Database Design Document – Restaurant Management System

1. Introduction

1.1 Purpose

This document provides the database design for the Restaurant Management System, including the schema, relationships, indexing strategies, and data integrity constraints.

1.2 Scope

The database will store and manage customer information, reservations, table availability, menu details, staff roles, and system notifications. It will ensure data integrity, security, and efficient retrieval to provide a seamless restaurant management experience.

1.3 Target Audience

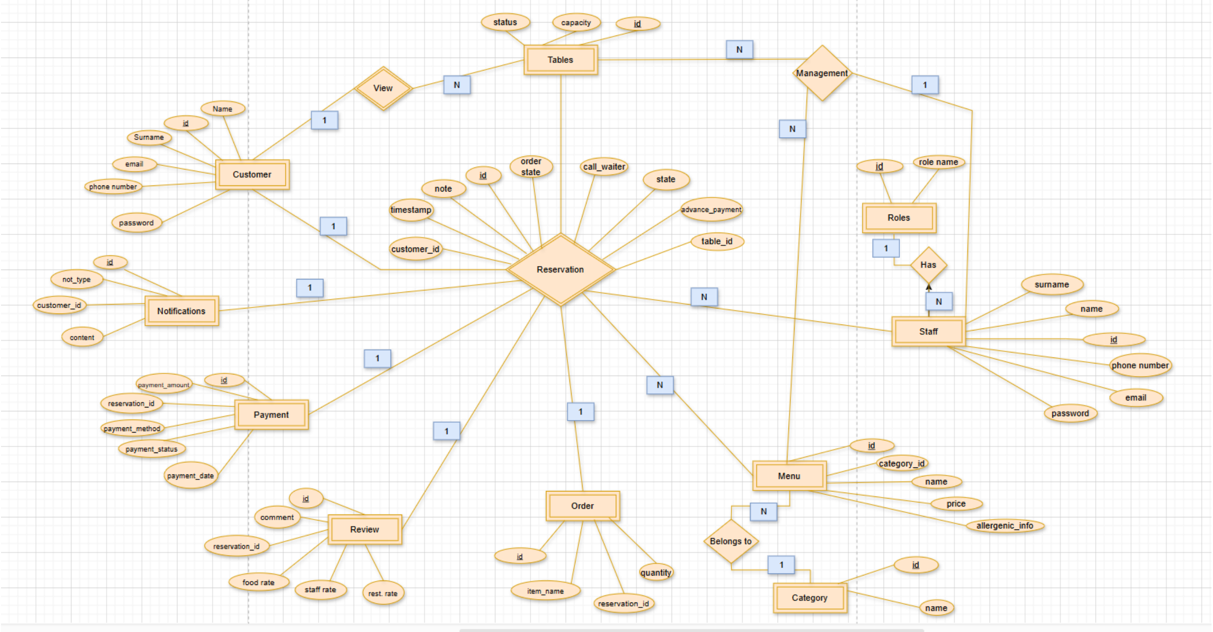
- Database administrators
- Backend developers
- System architects

2. Database Architecture

2.1 Database Management System

- **Type**: Relational Database Management System (RDBMS)
- **Preferred DBMS**: PostgreSQL
- **Normalization**: Third Normal Form (3NF) to optimize storage and eliminate redundancy

2.2 ER Diagram



ER Diagram of Restaurant Management System

3. Database Schema

3.1 Customer table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique customer identifier
name	VARCHAR(50)	NOT NULL	Customer's first name
surname	VARCHAR(50)	NOT NULL	Customer's last name
email	VARCHAR(100)	UNIQUE, NOT NULL	Unique email address
phone_number	VARCHAR(15)	UNIQUE,NOT NULL	Customer's phone number
password	VARCHAR(255)	UNIQUE,NOT NULL	Hashed password

3.2 Tables Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique table identifier
capacity	INT	NOT NULL	Maximum seating capacity
r.status	ENUM	reserved', 'empty')	It provides information about the reservation status.

3.3 Reservation Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique reservation identifier
customer_id	INT (FK)	REFERENCES customer(id) ON DELETE CASCADE	Reference to Customer (customer_id)
date	DATE	NOT NULL	Reservation date
time	TIME	NOT NULL	Reservation time
order_status	TEXT	'pending', 'confirmed', 'canceled')	('pending', 'confirmed', 'canceled')
t.status	TEXT	('active', 'completed', 'canceled')	('active', 'completed', 'canceled')
call_waiter	BOOLEAN	DEFAULT	Request waiter assistance
advance_payment	BOOLEAN	DEFAULT	Indicates advance payment status
in_restaurant	BOOLEAN	DEFAULT	Is customer in restaurant

3.4 Roles Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique role identifier
roleName	TEXT	NULLABLE	Role name (e.g., Manager, Chef,Waiter)

3.5 Staff Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique staff identifier
name	VARCHAR(50)	NOT NULL	Staff member's first name
surname	VARCHAR(50)	NOT NULL	Staff member's last name
phone_number	VARCHAR(15)	UNIQUE,NOT NULL	Staff member's phone number
email	VARCHAR(100)	UNIQUE,NOT NULL	Staff member's email
role_id	INT (FK)	REFERENCES roles(id) ON DELETE CASCADE	Reference to Roles (id)
password	VARCHAR(255)	UNIQUE,NOT NULL	Hashed password

3.6 Menu Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique menu item identifier
category_id	INT (FK)	REFERENCES category(id) ON DELETE CASCADE	Reference to Category (id)
itemName	VARCHAR(100)	NOT NULL	Menu item name
price	DECIMAL(10,2)	NOT NULL	Price of the menu item
allergens	TEXT	NULLABLE	Allergenic information

3.7 Category Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique category identifier
category_name	TEXT	(‘food’, ’drinks’, ‘desserts’)	Name of the category

3.8 Order Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique order identifier
reservation_id	INT (FK)	REFERENCES roles(id) ON DELETE CASCADE	Reference to Reservation (id)
item_name	VARCHAR(100)	NOT NULL	Ordered item name
quantity	INT	NOT NULL	Quantity of ordered item

3.9 Review Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique review identifier
reservation_id	INT (FK)	REFERENCES rezervation(id) ON DELETE CASCADE	Reference to Reservation (id)
comment	TEXT	NULLABLE	Review comment
food_rate	INT (1-5)	CHECK (rating BETWEEN 1 AND 5)	Food rating (1 to 5)
staff_rate	INT (1-5)	CHECK (rating BETWEEN 1 AND 5)	Staff service rating (1 to 5)
restaurant_rate	INT (1-5)	CHECK (rating BETWEEN 1 AND 5)	Restaurant ambiance rating (1 to 5)

3.10 Notifications Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique notification identifier
reservation_id	INT (FK)	REFERENCES rezervation(id) ON DELETE CASCADE	Reference to Reservations(id)
notification_type	**TEXT**	NULLABLE	(‘call waiter’, reservation update)
text	TEXT	NULLABLE	Notification content

3.11 Payment Table

Column	Type	Constraints	Description
id	INT (PK)	AUTO_INCREMENT, PRIMARY KEY	Unique payment identifier
reservation_id	INT (FK)	REFERENCES rezervation(id) ON DELETE CASCADE	Reference to Reservation (id)
amount	DECIMAL(10,2)	NOT NULL	Total payment amount
payment_method	VARCHAR(50)	(Cash, Credit Card,)	Payment method
status	**TEXT**	'pending', 'completed', 'failed')	It checks the payment status.
payment_date	DATETIME	NOT NULL	Date and time of payment

4. Indexing Strategy

• **Primary Keys**: Used on `id` fields to ensure uniqueness and fast lookups.

• **Foreign Keys:** Enforce referential integrity between tables.

• **Indexes:**

- Index on phone_number in Customer Table(For quick access to customer information)
- Index on food_rate in Review Table (For quick access to customer information)
- Index on payment_status in Payment Table(For filtering based on payment status)

5. Security Measures

• **Data Encryption:**

- Passwords stored using bcrypt hashing.
- Sensitive data encrypted using AES-256.

• **Access Control:**

- Role-based access control (RBAC) to restrict unauthorized actions.

• **Audit Logging:**

- All user actions logged for monitoring and security.

6. Scalability & Performance Optimization

• Read Replicas: Database replication for load balancing.

• Caching Mechanism: Implement Redis or Memcached for frequent queries.

• Partitioning Strategy: Time-based partitioning for logs and history tables.

• Connection Pooling: Optimized DB connections for performance.

7. Backup & Disaster Recovery Plan

• Database Backups: Daily full backups, hourly incremental backups.

• File Storage Backups: Version-controlled backups of static assets.

• Failover Strategy: Multi-region deployment for high availability.

• Recovery Time Objective (RTO): Less than 15 minutes.

• Recovery Point Objective (RPO): Less than 5 minutes

8. Conclusion

This database design document provides a robust foundation for the Online Restauran Management System. It ensures scalability, security, and efficiency while maintaining data integrity and accessibility.