Join in SQL

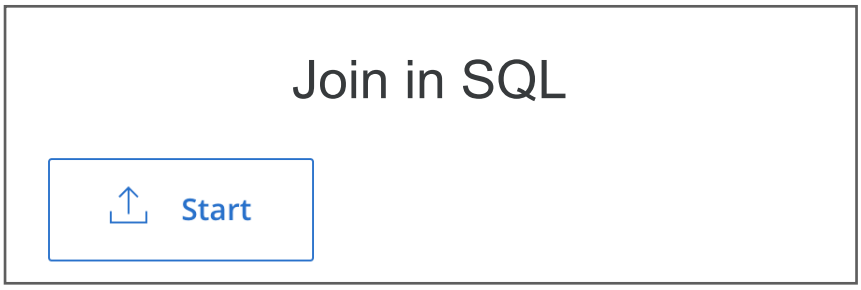
Welcome to this Lab activity

In this lab activity you will be exploring how to add more than one table to your database. Furthermore, you will see how to setup the tables with foreign keys and insert dummy data. Finally you will execute some basic and advanced SQL statements including JOIN to retrieve details about the schema and the data it contains. For the purpose of this lab you will be working with the Terminal panel inside Visual Studio Code.

Start the Lab environment application

It is simple to launch a lab exercise. You only need to click on the button “Start” below the activity title to enter a lab environment.

Let’s explore this lab activity. Go ahead and click on the “Start” button!



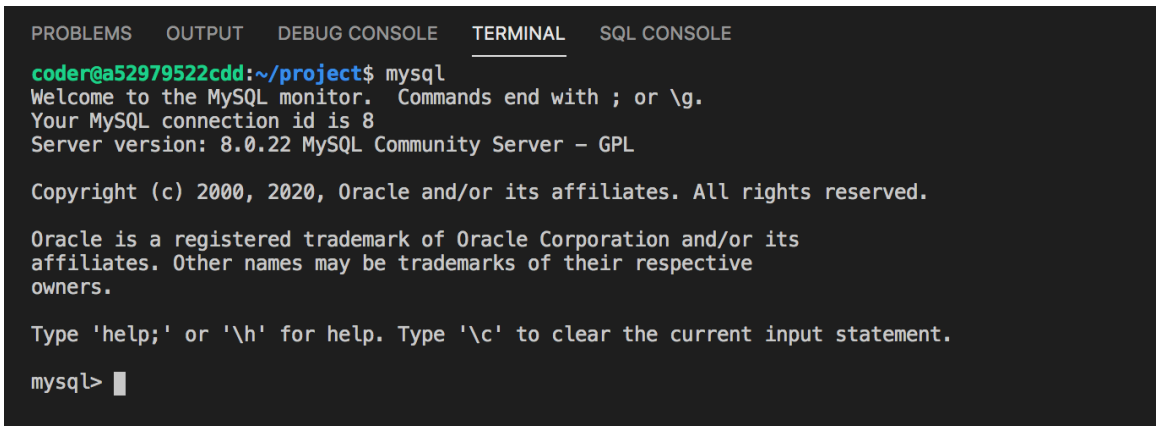
Task 1: Accessing the MySQL interactive shell

The folder structure has already been partially constructed for you and organised into different topics. For the purpose of this lab, you are not required to make any changes to the folder structure. You can see a folder called “topic8” inside this lab environment; it is only there as a reference for you and you are not required to add any content to it. Let’s get started!

In order to access your mysql interactive shell use the Visual Studio Code Terminal and run the following command:

- **mysql**: type this command and press *Enter*. This command will log you into mysql shell as the root(default) user.

If you have successfully followed all the above steps you should now be logged in inside mysql and see the following result on the Terminal:



Task 2: Create a new database

- First see what databases you already have linked to your virtual server:

SHOW DATABASES;

- You will already have a couple, including the “myBookshop” database you created previously.
- To create a new database, use:

CREATE DATABASE mysecondBookshop;

In the above, “mysecondBookshop" is the name of the new database. You can call your database what you like, this name is suggested here because it relates to the database design we discussed before.

- To check if your new database was successfully created, use:

SHOW DATABASES;

Task 3: Create tables

- Switch to the new database you created:

USE <DATABASE_NAME>;

- Replace <DATABASE_NAME> with the name of your database (“mysecondBookshop”)
- Create a table named “Publisher” based on the database design you see below, what kind of field types are the best to be used for this table?

I'll give you the answer for the first table:

CREATE TABLE Publisher (id INT AUTO_INCREMENT, name VARCHAR(50), address VARCHAR(100) , PRIMARY KEY(id));

- Can you create another table named “Book” based on the database design you can see above?
- What sort of association would you use to connect publisher and book tables?
- Please note that the “Book” table includes a foreign key to the publisher table. How can you create a table including foreign keys?

CREATE TABLE Book (id INT AUTO_INCREMENT, publisher_id INT, name VARCHAR(50), price DECIMAL(5, 2) unsigned, PRIMARY KEY (id), FOREIGN KEY(publisher_id) REFERENCES Publisher(id));

- See what tables were created:

SHOW TABLES;

- See what fields have been defined within a given table:

DESCRIBE TableName;

Replace TableName with the name of your tables.

Task 4: Input the dummy data

- To insert some dummy data into the database, you need to use INSERT INTO statement, If you are not sure how to do this refer to the “Insert dummy data in the database” lab instructions.
- Insert dummy data into the Publisher table, including the rows that you can see in database design above.

INSERT INTO Publisher (name, address) VALUES ('McGrawHill', 'Somewhere1');

- Insert dummy data into book. I'll give you an example of how you can insert data in a table (book) with a foreign key:

INSERT INTO Book (publisher_id, name, price) VALUES ((SELECT id FROM Publisher WHERE Publisher.name = 'McGrawHill'), 'Database Book', 40.25);

- Here we are inserting a new record for a book named ‘Database Book’, with a price of 40.25£ and published by ‘McGrawHill’

- Add two more publishers with the following data to your database:

  name: ‘Oxford press’
  address: ‘somewhere2’

  name: ‘Cambridge press’
  address: ‘somewhere3’

- Add two more books to your book table with the following data to your database:

  publisher: ‘Cambridge press’
  name: ‘Node.js book’
  price: 25

  publisher: ‘Oxford press’
  name: ‘Express book’
  price: 31.99

Task 5: Query the data in mySQL shell

Now that you have inserted the data, you are able to perform SQL queries on it.
You may use the wildcard (*) to return all the fields in a table:

SELECT * FROM TableName;

- See what data is in the “Book” and “Publisher” tables.

- List all the books published by “McGrawHill"

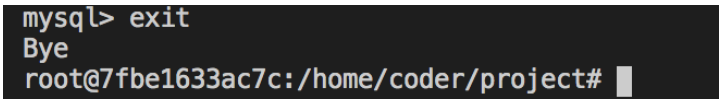
SELECT Book.name, Book.price FROM Book JOIN Publisher ON Book.publisher_id = Publisher.id WHERE Publisher.name = 'McGrawHill';

Task 6: Exit mysql shell

Exiting the mysql shell is very straight forward. In your Terminal panel type the following command:

- **exit**: type this command and press *Enter*. This command will log you out from your mysql virtual server.

If you have successfully exited the database you will get the following confirmation:



End of Section

Congratulations for completing this section. In the next lab activity you will be practicing more with database operations.