

CRUD Operations in MySQL:

XAMPP Control Panel:

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. The clear user interface logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

- Config: allows you to configure the XAMPP as well as the individual components
- Netstat: shows all running processes on the local computer
- Shell: opens a UNIX shell
- Explorer: opens the XAMPP folder in Windows Explorer
- Services: shows all services currently running in the background
- Help: offers links to user forums
- Quit: closes the XAMPP Control Panel

Starting modules:

- Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under 'Actions'. You can see which modules have been started because their names are highlighted green under the 'Module' title.
- If a module can't be started as a result of an error, you'll be informed of this straight away in red font. A detailed error report can help you identify the cause of the issue.

Port and Port number:

- A port is a virtual point where network connections start and end.
- A port number is a way to identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server.
- All network-connected devices come equipped with standardized ports that have an assigned number. These numbers are reserved for certain protocols and their associated function.
- Hypertext Transfer Protocol (HTTP) messages, for example, always go to port 80 -- one of the most commonly used ports.

Setting up XAMPP:

- A common source of error connected with Apache is blocked ports. If you're using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it's likely that the Tomcat port is being blocked, meaning the web server can't be started. There are three ways to solve this issue:
1. **Change the conflicting port:** Let's assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the

program and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.

2. **Change the XAMPP module port settings**

3. **End the conflicting program:** The simplest way to avoid port conflicts in the short term is to end the conflicting program (Skype in this case). If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved.

Module administration:

- You have an 'Admin' option located on the Control Panel for every module in your XAMPP.
- Click on the Admin button of your Apache server to go to the web address of your web server.
- The Control Panel will now start in your standard browser, and you'll be led to the **dashboard of your XAMPP's local host**.
- The dashboard features numerous links to websites for useful information as well as the open source project BitNami, which offers you many different applications for your XAMPP. Alternatively, you can reach the dashboard through *localhost/dashboard/*.
- You can use the Admin button of your database module to open phpMyAdmin. Here, you can manage the databases of your web projects that you're testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via *localhost/phpmyadmin/*.

Basic SQL:

Step 1:

- First of all start your XAMPP or WAMPP server. Then go to the link bar of your browser and type localhost/phpmyadmin and click enter.
- Then click on the new button that is available on the sidebar of that page.

Step 2:

- After click on the new button, put a name of your own choice on the create database field and click on the create button and you get a successful message on the screen that you create database successfully and will appear in the database list.
- Now we are going to see how to create a table in PhpMyAdmin under the database that we have created currently.
- You can't create table if you not have a database so first create database.

Step 3

- Click on the database name in which under you create a table.

- Here you have two option to create table first one is using **structure** and second one is using **SQL**. If you want to create table in structure option then put your table name on the create table name field and choose columns and click on the go button.
- Now we are going to see how to **create a table in PhpMyAdmin** under the database that you have create currently.
- You can't create table if you not have a database so first create database.

Step 4:

- To create new tables inside a database, open the phpMyAdmin tool, click on the Databases tab and click on the name of the desired database.
- On the new page that opens you will see a list of all the current tables inside the database and a section named Create table. In that section, in the Name field, input the desired new name of the table and then select the number of columns that the table should have via the Number of columns drop-down.
- When ready, click on Go to create the table.

On the next page, you can configure the structure of the columns in the new table. The different fields there are:

- **Name** – The name of the column;
- **Type** – The type of data, which will be stored in the corresponding column. More details about the possible choices can be found in the official MySQL Data Types documentation;
- **Length/Values** – The length of the field;
- **Default** – With this option, you can specify if the fields in the column would have a default value. This is useful when you want to have timestamps for the entries in each row;
- **Collation** – The data collation for each of the fields (Collation is the assembly of written information into a standard order. Many systems of collation are based on numerical order or alphabetical order, or extensions and combinations thereof. Collation is a fundamental element of most office filing systems, library catalogs, and reference books).
- **Attributes** – assign any special attributes to the fields;
- **Null** – Define whether the field value can be *NULL*;
- **Index** – Set the Index of the row;
- **AI** – Short for Auto Increment. If this option is enabled then the values in the fields of the column will be auto incremented;
- **Comments** – Here add comments, which will be included in the database SQL code.

When ready, click on **Save** to create the new table.

Step 5:

- 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.

Connecting to mySQL server:

- After opening the command line for mySQL server, run the following command on your shell:

```
mysql -h 127.0.0.1 -u root
```

After this command you would see the following details and you will be ready to do operations with mysql server:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 1

Server version: 5.6.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql>
```

Now, we are ready to create our database in mysql.

Creating a Database:

- To create a new database in MySQL, you use the CREATE DATABASE statement with the following syntax:

```
CREATE DATABASE IF NOT EXISTS database_name
```

```
[CHARACTER SET charset_name]
```

```
[COLLATE collation_name]
```

- In this syntax,
 1. Specify name of the database after the CREATE DATABASE keywords. The database name must be unique within a MySQL server instance. If you attempt to create a database with a name that already exists, MySQL will issue an error.
 2. Use the IF NOT EXISTS option to conditionally create a database if it doesn't exist.
 3. Specify the character set and collation for the new database. If you skip the CHARACTER SET and COLLATE clauses, MySQL will the default character set and collation for the new database.
 4. Next, we can check the current databases available on the server using the SHOW DATABASE statement. This step is optional.

```
SHOW DATABASES;
```

Dropping a Database:

The DROP DATABASE statement drops all tables in the database and deletes the database permanently. Therefore, you need to be very careful when using this statement.

The following shows the syntax of the DROP DATABASE statement:

DROP DATABASE [IF EXISTS] database_name;

- In this statement, you specify the name of the database which you want to delete after the DROP DATABASE keywords.
- If you drop a database that does not exist, MySQL will issue an error.
- To prevent an error from occurring if you delete a non-existing database, you can use the IF EXISTS option. In this case, MySQL will terminate the statement without issuing any error.
- The DROP DATABASE statement returns the number of tables it deleted.
- In MySQL, the schema is the synonym for the database. Therefore, you can use them interchangeably:

DROP SCHEMA [IF EXISTS] database_name;

Tables in MySQL:

- A table is used to organize data in the form of rows and columns and used for both storing and displaying records in the structure format.

Data types in MySQL:

- A database table contains multiple columns with specific data types such as numeric or string.
- MySQL provides more data types other than just numeric and string.
- Each data type in MySQL can be determined by the following characteristics:
 1. The kind of values it represents.
 2. The space that takes up and whether the values are a fixed-length or variable length.
 3. The values of the data type can be indexed or not.
 4. How MySQL compares the values of a specific data type.

INT data type:

- In MySQL, INT stands for the integer that is a whole number.
- An integer can be written without a fractional component e.g., 1, 100, 4, -10, and it cannot be 1.2, 5/3, etc.
- An integer can be zero, positive, and negative.
- MySQL supports all standard SQL integer types INTEGER or INT and SMALLINT.
- In addition, MySQL provides TINYINT, MEDIUMINT, and BIGINT as extensions to the SQL standard.
- MySQL INT data type can be signed and unsigned.

BIT data type:

- The BIT type that allows you to store bit values. Here is the syntax:

BIT(n)

- It can store up to n-bit values. The n can range from 1 to 64.

CHAR data type:

- It is a fixed-length character type in MySQL.
- You often declare the CHAR type with a length that specifies the maximum number of characters that you want to store.
- For example, CHAR(20) can hold up to 20 characters.
- The length of the CHAR data type can be any value from 0 to 255.

BOOLEAN data type:

- MySQL does not have built-in Boolean type. However, it uses [TINYINT(1)] instead. To make it more convenient, MySQL provides BOOLEAN or BOOL as the synonym of TINYINT(1).
- In MySQL, zero is considered as false, and non-zero value is considered as true.
- To use Boolean literals, you use the constants TRUE and FALSE that evaluate to 1 and 00 respectively. See the following example:

```
SELECT true, false, TRUE, FALSE, True, False;  
-- 1 0 1 00 1 0
```

Creating a Table:

The CREATE TABLE statement allows you to create a new table in a database.

The following illustrates the basic syntax of the CREATE TABLE statement:

```
CREATE TABLE [IF NOT EXISTS] table_name(  
column_1_definition,  
column_2_definition,  
...,  
table_constraints  
)
```

- Let's examine the syntax in greater detail.
 1. First, you specify the name of the table that you want to create after the CREATE TABLE keywords. The table name must be unique within a database. The IF NOT EXISTS is optional. It allows you to check if the table that you create already exists in the database. If this is the case, MySQL will ignore the whole statement and will not create any new table.

2. Second*,* you specify a list of columns of the table in the column_list section, columns are separated by commas.
- As we have seen in above syntax, there are columns definitions we need to define while creating a table, so let's see how we can define the columns in a table:
 - The following shows the syntax for a column's definition:

`column_name data_type(length) [NOT NULL] [DEFAULT value] [AUTO_INCREMENT] column_constraint;`

- Here are the details:
 1. The column_name specifies the name of the column. Each column has a specific data type and optional size e.g., VARCHAR(255)
 2. The NOT NULL constraint ensures that the column will not contain NULL. Besides the NOT NULL constraint, a column may have additional constraint such as CHECK, and UNIQUE.
 3. The DEFAULT specifies a default value for the column.
 4. The AUTO_INCREMENT indicates that the value of the column is incremented by one automatically whenever a new row is inserted into the table. Each table has a maximum one AUTO_INCREMENT column.
- Now, at the end of the column definition we can see the word column_constraint, so, after the column list, you can define table constraints such as UNIQUE, CHECK, PRIMARY KEY and FOREIGN KEY.
- For example, if you want to set a column or a group of columns as the primary key, you use the following syntax:

PRIMARY KEY (col1,col2,...)

The following statement creates a new table named tasks:

```
CREATE TABLE IF NOT EXISTS tasks (
  task_id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  start_date DATE,
  due_date DATE,
  status TINYINT NOT NULL,
  priority TINYINT NOT NULL,
  description TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

The tasks table has the following columns:

- The task_id is an auto-increment column. If you use the INSERT statement to insert a new row into the table without specifying a value for the task_id column, MySQL will automatically generate a sequential integer for the task_id starting from 1.

- The title column is a variable character string column whose maximum length is 255. It means that you cannot insert a string whose length is greater than 255 into this column. The NOT NULL constraint indicates that the column does not accept NULL. In other words, you have to provide a non-NULL value when you insert or update this column.
- The start_date and due_date are DATE columns. Because these columns do not have the NOT NULL constraint, they can store NULL. The start_date column has a default value of the current date. In other words, if you don't provide a value for the start_date column when you insert a new row, the start_date column will take the current date of the database server.
- The status and priority are the TINYINT columns which do not allow NULL.
- The description column is a TEXT column that accepts NULL.
- The created_at is a TIMESTAMP column that accepts the current time as the default value.

The task_id is the primary key column of the tasks table. It means that the values in the task_id column will uniquely identify rows in the table.

Dropping a Table:

- To remove existing tables, you use the MySQL DROP TABLE statement.
- Syntax of the DROP TABLE statement:

**DROP [TEMPORARY] TABLE [IF EXISTS] table_name [, table_name] ...
[RESTRICT | CASCADE]**

- The DROP TABLE statement removes a table and its data permanently from the database. In MySQL, you can also remove multiple tables using a single DROP TABLE statement, each table is separated by a comma (,).
- The TEMPORARY option allows you to remove temporary tables only. It ensures that you do not accidentally remove non-temporary tables.
- The IF EXISTS option conditionally drop a table only if it exists. If you drop a non-existing table with the IF EXISTS option, MySQL generates a NOTE, which can be retrieved using the SHOW WARNINGS statement.
- Note that the DROP TABLE statement only drops tables. It doesn't remove specific user privileges associated with the tables. Therefore, if you create a table with the same name as the dropped one, MySQL will apply the existing privileges to the new table, which may pose a security risk.
- The RESTRICT and CASCADE options are reserved for the future versions of MySQL.
- To execute the DROP TABLE statement, you must have DROP privileges for the table that you want to remove.

Keys in mySQL:

- A key as the term itself indicates, **unlocks access to the tables**.
- If you know the key, you know how to identify specific records and the relationships between the tables.
- Each key consists of one or more fields, or field prefix. The order of columns in an index is significant.

Primary Keys:

- A primary key is a column or a set of columns that uniquely identifies each row in the table. The primary key follows these rules:
 1. A primary key must contain unique values. If the primary key consists of multiple columns, the combination of values in these columns must be unique.
 2. A primary key column cannot have NULL values. Any attempt to insert or update NULL to primary key columns will result in an error. Note that MySQL implicitly adds a NOT NULL constraint to primary key columns.
 3. A table can have one and only one primary key.
 4. A primary key column often has the AUTO_INCREMENT attribute that automatically generates a sequential integer whenever you insert a new row into the table.
- When you define a primary key for a table, MySQL automatically creates a new index called PRIMARY.

Unique key:

- KEY is the synonym for INDEX. A UNIQUE index ensures that values in a column must be unique.
- Unlike the PRIMARY index, MySQL allows NULL values in the UNIQUE index. In addition, a table can have multiple UNIQUE indexes.
- Suppose that email and username of users in the users table must be unique. To enforce these rules, you can define UNIQUE indexes for the email and username columns as the following statement:
- Add a UNIQUE index for the username column:

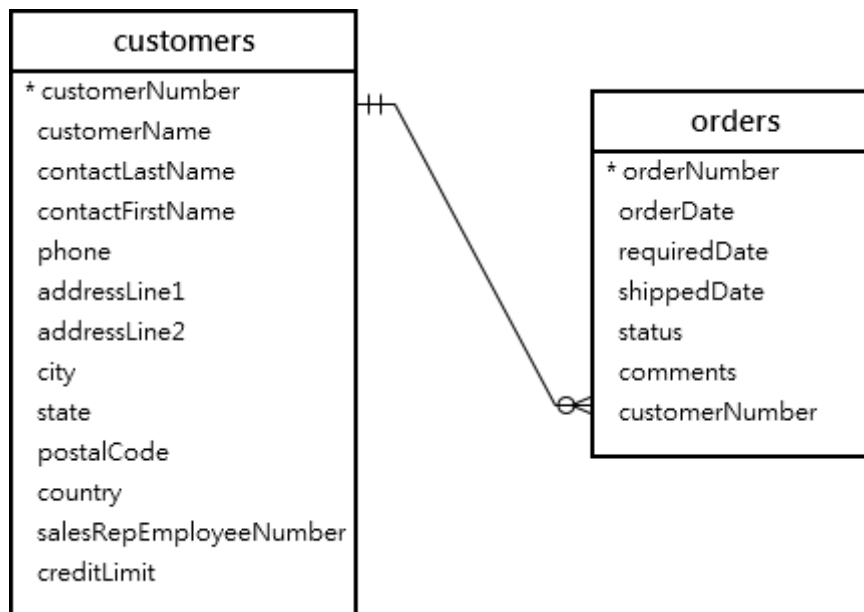
```
ALTER TABLE users
ADD UNIQUE INDEX username_unique (username ASC) ;
```

Add a UNIQUE index for the email column:

```
ALTER TABLE users
ADD UNIQUE INDEX email_unique (email ASC) ;
```

Foreign Keys:

- A foreign key is a column or group of columns in a table that links to a column or group of columns in another table. The foreign key places constraints on data in the related tables, which allows MySQL to maintain referential integrity.
- Let's take a look at the following customers and orders tables.



- In this diagram, each customer can have zero or many orders and each order belongs to one customer.
- The relationship between customers table and orders table is one-to-many. And this relationship is established by the foreign key in the orders table specified by the customerNumber column.
- The customerNumber column in the orders table links to the customerNumber primary key column in the customers table.
- The customers table is called the **parent table** or **referenced table**, and the orders table is known as the **child table** or **referencing table**.
- Typically, the foreign key columns of the child table often refer to the primary key columns of the parent table.
- A table can have more than one foreign key where each foreign key references to a primary key of the different parent tables.
- Once a foreign key constraint is in place, the foreign key columns from the child table must have the corresponding row in the parent key columns of the parent table or values in these foreign key column must be NULL (see the SET NULL action example below).
- For example, each row in the orders table has a customerNumber that exists in the customerNumber column of the customers table. Multiple rows in the orders table can have the same customerNumber.

Now, as we have learnt about the keys, next we can start learning operations in the next class.