Rendezvous Restaurant Website Database Documentation

1. Database Overview

The **Rendezvous_DB** is a relational database designed to support the functionality of the **Rendezvous Restaurant Website**. The database enables management of user accounts, menu items, customer orders, shopping carts, notifications, and reservations. This database structure ensures data integrity, scalability, and efficient handling of customer interactions.

2. Database Tables

The database consists of multiple tables, each serving a specific purpose for the system's operations. Below is a detailed description of each table and its relationships.

2.1. Users Table

The `users` table stores information about the customers and administrators of the restaurant.

Column Name	Data Type	Description
`user_id`	INT (Primary Key)	Auto-incrementing unique identifier for each user.
`user_name`	VARCHAR(255)	The name of the user.
`user_email`	VARCHAR(255) UNIQUE	The email of the user, must be unique for each user.
`user_password`	VARCHAR(255)	Hashed password stored securely using bcrypt.
`user_role`	ENUM('customer', 'admin')	The role of the user, can be either 'customer' or 'admin'. Default is 'customer'.
`created_at`	TIMESTAMP	Timestamp indicating when the user account was created.
`updated_at`	TIMESTAMP	Timestamp that updates whenever user details are modified.

Relationships:

- Linked to the `orders`, `cart`, `notifications`, and `reservations` tables through the `user_id` column.

2.2. Categories Table

The `categories` table stores different categories for menu items, ensuring that items are properly classified.

Column Name	Data Type	Description
`category_id`	INT (Primary Key)	Auto-incrementing unique identifier for each category.
`category_name`	VARCHAR(255) UNIQUE	Name of the category (e.g., Desserts, Main Meals, etc.).
`created_at`	TIMESTAMP	Timestamp indicating when the category was created.

Relationships:

- The `category_id` is referenced in the `menu_items` table to classify items.

2.3. Menu Items Table

The `menu_items` table stores the restaurant's menu items and their details.

Column Name	Data Type	Description
`menu_item_id`	INT (Primary Key)	Auto-incrementing unique identifier for each menu item.
`menu_item_name`	VARCHAR(255)	Name of the menu item.
`menu_item_price`	DECIMAL(10, 2)	Price of the menu item.
`menu_item_image`	VARCHAR(255)	Filename for the image representing the menu item.
`category_id`	INT	Foreign key that references the `categories` table.

`created_at`	TIMESTAMP	Timestamp indicating when the menu item was created.
`updated_at`	TIMESTAMP	Timestamp that updates when menu item details are modified.

Relationships:

- References `category_id` from the `categories` table.
- Referenced in the `cart` and `order_items` tables.

2.4. Cart Table

The `cart` table stores items that customers add to their shopping carts.

Column Name	Data Type	Description
`cart_id`	INT (Primary Key)	Auto-incrementing unique identifier for each cart entry.
`user_id`	INT	Foreign key referencing the `users` table.
`menu_item_id`	INT	Foreign key referencing the `menu_items` table.
`quantity`	INT	Number of units of the menu item in the cart.
`created_at`	TIMESTAMP	Timestamp indicating when the cart entry was created.
`updated_at`	TIMESTAMP	Timestamp that updates whenever the cart entry is modified.

Relationships:

- References `user_id` from the `users` table and `menu_item_id` from the `menu_items` table.

2.5. Notifications Table

The `notifications` table stores messages sent to users, including information about order status, promotions, etc.

Column Name	Data Type	Description
`notification_id`	INT (Primary Key)	Auto-incrementing unique identifier for each notification.
`user_id`	INT	Foreign key referencing the `users` table.
`notification_message`	TEXT	The content of the notification sent to the user.
`is_read`	BOOLEAN	Indicates whether the notification has been read (TRUE/FALSE).
`created_at`	TIMESTAMP	Timestamp indicating when the notification was created.

Relationships:

- References `user_id` from the `users` table.

2.6. Orders Table

The `orders` table stores customer orders, including details of payment and order status.

Column Name	Data Type	Description
`order_id`	INT (Primary Key)	Auto-incrementing unique identifier for each order.
`user_id`	INT	Foreign key referencing the `users` table.
`total_amount`	DECIMAL(10, 2)	Total cost of the order.
`payment_status`	ENUM('pending', 'paid', 'failed')	Tracks the payment status of the order.
`order_status`	ENUM('pending', 'completed', 'cancelled')	Tracks the order’s status.
`created_at`	TIMESTAMP	Timestamp indicating when the order was placed.
`updated_at`	TIMESTAMP	Timestamp that updates when order details are modified.

Relationships:

- References `user_id` from the `users` table.
- Referenced in the `order_items` table.

2.7. Order Items Table

The `order_items` table links orders with the specific items ordered by customers.

Column Name	Data Type	Description
`order_item_id`	INT (Primary Key)	Auto-incrementing unique identifier for each order item.
`order_id`	INT	Foreign key referencing the `orders` table.
`menu_item_id`	INT	Foreign key referencing the `menu_items` table.
`quantity`	INT	Number of units ordered for this menu item.
`item_price`	DECIMAL(10, 2)	Price of the menu item at the time of ordering.
`created_at`	TIMESTAMP	Timestamp indicating when the order item was added.

Relationships:

- References `order_id` from the `orders` table and `menu_item_id` from the `menu_items` table.

2.8. Reservations Table

The `reservations` table stores customer reservations for dining at the restaurant.

Column Name	Data Type	Description
`reservation_id`	INT (Primary Key)	Auto-incrementing unique identifier for each reservation.
`user_id`	INT	Foreign key referencing the `users` table.
`reservation_name`	VARCHAR(255)	Name of the person making the reservation.
`reservation_email`	VARCHAR(255)	Email for confirmation.
`reservation_date`	DATETIME	Date and time of the reservation.
`reservation_guests`	INT	Number of guests for the reservation.
`reservation_special_request`	TEXT	Any special requests made by the customer.
`created_at`	TIMESTAMP	Timestamp indicating when the reservation was made.

Relationships:

- References `user_id` from the `users` table.

---

3. Database Relationships

- **Users**: Linked to `orders`, `cart`, `notifications`, and `reservations`.
- **Categories**: Linked to `menu_items`.
- **Orders**: Linked to `order_items`.
- **Menu Items**: Linked to `cart` and `order_items`.
- **Foreign Key Constraints**: Cascading delete is applied to ensure that associated records are removed when a user or an order is deleted, maintaining database integrity.

---

4. Sample Data Insertion

4.1. Categories

```
INSERT INTO categories (category_name) VALUES
('Desserts'), ('Sandwiches'), ('Light Meals'), ('Main Meals'),
('Cold Beverages'), ('Hot Beverages');
```

4.2. Menu Items

```
INSERT INTO menu_items (menu_item_name, menu_item_price, menu_item_image, category_id) VALUES
('Delectable Chocolate Cake', 30.00, 'chocolate-cake.jpg', 1),
('Cheese & Tomato', 22.00, 'Cheese-Tomato.jpg', 2),
('Creamy Chicken Fettuccine', 45.00, 'Chicken-Fettuccine.jpg', 4),
```

```
('Cool-drinks', 17.00, 'drinks.jpg', 5),
('Espresso', 15.00, 'Espresso.jpg', 6);
```