Hands-on Lab: Built-in functions - Aggregate, Scalar, String, Date and Time Functions 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:

- Compose queries consting of built in functions and check the results.

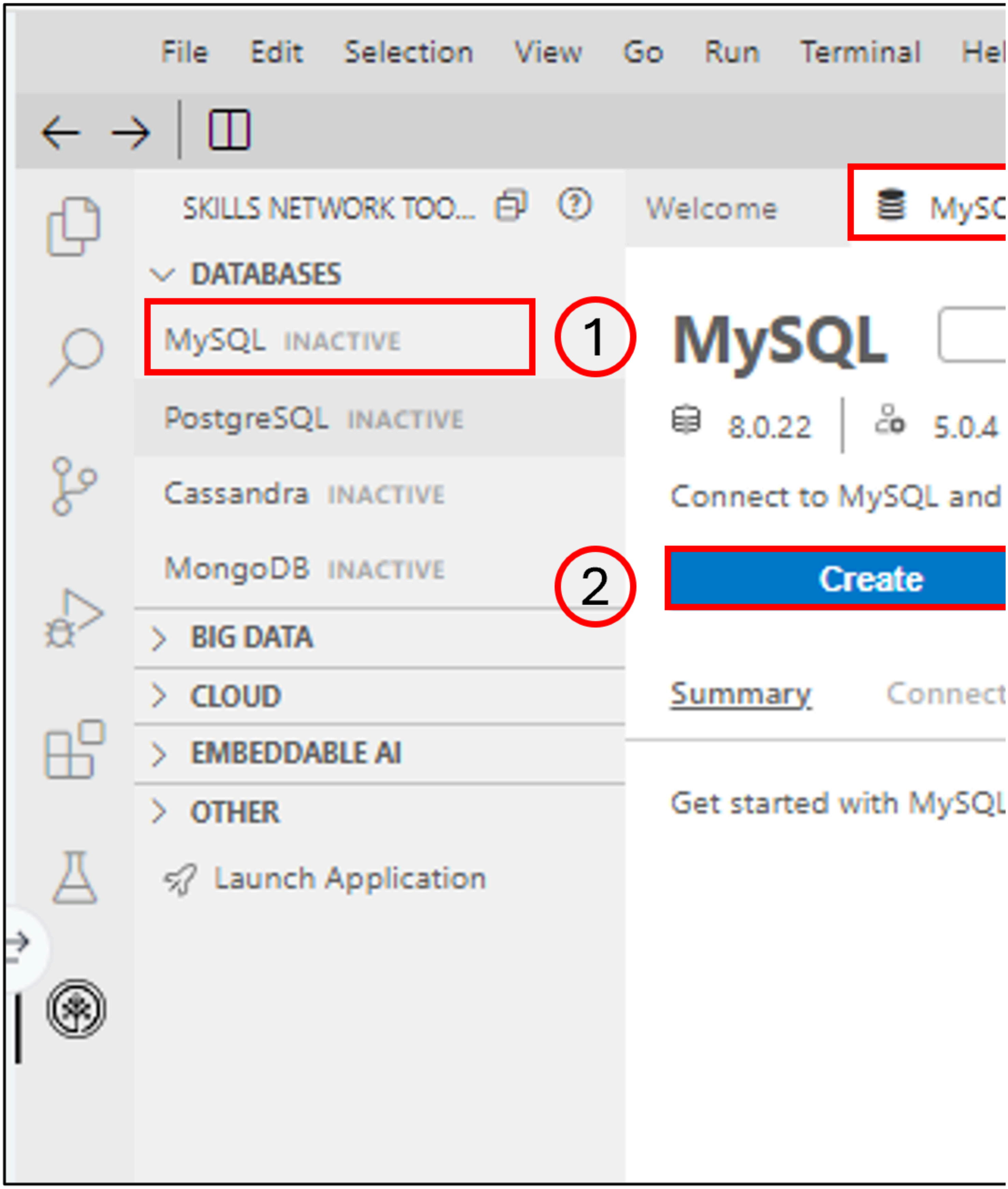
Exercise

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.

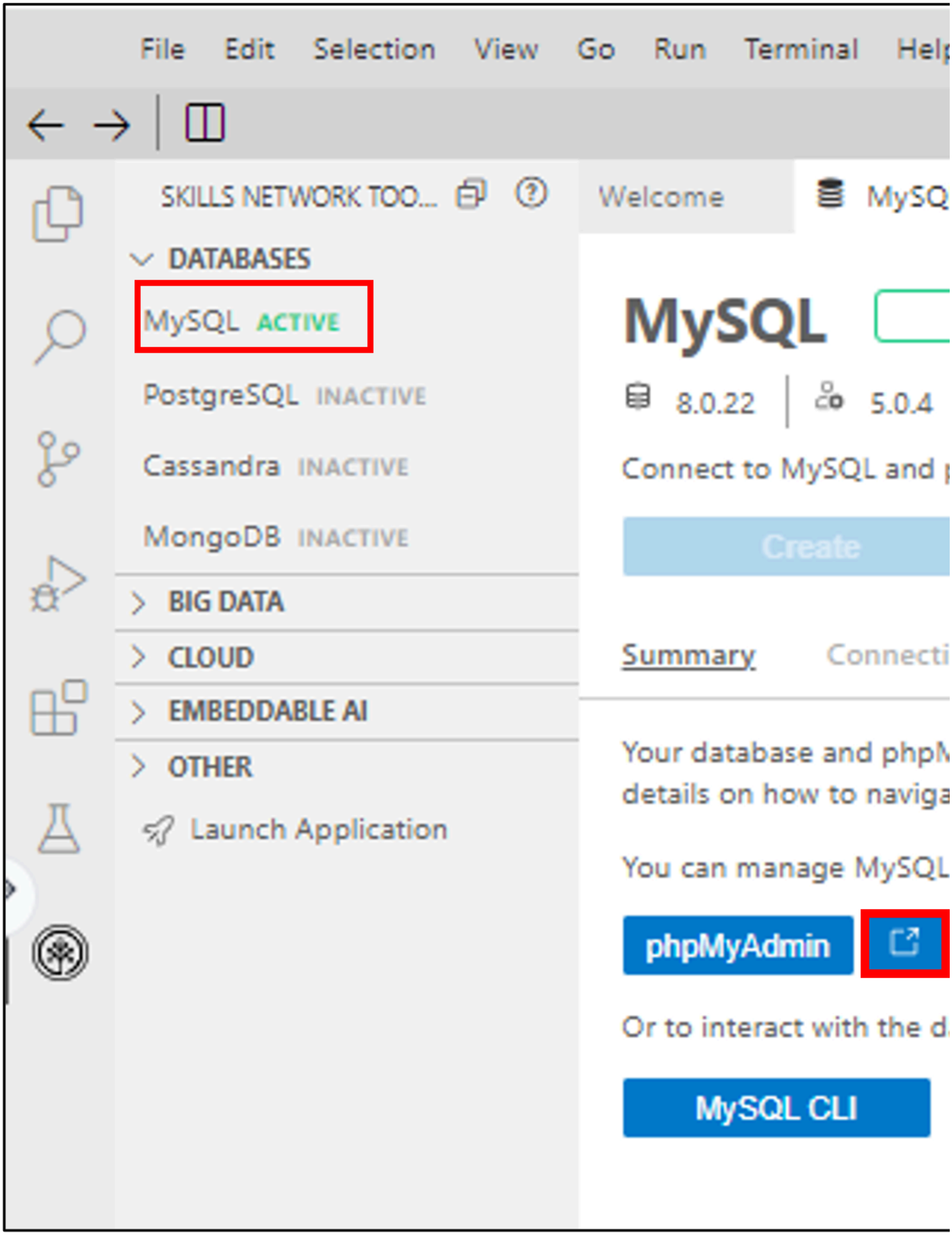
Task A: Create a database

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

To start the MySQL click **Create**.



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



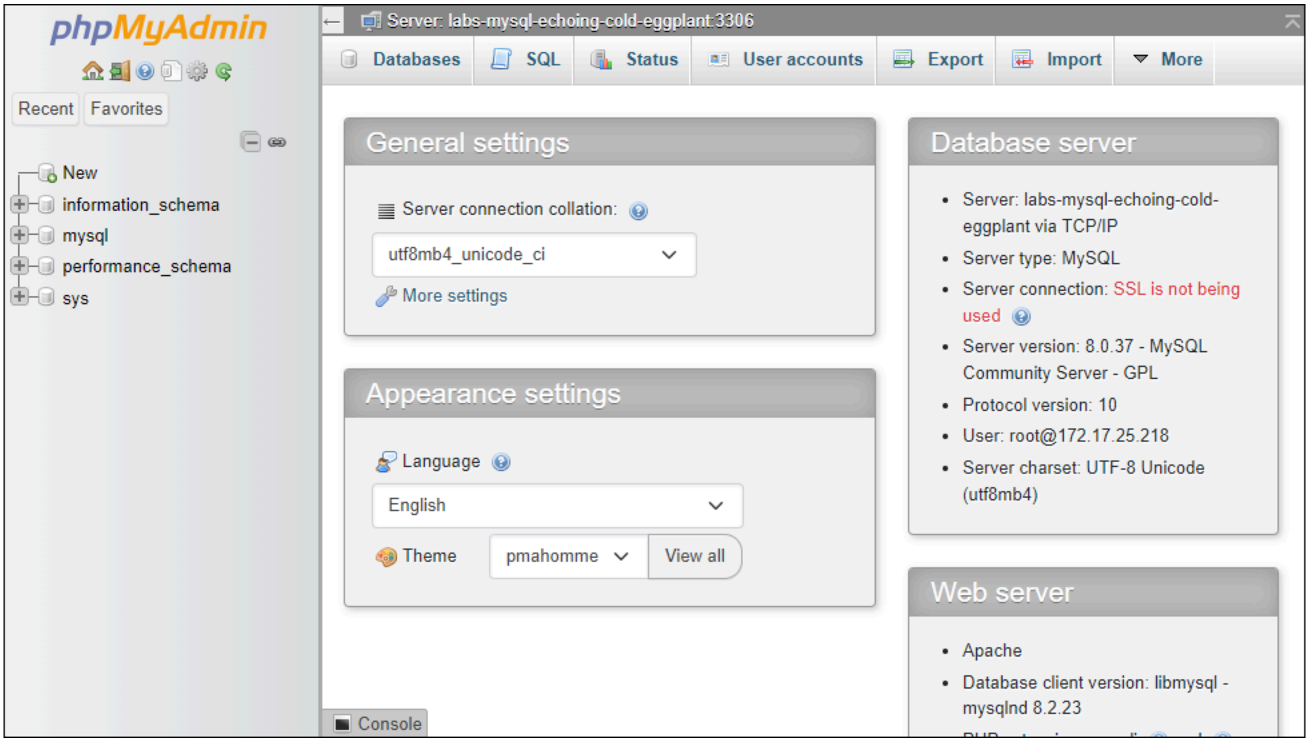
Connect to MySQL and

Your database and phpM
details on how to naviga

You can manage MySQL

Or to interact with the d

3. You will see the phpMyAdmin GUI tool.



General settings

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme View all

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)

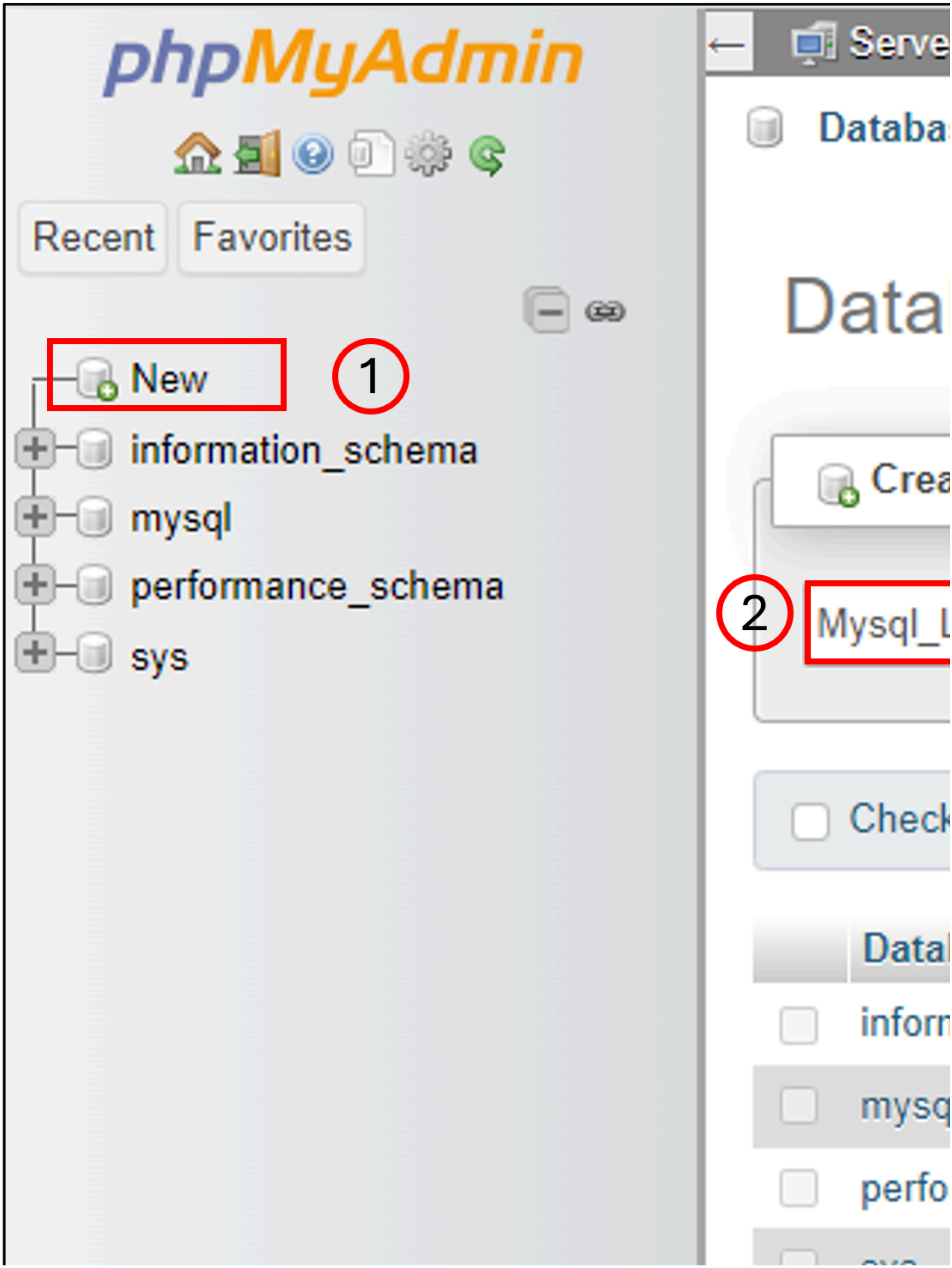
Web server

- 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 click **Create**.

   The encoding will be left as **utf8mb4_0900_ai_ci**. 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
Serve
Databa
Data
Crea
2
Mysql_L
Check
Data
inforr
mysq
perfo
sys

Total: 4

Compose and run the following queries in the textarea of the **SQL** page. Click **Go** to execute the queries and observe the the results.

Note: The solutions are provided at the end of this lab, but please try to compose the queries on your own before checking the solutions.

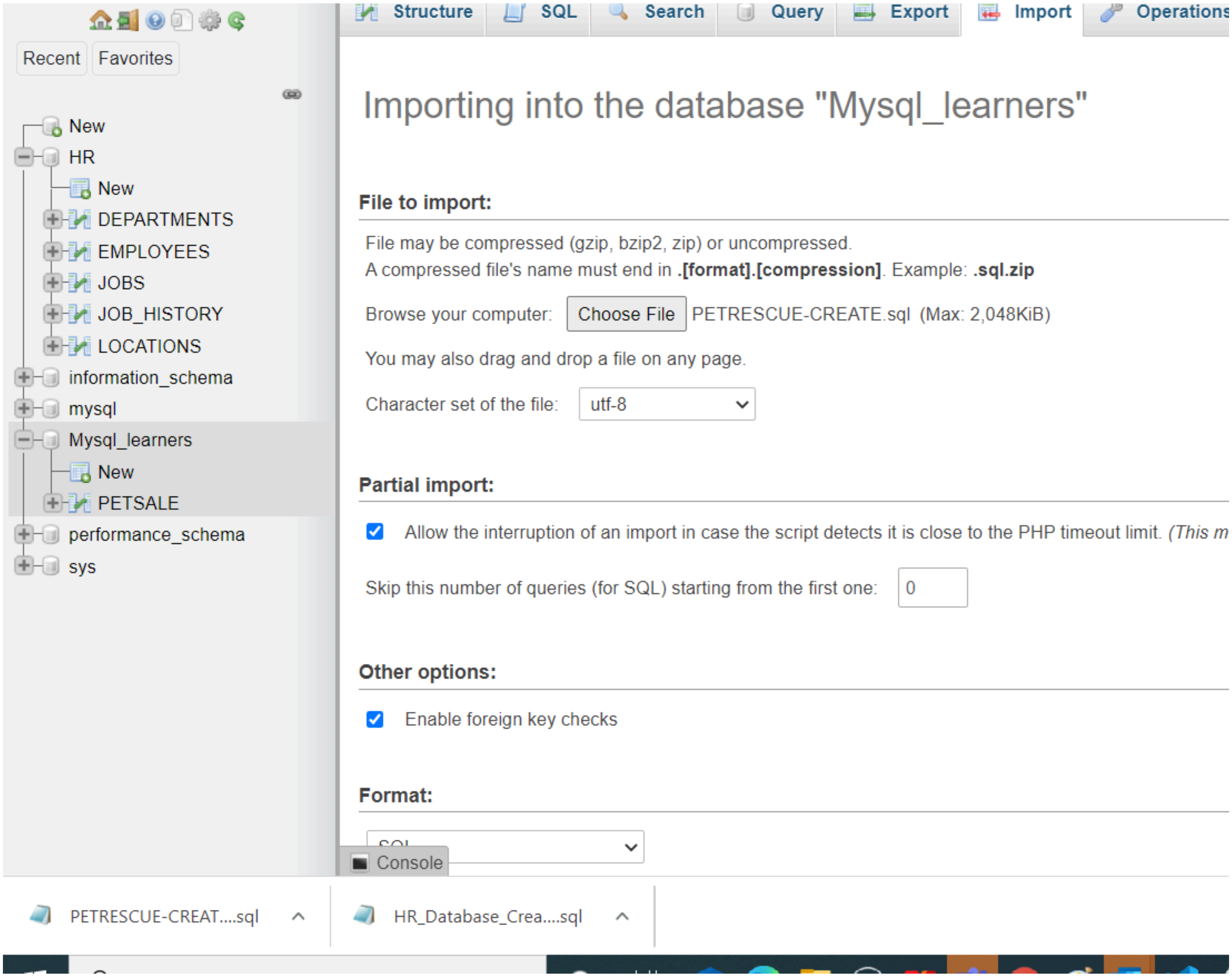
Exercise 1: Create the Pet Rescue table

Rather than create the table manually by typing the DDL commands in the SQL editor, you will execute a script containing the create table command.

1. Download the script file PETRESCUE-CREATE.sql

Note: To download, just right-click on the link above and click on **Save As..** or **Save Link As…** depending on your browser. Save the file as a .sql file and not HTML.

2. Next load the sql to your database using the Import option.



Importing into the database "Mysql_learners"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: PETRESCUE-CREATE.sql (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This m*

Skip this number of queries (for SQL) starting from the first one: 0

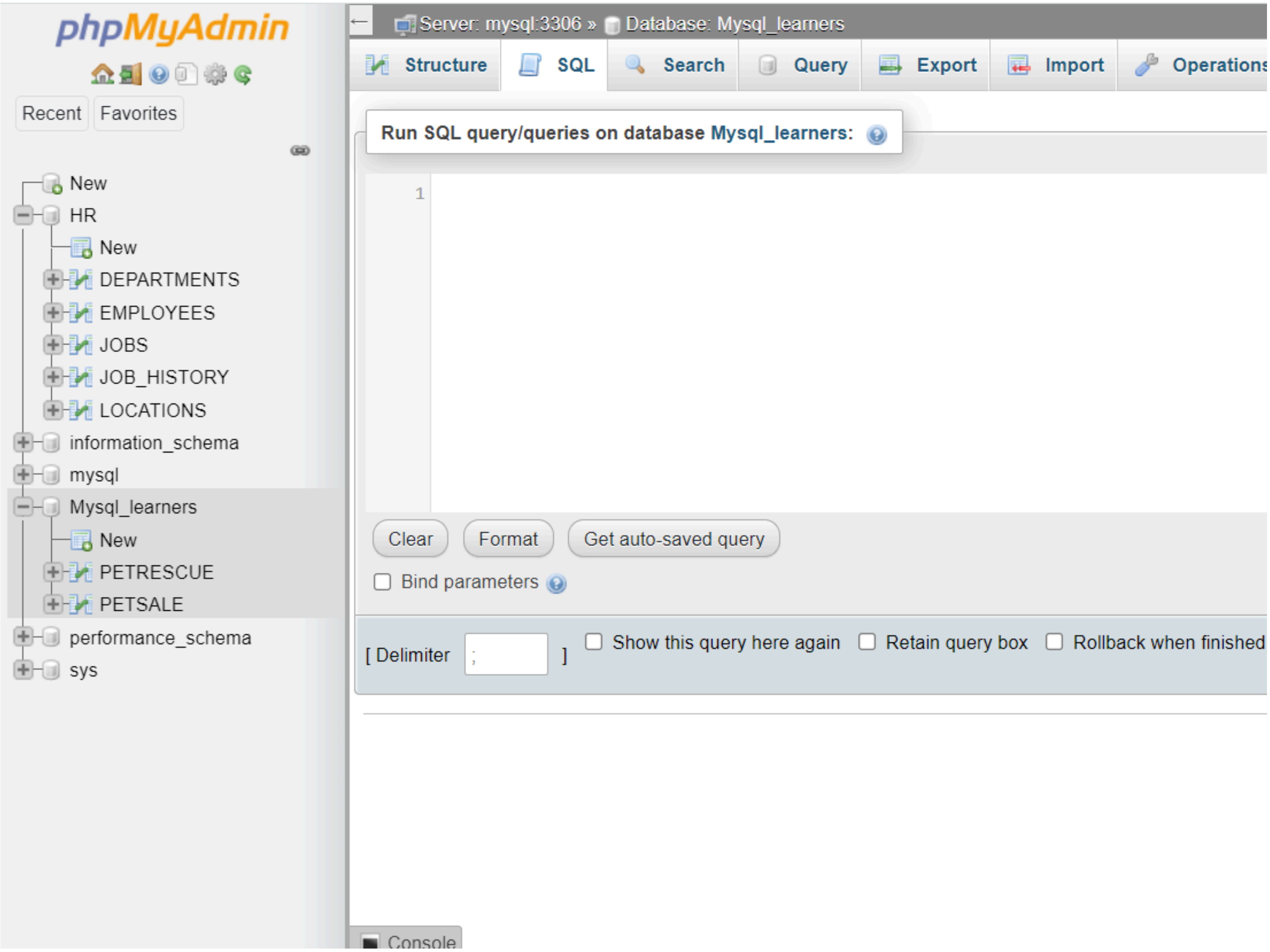
Other options:

☑ Enable foreign key checks

Format:

SQL

3. Once the table is loaded open the sql editor to start executing the functions.



Exercise 2: Aggregate Functions

Query A1: Enter a function that calculates the total cost of all animal rescues in the PETRESCUE table.

Query A2: Enter a function that displays the total cost of all animal rescues in the PETRESCUE table in a column called SUM_OF_COST.

Query A3: Enter a function that displays the maximum quantity of animals rescued.

Query A4: Enter a function that displays the average cost of animals rescued.

Query A5: Enter a function that displays the average cost of rescuing a dog.

Exercise 3: Scalar and String Functions

Query B1: Enter a function that displays the rounded cost of each rescue.

Query B2: Enter a function that displays the length of each animal name.

Query B3: Enter a function that displays the animal name in each rescue in uppercase.

Query B4: Enter a function that displays the animal name in each rescue in uppercase without duplications.

Query B5: Enter a query that displays all the columns from the PETRESCUE table, where the animal(s) rescued are cats. Use **cat** in lower case in the query.

Exercise 4: Date and Time Functions

Query C1: Enter a function that displays the day of the month when cats have been rescued.

Query C2: Enter a function that displays the number of rescues on the 5th month.

Query C3: Enter a function that displays the number of rescues on the 14th day of the month.

Query C4: Animals rescued should see the vet within three days of arrivals. Enter a function that displays the third day from each rescue.

Query C5: Enter a function that displays the length of time the animals have been rescued; the difference between today's date and the rescue date.

Lab Solutions

Exercise 2: Aggregate Functions

Query A1: Enter a function that calculates the total cost of all animal rescues in the PETRESCUE table.

```
select SUM(COST) from PETRESCUE;
```

Query A2: Enter a function that displays the total cost of all animal rescues in the PETRESCUE table in a column called SUM_OF_COST.

```
select SUM(COST) AS SUM_OF_COST from PETRESCUE;
```

Query A3: Enter a function that displays the maximum quantity of animals rescued.

```
select MAX(QUANTITY) from PETRESCUE;
```

Query A4: Enter a function that displays the average cost of animals rescued.

```
select AVG(COST) from PETRESCUE;
```

Query A5: Enter a function that displays the average cost of rescuing a dog.
Hint - Bear in my the cost of rescuing one dog on day, is different from another day. So you will have to use and average of averages.

```
select AVG(COST/QUANTITY) from PETRESCUE where ANIMAL = 'Dog';
```

Exercise 3: Scalar and String Functions

Query B1: Enter a function that displays the rounded cost of each rescue.

```
select ROUND(COST) from PETRESCUE;
```

Query B2: Enter a function that displays the length of each animal name.

```
select LENGTH(ANIMAL) from PETRESCUE;
```

Query B3: Enter a function that displays the animal name in each rescue in uppercase.

```
select UCASE(ANIMAL) from PETRESCUE;
```

Query B4: Enter a function that displays the animal name in each rescue in uppercase without duplications.

```
select DISTINCT(UCASE(ANIMAL)) from PETRESCUE;
```

Query B5: Enter a query that displays all the columns from the PETRESCUE table, where the animal(s) rescued are cats. Use **cat** in lower case in the query.

```
select * from PETRESCUE where LCASE(ANIMAL) = 'cat';
```

Exercise 4: Date and Time Functions

Query C1: Enter a function that displays the day of the month when cats have been rescued.

```
select DAY(RESCUEDATE) from PETRESCUE where ANIMAL = 'Cat';
```

Query C2: Enter a function that displays the number of rescues on the 5th month.

```
select SUM(QUANTITY) from PETRESCUE where MONTH(RESCUEDATE)='05';
```

Query C3: Enter a function that displays the number of rescues on the 14th day of the month.

```
select SUM(QUANTITY) from PETRESCUE where DAY(RESCUEDATE)='14';
```

Query C4: Animals rescued should see the vet within three days of arrivals. Enter a function that displays the third day from each rescue.

```
select DATE_add(RESCUEDATE, INTERVAL 3 DAY) from PETRESCUE;
```

Query C5: Enter a function that displays the length of time the animals have been rescued; the difference between today's date and the rescue date.

```
select DATEDIFF(CURRENT_TIMESTAMP,RESCUEDATE) from PETRESCUE;
```

Summary

You can now compose and run queries, check results and view the logs. You will use these skills in later labs.

Author(s)

Lakshmi Holla

Malika Singla

Skills Network