

# MySQL

Заняття #2

- Модифікація таблиць
- Внесення даних
- Оператори select, where



## Модифікація структури таблиці. Інструкція ALTER TABLE

Якщо таблиця вже була раніше створена, і її необхідно змінити, то для цього застосовується команда **ALTER TABLE**. Її скорочений формальний синтаксис:

### Додавання нового стовпця:

```
ALTER TABLE <ім'я таблиці>
```

```
ADD COLUMN <ім'я стовп> <тип(розмір)> [атрибут стовпця];
```

### Видалення стовпця з таблиці:

```
ALTER TABLE <ім'я таблиці>
```

```
DROP COLUMN <ім'я стовп>;
```

При цьому видаляються усі дані стовпця. Якщо стовпець є первинним ключем, на який посилається зовнішній ключ з іншої таблиці, то видалення не відбудеться. Спочатку потрібно скоректувати дані в іншій таблиці.

## Модифікація структури таблиці. Інструкція ALTER TABLE

### **Модифікування стовпця:**

ALTER TABLE <ім'я таблиці>

MODIFY COLUMN <ім'я стовп> <тип(розмір)> [атрибут стовпця];

### **Зміна значення по замовчуванню:**

ALTER TABLE <ім'я таблиці>

ALTER COLUMN <ім'я стовп> SET DEFAULT <значення>;

## Інструкція внесення даних (INSERT)

### Найпростіший синтаксис:

```
INSERT INTO <ім'я_таблиці>  
VALUES (<зн.стовпця1>, ..., <зн.стовпцяN>);
```

Ім'я таблиці має бути визначеною командою CREATE TABLE. Кожне значення, що додається, має співпадати з типом даних стовпця, в який воно додається. Інструкція не виводить результатів, але СУБД повинна видати підтвердження того, що дані були додані. Якщо необхідно ввести невизначене значення, то у відповідне поле додається значення NULL.

**Приклад:** щоб додати перший рядок у таблицю Sellers, можна використати команду:

```
INSERT INTO Sellers  
VALUES (1001, 'Peel', 'London', 0.12);
```

Можна вказувати конкретні стовпці, в які потрібно додати значення. Це дозволяє додавати дані в довільному порядку:

```
INSERT INTO Sellers (snum, sname, comm)  
VALUES (1001, 'Peel', 0.12);
```

## Визначення SQL-запиту на вибірку

**Запит на вибірку даних** – це команда або інструкція, яка дається СУБД, щоб вона вивела певну інформацію з таблиць бази даних.

Мінімальний синтаксис інструкції **SELECT**

**SELECT** стовпці **FROM** таблиця;

Аргументи **SELECT** та **FROM** є обов'язковими в SQL-виразі.

## Основні аргументи інструкції SELECT

- ✓ **\*** – вказує, що вибрано усі стовпці заданої таблиці або таблиць;
- ✓ **вираз-стовпець** – ім'я стовпця або вираз з декількох імен, з яких вибираються дані. Якщо включити декілька стовпців, то вони будуть вибиратись за вказаним порядком;
- ✓ **псевдонім** – ім'я або імена, які стануть заголовками стовпців у результаті запиту;
- ✓ **таблиця** – ім'я таблиці, з якої вибираються записи.

## Фільтрація даних. оператор WHERE

Найчастіше необхідно використовувати не всі дані з БД, а тільки ті, які відповідають певній умові. Для фільтрації даних в команді SELECT застосовується оператор WHERE, після якого вказується умова:

### **WHERE умова**

Якщо умова істинно, то рядок потрапляє в результуючу вибірку. Як можна використовувати операції порівняння, які порівнюють два вирази:

- =: Порівняння на рівність
- !=: Порівняння на нерівність
- <>: Порівняння на нерівність
- <: Менше ніж
- >: Більше ніж
- <=: Менше ніж або дорівнює
- >=: Більше ніж або дорівнює

Наприклад, виберемо всіх країни, в яких континент є Європа:

```
select * from world.country where continent = 'Europe';
```



## Фільтрація даних. оператор WHERE

Логічні оператори дозволяють об'єднати кілька умов. В MySQL можна використовувати такі логічні оператори:

**AND: операція логічного І. Вона об'єднує два вирази:**

- вираз1 AND вираз2

Тільки якщо обидва цих вираження одночасно істинними, то і загальне умова оператора AND також було це слово. Тобто якщо і перша умова істинно, і друге.

**OR: операція логічного АБО. Вона також об'єднує два вирази:**

- вираз1 OR вираз2

Якщо хоча б одне з цих виразів істинно, то загальне умова оператора OR також було це слово. Тобто якщо або перша умова істинно, або друге.

**NOT: операція логічного заперечення:**

- NOT вираз

Якщо вираз в цій операції помилково, то загальне умова істинно.

## Фільтрація даних. оператор WHERE

### Пріоритет операцій

В однієї умови при необхідності ми можемо об'єднувати кілька логічних операцій. Однак слід враховувати, що найбільш пріоритетним операцією, яка виконується в першу чергу, є NOT, менш пріоритетна - AND і операція з найменшим пріоритетом - OR. наприклад:

```
SELECT * FROM Products  
WHERE Manufacturer = 'Samsung' OR NOT Price > 30000 AND ProductCount > 2;
```

В даному випадку спочатку обчислюється вираз NOT Price > 30000, тобто ціна повинна бути менше або дорівнює 30000.

Потім обчислюється вираз NOT Price > 30000 AND ProductCount > 2, тобто ціна повинна бути менше або дорівнює 30000 і одночасно кількість товарів повинно бути більше 2.

В кінці обчислюється оператор OR - або ціна повинна бути менше або дорівнює 30000 і одночасно кількість товарів повинно бути більше 2, або виробником повинен бути Samsung.

# Додаткові джерела інформації:

## Посилання

[https://www.w3schools.com/sql/sql\\_alter.asp](https://www.w3schools.com/sql/sql_alter.asp)

[https://www.w3schools.com/sql/sql\\_insert.asp](https://www.w3schools.com/sql/sql_insert.asp)

[https://www.w3schools.com/sql/sql\\_select.asp](https://www.w3schools.com/sql/sql_select.asp)

[https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)

[https://www.w3schools.com/sql/sql\\_and\\_or.asp](https://www.w3schools.com/sql/sql_and_or.asp)