

SQL3 - Assessment

Introduction

This week you will be retrieving the records from the Food Order App database.

All the create and insert operations should be performed using **MySQL**. Before proceeding with the task make sure you have the Food Order App database created with enough records in the database.

You can follow the below mentioned steps to create the database and insert records in the database -

1. Download and extract the **SQL3 Assessment** folder from the zip folder provided under the Source Code section. All the necessary files are provided within this folder.
2. Open **MySQL Workbench**.
3. In MySQL Workbench, open the **FoodOrderApp_DDL Commands.sql** file provided in the Source code folder and execute this file to create the database and the tables in the database.
4. You can refer to the **FoodOrderApp_Database Design Document.pdf** and **FoodOrderApp_ER Diagram.pdf** files provided in the source code to understand the design of the database.
5. Now open the **FoodOrderApp_Insert Commands.sql** file in the MySQL workbench and execute it to insert the records in the tables.
6. Now open the **SelectQueries1.sql** file to write and execute the SQL queries as mentioned in the task1 below.
7. Then open the **SelectQueries2.sql** file to write and execute the SQL queries as mentioned in the task2 below.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

Your task is as follows -

1. **Retrieve records (SelectQueries1.sql file) (3 points each)**
 - 1.1. Display all cuisine names.
 - 1.2. Display description of 'Cake' category.
 - 1.3. Select all restaurants where restaurant names end with 'Junction'.
 - 1.4. Display details of all vegetarian food items.
 - 1.5. Retrieve all menu items with food items that have a price equal to or less than \$10.
 - 1.6. Retrieve cartId having the maximum number of food items.
 - 1.7. Retrieve all the shipping details where emailId contains 'doe'.
 - 1.8. Retrieve the order with the highest price.
 - 1.9. Retrieve the most recently updated order.
 - 1.10. How many total units of food items were purchased in orderId 1?

2. **Retrieve records (SelectQueries2.sql file) (6 points each)**
- 2.1. Retrieve number of food items for each cuisine
 - 2.2. Retrieve category names in the order of highest to lowest no. of food items
 - 2.3. Retrieve cuisine name as CuisineName, food item name as name FoodItemName and description as Description of all the food items belonging to 'Italian' cuisine.
 - 2.4. Retrieve details of food items, including their name, vegetarian status, and cuisine name, for only those food items that are vegetarian.
 - 2.5. Retrieve the details of users who have items in their cart along with the total order amount.

Program Organization

- You will be getting a zip folder containing a folder named **SQL3 Assessment** which has all the required files.
- The **FoodOrderApp_ER Diagram.pdf** and **FoodOrderApp_Database Design Document.pdf** are for your reference which demonstrates the basic design of all the tables of the **food_order_app** database.
- The **FoodOrderApp_DDL Commands.sql** file contains the commands to create the **food_order_app** database and to create the tables in the database.
- The **FoodOrderApp_Insert Commands.sql** file contains the commands to insert records in the tables in the database.
- The **SelectQueries1.sql** and **SelectQueries1.sql** files are provided which you need to modify to write the select queries as mentioned in the problem statement.

Evaluation Rubric

Total Project Points: 60

- Correctness:
Correctness of implementation
 - Problem statement - point 1 (50%) : 30 Points
 - Problem statement - point 2 (50%) : 30 Points

Program Instructions

- The **SQL3 Assessment** folder should have all the SQL files.
- The **SQL3 Assessment** folder should be compressed to zip/rar.
- Project will not be evaluated if the submitted project is not in the zip/rar format.