Hands-on Lab: CREATE, ALTER, TRUNCATE, DROP into Tables in MySQL using phpMyAdmin

Estimated time *(minutes)*: 30 minutes

In this lab, you will learn how to create tables and load data in the MySQL database server by using the phpMyAdmin graphical user interface (GUI) tool.

Software Used in this Lab

In this lab, you will use [MySQL](), MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.

To complete this lab you will utilize MySQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database Used in this Lab

Mysql_learners database has been used in this lab.

Objectives

After completing this lab, you will be able to use phpMyAdmin to perform MySQL to:

- Create a database
- Create tables with, or without, constraints
- Insert data into tables using the editing pane
- Remove all rows from an existing table without deleting the table itself
- Delete an existing table or database

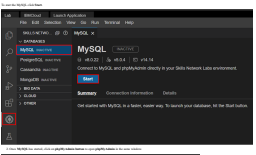
Exercise

In this exercise, through different tasks, you will learn how to create tables and load data in the MySQL database server using the phpMyAdmin graphical user interface (GUI) tool.

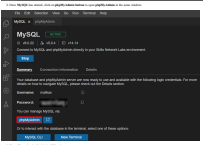
Task A: Create a database

1. Click on **Skills Network Toolbox**, in **Database** section, click **MySQL**.

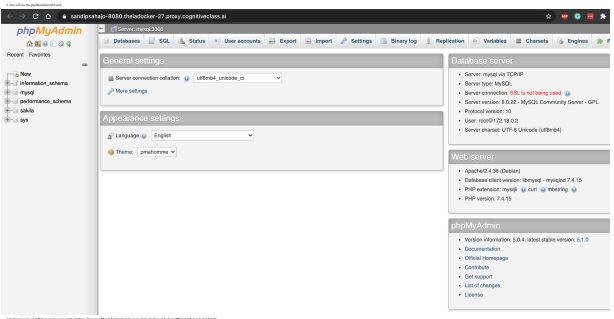
To start the MySQL, click **Start**.



2. Once **MySQL** has started, click on **phpMyAdmin button** to open **phpMyAdmin** in the new window.



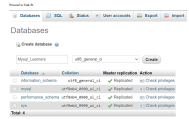
3. Now click on the phpMyAdmin icon to open.



4. In the first time, click **New** to create a new empty database. Then enter **Mysql_Learners** as the name of the database and select **utf8_general_ci** and click **Create**.

GUI is a lot more commonly used whenever something is to create or alter.

Pressed on Task B.



In this lab, you will learn some commonly used MySQL Data Definition Language statements (DDL). First you will learn the CREATE statement which is used to create a new table in a database. Next, you will learn the ALTER statement which is used to add, delete, or modify columns in an existing table. Then, you will learn the TRUNCATE statement which is used to remove all rows from an existing table without deleting the table itself. Lastly, you will learn the DROP statement which is used to delete an existing table in a database.

How does the syntax of a CREATE statement look?

```
CREATE TABLE table_name (
    column_name1,
    column_name2,
    column_name3,
    ....
);
```

How does the syntax of an ALTER statement look?

```
ALTER TABLE table_name
    ADD column_name1 data_type,
    ADD column_name2 data_type,
    ADD column_name3 data_type,
    ....
    DROP column_name1,
    DROP column_name2,
    DROP column_name3,
    ....
    MODIFY column_name1 new_data_type,
    MODIFY column_name2 new_data_type,
    ....
    RENAME COLUMN column_name1 TO new_column_name1,
    RENAME COLUMN column_name2 TO new_column_name2,
    ....
    ADD CONSTRAINT constraint_name1 constraint_type1,
    ADD CONSTRAINT constraint_name2 constraint_type2,
    ....
    DROP CONSTRAINT constraint_name1,
    DROP CONSTRAINT constraint_name2,
    ....
    RENAME TO new_table_name;
```

How does the syntax of a TRUNCATE statement look?

```
TRUNCATE TABLE table_name;
```

How does the syntax of a DROP statement look?

```
DROP TABLE table_name;
```

In this exercise, you will use the CREATE statement to create two new tables using DDL.

1. Type/paste the create two tables, **PETSALE** and **PET**. To create the two tables PETSALE and PET, copy the code below and paste it in the terminal of the **SQL** page. Click **Go**.

```
CREATE TABLE PETSALE(
    ID INT,
    PET INT,
    SALEPRICE DECIMAL(6,2),
    PROFIT DECIMAL(6,2),
    SALEDATE DATE,
    QUANTITY INT
);
CREATE TABLE PET(
    ID INT,
    PETNAME CHAR(16),
    PETTYPE CHAR(4),
    FOOD CHAR(10),
    BIRTHDAY DATE,
    STATUS CHAR(10),
    NAME CHAR(16)
);
```

2. Now insert some records into the two newly created tables and show all the records of the two tables. Copy the code below and paste it in the terminal of the **SQL** page. Click **Go**.

```
INSERT INTO PETSALE VALUES(
    (1, 4091, 100.00, 10.00, '2018-05-08', 5),
    (2, 4090, 500.00, 75.00, '2018-05-09', 2),
    (3, 4088, 70.00, 10.00, '2018-05-10', 1),
    (4, 4085, 90.00, 15.00, '2018-05-11', 3),
    (5, 4083, 40.00, 5.00, '2018-05-12', 1),
    (6, 4092, 60.00, 10.00, '2018-05-13', 2),
    (7, 4089, 80.00, 15.00, '2018-05-14', 4),
    (8, 4087, 110.00, 20.00, '2018-05-15', 6),
    (9, 4086, 150.00, 25.00, '2018-05-16', 3),
    (10, 4084, 45.00, 5.00, '2018-05-17', 1),
    (11, 4082, 120.00, 20.00, '2018-05-18', 2),
    (12, 4081, 200.00, 30.00, '2018-05-19', 4),
    (13, 4080, 75.00, 10.00, '2018-05-20', 1),
    (14, 4079, 300.00, 50.00, '2018-05-21', 5),
    (15, 4078, 55.00, 8.00, '2018-05-22', 2),
    (16, 4077, 180.00, 25.00, '2018-05-23', 3),
    (17, 4076, 95.00, 12.00, '2018-05-24', 1),
    (18, 4075, 250.00, 40.00, '2018-05-25', 4),
    (19, 4074, 65.00, 9.00, '2018-05-26', 2),
    (20, 4073, 140.00, 18.00, '2018-05-27', 3),
    (21, 4072, 85.00, 11.00, '2018-05-28', 1),
    (22, 4071, 320.00, 60.00, '2018-05-29', 5),
    (23, 4070, 50.00, 7.00, '2018-05-30', 2),
    (24, 4069, 190.00, 28.00, '2018-05-31', 4),
    (25, 4068, 130.00, 22.00, '2018-06-01', 3),
    (26, 4067, 60.00, 8.00, '2018-06-02', 1),
    (27, 4066, 400.00, 70.00, '2018-06-03', 6),
    (28, 4065, 105.00, 14.00, '2018-06-04', 2),
    (29, 4064, 175.00, 24.00, '2018-06-05', 4),
    (30, 4063, 90.00, 13.00, '2018-06-06', 1);
SELECT * FROM PETSALE;
```

Task B: ALTER using ADD COLUMN

1. Add a new **QUANTITY** column to the **PETSALE** table and show the altered table. Copy the code below and paste it in the terminal of the **SQL** page. Click **Go**.

```
ALTER TABLE PETSALE
    ADD COLUMN QUANTITY INT(2);
SELECT * FROM PETSALE;
```

2. Now update the newly added **QUANTITY** column of the **PETSALE** table with some values and show all the records of the table. Copy the code below and paste it in terminal of the **SQL** page. Click **Go**.

```
UPDATE PETSALE SET QUANTITY = 5 WHERE ID = 1;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 2;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 3;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 4;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 5;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 6;
UPDATE PETSALE SET QUANTITY = 4 WHERE ID = 7;
UPDATE PETSALE SET QUANTITY = 6 WHERE ID = 8;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 9;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 10;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 11;
UPDATE PETSALE SET QUANTITY = 4 WHERE ID = 12;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 13;
UPDATE PETSALE SET QUANTITY = 5 WHERE ID = 14;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 15;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 16;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 17;
UPDATE PETSALE SET QUANTITY = 4 WHERE ID = 18;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 19;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 20;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 21;
UPDATE PETSALE SET QUANTITY = 6 WHERE ID = 22;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 23;
UPDATE PETSALE SET QUANTITY = 4 WHERE ID = 24;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 25;
UPDATE PETSALE SET QUANTITY = 1 WHERE ID = 26;
UPDATE PETSALE SET QUANTITY = 5 WHERE ID = 27;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 28;
UPDATE PETSALE SET QUANTITY = 4 WHERE ID = 29;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 30;
SELECT * FROM PETSALE;
```

Task B: ALTER using DROP COLUMN

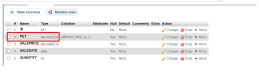
1. Delete the **PROFIT** column from the **PETSALE** table and show the altered table. Copy the code below and paste it in the terminal of the **SQL** page. Click **Go**.

```
ALTER TABLE PETSALE
    DROP COLUMN PROFIT;
SELECT * FROM PETSALE;
```

Task C: ALTER using ALTER COLUMN

1. Change the data type to **VARCHAR(20)** type for the column **PET** of the table **PETSALE** and show the altered table. Copy the code below and paste it in the terminal of the **SQL** page. Click **Go**.

```
ALTER TABLE PETSALE CHANGE "PET" "PET" VARCHAR(20);
SELECT * FROM PETSALE;
```

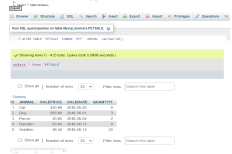


Task D: ALTER using RENAME COLUMN

1. Rename the column **PET** to **ANIMAL** of the **PETSALE** table and select the altered table. Copy the code below and paste it in the textarea of the **SQL** page. Click Go.

```
-- 1
-- 2
-- 3
-- ALTER TABLE "PETSALE" CHANGE "PET" "ANIMAL" varchar(20);
-- 4
-- select * from PETSALE;
```

Go



In this exercise, you will use the TRUNCATE statement to remove all rows from an existing table created in exercise 1 without deleting the table itself.

1. Remove all rows from the **PET** table and show the empty table. Copy the code below and paste it in the textarea of the **SQL** page. Click Go.

```
-- 1
-- 2
-- 3
-- TRUNCATE TABLE PET;
-- 4
-- SELECT * FROM PET;
```

Go

In this exercise, you will use the DROP statement to delete an existing table created in exercise 1.

1. Delete the **PET** table and verify if the table still exists or not (SELECT statement won't work if a table doesn't exist). Copy the code below and paste it in the textarea of the **SQL** page. Click Go.

```
-- 1
-- 2
-- 3
-- DROP TABLE PET;
-- 4
-- SELECT * FROM PET;
```

Go

Congratulations! You have completed this lab, and you are ready for the next topic.

Change Log

Date	Version	Changed by	Change Description
2021-10-20	0.1	Appakkudimani Kumar	Updated instructions
2020-07-27	0.1	Lakshmi Holla	Updated formatting
2020-07-24	0.1	Lakshmi Holla	Added instructions for dropping, renaming, altering, truncating commands
2020-07-09	0.1	Lakshmi Holla	Modified Regulatory Version

© IBM Corporation 2021. All rights reserved.