

# SELECT

```
SELECT *  
FROM table_name
```

Команда `SELECT *` повертає вибірку даних з таблиці

# SELECT для декількох колонок

```
SELECT column_name1, column_name2, ...  
FROM table_name
```

Команда SELECT \* повертає вибірку даних лише вказаних колонок таблиці

# ORDER BY

```
SELECT *  
FROM table_name  
ORDER BY column_name
```

Команда ORDER BY сортує результуючу вибірку за значенням певного стовпця

# WHERE

```
SELECT column_name, column_name  
FROM table_name  
WHERE column_name operator value;
```

WHERE використовується для отримання тільки тих записів, які задовольняють заданому критерію.

# COUNT()

```
SELECT COUNT(*)  
FROM table_name;
```

COUNT() повертає кількість записів, які відповідають певному критерію

# LIKE

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name LIKE pattern;
```

LIKE оператор використовується для пошуку певного шаблону в колонці.

LIKE "%"

```
SELECT * FROM Table_Name  
WHERE Column_Name LIKE 's%';
```

Знак “%” в шаблоні використовується для заміни послідовності літер.

```
LIKE "_"
```

```
SELECT * FROM Table_Name
```

```
WHERE Column_Name LIKE 's_';
```

Знак “\_” в шаблоні використовується для заміни послідовності літер.



# BETWEEN

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

BETWEEN вибирає значення в межах діапазону. Значення можуть бути цифри, текст або дати.

# INSERT INTO

```
INSERT INTO table_name  
VALUES (value1,value2,value3,...);
```

Оператор INSERT INTO використовується для вставки нових записів в таблиці.

# INSERT INTO для певних колонок

```
INSERT INTO (column_name1, column_name2,...);  
VALUES (value1,value2,value3,...);
```

Оператор INSERT INTO використовується для вставки даних лише для вказаних колонок.

# UPDATE

```
UPDATE table_name  
SET column1=value1,column2=value2,...  
WHERE some_column=some_value;
```

Оператор UPDATE використовується для оновлення записів в таблиці.

# DELETE

```
DELETE FROM table_name  
WHERE some_column=some_value;
```

Оператор DELETE використовується для видалення записів в таблиці.

# Агрегирование и групповые функции COUNT, SUM, AVG, MAX, MIN

```
SELECT FUNC(column_name)  
FROM table_name;
```

Дані функції використовуються для отримання загальної інформації про колонку в таблиці

# AS

```
SELECT column_name1 AS "name1",  
column_name2 AS "name2"  
FROM table_name AS new_table_name;
```

AS заміняє назву результуючої колонки або таблиці, для покращення сприйняття даних.

# SELECT для багатьох таблиць

```
SELECT *  
FROM table_name1, table_name2;
```

Повертає дані з багатьох таблиць



# SELECT FROM SELECT

```
SELECT column_name1, column_name2  
FROM (  
    SELECT *  
    FROM table_name  
)  
AS test_table;
```

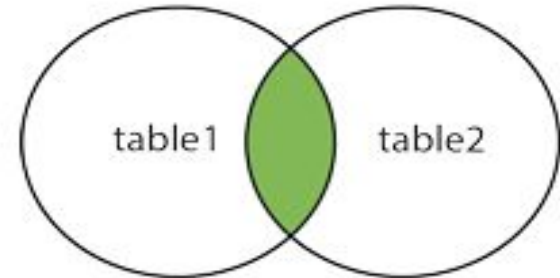
Зовнішній SELECT використовує результат внутрішнього SELECT як таблицю, з якою буде працювати

# CREATE TABLE AS SELECT

```
CREATE TABLE new_table  
AS (  
    SELECT column1, column2  
    FROM exist_table  
    WHERE column3 > some_value  
);
```

Створює нову таблицю, що матиме структуру,  
як і результуюча таблиця внутрішнього запиту  
**SELECT**

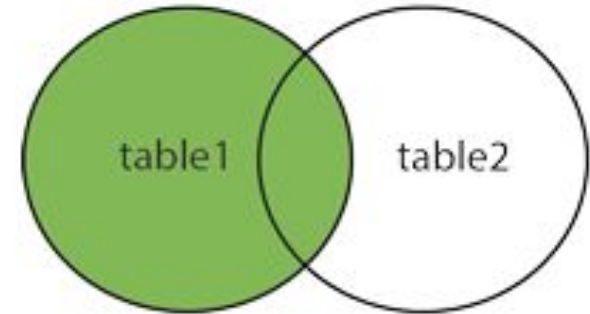
# INNER JOIN



```
SELECT table_1.column_1, table_1.column_2,  
table_2.column_2  
FROM table_1  
INNER JOIN table_2  
WHERE table_1.column_1 = table_2.column_1;
```

Даний запит поверне таблицю, що буде містити дані з двох таблиць, що задовольняють умову

# LEFT JOIN



```
SELECT table_1.column_1, table_1.column_2,  
table_2.column_2  
FROM table_1  
LEFT JOIN table_2  
ON table_1.column_1 = table_2.column_1;
```

Даний запит поверне таблицю, що буде містити всі дані з першої таблиці та дані з другої таблиці, що задовольняють умову