1. Introduction

This document presents the **high-level database design** for an **Online Food Delivery Platform** (similar to Uber Eats, Glovo, or DoorDash). The platform allows **customers** to browse restaurants, place orders, make payments, and track deliveries. **Drivers** deliver orders, while **restaurant owners** manage menus.

The goal of this design is to capture the **core entities**, their **attributes**, and the **relationships** between them, ensuring referential integrity with **primary keys (PK), foreign keys (FK), unique constraints, and check constraints**.

2. Core Entities & Attributes

2.1 Users

Represents all platform users: customers, drivers, restaurant owners, and admins.

- `user_id` (PK) – BIGINT, auto-increment
- `name` – VARCHAR(100), not null
- `email` – VARCHAR(100), unique, not null
- `phone` – VARCHAR(20), unique
- `password_hash` – VARCHAR(255), not null
- `role` – ENUM('CUSTOMER','DRIVER','RESTAURANT_OWNER','ADMIN'), not null
- `created_at` – TIMESTAMP, default current timestamp

Constraints:

- Each email and phone must be unique.

- Role ensures authorization boundaries.

2.2 Restaurants

Represents restaurants managed by users with the `RESTAURANT_OWNER` role.

- `restaurant_id` (PK) – BIGINT, auto-increment
- `owner_id` (FK → Users.user_id), not null
- `name` – VARCHAR(150), not null
- `address` – TEXT, not null
- `phone` – VARCHAR(20)

Constraints:

- Each restaurant belongs to a registered owner.

2.3 MenuItems

Represents food and drinks offered by restaurants.

- `item_id` (PK) – BIGINT, auto-increment
- `restaurant_id` (FK → Restaurants.restaurant_id), not null
- `name` – VARCHAR(150), not null
- `description` – TEXT
- `price` – DECIMAL(10,2), not null

- `available` – BOOLEAN, default true

Constraints:

- A menu item must be linked to exactly one restaurant.
- Price must be positive.

2.4 Orders

Represents customer orders from restaurants.

- `order_id` (PK) – BIGINT, auto-increment
- `customer_id` (FK → Users.user_id), not null
- `restaurant_id` (FK → Restaurants.restaurant_id), not null
- `order_status` – ENUM('PENDING','ACCEPTED','PREPARING','DELIVERING','COMPLETED','CANCELLED'), default 'PENDING'
- `total_amount` – DECIMAL(10,2), not null
- `created_at` – TIMESTAMP, default current timestamp

Constraints:

- Each order belongs to a single customer and a single restaurant.
- Total amount must equal the sum of associated `OrderItems`.

2.5 OrderItems

Represents individual menu items inside an order.

- `order_item_id` (PK) – BIGINT, auto-increment
- `order_id` (FK → Orders.order_id), not null
- `item_id` (FK → MenuItems.item_id), not null
- `quantity` – INT, not null, check > 0
- `price` – DECIMAL(10,2), not null

Constraints:

- Each order can have multiple items.
- Quantity must be greater than 0.

2.6 Payments

Represents payments made for orders.

- `payment_id` (PK) – BIGINT, auto-increment
- `order_id` (FK → Orders.order_id, UNIQUE), not null
- `amount` – DECIMAL(10,2), not null
- `payment_method` – ENUM('CREDIT_CARD','DEBIT_CARD','WALLET','CASH_ON_DELIVERY'), not null
- `payment_status` – ENUM('PENDING','SUCCESS','FAILED'), default 'PENDING'

- `paid_at` – TIMESTAMP

Constraints:

- One order maps to exactly one payment record.

2.7 Deliveries

Represents delivery assignments and tracking.

- `delivery_id` (PK) – BIGINT, auto-increment
- `order_id` (FK → Orders.order_id, UNIQUE), not null
- `driver_id` (FK → Users.user_id), not null
- `Delivery_status` → ENUM('ASSIGNED','PICKED_UP','ON_THE_WAY','DELIVERED','FAILED'), default 'ASSIGNED'
- `estimated_time` – TIMESTAMP
- `delivered_at` – TIMESTAMP

Constraints:

- Each order has at most one delivery.
- Each delivery must be assigned to a driver.

2.8 Reviews

Represents customer feedback on orders.

- `review_id` (PK) – BIGINT, auto-increment
- `order_id` (FK → Orders.order_id), not null
- `customer_id` (FK → Users.user_id), not null
- `rating` – INT, check between 1 and 5
- `comment` – TEXT
- `created_at` – TIMESTAMP, default current timestamp

Constraints:

- A customer can only review orders they placed.
- Rating must be between 1 and 5.

---

3. Relationships Summary

- **Users (1) → (Many) Restaurants**
- **Users (1) → (Many) Orders**
- **Users (1) → (Many) Deliveries**
- **Restaurants (1) → (Many) MenuItems**
- **Restaurants (1) → (Many) Orders**
- **Orders (1) → (Many) OrderItems**
- **Orders (1) → (1) Payments**
- **Orders (1) → (1) Deliveries**

- **Orders (1) → (Many) Reviews**
- **MenuItems (1) → (Many) OrderItems**

---

4. Entity Relationship Diagram (ERD)

At high-level (textual view):

```
Users (user_id) ----------------< Restaurants (owner_id)
Users (user_id) ----------------< Orders (customer_id)
Users (user_id) ----------------< Deliveries (driver_id)

Restaurants (restaurant_id) ----< MenuItems (restaurant_id)
Restaurants (restaurant_id) ----< Orders (restaurant_id)

Orders (order_id) ---------------< OrderItems (order_id)
Orders (order_id) --------------- Payments (order_id)
Orders (order_id) --------------- Deliveries (order_id)
Orders (order_id) ---------------< Reviews (order_id)

MenuItems (item_id) -------------< OrderItems (item_id)
```