DevifyX Assignment: Advanced Restaurant Ordering System (MySQL-Only)

Objective

Design and implement a comprehensive **Restaurant Ordering System** using **MySQL only**. This system should model real-world restaurant operations in detail and include advanced features beyond basic order and inventory management.

Expanded Requirements

You will build the database layer including schema, stored procedures, triggers, and views that support the following capabilities:

1. Menu Management

- Maintain menu items with detailed attributes:
  - Name, description, price, category, availability
  - Support for **add-ons** or **modifiers** (e.g., extra cheese, no onions) with separate pricing.
- Categorize menu items hierarchically (e.g., Drinks > Alcoholic > Beer).

2. Order Management

- Place and update orders linked to tables and customers.
- Support order item customizations and modifiers.
- Track detailed order statuses:
  - Pending, confirmed, preparing, ready for serving, served, canceled, refunded.
- Support **partial order cancellation** (cancel individual items from an order).
- Maintain **order history** for auditing and reporting.

3. Inventory Management

- Track inventory at the ingredient level.
- Implement **inventory reservation** when orders are placed, preventing overbooking.
- Handle **restocking events** with logs.
- Auto-flag **low-stock alerts** and send notifications (simulate with status flags).

- Support inventory usage reports over time.

4. **Table Management**
   - Manage tables with attributes:
     - Table number, capacity, location/section, status (available, occupied, reserved, cleaning).
   - Support **table reservations** with customer details and booking times.
   - Track seating arrangements and assign orders accordingly.

5. **Kitchen Queue and Workflow**
   - Manage kitchen queue with priorities:
     - Normal, high-priority (VIP orders, rush orders).
   - Track preparation times and estimated completion.
   - Allow marking items as **partially prepared** (e.g., sauce ready, waiting for grill).
   - Support **multiple kitchen stations** (grill, salad, drinks) and route items accordingly.

6. **Reporting & Analytics (via Views and Queries)**
   - Generate reports such as:
     - Daily sales per menu item and category
     - Inventory usage and restock frequency
     - Average preparation time per order and per kitchen station
     - Table turnover rates and peak hours
     - Customer order history and frequent items

7. **User and Role Management**
   - Implement simple user roles (e.g., waiter, chef, manager) with different permissions.
   - Track which user created or updated orders and inventory.

Additional Technical Requirements

- Use **foreign keys, constraints, and indexes** appropriately to ensure data integrity and performance.
- Design stored procedures to handle:
  - Complex order placement with inventory reservation and modifier handling.
  - Status updates with cascading changes (e.g., when all items are served, update order and table status).
  - Inventory restocking and notifications for low stock.
  - Reservation booking and cancellation logic.

- Use **triggers** for automating:
  - Inventory decrement on order confirmation.
  - Status update cascades.
  - Audit logging for critical changes (order updates, inventory adjustments).
- Create **views** for real-time dashboards covering orders, inventory, kitchen queue, and reservations.

Deliverables

- Full MySQL scripts with:
  - Tables, relationships, constraints.
  - Stored procedures and functions.
  - Triggers.
  - Views.
  - Sample data demonstrating all features.
- A README file explaining:
  - Schema design and rationale.
  - How to run and test each feature.
  - Any assumptions or limitations.
- Video Demo

Instructions

- You may use GPT or any AI tools as assistants in your development.
- Write clean, maintainable, and well-commented SQL code.
- Think about real-world constraints like concurrency, data consistency, and scalability.
- Test your system with realistic data scenarios.

Evaluation Criteria

- Completeness and correctness of features.
- Quality and optimization of database design.
- Use of advanced MySQL features (procedures, triggers, views).
- Handling of complex business logic and edge cases.
- Clarity of documentation and code comments.
- Demonstrated understanding of transactional integrity and performance considerations.

Submission

- Deadline:- 5 Days
- Submission Form:- https://forms.gle/HZxnwbzDnmLzMsqTA