

Create a database and a table

Welcome to this lab activity

In this lab activity, you will explore how to use the MySQL shell and how to create a database and a table with it. You will then insert some data into the table and query it to retrieve the data.

The method by which you access the MySQL shell will differ based on your setup. Please refer to the installation instructions earlier in this module for details.

Task 1: Start the MySQL interactive shell

Start the MySQL shell, logging in with the root user and password.

When you start the MySQL shell, you should see the MySQL prompt:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Task 2: Creating a new database

Before creating a new database, let's check which databases are already set up in your MySQL server.

1. Run the following command in the MySQL shell:

```
SHOW DATABASES ;
```

type this command and press **Enter**.

You will see that you already have some databases in MySQL; these are just system-managed databases, and you can ignore them:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sys        |
+-----+
4 rows in set (0.00 sec)
```

2. Now create a new database called `myBookshop`.

In the MySQL shell, run the following command:

```
CREATE DATABASE myBookshop;
```

This command will create a new database called `myBookshop` in your virtual server.

In the above, `myBookshop` is the name of the database. You may call your database whatever you like, but `myBookshop` is the name we will use in the remaining instructions.

Please note: some names are case-sensitive; `myBookshop` is different from `mybookshop`! Please use the same names as instructed on the lab instructions to avoid any confusion. Commands, on the other hand, are not case-sensitive; `CREATE` is the same as `create`.

3. Now that you have created your new database, you can view it by running the following command once again:

```
SHOW DATABASES;
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| myBookshop |
| mysql      |
| performance_schema |
| sys        |
+-----+
5 rows in set (0.00 sec)
```

As you can see the `myBookshop` database is now part of your databases list.

Task 3: Creating a table in your database

It is now time to add a table to your newly created database.

4. First, you need to do is to select your `myBookshop` database. In the MySQL shell, enter the following command:

```
USE myBookshop;
```

This command will switch to the `myBookshop` database.

If you have successfully switched to the database, you will receive the following confirmation:

```
mysql> USE myBookshop;
Database changed
mysql> █
```

5. Now that you have correctly switched to your `myBookshop` database, go ahead and create a table name `books` by running the following command on the Terminal:

```
CREATE TABLE books (
id INT AUTO_INCREMENT,
name VARCHAR(50),
price DECIMAL(5, 2),
PRIMARY KEY(id));
```

This command will create a new table called `books` in your currently selected database.

Note that when you press ENTER, MySQL will change the prompt to `'->'` to indicate that the command has not completed. The command will only be completed when you enter the semi-colon at the end.

If you have successfully created a new table, you will get the following confirmation:

```
mysql> CREATE TABLE books (
-> id INT AUTO_INCREMENT,
-> name VARCHAR(50),
-> price DECIMAL(5, 2),
-> PRIMARY KEY(id));
Query OK, 0 rows affected (0.02 sec)
```

6. You can now view your newly created table by running the following command:

```
SHOW TABLES;
```

This command will show you all the tables in your currently selected database.

```
+-----+
| Tables_in_mybookshop |
+-----+
| books                 |
+-----+
1 row in set (0.00 sec)
```

7. Finally, view the structure of the `books` table. Run the following command:

```
DESCRIBE books;
```

```
mysql> DESCRIBE books;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50)   | YES  |     | NULL    |                |
| price | decimal(5,2)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

You can see from the above picture that you have successfully created a new table called `books` with 3 fields (`id`, `name`, `price`) in your `myBookshop` database.

Task 4: Adding dummy data to your database

At the moment the `books` table does not contain any data. It only has the data structure for each entry. Let's see how you can add data to the `books` table.

8. Run the following command in the MySQL shell:

```
INSERT INTO books (name, price)
VALUES('Don Quixote', 19.99),
('Atlas of the World', 25.00),
('World History', 31.99);
```

This command will insert three new books in your database.

If you have successfully run the above command, you will receive the following confirmation message:

```
mysql> INSERT INTO books (name, price)
-> VALUES('Don Quixote', 19.99),
-> ('Atlas of the World', 25.00),
-> ('World History', 31.99);
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

You have successfully added three books to your `books` table in your `myBookshop` database.

9. Add two more books of your choosing to the `books` table using the `INSERT` command.

Task 5: Query the data in the database

Let's now see how you can access the data that you entered into the `books` table.

In order to read data from a database with SQL, you need to use the `SELECT` statement.

The basic format of the `SELECT` statement looks like this:

```
SELECT fileName1, fileName2 FROM TableName;
```

`fileName1` and `fileName2` are the field names that you want to retrieve and `TableName` is the name of the table where you want to retrieve the data from.

You may also use the wildcard (*) to return all the fields in a table:

```
SELECT * FROM TableName;
```

10. To select all the books from the books table, you can run the following command in the MySQL shell:

```
SELECT * FROM books;
```

You should get the following result:

```
mysql> SELECT * FROM books;
+----+-----+-----+
| id | name          | price |
+----+-----+-----+
|  1 | Don Quixote   | 19.99 |
|  2 | Atlas of the World | 25.00 |
|  3 | World History | 31.99 |
+----+-----+-----+
3 rows in set (0.00 sec)
```

11. Now try running the following command:

```
SELECT * FROM books LIMIT 2;
```

Here, the asterisk (*) is a wildcard, which means 'all'. Including the `LIMIT` clause restricts how many records (rows) are returned in the result set. By running the above command, you should only see the first two books in the database.

`SELECT` can be combined with various **clauses** to restrict/filter the records returned in the result set. For example, you can use the `WHERE` clause to return only records matching specific criteria.

12. Try running the following query:

```
SELECT name, price FROM books WHERE id=2;
```

What do you see?

13. Now that you are familiar with the basics of SQL queries, try to write a statement to retrieve all the books that cost more than £20.

Task 6: Exit MySQL shell

- 14.** Exiting the MySQL shell is very straightforward. In your MySQL shell, type the following command:

```
exit
```

Note that your database, table and data will remain stored in the MySQL database on your system, ready for you when you log into MySQL shell again.

Task 7: Explore further

When tackling these lab activities, it's always good to stretch yourself by doing some and attempting some changes on your own.

Can you add two more books to your 'books' table?

End of lab

Congratulations on completing this section.

You have successfully connected to the MySQL shell and created a database and a table within the database. You then added some data to the table and queried it in various ways.

In the next lab, you will practise these operations on a new database.