

# Hands-on 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

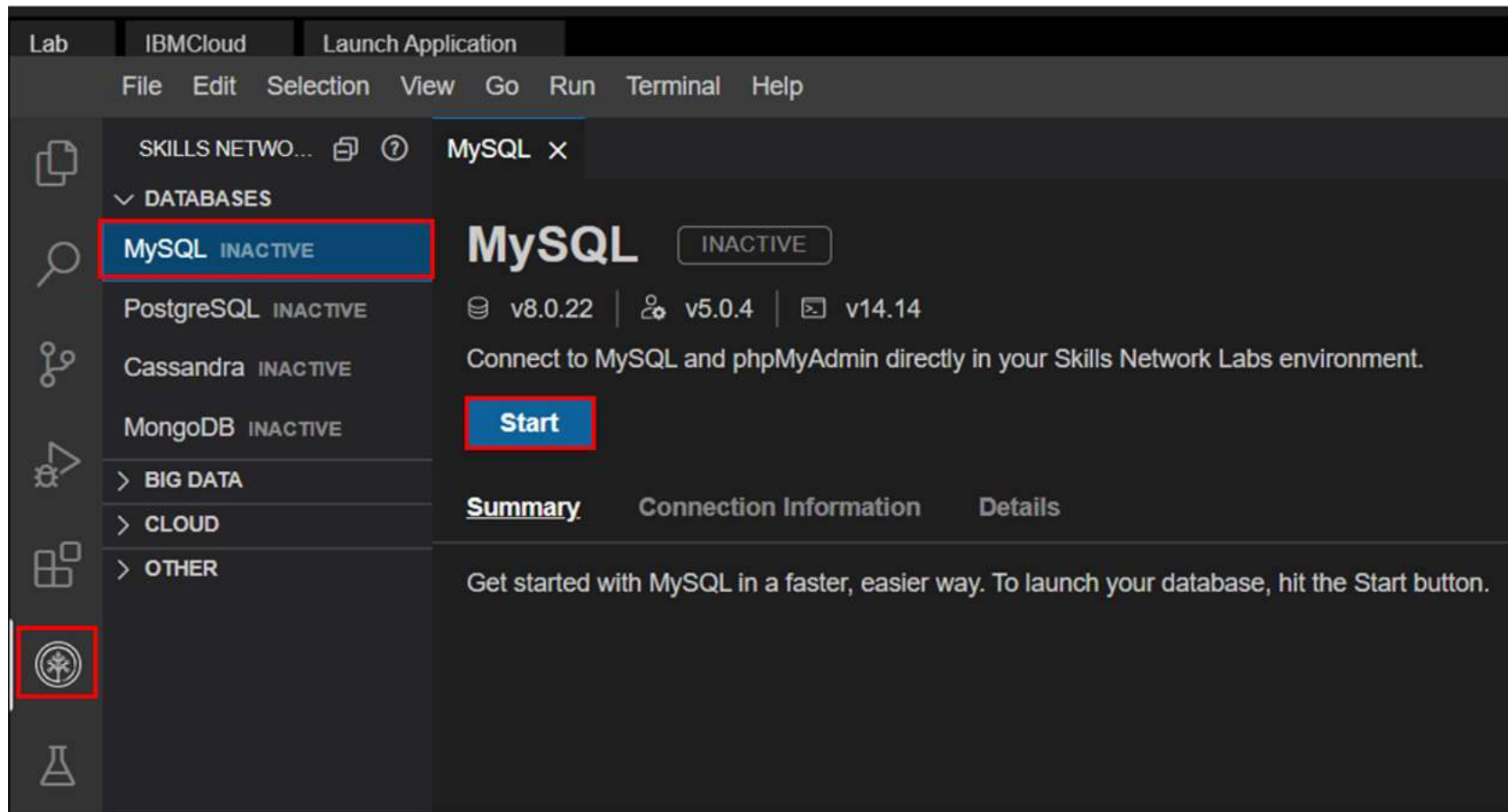
## Exercise

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 **Start**.



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

The screenshot shows a web-based interface for managing MySQL and phpMyAdmin. At the top, there is a menu bar with options: File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the menu, there are tabs for 'MySQL' and 'phpMyAdmin'. The 'MySQL' tab is active, indicated by a green 'ACTIVE' label. The main content area displays the MySQL version (v8.0.22), phpMyAdmin version (v5.0.4), and a connection status (v14.14). A message states: 'Connect to MySQL and phpMyAdmin directly in your Skills Network Labs environment.' Below this message is a blue 'Stop' button. There are three tabs: 'Summary', 'Connection Information', and 'Details'. The 'Summary' tab is selected. The summary text reads: 'Your database and phpMyAdmin server are now ready to use and available with the following login credentials. For more details on how to navigate MySQL, please check out the Details section.' Below this, the 'Username' is 'malikas' and the 'Password' is masked. There are copy icons next to both fields. Below the password field, it says 'You can manage MySQL via:' followed by a blue button labeled 'phpMyAdmin' with an external link icon. At the bottom, it says 'Or to interact with the database in the terminal, select one of these options:' followed by two buttons: 'MySQL CLI' and 'New Terminal'.

File Edit Selection View Go Run Terminal Help

MySQL x phpMyAdmin

# MySQL

ACTIVE

v8.0.22 | v5.0.4 | v14.14

Connect to MySQL and phpMyAdmin directly in your Skills Network Labs environment.

Stop

Summary Connection Information Details

Your database and phpMyAdmin server are now ready to use and available with the following login credentials. For more details on how to navigate MySQL, please check out the Details section.

Username: malikas

Password:

You can manage MySQL via:

phpMyAdmin

Or to interact with the database in the terminal, select one of these options:

MySQL CLI New Terminal

3. You will see the phpMyAdmin GUI tool.

← → ↻ 🏠 🔒 sandipsahajo-8080.theiadocker-27.proxy.cognitiveclass.ai

# phpMyAdmin

🏠 📊 ⓘ 📄 ⚙️ 💰

Recent Favorites

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

← 🖨️ Server: mysql:3306

**Databases** **SQL** **Status** **User accounts** **Export**

## General settings

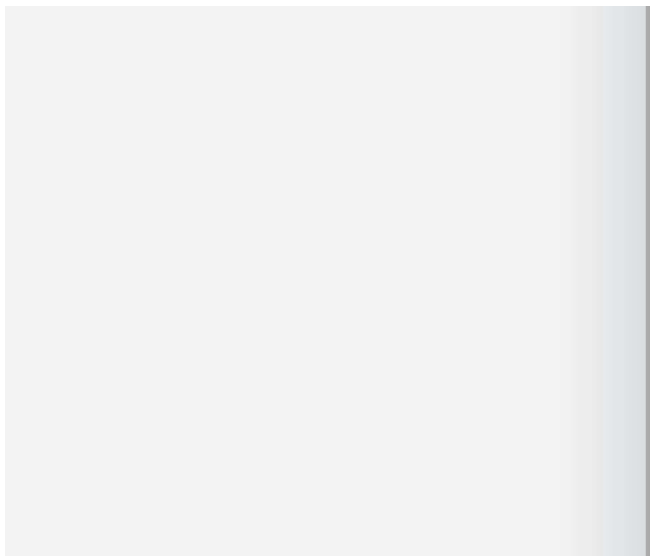
☰ Server connection collation: ⓘ utf8mb4\_unicode\_ci ▼

🔑 [More settings](#)

## Appearance settings

🗣️ Language ⓘ English ▼

🎨 Theme: pmahomme ▼



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 **utf8\_general\_ci** and click **Create**.

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

Proceed to Task B.



## Databases

**Create database**

	Database	Collation	Master replication	Action
<input type="checkbox"/>	information_schema	utf8_general_ci	✓ Replicated	Check privileges
<input type="checkbox"/>	mysql	utf8mb4_0900_ai_ci	✓ Replicated	Check privileges
<input type="checkbox"/>	performance_schema	utf8mb4_0900_ai_ci	✓ Replicated	Check privileges
<input type="checkbox"/>	sys	utf8mb4_0900_ai_ci	✓ Replicated	Check privileges
<b>Total: 4</b>				

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?

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

1. CREATE TABLE table_name (
2.     column1 datatype,
3.     column2 datatype,
4.     column3 datatype,
5.     ....
6. );
```

Copied!

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

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

1. ALTER TABLE table_name
2. ADD COLUMN column_name data_type column_constraint;
3.
4. ALTER TABLE table_name
5. DROP COLUMN column_name;
6.
7. ALTER TABLE table_name
8. ALTER COLUMN column_name SET DATA TYPE data_type;
9.
10. ALTER TABLE table_name
11. CHANGE current_column_name new_column_name;
```

Copied!

#### How does the syntax of a TRUNCATE statement look?

```
1. 1

1. TRUNCATE TABLE table_name;
```

Copied!

#### How does the syntax of a DROP statement look?

```
1. 1

1. DROP TABLE table_name;
```

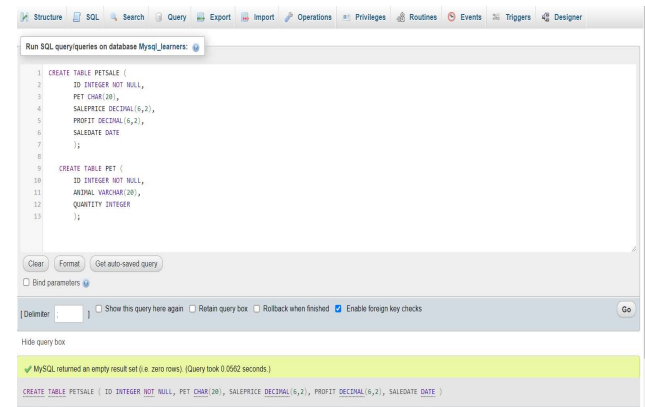
Copied!

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 PETSAL and PET, copy the code below and paste it to the textarea of the **SQL** page. 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 PETALE (
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!



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**.

```
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 PETALE 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!

1. INSERT INTO PETSale VALUES

2. (1, 'Cat', 450.00, 100.47, '2018-05-28'),

3. (2, 'Dog', 666.66, 150.76, '2018-06-01'),

4. (3, 'Parrot', 50.00, 8.90, '2018-06-04'),

5. (4, 'Hamster', 60.00, 12.00, '2018-06-11'),

6. (5, 'Goldfish', 48.48, 3.50, '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;

Showing rows 0 - 4 (5 total. Query took 0.0000 seconds.)

SELECT \* FROM PETSale

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

+ Options

ID	PET	SALEPRICE	PROFIT	SALEDATE
1	Cat	450.00	100.47	2018-05-28
2	Dog	666.66	150.76	2018-06-01
3	Parrot	50.00	8.90	2018-06-04
4	Hamster	60.00	12.00	2018-06-11
5	Goldfish	48.48	3.50	2018-06-14

Showing rows 0 - 2 (3 total. Query took 0.0000 seconds.)

SELECT \* FROM PET

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

+ Options

ID	ANIMAL	QUANTITY
1	Cat	3
2	Dog	4
3	Hamster	2

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

1. 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..

```
1. 1
2. 2
3. 3
4. 4
5. 1. ALTER TABLE PETSale
6. 2. ADD COLUMN QUANTITY INTEGER;
7. 3.
8. 4. SELECT * FROM PETSale;
```

Copied!

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0481 seconds.)

ALTER TABLE PETSale ADD COLUMN QUANTITY INTEGER

[Edit view][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 view][Edit][Explain SQL][Create PHP code][Refresh]

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

Options

ID	PET	SALEPRICE	PROFIT	SALEDATE	QUANTITY
1	Cat	450.00	100.47	2018-05-28	NULL
2	Dog	666.66	150.76	2018-06-01	NULL
3	Parrot	50.00	8.90	2018-06-04	NULL
4	Hamster	60.00	12.00	2018-06-11	NULL
5	Goldfish	48.48	3.50	2018-06-14	NULL

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 to textarea 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!

```
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	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	8.90	2018-06-04
4	Hamster	60.60	12.00	2018-06-11
5	Goldfish	48.48	3.50	2018-06-14

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 to the textarea 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!

Run SQL query/queries on table **Mysql\_learners.PETSALE**:

```

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

```

+ Options

ID	PET	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

☐ Show all | Number of rows: 25  Filter rows:

## Task C: ALTER using ALTER COLUMN

1. 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**.

```

1. 1
2. 2
1. ALTER TABLE PETSALE CHANGE PET PET VARCHAR(20);
2. SELECT * FROM PETSALE;

```

Copied!

Run SQL query/queries on table Mysql\_learners.PETSALE:

```

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

```

+ Options

ID	PET	SALEPRICE	SALEDATE
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

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int			No	None			Change Drop More
2	PET	varchar(20)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	SALEPRICE	decimal(6,2)			Yes	NULL			Change Drop More
4	SALEDATE	date			Yes	NULL			Change Drop More
5	QUANTITY	int			Yes	NULL			Change Drop More

## 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**.

```

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

```

Copied!

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)

Run SQL query/queries on table **Mysql\_learners.PETSALE**:

```
1 ALTER TABLE `PETSALE` CHANGE `PET` `ANIMAL` varchar(20);
```

✓ Showing rows 0 - 4 (5 total, Query took 0.0006 seconds.)

```
select * from `PETSALE`
```

☐ Show all | Number of rows: 25 ▼ Filter rows:

- Options

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

☐ Show all | Number of rows: 25 ▼ Filter rows:

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**.

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

Copied!

The screenshot shows a MySQL web interface with a top navigation bar containing icons and labels for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operations. Below the navigation bar, a status bar indicates the current context: "Run SQL query/queries on table Mysql\_learners.PET\$ALE:". The main area is a text editor with two lines of SQL code: "1 TRUNCATE TABLE PET ;" and "2 SELECT \* FROM PET;". To the right of the text editor, a panel displays the query results. At the top of this panel, a green message states: "MySQL returned an empty result set (i.e. zero rows)". Below this, the SQL query "SELECT \* FROM PET" is shown. Underneath the query, there are three column headers: "ID", "ANIMAL", and "QUANTITY". A button labeled "Query results operations" is visible, and at the bottom of the panel, there is a "Create view" button with a small icon.

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**.

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

Copied!

The screenshot shows a MySQL IDE interface. The top toolbar includes buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operations. Below the toolbar, a status bar indicates "Run SQL query/queries on table Mysql\_learners.PETSALE:". The main query editor contains two lines of SQL code:

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

On the right side, there is a panel with a "Delimit" dropdown set to ";" and a checkbox for "Show this query h". Below this is a "Hide query box" link. An "Error" message is displayed in a pink box:

**Error**

SQL query: [Copy](#) ⓘ

```
SELECT * FROM PET LIMIT 0, 25
```

**MySQL said:** ⓘ

#1146 - Table 'Mysql\_learners.PET' does not exist

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

[Lakshmi Holla](#)

[Malika Singla](#)

## Changelog

Date	Version	Changed by	Change Description
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

©BM Corporation 2021. All rights reserved.