Study Unit 3: WEB-BASED DATABASE DEVELOPMENT

Outline

- What Can MySQL Do?
- MYSQL Database Console (Localhost) GUI
- MySQL Basic Query

Learning Outcomes of Study Unit 3

Upon completion of this study unit, you should be able to

1.1 MySQL

- ○ What can MySQL do?
- ○ Using MySQL in your website
- ○ Semicolon after MySQL Statement?
- ○ Some of the Most Important MySQL Commands

1.2 MySQL Data Types (Version 8.0)

- Numeric Data Types
- Date and Time Types
- String Types
- Hardware

1.3 MySQL phpMyAdmin

- What is phpMyAdmin?
- How to access phpMyAdmin?
-

1.1 MySQL

1.1.1 What can MySQL do?

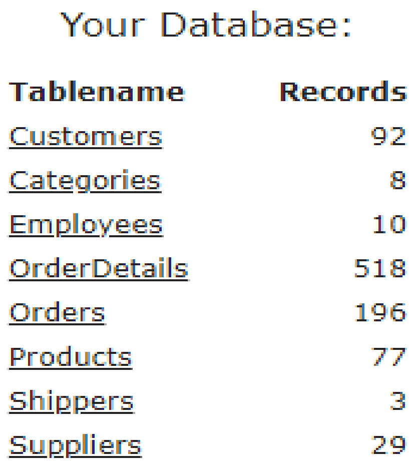
- MySQL can execute queries against a database
- MySQL can retrieve data from a database
- MySQL can insert records in a database
- MySQL can update records in a database
- MySQL can delete records from a database
- MySQL can create new databases
- MySQL can create new tables in a database
- MySQL can create stored procedures in a database
- MySQL can create views in a database
- MySQL can set permissions on tables, procedures, and views

1.1.2 Using MySQL in your website

To build a web site that shows data from a database, you will need:

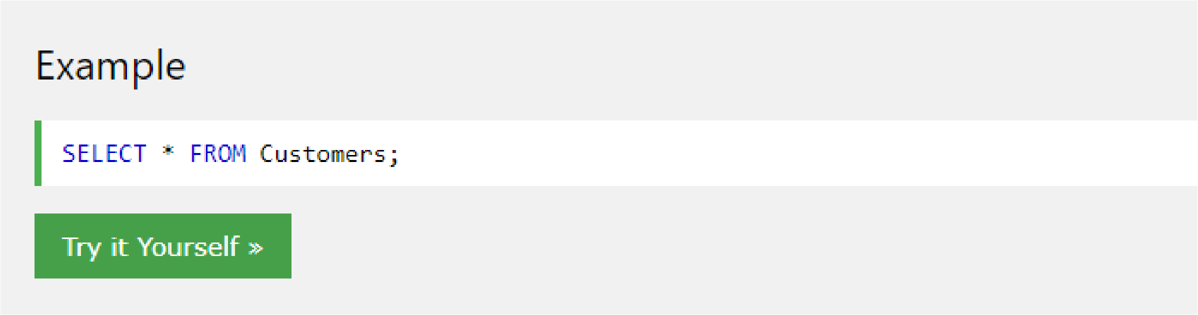
- An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
- To use a server-side scripting language, like PHP or ASP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

- RDBMS stands for Relational Database Management System.
- RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.
- Look at the "Customers" table:



Your Database:

Tablename	Records
Customers	92
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29



Example

```
SELECT * FROM Customers;
```

Note: The above Query will select all 92 recorde of the Customers table in a Database.

1.1.3 Semicolon after MySQL Statement?

Some database systems require a semicolon at the end of each MySQL statement.

Semicolon is the standard way to separate each MySQL statement in database systems that allow more than one MySQL statement to be executed in the same call to the server.

In this tutorial, we will use semicolon at the end of each MySQL statement.

1.1.4 Some of the Most Important MySQL Commands

- `SELECT` - extracts data from a database
- `UPDATE` - updates data in a database
- `DELETE` - deletes data from a database
- `INSERT INTO` - inserts new data into a database
- `CREATE DATABASE` - creates a new database
- `ALTER DATABASE` - modifies a database
- `CREATE TABLE` - creates a new table
- `ALTER TABLE` - modifies a table
- `DROP TABLE` - deletes a table
- `CREATE INDEX` - creates an index (search key)
- `DROP INDEX` - deletes an index

But in this Lecture we shall stick with four basic following commands:

- INSERT INTO - inserts new data into a database
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database

1.2 MySQL Data Types (Version 8.0)

In MySQL there are three main data types: string, numeric, and date and time.

Properly defining the fields in a table is important to the overall optimization of your database. You should use only the type and size of field you really need to use. For example, do not define a field 10 characters wide, if you know you are only going to use 2 characters. These type of fields (or columns) are also referred to as data types, after the **type of data** you will be storing in those fields.

MySQL uses many different data types broken into three categories −

- Numeric
- Date and Time
- String Types.

Let us now discuss them in detail.

Numeric Data Types

MySQL uses all the standard ANSI SQL numeric data types, so if you're coming to MySQL from a different database system, these definitions will look familiar to you.

The following list shows the common numeric data types and their descriptions −

- **INT** − A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295. You can specify a width of up to 11 digits.
- **TINYINT** − A very small integer that can be signed or unsigned. If signed, the allowable range is from -128 to 127. If unsigned, the allowable range is from 0 to 255. You can specify a width of up to 4 digits.
- **SMALLINT** − A small integer that can be signed or unsigned. If signed, the allowable range is from -32768 to 32767. If unsigned, the allowable range is from 0 to 65535. You can specify a width of up to 5 digits.

- **MEDIUMINT** − A medium-sized integer that can be signed or unsigned. If signed, the allowable range is from -8388608 to 8388607. If unsigned, the allowable range is from 0 to 16777215. You can specify a width of up to 9 digits.
- **BIGINT** − A large integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned, the allowable range is from 0 to 18446744073709551615. You can specify a width of up to 20 digits.
- **FLOAT(M,D)** − A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.
- **DOUBLE(M,D)** − A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.
- **DECIMAL(M,D)** − An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.

Date and Time Types

The MySQL date and time datatypes are as follows −

- **DATE** − A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.
- **DATETIME** − A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.
- **TIMESTAMP** − A timestamp between midnight, January 1st, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).
- **TIME** − Stores the time in a HH:MM:SS format.
- **YEAR(M)** − Stores a year in a 2-digit or a 4-digit format. If the length is specified as 2 (for example YEAR(2)), YEAR can be between 1970 to 2069 (70 to 69). If the length is specified as 4, then YEAR can be 1901 to 2155. The default length is 4.

String Types

Although the numeric and date types are fun, most data you'll store will be in a string format. This list describes the common string datatypes in MySQL.

- **CHAR(M)** − A fixed-length string between 1 and 255 characters in length (for example CHAR(5)), right-padded with spaces to the specified length when stored. Defining a length is not required, but the default is 1.
- **VARCHAR(M)** − A variable-length string between 1 and 255 characters in length. For example, VARCHAR(25). You must define a length when creating a VARCHAR field.
- **BLOB or TEXT** − A field with a maximum length of 65535 characters. BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data. The difference between the two is that the sorts and comparisons on the stored data are **case sensitive** on BLOBs and are **not case sensitive** in TEXT fields. You do not specify a length with BLOB or TEXT.
- **TINYBLOB or TINYTEXT** − A BLOB or TEXT column with a maximum length of 255 characters. You do not specify a length with TINYBLOB or TINYTEXT.
- **MEDIUMBLOB or MEDIUMTEXT** − A BLOB or TEXT column with a maximum length of 16777215 characters. You do not specify a length with MEDIUMBLOB or MEDIUMTEXT.
- **LONGBLOB or LONGTEXT** − A BLOB or TEXT column with a maximum length of 4294967295 characters. You do not specify a length with LONGBLOB or LONGTEXT.
- **ENUM** − An enumeration, which is a fancy term for list. When defining an ENUM, you are creating a list of items from which the value must be selected (or it can be NULL). For example, if you wanted your field to contain "A" or "B" or "C", you would define your ENUM as ENUM ('A', 'B', 'C') and only those values (or NULL) could ever populate that field.

1.3 MySQL (phpMyAdmin?)

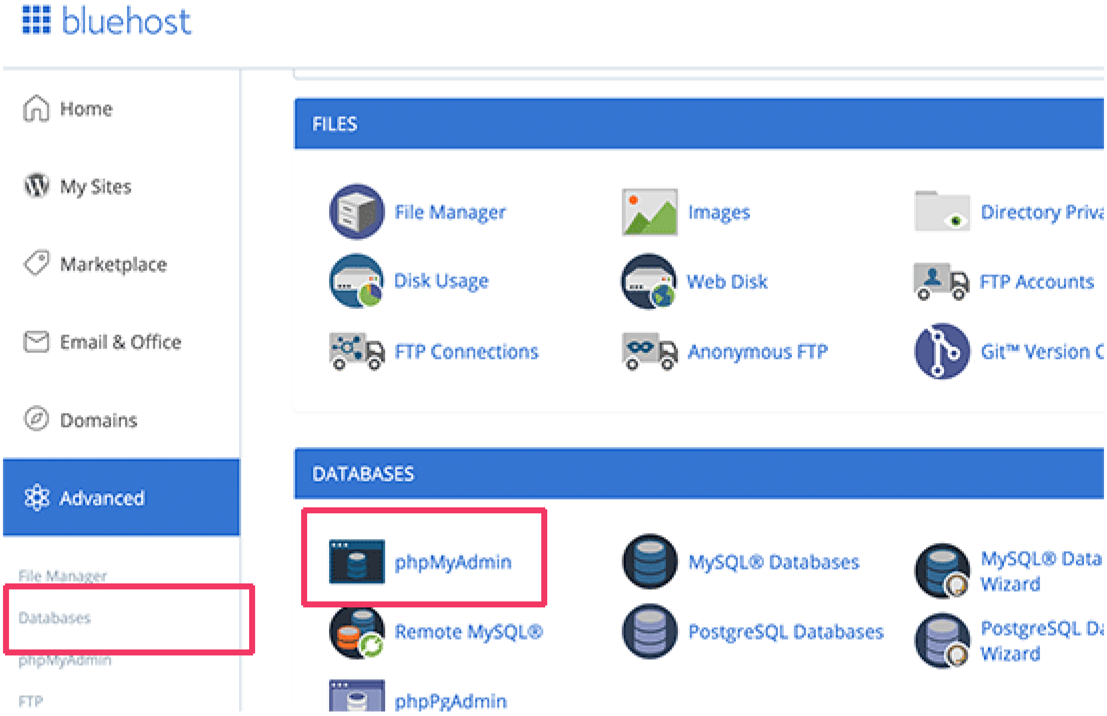
1.3.1 What is phpMyAdmin?

PhpMyAdmin is a web-based software that allows you to manage MySQL databases using your web browser. It offers an easy to use interface that allows you to run MySQL commands and database operations.

You can also use it to browse and edit database tables, rows, and fields. It also allows you to import, export, or delete all data inside a database.

1.3.2 How to access phpMyAdmin?

PhpMyAdmin comes pre-installed with all top WordPress hosting companies. You can find it under the Databases section of your hosting account’s cPanel dashboard. Below is an example screenshot from Bluehost control panel:



Depending on your hosting provider, your cPanel interface may look different than the above screenshot. You would still be able to find phpMyAdmin icon under the databases section.

Clicking on it will open the phpMyAdmin interface where you can select your WordPress database from the left column. After, that phpMyAdmin will display all tables in your WordPress database.

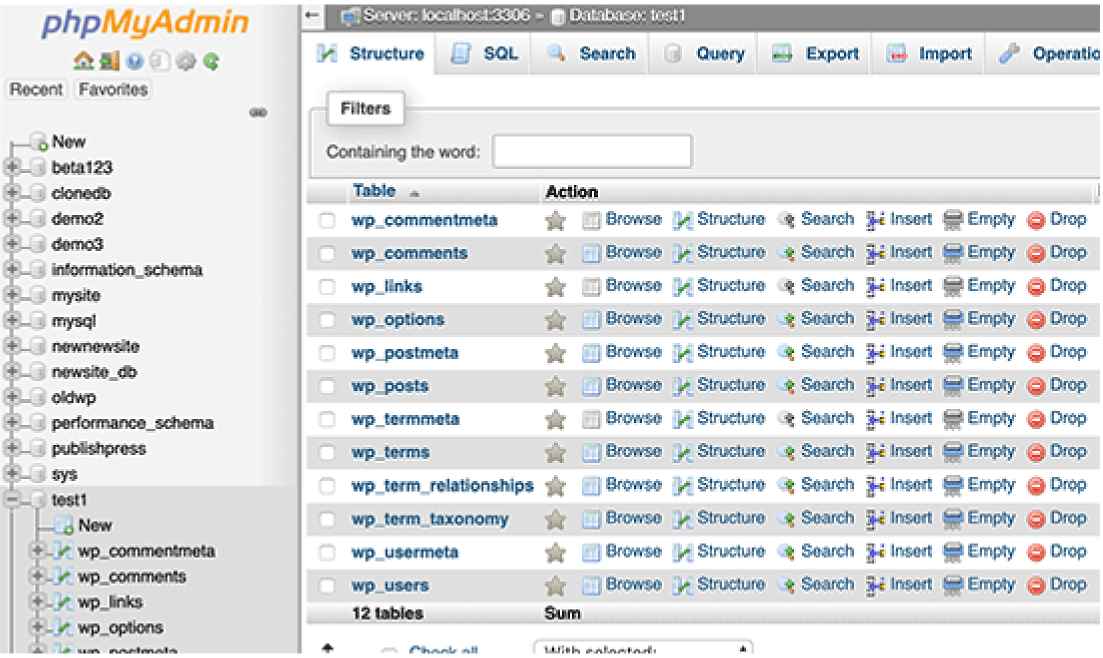
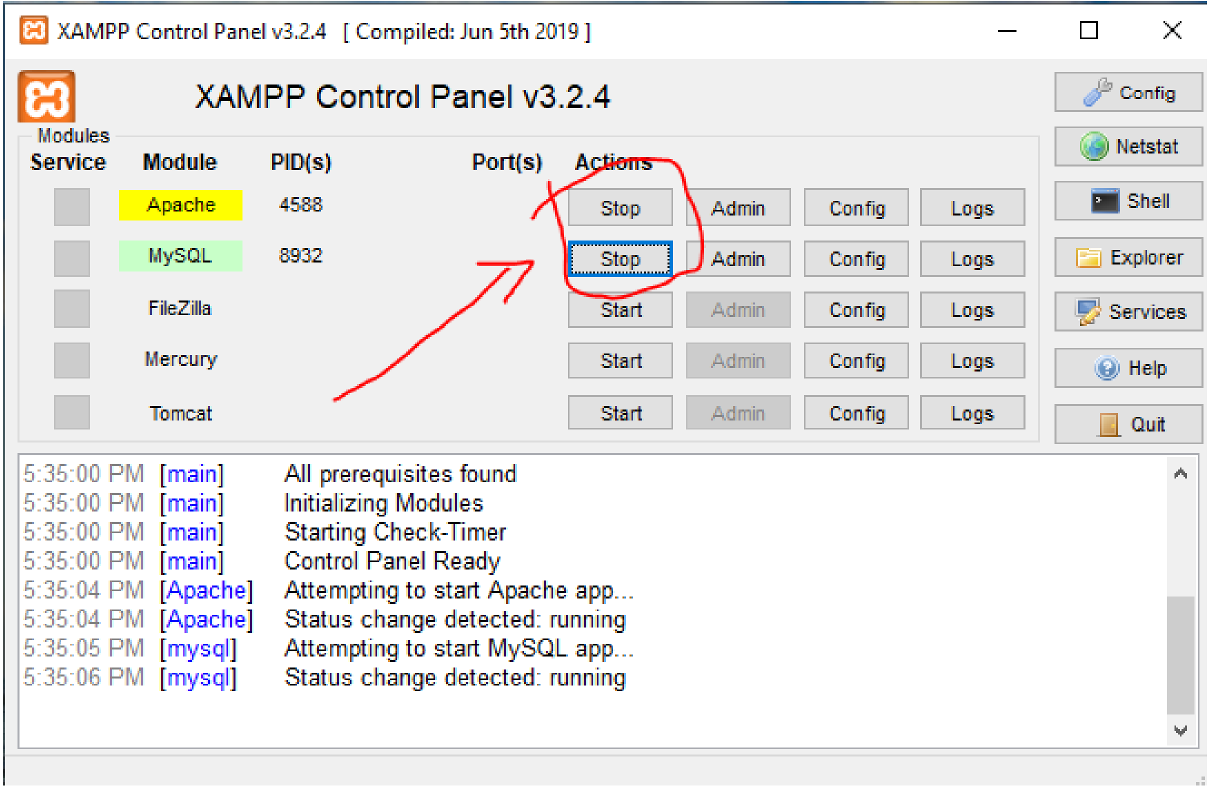


	Table ▲	Action						
□	wp_commentmeta	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_comments	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_links	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_options	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_postmeta	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_posts	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_termmeta	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_terms	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_term_relationships	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_term_taxonomy	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_usermeta	☆	Browse	Structure	Search	Insert	Empty	Drop
□	wp_users	☆	Browse	Structure	Search	Insert	Empty	Drop
	12 tables	Sum						

1.3.3 How to access phpMyAdmin on XAMPP?

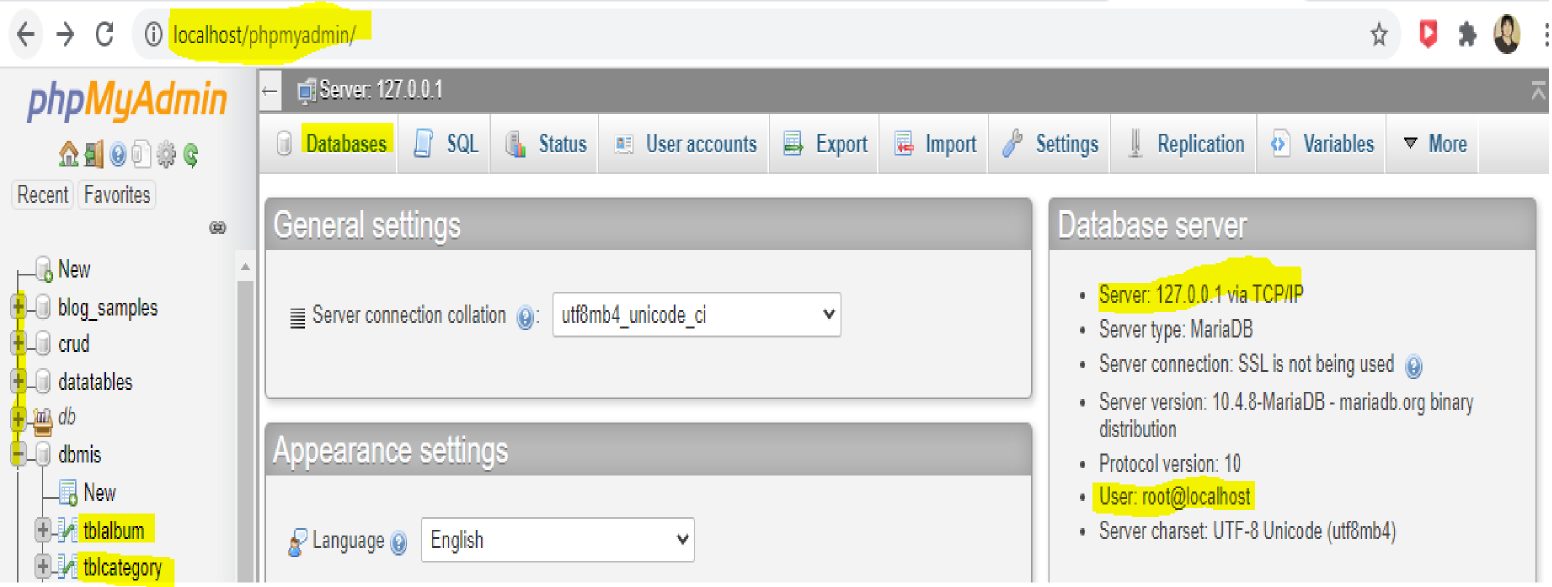
Start your XAMPP Server on your computer and Run APACHE and MYSQL services


XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	4588		Stop	Admin	Config	Logs
	MySQL	8932		Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
5:35:00 PM  [main]     All prerequisites found
5:35:00 PM  [main]     Initializing Modules
5:35:00 PM  [main]     Starting Check-Timer
5:35:00 PM  [main]     Control Panel Ready
5:35:04 PM  [Apache]   Attempting to start Apache app...
5:35:04 PM  [Apache]   Status change detected: running
5:35:05 PM  [mysql]    Attempting to start MySQL app...
5:35:06 PM  [mysql]    Status change detected: running
```

In Chrome Browser write http://localhost/phpmyadmin
To access databases.

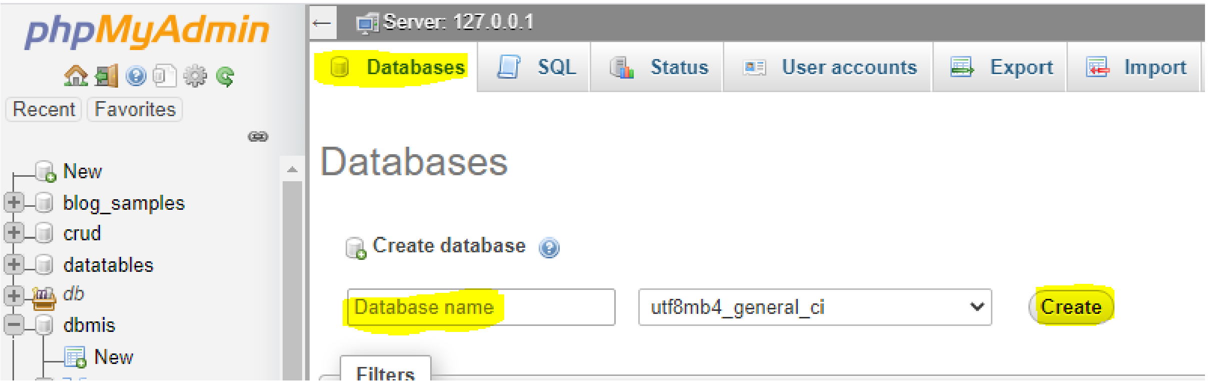


Server connection collation: utf8mb4_unicode_ci

Language: English

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.8-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Click on Databases tab than write Database name and click to create.



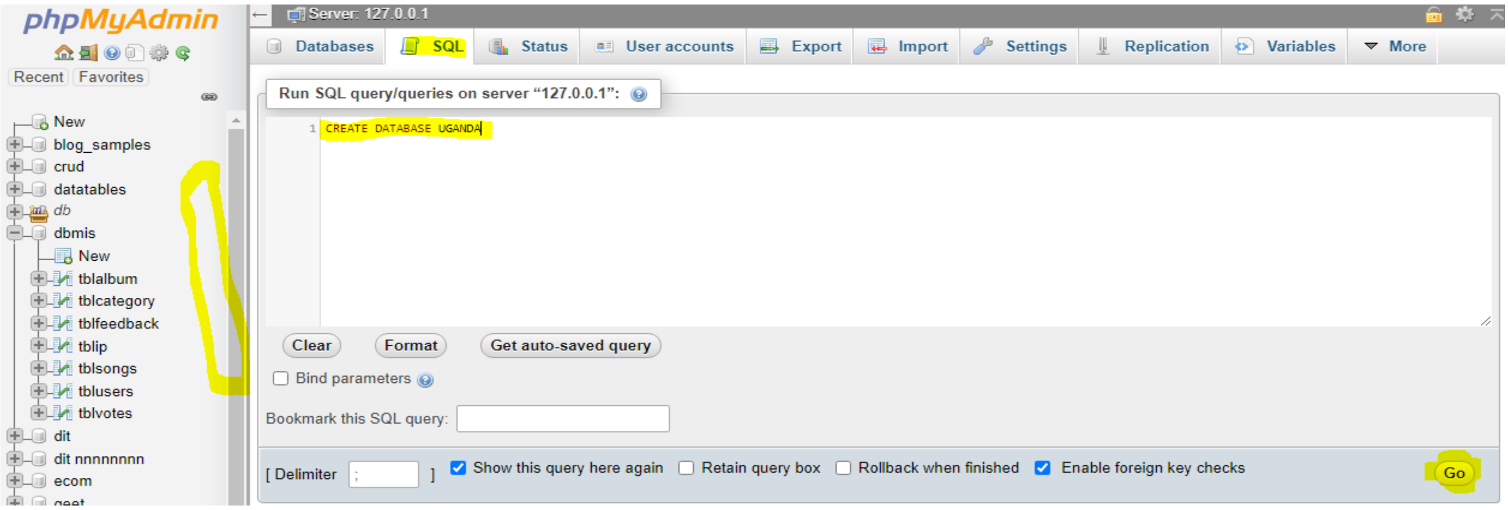
The SQL CREATE DATABASE Statement

The `CREATE DATABASE` statement is used to create a new SQL database.

Syntax

```
CREATE DATABASE databasename;
```

Create database Through command prompt click on SQL tab write a command
MYSQL = >   CREATE DATABASE KAMPALA



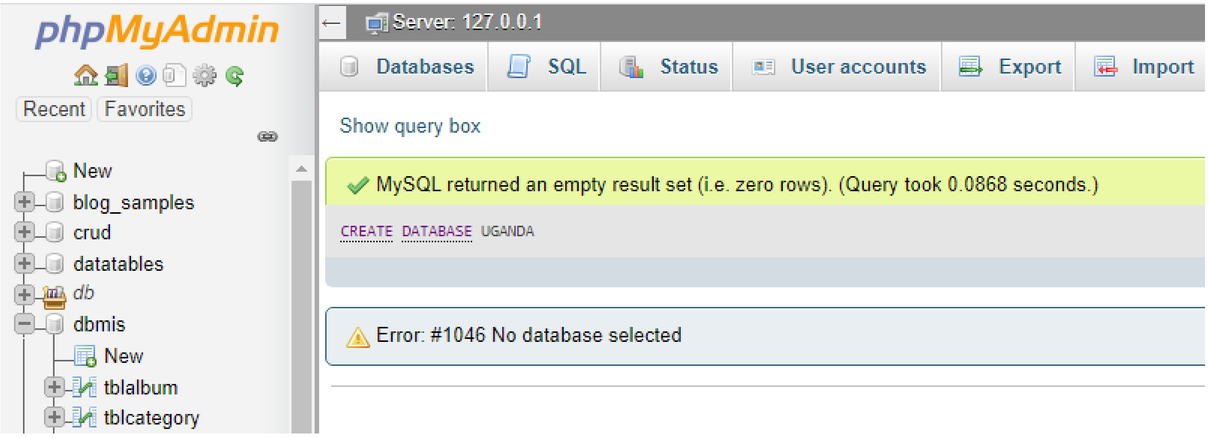
Run SQL query/queries on server "127.0.0.1":

```
CREATE DATABASE UGANDA
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0868 seconds.)

```
CREATE DATABASE UGANDA
```

⚠ Error: #1046 No database selected

1.4 DROP DATABASE

The SQL DROP DATABASE Statement

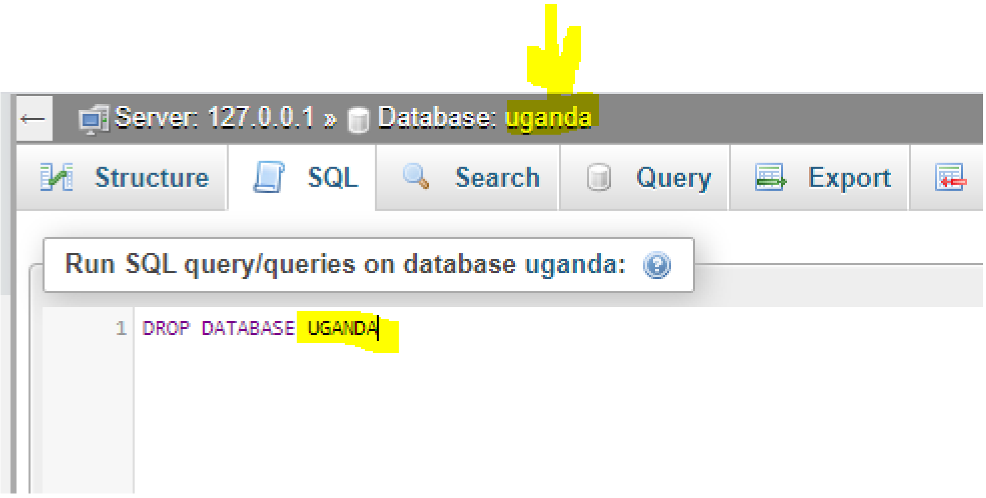
The `DROP DATABASE` statement is used to drop an existing SQL database.

Syntax

```
DROP DATABASE databasename;
```

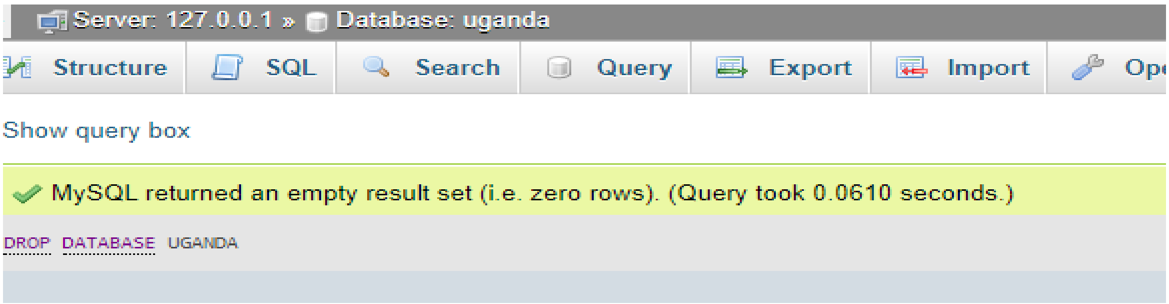
Note: Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!

Click on SQL tab and write Drop statement and click on go…..



←
Server: 127.0.0.1 » Database: uganda
Structure
SQL
Search
Query
Export
Run SQL query/queries on database uganda:

```
DROP DATABASE UGANDA
```



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0610 seconds.)

```
DROP DATABASE UGANDA
```

1.5 MySQL Backup (Export) Database Statement

select a Database like **crud** -> click on Export Tab

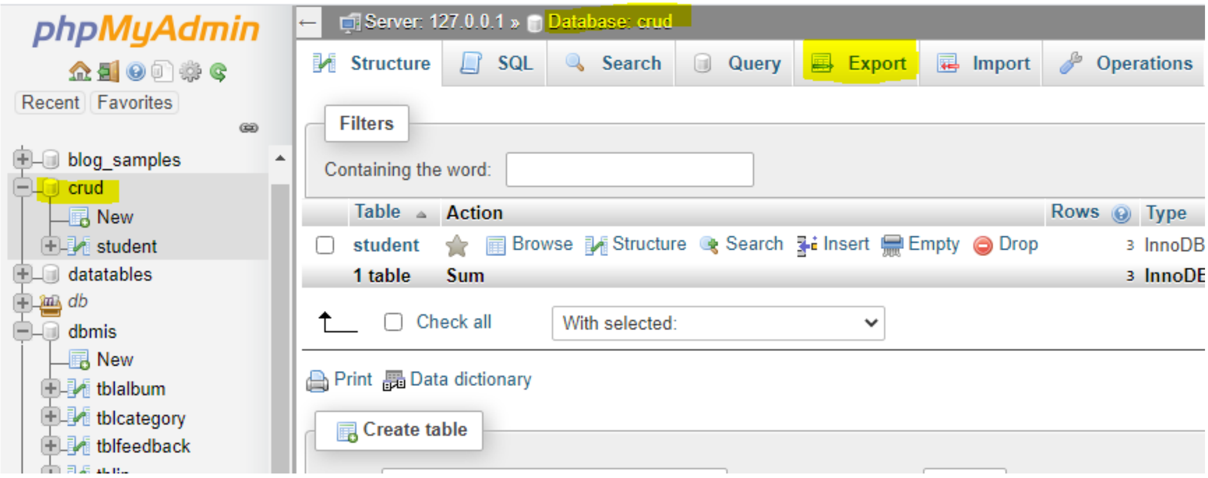
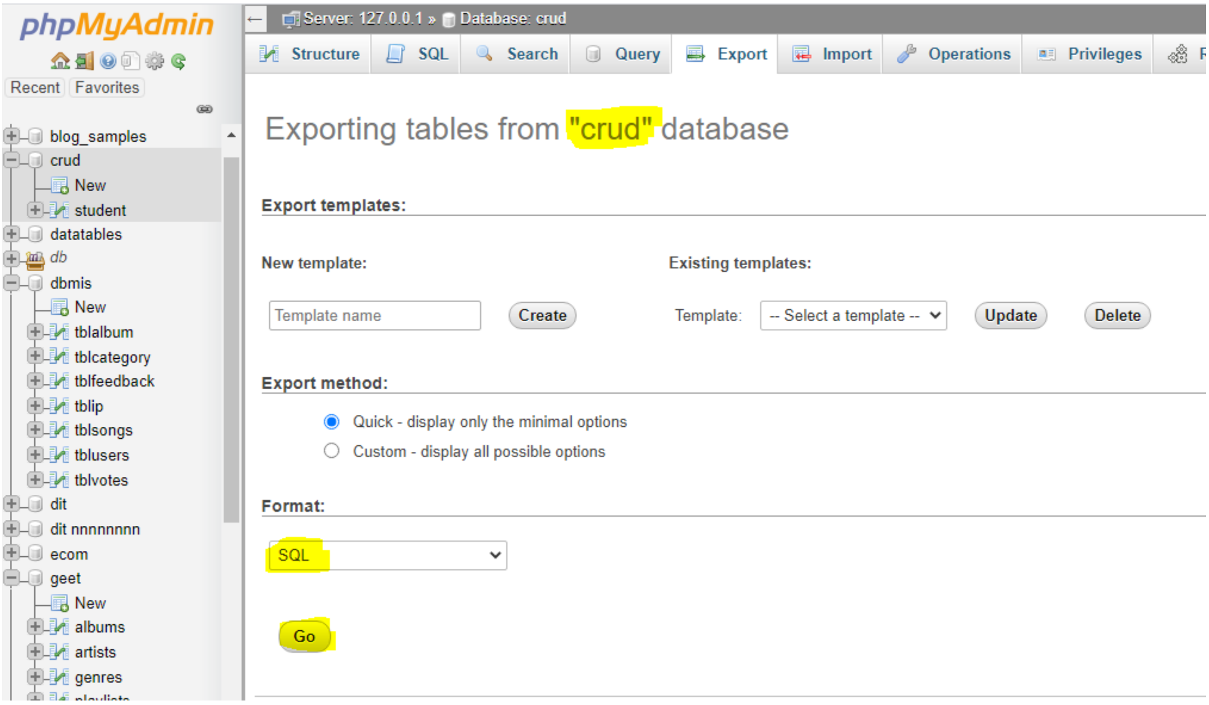


Table	Action	Rows	Type
student	Browse Structure Search Insert Empty Drop	3	InnoDB
1 table	Sum	3	InnoDB

Select option SQL and click on go



Exporting tables from "crud" database

Export templates:

New template:

Existing templates:

Export method:

● Quick - display only the minimal options

○ Custom - display all possible options

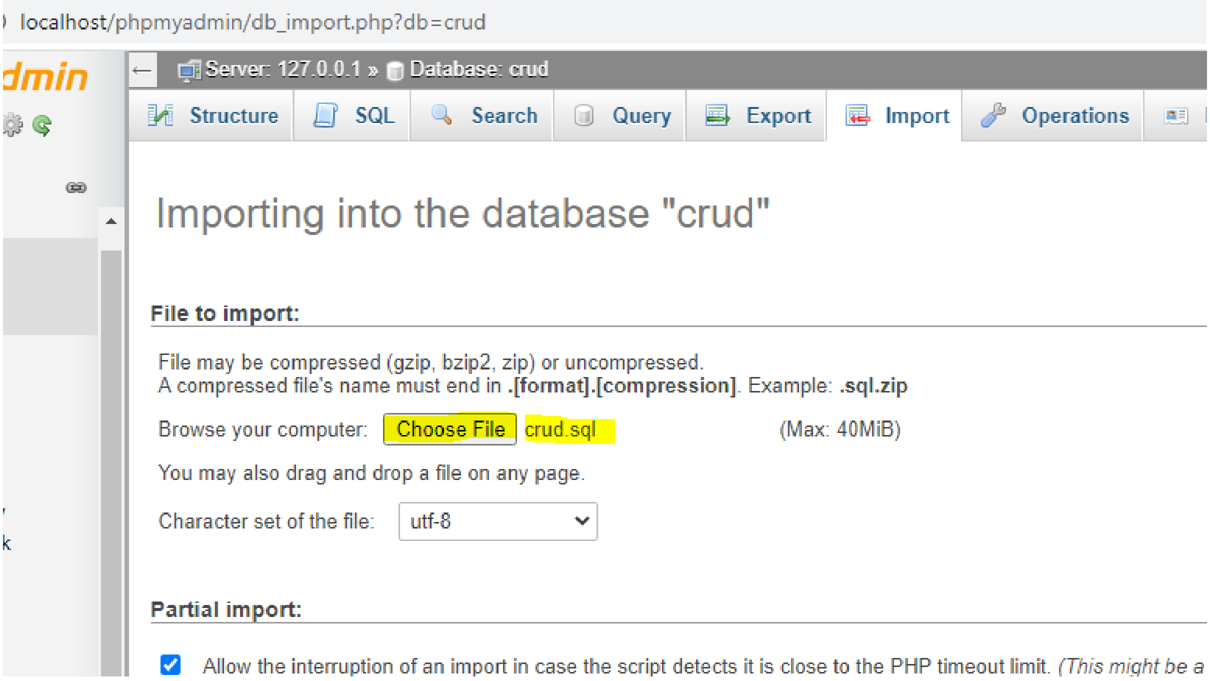
Format:

SQL

Your database with name like crud.sql will be download ...

1.6 MySQL Backup (Import) Database Statement

Click on Import Option and choose File ->Path of your existing DB and click on go to import DB file


localhost/phpmyadmin/db_import.php?db=crud
dmin
Server: 127.0.0.1 » Database: crud
Structure
SQL
Search
Query
Export
Import
Operations

Importing into the database "crud"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File crud.sql (Max: 40MiB)

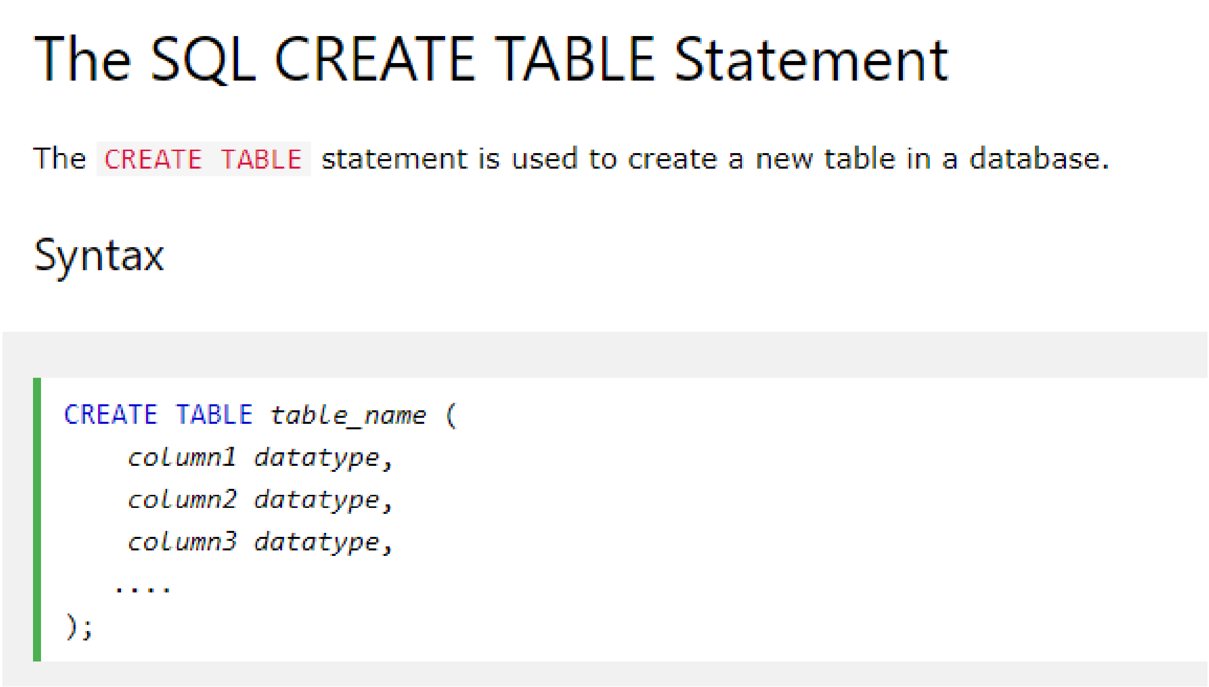
You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a*

1.7 MYSQL Create Table Statement



The SQL CREATE TABLE Statement

The `CREATE TABLE` statement is used to create a new table in a database.

Syntax

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

Tip: For an overview of the available data types, go to our complete Data Types Reference.

SQL CREATE TABLE Example

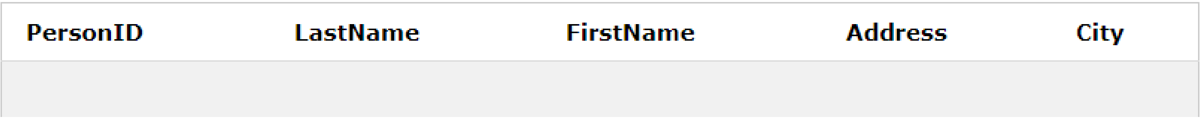
The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

The PersonID column is of type int and will hold an integer.

The LastName, FirstName, Address, and City columns are of type varchar and will hold characters, and the maximum length for these fields is 255 characters.

The empty "Persons" table will now look like this:



PersonID	LastName	FirstName	Address	City

Tip: The empty "Persons" table can now be filled with data with the SQL INSERT INTO statement.

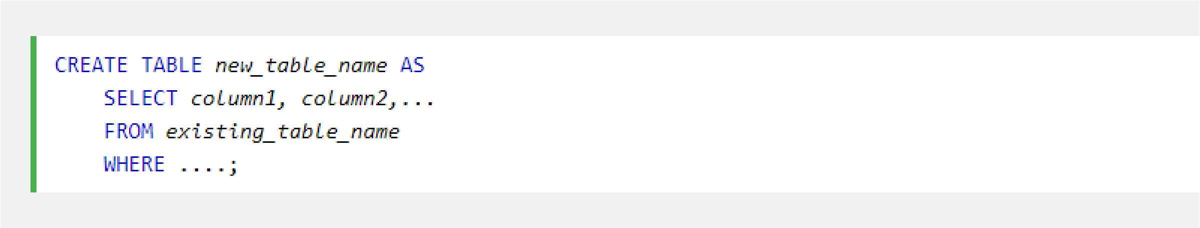
1.8 Create Table Using Another Table OR Copy a table

A copy of an existing table can also be created using `CREATE TABLE`.

The new table gets the same column definitions. All columns or specific columns can be selected.

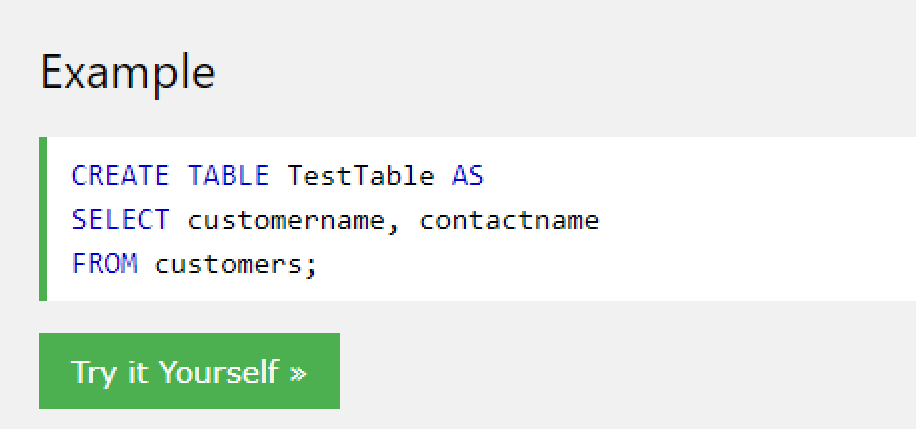
If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

Syntax



```
CREATE TABLE new_table_name AS
    SELECT column1, column2,...
    FROM existing_table_name
    WHERE ....;
```

The following SQL creates a new table called "TestTables" (which is a copy of the "Customers" table):



Example

```
CREATE TABLE TestTable AS
SELECT customername, contactname
FROM customers;
```

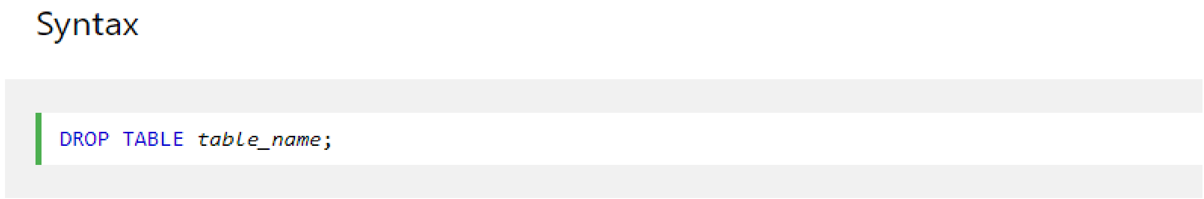
CREATE TABLE Duplicate AS

SELECT Address, City

FROM persons;

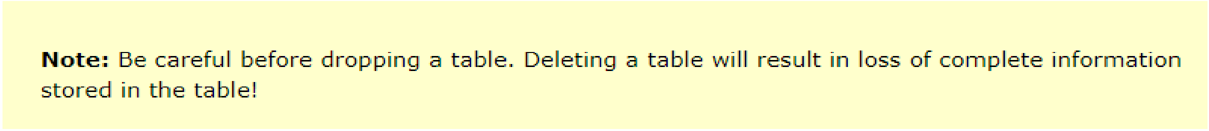
1.9 MYSQL Drop Table Statement

The DROP TABLE statement is used to drop an existing table in a database.



Syntax

```
DROP TABLE table_name;
```



Note: Be careful before dropping a table. Deleting a table will result in loss of complete information stored in the table!

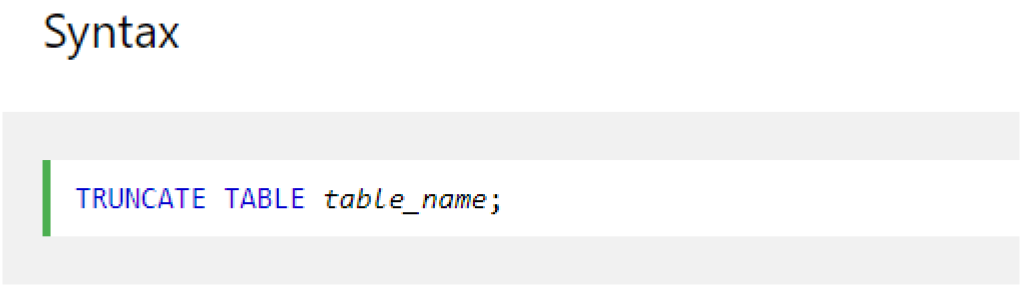
The following MySQL statement drops the existing table "Shippers":

```
DROP TABLE Duplicate;
```

SQL TRUNCATE TABLE

1.9 MYSQL Truncate Table Statement

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.



Syntax

```
TRUNCATE TABLE table_name;
```

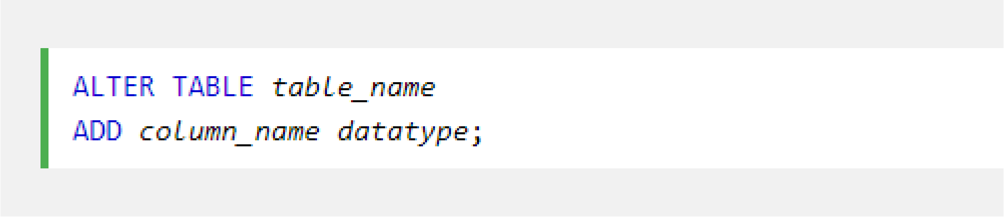
1.10 MYSQL Alter Table Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

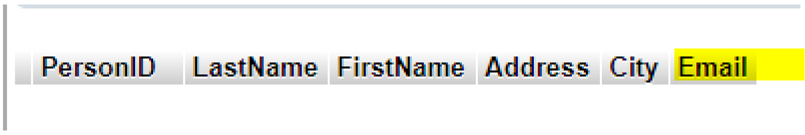


```
ALTER TABLE table_name
ADD column_name datatype;
```

EXAMPLE

```
ALTER TABLE persons
ADD Email varchar(255);
```

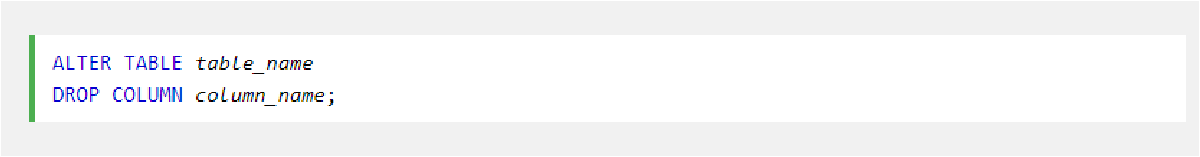
OUTPUT



PersonID	LastName	FirstName	Address	City	Email

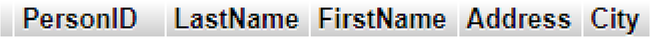
ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):



```
ALTER TABLE table_name
DROP COLUMN column_name;
```

```
ALTER TABLE persons
DROP COLUMN Email;
```

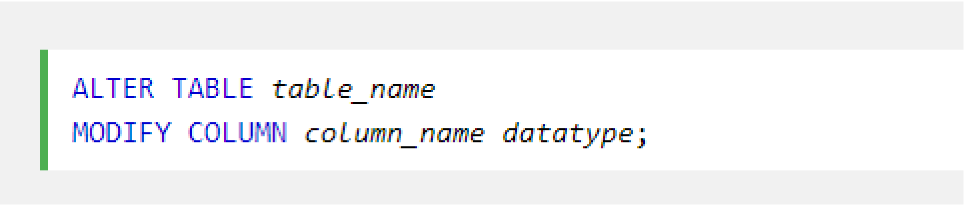


PersonID	LastName	FirstName	Address	City

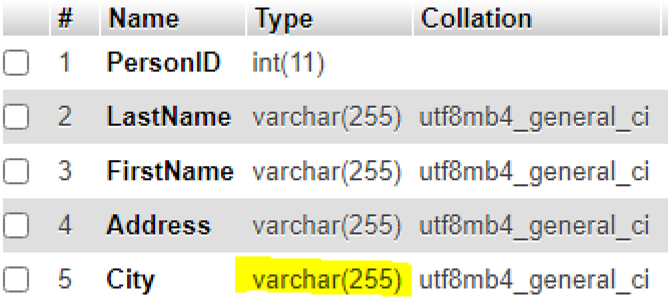
ALTER TABLE - ALTER/MODIFY COLUMN

To change the data type of a column in a table, use the following syntax:

My SQL / Oracle (prior version 10G):

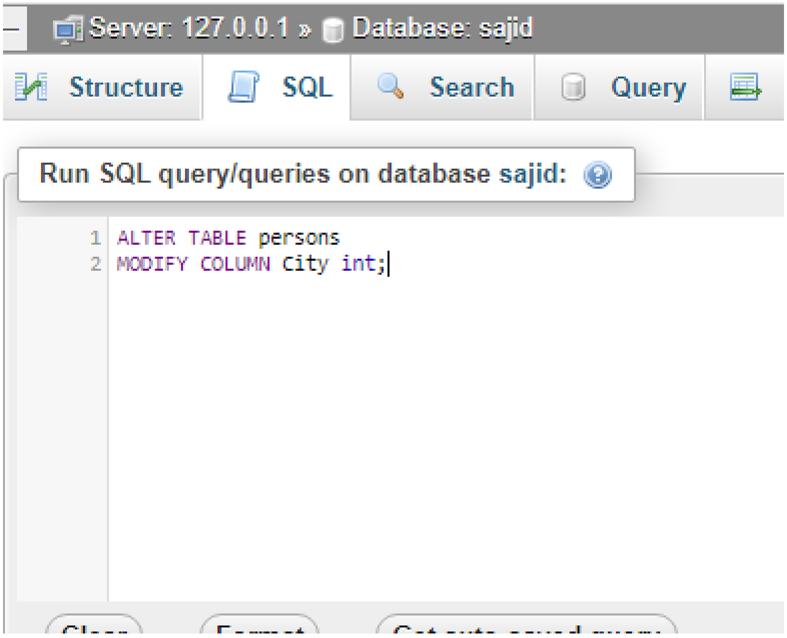


```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```



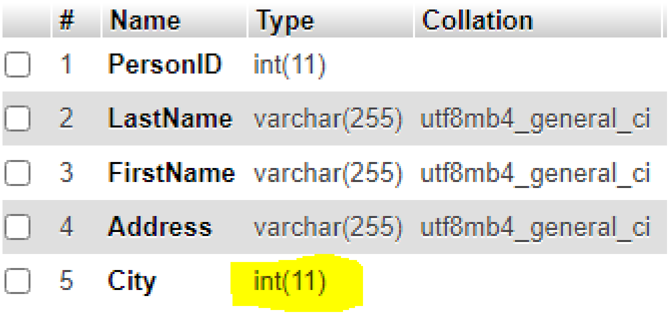
	#	Name	Type	Collation
□	1	PersonID	int(11)	
□	2	LastName	varchar(255)	utf8mb4_general_ci
□	3	FirstName	varchar(255)	utf8mb4_general_ci
□	4	Address	varchar(255)	utf8mb4_general_ci
□	5	City	varchar(255)	utf8mb4_general_ci

```
ALTER TABLE persons
MODIFY COLUMN City int;
```


Server: 127.0.0.1 » Database: sajid
Structure
SQL
Search
Query
Run SQL query/queries on database sajid:
Clear
Format
Get auto-saved query

```
ALTER TABLE persons
MODIFY COLUMN City int;
```

Output



	#	Name	Type	Collation
□	1	PersonID	int(11)	
□	2	LastName	varchar(255)	utf8mb4_general_ci
□	3	FirstName	varchar(255)	utf8mb4_general_ci
□	4	Address	varchar(255)	utf8mb4_general_ci
□	5	City	int(11)	

Self-Review Questions (SRQ) For Study Session 3

Now that you have completed this study unit, you can assess how well you have achieved its Learning Outcomes by answering these questions. Write your answers in your Study Diary and discuss them with your Tutor at the next Study Support Meeting or Online interactive sessions.

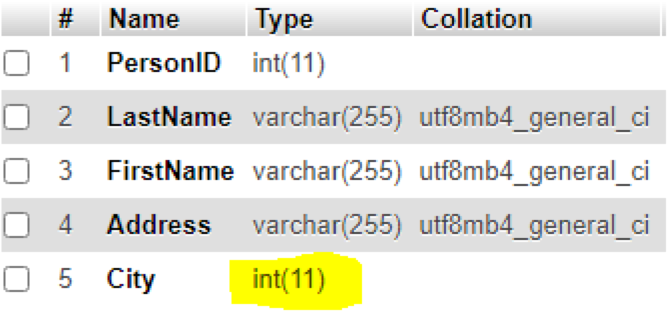
1. What can MySQL do?
2. To build a website what we need to show data from database.
3. How many types of MySQL data types describe in detail?
4. How to create database with command and GUI?
5. How to Drop database with command and GUI?
6. How to backup database with GUI?
7. How to create table with a statement?
8. How to Copy a table with MySQL Query?
9. How to Drop a table in MySQL?
10. What is the meaning of truncate table? Write a statement to truncate a table.
11. What is the meaning of Alter Table?

12. How ALTER TABLE statement is used to add, delete, or modify columns in an existing following table

    Add a Column Name (Email)

    Delete a Column Name (Address)

    Modify a Column Type int(11) to (varchar 255)



	#	Name	Type	Collation
☐	1	PersonID	int(11)	
☐	2	LastName	varchar(255)	utf8mb4_general_ci
☐	3	FirstName	varchar(255)	utf8mb4_general_ci
☐	4	Address	varchar(255)	utf8mb4_general_ci
☐	5	City	int(11)	

References and Additional Reading Materials

https://www.w3schools.com/sql/

https://www.tutorialspoint.com/mysql/index.htm