DESIGNING AN RDBMS FOR THE MANAGEMENT OF A PIZZA SHOP

Abhilash Antony

Introduction…

In today's fast-paced food service industry, effective management of operations is crucial for success. This report presents the design and implementation of a Relational Database Management System (RDBMS) for Pizza Oven, an imaginary pizza shop dedicated to delivering high-quality products and services. The database system is designed to streamline various aspects of the business, including order processing, inventory management, and staff scheduling.

The importance of database management in a pizza shop cannot be overstated. A well-structured database ensures accurate tracking of customer orders, ingredients, and inventory levels, enabling the shop to operate efficiently. By employing normalization techniques, the database minimizes redundancy and maintains data integrity, further enhancing its effectiveness. It also facilitates better customer relationship management by maintaining detailed records of customer preferences and delivery addresses. This leads to improved customer satisfaction and repeat business, which are essential in a competitive market.

The primary objectives of this database system include:

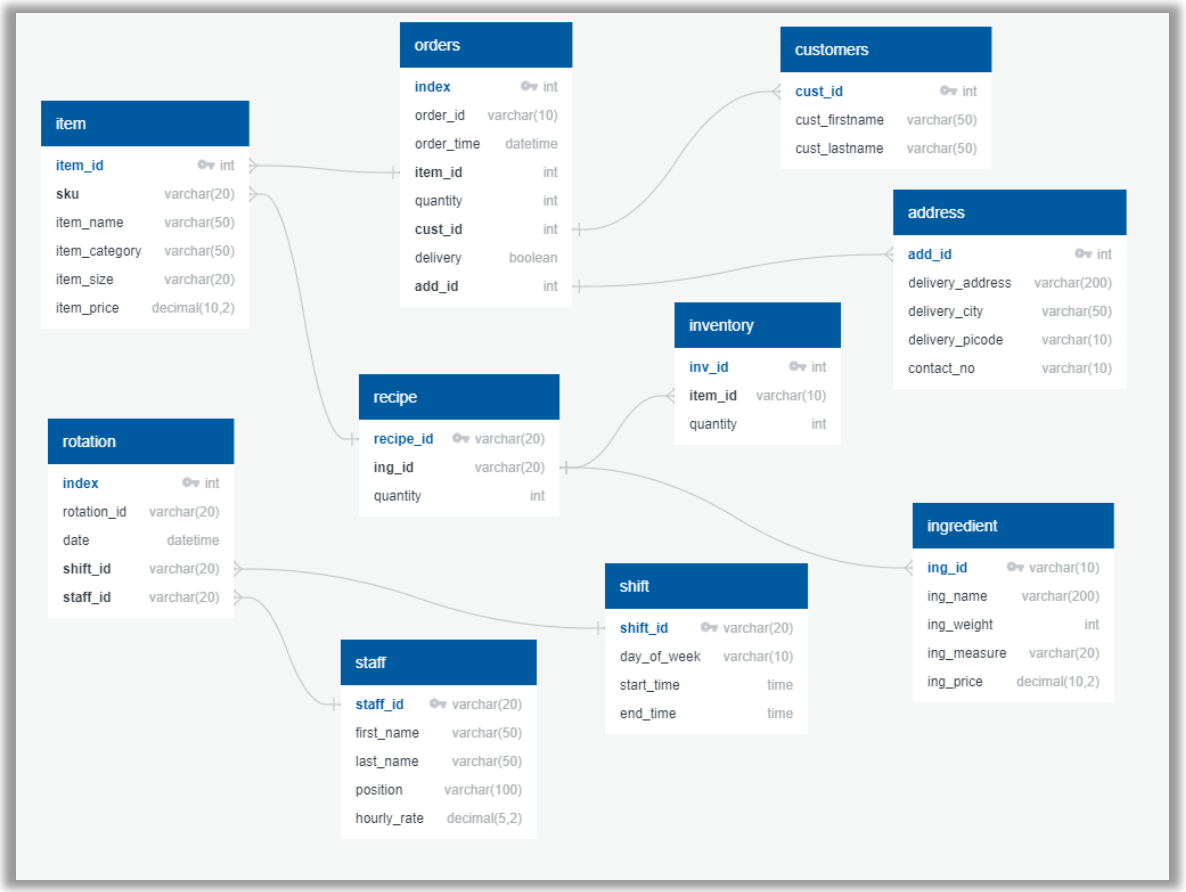
1. **Centralized Data Management:** To create a centralized repository for all critical data, including customer information, orders, inventory, and staff details, ensuring easy access and management.
2. **Enhanced Operational Efficiency:** To streamline the order-taking process and reduce errors through organized data storage, enabling staff to focus more on customer service.
3. **Inventory Control:** To monitor ingredient levels in real-time, helping to prevent shortages and waste, thereby optimizing costs.
4. **Staff Management:** To maintain comprehensive records of staff schedules and shifts, enhancing workforce management.

Through this project, we aim to establish a robust and efficient database system that meets the specific needs of Pizza Oven

Database design…

The Entity-Relationship Diagram (ERD) serves as a visual representation of the entities involved in the Pizza Oven database and the relationships between them. It outlines how various elements such as customers, orders, ingredients, items, staff, and addresses interact within the system.

The ERD helps in understanding the structure of the database, identifying key components, and clarifying how data flows throughout the pizza shop's operations. Each entity is represented by a rectangle, while relationships are depicted with diamonds or lines connecting the entities, illustrating their associations and cardinalities.



This ERD provides a foundation for the database design, ensuring all necessary components are covered for the efficient functioning of Pizza Oven.

Database Schema…

The schema for the Pizza Oven Database consists of multiple interconnected tables, each designed to store specific information relevant to the pizza shop's operations. Below is a description of each table:

Customers Table: This table stores information about the customers who place orders at the pizza shop.

- `cust_id`: Unique identifier for each customer (Primary Key).
- `cust_firstname`: Customer's first name.
- `cust_lastname`: Customer's last name.

Address Table: This table holds delivery addresses associated with customer orders.

- `add_id`: Unique identifier for each address (Primary Key).
- `delivery_address`: The specific delivery address.
- `delivery_city`: The city of the delivery address.
- `delivery_picode`: The postal code of the delivery address.
- `contact_no`: Contact number for the delivery.

Ingredient Table: This table maintains details about the ingredients used in various menu items.

- `ing_id`: Unique identifier for each ingredient (Primary Key).
- `ing_name`: Name of the ingredient.
- `ing_weight`: Weight of the ingredient.
- `ing_measure`: Measurement unit (e.g., grams, liters).
- `ing_price`: Price per unit of the ingredient.

Recipe Table: This table outlines the recipes for each menu item, linking ingredients with the corresponding quantities needed.

- `recipe_id`: Unique identifier for each recipe (Primary Key).
- `ing_id`: Identifier for the ingredient used in the recipe (Foreign Key referencing `Ingredient` table).
- `quantity`: Quantity of the ingredient required for the recipe.

Item Table: This table contains information about the menu items available at the pizza shop.

- `item_id`: Unique identifier for each item (Primary Key).
- `sku`: Stock Keeping Unit (SKU) linked to the recipe (Foreign Key referencing `Recipe` table).
- `item_name`: Name of the menu item.
- `item_category`: Category of the item (e.g., pizza, drink).
- `item_size`: Size of the item (e.g., small, medium, large).
- `item_price`: Price of the item.

Orders Table: This table tracks customer orders, including details about the items ordered and their quantities.

- `index`: Unique index for each order entry (Primary Key).
- `order_id`: Unique identifier for the order.
- `order_time`: Timestamp of when the order was placed.
- `item_id`: Identifier for the ordered item (Foreign Key referencing `Item` table).

- `quantity`: Quantity of the item ordered.
- `cust_id`: Identifier for the customer placing the order (Foreign Key referencing `Customers` table).
- `delivery`: Indicates whether the order is for delivery (boolean).
- `add_id`: Identifier for the delivery address (Foreign Key referencing `Address` table).

Inventory Table: This table manages the stock levels of ingredients required for preparing menu items.

- `inv_id`: Unique identifier for each inventory entry (Primary Key).
- `item_id`: Identifier for the item in inventory (Foreign Key referencing `Recipe` table).
- `quantity`: Quantity of the ingredient available in stock.

Staff Table: This table holds information about the staff members working at the pizza shop.

- `staff_id`: Unique identifier for each staff member (Primary Key).
- `first_name`: Staff member's first name.
- `last_name`: Staff member's last name.
- `position`: Position of the staff member (e.g., cashier, chef).
- `hourly_rate`: Hourly wage of the staff member.

Shift Table: This table describes the working shifts for the staff members.

- `shift_id`: Unique identifier for each shift (Primary Key).
- `day_of_week`: Day of the week for the shift.
- `start_time`: Start time of the shift.
- `end_time`: End time of the shift.

Rotation Table: This table tracks staff rotation through different shifts.

- `index`: Unique index for each rotation entry (Primary Key).
- `rotation_id`: Unique identifier for each rotation (e.g., a specific schedule).
- `date`: Date of the shift.
- `shift_id`: Identifier for the assigned shift (Foreign Key referencing `Shift` table).
- `staff_id`: Identifier for the staff member assigned to the shift (Foreign Key referencing `Staff` table).

The relationships between the tables in the Pizza Oven database are established through primary and foreign keys, ensuring data integrity and enabling efficient data retrieval. Below are the key relationships:

- Customers and Orders: One-to-many relationship via cust_id.
- Orders and Address: Each order links to a delivery address using add_id.
- Orders and Item: Multiple items can be in a single order through item_id.
- Item and Recipe: sku (stock keeping unit) links menu items to recipes.
- Recipe and Ingredient: Recipes reference ingredients via ing_id.
- Inventory and Recipe: Tracks ingredient quantities using item_id.
- Staff and Rotation: Staff shifts are managed via staff_id.
- Shift and Rotation: Shifts are linked to rotations through shift_id.

This schema supports Pizza Oven’s daily operations by accurately tracking orders, ingredients, and staff schedules.

Conclusion

In conclusion, the database design for the Pizza Oven project provides a robust framework for managing essential operations within the pizza shop. By implementing a well-structured schema comprising tables for customers, addresses, ingredients, recipes, items, orders, inventory, and staff, the database system ensures efficient tracking and management of data related to customer orders, inventory levels, and staff schedules.

The relationships established between these tables facilitate seamless data retrieval and integrity, enhancing the overall functionality of the pizza shop's operations. This comprehensive approach to database management not only improves operational efficiency but also supports effective decision-making for future growth.

For those interested in the implementation details, the complete SQL code for the database design can be accessed on my GitHub repository.