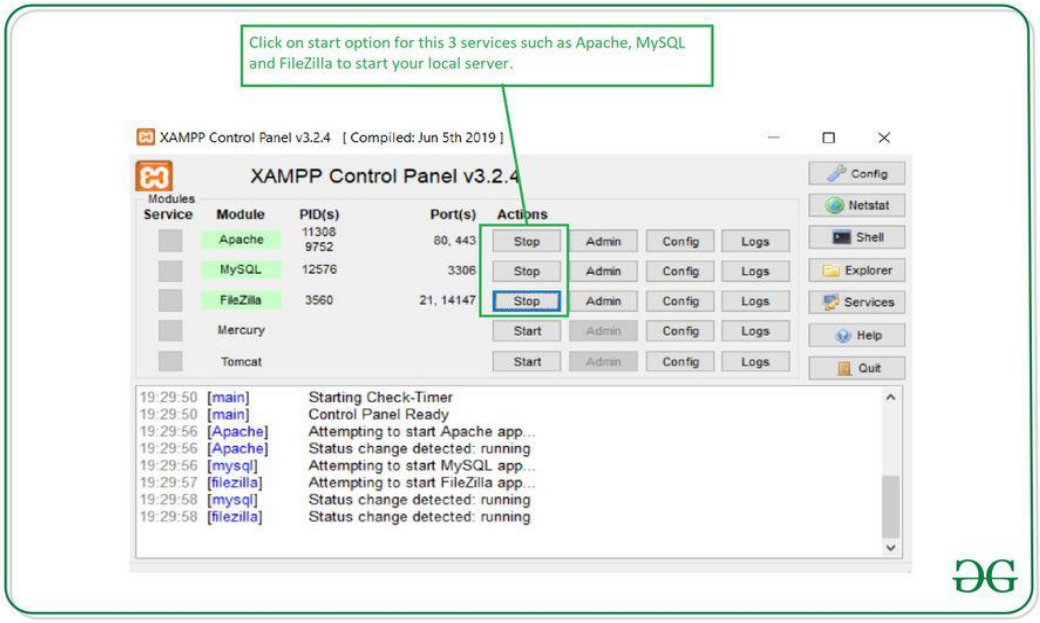
Introduction to PHPMyAdmin

Definition

phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

Creating Database Using PHPMyAdmin

Step 1: Start your XAMPP server as shown in below screen

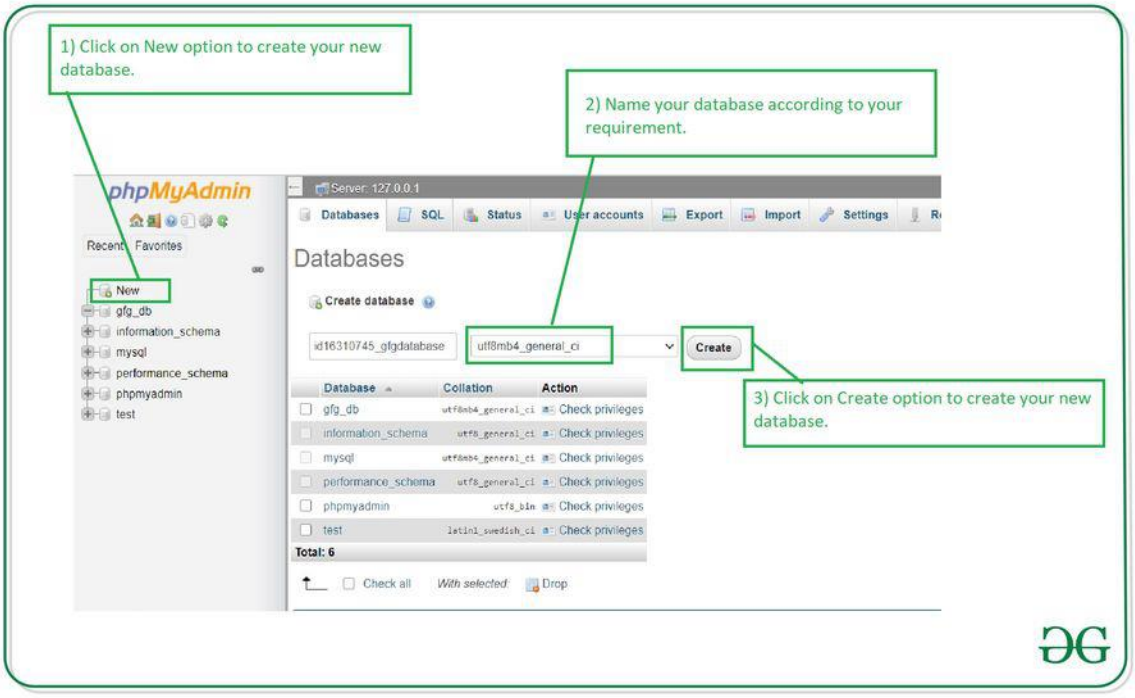


For starting your server on Xampp, we simply have to start the services in the Xampp console which is shown on above screenshot.

Step 2: Creating a new database

Visit the URL: ***localhost/phpmyadmin/***

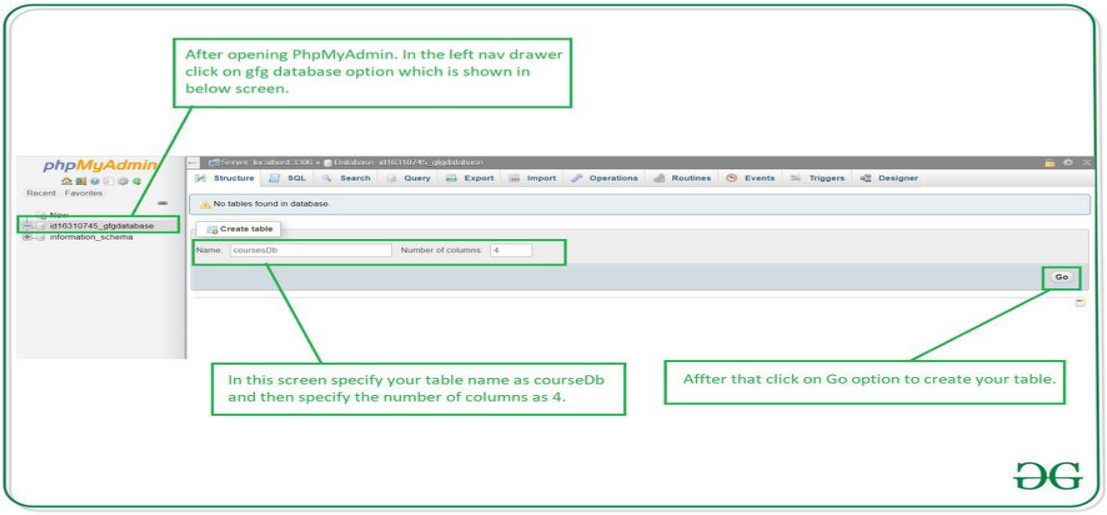
Click on New option which is shown in below screen. After that we have to enter our database name and then click on Create option to **create a new** database.



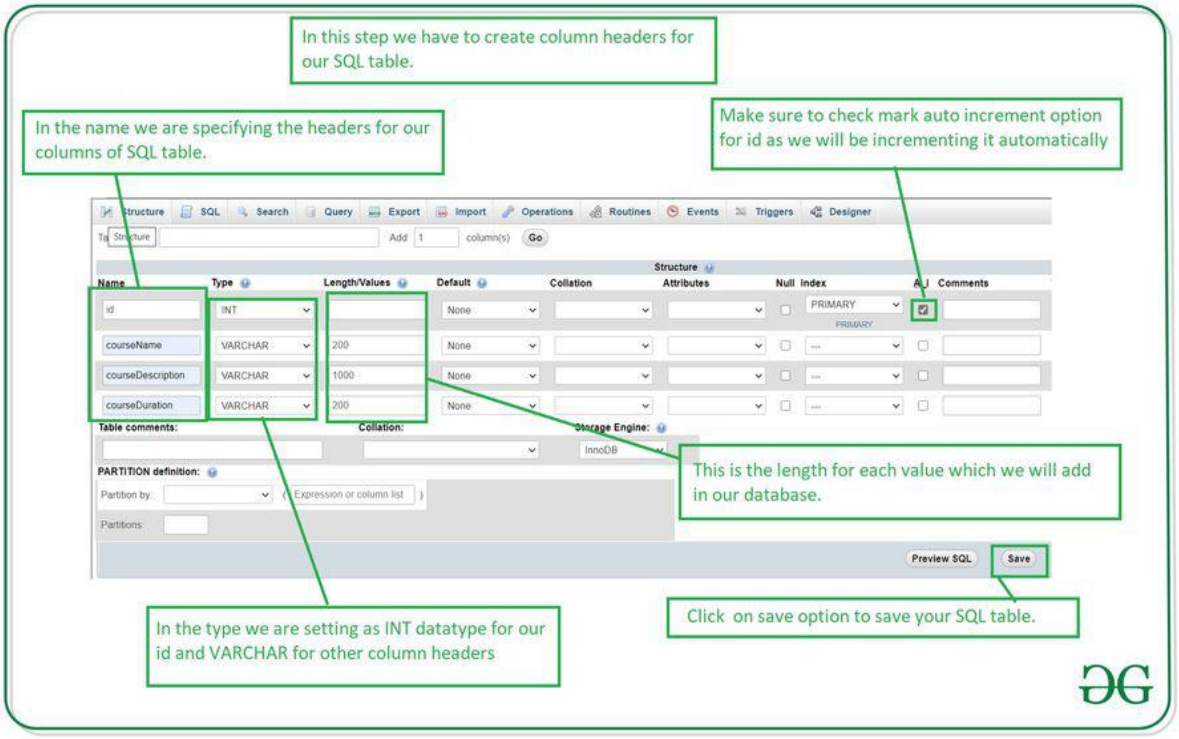
After creating a new database we have to move forward for creating a **new SQL table.**

Step 3: Creating a new table for storing our data

After creating your database, see below screen. In this screen we will be creating a new table. For creating a new table specify your table name and number of columns. In our case we are specifying our table name as **courseDb** and specify the number of columns as **4**.



After clicking on Go option your table will be created. Now we have to add column heading for our SQLite table. The 4 different columns are as follows. The 1st column is for **id**, 2nd column is for **courseName**, 3rd column is for **courseDescription** and 4th column is for **courseDuration.** For adding this column heading you will get to see below screen.



Steps to Create the SQL table:

1. Enter the name for your columns in name section which is shown in above screen.

2. After that enter the datatype of the data which we will be adding in our database. In this we are specifying **INT** for our ID as our ID will be an integer. And in SQL table **VARCHAR** is used for storing strings.

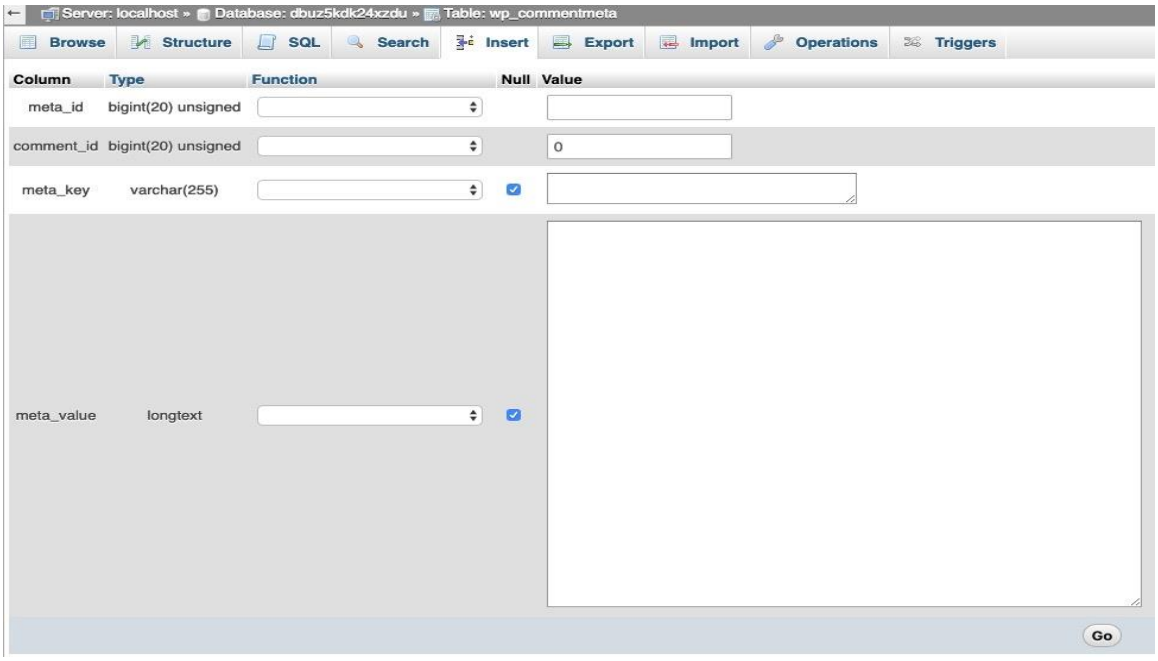
3. In the length/ values section we have to specify the maximum length of the data which we will be adding in our table. We will not be specifying length for our ID because it is added automatically.

4. After that we have to auto increment our ID. For this we have to check the box which is shown in above screenshot so that it will be incremented automatically on adding new data.

Now our database has been created successfully.

Insert Data in SQL Table Using PHPMyAdmin

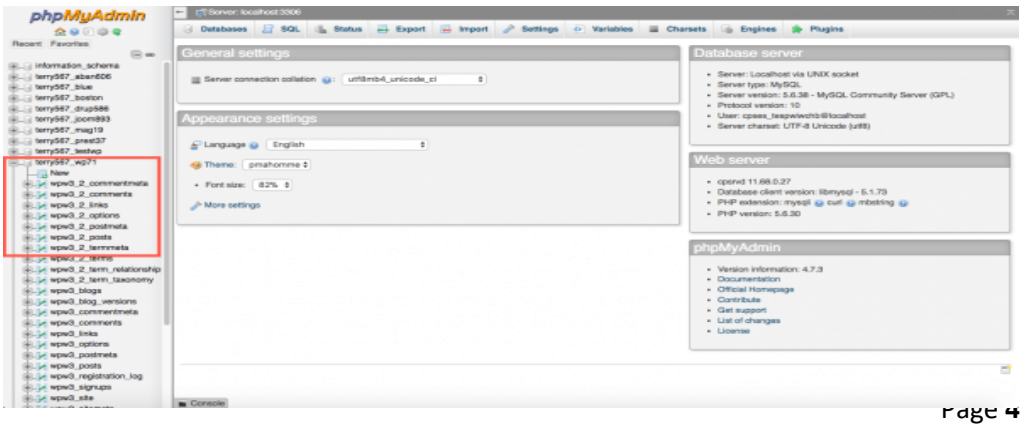
To add records inside a database table, open the table with phpMyAdmin and click on the **Insert** tab.



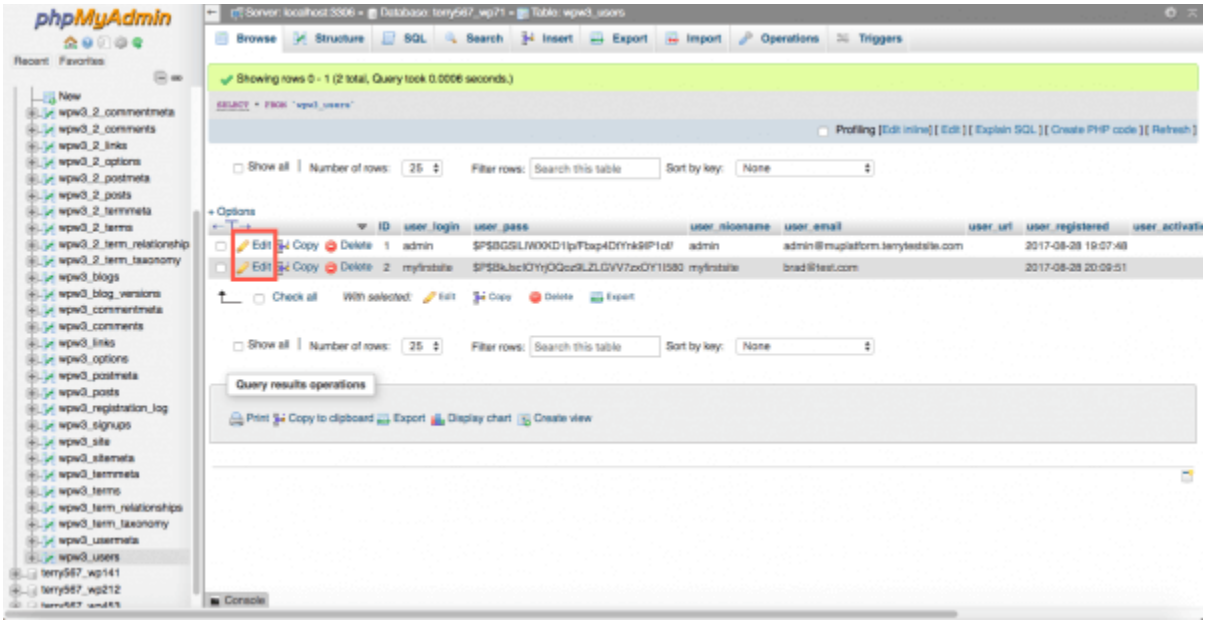
Enter the desired data in the corresponding fields and click on **Go** to store it. You can see the newly inserted record by clicking on the **Browse** tab.

Update SQL Table Data Using PHPMyAdmin

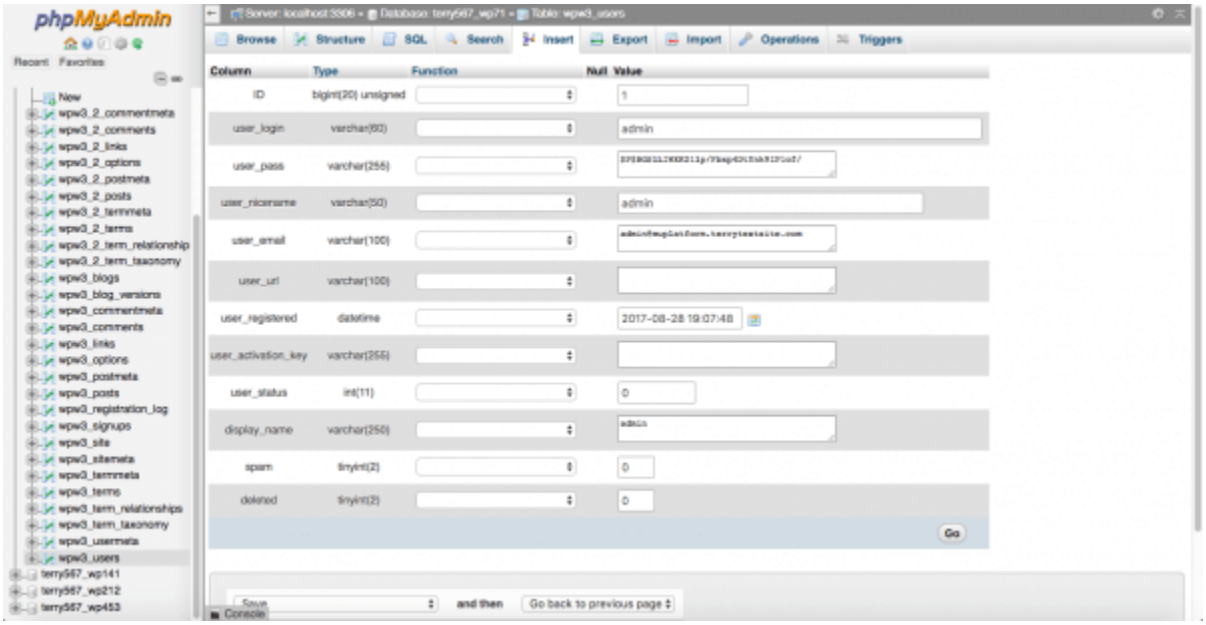
1. Use the navigation tree in the left sidebar to locate the database table you wish to modify.



2. The database fields are displayed in the right pane. Use the **Edit** links in each field listing to modify an individual field.

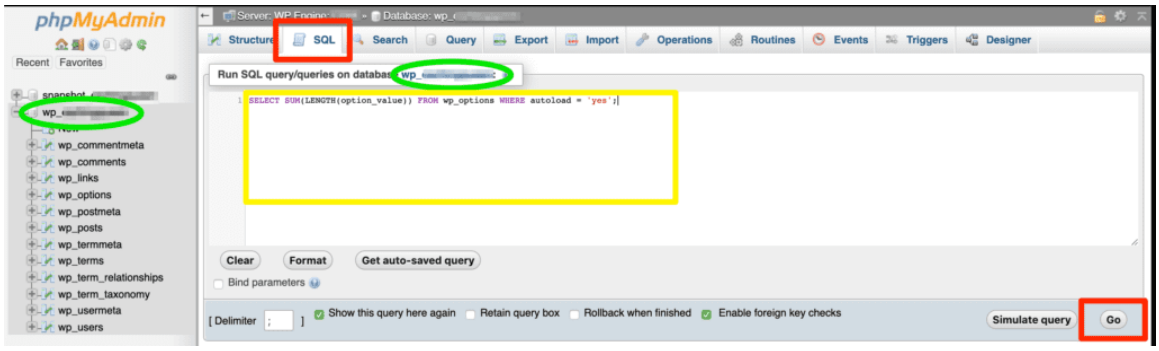


3. You can now proceed to modify the data within the field.



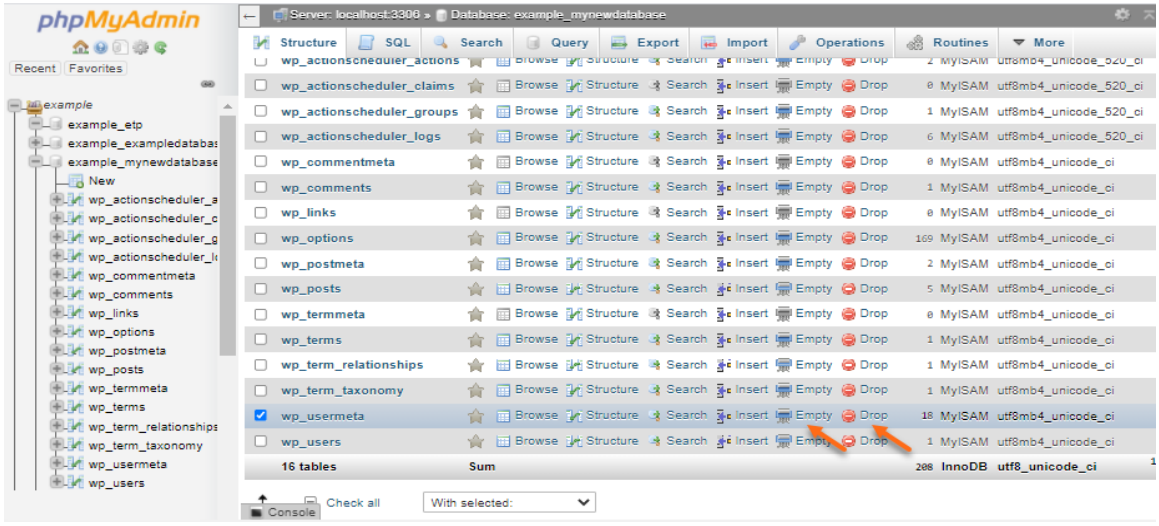
Run Query in phpMyAdmin (Select Statemnets)

1. Select the database you'd like to run a query on
2. Select **SQL** Tab
3. Write or paste your query in the text field
4. Click **GO**



Delete SQL Table Data Using PHPMyAdmin

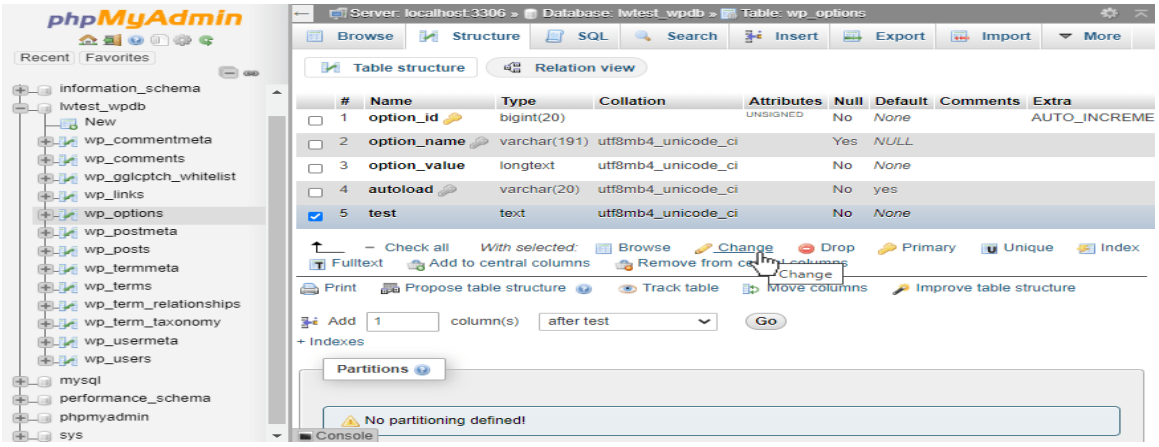
- Select **Drop** from the list if you wish the table to be completely gone.
- If you want to clear the data in the table and keep its structure, select **Empty**.



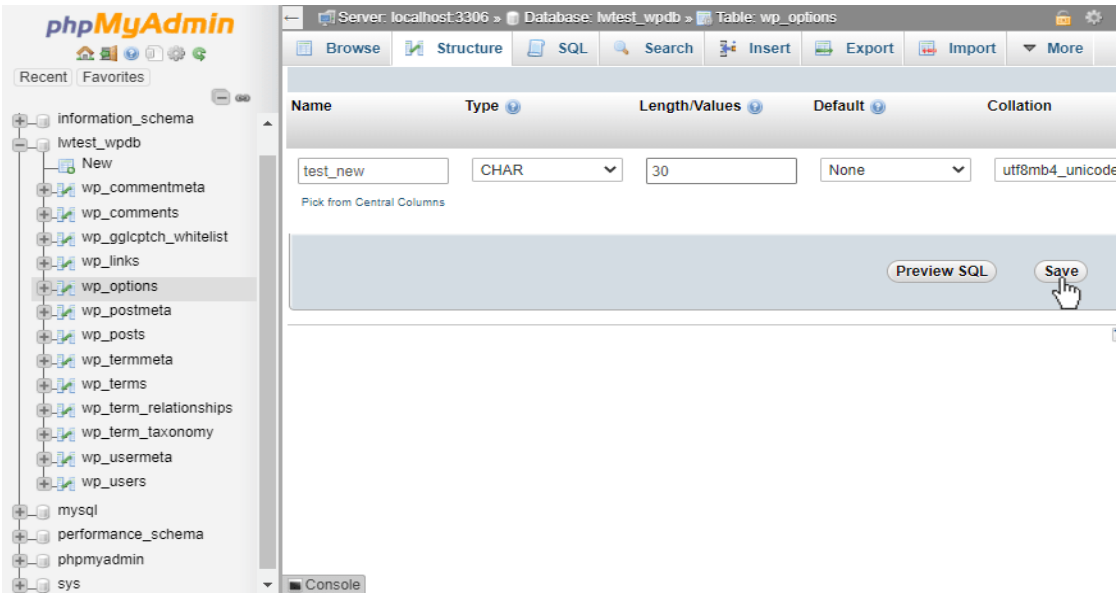
Alter SQL Table Using PHPMyAdmin

- Click on the name of the database where the table resides.
- Click on the table name.
- Click on the Structure tab.

On the Structure page, it lists the existing column fields and the attributes each one is currently set to use. To begin the modification, **check the box next** to the column name of one or more fields that should be modified.



- Then, click on the Change option link below the list of tables, and start changing the column data.



- Change whatever you want to change.

- Finally, Click Save Button.

Basic SQL Queries

1. Creating Database

```
mysql> CREATE DATABASE BAU;
```

2. Selecting Database

Creating a database does not select it for use; you must do that explicitly. To make BAU the current database, use this statement:

```
mysql> USE BAU;
```

3. Displaying Database

To display the created database, use the next command:

```
mysql> SHOW BAU;
```

4. Creating Tables

To create table in BAU database called student

```
mysql> CREATE TABLE student (ID int PRIMARY KEY, name VARCHAR(20)
NOT NULL, country VARCHAR(20), sex CHAR(1) );
```

5. Insert Data into Table

- **To insert data to all columns, the order as in table structure is required**

```
mysql> INSERT INTO student
       VALUES (100,'Diana','Zarqa','f');
```

- **If the country not defined yet, you can fill with null.**

```
mysql> INSERT INTO student
      VALUES (100,'Diana',Null,'f');
```

- **To insert data only to a specific columns in the table not to all**

```
mysql> INSERT INTO student (ID,name)
    VALUES (100,'Diana');
```

- **Be aware of the order, and the varchar data should be in ' '**

```
mysql> INSERT INTO student (name, ID)
       VALUES ('Sarah',200);
```

6. Update Data into Table

```
mysql> UPDATE Student
SET country = 'Amman', sex = 'f'
WHERE ID = 200;
```

- You should use the where clause to determine the required records, if not used all data in the table will be modified.

7. Select Data from Table

```
SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy;
```

1. To retrieve everything from a table

```
mysql> SELECT * FROM pet;
```

```
+----------+--------+---------+------+------------+------------+
| name     | owner  | species | sex  | birth      | death      |
+----------+--------+---------+------+------------+------------+
| Fluffy   | Harold | cat     | f    | 1993-02-04 | NULL       |
| Claws    | Gwen   | cat     | m    | 1994-03-17 | NULL       |
| Buffy    | Harold | dog     | f    | 1989-05-13 | NULL       |
| Fang     | Benny  | dog     | m    | 1990-08-27 | NULL       |
| Bowser   | Diane  | dog     | m    | 1979-08-31 | 1995-07-29 |
| Chirpy   | Gwen   | bird    | f    | 1998-09-11 | NULL       |
| Whistler | Gwen   | bird    | NULL | 1997-12-09 | NULL       |
| Slim     | Benny  | snake   | m    | 1996-04-29 | NULL       |
| Puffball | Diane  | hamster | f    | 1999-03-30 | NULL       |
+----------+--------+---------+------+------------+------------+
```

2. Selecting Particular Rows

```
mysql> SELECT * FROM pet WHERE name = 'Bowser';
```

```
+--------+-------+---------+------+------------+------------+
| name   | owner | species | sex  | birth      | death      |
+--------+-------+---------+------+------------+------------+
| Bowser | Diane | dog     | m    | 1989-08-31 | 1995-07-29 |
+--------+-------+---------+------+------------+------------+
```

```
mysql> SELECT * FROM pet WHERE birth >= '1998-1-1';
```

```
+----------+-------+---------+------+------------+-------+
| name     | owner | species | sex  | birth      | death |
+----------+-------+---------+------+------------+-------+
| Chirpy   | Gwen  | bird    | f    | 1998-09-11 | NULL  |
| Puffball | Diane | hamster | f    | 1999-03-30 | NULL  |
+----------+-------+---------+------+------------+-------+
```

```
mysql> SELECT * FROM pet WHERE species = 'snake' OR species = 'bird';
+----------+-------+---------+------+------------+-------+
| name     | owner | species | sex  | birth      | death |
+----------+-------+---------+------+------------+-------+
| Chirpy   | Gwen  | bird    | f    | 1998-09-11 | NULL  |
| Whistler | Gwen  | bird    | NULL | 1997-12-09 | NULL  |
| Slim     | Benny | snake   | m    | 1996-04-29 | NULL  |
+----------+-------+---------+------+------------+-------+
```

```
mysql> SELECT * FROM student WHERE ID BETWEEN 100 AND 300;
```

Example2:

```
mysql> SELECT ID, Name
    -> FROM Employeedetails
    -> WHERE ID BETWEEN 4 AND 7;
```

This will result in the following:

```
+----+---------+
| ID | Name    |
+----+---------+
|  4 | Rahul   |
|  5 | Arun    |
|  6 | Barbara |
|  7 | Suchi   |
+----+---------+
4 rows in set (0.04 sec)
```

3. Selecting Particular Columns

```
mysql> SELECT name, birth FROM pet;
```

```
+----------+------------+
| name     | birth      |
+----------+------------+
| Fluffy   | 1993-02-04 |
| Claws    | 1994-03-17 |
| Buffy    | 1989-05-13 |
| Fang     | 1990-08-27 |
| Bowser   | 1989-08-31 |
| Chirpy   | 1998-09-11 |
| Whistler | 1997-12-09 |
| Slim     | 1996-04-29 |
| Puffball | 1999-03-30 |
+----------+------------+
```

```
mysql> SELECT owner FROM pet;
+--------+
| owner  |
+--------+
| Harold |
| Gwen   |
| Harold |
| Benny  |
| Diane  |
| Gwen   |
| Gwen   |
| Benny  |
| Diane  |
+--------+
```

```
mysql> SELECT DISTINCT owner FROM pet;
+--------+
| owner  |
+--------+
| Benny  |
| Diane  |
| Gwen   |
| Harold |
+--------+
```

```
mysql> SELECT name, species, birth FROM pet
       WHERE species = 'dog' OR species = 'cat';
+--------+---------+------------+
| name   | species | birth      |
+--------+---------+------------+
| Fluffy | cat     | 1993-02-04 |
| Claws  | cat     | 1994-03-17 |
| Buffy  | dog     | 1989-05-13 |
| Fang   | dog     | 1990-08-27 |
| Bowser | dog     | 1989-08-31 |
+--------+---------+------------+
```

8. Delete Data from Table

- **Delete a particular or specific number of rows using where condition**

```
DELETE FROM pet WHERE name = 'Fang'
```

- **To delete all records in the table**

  ```
  DELETE * FROM pet;
  ```

  Or

  Use TRUNCATE TABLE it is similar to a DELETE statement that deletes all rows.

  ```
  Truncate table pet;
  ```