

# MySQL

Заняття #4

- Оператор сортування order by
- Оператор limit
- Агрегативні функції
- Групування
- Підзапити



## Оператор ORDER BY

Оператор **ORDER BY** сортує значення по одному або декількох стовпцях. Наприклад, впорядкуємо вибірку з таблиці Products по стовпцю Price:

```
SELECT * FROM Products ORDER BY Price;
```

Фактично за замовчуванням використовується оператор **ASC**, який сортує по зростанню:

```
SELECT * FROM Products ORDER BY Price ASC;
```

Можна упорядкувати дані за псевдонімом стовпчика, який визначається за допомогою оператора **AS**:

```
SELECT ProductName, ProductCount * Price AS TotalSum FROM Products ORDER BY TotalSum;
```

За замовчуванням дані сортуються за зростанням, проте за допомогою оператора **DESC** можна задати сортування за спаданням.

```
SELECT ProductName, ProductCount FROM Products ORDER BY ProductCount DESC;
```

**ASC** і **DESC** можна окремо застосовувати для різних стовпців сортуючи по зростанню і зменшенню:

```
SELECT ProductName, Price, Manufacturer FROM Products  
ORDER BY Manufacturer ASC, ProductName DESC;
```

## Агрегатна функція COUNT

**COUNT** (параметр) – обчислює кількість записів, вказаних у параметрі.

Якщо необхідно отримати кількість усіх записів, то в якості параметра вказується \* (або унікальний ідентифікатор):

```
SELECT COUNT(*) FROM Products
```

Якщо в якості параметра вказано ім'я стовпця, то функція поверне кількість записів, в яких цей стовпець має не **NULL** значення.

```
SELECT COUNT(Manufacturer) FROM Products
```

Щоб знати, скільки різних значень має стовпець, перед його іменем вказується ключове слово **DISTINCT**.

## Агрегатна функція AVG

**AVG** (параметр) – обчислює середнє арифметичне значень стовпця, вказаного в параметрі.

Параметр може представляти собою вираз, який містить ім'я стовпця. Тоді використання **DISTINCT** не дозволяється.

Наприклад, знайдемо середню ціну товарів з бази даних:

```
SELECT AVG(Price) AS Average_Price FROM Products
```

На етапі вибірки можна застосовувати фільтрацію. Наприклад, знайдемо середню ціну для товарів певного виробника:

```
SELECT AVG(Price) FROM Products WHERE Manufacturer='Apple'
```

Також можна знаходити середнє значення для більш складних виразів. Наприклад, знайдемо середню суму всіх товарів, з огляду на їх кількість:

```
SELECT AVG(Price * ProductCount) FROM Products
```

З функцією **AVG** можна використовувати лише числові поля.

## Агрегатна функція SUM

**SUM** (параметр) – обчислює суму значень стовпця, вказаного як параметр.

Параметр може представляти собою вираз, який містить ім'я стовпця. Тоді використання **DISTINCT** не дозволяється.

Наприклад, підрахуємо загальна кількість товарів:

```
SELECT SUM(ProductCount) FROM Products
```

На етапі вибірки можна застосовувати фільтрацію. Наприклад, знайдемо кількість товарів певного виробника:

```
SELECT SUM(ProductCount) FROM Products WHERE Manufacturer='Apple'
```

Також замість імені стовпця може передаватися обчислюється вираз. Наприклад, знайдемо загальну вартість всіх наявних товарів:

```
SELECT SUM(ProductCount * Price) FROM Products
```

З функцією **SUM** можна використовувати лише числові поля.

## Агрегатні функції MIN та MAX

**MAX** (параметр) – обчислює найбільше з усіх вибраних значень стовпця.

**MIN** (параметр) – обчислює найменше з усіх вибраних значень стовпця.

Наприклад, знайдемо мінімальну та максимальну ціну серед товарів:

```
SELECT MIN(Price), MAX(Price) FROM Products
```

Дані функції також ігнорують значення **NULL** і не враховують їх при підрахунку.

З функціями **MAX** і **MIN** можуть використовуватись і числові, і символьні поля.

## Групування

**GROUP BY** служить для групування записів за значеннями одного або декількох стовпців.

Якщо в SQL-виразі використовується оператор **WHERE**, який задає фільтр записів, то оператор **GROUP BY** знаходиться і виконується після нього.

Для визначення, які записи повинні увійти в групи, служить оператор **HAVING**, який використовується разом з **GROUP BY**.

Якщо оператор **HAVING** не використовується, то групуванню підлягають усі записи, відфільтровані оператором **WHERE**.

Якщо **WHERE** не використовується, то групуються усі записи таблиці.



## Використання агрегатних функцій з групуванням

Фраза **GROUP BY** дозволяє визначати підмножину значень в деякому стовпці і застосовувати агрегатну функцію до цієї підмножини. Потрібно утворювати групи і лише тоді виконувати певні операції.

Приклад – групувати продавця, і тоді рахувати певні операції з ним.

Тоді необхідно оголосити стовпці та агрегатні функції у фразі **SELECT**.

Приклад. Знайти найбільшу суму, отриману кожним продавцем:

```
SELECT snum, MAX(amt) FROM Orders GROUP BY snum;
```

Приклад. Знайти найбільший платіж, який проводив кожен продавець кожен день:

```
SELECT snum, odate, MAX(amt) FROM Orders GROUP BY snum, odate;
```

## Використання підзапитів

При використанні підзапитів у предикатах, які використовують операції порівняння, необхідно, щоб результат підзапиту видавав лише один рядок. В іншому випадку команда не виконається.

Якщо в результаті підзапиту не буде ніяких значень, то інструкція виконається, але не видасть ніяких результатів. Предикат, в якому розміщений такий підзапит, є невідомий і має такий ефект як невірний, тому команда не має результатів.

Приклад. Вивести інформацію про усі операції купівлі-продажу, які обслуговуються продавцем Motika, припустивши, що його номер нам невідомий.

```
SELECT * FROM Orders  
WHERE snum =  
(SELECT snum FROM Sellers  
WHERE sname = 'Motika');
```

Стандарт ANSI забороняє додавати для порівняння два підзапити: <підзапит> <оператор> <підзапит>.

## Використання агрегатних функцій у підзапитах

Приклад. Вивести усі операції за 3 жовтня з платежем вище середнього:

```
SELECT * FROM Orders  
WHERE odate = '2009-10-03'  
AND amt >  
(SELECT AVG(amt)  
FROM Orders);
```

Приклад. Порахувати кількість замовників з рейтингом, вищим середнього, у місті San Jose:

```
SELECT rating, COUNT(cnum)  
FROM Customers  
GROUP BY rating  
HAVING rating >  
(SELECT AVG(rating)  
FROM Customers  
WHERE city = 'SanJose');
```

# Додаткові джерела інформації:

## Посилання

[https://www.w3schools.com/sql/sql\\_orderby.asp](https://www.w3schools.com/sql/sql_orderby.asp)

[https://www.w3schools.com/sql/sql\\_ref\\_top.asp](https://www.w3schools.com/sql/sql_ref_top.asp)

[https://www.w3schools.com/sql/sql\\_min\\_max.asp](https://www.w3schools.com/sql/sql_min_max.asp)

[https://www.w3schools.com/sql/sql\\_count\\_avg\\_sum.asp](https://www.w3schools.com/sql/sql_count_avg_sum.asp)

[https://www.w3schools.com/sql/sql\\_groupby.asp](https://www.w3schools.com/sql/sql_groupby.asp)

[https://www.w3schools.com/sql/sql\\_having.asp](https://www.w3schools.com/sql/sql_having.asp)