



Робота з базами даних

Як зберігати список
zareestrovanih na сайті
користувачів?

Можна зберігати їх у текстовому файлі.

Кожен новий користувач буде дописуватися в кінець файлу.

А якщо користувачів мільйон?
Десять мільйонів?

Як знайти потрібного
користувача?

Просканувати весь файл від
початку до кінця.

Проблема: що більший файл, то
довше триватиме пошук.

Як вирішується проблема пошуку
в книжках?

У книзі створюється зміст або
алфавітний покажчик.

Тоді легше знайти ту чи іншу
інформацію. Достатньо знайти її у
змісті і одразу відкрити відповідну
сторінку.

Із текстовим файлом можна
зробити так само.

Створити окремо **індекси** (зміст),
щоб можна легко і швидко знайти
потрібну інформацію.

Можна навіть створити кілька
індексів.

У книзі зміст подає інформацію,
згруповану за темами, а
алфавітний покажчик — за
алфавітом.

Так можна створити й індекси для пошуку користувача за датою реєстрації, алфавітом, місцем проживання тощо.

Для цього існують **бази даних**.

По суті, **база даних** — це набір таблиць (які можуть бути пов'язані між собою) та їхніх індексів (для швидкого пошуку).

Навіщо бази даних, якщо є таблиці **Microsoft Excel**?

	A	B	C	D	E
1	Логін	Пароль	Дата реєстрації	Електронна пошта	
2	user1	111111	01.10.2019	user1@gmail.com	
3	user2	123321	08.10.2019	user2@gmail.com	
4	banana	@#\$^&*()	14.10.2019	banana@example.com	
5					
6					
7					
8					
9					
10					
11					
12					

РНР запросто може працювати з
Excel-документами.

Однак, як і у випадку з текстовими
файлами, доведеться вручну
програмувати операції пошуку,
вставки і видалення рядків.

При роботі з базами даних цього програмувати не потрібно.

Є багато програм із уже готовими командами для роботи з БД.

Програми для роботи з БД
називаються **системами**
керування базами даних (СКБД).

Англ. **Database Management**
System (DBMS).

Наразі існує багато СКБД.

Найвідоміші серед них:

MySQL

PostgreSQL

SQLite

MS SQL Server

MS Access

Є також повністю безкоштовний
різновид MySQL під назвою
MariaDB.

Більшість СКБД зберігають
таблиці та їхні індекси в окремих
файлах.

SQLite і **MS Access** зберігають цілу
БД в одному файлі.

Більшість СКБД мають майже
однаковий набір команд для
пошуку, додавання та видалення
даних з БД.

Ці команди утворюють мову **SQL**
(**Structured Query Language**).

SQL не є окремою незалежною мовою програмування.

Це скоріше мова взаємодії користувача з БД.

Майже кожна мова програмування
дозволяє працювати з БД за
допомогою SQL-команд.

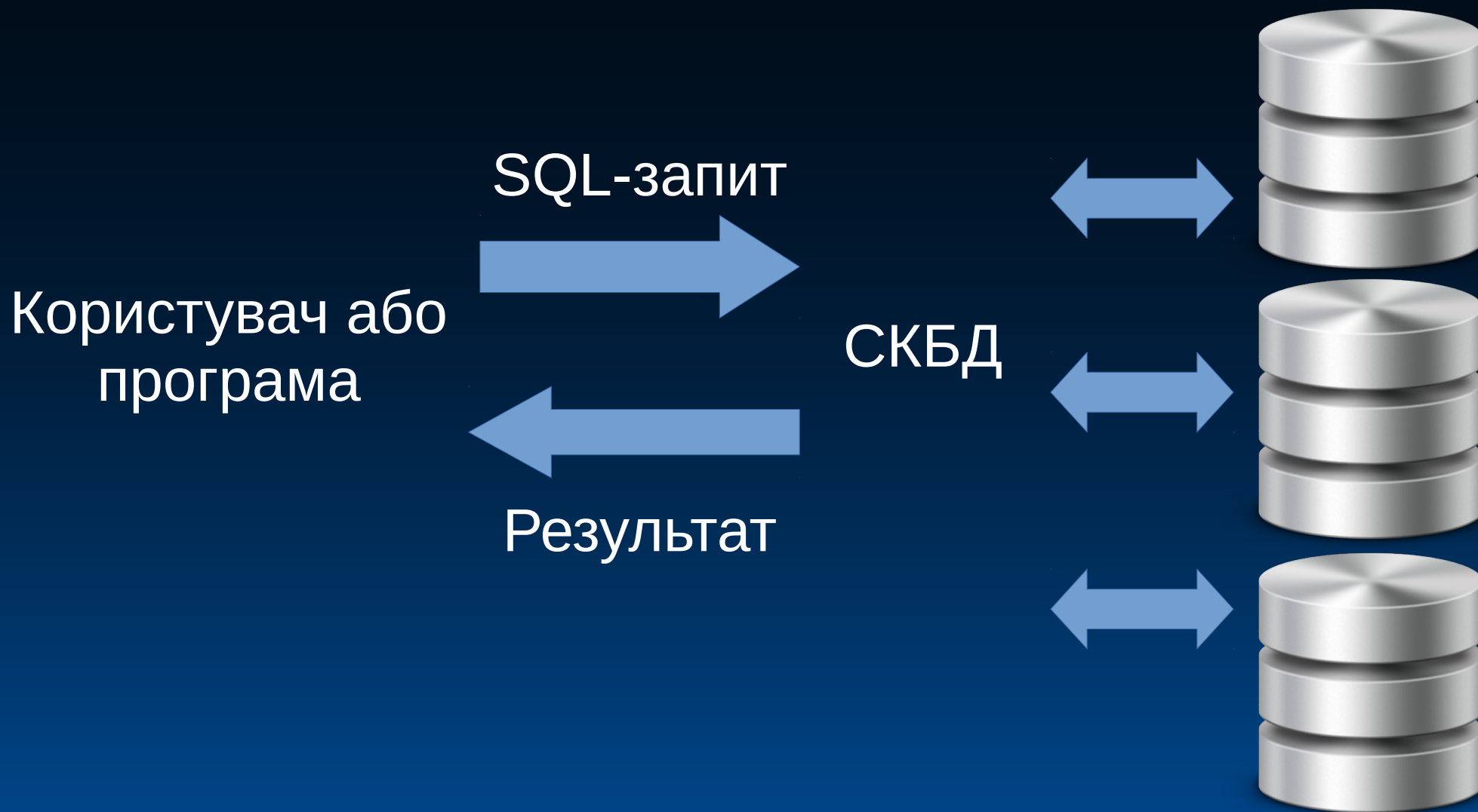
Користувач також може працювати напряму з БД через консоль і SQL-команди.

```
| test |
+-----+
1 row in set (0.02 sec)

mysql> use software;
Database changed
mysql> CREATE table Apple(id int, name varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> desc Apple;
+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11)       | YES  |     | NULL    |       |
| name  | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql>
```



Кожна БД складається з таблиць.

Таблиці прийнято називати малими латинськими літерами без пробілів.

Наприклад, **users**, **posts**,
comments.

Таблиці прийнято називати іменниками **у множині**, оскільки таблиця містить інформацію не про один об'єкт, а багато.

Кожна таблиця складається з
рядків і стовпців (полів).

Поля також прийнято називати
малими латинськими літерами без
пробілів.

Наприклад, username, login,
password, email, reg_date.

Наприклад, структура (список полів) таблиці **users**:

- **username** (текст)
- **login** (текст)
- **password** (текст)
- **email** (текст)
- **age** (число)
- **reg_date** (дата)

Припустимо, в таблиці **users** є 2 користувачі з однаковим іменем **username**.

Як їх розрізнити?

Прийнято кожному рядку давати унікальний номер (**id** - ідентифікатор).

Щоб розрізняти схожі дані,
прийнято, що кожна таблиця
повинна мати поле, значення
якого різне для кожного рядка.

Таке поле, яке має унікальне
значення для кожного рядка,
називається **первинним ключем**
(primary key).

Зазвичай його називають **id**.

Нова структура таблиці **users**:

- **id** (число)
- **username** (текст)
- **login** (текст)
- **password** (текст)
- **email** (текст)
- **age** (число)
- **reg_date** (дата)

id	username	login	password	email	reg_date
1	Bob	bob	111111	bob@example.com	2019-10-01
2	Alice	alice	123456	alice@meta.ua	2019-10-08
3	Bob	bob	password	bob74@gmail.com	2019-10-14

У багатьох СКБД є можливість
зробити так, щоб поле **id**
збільшувалось автоматично для
кожного нового рядка таблиці.

Зазвичай, така властивість
називається **AUTO_INCREMENT**.

У даному курсі ми будемо
використовувати СКБД **MySQL**
(або **MariaDB**).

Ця СКБД постачається у складі
пакету програм **XAMPP**.





XAMPP Control Panel v3.2.1

Config

Netstat

Shell

Explorer

Services

Help

Quit

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	MySQL			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	FileZilla			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	Mercury			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	Tomcat			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>

오 후 7:22:02 [main] most application stuff but whenever you do something with services
오 후 7:22:02 [main] there will be a security dialogue or things will break! So think
오 후 7:22:02 [main] about running this application with administrator rights!
오 후 7:22:02 [main] XAMPP Installation Directory: "c:\xampp"
오 후 7:22:02 [main] Checking for prerequisites
오 후 7:22:07 [main] All prerequisites found
오 후 7:22:07 [main] Initializing Modules
오 후 7:22:07 [main] Starting Check-Timer
오 후 7:22:07 [main] Control Panel Ready

Щоб не працювати з MySQL через консоль, є безкоштовний зручний веб-додаток **phpMyAdmin**, який також постачається у складі **XAMPP**.

Його можна відкрити у браузері, ввівши **localhost/phpmyadmin**



Welcome to phpMyAdmin

Language

English ▼

Log in ⓘ

Username:

Password:

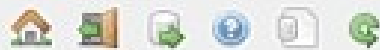
Go

Для роботи з БД MySQL потрібно ввести окремий логін і пароль.

За замовчуванням, XAMPP у Windows встановлює логін **root**, пароль порожній.

Створити нову БД можна в консолі
за допомогою SQL-команд.

Але для початку зробимо це в
phpMyAdmin клацанням мишею.



- coaching
- coaching2
- coaching3
- coolest
- coolio2
- coolio3
- databasename
- drupal
- drupal5
- drupal6
- drupal7
- drupal8
- drupal10

localhost



Databases



SQL



Status



Processes



Privileges

General Settings



Change password



MySQL connection collation : utf8_general_ci



Appearance Settings



Language : English



Theme / Style: pmahomme



Font size: 82%



More settings

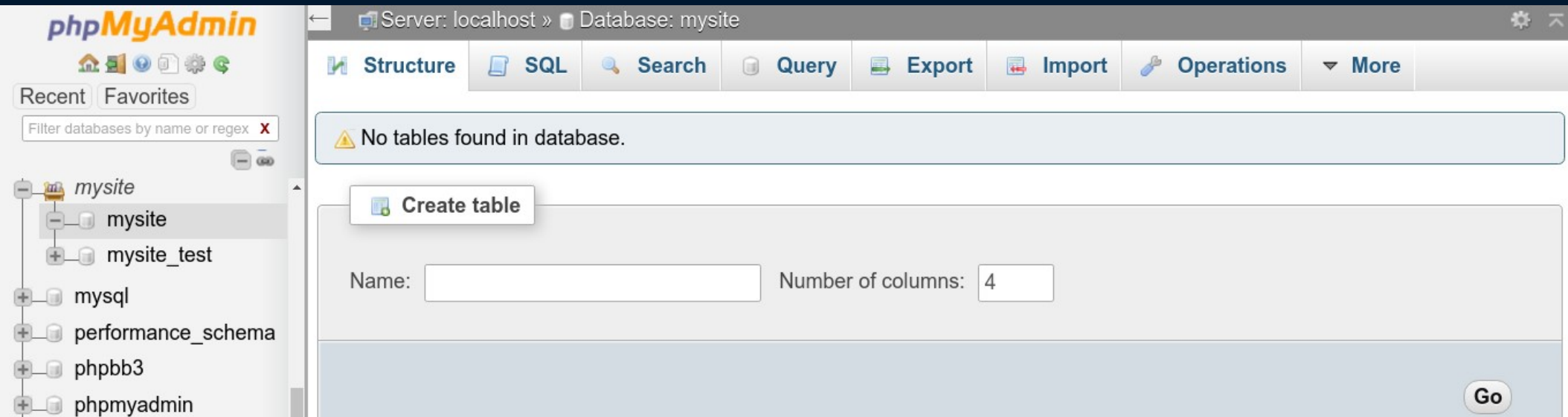
Введемо назву нової БД (без пробілів) і задамо кодування символів.

Рекомендується
utf8mb4_unicode_ci.

Databases

 Create database 

Новостворена БД поки що не має таблиць.



Введемо назву нової таблиці
users. І кількість полів у ній.

 Create table

Name:

Number of columns:

Go

Далі заповнюємо назву і тип кожного поля таблиці.

Table name:	<input type="text" value="users"/>	Add	<input type="text" value="1"/>	column(s)	<input type="button" value="Go"/>	
Name	Type	Length/Values	Default	Collation	Attributes	Null
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						
<input type="text"/>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Pick from Central Columns						

Для цілих чисел залишаємо **INT**.

Для поля **id** ставимо галочку **A_I**
(**AUTO_INCREMENT**).

Для короткого тексту (логін,
пароль, e-mail) обираємо
VARCHAR і вписуємо довжину
(length) **255**.

У MySQL є також тип даних **TEXT**,
але він, як правило,
використовується для довгого
тексту.

Наприклад, текст статті чи
коментаря.

Для дати обираємо тип
TIMESTAMP і обираємо в default
CURRENT_TIMESTAMP.


Тоді поточна дата й час
підставлятимуться автоматично
при додаванні нового рядка
таблиці.


У MySQL є також типи даних **DATE** і **DATETIME**, але вони не враховують часовий пояс поточного користувача.




















У phpMyAdmin можна також вказати тип поля **BOOLEAN**, однак у MySQL немає такого типу даних.


Насправді це поле збережеться як **TINYINT(1)**, тобто ціле число.

Таблиця з полями








 Table structure


 Relation view


#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	id 	int(11)		No	None	AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	username	varchar(255) utf8mb4_unicode_ci		No	None		 Change  Drop  More
<input type="checkbox"/>	3	login	varchar(255) utf8mb4_unicode_ci		No	None		 Change  Drop  More
<input type="checkbox"/>	4	password	varchar(255) utf8mb4_unicode_ci		No	None		 Change  Drop  More
<input type="checkbox"/>	5	reg_date	timestamp		No	CURRENT_TIMESTAMP		 Change  Drop  More
<input type="checkbox"/>	6	admin	tinyint(1)		No	None		 Change  Drop  More





 ☐ Check all


With selected:

 Browse  Change  Drop  Primary  Unique  Index  Add to central columns


 Remove from central columns


 Print view

 Propose table structure  Track table  Move columns  Improve table structure

 Add

column(s)



 Go

+ Indexes

На вкладці **Browse** видно, що в таблиці немає рядків.

Server: localhost » Database: mysite » Table: users

Browse Structure SQL Search Insert Export Import

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0012 seconds.)

```
SELECT * FROM `users`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)]









Query results operations


Create view

Новий рядок до таблиці можна
вставити за допомогою SQL-
команди.

Але для початку зробимо це в
phpMyAdmin.

Вкладка Insert.

 Browse	 Structure	 SQL	 Search	 Insert	 Export	 Import	 Privilege
-----------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------

Column	Type	Function	Null	Value
id	int(11)	<input type="text"/>		<input type="text"/>
username	varchar(255)	<input type="text"/>		<input type="text" value="Bob"/>
login	varchar(255)	<input type="text"/>		<input type="text" value="bob"/>
password	varchar(255)	<input type="text"/>		<input type="text" value="123456"/>
reg_date	timestamp	<input type="text"/>		<input type="text" value="CURRENT_TIMESTAMP"/> 
admin	tinyint(1)	<input type="text"/>		<input type="text" value="0"/>

Go

Поле **id** залишаємо порожнім.
Воно заповниться автоматично
завдяки властивості
AUTO_INCREMENT.

Перший рядок матиме **id = 1**,
наступний **2** і т.д.

Якщо є поле дата (TIMESTAMP),
його теж залишаємо порожнім.
Воно заповниться автоматично
завдяки властивості
CURRENT_TIMESTAMP.

✓ 1 row inserted.
Inserted row id: 1

```
INSERT INTO `users` (`id`, `username`, `login`, `password`, `reg_date`, `admin`) VALUES (NULL, 'Bob', 'bob', '123456',  
CURRENT_TIMESTAMP, '0');
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Run SQL query/queries on table **mysite.users**: ⓘ

```
1 INSERT INTO `users` (`id`, `username`, `login`, `password`, `reg_date`,  
  `admin`) VALUES (NULL, 'Bob', 'bob', '123456', CURRENT_TIMESTAMP, '0');
```

Columns

id
username
login
password
reg_date
admin

На вкладці **Browse** видно, що таблиця має 1 рядок.

Browse Structure SQL Search Insert Export Import

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `users`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Edit](#)]

☐ Show all | Number of rows: 25 | Filter rows:

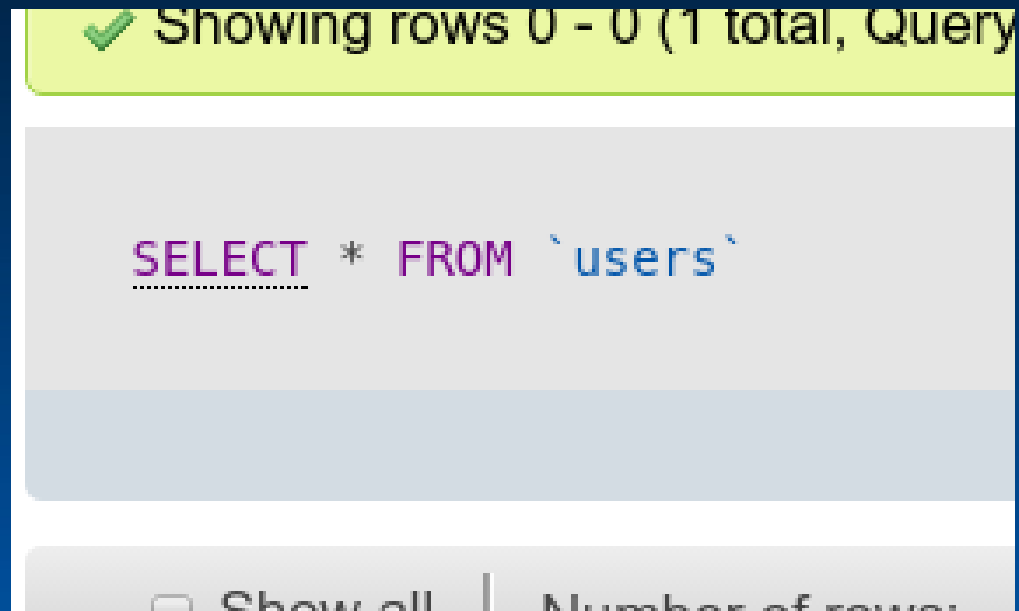
+ Options

		id	username	login	password	reg_date	admin
<input type="checkbox"/>	Edit Copy Delete	1	Bob	bob	123456	2019-10-14 22:49:56	0

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows:

Якщо придивитися, у верхній частині вікна **phpMyAdmin** написано, яку SQL-команду було виконано, щоб показати даний результат.



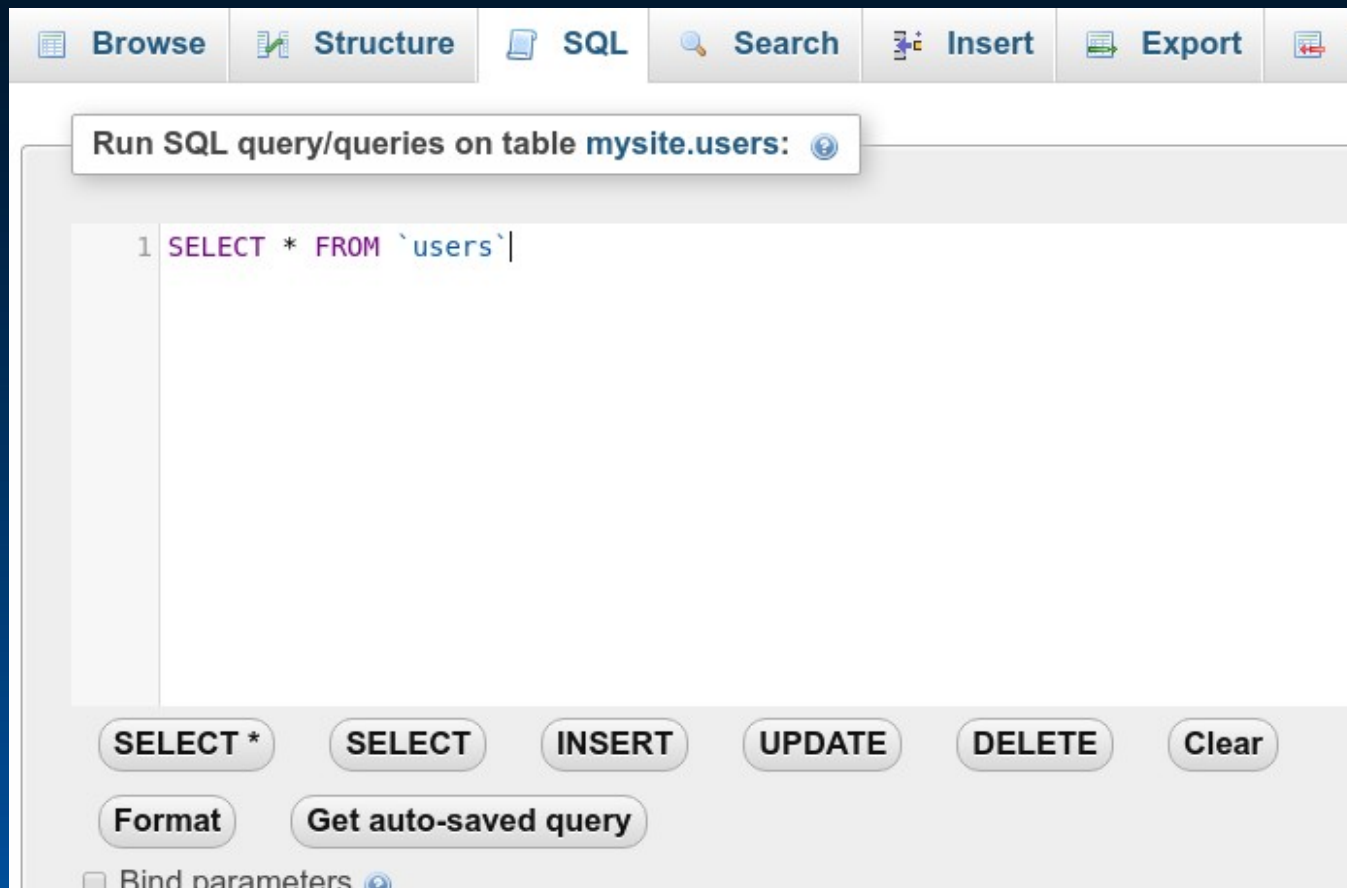
По суті, при кожному натисканні мишею за кадром виконується SQL-команда.

phpMyAdmin надає зручний інтерфейс і дозволяє не задумуватись про SQL.

Але нам, як програмістам, слід
навчитися працювати з БД за
допомогою команд **SQL**.

SQL-команди називаються
запитами (queries).

У phpMyAdmin можна також
вводити SQL-запити напряду на
вкладці **SQL**.



Існують 4 основні типи операцій з
БД.

1. Пошук інформації в БД.

```
SELECT * FROM users;
```

* означає витягти всі поля таблиці

Крапка з комою в кінці запиту
необов'язкова.

```
SELECT login,password FROM  
users
```

Витягти не всі поля таблиці, а
лише поля **login** і **password**

```
SELECT login,password FROM  
users WHERE id=2
```

Витягти логін і пароль лише
користувача, в якого **id=2**

```
SELECT login,password FROM  
users WHERE id=1 OR id=2
```

Витягти логін і пароль
користувача, в якого **id=1** або **id=2**

Тобто результат матиме
інформацію про обох користувачів
(2 рядки).


```
SELECT login,password FROM  
users WHERE id IN (1,2)
```

Te came

2. Додавання інформації до БД.

```
INSERT INTO users  
(login,password,email) VALUES  
( 'alice', '123456', 'alice@meta.ua')
```

Текст може бути як в одинарних,
так і в подвійних лапках.

3. Редагування інформації в БД.

```
UPDATE users SET  
    password='hello',  
    email='bobb@b@gmail.com'  
    WHERE id=1
```

Вказувати WHERE обов'язково,
інакше будуть відредаговані ВСІ
рядки!

4. Видалення інформації з БД.

DELETE FROM users WHERE id=2

Вказувати WHERE обов'язково,
інакше будуть видалені ВСІ рядки!

Запит **SELECT** повертає
результат.

Інші типи запитів — ні.

Примітка: назви таблиць та полів дозволяється брати в кутові лапки, але тільки в кутові.

```
SELECT `login`, `password` FROM  
`users` WHERE id=2
```

Кутові лапки знаходяться зліва від цифри 1 на клавіатурі (англійською мовою).

Як виконувати SQL-запити у PHP-кодi?

Є 2 способи: **процедурний** і **об'єктно-орієнтований**.

Працюють вони ідентично.

1. Процедурний стиль.

```
<?php
$link = mysqli_connect("localhost",
"логін_БД", "пароль_БД", "назва_БД");

if (mysqli_connect_errno() != 0) {
    die(mysqli_connect_error());
}
```


Змінна **\$link** міститиме вказівник на з'єднання з БД.

Функція **mysqli_connect_errno()** повертає 0, якщо вдалося успішно підключитися до БД.

Якщо вона повернула не 0, то слід перевірити, чи правильно вказано логін, пароль або назва БД.

Функція `mysqli_connect_error()`
містить словесний опис помилки,
якщо вона виникла.

```
$query = "SELECT * FROM users";
```

```
$result = mysqli_query($link, $query);
```

```
if(mysqli_errno($link) != 0) {  
    die(mysqli_error($link));  
}
```

```
while($row = mysqli_fetch_assoc($result)) {  
    echo $row["username"];  
    echo $row["login"];  
    echo $row["email"];  
}
```

Будь-які запити до БД у PHP виконуються за допомогою функції `mysqli_query`.

Перший параметр — до якої БД звертаємось (може бути відкрито кілька БД одночасно).

Другий параметр — текст SQL-запиту.

Змінна **\$result** міститиме
“запакований” результат з БД, усі
рядки одразу.

Зчитувати треба їх звідти в циклі
по одному рядку за допомогою
функції **mysqli_fetch_assoc**.

Коли рядків більше немає,
функція поверне **NULL**, і цикл
припиниться.

У кінці можна закрити з'єднання з
БД функцією

```
mysqli_close($link);
```

Але це не обов'язково. Усі
з'єднання з БД закриються
автоматично після завершення
PHP-скрипту.

2. Об'єктно-орієнтований стиль.

```
<?php
$mysqli = new mysqli("localhost",
"логін_БД", "пароль_БД",
"назва_БД");

if ($mysqli->connect_errno != 0) {
    die($mysqli->connect_error);
}
```

Тут змінна `$mysqli` — це об'єкт
класу `mysqli`.


```
$query = "SELECT * FROM users";
```

```
$result = $mysqli->query($query);
```

```
if($mysqli->errno != 0) {  
    die($mysqli->error);  
}
```

```
while($row = $result->fetch_assoc()) {  
    echo $row["username"];  
    echo $row["login"];  
    echo $row["email"];  
}
```

Тут змінна `$result` — це об'єкт
класу `mysqli_result`.

У кінці можна закрити з'єднання з
БД функцією

```
$mysqli->close( );
```

Але це не обов'язково.