# Exercise: Create optimized queries to manage and analyze data

## **Scenario**

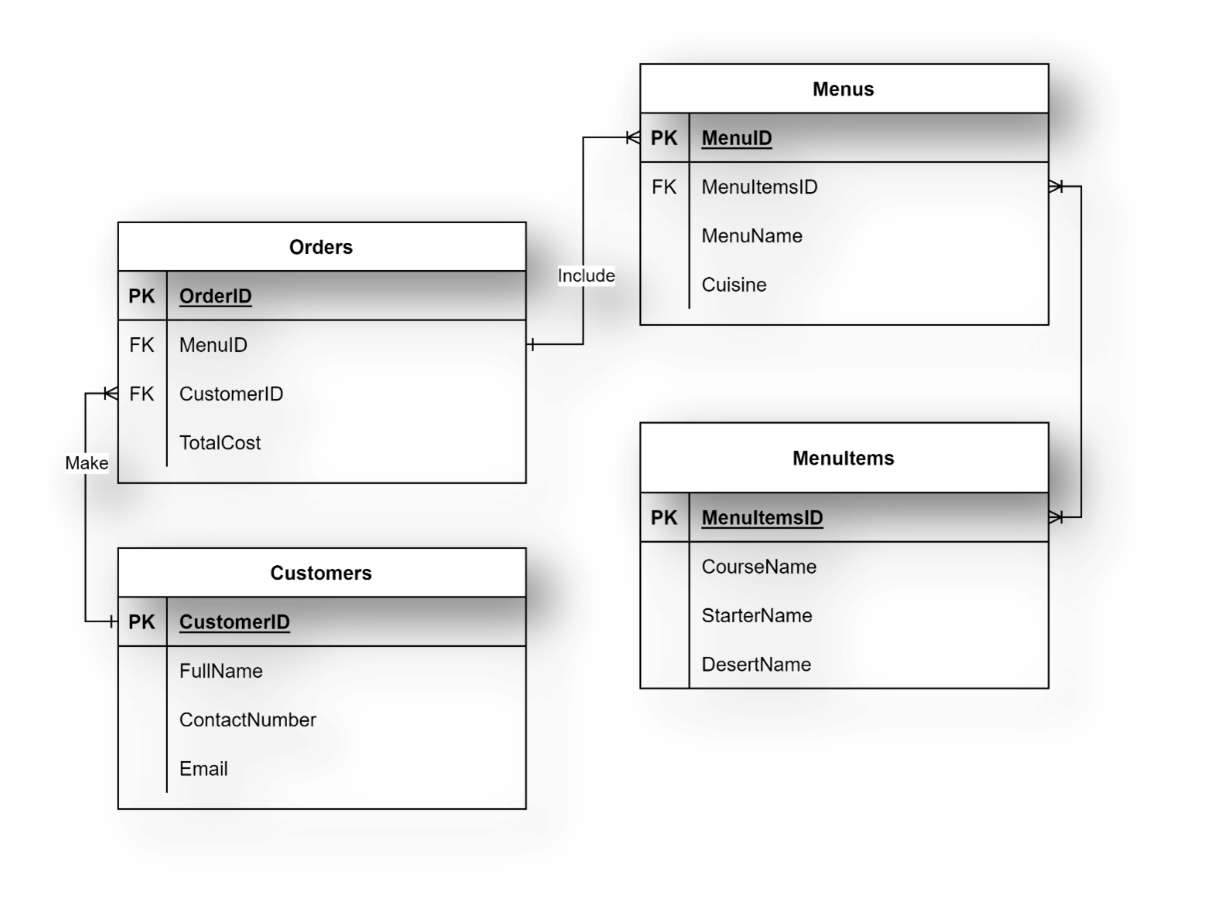
Little Lemon need to query the data in their database. To do this, they need your help with creating optimized queries using stored procedures and prepared statements.

## **Prerequisites**

In the previous module, you developed a data model for Little Lemon and implemented it in your MySQL server. Your database should now contain several tables including the following:

* Menus,
* Orders,
* MenuItems,
* and Customers.

These tables might have different names within your database. However, they should still contain similar data to, and resemble the relationships in, the following diagram:



You also need to use MySQL Workbench SQL editor to write the required stored procedures and prepared statements.

## **Task 1**

In this first task, Little Lemon need you to create a procedure that displays the maximum ordered quantity in the Orders table.

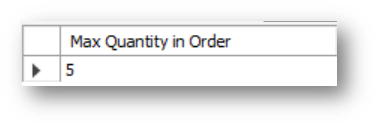
Creating this procedure will allow Little Lemon to reuse the logic implemented in the procedure easily without retyping the same code over again and again to check the maximum quantity.

You can call the procedure GetMaxQuantity and invoke it as follows:

1

CALL GetMaxQuantity();

The output result of your query (depending on the data populated in your database) should be similar to the output in the following screenshot:



## **Task 2**

In the second task, Little Lemon need you to help them to create a prepared statement called GetOrderDetail. This prepared statement will help to reduce the parsing time of queries. It will also help to secure the database from SQL injections.

The prepared statement should accept one input argument, the CustomerID value, from a variable.

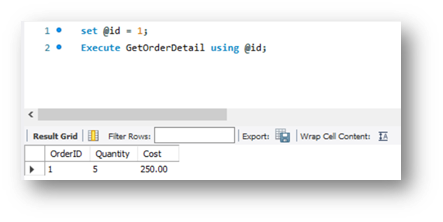
The statement should return the order id, the quantity and the order cost from the Orders table.

Once you create the prepared statement, you can create a variable called id and assign it value of 1.

Then execute the GetOrderDetail prepared statement using the following syntax:

SET @id = 1;

EXECUTE GetOrderDetail USING @id;

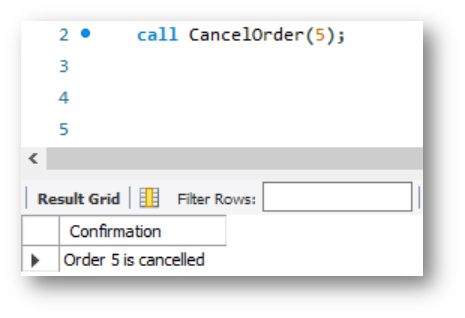


## **Task 3**

Your third and final task is to create a stored procedure called CancelOrder. Little Lemon want to use this stored procedure to delete an order record based on the user input of the order id.

Creating this procedure will allow Little Lemon to cancel any order by specifying the order id value in the procedure parameter without typing the entire SQL delete statement.

If you invoke the CancelOrder procedure, the output result should be similar to the output of the following screenshot:



## **Conclusion**

In this exercise, you helped Little Lemon to query the data in their database by creating optimized queries using stored procedures and prepared statements.