**Lab Manual**

**Task 3: Using Clauses, Operators, and Functions in Queries System Name**: Online Food Ordering System

**Aim**: To perform query processing on databases for different retrieval results using **DML** and **DRL operations** with aggregate functions, date functions, string functions, set clauses, and operators.

CREATE TABLE Customer ( Cust\_ID INT PRIMARY KEY,

Cust\_Name VARCHAR(100), Cust\_Contact VARCHAR(10) UNIQUE, Cust\_Email VARCHAR(100) NOT NULL,

Cust\_Address VARCHAR(100)

);

CREATE TABLE Restaurant ( Rest\_ID INT PRIMARY KEY,

Rest\_Name VARCHAR(100), Rest\_Location VARCHAR(200), Rest\_Contact VARCHAR(10)

);

CREATE TABLE Menu\_Item ( Item\_ID INT PRIMARY KEY,

Item\_Name VARCHAR(100), Price INT CHECK (Price > 0), Category VARCHAR(50), Rest\_ID INT,

FOREIGN KEY (Rest\_ID) REFERENCES Restaurant(Rest\_ID)

);

CREATE TABLE OrderTable ( Order\_ID INT PRIMARY KEY,

Cust\_ID INT, Order\_Date DATE, Order\_Total INT,

Payment\_Status VARCHAR(50),

FOREIGN KEY (Cust\_ID) REFERENCES Customer(Cust\_ID)

);

CREATE TABLE Delivery ( Delivery\_ID INT PRIMARY KEY, Order\_ID INT,

Delivery\_Status VARCHAR(50), Delivery\_Time DATE,

FOREIGN KEY (Order\_ID) REFERENCES OrderTable(Order\_ID)

);

1. **Insert Values into Customer Table**

INSERT INTO Customer VALUES (1, 'Alice', '9876543210', ['alice@example.com',](mailto:%27alice@example.com) 'Street 123');

INSERT INTO Customer VALUES (2, 'Bob', '9123456789', ['bob@example.com',](mailto:%27bob@example.com) 'Street 456');

INSERT INTO Customer VALUES (3, 'Charlie', '9988776655', ['charlie@example.com',](mailto:%27charlie@example.com) 'Street 789');

1. **Insert Values into Restaurant Table**

INSERT INTO Restaurant VALUES (1, 'Food Paradise', 'Downtown', '9988771234'); INSERT INTO Restaurant VALUES (2, 'Tasty Treats', 'Uptown', '8877664321'); INSERT INTO Restaurant VALUES (3, 'Global Eats', 'City Center', '7766552211');

1. **Insert Values into Menu\_Item Table**

INSERT INTO Menu\_Item VALUES (1, 'Pizza', 500, 'Italian', 1); INSERT INTO Menu\_Item VALUES (2, 'Burger', 300, 'Fast Food', 1); INSERT INTO Menu\_Item VALUES (3, 'Sushi', 800, 'Japanese', 2);

INSERT INTO Menu\_Item VALUES (4, 'Pasta', 400, 'Italian', 1);

INSERT INTO Menu\_Item VALUES (5, 'Noodles', 350, 'Chinese', 3);

1. **Insert Values into OrderTable Table**

INSERT INTO OrderTable VALUES (1, 1, '2025-01-20', 800, 'Paid');

INSERT INTO OrderTable VALUES (2, 2, '2025-01-21', 500, 'Unpaid');

INSERT INTO OrderTable VALUES (3, 3, '2025-01-22', 700, 'Paid');

1. **Insert Values into Delivery Table**

INSERT INTO Delivery VALUES (1, 1, 'Delivered', '2025-01-20 14:30:00');

**First insert into OrderTable**

INSERT INTO OrderTable VALUES (2, 2, '2025-01-25', 500, 'Pending');

**-- Then insert into Delivery**

INSERT INTO Delivery VALUES (2, 2, 'Pending', NULL);

INSERT INTO Delivery VALUES (3, 3, 'Delivered', '2025-01-22 16:00:00');

**Task 3: Using Clauses, Operators, and Functions in Queries Queries and Outputs**

1. ***Aggregate Functions***

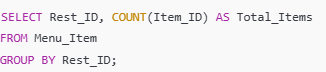
**Query 1: Find the total revenue generated by all orders.**

****

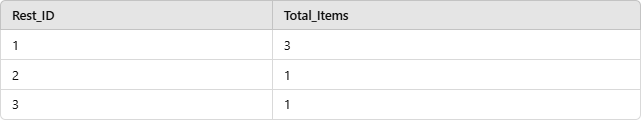
**Output:**



**Query 2: Find the total number of menu items offered by each restaurant.**

****

**Output:**

****

**Query 3: Find the average price of all menu items.**

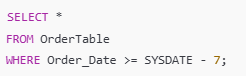
****

**Output:**

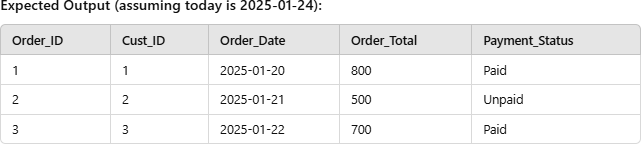
****

1. ***Date Functions***

**Query 1: Retrieve orders placed in the last 7 days.**

****

**Output:**



**Query 2: Find the orders and their delivery time if delivered today.**

**SELECT OrderTable.Order\_ID, Delivery.Delivery\_Time**

**FROM OrderTable**

**INNER JOIN Delivery**

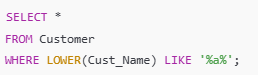
**ON OrderTable.Order\_ID = Delivery.Order\_ID;**

**Output:**

****

1. ***String Functions***

**Query 1: Find all customers whose names contain the letter 'a'.**

****

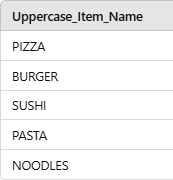
**Output:**

****

**Query 2: Display all menu item names in uppercase.**

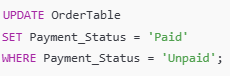
****

**Output:**

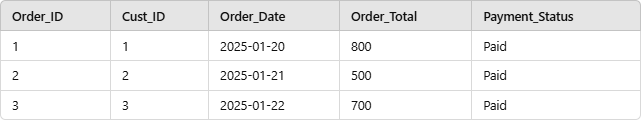


1. ***Set Clauses***

**Query 1: Update the payment status of all unpaid orders to 'Paid'.**

****

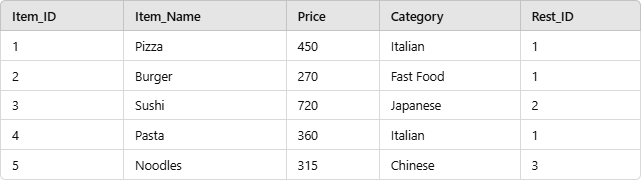
**Output:**

****

**Query 2: Reduce the price of all menu items by 10%.**

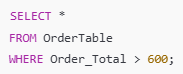
****

**Output:**

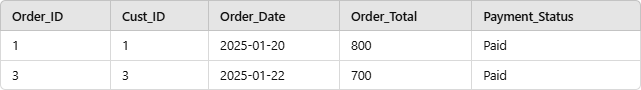
****

1. ***Operators***

**Query 1: Retrieve orders where the total is greater than 600.**

****

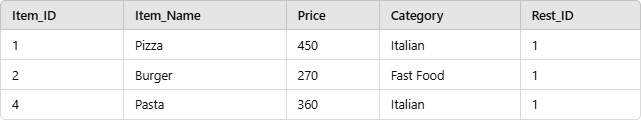
**Output:**

****

**Query 2: Retrieve menu items that belong to category 'Italian' or have a price less than 350.**

****

**Output:**

****