Restaurant Management System

Sasha DiVall, Simone Kaplunov, Harshal Shinoy Thachapully
DS5110: Introduction to Data Management
Prof. Roi Yehoshua
Northeastern University, Fall 2023

Introduction

The goal of this database management system is to support the fast-paced and complex nature of restaurants. From managing reservations to maintaining an inventory, it is necessary for managers and chefs of the restaurant to have only the most up-to-date data. That is why the objectives of our system are as follows:

- **Data Storage** - store and manage data efficiently.
- **Data Retrieval** - provide users with quick and reliable access to the data.
- **Data Analysis** - allow for real-time analysis of the data, guiding managers and chefs in making important decisions for their business.
- **Data Security** - maintain a high level of security and keep certain data protected from public view.
- **Data Concurrency** - allow multiple users to access the information in the database at the same time.
- **Scalability** - design the system to scale easily as the data volume and amount of users grow over time.
- **Referential Integrity** - maintain data accuracy through proper data validation and data constraints; ensure that data is always up to date.

Database Design

In designing the database, our decisions were largely informed by the objectives of our system, particularly data security and referential integrity. To ensure these objectives, we designed our database to be in third normal form (3NF), meaning for each non-trivial functional dependency X->Y, at least one of the following holds:

- X is a super key
- Each element of Y is part of some candidate key (Y is a prime attribute).

3NF was enforced in our data design by decomposing the relations in the database to satisfy the above requirements. Another key design choice we made was to not include any redundant data. This was accomplished through the use of foreign key constraints and a lack of repetition of data across tables.

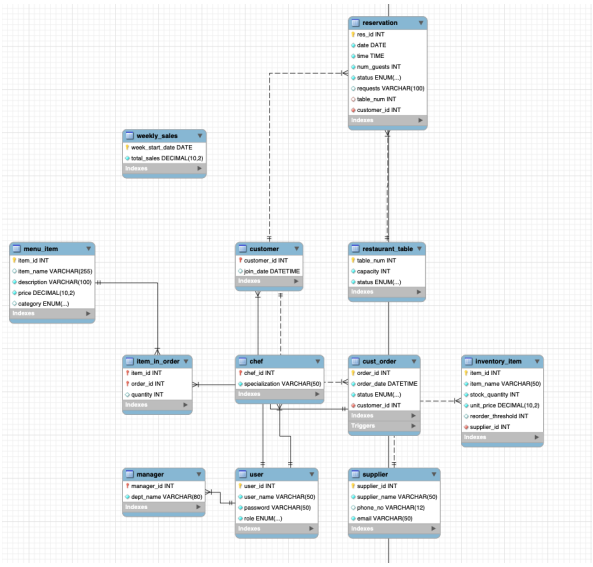

Figure 1: Entity-Relationship Diagram

Stored Procedures, Functions, and Trigger

Trigger

- `update_weekly_sales`
  - Description: A trigger that is activated after an insert, update, or delete operation on the `cust_order` table. It updates the `weekly_sales` table to reflect changes in weekly sales data.
  - Purpose: Ensures that the `weekly_sales` table is always up-to-date, providing real-time information on the restaurant's performance.

Stored Procedures

- `get_user_orders`
  - Description: A stored procedure that retrieves orders associated with a specific user from the `cust_order` table.

    - Purpose: Provides a convenient way to fetch order information for a given user, aiding in order tracking and customer service.
- `add_user_with_role`
    - Description: A stored procedure that adds a new user with a specified role (Manager, Chef, or Customer) to the `user` table and assigns role-specific permissions.
    - Purpose: Simplifies the process of adding new users while ensuring proper role-based access control.

Functions

- `num_reserved`
    - Description: A function that calculates the number of reservations for a given date and time from the `reservation` table.
    - Purpose: Supports efficient reservation management by providing information on the number of reservations for a specific timeframe.
- `calculate_inventory_difference`
    - Description: A function that calculates the difference in inventory before and after an update operation on the `inventory_item` table.
    - Purpose: Helps in tracking changes in inventory levels, facilitating inventory management and order fulfillment.
- `get_order_total`
    - Description: A function that calculates the total cost of items in a customer's order from the `item_in_order` table.
    - Purpose: Enables the application to display accurate order totals, improving transparency for customers during the checkout process.

These stored procedures and functions contribute to the overall efficiency and functionality of the database management system, aligning with the objectives of data retrieval, data analysis, and data security outlined in the system's goals. The trigger ensures that weekly sales data is consistently updated, supporting real-time decision-making for restaurant management.

Data Collection

The data that populates our database was fabricated, AI-generated data that resembles a traditional menu. The data includes fake users with randomly-generated usernames and passwords. However, our database designs allows for easy addition of new customers who are able to choose their own username and password and the addition of new employees who may choose their own username and password.

Application Description

Customer Interface

The opening screen of the application presents an option to either “Log In” or “Sign Up”. New customers sign up, and enter a username and password. To ensure data accuracy, users who choose a username that belongs to another user are prompted to select a new username until they choose a unique username that does not exist in the database. Existing customers, managers, and chefs are able to login with their username and password. In order to maintain the objective of data privacy, the application determines the role of the user based on their username, and presents them with a different interface depending on their role. Customers are presented with the option to make a reservation, modify an existing reservation, place an order, view their active orders’ status, and view their past orders. When making a reservation, the user is prompted to enter the date they wish to reserve a table for, the time of the reservation, the number of guests, and any optional requests. The application then inserts this reservation information into the database immediately. If the user would like to modify a reservation, they are prompted which modification they would like to perform: cancel the reservation, change the date or time, or change the number of guests. Upon modification, the application automatically updates the reservation in the database. To place an order, the user is able to view the menu, view their cart, or checkout at any time. The application extracts the menu information from the database and stores the data in a user-friendly dataframe. When a user selects an item, they view more details about an item, including the description and the price. They can add any quantity to their cart. When viewing their cart, the get_order_total procedure calculates the total price of their cart and allows them to checkout. Once the user checks out, that is when the application commits the new order to the database, allowing chefs in the kitchen to

view the order. Users are able to check the status of their active orders and are notified when the order is complete. Should they choose to reorder an existing order, they are able to view their past orders.

Manager Interface

The Manager Interface plays a pivotal role in overseeing various aspects of restaurant management. It provides tools for managers to efficiently handle administrative tasks, ensuring the smooth operation of the restaurant. Here are the key functionalities embedded in the Manager Interface:

1. Adding a New Employee:
   - Managers can easily add new employees to the system, streamlining the onboarding process.
   - The interface prompts managers to input essential employee details, such as name, password, and role.
   - The system ensures data consistency and integrity by inserting information into the 'user' table and the corresponding role table.
2. Viewing/Updating the Menu:
   - Managers have the capability to view the current menu, including all menu items and their respective prices.
   - A user-friendly display, powered by pandas DataFrame, presents the menu to the manager.
   - The system allows managers to update menu items, facilitating quick adjustments to prices to align with business strategies.
3. Viewing/Updating Inventory:
   - The Inventory Management feature enables managers to keep track of stock levels for various items.
   - Managers can view the current inventory, providing insights into stock quantities for each item.
   - The system empowers managers to update inventory levels, ensuring accuracy in stock management.
4. Viewing Reservations:
   - The interface provides a snapshot of current reservations, offering details such as reservation date, time, and the number of guests.
   - Managers can easily monitor and plan for upcoming reservations, contributing to efficient table management.
5. Exit Option:
   - To enhance usability, the interface includes an option for managers to exit the interface seamlessly.

This comprehensive suite of functionalities empowers restaurant managers to make informed decisions, streamline operations, and maintain control over critical aspects of the business. The real-time nature of the system ensures that managers have access to the most up-to-date information, supporting them in adapting to the dynamic nature of the restaurant industry.

The interface leverages database transactions and error handling to guarantee data consistency and integrity, even in the face of unexpected events. Future iterations may include additional features and optimizations based on user feedback and evolving business requirements.

As it is vital for the manager to view how well the restaurant is performing and what services or products are doing well and to focus on them, the manager also has the option to view reports.

These include:

1. Most commonly ordered menu items

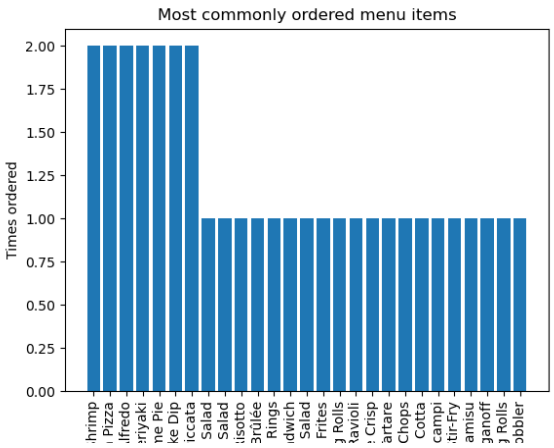

Figure 2: Most Commonly Ordered Menu Items Report

2. Sales over the week

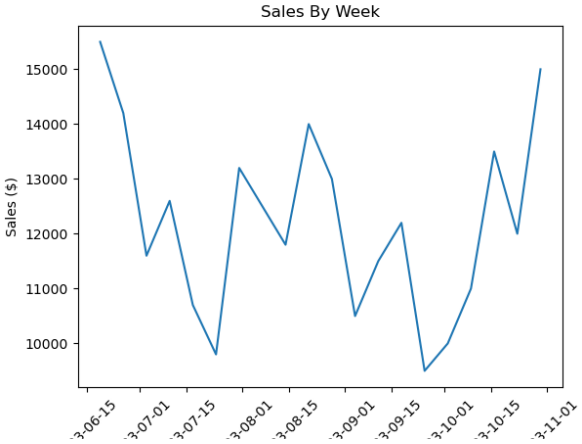

Figure 3: Sales Trends Over the Weeks

3. Number of orders per date

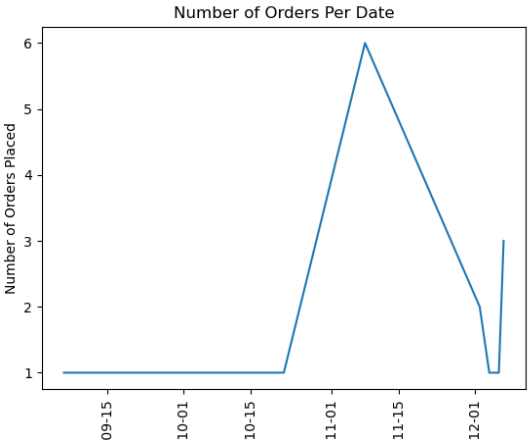

Figure 4: Order Count Trends Per Date

Chef Interface

The Chef Interface is a crucial component of the restaurant management system, tailored to meet the specific needs of chefs. It offers a set of functionalities to streamline kitchen operations and ensure effective communication between the culinary team and other restaurant staff. Here are the primary features integrated into the Chef Interface:

- Update an Order:
  - Chefs can efficiently update the status and details of existing orders through a straightforward process.
  - The interface facilitates seamless communication within the kitchen, allowing chefs to manage order progress.
- View/Update Inventory:
  - The Chef Interface provides tools for chefs to access and update inventory information.
  - Chefs can monitor and make necessary adjustments to ingredient quantities, contributing to efficient kitchen management.

The Chef Interface uses the MySQL database to store and retrieve order and inventory data, enhancing collaboration among kitchen staff and supporting the overall efficiency of the restaurant.

As the system continues to evolve, additional functionalities and optimizations may be introduced based on user feedback and the dynamic requirements of the restaurant industry. The Chef Interface plays an important role in making sure that kitchen operations align with the broader goals of the restaurant.

Conclusions and Future Work

In conclusion, the development of our Restaurant Management System represents a significant milestone in addressing the intricate challenges faced by the restaurant industry. Through meticulous database design and a user-centric application, our system provides an efficient and intuitive solution for managers, chefs, and customers. The Manager Interface stands out as a cornerstone of our system, offering a suite of functionalities crucial for managerial decision-making and operational control. Managers can seamlessly add new employees, view and update the menu, manage inventory, and monitor reservations. The interface's intuitive design and real-time data updates empower managers to navigate the complexities of restaurant management with ease.

Key Learnings:
Throughout this project, we gained valuable insights into the critical importance of database design in achieving data consistency, integrity, and security. The development process emphasized the significance of user-centric interfaces to enhance the overall user experience.

Reflections on Time Constraints:
With more time, we envision further enhancements to the system. Specifically, additional features in both the Manager and Chef Interfaces could be explored. For managers, an extended suite of analytics and reporting tools could offer deeper insights into sales trends and customer preferences. In the Chef Interface, optimization tools for kitchen efficiency and ingredient tracking might be valuable.

Advice for Future DS 5110 Students:

- Thoroughly Understand Requirements: Invest time in comprehensively understanding the requirements before diving into design and implementation. A clear understanding at the beginning saves time in the long run.
- Iterative Development: Adopt an iterative development approach. Regularly test and refine your system, incorporating feedback from users and stakeholders. This approach ensures that the system aligns closely with real-world needs.
- Data Security Considerations: Prioritize data security from the outset. Implement secure coding practices and regularly audit the system for vulnerabilities. Robust data security measures are essential, especially when dealing with sensitive information.
- Continuous Documentation: Maintain detailed and up-to-date documentation throughout the project. Well-documented code, database schemas, and system architecture aid in understanding and collaboration, both during the development phase and for future enhancements.

In conclusion, our Restaurant Management System lays a foundation for further innovation and refinement. It is a testament to the application of data management principles in solving real-world challenges. As we look to the future, we are excited about the potential for continuous improvement and adaptation to the evolving needs of the restaurant industry.