12|8|2

Task-3

using claues. operators, and Function in Queries

System Name ÷ Online Food Ordering System

Aim÷ To perform query processing on database for different retrival 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). NOTNULL
   Cust-Address VARCHAR(100)
   );
  CREATE TABLE Restaurant
    (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 Menu-item
 (item-ID INT. PRIMARY KEY
   item-Name VARCHAR(100),
   Price INT CHCEK (Price>a);
   category VARCHAR(50);
     Rest-ID INT,
```

2. Date Functions

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

```
SELECT*
FROM Order Table
WHERE order_Date>=SYSDATE-7;
```

Output:

order_ID	Cust_ID	order_Date	order_Total	Payment_Status
1	1	2025-01-20	1800	Paid
2	2	2025-01-21	500	unpaid
3	3	2025-01-22	1700	Paid

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

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

Output!

Cust_ID	Cust_Name	Cust_contact	Cust_Email	Cust_Address
1	Alice	9876543210	alice@Example.com	street123
3	Charlie	9988776655	charlie@Example.com	street789

Query2: Display all menu item names in uppercase.

```
SELECT UPPER(Item_Name) AS uppercase_Item_Name
FROM Menu_Items;
```

Output:

uppercase_Item_Name
PIZZA
BURGER
SUSHI
PASTA
NOODLES

```
FOREIGN KEY (Rest_ID) REFERENCES
  Restaurant (Rest_ID);
  );
 CREATE TABLE OrderTable
  (Order_ID INT PRIMARY KEY,
     Cust_ID INT
     Order_ Date,
     Order_ Total INT
    Payment_ Status VARCHAR(50)
   FORIGN KEY(cust_ID) REFERENCES
     OrderTable (Order_ID)
     );
```

1. INSERT values into Customer Table

```
 INSERT INTO Customers VALUS (1, 'Alice'
  '9876543210', 'alice@ Example.com', 'street
  123');
INSERT INTO Customer VALUES (2, 'Bob',
'9123456789', 'bob @ gmail.com', 'street456
INSERT INO Customer VALUES (3, 'Charlie',
 '9988776655', charlie @ example.com',
  'street 789');
```

2. Insert values into Restaurant Table

```
   INSERT INTO Restaurant ValuEs
    (1, 'Food Paradise', 'Downtown', '9988771234')
   INSERT INTO Restaurant VALUES (2, 'Tasty
    Treats', 'uptown', '8877664321'),
```

Query1: Update the payment status of all unpaid orders to 'Paid'

```
UPDATE Order Table
SET Payment_status = 'Paid'
WHERE Payment_Status = 'unpaid';
```

output:

Order-ID	Cust_ID	Order-Date	Order-Total	Payment-status
1	1	2025-01-20	800	Paid
2	2	2025-01-21	500	Paid
3	3	2025-01-22	700	Paid

Query2: Reduce the price of all menu items by 10%

```
UPDATE Menu_Item
SET Price = Price * 0.9;
```

Output:

Item-ID	Item_Name	Price	Category	Rest-ID
1	Pizza	450	Italian	1
2	Burger	270	Fast Food	1
3	Sushi	720	Japanese	2
4	Pasta	360	Italian	1
5	Noodles	315	Chinese	3

5. Operators

Query1: Retrieve orders where the total is greater than 600.

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

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

Query2: 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;
```

Item-ID	Item-Name	Price	Category	Rest-ID
	Pizza	450	Italian	1
2	Burger	270	Fast Food	1
4	Pasta	360	Italian	1

INSERT INTO Restaurant VALUES(3, 'Global Eats', Citycenter, '7766552211');

3. INSERT INTO Menu-item values Table

INSERT INTO Menu-item VALUES(1, 'Pizza', 500, 'Italian', 1);
INSERT INTO Menu-item VALUES(2, 'Burgi', 300, 'Fast Food', 1);

INSERT INTO Menu-item VALUES(3, 'sushi' 800, 'Fast Food', 1);

INSERT INTO Menu_item VALUES(4, 'pasta', 400, 'Italian', 1);

4. INSERT VALUES into Order Table
INSERT INTO order Table VALUES (1, 1, '2025.01-20', 800, paid')

INSERT INTO order Table VALUES (2, 2, '2025-01-21', 500, 'unpaid');
INSERT INTO order Table VALUES (3, 3, '2025-01-22', 700, 'paid');

5. Insert values into Delivary Table
Insert INTO Delivery VALUES (1, 1, 'Delivered, '2025-01-20', 14:30:00');
First
INSERT into order Table
INSERT INTO order Table VALUES (2, 2, '2025-01-25', 500);
Then INSERT INTO Delivery VALUS(2, 2, 'Delivered, '2025-01-22 16:00:00');

Result : Task Executed Sucessfully

VEL TECH
EX NO.
PERFORMANCE (5)
RESULT AND ANALYSIS (3)
VIVA VOCE (2)
RECORD (?)
TOTAL (10)
SIGN WITH DATE