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 : Tracks delivery progress of each order.

Database Design:-

* 1) Tables [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
);
```

Update delivery status:-

Updates the stahs 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;
```

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;
```

```
FOREIGN-KEY (Customer-id) References customers(customer-id)
);
```

Columns:- feed back-id, order-id, customer id, rating, comment, created at

Purpose:- Colleut customer feedback to improve Service Quaity.

* Delivery - status table :-

Trauks 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,
    FORIEGN 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:- Track progress of each order delivery & update in real time.

columns:- order-item-id, order-id, item-id, quantity, price.

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

* Delivery - Drivers Table :-

Information about driver 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),
```

* 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

Purpoce :- 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 RRIMARY 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) Reference Items (item-id)
);
```