

기본 SQL 작성하기_DML 활용



데이터조작어(DML:DATA MANIPULATION LANGUAGE)

```
INSERT INTO 테이블_이름 (컬럼1, 컬럼2, 컬럼3, ...)  
VALUES (값1, 값 2, 값 3, ...);
```

```
INSERT INTO nation(code, name, number) VALUES (1,'korea', 50) ;
```

nation 테이블 code, name, number 컬럼에 각각 1, korea, 50의 값을 입력한다

- 컬럼 리스트의 개수와 값 리스트의 개수가 동일
- 컬럼리스트를 생략하면 모든 컬럼을 의미하므로 컬럼 개수 만큼 값을 입력해야 한다. 또한 값들은 컬럼 순서와 동일하게 입력해야 한다.

예) INSERT INTO nation VALUES(1,'korea', 50, 10, 21, 2);

code	name	number	game	now_rank	last_rank
1	KOREA	50	10	21	2

데이터조작어(DML:DATA MANIPULATION LANGUAGE)

[예제] company 데이터베이스를 정의하고 employee, department 테이블을 정의하고 테이블에 예제 데이터를 입력해 보자

employee테이블

컬럼명	데이터타입	설명
emp_no	int	직원 번호(AUTO_INCREMENT, 초기값 1001) 기본키(PK)
name	varchar(20)	이름(NOT NULL)
department	int	소속 부서 번호
position	varchar(20)	직급
gender	Enum("M", "F")	성별
hire_date	Date	고용날짜
salary	int	급여

department테이블

컬럼명	데이터타입	설명
dept_no	int	부서 번호(기본키)
dept_name	varchar(15)	부서 이름(NOT NULL)
location	varchar(15)	부서 위치(NOT NULL)

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

department

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

INSERT INTO 문

복수 개의 데이터를 입력

```
INSERT INTO employee VALUES
```

```
(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);
```

```
INSERT INTO department VALUES
```

```
(1, '영업부', '대구'),  
(2, '인사부', '서울'),  
(3, '총무부', '대구'),  
(4, '기획부', '서울');
```

SELECT 문

```
SELECT 컬럼1, 컬럼2, ...  
FROM 테이블_이름;
```

```
SELECT name  
FROM employee;
```

employee 테이블에서
이름 검색

결과

name
구창민
김민서
이은영
한성일

모든 컬럼을 검색하고 싶다면 * 를 사용한다.

```
SELECT *  
FROM employee;
```

employee 테이블에서 모든 컬럼 검색
모든 테이블 내용이 출력

SELECT 문

```
SELECT 컬럼1, 컬럼2, ...  
FROM 테이블_이름  
WHERE 조건식;
```

조건에 맞는 레코드 중
컬럼 출력

```
SELECT name ← name 컬럼만 출력  
FROM employee ← employee테이블에서  
WHERE position = '과장' ← position이 과장인 직원 검색
```

and, or, not 을 써서 조건식을 만들 수 있다.

```
SELECT name  
FROM employee  
WHERE position = '과장' OR '부장' ← position이 과장이거나 부장인  
직원의 이름
```

SELECT 문

```
SELECT DISTINCT 컬럼1, 컬럼2, ...  
FROM 테이블_이름;
```

```
SELECT DISTINCT location  
FROM department;
```

부서테이블에서 location 출력
중복제거하고 출력

location
대구
서울
대구
서울



location
대구
서울

SELECT 문

SELECT 컬럼1, 컬럼2, ...

FROM 테이블_이름

ORDER BY 컬럼1, 컬럼2,

컬럼1, 컬럼2 순으로 정렬하여 출력

ASC(ASCENDING) : 오름차순

DESC(DESCENDING) : 내림차순

ASC나 DESC 생략하는 경우 ASC로 정렬

SELECT *

FROM *employee*

월급이 많은 순서로 직원의 모든 정보 출력

ORDER BY salary DESC;

SELECT 문(다양한 함수)

최대값 max(), 최소값 min()

SELECT MAX(컬럼명) FROM 테이블_이름

select max(salary) from employee;

max(salary)

5500000

개수 검색 count()

SELECT COUNT(컬럼명) FROM 테이블_이름 WHERE 조건식;

select count(empno) from employee;

count(emp_no)

4

평균값 avg()

SELECT AVG(컬럼명) FROM 테이블_이름 WHERE 조건식;

select avg(salary) from employee;

avg(salary)

4500000.0000

SELECT 문(LIKE, LIMIT)

특정 패턴의 값을 찾을 쓰는 연산자 LIKE

SELECT 컬럼1, 컬럼2, ... **FROM** 테이블_이름
WHERE 컬럼명 **LIKE** 패턴 ;

select name, department, position from employee where name like '김%';

와일드 카드 문자

% : 여러 글자 _ : 한 개의 글자

name	department	position
김민서	1	사원

결과의 개수 제한 LIMIT

SELECT 컬럼1, 컬럼2, .. **FROM** 테이블_이름
ORDER BY 컬럼명 **LIMIT** 개수;

*SELECT name, hire_date FROM employee
ORDER BY hire_date LIMIT 2;*

name	hire_date
이은영	1990-09-01
한성일	1993-04-01

UPDATE 문

테이블에 저장된 값을 수정

```
UPDATE 테이블_이름  
SET 컬럼1 = 값1, 컬럼2 = 값2, ...  
WHERE 조건식;
```

```
UPDATE employee  
SET position = '대리'  
WHERE emp_no = 1002 ;
```

김민서 사원의 직급이
사원에서 대리로 변경

```
UPDATE department  
SET location = '인천'  
WHERE dept_name = '인사부';
```

인사부의 위치를
인천으로 변경

DELETE 문

테이블에 저장된 값을 삭제

```
DELETE FROM 테이블_이름  
WHERE 조건식;
```

```
DELETE FROM employee  
WHERE emp_no=1004;
```

emp_no가 1004인 직원 정보 삭제

```
DELETE FROM employee
```

모든 테이블 정보 삭제

주의 !! 테이블의 모든 내용이 삭제되므로 주의해서 사용.