Day 6: Online Food Delivery Platform

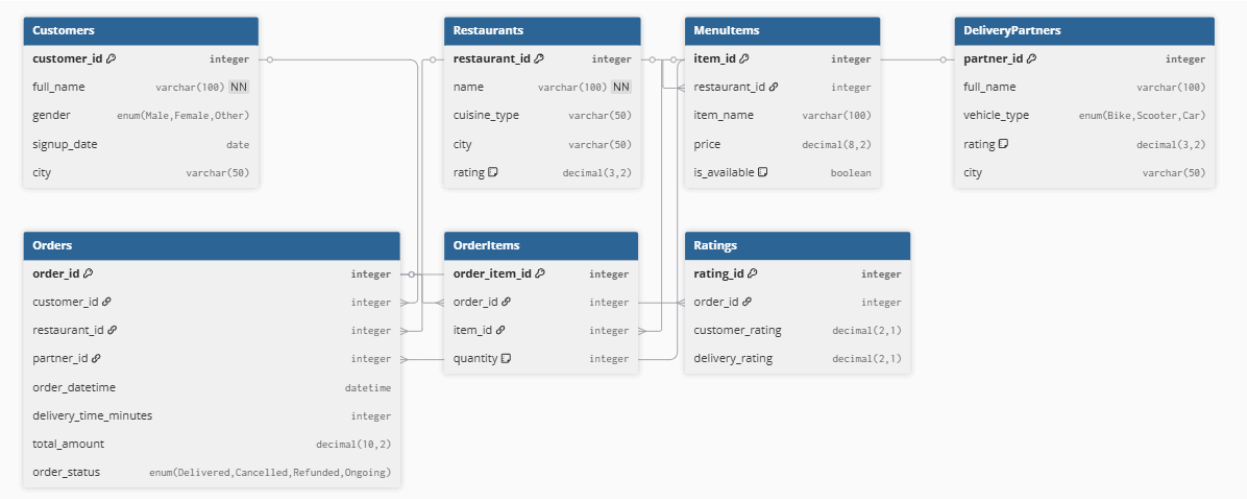
🎯 Business Context

A large-scale **food delivery startup** (similar to Swiggy/Zomato/UberEats) wants to analyze:

- **Order performance**, **restaurant ratings**, **delivery time efficiency**, and **customer satisfaction.**
- SQL Analysts and Data Engineers build analytics queries across **customers, restaurants, delivery partners, and orders**.
- The data is used to **optimize delivery routes**, **boost high-performing restaurants**, and **reduce delays.**

🛍️ Database Schema: FoodDeliveryDB

This schema is **moderately normalized** for OLAP analysis.



1️⃣ Customers

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    full_name VARCHAR(100) NOT NULL,
    gender ENUM('Male', 'Female', 'Other'),
    signup_date DATE,
    city VARCHAR(50)
);
```

2️⃣ Restaurants

```
CREATE TABLE Restaurants (
    restaurant_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    cuisine_type VARCHAR(50),
    city VARCHAR(50),
    rating DECIMAL(3,2) CHECK (rating BETWEEN 0 AND 5)
);
```

3️⃣ MenuItems

```
CREATE TABLE MenuItems (
    item_id INT PRIMARY KEY,
    restaurant_id INT,
    item_name VARCHAR(100),
    price DECIMAL(8,2),
    is_available BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id)
);
```

4️⃣ DeliveryPartners

```
CREATE TABLE DeliveryPartners (
    partner_id INT PRIMARY KEY,
    full_name VARCHAR(100),
    vehicle_type ENUM('Bike', 'Scooter', 'Car'),
    rating DECIMAL(3,2) CHECK (rating BETWEEN 0 AND 5),
    city VARCHAR(50)
);
```

5️⃣ Orders

```
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    restaurant_id INT,
    partner_id INT,
    order_datetime DATETIME,
    delivery_time_minutes INT,
    total_amount DECIMAL(10,2),
    order_status ENUM('Delivered', 'Cancelled', 'Refunded', 'Ongoing'),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
    FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id),
    FOREIGN KEY (partner_id) REFERENCES DeliveryPartners(partner_id)
);
```

6️⃣ OrderItems

```
CREATE TABLE OrderItems (
    order_item_id INT PRIMARY KEY,
    order_id INT,
    item_id INT,
    quantity INT CHECK (quantity > 0),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
```

```
    FOREIGN KEY (item_id) REFERENCES MenuItems(item_id)
);
```

7️⃣ Ratings

```
CREATE TABLE Ratings (
    rating_id INT PRIMARY KEY,
    order_id INT,
    customer_rating DECIMAL(2,1),
    delivery_rating DECIMAL(2,1),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);
```

📊 ERD Overview (Textual)

```
Customers (1)——< (M) Orders (1)——< (M) OrderItems >——(1) MenuItems
>——(M) Restaurants
        |
        └———————— Ratings ——< (1) Orders
DeliveryPartners (1)——< (M) Orders
```

🧮 Sample Data

```
INSERT INTO Customers VALUES
(1, 'Arjun Mehta', 'Male', '2023-05-10', 'Mumbai'),
(2, 'Sara Khan', 'Female', '2023-06-15', 'Delhi'),
(3, 'Rohan Das', 'Male', '2023-04-01', 'Bangalore'),
(4, 'Meera Iyer', 'Female', '2023-07-20', 'Mumbai');

INSERT INTO Restaurants VALUES
```

```
(1, 'SpiceVilla', 'Indian', 'Mumbai', 4.3),
(2, 'SushiZen', 'Japanese', 'Delhi', 4.7),
(3, 'UrbanBites', 'Continental', 'Bangalore', 4.2),
(4, 'QuickBite', 'Fast Food', 'Mumbai', 3.9);

INSERT INTO MenuItems VALUES
(1, 1, 'Paneer Tikka', 250.00, TRUE),
(2, 1, 'Dal Makhani', 180.00, TRUE),
(3, 2, 'Salmon Sushi', 600.00, TRUE),
(4, 3, 'Pasta Alfredo', 350.00, TRUE),
(5, 4, 'Veg Burger', 120.00, TRUE);

INSERT INTO DeliveryPartners VALUES
(1, 'Ravi Kumar', 'Bike', 4.8, 'Mumbai'),
(2, 'Amit Verma', 'Car', 4.5, 'Delhi'),
(3, 'Rajesh Nair', 'Scooter', 3.9, 'Bangalore'),
(4, 'Deepak Singh', 'Bike', 4.2, 'Mumbai');

INSERT INTO Orders VALUES
(1, 1, 1, 1, '2024-09-10 12:00:00', 35, 430.00, 'Delivered'),
(2, 2, 2, 2, '2024-09-10 13:30:00', 45, 600.00, 'Delivered'),
(3, 3, 3, 3, '2024-09-11 19:00:00', 70, 350.00, 'Delivered'),
(4, 4, 4, 4, '2024-09-12 18:15:00', NULL, 120.00, 'Cancelled'),
(5, 1, 1, 4, '2024-09-13 21:00:00', 50, 180.00, 'Delivered');

INSERT INTO OrderItems VALUES
(1, 1, 1, 1),
(2, 1, 2, 1),
(3, 2, 3, 1),
(4, 3, 4, 1),
(5, 5, 2, 1);

INSERT INTO Ratings VALUES
(1, 1, 4.5, 5.0),
(2, 2, 4.8, 4.7),
(3, 3, 3.9, 3.5),
```

```
(4, 5, 4.2, 4.0);
```

🧠 5 SQL Questions (Increased Difficulty)

🟢 (Easy)

Q1: List all customers and their most recent order date.

Hint: Use `MAX(order_datetime)` grouped by `customer_id`.

🟡 (Medium)

Q2: Find the average delivery time and total orders per city.

Hint: Join `Orders`, `Restaurants`, group by restaurant city, and exclude NULL `delivery_time_minutes`.

🟠 (Hard)

Q3: Identify top 3 restaurants with highest **average customer rating**.

Hint: Join `Ratings` → `Orders` → `Restaurants` and use `AVG(customer_rating)` with `ORDER BY`.

🔵 (Difficult)

Q4: For each delivery partner, find their **average delivery time** and **number of unique customers served**.

Hint: Use `COUNT(DISTINCT customer_id)` and group by `partner_id`.

Optimize with index on `(partner_id, order_status)`.

🔴 (Expert)

Q5: Using **window functions**, determine for each city:

- The top restaurant by **total revenue**, and
- The **rank of each restaurant** by revenue.

Output columns: `city, restaurant_name, total_revenue, rank_in_city`.

Hint: Use `SUM(total_amount)` grouped by restaurant and `RANK() OVER (PARTITION BY city ORDER BY SUM(total_amount) DESC)`.

⚙️ Advanced Optimization Tips

- Add **composite indexes**:
  - `(restaurant_id, order_status)` in `Orders`
  - `(partner_id, delivery_time_minutes)` for time analysis
- Use **temporary tables or CTEs** to pre-aggregate revenue for reports.
- **Partition orders** by `order_datetime` for performance scaling.
- **Normalize MenuItems** to prevent redundancy; consider `MenuCategory` in production.