

# Lab: CREATE, ALTER, TRUNCATE, DROP into Tables in MySQL using phpMyAdmin



**Estimated time needed:** 20 minutes

In this lab, you will learn how to create tables and load data in the MySQL database service 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 with MySQL to:

- Create a database.
- Create a new table in a database.
- Add, delete, or modify columns in an existing table.
- Remove all rows from an existing table without deleting the table itself.
- Delete an existing table in a database

## Exercise

In this exercise through different tasks, you will learn how to create tables and load data in the MySQL database service 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 **Create**.

The screenshot shows the Skills Network Tools interface. On the left, there's a sidebar with various icons and sections: SKILLS NETWORK TOO... (with a file icon), Welcome (with a person icon), DATABASES (with a database icon), BIG DATA (with a play button icon), CLOUD (with a cloud icon), EMBEDDABLE AI (with a brain icon), OTHER (with a gear icon), and Launch Application (with a flask icon). The DATABASES section is expanded, showing MySQL INACTIVE (highlighted with a red box and circled with a red number 1), PostgreSQL INACTIVE, Cassandra INACTIVE, and MongoDB INACTIVE. To the right of the sidebar, there's a large MySQL card. The card has a title 'MySQL' with a 'C' icon, version numbers 8.0.22 and 5.0.4, a 'Connect to MySQL and...' link, and a prominent blue 'Create' button. Below the card, there are 'Summary' and 'Connect' links, and a 'Get started with MySQL' section.

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

The screenshot shows the Skills Network Tools interface. On the left, there's a sidebar with various icons: a file icon, a magnifying glass, a gear, a triangle, a flask, and a tree. Below these are sections for 'DATASES', 'BIG DATA', 'CLOUD', 'EMBEDDABLE AI', 'OTHER', and 'Launch Application'. The 'DATASES' section has a dropdown menu open, showing 'MySQL ACTIVE' (which is highlighted with a red box), 'PostgreSQL INACTIVE', 'Cassandra INACTIVE', and 'MongoDB INACTIVE'. To the right of the sidebar is a main panel titled 'MySQL'. It displays the MySQL logo, version 8.0.22, and port 5.0.4. Below this is a 'Create' button. Further down, there are 'Summary' and 'Connect' tabs, followed by a large text area with details about managing the database. At the bottom, there are two buttons: 'phpMyAdmin' (with a red box around it) and 'MySQL CLI'.

File Edit Selection View Go Run Terminal Help

← → | □

SKILLS NETWORK TOO... ⚙️ ⓘ Welcome MySQL

DATABASES

MySQL ACTIVE

PostgreSQL INACTIVE

Cassandra INACTIVE

MongoDB INACTIVE

> BIG DATA

> CLOUD

> EMBEDDABLE AI

> OTHER

Launch Application

**MySQL**

8.0.22 | 5.0.4

Connect to MySQL and ...

Create

Summary Connect

Your database and phpMyAdmin details on how to navigate and manage MySQL.

You can manage MySQL using the phpMyAdmin interface or the MySQL CLI.

**phpMyAdmin** ↗

Or to interact with the database, use the MySQL CLI.

**MySQL CLI**

3. You will see the phpMyAdmin GUI tool.

The screenshot shows the phpMyAdmin interface with the following details:

- General settings:** Server connection collation is set to `utf8mb4_unicode_ci`.
- Appearance settings:** Language is set to English, Theme is pmahomme.
- Database server:**
  - Server: labs-mysql-echoing-cold-eggplant via TCP/IP
  - Server type: MySQL
  - Server connection: **SSL is not being used**
  - Server version: 8.0.37 - MySQL Community Server - GPL
  - Protocol version: 10
  - User: root@172.17.25.218
  - Server charset: UTF-8 Unicode (utf8mb4)
- Web server:**
  - Apache
  - Database client version: libmysql - mysqlnd 8.2.23

4. In the tree-view, click **New** to create a new empty database. Then enter **Mysql\_Learners** as the name of the database and select **utf8mb4\_0900\_ai\_ci** and click **Create**.

UTF-8 is the most commonly used character encoding for content or data.  
Proceed to Task B.

# phpMyAdmin

Recent Favorites

New ①

- + information\_schema
- + mysql
- + performance\_schema
- + sys

Create ② Mysql\_L

Check

Data

inform

mysql

perfo

sys

The screenshot shows the phpMyAdmin interface. On the left, there's a tree view of databases: 'information\_schema', 'mysql', 'performance\_schema', and 'sys'. A red box highlights the 'New' button, and a red circle with the number '1' is placed next to it. On the right, there's a 'Create' dialog box. Inside the dialog, a red box highlights the 'Mysql\_L' text field, and a red circle with the number '2' is placed next to it. Below the dialog, there are several checkboxes for other database names: 'inform', 'mysql', 'perfo', and 'sys'. The top right corner of the interface shows a 'Server' icon and the word 'Database'.



In this lab, you will learn some commonly used DDL (Data Definition Language) statements of SQL. 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 (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

#### How does the syntax of an ALTER statement look?

```
ALTER TABLE table_name
ADD COLUMN column_name data_type column_constraint;
ALTER TABLE table_name
DROP COLUMN column_name;
ALTER TABLE table_name
ALTER COLUMN column_name SET DATA TYPE data_type;
ALTER TABLE table_name
CHANGE current_column_name new_column_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 Db2.

1. You need to create two tables, **PETSALE** and **PET**. To create the two tables PETSALE and PET, copy the code below and paste it to the textarea of the **SQL** page. Click **Go**.

```
CREATE TABLE PETSALE (
```

```

ID INTEGER NOT NULL,
PET CHAR(20),
SALEPRICE DECIMAL(6,2),
PROFIT DECIMAL(6,2),
SALEDATE DATE
);

CREATE TABLE PET (
ID INTEGER NOT NULL,
ANIMAL VARCHAR(20),
QUANTITY INTEGER
);

```

The screenshot shows the MySQL Workbench interface with the following details:

- Structure** tab is selected.
- SQL** tab is open, displaying the SQL code for creating the tables.
- Run SQL query/queries on database Mysql\_learners:** The code is as follows:
 

```

1 CREATE TABLE PETSAL (
2   ID INTEGER NOT NULL,
3   PET CHAR(20),
4   SALEPRICE DECIMAL(6,2),
5   PROFIT DECIMAL(6,2),
6   SALEDATE DATE
7 );
8
9 CREATE TABLE PET (
10  ID INTEGER NOT NULL,
11  ANIMAL VARCHAR(20),
12  QUANTITY INTEGER
13 );

```
- Buttons:** Clear, Format, Get auto-saved query, Bind parameters, Delimiter, Show this query here again, Retain query box, Rollback when finished, Enable foreign key checks, Go.
- Output:** Shows a green bar indicating success: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0562 seconds.)". Below it, the CREATE TABLE statements are shown.

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 to the textarea of the **SQL** page. Click **Go**.

```

INSERT INTO PETSAL VALUES
(1,'Cat',450.09,100.47,'2018-05-29'),
(2,'Dog',666.66,150.76,'2018-06-01'),
(3,'Parrot',50.00,8.9,'2018-06-04'),
(4,'Hamster',60.60,12,'2018-06-11'),
(5,'Goldfish',48.48,3.5,'2018-06-14');

INSERT INTO PET VALUES
(1,'Cat',3),
(2,'Dog',4),
(3,'Hamster',2);

SELECT * FROM PETSAL;
SELECT * FROM PET;

```

The screenshot shows the MySQL Workbench interface with the following details:

- SQL** tab is selected, showing the inserted data into PETSAL and PET tables.
- PETSAL** table data:
 

| ID | PET      | SALEPRICE | PROFIT | SALEDATE   |
|----|----------|-----------|--------|------------|
| 1  | Cat      | 450.09    | 100.47 | 2018-05-29 |
| 2  | Dog      | 666.66    | 150.76 | 2018-06-01 |
| 3  | Parrot   | 50.00     | 8.9    | 2018-06-04 |
| 4  | Hamster  | 60.60     | 12.00  | 2018-06-11 |
| 5  | Goldfish | 48.48     | 3.50   | 2018-06-14 |
- PET** table data:
 

| ID | ANIMAL  | QUANTITY |
|----|---------|----------|
| 1  | Cat     | 3        |
| 2  | Dog     | 4        |
| 3  | Hamster | 2        |
- Output:** Shows a green bar indicating success: "Showing rows 0-4 (5 total). Query took 0.0000 seconds." and "Showing rows 0-2 (3 total). Query took 0.0003 seconds.".

In this exercise, you will use the ALTER statement to add, delete, or modify columns in two of the existing tables created in exercise 1.

## Task A: ALTER using ADD COLUMN

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

```
ALTER TABLE PETSALE
ADD COLUMN QUANTITY INTEGER;
SELECT * FROM PETSALE;
```

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0461 seconds.)

ALTER TABLE PETSALE ADD COLUMN QUANTITY INTEGER

[Edit inline](#) [Edit](#) [Create PHP code](#)

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total). Query took 0.0005 seconds.

SELECT \* FROM PETSALE

[Profiling](#) [Edit inline](#) [Edit](#) [Explain SQL](#) [Create PHP code](#) [Replies](#)

Show all Number of rows: 25 Filter rows Search this table

| ID | PET      | SALEPRICE | PROFIT | SALEDATE   | QUANTITY |
|----|----------|-----------|--------|------------|----------|
| 1  | Cat      | 450.09    | 100.47 | 2018-05-29 | NULL     |
| 2  | Dog      | 666.66    | 150.76 | 2018-06-01 | NULL     |
| 3  | Pam      | 50.00     | 8.90   | 2018-06-04 | NULL     |
| 4  | Hamster  | 60.60     | 12.00  | 2018-06-11 | NULL     |
| 5  | Goldfish | 48.48     | 3.91   | 2018-06-14 | NULL     |

- 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 to textarea of the **SQL** page. Click **Go..**

```
UPDATE PETSALE SET QUANTITY = 9 WHERE ID = 1;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 2;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 3;
UPDATE PETSALE SET QUANTITY = 6 WHERE ID = 4;
UPDATE PETSALE SET QUANTITY = 24 WHERE ID = 5;
SELECT * FROM PETSALE;
```

```
UPDATE PETSALE SET QUANTITY = 9 WHERE ID = 1;
UPDATE PETSALE SET QUANTITY = 3 WHERE ID = 2;
UPDATE PETSALE SET QUANTITY = 2 WHERE ID = 3;
UPDATE PETSALE SET QUANTITY = 6 WHERE ID = 4;
UPDATE PETSALE SET QUANTITY = 24 WHERE ID = 5;

SELECT * FROM PETSALE;
```

| ID |
|----|
| 1  |
| 2  |
| 3  |
| 4  |
| 5  |

## Task B: ALTER using DROP COLUMN

- Delete the **PROFIT** column from the **PETSALE** table and show the altered table. Copy the code below and paste it to the textarea of the **SQL** page. Click **Go..**

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

The screenshot shows the MySQL Workbench interface. The top navigation bar includes 'Browse', 'Structure', 'SQL', 'Search', and 'Insert' tabs. Below the tabs, a message says 'Run SQL query/queries on table Mysql\_learners.PETSALE:'. The SQL code area contains the following:

```

1 ALTER TABLE PETSALE
2     DROP COLUMN PROFIT;
3
4 SELECT * FROM PETSALE;

```

To the right, a results grid displays data from the PETSALE table:

| + Options | ID | PET      | SALEPRICE | SALEDATE   | QUA |
|-----------|----|----------|-----------|------------|-----|
|           | 1  | Cat      | 450.09    | 2018-05-29 |     |
|           | 2  | Dog      | 666.66    | 2018-06-01 |     |
|           | 3  | Parrot   | 50.00     | 2018-06-04 |     |
|           | 4  | Hamster  | 60.60     | 2018-06-11 |     |
|           | 5  | Goldfish | 48.48     | 2018-06-14 |     |

Below the grid, there are buttons for 'Show all' and 'Number of rows: 25'.

### Task C: ALTER using ALTER COLUMN

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

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

The screenshot shows the MySQL Workbench interface. The top navigation bar includes 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations' tabs. Below the tabs, a message says 'Run SQL query/queries on table Mysql\_learners.PETSALE:'. The SQL code area contains the following:

```

1
2 ALTER TABLE PETSALE CHANGE `PET` `PET` VARCHAR(20);
3
4 SELECT * FROM PETSALE;

```

At the bottom, the 'Table structure' tab is selected, showing the table definition:

| # | Name       | Type         | Collation          | Attributes | Null | Default | Comments | Extra | Action  |
|---|------------|--------------|--------------------|------------|------|---------|----------|-------|---|
| 1 | ID         | int          |                    |            | No   | None    |          |       | <span style="color: blue;">Change</span> <span style="color: red;">Drop</span> <span style="color: blue;">More</span> |
| 2 | <b>PET</b> | varchar(20)  | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |       | <span style="color: blue;">Change</span> <span style="color: red;">Drop</span> <span style="color: blue;">More</span> |
| 3 | SALEPRICE  | decimal(6,2) |                    |            | Yes  | NULL    |          |       | <span style="color: blue;">Change</span> <span style="color: red;">Drop</span> <span style="color: blue;">More</span> |
| 4 | SALEDATE   | date         |                    |            | Yes  | NULL    |          |       | <span style="color: blue;">Change</span> <span style="color: red;">Drop</span> <span style="color: blue;">More</span> |
| 5 | QUANTITY   | int          |                    |            | Yes  | NULL    |          |       | <span style="color: blue;">Change</span> <span style="color: red;">Drop</span> <span style="color: blue;">More</span> |

## Task D: ALTER using RENAME COLUMN

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

```
ALTER TABLE `PETSALE` CHANGE `PET` `ANIMAL` varchar(20);
SELECT * FROM PETSALE;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations.
- SQL Editor:** Run SQL query/queries on table Mysql\_learners.PETSALE: `ALTER TABLE `PETSALE` CHANGE `PET` `ANIMAL` varchar(20);`
- Output Window:** Shows a green success message: "Showing rows 0 - 4 (5 total, Query took 0.0006 seconds.)" followed by the result of the `select * from `PETSALE`` query.
- Table View:** A table named PETSALE with columns ID, ANIMAL, SALEPRICE, SALEDATE, and QUANTITY. The data is as follows:
 

| ID | ANIMAL   | SALEPRICE | SALEDATE   | QUANTITY |
|----|----------|-----------|------------|----------|
| 1  | Cat      | 450.09    | 2018-05-29 | 9        |
| 2  | Dog      | 666.66    | 2018-06-01 | 3        |
| 3  | Parrot   | 50.00     | 2018-06-04 | 2        |
| 4  | Hamster  | 60.60     | 2018-06-11 | 6        |
| 5  | Goldfish | 48.48     | 2018-06-14 | 24       |
- Filter and Row Selection:** Options to Show all, Number of rows (set to 25), Filter rows, and a search bar.

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 to the textarea of the **SQL** page. Click **Go**.

```
TRUNCATE TABLE PET ;
SELECT * FROM PET;
```

```
1 TRUNCATE TABLE PET ;
2 SELECT * FROM PET;
```

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 to the textarea of the **SQL** page. Click **Go**.

```
DROP TABLE PET;
SELECT * FROM PET;
```

```
1 DROP TABLE PET;
2 SELECT * FROM PET;
```

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

[Lakshmi Holla](#)

[Malika Singla](#)

© Copyright IBM Corporation. All rights reserved.