**SOFTWARE ENGINEERING 1 PRELIM EXAM**

**A BUSINESS ESTABLISHMENT'S DATABASE**

**NAME: Ronnie Angelo C. Quiao**

**SECTION: UCOS 3-1**

**INSTRUCTIONS:**

* Please keep in mind that this activity should be done **INDIVIDUALLY ONLY**. This will serve as an individual assessment to identify **how much you've learned** from the prelim period. Please follow the format on the next page.
* Your goal for your prelim exam is to come up with a **software engineering scenario** and then create a database for this concept.
* For example, a restaurant asking you to create an order management system or a school needing a library management system and your goal is to create a database for the system.
* The database should have a **minimum of 5 tables**. The finals exam project consists of two parts:
  + Part 1: Data Model
  + Part 2: Creating tables, inserting records, and reading records.
* **FOR PART 1**, I want you to create the data model of your database using this website only: <https://dbdiagram.io/d>. Please highlight the relationships so that we can see which entities have one to one, one to many or many to many relationships.
* **FOR PART 2**, please show the sql queries listed below. You can use the SQL editor from <https://www.programiz.com/sql/online-compiler/>

1. SQL query for creating the tables.
2. SQL query for inserting records into the tables *(You can use AI to generate random data for you to finish this part much faster)*.
3. Three to five (3-5) SQL queries that demonstrate the use of any of these: WHERE, JOIN, GROUP BY, and aggregate functions.

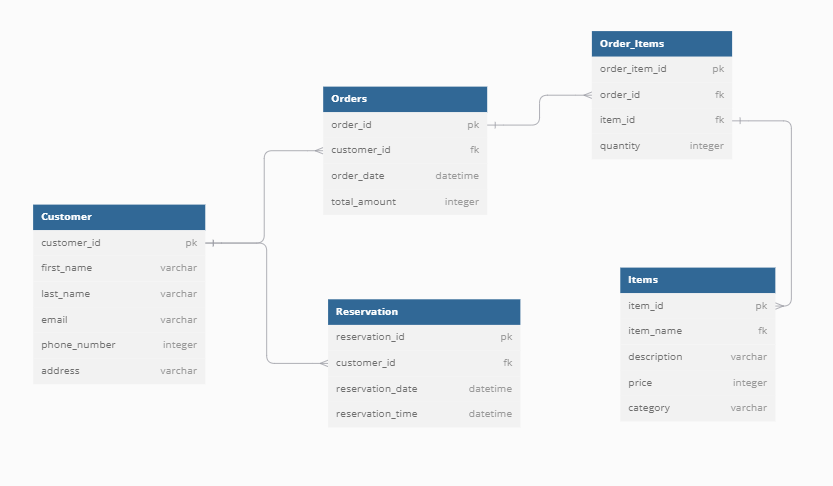
***TITLE HERE***

*“RESTAURANT MANAGEMENT SYSTEM FOR A BUSINESS”*

***DESCRIPTION HERE***

A Restaurant Management System (RMS) is a comprehensive software solution designed to streamline and automate various operational aspects of a restaurant business. It provides a centralized platform for managing orders, inventory, customer information, employee scheduling, and financial transactions. RMS can be tailored to fit the specific needs of different types of restaurants, from small cafes to large chains.

**PART 1: DATA MODEL**

**

***add a 2-3 sentence description about the data model***

*This data model provides a foundational structure for managing customer information, orders, products, and reservations in an e-commerce system. By understanding the entities, attributes, and relationships, you can effectively design and implement database queries and applications to support various business operations.*

**PART 2: SQL QUERIES**

**SQL QUERY FOR CREATING THE TABLES**

*CREATE TABLE Customer (*

*customer\_id INTEGER PRIMARY KEY AUTOINCREMENT,*

*first\_name VARCHAR(50) NOT NULL,*

*last\_name VARCHAR(50) NOT NULL,*

*email VARCHAR(100) UNIQUE NOT NULL,*

*phone\_number VARCHAR(15),*

*address VARCHAR(255)*

*);*

*CREATE TABLE Orders (*

*order\_id INTEGER PRIMARY KEY AUTOINCREMENT,*

*customer\_id INTEGER NOT NULL,*

*order\_date DATETIME NOT NULL,*

*total\_amount DECIMAL(10, 2) NOT NULL,*

*FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)*

*);*

*CREATE TABLE Items (*

*item\_id INTEGER PRIMARY KEY AUTOINCREMENT,*

*item\_name VARCHAR(50) NOT NULL,*

*description VARCHAR(255),*

*price DECIMAL(10, 2) NOT NULL,*

*category VARCHAR(50)*

*);*

*CREATE TABLE Order\_Items (*

*order\_item\_id INTEGER PRIMARY KEY AUTOINCREMENT,*

*order\_id INTEGER NOT NULL,*

*item\_id INTEGER NOT NULL,*

*quantity INTEGER NOT NULL CHECK (quantity > 0),*

*FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),*

*FOREIGN KEY (item\_id) REFERENCES Items(item\_id)*

*);*

*CREATE TABLE Reservation (*

*reservation\_id INTEGER PRIMARY KEY AUTOINCREMENT,*

*customer\_id INTEGER NOT NULL,*

*reservation\_date DATE NOT NULL,*

*reservation\_time TIME NOT NULL,*

*FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)*

*);*

**SQL QUERY FOR INSERTING RECORDS IN ATABLE**

*INSERT INTO Customer (first\_name, last\_name, email, phone\_number, address)*

*VALUES*

*('John', 'Doe', 'john.doe@example.com', '1234567890', '123 Elm Street'),*

*('Jane', 'Smith', 'jane.smith@example.com', '0987654321', '456 Oak Avenue'),*

*('Michael', 'Johnson', 'michael.j@example.com', '1122334455', '789 Maple Road'),*

*('Emily', 'Davis', 'emily.davis@example.com', '2233445566', '321 Pine Lane'),*

*('David', 'Miller', 'david.miller@example.com', '3344556677', '654 Birch Boulevard');*

*INSERT INTO Orders (customer\_id, order\_date, total\_amount)*

*VALUES*

*(1, '2024-09-10 10:00:00', 1200.50),*

*(2, '2024-09-11 14:30:00', 599.99),*

*(3, '2024-09-12 12:15:00', 750.00),*

*(4, '2024-09-13 16:45:00', 450.25),*

*(5, '2024-09-14 09:30:00', 999.99);*

*INSERT INTO Items (item\_name, description, price, category)*

*VALUES*

*('Pizza', 'Cheese pizza with tomato sauce and toppings', 12.99, 'Food'),*

*('Burger', 'Grilled beef patty with lettuce, tomato, and cheese', 8.99, 'Food'),*

*('Pasta', 'Penne pasta with marinara sauce', 10.99, 'Food'),*

*('Salad', 'Fresh mixed greens with vegetables and dressing', 7.49, 'Food'),*

*('Ice Cream', 'Vanilla ice cream served with chocolate syrup', 4.99, 'Dessert');*

*INSERT INTO Order\_Items (order\_id, item\_id, quantity)*

*VALUES*

*(1, 1, 1),*

*(2, 2, 1),*

*(3, 4, 1),*

*(4, 3, 2),*

*(5, 5, 1);*

*INSERT INTO Reservation (customer\_id, reservation\_date, reservation\_time)*

*VALUES*

*(1, '2024-09-15', '18:30:00'),*

*(2, '2024-09-16', '19:00:00'),*

*(3, '2024-09-17', '20:00:00'),*

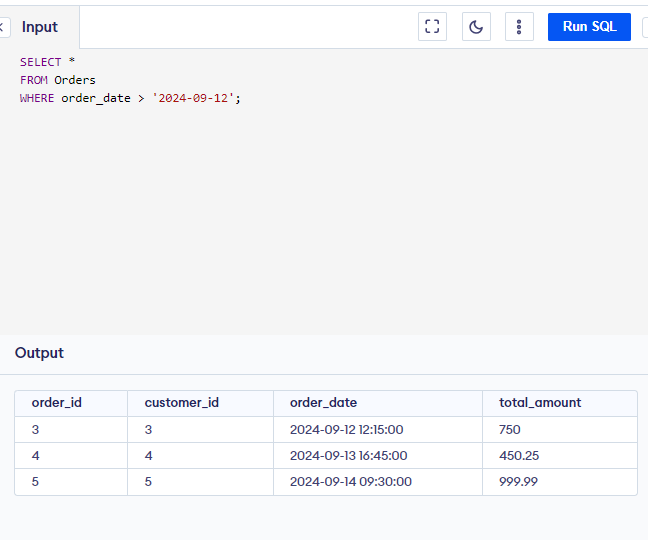
*(4, '2024-09-18', '17:30:00'),*

*(5, '2024-09-19', '21:00:00');*

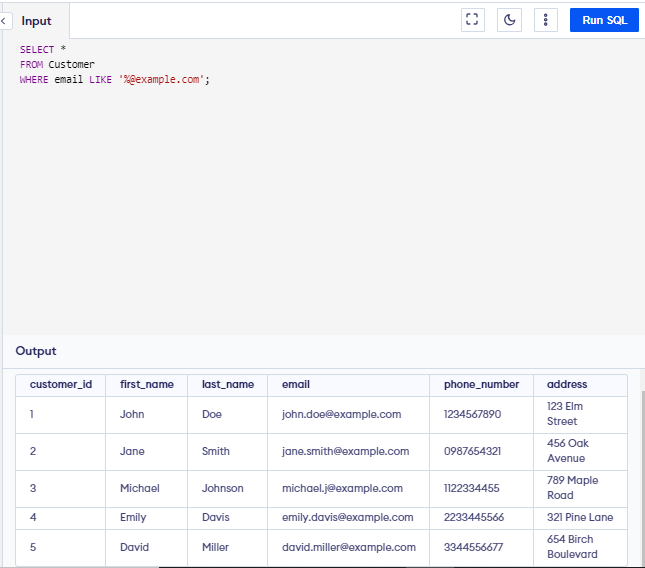
**SQL QUERIES DEMONSTRATING WHERE, JOIN, GROUP BY SQL STATEMENTS.**

*Show 3-5 SQL queries demonstrating any of the following WHERE, JOIN, GROUP BY, and aggregate functions.*

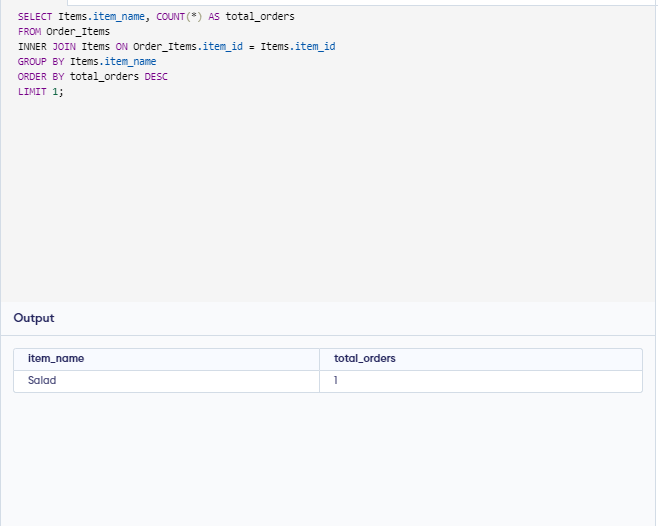
* *Find all orders placed after a specific date*

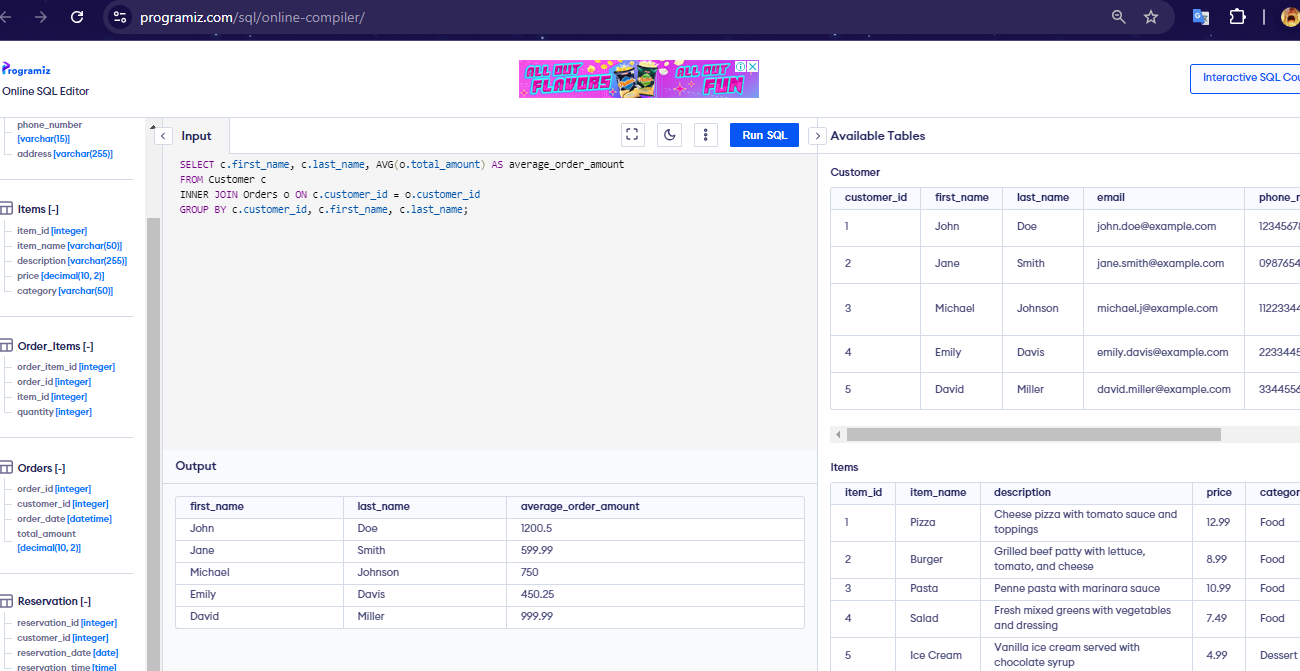


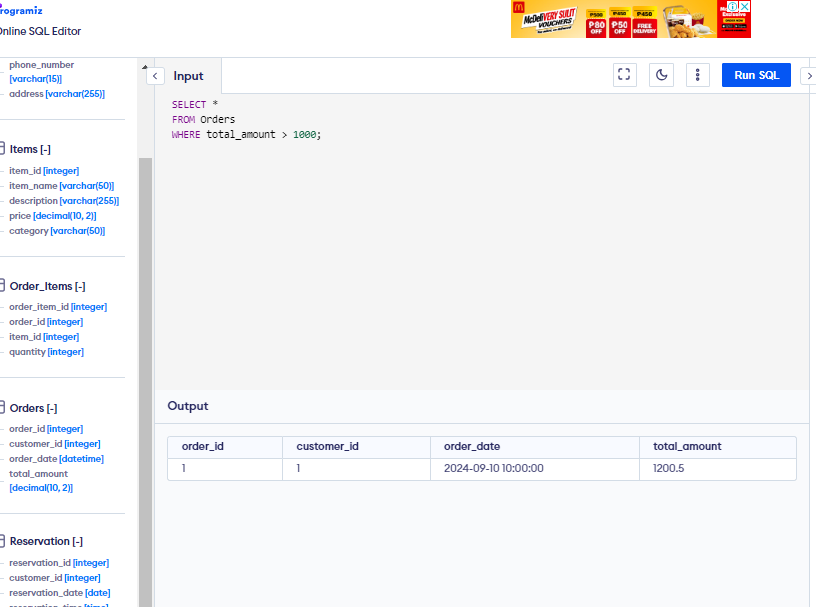
* Find all customers with email addresses ending in "@example.com"

**

* Find the most popular food item (JOIN, GROUP BY, COUNT)

**

* **Calculate the average order amount per customer (JOIN, GROUP BY, AVG):
* Find all orders with a total amount exceeding $1000 (WHERE)

**