

SQL

DDL

- **CREATE**
- **ALTER**
- **DROP**

DML

- **SELECT**
- **INSERT**
- **DELETE**
- **UPDATE**

DCL

- **GRANT**
- **REVOKE**
- **COMMIT**
- **ROLLBACK**

DDL

- Data **Definition** Language
데이터 정의어

데이터베이스나 테이블을
생성(**CREATE**)
변경(**ALTER**)
제거(**DROP**)

DML

- Data **Manipulation** Language
데이터 조작어

데이터를

검색(**SELECT**) 삽입(**INSERT**)

삭제(**DELETE**) 수정(**UPDATE**)

DCL

- Data **Control** Language
데이터 제어어

권한을

부여(**GRANT**)

처리를

적용(**COMMIT**)

COMMIT과 **ROLLBACK**은
TCL로 분류하기도 함!

제거(**REVOKE**)

취소(**ROLLBACK**)

기본 활용

- DB 활용 기본

```
Mysql> SHOW DATABASES;
```

```
Mysql> USE DB이름;
```

```
Mysql> SHOW TABLES;
```

```
SHOW TABLE STATUS;
```

```
Mysql> DESC TABLE이름;
```

DDL

- **CREATE**

```
Mysql> CREATE DATABASE DB이름;
```

```
Mysql> CREATE TABLE TABLE이름 (  
    열이름 데이터타입 제약조건,  
    열이름 데이터타입,  
    ...);
```

DDL

- 제약조건

데이터의 **무결성**을 위한 조건

키, null의 허용 여부 등

DDL

- **CREATE**

예시)

```
Mysql> CREATE TABLE testtbl (  
        col1 int(5) not null primary key,  
        col2 varchar(8),  
        ...);
```

#값이 null이면 안되고 중복되는 데이터x

DDL

- 현재 존재하는 DB 목록을 확인하세요.
- 'sqlDB'라는 DB를 만들고 사용하도록 선택하세요.
- 다음 정보에 맞게 테이블을 생성하고 테이블 정보를 출력하세요.

DDL

TABLE명 : userTbl

속성명	데이터타입	NN	PK
userID	CHAR(8)	O	O
name	VARCHAR(10)	O	
birthyear	INT	O	
addr	CHAR(2)	O	
mobile1	CHAR(3)		
mobile2	CHAR(8)		
height	SMALLINT		
mDate	DATE		

DDL

TABLE명 : prodTbl

속성명	데이터타입	NN	PK
num	INT	O	O
userID	CHAR(8)	O	
prodName	CHAR(6)	O	
groupName	CHAR(4)		
price	CHAR(5)		
count	SMALLINT	O	

DDL

- **ALTER**

```
Mysql> ALTER TABLE TABLE이름  
        RENAME 새TABLE이름;
```

```
Mysql> ALTER TABLE TABLE이름  
        MODIFY 열이름 새데이터타입;
```

DDL

- **sqlDB의 테이블 중 prodTbl의 이름을 buyTbl로 바꾸세요.**
- **buyTbl의 속성 중 price의 데이터 타입을 INT로 바꾸세요.**
- **변경을 마친 후 테이블 description을 출력하세요.**

DDL

- **ALTER**

```
Mysql> ALTER TABLE TABLE이름  
        MODIFY 열이름 데이터타입  
        PRIMARY KEY;
```

```
Mysql> ALTER TABLE TABLE이름  
        MODIFY 열이름 데이터타입  
        AUTO_INCREMENT;
```

DDL

- **ALTER**

```
Mysql> ALTER TABLE TABLE이름  
        MODIFY 열이름 데이터타입  
        NOT NULL;
```

```
Mysql> ALTER TABLE TABLE이름  
        CHANGE 열이름 새열이름  
        데이터타입;
```

DDL

- buyTbl의 속성 중 num에 auto_increment를 적용하세요.
- buyTbl의 속성 중 count를 amount로 변경하세요.
- buyTbl의 속성 중 price에 NULL값이 올 수 없도록 변경하세요.

DDL

- **ALTER**

```
Mysql> ALTER TABLE TABLE이름  
      ADD 제약조건;
```

```
Mysql> ALTER TABLE TABLE이름  
      ADD CONSTRAINT  
      FOREIGN KEY(열이름)  
      REFERENCES  
      참조테이블(열이름);
```

DDL

- buyTbl의 속성 중 userID를 외래키로 지정하고, userTbl의 userID를 참조하도록 제약조건을 추가하세요.
- 모두 끝낸 후, buyTbl의 description을 출력하고 다음 정보와 비교하세요.

DDL

TABLE명 : buyTbl

속성명	데이터타입	NN	KEY
num	INT	O	PRI
userID	CHAR(8)	O	MUL
prodName	CHAR(6)	O	
groupName	CHAR(4)		
price	INT	O	
amount	SMALLINT	O	

AUTO_INCREMENT(num)

DDL

- **DROP**

```
Mysql> DROP DATABASE DB이름;
```

```
Mysql> DROP TABLE TABLE이름;
```

DDL

- testDB 데이터베이스를 생성하고, 속성으로 id char(16)를 가지는 testTbl을 생성한 후,

DROP 명령어로 테이블, DB 순서로 삭제하세요.

DML

- **SELECT – 기본 형식**

```
Mysql> SELECT 열이름  
        FROM 테이블이름;
```

```
Mysql> SELECT * FROM DB.TABLE;
```

```
Mysql> SELECT 열이름 AS 별칭  
        FROM 테이블이름;
```

DML

- **employees DB의 titles 테이블 내용을 검색하세요.**
- **employees DB의 employees 테이블에서 first_name 열만 검색하세요.**

DML

- employees DB의 employees 테이블에서 gender, first_name, emp_no 열을 순서대로 나오게 검색하세요.

DML

- employees DB의 employees 테이블에서

first_name은 '이름'으로,
gender는 '성별'로,
hire_date는 '입사일'로

별칭을 사용하여 검색하세요.

DML

- INSERT – 기본 형식

```
Mysql> INSERT INTO 테이블이름  
      (열이름, 열이름, ...)  
      VALUES (데이터, 데이터, ...);
```

```
Mysql> INSERT INTO 테이블이름  
      VALUES (데이터, 데이터, ...);
```

DML

- INSERT를 이용하여 다음 데이터를 해당하는 테이블에 입력하세요.
(auto_increment 적용된 열은 null입력)
- 입력이 완료되면 SELECT를 이용하여 전체 데이터를 확인하세요.
- 저장한 sql문을 이용해 Linux에도 동일한 DB를 생성하세요.

DML

TABLE명 : userTbl

I D	이름	생년	주소	전화1	전화2	키	가입일
BBK	바비킴	1973	서울	010	0000000	176	2013-05-05
EJW	은지원	1972	경북	011	8888888	174	2014-03-03
JKW	조관우	1965	경기	018	9999999	172	2010-10-10
JYP	조용필	1950	경기	011	4444444	166	2009-04-04
KBS	김범수	1979	경남	011	2222222	173	2012-04-04
KKH	김경호	1971	전남	019	3333333	177	2007-07-07
LJB	임재범	1963	서울	016	6666666	182	2009-09-09
LSG	이승기	1987	서울	011	1111111	182	2008-08-08
SSK	성시경	1979	서울			186	2013-12-12
YJS	윤종신	1969	경남			170	2005-05-05

DML

TABLE명 : buyTbl

번호	ID	상품명	그룹명	가격	수량
1	KBS	운동화		30	2
2	KBS	노트북	전자	1000	1
3	JYP	모니터	전자	200	1
4	BBK	모니터	전자	200	5
5	KBS	청바지	의류	50	3
6	BBK	메모리	전자	80	10
7	SSK	책	서적	15	5
8	EJW	책	서적	15	2
9	EJW	청바지	의류	50	1
10	BBK	운동화		30	2
11	EJW	책	서적	15	1
12	BBK	운동화		30	2

DML

- **SELECT – WHERE**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        WHERE 조건;
```

DML

- userTbl에서 이름이 '김경호'인 회원을 검색하세요.
- userTbl에서
1970년 이후에 출생하고
키가 182 이상인 사람의
아이디와 이름을 검색하세요.

DML

- userTbl에서
1970년 이후에 출생했거나
키가 182 이상인 사람의
아이디와 이름을 검색하세요.
- userTbl에서 키가 180이상,
183이하인 사람을 검색하세요.

DML

- **SELECT – BETWEEN AND**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        WHERE 열이름  
        BETWEEN A AND B;
```

DML

- userTbl에서 주소가
'경남'이거나 '전남'이거나
'경북'인 사람을 검색하세요.

DML

- **SELECT - IN**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        WHERE 열이름  
        IN ('A', 'B', 'C');
```

DML

- **SELECT - 와일드카드**

```
Mysql> SELECT name, height  
        FROM userTbl  
        WHERE name LIKE '김%';
```

```
Mysql> SELECT name, height  
        FROM userTbl  
        WHERE name LIKE '_종신';
```

DML

- 와일드카드

% : 글자 길이 무관

_ : 글자 하나

- 예시) _용% : 조용필, 사용한 사람
 이용해주셔서 감사합니다

DML

- userTbl에서 '김경호'보다 키가 크거나 같은 사람을 검색하세요.

DML

- userTbl에서 '김경호'보다 키가 크거나 같은 사람을 검색하세요.
- `select name, height from usertbl where name='김경호';` : 177
- `select name, height from usertbl where height >= 177;`

DML

- **SELECT - SubQuery**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        WHERE 열이름 연산자 (  
            SELECT 열이름  
            FROM 테이블이름  
            WHERE 조건);
```


DML

- userTbl에서 '김경호'보다 키가 크거나 같은 사람을 검색하세요.

SubQuery, 하위쿼리
쿼리문 안의 쿼리문

DML

- userTbl에서 '김경호'보다 키가 크거나 같은 사람을 검색하세요.

```
Mysql> select name, height  
       from usertbl  
       where height >= (  
           select height  
           from usertbl  
           where name='김경호');
```

DML

- userTbl에서 주소가 '경남'인 사람의 키보다 크거나 같은 사람을 검색하세요.

DML

- userTbl에서 주소가 '경남'인 사람의 키보다 크거나 같은 사람을 검색하세요.

```
Mysql> select name, height  
       from usertbl  
       where height >= (  
           select height  
           from usertbl  
           where addr='경남');
```

DML

- 오류 : 서브쿼리가 2개 이상의 값을 반환했습니다.

```
Mysql> select height  
from usertbl  
where addr='경남';
```

	height
▶	173
	170

DML

- 연산자 **ANY** (SubQuery)
하위쿼리를 만족하는 값들 중 어느 하나만 조건을 만족해도 될 때
(**OR**과 유사)
- 연산자 **ALL** (SubQuery)
하위쿼리를 만족하는 값들 모두에 대해서 조건이 만족해야 할 때
(**AND**와 유사)

DML

- userTbl에서 주소가 '경남'인 사람의 키보다 크거나 같은 사람을 검색하세요.

```
Mysql> select name, height  
        from usertbl  
        where height >= ANY/ALL(  
            select height  
            from usertbl  
            where addr='경남');
```

DML

- ANY

	name	height
▶	바비킴	176
	은지원	174
	조관우	172
	김범수	173
	김경호	177
	임재범	182
	이승기	182
	성시경	186
	윤종신	170

- ALL

	name	height
▶	바비킴	176
	은지원	174
	김범수	173
	김경호	177
	임재범	182
	이승기	182
	성시경	186

DML

- **SELECT – ANY와 IN**

```
Mysql> select name, height  
        from usertbl  
        where height = ANY(  
                        select height  
                        from usertbl  
                        where addr='경남');
```

DML

- **SELECT – ANY와 IN**

```
Mysql> select name, height  
        from usertbl  
        where height IN (  
                select height  
                from usertbl  
                where addr='경남');
```

DML

- **SELECT – ORDER BY**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        [WHERE 조건]  
        ORDER BY 열이름 ASC/DESC;
```

DML

- ORDER BY
기본적으로 **오름차순**(ASC)

정렬의 기준이 되는 열이 꼭
검색 대상일 필요는 없다.

```
Mysql> SELECT userID  
        FROM userTbl  
        ORDER BY height;
```

DML

- **ORDER BY**

키가 큰 순서대로 정렬하되,
키가 같으면 이름 순으로 정렬하기

```
Mysql> SELECT name, height  
        FROM userTbl  
        ORDER BY height DESC,  
                 name ASC;
```

DML

- **userTbl에서 회원들의 거주 지역이 몇 군데인지 세어보세요.**

DML

- userTbl에서 회원들의 거주 지역이 몇 군데인지 세어보세요.

```
Mysql> select * from usertbl;
```

```
Mysql> select addr from usertbl;
```

```
Mysql> select addr from usertbl  
order by addr;
```

DML

- **SELECT – DISTINCT**

```
Mysql> SELECT DISTINCT 열이름  
        FROM 테이블이름  
        [WHERE 조건]  
        [ORDER BY 열이름];
```


DML

- **SELECT – LIMIT**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        [WHERE 조건]  
        [ORDER BY 열이름]  
        LIMIT 시작위치, 개수;  
        = LIMIT 개수 OFFSET 위치
```

DML

- buyTbl에서 각 회원 당 총 구매량을 구하세요.
- buyTbl에서 각 회원 당 총 구매액을 구하세요.

DML

- **SELECT – GROUP BY**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        [WHERE 조건]  
        GROUP BY 열이름  
        [ORDER BY 열이름];
```

DML

- **SELECT – SUM()함수**

```
Mysql> SELECT SUM(열이름)  
        FROM 테이블이름  
        [WHERE 조건]  
        GROUP BY 열이름  
        [ORDER BY 열이름];
```

DML

- **SELECT – SUM()함수**

```
Mysql> SELECT userID AS '사용자',  
          SUM(amount) AS  
          '총 구매 개수'  
FROM buyTbl  
GROUP BY userID;
```

DML

- **SELECT – SUM() 함수**

```
Mysql> SELECT userID AS '사용자',  
          SUM(amount*price) AS  
          '총 구매액'  
FROM buyTbl  
GROUP BY userID;
```

DML

• 집계 함수

함수명	설명
SUM()	합계
AVG()	평균
MIN()	최소값
MAX()	최대값
COUNT()	행의 개수
COUNT(DISTINCT)	행의 개수(중복 1개로 처리)
STDEV()	표준편차
VAR_SAMP()	분산

DML

- 전체 구매자가 구매한 **물품 개수의 평균**을 구하세요.
- 각 사용자가 **한 번 구매 시 평균적으**로 물건을 몇 개 구매했는지 구하세요.
- 휴대폰 번호를 입력한 **회원 수**를 구하세요.
- **가장 큰 키와 가장 작은 키의 회원 이**름을 출력하세요.

DML

- **SELECT – GROUP BY**

```
Mysql> SELECT AVG(amount)  
        FROM buyTbl;
```

```
Mysql> SELECT userID, AVG(amount)  
        FROM buyTbl  
        GROUP BY userID;
```

```
Mysql> SELECT COUNT(mobile1)  
        FROM userTbl;
```

DML

- **SELECT – GROUP BY**

```
Mysql> SELECT name, height
        FROM userTbl
        WHERE height = (
            SELECT MAX(height)
            FROM userTbl) OR
            height = (
            SELECT MIN(height)
            FROM userTbl);
```

DML

- 총 구매액이 1000 이상인 사용자를 검색하세요.

DML

- **SELECT – GROUP BY HAVING**

```
Mysql> SELECT 열이름  
        FROM 테이블이름  
        [WHERE 조건]  
        GROUP BY 열이름  
        HAVING 조건  
        [ORDER BY 열이름];
```

DML

- **SELECT – GROUP BY HAVING**

```
Mysql> SELECT userID,  
          SUM(amount*price)  
          AS '총구매액'  
FROM buyTbl  
GROUP BY userID  
HAVING 총구매액 >= 1000  
[ORDER BY 총구매액];
```

DML

- buyTbl에서 상품 분류별에 따라
각 분류별 판매액의 합계(소합계)와
총 판매액을 확인하세요.

DML

- **SELECT – GROUP BY WITH ROLLUP**

```
Mysql> SELECT groupName,  
          SUM(amount*price)  
FROM buyTbl  
GROUP BY groupName
```

DML

- **SELECT – GROUP BY WITH ROLLUP**

```
Mysql> SELECT num, groupName,  
          SUM(amount*price)  
FROM buyTbl  
GROUP BY  
        groupName, num  
WITH ROLLUP;
```


DML

- **INSERT + SELECT**

```
Mysql> INSERT INTO 테이블이름  
        SELECT 열이름  
        FROM 테이블이름;
```

열개수와 데이터타입 등이 일치해야 함

DML

- 다음과 같은 테이블을 sqlDB내에 만들고 employees DB의 employees 테이블에서 emp_no, first_name, last_name의 데이터를 가져오세요.

속성	데이터타입
id	int
fname	varchar(50)
lname	varchar(50)

TABLE명 : testTbl

DDL

DML

- **CREATE + SELECT**

```
Mysql> CREATE TABLE 테이블이름 (  
        SELECT 복사할 열  
        FROM 복사할 테이블);
```

DDL

DML

- buyTbl을 buyTbl2로 복사하세요.
 - buyTbl에서 userID만 복사해서 buyTbl3을 만드세요.
- ※ 제약 조건은 복사되지 않음.

DML

- **UPDATE SET**

```
Mysql> UPDATE 테이블이름  
        SET 열1=값1,  
            열2=값2, ...  
        WHERE 조건;
```

DML

- 만들었던 testTbl에서 fname이 'Kyichi'인 사람의 전화번호를 '없음'으로 변경하세요.
- ※ WHERE 조건을 빼면
전체 테이블의 내용이 바뀌므로 주의.

DML

- **DELETE**

```
Mysql> DELETE FROM 테이블 이름  
WHERE 조건;
```

DML

- 만들었던 testTbl에서 fname이 'Aamer'인 사람의 데이터를 삭제하세요.

※ WHERE 조건을 빼면
전체 데이터가 삭제되므로 주의.

DML

- `employees.employees` 테이블을 `bigTbl1`, `bigTbl2`, `bigTbl3` 으로 복사하세요.
- 다음 3가지 명령을 각각 사용하여 각 테이블을 지워보세요.

DDL

DML

- **DELETE / TRUNCATE / DROP**

```
Mysql> DELETE FROM 테이블이름;
```

```
Mysql> TRUNCATE TABLE 테이블이름;
```

```
Mysql> DROP TABLE 테이블이름;
```

DDL

DML

- **트랜잭션(TRANSACTION)**

**DBMS에서
한번에 처리되는 연산의 단위
작업의 단위**

- **DML은 트랜잭션을 발생시킴**
- **DDL은 트랜잭션을 발생시키지 않음**

DDL

DML

- **DELETE는 DML**
- **TRUNCATE와 DROP은 DDL**
- **DELETE는 트랜잭션 로그를 기록하기 때문에 속도가 느림**
- **확실히 불필요한 데이터의 경우 DDL을 이용하는게 유리**

DCL

- **GRANT / REVOKE**

```
Mysql> GRANT 권한  
        ON DB.TABLE  
        TO 사용자;
```

```
Mysql> SHOW GRANTS FOR 사용자;
```

```
Mysql> REVOKE 권한  
        ON DB.TABLE  
        FROM 사용자;
```

DCL

- Linux에 만들었던
director, ceo, staff 계정의 권한을
REVOKE 명령어로 취소해보세요.

DCL

- **COMMIT / ROLLBACK**

```
Mysql> SELECT @@AUTOCOMMIT;
```

```
Mysql> SET AUTOCOMMIT=0;
```

DCL

- **AUTOCOMMIT을 끈 후
연습용 테이블에서
데이터를 삭제, 추가, 변경하고**

**ROLLBACK을 이용해
트랜잭션을 되돌려보세요.**