Assignment -01

Name :- P. Venkata Sai pradeep Reddy

Reg No :- 19231364

Subject code :- Database management System - CSA0593

Assignment - 01

CSA 0593 DBMS 192311364 P. Venkata Sai Pradeep Redd

(1) Scenario Breakdown:-

In food delivery service, there are several core components.

1) Customers : Users of service who place orders.
2) Restaurants : Provides food items available for order.
3) Items : Menu items listed by each restaurant.
4) orders : orders placed by customers, containing multiple items
5) Delivery drivers : Drivers responsible for delivering orders
6) Feedback : Customer feed back on orders & Service Quality
7) Delivery status : Traces delivery progress of each order.

Database Design:-

* 1) Table [Customer Table]

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    Phone VARCHAR (20),
    address TEXT NOT NULL,
    Created_at TIMESTAMP DEFAULT Current_TIMESTAMP
);
```

* Restaurants Table :-

```
CREATE TABLE Restaurants(
    restaurant_id INT primary_key Auto_increment,
    Name VARCHAR(200) NOT NULL
    address TEXT NOT NULL,
    phone VARCHAR(20),
    created at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Columns :- restaurant_id, name, address, phone and created_at

Purpose :- provide restaurant details, essential for connecting menu items and facilitating delivery logistics.

* Items Table :-

Stores individual items within an order.

```
CREATE TABLE order_items (
    Order Item_id INT PRIMARY KEY Auto_INCREMENT,
    Order_id , INT,
    item_id INT,
    Quantity INT NOT NULL,
    price Decimal ( 10, 2) NOT NULL,
    FOREIGN key (order_id) References orders( order_id)
                ON DELETE CASCADE,
    FOREIGN key( item_id) References Items (item_id)
);
```

Columns:- order_item_id, order_id, item_id, quantity, price.

Purpose:- Manage items within each order, including quantity & price.

* Delivery_Drivers Table :-

Information about drivers available for delivery.

```
CREATE TABLE Delivery_Drivers(
    driver_id INT primary key Auto_INCREMENT,
    Name VARCHAR(100),
    phone VARCHAR(20),
    is_available BOOLEAN DEFAULT TRUE.
```

Column:- driver_id, name, phone, is_available.

Purpose:- Track driver availability and facilitate assignment to orders.

* Feedback table :-

Stores feedback from customers related to orders.

```
CREATE TABLE feedback(
    feedback_id INT primary key Auto_INCREMENT,
    order_Id INT,
    Customer_ID INT,
    rating INT Check (rating Between 1 AND 5),
    comment TEXT
    created_at TIMESTAMP DEFAULT Current_TIMESTAMP
    FOREIGN_key (order_id) References orders(order_id),
```

FOREIGN_KEY (Customer_id) References customers (customer_id)
);

Columns :- feedback_id, order_id, customer id, rating, comment, created at

Purpose :- Collect customer feedback to improve Service Quality.

* Delivery - status table :-

Tracks and status & driver information for each order.

CREATE TABLE Delivery_status(
status_id INT primary key, Auto_increment,
order_id INT,
delivery_id INT,
status ENMU ('Pending', 'in Transit', 'Delivered') Default 'Pending',
last_updated TIMESTAMP DEFAULT Current_TIMESTAMP ON UPDATE Current_TIME STAMP,
FOREIGN Key (Order_id) References orders(order_id),
FOREIGN key(deliver_id) References Delivery_Drivers(driver_id)
);

Columns :- status_id, order_id, driver_id, status, last_updated.

Purpose :- Trans progress of each order delivery & update in real time.

Sorted procedures :-

1) place an order :- Inserts a new order & associated items

```
CREATE PROCEDURE placeorder(
    IN p_customer_id INT,
    IN p_restaurant_id INT,
    IN p_items JSON.
)
BEGIN
    DECLARE order_total (Decimal (10,2) Default 0);
    START TRANSACTION;
    INSERT INTO orders (Customer_id), restaurant_id, total_amount)
    Values (p_customer_id, p_restaurant_id, 0);
    SET @ order_id = LAST_INSERT_ID();
    DECLARE item_id INT;
    DECLARE quantity INT;
    DECLARE price Decimal (10,2);
    For Each item IN JSON_EXRTACT (p_items, '$[*]') DO
        SET item_id = item -> '$.item_id';
        SET quantity = item -> '$.quantity';
    SELECT price into price From items where item_id
                                = item_id;
    INSERT_INTO order_items (order_id, item_id, quantity, price)
    Values(@ order_id, item_id, quantity, price);
    SET order_total = order_total + (quantity * price);
END FOR;
```

* In Conclusion, the proposed customer order management system for a food delivery Service provides a robust, Scalable database structure tailored to handle essential operations, including customer management, restaurant menus, order processing, delivery tracking, and customer feed back.

The Solution not only meets immediate requirements but also lays a strong foundation for future enhancements, Such as Personalized recomendations, dynamic pricing, or advanced delivery logistics. Overall This structured approach aligns with goals of modern food delivery Service.

Conclusion :- This customer order management System provides a scalable, efficient Solution for handling orders, delivery logistics and customer feedback in a food delivery Service.

Update delivery status :-

Updates the status of an order in delivery status table.

```
CREATE TABLE Updatedeliverystatus(
  IN p_order_id INT,
  IN p_status ENUM('pending', 'In Transit', 'Delivered',

BEGIN
    UPDATE Delivery_status SET status = p_state WHERE
  order_id = p_order_id;

END;
```

3 Assign Driver :-

```
CREATE TABLE Assignment(
    IN p_order_id INT,
    IN p_driver_id INT
)

BEGIN
 UPDATE Delivery_Drivers SET is_available = FALSE WHERE
driver_id = p_driver_id;
INSERT INTO Delivery_status (order_id, driver_id, status)
 VALUES (p_order_id, p_drivers_id, 'pending');
END;
```