**Database Application Development**

Fall 2022

Group Project

**A Restaurant Management DB System**

Assume that your team has been in contract with the headquarters of a company that owns several restaurants in different states in the US. Your team is to provide software that manages these restaurants, by performing several operations in the software. These operations actually manipulate various components of a database. Example operations are to establish a new restaurant, update a menu, hire restaurant employees, place an order by a customer, provide financial statements, etc. In addition, there are reports to be run for the managers of the restaurants, to observe how the business is doing in a specific state, or during a specific month or quarter of the year, etc.

Your team will implement the above by designing a database, creating the appropriate tables, and then writing, testing and deploying PL/SQL stored procedures that implement the operations and the reports.

This project is a group project. The work is to be divided into tasks which are to be distributed among the group members, each one will work individually on their own tasks and also consolidate the individual tasks with the tasks of the other group members into an integrated project. The entire project is to be completed incrementally based on deliverables. Some deliverables are group deliverables meaning that the entire group must work together to accomplish them, while other deliverables are individual where each group member is to work on their own and upload their completed deliverable separately.

**TABLES**

**Cuisine Types**: Contains a cuisine type ID and the cuisine type names: American, Indian, Italian, BBQ, Ethiopian.

**Restaurants**: Each restaurant has an ID, name, street address, city, state, zip, and the cuisine type it specializes in.

**Waiters**: Each waiter has an ID, name, ID of restaurant they work at.

**Menu Items**: This is the food that is served in each restaurant and shows up as items in the menu. Contains cuisine type ID, menu item ID, name of item for each cuisine type, and price.

* American cuisine menu items: burger, fries, pasta, salad, salmon
* BBQ cuisine menu items: steak, pork loin, fillet mignon
* Indian cuisine menu items: dal soup, rice, tandoori chicken, samosa
* Italian cuisine menu items: lasagna, meatballs, spaghetti, pizza
* Ethiopian cuisine menu items: meat chunks, legume stew, flatbread

Since there is one large company that owns all these restaurants, there is a single consolidated menu. So for simplicity let us assume that:

* All restaurants of the same type (e.g. BBQ) have the same plates (steak, burger, etc.)
* The same plate has the same price across all restaurants that offer it.

**Restaurant Inventory**: Restaurants need to stock food items (menu items) to serve to customers. This table has food inventory for each restaurant. Include menu item ID, menu item name, restaurant ID, and quantity for the menu item. This table is populated with information from the Menu Items table. Every time food is ordered from the menu, the appropriate quantity must be reduced accordingly.

**Customers**: Contains customer ID, name, email, street address, city, state, zip, credit card number.

**Orders**: This table contains information about an order of a customer at a restaurant. Include order ID, restaurant ID, customer ID, order date, menu item ID, waiter ID, amount paid (order amount without tip), tip (calculated as 20% of the order amount).

Assumptions: Each order contains a single menu item.

**OPERATIONS**

These are the operations that need to be implemented by each member and they correspond to the functional operations within our restaurant ecosystem. Their purpose is to implement each numbered operation listed below as a PL/SQL **stored procedure** or **function**. For example, Add New Cuisine Type / Restaurant / Customer / Waiter / Menu / Inventory / Order etc. Create reports that show how each restaurant is doing regarding orders, food inventory, waiters’ tips, etc. A detailed set of operations to be performed is shown below.

**NOTE: All IDs must be automatically created using sequences.**

* **Member 1**: Responsible for the Cuisine Types and Restaurants tables. Must create these tables.
  1. Add cuisine type: Given the name of a cuisine type, add it to the table. The cuisine type is the input parameter to the PL/SQL procedure.
  2. Add restaurant: Add a new restaurant in the table with all pertinent input information.
  3. Display restaurant by cuisine: Given a cuisine type, show name and address about all restaurants that offer that cuisine.
  4. Report Income by state. Generate a report that lists the income of restaurants per cuisine type and per state.
* **Member 2**: Responsible for the Waiters table. Must create the table.

1. Hire waiter: Given all pertinent information as parameters, hire a waiter at a restaurant. The waiter name and restaurant ID must be input parameters. Use the function FIND\_RESTAURANT\_ID (see helper functions below) first to get the rID.
2. Show list of waiters: Given a restaurant ID (you need to call the appropriate helper function), show all info about each employee
3. Report tips: Show total tips by each waiter.
4. Report tips by state: Show total tips earned by waiters per state.

* **Member 3**: Responsible for the Menu Items and Restaurant Inventory tables. Must create these tables.

1. Create menu item: Given a cuisine type id, create a menu item (name and price) for that cuisine type. Use the function FIND\_CUISINE\_TYPE\_ID (see helper functions below) first to get the ID.
2. Add menu item to Restaurant Inventory: Given all pertinent information, add a menu item with a given quantity to a given restaurant in the Restaurant Inventory table. You will need to call helper functions to find IDs (see helper functions below).
3. Update menu item inventory: Given a restaurant id, a menu item id, along with a given quantity, reduce the inventory of that menu item by the amount specified by the quantity. This is to keep the inventory updated every time there is an order of an item.
4. Report Menu items: Generate a report to show totals of each menu item by type of cuisine.

* **Member 4**: Responsible for the Orders table. Must create this table.

1. Place an order: Given all required information, add an order in the Orders table. Use the FIND\_x\_ID helper functions (where x is the name of a table - see helper functions below) first to retrieve the IDs that are needed.
2. List all orders at a given restaurant on a given date.
3. List the most popular menu item ordered for each cuisine type
4. Report: Generate a report showing the top 3 restaurants of each state. The ranking is based on the total of ‘amount paid’ per restaurant per state.

* **Member 5**: Responsible for the Customers table. Must create this table.

1. Add a customer: Given all necessary information add a customer to the DB
2. List names of all customers who live in a given zip code
3. Report: Generate a report with the names of customers who spent the most money (top 3) so we can send them discount coupons, and also the names of the most frugal customers (bottom 3).
4. Report: States of generous customers. Generate a report that lists the states based on customers who tip generously. Show the total amount of tips by state in descending order of tip amount.