Rendezvous Restaurant Database Documentation

1. Overview

This document provides a detailed overview of the Rendezvous Restaurant web application database. It outlines the database structure, tables, relationships, constraints, and key SQL definitions to ensure a well-structured, normalized relational database using MySQL.

2. Database Structure

The database consists of **ten** main tables:

1. **Users** - Stores customer and staff details.
2. **Tables** - Represents the physical tables available in the restaurant.
3. **Reservations** - Manages table reservations.
4. **Categories** - Organizes menu items into categories.
5. **Menu** - Stores food and drink options.
6. **Orders** - Records all orders (dine-in, takeaway, and delivery).
7. **Order_Items** - Links orders with menu items.
8. **Takeaway** - Tracks takeaway-specific orders.
9. **Payment** - Handles transaction details.
10. **Indexes** - Optimizes database performance.

3. Table Definitions

3.1 Users Table

Stores information about customers, staff, and admins.

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    role ENUM('admin', 'staff', 'customer') NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

3.2 Tables Table

Represents the restaurant's physical tables.

```
CREATE TABLE tables (
    id INT AUTO_INCREMENT PRIMARY KEY,
    table_number VARCHAR(10) UNIQUE NOT NULL,
    location VARCHAR(50) NOT NULL,
    seats INT NOT NULL,
    is_available BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

3.3 Categories Table

Groups menu items into categories.

```
CREATE TABLE categories (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

3.4 Menu Table

Stores restaurant menu items.

```
CREATE TABLE menu (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    image VARCHAR(255),
    price DECIMAL(10, 2) NOT NULL,
    category_id INT NOT NULL,
    is_available BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE RESTRICT
);
```

3.5 Orders Table

Records both dine-in and takeaway orders.

```
CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    order_type ENUM('dine-in', 'takeaway', 'delivery') NOT NULL,
    total_price DECIMAL(10, 2) NOT NULL,
    status ENUM('pending', 'preparing', 'ready', 'delivered', 'completed', 'cancelled') NOT NULL DEFAULT 'pending',
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE RESTRICT
);
```

3.6 Order_Items Table

Tracks items in each order.

```
CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    menu_id INT NOT NULL,
    quantity INT NOT NULL,
    price_at_time DECIMAL(10, 2) NOT NULL,
    special_instructions TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE CASCADE,
    FOREIGN KEY (menu_id) REFERENCES menu(id) ON DELETE RESTRICT
);
```

3.7 Reservations Table

Stores reservations made by users.

```
CREATE TABLE reservations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    table_id INT NOT NULL,
    time DATETIME NOT NULL,
    number_of_people INT NOT NULL,
    status ENUM('pending', 'confirmed', 'cancelled', 'completed') NOT NULL DEFAULT 'pending',
    notes TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE RESTRICT,
    FOREIGN KEY (table_id) REFERENCES tables(id) ON DELETE RESTRICT
);
```

3.8 Takeaway Table

Tracks takeaway-specific orders.

```
CREATE TABLE takeaway (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT UNIQUE NOT NULL,
    status ENUM('preparing', 'ready_for_pickup', 'picked_up') NOT NULL DEFAULT 'preparing',
    estimated_pickup_time DATETIME,
    pickup_notes TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE CASCADE
);
```

3.9 Payment Table

Handles payments made by users.

```
CREATE TABLE payment (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT UNIQUE NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    payment_method ENUM('cash', 'credit_card', 'debit_card', 'mobile_payment', 'online') NOT NULL,
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    transaction_id VARCHAR(100),
    status ENUM('pending', 'completed', 'failed', 'refunded') NOT NULL DEFAULT 'pending',
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE RESTRICT
);
```

4. Indexing & Performance Optimization

```
CREATE INDEX idx_orders_user_id ON orders(user_id);
CREATE INDEX idx_order_items_order_id ON order_items(order_id);
CREATE INDEX idx_order_items_menu_id ON order_items(menu_id);
CREATE INDEX idx_menu_category_id ON menu(category_id);
CREATE INDEX idx_reservations_user_id ON reservations(user_id);
CREATE INDEX idx_reservations_table_id ON reservations(table_id);
CREATE INDEX idx_reservations_time ON reservations(time);
```