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, Cust_Name VARCHAR(100), Cust_contact
VARCHAR(10) UNIQUE, Cust_Email VARCHAR(100) NOT NULL, Cust_Address
VARCHAR(100)
);

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)
);
```

Customer

Cust_ID	Cust_Name	Cust_contact	Cust_Address
1	Alice	9876543210	street 123
2	Bob	9123456789	street 456
3	Charlie	9988776655	street 789

Menu_Item

Item	Item_Name	Price	Category	Rest_ID
1	Pizza	500	Italian	1
2	Burger	300	Fast Food	1
3	Sushi	800	Japanese	2
4	Pasta	400	Italian	1
5	Noodles	350	Chinese	3

Order_Table

Order_ID	Cust_ID	Order_Date	Order_Total	Payment Status
1	1	2025-01-20	800	Paid
2	2	2025-01-21	500	Unpaid
3	3	2025-01-22	700	Paid

Restaurant :-

Rest_ID	Rest_Name	Rest_Location	Rest_contact
1	Food Paradise	DownTown	9988771234
2	Tasty Treats	Uptown	87766321
3	Global Estate	City Center	7766522111

Delivery :-

Delivery	Order_ID	Delivery_status	Delivery_Time
1	1	Delivered	2025-01-2014:30:00
2	2	Pending	–
3	3	Delivered	2025-01-22 16:00:00

1. Output Query 1

Total Revenue
2000

2. Output Query 2

Rest_ID	Total_Items
1	3
2	1
3	1

```
CREATE TABLE
   Order Table (
   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', 'street 123');

INSERT INTO Customer VALUES(2, 'Bob', '9123456789', 'bob@exam
    com', 'street 456');
```

Output :

Average_Price
470

Cust_Name
Alice
Charlie

Output :

Cust_ID	Cust_Name	Cust_Cont	Cust_Email	Cust_Address
1	Alice	9876543210	alice@example.com	Street 123
2	Charlie	9988776655	charlie@example.com	Street 789

Output :-

Upper_Case_Item_Name
Pizza
Burger
Sushi
Pasta
Noodles

Output :

Order_ID	Cust_ID	Order_date	Order_total	Payment
1	1	2025-01-20	800	Paid
2	2	2025-01-21	800	Paid
3	3	2025-01-22	700	Paid

```
INSERT INTO Customer VALUES (3, 'charlie', '9988776655', 'charlie
@example.com', 'Street 789');
```

2. Insert Values into Restaurant Table

```
INSERT INTO Restaurant VALUES (1, 'Food Paradise', 'Downtown', '9988
771234'); INSERT INTO Restaurant VALUES (2, 'Tasty Treats', 'Uptown',
8877664321');
INSERT INTO Customer VALUES(3,'charlie', '9988776655', 'charlie@
example.com', 'Street 789');
```

3. 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);
```

4. Insert Values into OrderTable Tables

```
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');
```

5. Insert Values into Delivery Table

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

Output :-

Item_ID	Item Name	Price	Category	Rest_ID
1	Pizza	450	Italain	1
2	Burger	270	Fast Food	1
3	Sushi	720	Japanese	2
4	Pasta	360	Italian	1
5	Noodles	315	chinese	3

Output :-

Order_ID	Cust_ID	Order_Date	Order_Total	Payment_status
1	1	2025-01-20	800	Paid
3	3	2025-01-22	700	Paid

Output :-

Item_id	Item_Name	Price	Category	Rest_ID
1	Pizza	450	Italian	1
2	Burger	270	Fast Food	1
4	Pasta	360	Italian	1

Cust_ID	Order_Total
1	800

Item_ID	Item_Name	Price
3	Sushi	720

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 Qua Clauses, Operators, and Functions in Queries and Outputs

1. Aggregate Functions

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

```
SELECT SUM(Order_total) AS Total_Revenue FROM OrderTable;
```

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

```
SELECT Rest_Id, COUNT(Item_Id) AS Total_Items
FROM Menu_Item
Group by Rest_ID;
```

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

```
SELECT AVG(Price) AS Average_Price FROM Menu_Item;
```

2. Date Functions

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

```
SELECT *
FROM OrderTable
WHERE Order_date >= SYSDATE - 7;
```

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;
```

3. String functions

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

```
SELECT *
FROM Customer
WHERE LOWER(Cust_Name) LIKE "%a%";
```

Query 2: Display all menu item names in uppercase.

```
SELECT UPPER(Item_Name) AS Uppercase_Item_Name FROM MENU_
```

4. Set clauses

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

```
UPDATE OrderTable
SET Payment_status = 'Paid'
WHERE Payment_status = 'Unpaid';
```

6. Operators

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

```
SELECT *
FROM Order Table
WHERE Order_Total > 600;
```

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

```
SELECT *
FROM Menu_Item
WHERE category = 'Italian' OR Price < 350;
```