SET 3

Q1. Generate an AI-assisted database schema for an **Online Food Delivery Application**. Use VS code with Github copilot

- Include tables: `Restaurants`, `Customers`, `Orders`, `Menu_Items` with some random data.
- Write SQL queries to list all customers who ordered items costing above ₹500.

Q2. Use AI to convert a Python dictionary manipulation program into equivalent **JavaScript code**.

- Verify correctness by comparing outputs of both scripts.

Task1.sql

Run on active connection | Select block

```sql
-- Create Database
CREATE DATABASE FoodDeliveryApp;
USE FoodDeliveryApp;

-- Create Restaurants table
CREATE TABLE Restaurants (
    restaurant_id INT PRIMARY KEY,
    name VARCHAR(100),
    address VARCHAR(200),
    phone VARCHAR(15),
    rating DECIMAL(3,1)
);

-- Create Customers table
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(15),
    address VARCHAR(200)
);

-- Create Menu_Items table
CREATE TABLE Menu_Items (
    item_id INT PRIMARY KEY,
    restaurant_id INT,
    item_name VARCHAR(100),
```

```sql
    price DECIMAL(10,2),
    category VARCHAR(50),
    FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id)
);

-- Create Orders table
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    restaurant_id INT,
    item_id INT,
    order_date DATETIME,
    quantity INT,
    total_amount DECIMAL(10,2),
    status VARCHAR(20),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
    FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id),
    FOREIGN KEY (item_id) REFERENCES Menu_Items(item_id)
);

-- Insert sample data
INSERT INTO Restaurants VALUES
(1, 'Spice Garden', 'MG Road, Bangalore', '9876543210', 4.5),
(2, 'Pizza Hub', 'HSR Layout, Bangalore', '9876543211', 4.2),
(3, 'Chinese Box', 'Koramangala, Bangalore', '9876543212', 4.0);

INSERT INTO Customers VALUES

(1, 'Rahul Sharma', 'rahul@email.com', '9898989898', 'Indiranagar, Bangalore'),
(2, 'Priya Singh', 'priya@email.com', '9797979797', 'JP Nagar, Bangalore'),
(3, 'Amit Kumar', 'amit@email.com', '9696969696', 'Whitefield, Bangalore');

INSERT INTO Menu_Items VALUES
(1, 1, 'Butter Chicken', 550.00, 'Main Course'),
(2, 2, 'Supreme Pizza', 699.00, 'Pizza'),
(3, 3, 'Dim Sum', 450.00, 'Starters'),
(4, 1, 'Biryani', 600.00, 'Main Course'),
(5, 2, 'Pasta Alfredo', 525.00, 'Italian');

INSERT INTO Orders VALUES
(1, 1, 1, 1, '2023-11-20 13:00:00', 2, 1100.00, 'Delivered'),
(2, 2, 2, 2, '2023-11-20 14:00:00', 1, 699.00, 'Delivered'),
(3, 3, 1, 4, '2023-11-20 15:00:00', 1, 600.00, 'In Transit');

-- Query to list customers who ordered items costing above ₹500
SELECT DISTINCT c.name, c.email, m.item_name, m.price
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Menu_Items m ON o.item_id = m.item_id
WHERE m.price > 500
ORDER BY m.price DESC;
```

OUTPUT:

MySQL LOCAL-4: multiple query results

Menu_Items VALUES (1, 1, 'B... | INSERT INTO Orders VALUES (1, 1, 1, 1, '20... | -- Query to list customers who ordered it...

name	email	item_name	price
Filter...	Filter...	Filter...	Filter...
Priya Singh	priya@email.com	Supreme Pizza	699
Amit Kumar	amit@email.com	Biryani	600
Rahul Sharma	rahul@email.com	Butter Chicken	550

TASK2

sk2.py > ...

```python
# /c:/Users/DEEKSHA/Desktop/AIAC/Lab Test-4/Task2.py

student = {"name": "John", "age": 20, "grade": "A"}

# Adding new key-value pair
student["subject"] = "Mathematics"

# Updating existing value
student["age"] = 21

# Deleting a key-value pair
del student["grade"]

# Printing the object
print("Updated student info:", student)
print("Keys:", list(student.keys()))
```

OUTPUT:

```
Updated student info: {'name': 'John', 'age': 21, 'subject': 'Mathematics'}
Keys: ['name', 'age', 'subject']
PS C:\Users\DEEKSHA\Desktop\AIAC\Lab Test-4>
```

JS

```javascript
// JavaScript object (equivalent to Python dictionary)
let student = { name: "John", age: 20, grade: "A" };

// Adding new key-value pair
student.subject = "Mathematics";

// Updating existing value
student.age = 21;

// Deleting a key-value pair
delete student.grade;

// Printing the object
console.log("Updated student info:", student);
console.log("Keys:", Object.keys(student));
```

OUTPUT:

```
PS C:\Users\DEEKSHA\Desktop\AIAC\Lab Test-4> node Task2.js
Original object: { name: 'John', age: 20, grades: [ 85, 90, 92 ] }
Name: John
Age: 20
After adding course: {
  name: 'John',
  age: 20,
  grades: [ 85, 90, 92 ],
  course: 'Computer Science'
}
After modifying age: {
  name: 'John',
  age: 21,
  grades: [ 85, 90, 92 ],
  course: 'Computer Science'
}
After deleting grades: { name: 'John', age: 21, course: 'Computer Science' }
Name exists in object
Keys: [ 'name', 'age', 'course' ]
Values: [ 'John', 21, 'Computer Science' ]
```