ASSIGNMENT 3

SUBJECT- Data Driven Modelling

CASE STUDY - Online Food Delivery Management System

Student Details

Name : DHANASHRI SANJAY AHIRRAO
PRN : 202501080080
Class and Division : IT1 and IT-14
Date of Submission : 14-12-2025

Case Study Details

An online food delivery platform allows customers to order food from multiple restaurants. The system maintains restaurant details, menu items, customer records, and order information.

Part A (DDL) + Part B (DML)

1) CUSTOMER TABLE

```
CREATE TABLE CUSTOMER (
   customer_id NUMBER PRIMARY KEY,
   customer_name VARCHAR2(50) NOT NULL,
   phone VARCHAR2(10) UNIQUE NOT NULL,
   email VARCHAR2(50) UNIQUE,
   city VARCHAR2(30) CHECK (city IN ('Pune','Mumbai','Nashik'))
);
```

INSERT VALUE

```
INSERT INTO CUSTOMER VALUES
(1,'Aditi','9876543210','aditi@gmail.com','Pune');

INSERT INTO CUSTOMER VALUES
(2,'Rohan','9876543211','rohan@gmail.com','Mumbai');

INSERT INTO CUSTOMER VALUES
(3,'Sneha','9876543212','sneha@gmail.com','Pune');

INSERT INTO CUSTOMER VALUES
(4,'Aman','9876543213','aman@gmail.com','Nashik');

INSERT INTO CUSTOMER VALUES
(5,'Neha','9876543214','neha@gmail.com','Pune');
```

CUSTOMER

customer_id	customer_name	phone	email	city
1	Aditi	9876543210	aditi@gmail.com	Pune
2	Rohan	9876543211	rohan@gmail.com	Mumbai
3	Sneha	9876543212	sneha@gmail.com	Pune
4	Aman	9876543213	aman@gmail.com	Nashik
5	Neha	9876543214	neha@gmail.com	Pune

---

2) RESTAURANT TABLE

```
CREATE TABLE RESTAURANT (
   restaurant_id NUMBER PRIMARY KEY,
   restaurant_name VARCHAR2(50) NOT NULL,
   location VARCHAR2(30) NOT NULL,
   rating NUMBER CHECK (rating BETWEEN 1 AND 5)
);
```

INSERT VALUE

```
INSERT INTO RESTAURANT VALUES (101,'Food Plaza','Pune',4);
INSERT INTO RESTAURANT VALUES (102,'Spice Hub','Mumbai',5);
INSERT INTO RESTAURANT VALUES (103,'Tandoori House','Pune',4);
INSERT INTO RESTAURANT VALUES (104,'Pizza Corner','Nashik',3);
INSERT INTO RESTAURANT VALUES (105,'Burger Zone','Pune',4);
```

RESTAURANT

restaurant_id	restaurant_name	location	rating
101	Food Plaza	Pune	4
102	Spice Hub	Mumbai	5
103	Tandoori House	Pune	4
104	Pizza Corner	Nashik	3
105	Burger Zone	Pune	4

---

3) MENU_ITEM Table

```
CREATE TABLE MENU_ITEM (
   item_id NUMBER PRIMARY KEY,
   item_name VARCHAR2(50) NOT NULL,
   price NUMBER CHECK (price > 0),
   restaurant_id NUMBER,
   CONSTRAINT fk_restaurant
   FOREIGN KEY (restaurant_id) REFERENCES RESTAURANT(restaurant_id)
);
```

INSERT VALUE

```
INSERT INTO MENU_ITEM VALUES (201,'Paneer Tikka',250,101);
INSERT INTO MENU_ITEM VALUES (202,'Biryani',300,102);
INSERT INTO MENU_ITEM VALUES (203,'Pizza',450,104);
INSERT INTO MENU_ITEM VALUES (204,'Burger',180,105);
INSERT INTO MENU_ITEM VALUES (205,'Butter Chicken',380,103);
```

MENU_ITEM

item_id	item_name	price	restaurant_id
201	Paneer Tikka	250	101
202	Biryani	300	102
203	Pizza	450	104
204	Burger	180	105
205	Butter Chicken	380	103

---

4) ORDER_MASTER Table

```
CREATE TABLE ORDER_MASTER (
   order_id NUMBER PRIMARY KEY,
   customer_id NUMBER,
   order_date DATE,
   total_amount NUMBER CHECK (total_amount > 0),
   CONSTRAINT fk_customer
   FOREIGN KEY (customer_id) REFERENCES CUSTOMER(customer_id)
);
```

INSERT VALUE

```
INSERT INTO ORDER_MASTER VALUES (301,1, '11-12-2005' ,500);
INSERT INTO ORDER_MASTER VALUES (302,2, '12-12-2005' ,750);
INSERT INTO ORDER_MASTER VALUES (303,1, '02-02-2005' ,300);
INSERT INTO ORDER_MASTER VALUES (304,3, '20-02-2004' ,450);
INSERT INTO ORDER_MASTER VALUES (305,4, '17-09-2006' ,600);
```

order_id	customer_id	order_date	total_amount
301	1	11-12-2005	500
302	2	12-12-2005	800
303	5	02-02-2005	200
304	3	20-08-2004	300
305	4	17-09-2006	840

---

5) ORDER_ITEM Table

```
CREATE TABLE ORDER_ITEM (
   order_item_id NUMBER PRIMARY KEY,
   order_id NUMBER,
   item_id NUMBER,
   quantity NUMBER CHECK (quantity > 0),
   CONSTRAINT fk_order
   FOREIGN KEY (order_id) REFERENCES ORDER_MASTER(order_id),
   CONSTRAINT fk_item
   FOREIGN KEY (item_id) REFERENCES MENU_ITEM(item_id)
);
```

INSERT VALUE

INSERT INTO ORDER_ITEM VALUES (401,301,201,2);
INSERT INTO ORDER_ITEM VALUES (402,302,202,2);
INSERT INTO ORDER_ITEM VALUES (403,303,204,1);
INSERT INTO ORDER_ITEM VALUES (404,304,203,1);
INSERT INTO ORDER_ITEM VALUES (405,305,205,2);

order_item_id	order_id	item_id	quantity
401	301	201	2
402	302	202	3
403	303	203	1
404	304	204	4
405	305	205	5

Part C (DCL)

CREATE USER food_user IDENTIFIED BY food123;
GRANT CONNECT, RESOURCE TO food_user;
REVOKE RESOURCE FROM food_user;

Part D (Data Handling & Querying)

1. List All Orders

SELECT * FROM ORDER_MASTER;

Output

order_id	customer_id	order_date	total_amount
301	1	2005	500

2. Join Customer and Order tables

```
SELECT c.customer_name, o.order_id, o.total_amount
FROM CUSTOMER c
JOIN ORDER_MASTER o
ON c.customer_id = o.customer_id;
```

Output

customer_name	order_id	total_amount
Aditi	301	500

3. Count Total Orders

```
SELECT COUNT(*) AS total_orders FROM ORDER_MASTER;
```

total_orders
1

4. Group Orders per Restaurant

```
SELECT r.restaurant_name, COUNT(o.order_id)
FROM RESTAURANT r
JOIN MENU_ITEM m ON r.restaurant_id = m.restaurant_id
JOIN ORDER_ITEM oi ON m.item_id = oi.item_id
JOIN ORDER_MASTER o ON oi.order_id = o.order_id
GROUP BY r.restaurant_name;
```

restaurant_name	COUNT(o.order_id)
Food Plaza	1

5. Filter High-value orders using Having

```
SELECT customer_id, SUM(total_amount)
FROM ORDER_MASTER
GROUP BY customer_id
HAVING SUM(total_amount) > 600;
```

Part D

6. **Top 3 most Expensive Item**

```
SELECT item_name, price, restaurant_name
FROM MENU_ITEM m
JOIN RESTAURANT r
ON m.restaurant_id = r.restaurant_id
ORDER BY price DESC
FETCH FIRST 3 ROWS ONLY;
```

restaurant_id	item_name	total_amount
104	Pizza	450
103	Butter chicken	380
102	Biryani	300

7. **Customers with more than 2 Orders**

```
SELECT customer_id
FROM ORDER_MASTER
GROUP BY customer_id
HAVING COUNT(order_id) > 2;
```

Number of Orders	customer_id
5	305
4	304

8. Restaurant Wise Total Revenue

```
SELECT r.restaurant_name, SUM(o.total_amount) AS revenue
FROM RESTAURANT r
JOIN MENU_ITEM m ON r.restaurant_id = m.restaurant_id
JOIN ORDER_ITEM oi ON m.item_id = oi.item_id
JOIN ORDER_MASTER o ON oi.order_id = o.order_id
GROUP BY r.restaurant_name;
```

Total revenue	restaurant_name
1000	Food Plaza
3020	Spice Hub
4500	Tandoori House

9. Customer who Never Place an Order

```
SELECT customer_name
FROM CUSTOMER
WHERE customer_id NOT IN
(SELECT customer_id FROM ORDER_MASTER);
```

customer_id	customer_name
1	Aditi
2	Rohan
3	Sneha

Explanation

The database is designed to manage an online food delivery system using five related tables: Customer, Restaurant, Menu_Item, Order_Master, and Order_Item.
Primary keys uniquely identify each record, while foreign keys maintain relationships between tables.
Constraints like NOT NULL, UNIQUE, and CHECK ensure data accuracy and consistency.
DDL commands help create and structure the database.
DML commands allow inserting, updating, deleting, and modifying data.
DCL commands control user access and security.
Together, these SQL operations help manage real-world data efficiently and securely.