



## БАЗИ ДАННИ

# Работа с MySQL БД SQL

# Съдържание

- Databases, MySQL, SQL - същност и предназначение
- PhpMyAdmin – database IMPORT
- SQL заявки –
  - select,
  - read,
  - update,
  - delete

# Съдържание - 2

- Връзка между PHP и база данни /БД/

**БАЗИ ДАННИ**

**MySQL**

**SQL**

**phpMyAdmin**

**същност**

**предназначение**

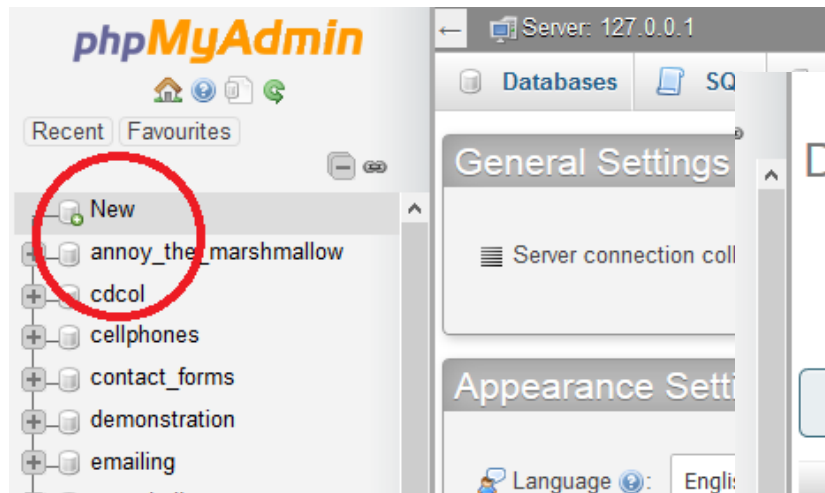
# База данни, MySQL, SQL

- База данни /БД/ - съвкупност от данни/информация/.
- **MySQL** – DBRMS – database relational management system или програма, чрез която записваме и използваме информацията, съхранена в БД.
- Как - чрез **SQL** заявки
- Какво е **SQL** – Standard Query Language
- **phpMyAdmin** – инструмент, написан с PHP. Използва се в браузъра. Служи за администриране/управление на MySQL.
  
- БД, които създаваме и достъпваме в **phpMyAdmin** представляват съвкупност от данни, организирани в логически свързани **таблици**.

# PhpMyAdmin – database, table IMPORT

# database IMPORT

1. Създаваме нова база данни в  
PhpMyAdmin



2. Избираме UTF8 general\_ci – за да нямаме  
проблеми с отпечатването на текстовете на  
кирилица

## Databases

Create database

foods

utf8\_general\_ci

Create



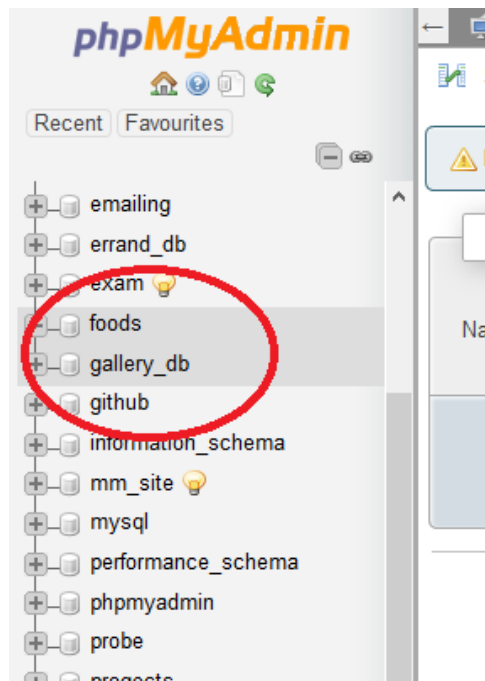
Note: Enabling the database statistics here might cause heavy traffic between the web server and the database.

Database

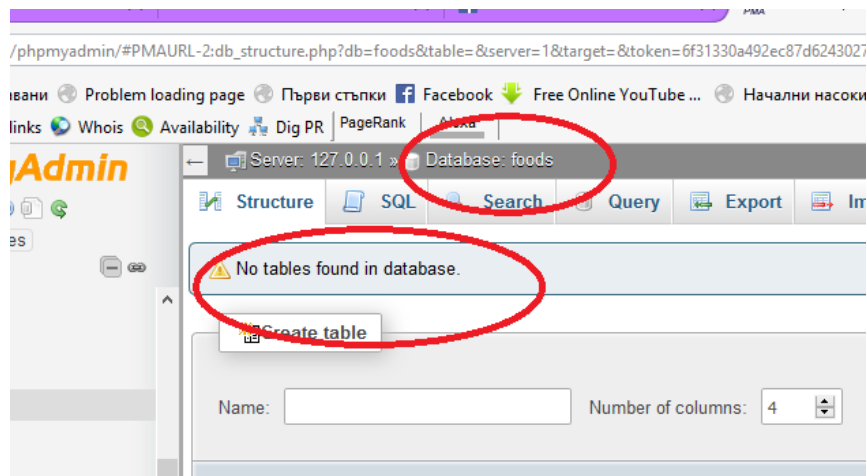
Collation

# database IMPORT - 2

3. Новата база данни/БД/ се появява в списъка при останалите -



4. Избираме новата, все още празна БД

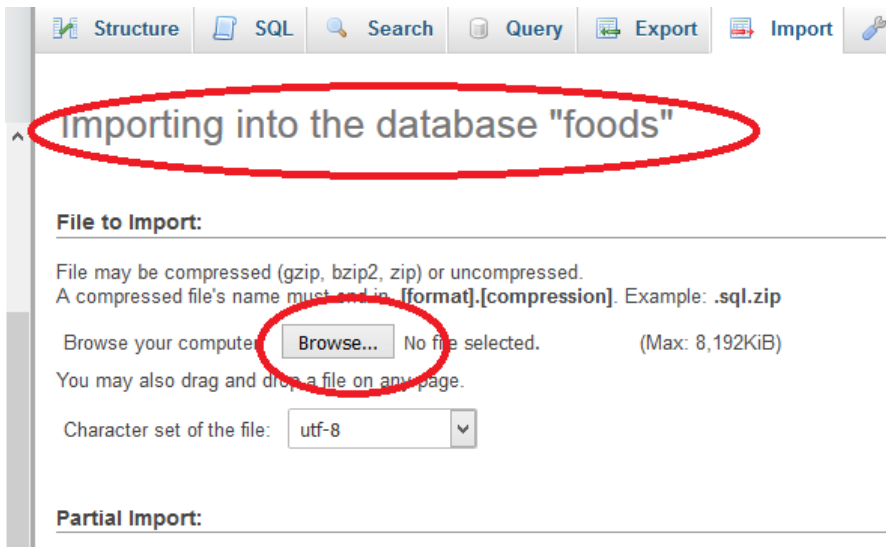


5. Натискаме IMPORT

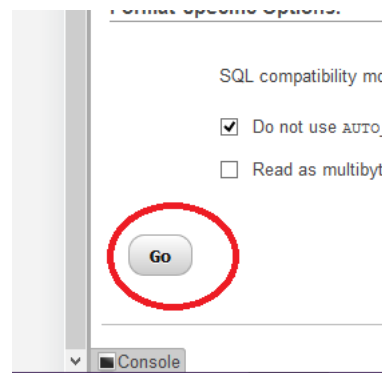


# database IMPORT - 3

6. Уверяваме се, че сме в БД, която трябва, и търсим БД, която ще импортираме.

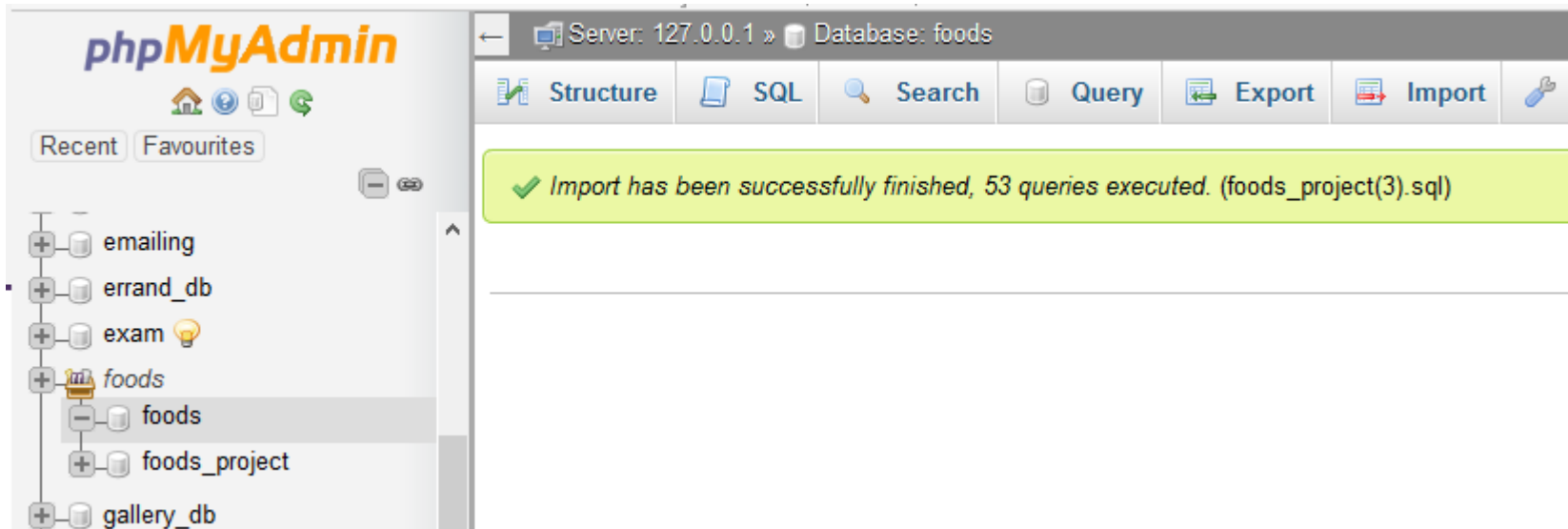


7. Натискаме GO – това ще импортира новата БД



# database IMPORT - 4

8. Новата БД е присъединена към останалите в PhpMyAdmin



**SQL заявки към БД**

# Заявки в БД

Написани с SQL. Наричаме ги – query/queries. Насочени са към БД, организирана в таблици.

Видове заявки –

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**

**INSERT query**

# INSERT query

Пребиване

Структура

SQL

Търсене

Вмъкване

Експорт

Импорт

Права

Операции

Run SQL query/queries on table demo\_shop.category: ?

```
1 INSERT INTO `category`(`category_id`, `image`, `date_added`, `date_modified`) VALUES ([value-1],[value-2],[value-3],[value-4])
```



SELECT \*

SELECT

INSERT

UPDATE

DELETE

Изчистване

Формат

Get auto-saved query

☐ Bind parameters ?

Bookmark this SQL query

# INSERT query

Въвежда данни в таблица

Синтаксис -

**INSERT INTO** таблица (колони) **VALUES** (стойности)

Когато въвеждаме стринг – ‘стринг’.


Датата е стринг – ‘yyyy-mm-dd’

*Стойностите в колона `id` обикновено се генерират автоматично и не се налага да ги добавяме ръчно!*

# INSERT query

Във всяка колона задължително трябва да бъде посочена стойност за въвеждане!

Или - трябва да е зададена стойност по подразбиране или да е разрешено колоната да няма стойност при INSERT заявка.

сти По подразбиране  Колагия

NULL	▼
Няма	
Дефиниран като:	
NULL	
CURRENT_TIMESTAMP	



**SELECT query**

# SELECT query

Прелистване Структура **SQL** Търсене Вмъкване Експорт Импорт Права Операции Проследяване Тригери

Run SQL query/queries on table demo\_shop.category:

```
1 SELECT * FROM `category` WHERE 1
```

Колони

- category\_id
- image
- date\_added
- date\_modified

**SELECT \*** SELECT INSERT UPDATE DELETE Изчистване Формат

Get auto-saved query

☐ Bind parameters

Bookmark this SQL query:

Разделител  ☒ Показване на заявката отново ☐ Запазване отворен прозореца за заявки ☐ Rollback when finished ☒ Включване на проверките за външен ключ **Изпълнение**

# SELECT query

The screenshot shows a database management interface with a menu bar at the top containing: Прелистване, Структура, SQL, Търсене, Вмъкване, Експорт, Импорт, Права, Операции, Проследяване, and Тригери. The 'SQL' tab is selected and highlighted with a red circle.

Below the menu bar, a text box says: Run SQL query/queries on table demo\_shop.category: ⓘ

The main area contains a SQL query editor with the following text:

```
1 SELECT * FROM `category` WHERE 1
```

Below the query editor is a toolbar with buttons: SELECT \*, SELECT, INSERT, UPDATE, DELETE, Изчистване, and Формат. A red arrow points to the 'SELECT' button.

Below the toolbar is a button: Get auto-saved query

Below that is a checkbox: ☐ Bind parameters ⓘ

Below that is a text box: Bookmark this SQL query: [ ]

At the bottom, there is a row of checkboxes and a button:

Разделител [ ; ] ☒ Показване на заявката отново ☐ Запазване отворен прозореца за заявки ☐ Rollback when finished ☒ Включване на проверките за външен ключ **Изпълнение**

On the right side, there is a panel titled 'Колони' (Columns) containing a list of column names: category\_id, image, date\_added, and date\_modified.

# SELECT query

Извлича данни от таблица/и

Синтаксис -

**SELECT** поле1/, поле2, поле3.../ **FROM** таблица

**SELECT** \* **FROM** таблица

Полета – името на колоната, която избираме или \*/всички колони/

Таблица – името на таблицата, от която извличаме информация

# SELECT query - 2

Условия – започват с **WHERE** условие1 /AND условие 2 ...

/не е задължително да присъстват в заявката/

*Примери -*

да се селектират стойности, които са по-големи > ,

по-малки <

равни =

различни от число <> ,

различни от стринг <>

# SELECT query - 3

*Примери -*

да се селектират стойности, които са еквивалентни =  
започват със LIKE "string%"  
завършват със LIKE "%string",  
притежават стринг LIKE "%string%"

# SELECT query - 4

*Примери -*

да се селектират стойности, които са влизат в изброена група от стойности.

```
SELECT column-names FROM table-name WHERE column-name IN (values)
```

```
SELECT Id, CompanyName, City, Country FROM Supplier WHERE Country IN ('USA', 'UK',  
'Japan')
```

# SELECT query - 5

*Примери за няколко WHERE условия заедно*

```
SELECT * FROM customers WHERE favorite_website = 'techonthenet.com'  
AND customer_id > 6000
```

```
SELECT * FROM products WHERE product_name = 'Pear' OR product_name = 'Apple'
```



**UPDATE query**

# UPDATE query

Обновява записи в таблица

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

Не е задължително да променяме стойностите във всички колони.

Ако липсва **WHERE** условието - ще променим таблицата и във всички редове в указаните колони в резултат на заявката ще имаме еднакви стойности.

Ако има **WHERE** ще се променят само редовете, които отговарят на условието.

**DELETE query**

# DELETE query

Изтрива записи от таблица!!!!

синтаксис

**DELETE FROM** *table\_name* **WHERE** *condition*;

Препоръчително е първо да се провери WHERE условието дали е коректно със SELECT заявка.

При некоректно зададено WHERE условие е възможно да изтрием таблица, данни, които не желаем.

# DELETE query

Често в практиката се използва допълнително поле в таблиците -

**date\_deleted**, със стойност по подразбиране NULL.

В скрипта записа не се изтрива с DELETE заявка, а се променя стойността на **date\_deleted**, като се записва момента на изпълнение на заявката, която е UPDATE. **Записа се “скрива” и впоследствие се изтрива от администратор, при определени условия.**

Със SELECT се вадят само записите, за които `date_deleted = NULL`.

# Връзка между РНР и БД

# Връзка между PHP и БД

## ресурси

Свързва PHP кода ни (от папка htdocs/) с база данни (в localhost/phpmyadmin)

## Видове връзки

MySQLi (обектно – ориентиран вариант)

**MySQLi(процедурен вариант)**

**PDO (PHP Data Objects)**

# Връзка между PHP и БД - 2

Създаване на връзката –

```
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

```
$conn = mysqli_connect('localhost', 'root', '', 'products');
```



# Връзка между PHP и БД - 3

*Проверка дали е реализирана връзката*

```
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
} else {  
    echo "Connected successfully !";  
}
```

# Връзка между PHP и БД - 4

**mysqli\_query()** - изпълнява заявка

**mysqli\_fetch\_assoc()** - итерира резултат (минава през редовете на таблицата един по един)

# Връзка между PHP и БД - 4

```
$q = "SELECT * FROM products";  
$result = mysqli_query($conn, $q);  
if (mysqli_num_rows($result) > 0) {  
    while($row = mysqli_fetch_assoc($result)){  
        var_dump($row);  
        //TO BE CONTINUED  
    }  
}
```

} //**\$row** – съвпада с ред от таблицата и е асоциативен масив.

//Цялата таблица в БД е двумерен масив – съвкупност от n-брой **\$row**-масиви

Сламки 😊



# Досадни грешки

- Датата е стринг! Подавайте я във формат 'уууу-mm-dd' при запис в БД
- Ключовете на масивите, които 'вадите' от БД /имената на колоните на таблиците в БД/ взимайте с copy&paste – много често ги преписваме грешно.
- Не изпускайте кавичките на ключовете!!!

/to be continued .../