



RESTAURANT MANAGEMENT SYSTEM

END-TO-END SQL-DRIVEN RESTAURANT MANAGEMENT SYSTEM WITH AUTOMATION & ANALYTICS.



PROJECT SUMMARY:

Developed a **SQL-based Restaurant Management System** to streamline operations such as customer management, ordering, billing, table allocation, payments, and inventory. Implemented **stored procedures, functions, and triggers** for workflow automation, including real-time inventory updates, loyalty-based billing, and price-change logging. Added **role-based access control** for secure data handling and built **analytical SQL reports** to track revenue, top-selling items, peak hours, and stock status for data-driven decision-making.

PROBLEM STATEMENT & OBJECTIVES:

Restaurants often struggle with managing day-to-day operations such as order processing, billing, table allocation, staff coordination, and inventory tracking. Manual processes lead to delays, errors, stock mismanagement, and lack of data visibility, making it difficult for management to monitor performance and make informed decisions.

Objective:

- To design a **centralized SQL database** for managing customers, staff, menu, orders, tables, payments, and inventory.
- To **automate routine restaurant operations** using stored procedures, functions, and triggers.
- To enable **real-time inventory tracking** and generate low-stock alerts for efficient stock management.
- To implement **role-based access control (RBAC)** for secure and controlled data access.



DATASET OVERVIEW:

The project uses a **structured relational database** designed to capture core restaurant operations. It consists of multiple interconnected tables that store information about customers, staff, menu items, orders, payments, dining tables, and inventory. The dataset supports end-to-end tracking of restaurant workflows, from order placement to billing and stock updates.

Data Entities Covered:

- **Customers & Staff** – personal details, loyalty points, roles, and contact information.
- **Menu & Categories** – item details, pricing, and menu classifications.
- **Orders & Order Items** – complete order lifecycle with item-level breakdown.
- **Payments** – billing, payment modes, and transaction details.
- **Dining Tables & Reservations** – table status and booking information.
- **Inventory & Ingredients** – stock availability with threshold limits.

TOOLS & APPROACH:

Tools Used:

PostgreSQL for database design, development, and execution of SQL queries.

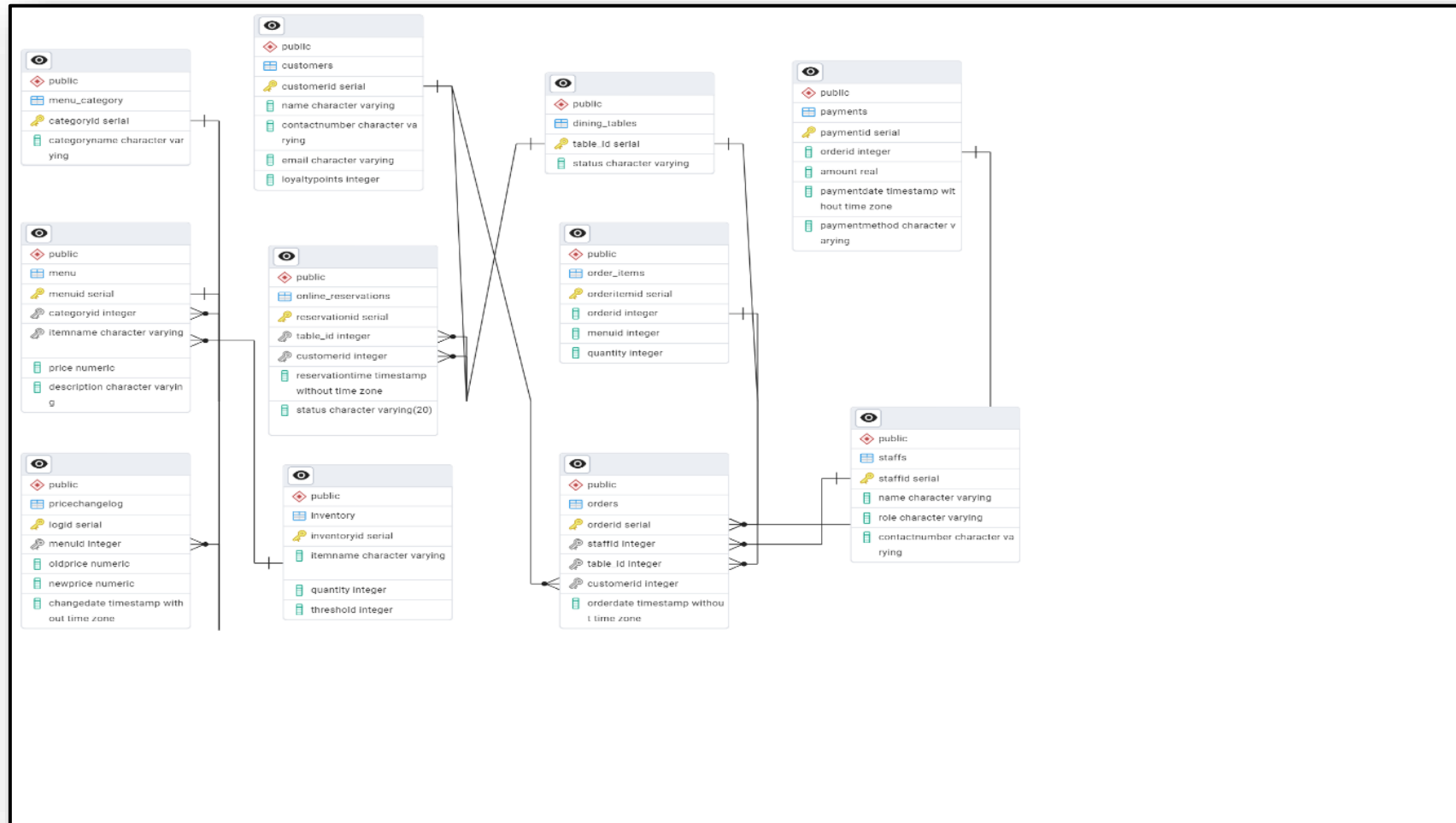
SQL Concepts Applied: DDL, DML, Joins, Views, Window Functions, Stored Procedures, Functions, Triggers, and RBAC.

Approach:

- Designed a **normalized relational database schema** to ensure data integrity and efficient storage.
- Implemented **stored procedures, functions, and triggers** to automate order processing, billing, inventory updates, and logging.
- Applied **role-based access control** to ensure secure user permissions and data protection.
- Developed **analytical SQL queries** to derive insights on sales performance, revenue trends, peak hours, and stock status.

DATABASE DESIGN:

ENTITY-RELATIONSHIP DIAGRAM:



DATABASE DESIGN:

The database is designed using **Third Normal Form (3NF)** to avoid redundancy and maintain data consistency. Primary and foreign key relationships ensure strong referential integrity across modules.

Core Modules:

- Customer & Staff Management
- Menu & Category Setup
- Order & Billing Management
- Inventory & Stock Monitoring
- Payment & Transaction Records
- Reservations and Table Allocation

IMPLEMENTATION:

Database Schema (DDL)

Created relational tables with keys, constraints, and relationships to structure the system.

Automation using SQL

Stored Procedures:

Register customers, process orders, update inventory, generate sales reports

Functions:

Calculate order totals with loyalty discounts, fetch loyalty points, identify low-stock items

Triggers:

Update table availability after booking

Deduct stock when order is placed

Log price changes in menu

Security & Access Control

Role-based access is defined for Admin, Manager, and Staff with restricted privileges as per role responsibilities.

KEY OUTCOMES:

- Automated routine operations, reducing manual workload and errors.
- Real-time stock tracking and alert mechanism for efficient inventory control.
- Improved transparency in billing, table management, and staff performance.
- Enabled data-driven decision-making through analytical SQL reports.

CONCLUSION:

The SQL-based Restaurant Management System successfully streamlines restaurant operations by integrating automation, secure data access, and performance analytics. By eliminating manual dependency, enhancing accuracy, and providing meaningful insights, this system improves overall restaurant efficiency. The project lays the foundation for future scalability into advanced reporting dashboards and application-based ordering systems.