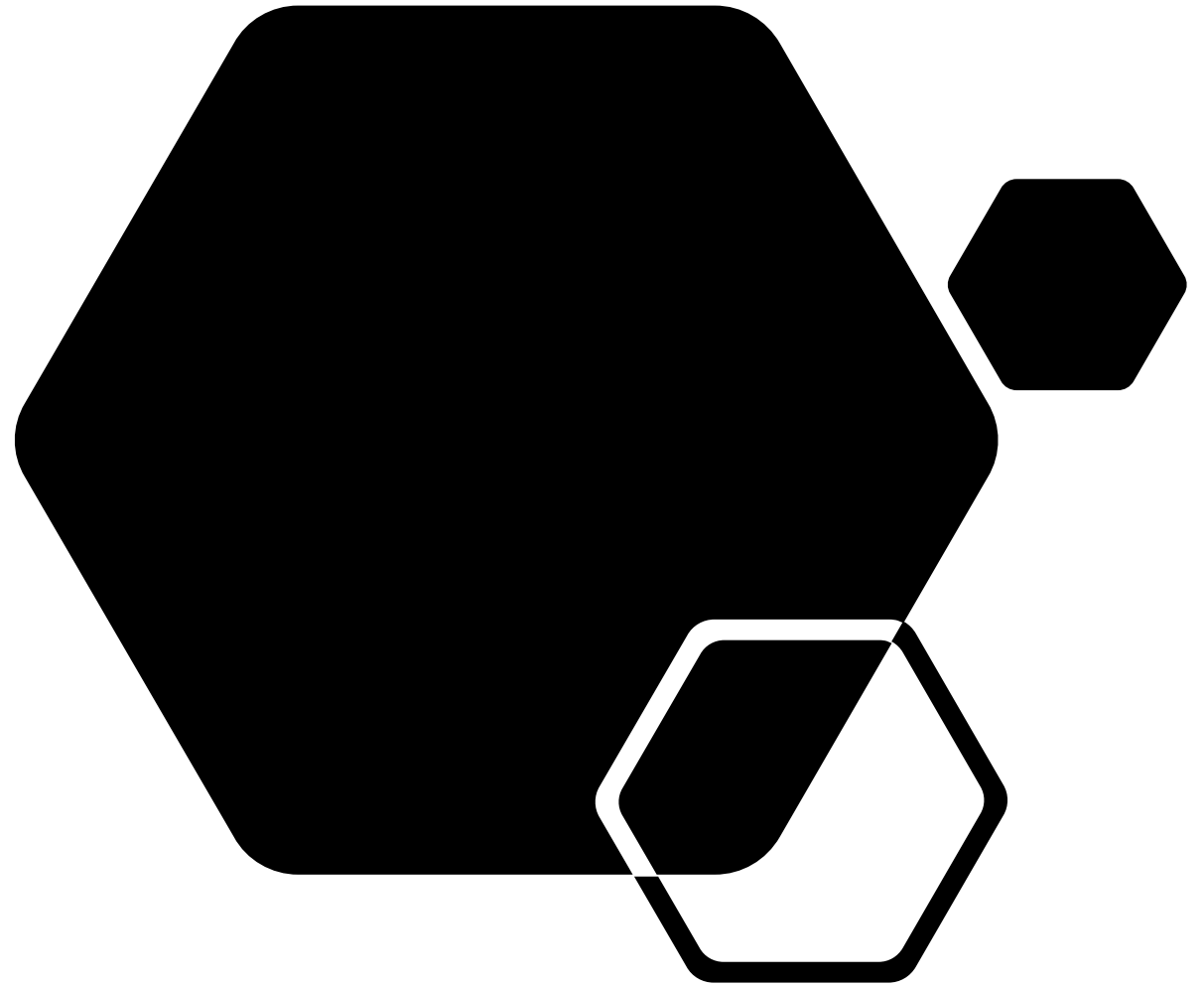


SQL Basic



Login

```
mysql -u root -p
```

```
mysql -h ip -u 계정 -p
```

Show Users

```
SELECT User, Host FROM mysql.user;
```

Create User

```
CREATE USER 'someuser'@'localhost' IDENTIFIED BY 'somepassword';
```

Delete User

```
DROP USER 'someuser'@'localhost';
```

Grant All Priveleges On All Databases

```
GRANT ALL PRIVILEGES ON * . * TO 'someuser'@'localhost';  
FLUSH PRIVILEGES;
```

| | |
|-----|--|
| 개발자 | DELETE, INSERT, SELECT, UPDATE |
| 설계자 | ALTER, CREATE, DROP, INDEX, INSERT, SELECT, UPDATE, DELETE |
| DBA | ALL |

Show Grants

```
SHOW GRANTS FOR 'someuser'@'localhost';
```

Remove Grants

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM  
'someuser'@'localhost';
```


Show Databases

```
SHOW DATABASES;
```

Create Database

```
CREATE DATABASE dbname;
```

Delete Database

```
DROP DATABASE dbname;
```

Select Database

```
USE dbname;
```

Create Table

```
CREATE TABLE users(  
    id INT AUTO_INCREMENT,  
    first_name VARCHAR(100),  
    last_name VARCHAR(100),  
    email VARCHAR(50),  
    password VARCHAR(20),  
    location VARCHAR(100),  
    dept VARCHAR(100),  
    is_admin TINYINT(1),  
    register_date DATETIME,  
    PRIMARY KEY(id)  
);
```

Add New Column

```
ALTER TABLE users ADD age VARCHAR(3);
```

Modify Column

```
ALTER TABLE users MODIFY COLUMN age INT(3);
```

Delete / Drop Table

```
DROP TABLE tablename;
```


Show Tables

```
SHOW TABLES;
```

Insert Row / Record

```
INSERT INTO users (first_name, last_name, email, password, location,  
dept, is_admin, register_date) values ('Brad', 'Traversy',  
'brad@gmail.com', '123456','Massachusetts', 'development', 1, now());
```

Insert Multiple Rows

```
INSERT INTO users (first_name, last_name, email, password, location,  
dept, is_admin, register_date) values ('Fred', 'Smith',  
'fred@gmail.com', '123456', 'New York', 'design', 0, now()), ('Sara',  
'Watson', 'sara@gmail.com', '123456', 'New York', 'design', 0,  
now()), ('Will', 'Jackson', 'will@yahoo.com', '123456', 'Rhode Island',  
'development', 1, now()), ('Paula', 'Johnson', 'paula@yahoo.com',  
'123456', 'Massachusetts', 'sales', 0, now()), ('Tom', 'Spears',  
'tom@yahoo.com', '123456', 'Massachusetts', 'sales', 0, now());
```

Update Row

```
UPDATE users SET email = 'freddy@gmail.com' WHERE id = 2;
```

Select



SELECT * FROM users;

SELECT first_name, last_name FROM users;

```
SELECT select_expr  
  [FROM table_references]  
  [WHERE where_condition]  
  [GROUP BY {col_name | expr | position}]  
  [HAVING where_condition]  
  [ORDER BY {col_name | expr | position}]
```

alias for columns

```
SELECT  
    [column_1 | expression] AS `alias name`  
FROM  
    table_name;
```

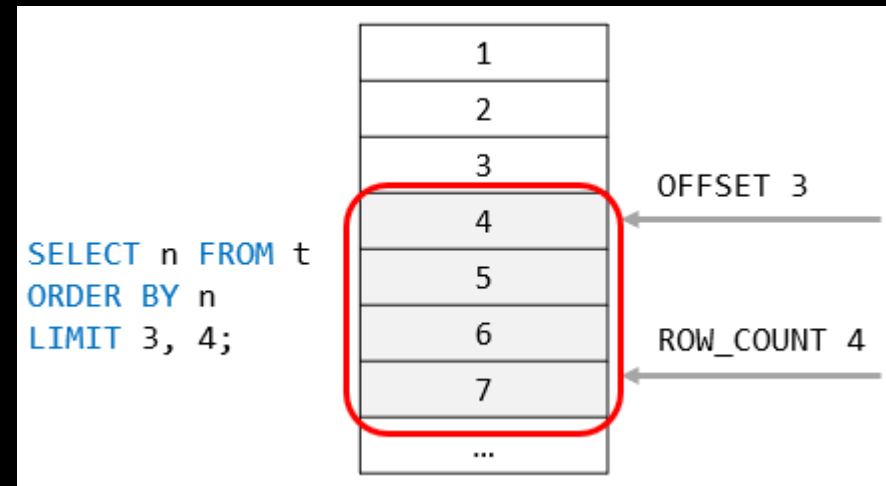
```
SELECT  
    CONCAT_WS(', ', lastName, firstname) AS `Full name`  
FROM  
    users;
```

```
SELECT * FROM employees e;
```

LIMIT

```
SELECT * FROM users LIMIT 5;
```

```
SELECT  
    select_list  
FROM  
    table_name  
LIMIT [offset,] row_count;
```



value IS NULL
value IS NOT NULL

IS NULL

SELECT

*

FROM

users

WHERE

email IS NULL

Where Clause

```
SELECT  
    select_list  
FROM  
    table_name  
WHERE  
    search_condition;
```

SELECT * FROM users WHERE location='Massachusetts';

SELECT * FROM users WHERE location='Massachusetts' AND dept='sales';

SELECT * FROM users WHERE is_admin = 1;

SELECT * FROM users WHERE is_admin > 0;



Order By (Sort)

```
SELECT  
    select_list  
FROM  
    table_name  
ORDER BY  
    column1 [ASC|DESC],  
    column2 [ASC|DESC],  
    ...;
```

```
SELECT * FROM users ORDER BY last_name ASC;  
SELECT * FROM users ORDER BY last_name DESC;
```



Concatenate Columns

```
SELECT CONCAT(first_name,' ', last_name) AS 'Name', dept  
FROM users;
```

Select Distinct Rows

```
SELECT DISTINCT  
  select_list  
FROM  
  table_name
```

SELECT DISTINCT location FROM users;



value BETWEEN low AND high;

Between (Select Range)

```
SELECT * FROM users WHERE age BETWEEN 20 AND 25;
```

Like (Searching)

```
SELECT * FROM users WHERE dept LIKE 'd%';
```

```
SELECT * FROM users WHERE dept LIKE 'dev%';
```

```
SELECT * FROM users WHERE dept LIKE '%t';
```

```
SELECT * FROM users WHERE dept LIKE '%e%';
```

expression LIKE pattern ESCAPE escape_character

- The percentage (%) wildcard matches any string of zero or more characters.
- The underscore (_) wildcard matches any single character.

Not Like

```
SELECT * FROM users WHERE dept NOT LIKE 'd%';
```

IN

```
value IN (value1, value2, value3,...)  
value NOT IN (value1, value2, value2)
```

```
SELECT * FROM users WHERE dept IN ('design', 'sales');
```


Create & Remove Index

```
CREATE INDEX idx_location On users(location);  
DROP INDEX idx_location ON users;
```

New Table With Foreign Key (Posts)

```
CREATE TABLE posts(  
  id INT AUTO_INCREMENT,  
  user_id INT,  
  title VARCHAR(100),  
  body TEXT,  
  publish_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY(id),  
  FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

Add Data to Posts Table

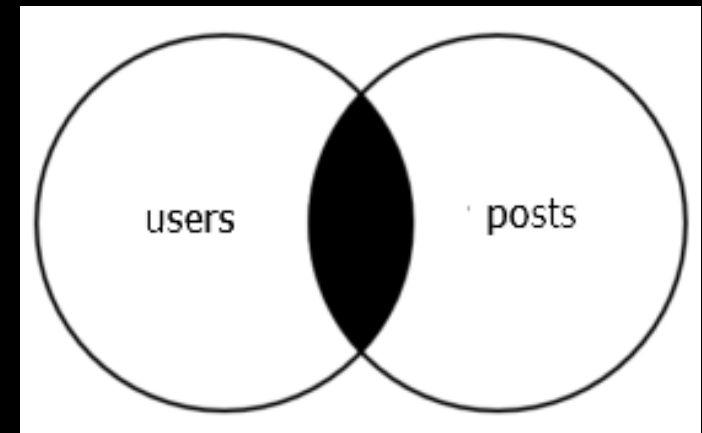
```
INSERT INTO posts(user_id, title, body) VALUES (1, 'Post One',  
'This is post one'),(3, 'Post Two', 'This is post two'),(1, 'Post  
Three', 'This is post three'),(2, 'Post Four', 'This is post four'),(5,  
'Post Five', 'This is post five'),(4, 'Post Six', 'This is post six'),(2,  
'Post Seven', 'This is post seven'),(1, 'Post Eight', 'This is post  
eight'),(3, 'Post Nine', 'This is post none'),(4, 'Post Ten', 'This is  
post ten');
```

INNER JOIN

```
SELECT
  u.first_name,
  u.last_name,
  p.title,
  p.publish_date
FROM users u
  INNER JOIN posts p
    ON u.id = p.user_id
ORDER BY p.title;
```

--첫번째 테이블
--두번째 테이블
--조인 조건

```
SELECT
  select_list
FROM t1
  INNER JOIN t2 ON join_condition1
  INNER JOIN t3 ON join_condition2
...;
```



New Table With 2 Foreign Keys

```
CREATE TABLE comments(  
    id INT AUTO_INCREMENT,  
    post_id INT,  
    user_id INT,  
    body TEXT,  
    publish_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY(id),  
    FOREIGN KEY(user_id) references users(id),  
    FOREIGN KEY(post_id) references posts(id)  
);
```

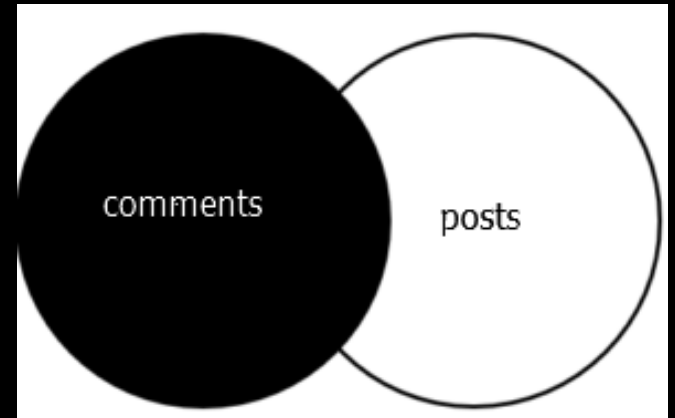
Add Data to Comments Table

```
INSERT INTO comments(post_id, user_id, body) VALUES (1, 3, 'This is comment one'),(2, 1, 'This is comment two'),(5, 3, 'This is comment three'),(2, 4, 'This is comment four'),(1, 2, 'This is comment five'),(3, 1, 'This is comment six'),(3, 2, 'This is comment six'),(5, 4, 'This is comment seven'),(2, 3, 'This is comment seven');
```

Left Join

```
SELECT  
c.body,  
p.title  
FROM comments c  
LEFT JOIN posts p ON p.id = c.post_id  
ORDER BY p.title;
```

```
SELECT  
    select_list  
FROM  
    t1  
LEFT JOIN t2 ON  
    join_condition;
```



Join Multiple Tables

```
SELECT
comments.body,
posts.title,
users.first_name,
users.last_name
FROM comments
INNER JOIN posts on posts.id = comments.post_id
INNER JOIN users on users.id = comments.user_id
ORDER BY posts.title;
```


Aggregate Functions

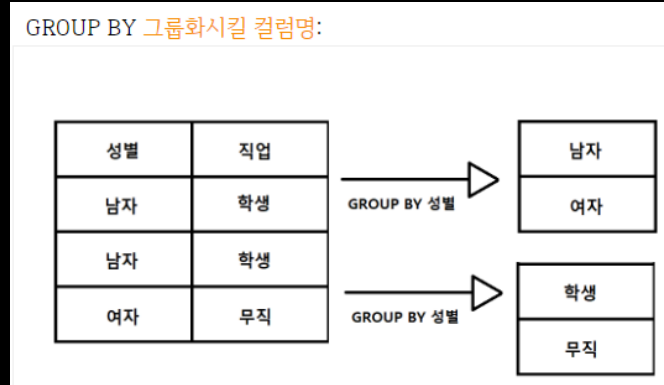
```
SELECT COUNT(id) FROM users;
```

```
SELECT MAX(age) FROM users;
```

```
SELECT MIN(age) FROM users;
```

```
SELECT SUM(age) FROM users;
```

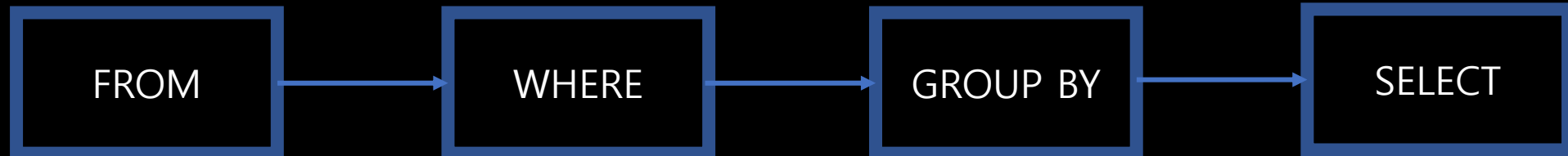
Group By



```
SELECT
    c1, c2,..., cn, aggregate_function(ci)
FROM
    table
WHERE
    where_conditions
GROUP BY c1 , c2,...,cn;
```

SELECT age FROM users GROUP BY age;

SELECT age FROM users WHERE age > 20 GROUP BY age;



Group By

```
SELECT  
    select_list  
FROM  
    table_name  
WHERE  
    search_condition  
GROUP BY  
    group_by_expression  
HAVING  
    group_condition;
```

SELECT age, avg(age) FROM users GROUP BY age HAVING avg(age) >=2;



SUB QUERY

```
SELECT * FROM users WHERE location=  
(SELECT location FROM users WHERE email = 'sara@gmail.com');
```