Hands-on Lab: CREATE, ALTER, TRUNCATE, DROP

Estimated time needed: 20 minutes

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

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 use 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.

Objectives

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

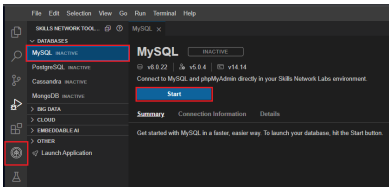
- Create a database.
- Create a new table in a database.
- Add, delete, or modify columns in an existing table.
- Remove all rows from an existing table without deleting the table itself.
- Delete an existing table in a database.

Task 1: Create a database

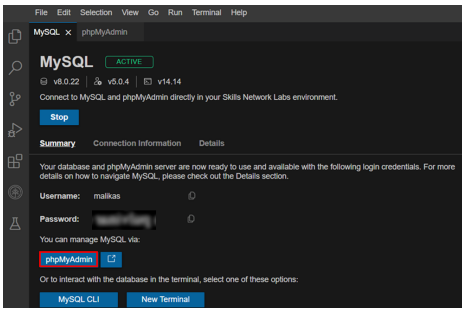
Follow the steps below to create a new database in the phpMyAdmin GUI of MySQL.

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

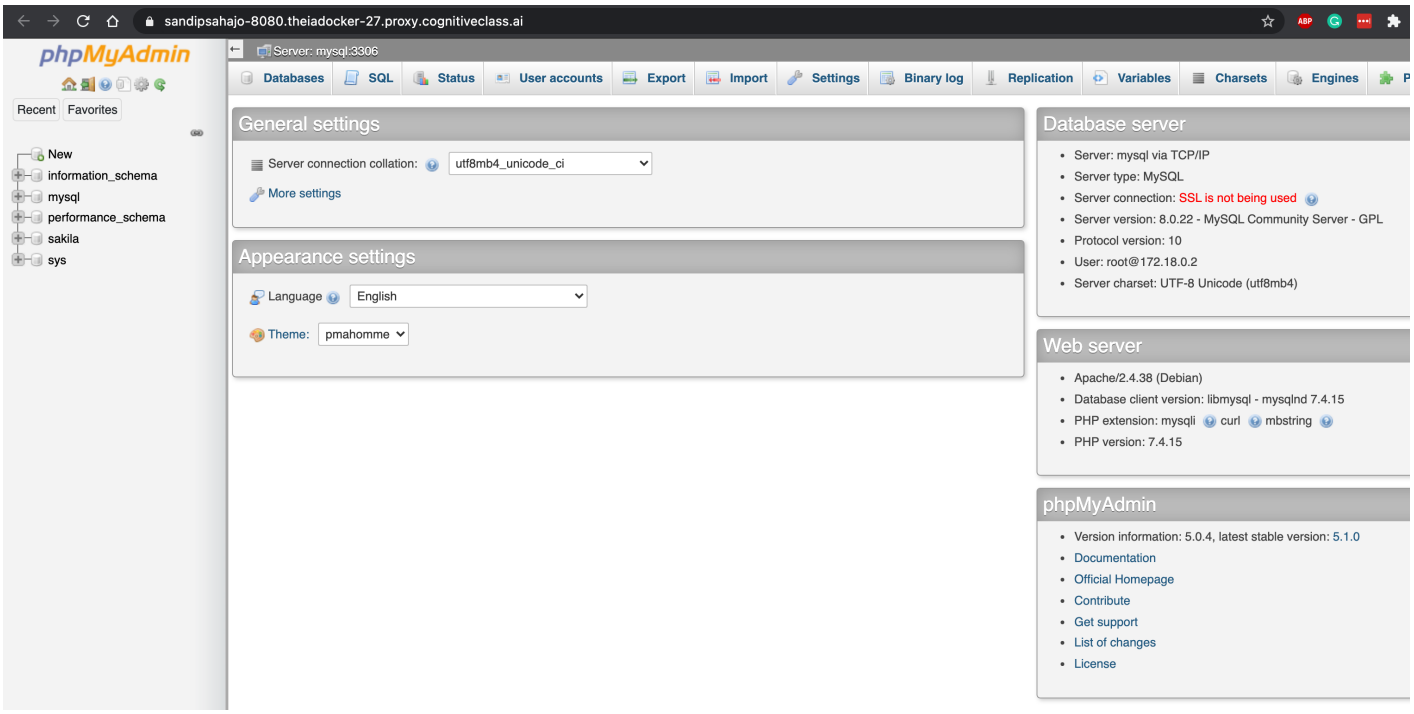
   To start the MySQL, click **Start**.



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

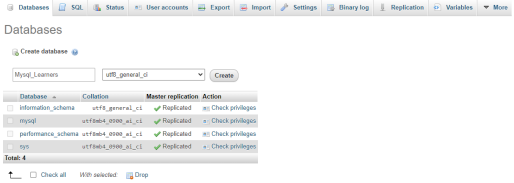


3. You will see the phpMyAdmin GUI tool.



4. In the tree view, click `New` to create a new empty database. Then, enter `Mysql_Learners` as the name of the database, select `utf8_general_ci` from the dropdown, and click `Create`.

   UTF-8 is the most commonly used character encoding for content or data.



Task 2a : CREATE statement

Now, you will use the CREATE statement to create two new tables.
Follow the instructions to complete this task.

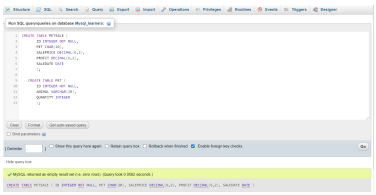
1. You need to create two tables, `PETSALE` and `PET`. To create the two tables, copy the code below and paste it into the text area of the `SQL` tab. Click `Go`.

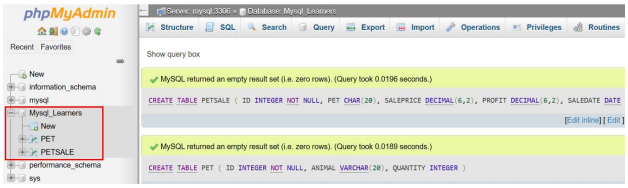
```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
```

```
 7. 7
 8. 8
 9. 9
10. 10
11. 11
12. 12
13. 13
```

```
1. CREATE TABLE PETSALE (
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.         );
```

Copied!





Task 2b: INSERT statement

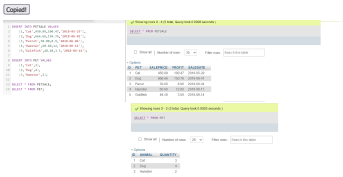
Now, insert some records into the two newly created tables. You can also add SELECT statements to print the contents of the tables once they are loaded with data.

Copy the code below and paste it into the text area of the SQL tab. Then, click **Go**.

```
 1. 1
 2. 2
 3. 3
 4. 4
 5. 5
 6. 6
 7. 7
 8. 8
 9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
```

```
1. INSERT INTO PETSALE VALUES
2.         (1,'Cat',450.09,100.47,'2018-05-29'),
3.         (2,'Dog',666.66,150.76,'2018-06-01'),
4.         (3,'Parrot',50.00,8.9,'2018-06-04'),
5.         (4,'Hamster',60.60,12,'2018-06-11'),
6.         (5,'Goldfish',48.48,3.5,'2018-06-14');
7.
8. INSERT INTO PET VALUES
9.         (1,'Cat',3),
10.         (2,'Dog',4),
11.         (3,'Hamster',2);
12.
13. SELECT * FROM PETSALE;
14. SELECT * FROM PET;
```

Copied!



Task 3: ALTER statement

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

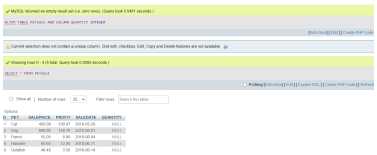
1. Adding a column

Add a new column named `QUANTITY` to the `PETSALE` table and display the altered table.
For this, copy the code below and paste it into the text area of the **SQL** page. Click **Go**.

```
1. 1
2. 2
3. 3
4. 4
```

```
1. ALTER TABLE PETSALE
2. ADD COLUMN QUANTITY INTEGER;
3.
4. SELECT * FROM PETSALE;
```

Copied!

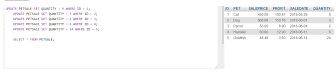


Now update the newly added **QUANTITY** column of the **PETSALE** table with some values and show all the table records. Copy the code below and paste it into text area of the **SQL** page. Click **Go**.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
```

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

Copied!



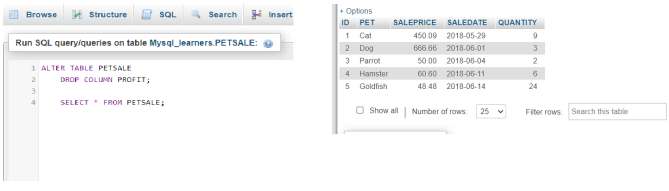
2. Deleting a column

Delete the **PROFIT** column from the **PETSALE** table and show the altered table. Copy the code below and paste it into the text area of the **SQL** page. Click **Go**.

```
1. 1
2. 2
3. 3
4. 4
```

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

Copied!



3. Modify a column

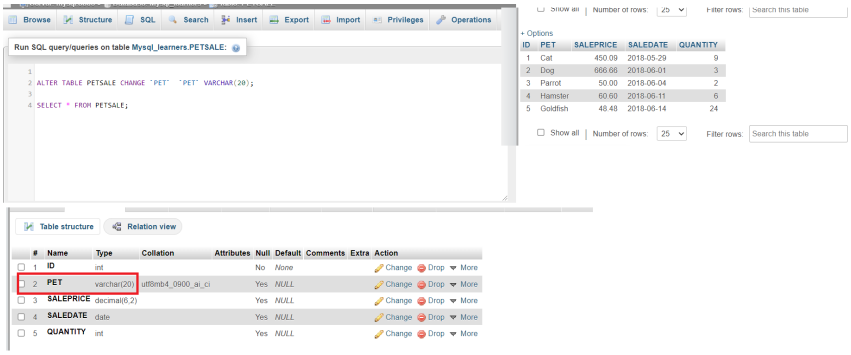
Change the data type to `VARCHAR(20)` type of the column `PET` of the table `PETSALE` and show the altered table. Copy the code below and paste it into the text area of the SQL page. Click **Go**.

```
1. 1
2. 2
3. 3
```

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

Copied!

You can click on the table name `PETSALE` in the tree structure on the left and then click on the `Structure` tab in the interface. You can then see the table structure shows the modified column data type, as shown in the image below.



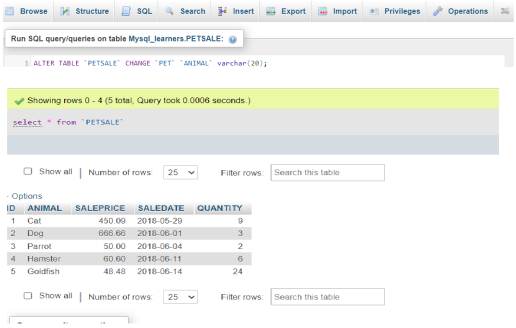
4. Rename a Column

Rename the column `PET` to `ANIMAL` of the `PETSALE` table and show the altered table. Copy the code below and paste it into the text area of the SQL page. Click Go.

```
1. 1
2. 2
3. 3

1. ALTER TABLE `PETSALE` CHANGE `PET` `ANIMAL` varchar(20);
2.
3. SELECT * FROM PETSALE;
```

Copied!



Task 4: TRUNCATE statement

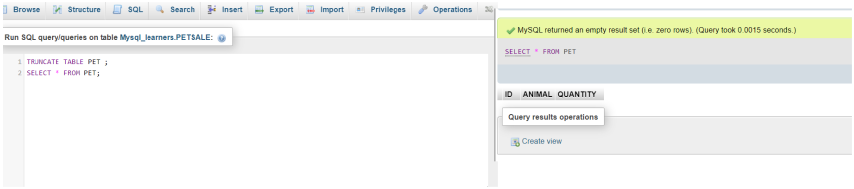
In this exercise, you will use the TRUNCATE statement to remove all rows from an existing table without deleting it.

Let's remove all rows from the `PET` table and show the empty table. Copy the code below and paste it into the text area of the SQL page. Click Go.

```
1. 1
2. 2
3. 3

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

Copied!



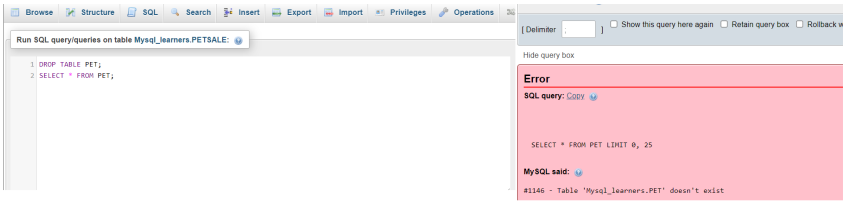
Task 5: DROP statement

Finally, you will use the DROP statement to delete an existing table. Let's delete the `PET` table and verify if the table still exists or not (the SELECT statement should give an error if a table doesn't exist). Copy the code below and paste it into the text area of the SQL page. Click Go.

```
1. 1
2. 2
3. 3

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

Copied!



Practice problems

Try the following problems for an enhanced practice of the concepts learned in this lab.

1. Create a new table in the database named `Toys` with attributes as ID (integer), Variety (variable length string), and Quantity (integer). Make sure the ID is not Null.

► Click here for the solution

2. Add the below-mentioned entries to the table using the INSERT statement.

ID	Variety	Quantity
1	Chew toy	20
2	Balls	50
3	Bowls	30
4	Foldable bed	40

► Click here for the solution

3. ALTER the length of 'Variety' in the table to 30 characters.

► Click here for the solution

4. TRUNCATE the table 'Toys'

► Click here for the solution

5. DROP the table 'Toys'

► Click here for the solution

Conclusion

Congratulations on successfully completing this lab.

By now, you have learned how to:

- Create a database in phpMyAdmin GUI on MySQL.

- Use the CREATE statement to create new tables in the database.
- Use the INSERT statement to add records to the tables.
- Use the ALTER statement to add, delete, rename, or modify the columns of an existing table.
- Use the TRUNCATE statement to delete the contents of an existing table (but not the table).
- Use the DROP statement to delete an entire table.

Author(s)

Lakshmi Holla

Malika Singla

Additional Contributor(s)

Abhishek Gagneja

Changelog

Date	Version	Changed by	Change Description
2023-10-10	0.7	Mercedes Schneider	QA Pass w/Edits
2023-10-07	0.6	Misty Taylor	ID Check
2023-09-09	0.5	Abhishek Gagneja	Updated instructions
2022-10-28	0.4	Appalabhaktula Hema	Updated instructions
2022-07-27	0.3	Lakshmi Holla	updated html tag
2022-06-04	0.2	Lakshmi Holla, Malika Singla	Updated the MySQL starting commands
2021-11-01	0.1	Lakshmi Holla, Malika Singla	Initial Version