차순 정렬된 결과를 볼 수 있다. 이 상태에서 추가적인 정렬이 가능할까? 물론 가능하다.

아래 쿼리문은 직급 기준 오름차순 정렬 후, 정렬된 데이터에서 추가적으로 이름 기준 내림차순 정렬을 한 쿼리이다.

```
SELECT * FROM EMP
ORDER BY JOB ASC, ENAME DESC;
```

위의 쿼리에서 보듯, 정렬 기준이 되는 컬럼을 쉼표를 통해 나열할 수 있으며, 정렬 기준이 되는 각각의 컬럼에 별도로 오름차순, 내림차순을 정해줘야 한다.

다음은 EMP테이블에서 직급이 사원인 직원을 제외한 데이터를 제외 후, 사원명, 직급, 급여, 부서번호를 조회하되, 먼저 부서번호 기준 내림차순 정렬 후, 부서번호가 같다면 급여가 높은 직원부터 조회하는 쿼리이다.

```
SELECT ENAME, JOB, SAL, DEPTNO FORM EMP
WHERE JOB != '사원'
ORDER BY DEPTNO DESC, SAL DESC;
```

Database | Like 연산자와 와일드카드

EMP 테이블에서 사원명에 '이 ' 가 포함된 사원의 모든 컬럼을 조회하려면 어떻게 해야할까?

```
SELECT * FROM EMP
WHERE ENAME = '이';
```

만약, 위와 같은 쿼리를 생각했다면 잘못된 생각이다.

위의 쿼리는 사원명에 '이'가 포함된 사원을 조회하는 것이 아니라, 사원명이 '이'와 일치하는 사원을 조회하기에 조회 결과 어떤 데이터도 조회되지 않는다.

이렇게 특정 정보가 포함된 정보를 조회할 때는 LIKE연산자와 와일드카드를 사용한다. 먼저 위 요구사항을 충족하는 쿼리문은 다음과 같다.

```
SELECT * FROM EMP
WHERE ENAME LIKE '%이%';
```

위 쿼리에서 작성한 '%'를 와일드카드라 부른다. 데이터베이스에서 와일드 카드는 '%'와 '_' 두 개가 존재한다.

- '%'는 0글자 이상의 어떤 글자를 의미한다. '_'는 어떤 한 글자를 의미한다.
- '%'와 '_' 와일드카드는 모두 어떤 글자를 의미하지만, '_'는 한 글자를 의미하고, '%'는 글자수 제한이 없다는 점이 차이점이다.
- '김%'이라 작성하면, 김으로 시작하는 모든 글자를 의미한다.(김, 김가, 김1, 김가나, 김가나다....)
- '김_'이라 작성하면, 김으로 시작하는 두 글자를 찾는다. '_가__'로 작성하면 두번째 글자가 '가'인 4글자를 찾는다.
- '_가%'이라 작성하면 두번째 글자가 '가'인 모든 글자를 찾는다. '%나__' 로 작성하면 마지막에서 세번째 글자가 '나'인 모든 글자를 찾는다.

Database | DISTINCT 키워드 - 1

DISTINCT 키워드를 이용해 데이터를 조회하면 조회 데이터 중 중복 데이터는 제거한 조회 결과를 볼 수 있다.

```
SELECT DEPTNO FROM EMP;
```

위 쿼리의 결과 EMP테이블의 데이터수만큼 부서번호가 조회되는 것을 확인할 수 있다. 하지만 부서번호는 10, 20, 30번 밖에 없기 때문에 똑같은 조회 데이터가 14번 조회되는 것은 뭔가 바람직하지 않는 듯 하다.

만약, EMP테이블에서 모든 직원이 소속된 부서번호의 종류를 알고 싶다면 어떻게 해야 할까. 이렇게 쿼리 결과 중복데이터는 제거 후 결과를 보고 싶다면 DISTINCT 연산자를 컬럼명 앞에 추가한다.

```
SELECT DISTINCT DEPTNO FROM EMP; -> EMP테이블에서 부서번호의 종류를 확인할 수 있는 쿼리
SELECT DISTINCT JOB FROM EMP; -> EMP테이블에서 직급의 종류를 확인할 수 있는 쿼리
```

위의 쿼리를 실행하여 확인해보면 부서번호는 3개의 데이터가 조회되고, 직급은 6개가 조회되는 것을 확인 할 수 있다.

다음 쿼리를 보자. 아래의 쿼리문은 DEPTNO 컬럼만 중복을 제거하라는 말일까? 아니면 DEPTNO, JOB 두 컬럼에서 모두 중복을 제거할까?

```
SELECT DISTINCT DEPTNO, JOB FROM EMP;
```

조회 쿼리문에서 DISTINCT 키워드를 한 번만 사용하면 작성한 모든 컬럼에서 중복을 제거하여 조회한다.

그렇기 때문에 각각의 컬럼 앞에 DISTINCT 키워드를 붙이면 오류가 나며, 쿼리 작성 시 나열한 컬럼 중 특정 컬럼만 중복을 제거하는 것은 불가능하다.

Database | DISTINCT 키워드 - 2

앞서 이야기한 다음 쿼리문을 실행하면 11개의 데이터가 조회되는 것을 확인할 수 있다.

```
SELECT DISTINCT DEPTNO, JOB FROM EMP;
```

말했듯이, DISTINCT 키워드를 사용하면 DISTINCT 다음에 작성한 컬럼에서만 중복을 제거하지 않고, 나열한 모든 컬럼을 기준으로 중복을 제거한다.

위의 코드에서는 부서번호와 직급을 조회 컬럼으로 작성했기에, 두 컬럼 모두에서 중복을 제거한다.

여러 컬럼에서의 중복 판단은 모든 컬럼의 값이 같을 경우에만 중복이라 판단하기에 해석에 주의를 요한다.

DISTINCT 키워드의 해석은 SQLD 시험에 의외로 자주 출제되는 내용이다.

Database | 최종 점검

1. EMP 테이블에서 커미션이 NULL이 아닌 사원 중, 급여가 350에서 650 사이인 사원들의 사원명, 급여, 커미션을 조회하되, 쿼리문 작성 시 BETWEEN 연산자를 사용하여 작성하시오.
2. 직급이 과장, 차장, 부장인 직원의 사번, 사원명, 직급을 조회하되, 직급 기준 오름차순으로 정렬하고, 쿼리 작성 시 IN 연산자를 사용하시오.
3. 부서번호가 10, 20인 부서에 소속된 직원 중, 이름에 '이'가 포함된 직원의 사번, 사원명, 부서번호, 급여를 조회하되, 부서번호 기준 내림차순으로 정렬 후, 부서번호가 같다면 급여가 낮은 순부터 조회하는 쿼리문을 작성하시오.
4. 이름이 '기 ' 로 끝나는 직원 중, 커미션은 NULL이고 급여는 400에서 800 사이인 직원의 모든 컬럼 정보를 조회하시오.
5. 다음과 같은 데이터가 있는 CLASS_INFO 테이블에서

   SELECT DISTINCT CLASS_NAME, TEACHAR FROM CLASS_INFO WHERE CLASS_NAME = '자바반'; 으로 작성한 쿼리 실행 결과 조회되는 튜플(Tuple)의 갯수는?

학급(CLASS_NAME)	강사(TEACHER)	강의실(CLASS_ROOM)
회계반	김회계	101호
회계반	김회계	102호
회계반	박회계	103호
자바반	윤자바	201호
자바반	홍자바	202호
자바반	최자바	203호