BUSINESS

At SushiGo, we believe sushi should be fresh, fast, and flavorful. Since opening in 2022 under the leadership of our CEO, João Ribeiro, our restaurant has operated exclusively from our location at **Campus de Campolide, 1070-312 Lisboa**, focusing on **delivery** and **takeaway orders**.

We’ve embraced a delivery and takeaway-only model, enabling us to dedicate ourselves fully to crafting high-quality sushi with speed and precision. While our team takes great care in preparing every order, deliveries are handled by trusted third-party services like Uber Eats, ensuring smooth and reliable service. Customers interact with us entirely through our website and mobile app, where they can explore our menu, place orders, and share feedback using our built-in rating system.

Our team includes **10 skilled staff members**, who prepare and manage orders, supported by **2 experienced managers** overseeing operations. To run our business efficiently, we’ve developed a **relational database** with 10 carefully designed tables. This database is focused on critical areas such as **customers, orders, employees, payments, and menu items**, while also incorporating key elements like **refunds, inventory tracking and logs of menu item changes**. By honing in on these entities, we’ve created a foundation for **analyzing and optimizing our business processes**.

SCHEMA

Now, let’s explore the structure of our relational database by walking through the Entity-Relationship Diagram and the corresponding SQL code for each table.

Starting with the **customers table**: Customers create accounts with their information and earn loyalty points, redeemable for discounts on future purchases. Once an account is set up, customers can place orders, which are recorded in the **orders table**. This table links each order to the corresponding customer and captures key details like the order date, total price, discount applied, order type – ‘Home Delivery’ or ‘Pickup at Store’ - , and delivery address if needed. Customers also have the option to rate their orders, providing valuable feedback.

Each order is linked to specific menu items, which are stored in the **menu\_items table**. This table holds details about each item, including its name, price, availability, and discontinuation information. To manage the many-to-many relationship between orders and menu items, the **order\_items table** connects them and records the quantity of each menu item in an order.

To manage the inventory and ingredient usage, the **ingredients** table tracks essential details such as the name, price, stock quantity, measurement unit, and last updated timestamp. To handle the many-to-many relationship between menu items and ingredients, the **menu\_item\_ingredients** table links them and specifies the quantity of each ingredient required to prepare a menu item. This setup ensures precise inventory tracking and efficient management of ingredient usage.

Once an order is completed, payment details are recorded in the **payments** table, which tracks the payment method, the finalamount including any applied discounts and tax, and the tax identification number (if provided). If multiple payment methods are used for a single order, separate entries are created for each method.

Linked to the **payments** table, the **refunds** table records the refund amount, reason, and method, ensuring proper tracking of refunded transactions.

The **employees** table manages staff information, including contact details, salary, and reporting structure. Employees handling payments and refunds are linked to transactions via their employee ID, providing accountability and proper oversight throughout these processes.

Finally, the **menu\_item\_logs** table tracks changes to menu items, specifically updates to price and availability. It is linked to the **menu\_items** table to identify the item being modified, which already contains information about the employee responsible for creating or updating it. This ensures accurate tracking of changes while maintaining accountability.

TRIGGERS

To automate this process, we implemented a trigger that detects updates to a menu item's price or availability. The trigger seamlessly **logs** individual changes as well as simultaneous updates, providing a clear and detailed audit trail. First, we demonstrate price and availability changes individually, followed by a scenario where both are updated at the same time, showcasing how the trigger efficiently captures all modifications.

Following the trigger that logs price and availability changes, we implemented another trigger designed to automate the updating of **ingredient stock** after an order is placed. This trigger ensures that whenever an item is ordered, the ingredients required to prepare that item are properly deducted from the available stock.

Here’s how the trigger works: After a new order item is added, the trigger updates the ingredient stock by calculating the required quantities based on the menu item ingredients and the quantity of the ordered item. This helps maintain accurate inventory levels in real-time.

INVOICE

Next, we focus on the crucial step of **generating the invoice.** To streamline this process, we created two key views:

**First, the Invoice Header:** This view gathers customer and company information, including the order subtotal, discounts, tax rates, and the final amount paid.

**Second, the Invoice Item Details:** This view lists all items purchased, showing quantities, unit prices, and the total for each item.

Let’s see this in action! By querying these views for a specific payment, we can generate a complete and accurate invoice.

These views work together to provide a comprehensive invoice, capturing all necessary details for both customer and company records, ensuring that the transaction is accurately documented.

QUERIES

Now, let's explore five key business questions that a CEO might focus on, such as profitability, customer segmentation, and trend analysis. After creating the initial queries, we used the **EXPLAIN** command to identify opportunities for optimization. By creating indexes—either on individual columns or combinations of columns - we were able to make small adjustments to some queries, improving performance for faster insights.

**Query1**

This query helps the CEO identify the most and least profitable menu items by comparing their selling prices to production costs, calculated from ingredient prices. The results reveal that the most profitable item is ‘Vin Festa 56 pieces’, yielding a profit of 36.2€, while the least profitable is ‘Nigiri Tuna’, with a profit of just 1.5€. Additionally, the analysis confirms that no items are operating at a loss, ensuring healthy profit margins across the menu. This insight is vital for strategic pricing decisions and optimizing the menu to enhance overall profitability.

**Query2**

This query provides the CEO with valuable insights into customer behavior by gender, looking at factors like age, spending, and preferences for home delivery versus in-store pickup. The results reveal that while all customer categories show a preference for home delivery, males have the highest proportion of deliveries, with 82%. Women, however, place the highest number of orders and spend the most per order, averaging 25.75€. Additionally, the average age of customers is in the early 30s, with younger demographics showing potential for growth. These insights can help shape marketing strategies and logistics planning, allowing for targeted campaigns and more efficient services based on customer age, preferences, and spending patterns.

**Query3**

This query helps the CEO identify key customer segments by age and order type. The ‘26-35’ age group stands out, driving the most revenue, especially through home deliveries, with a total of 1564€. Interestingly, the ‘36-45’ group has the highest average order value for home deliveries at 28.76€, despite fewer orders. These insights suggest that focusing on the ‘26-35’ group for targeted promotions could significantly boost revenue, while also considering higher-value orders from the ‘36-45’ group.

**Query4**

This query reveals to the CEO how the most popular menu items vary depending on when customers registered. For customers who registered in 2024, items like ‘Tartare Salmon Avocado’ and ‘Temakizushi Tuna’ are the top choices, each with 3 orders. In contrast, customers from 2022 show a preference for ‘Combo Mix 30 pieces’ and ‘California roll’, each with 4 orders. These insights help identify changing preferences over time, which can guide menu updates and targeted promotions for different customer groups based on when they joined.

**Query5**

This query helps the CEO identify year-over-year trends in customer growth, order volume, spending, and refunds. The results show that in 2023, total revenue peaked with the highest average payment per order, reflecting strong customer spending. In contrast, 2024 saw a notable balance between new and returning customers, though spending was lower overall. Meanwhile, 2022 stood out for its influx of new customers but also had the highest number of refunds. These insights are crucial for understanding customer retention, order patterns, and how the business is evolving over time, allowing for better resource allocation and targeted strategies.

CONCLUSION

In conclusion, this project provides a comprehensive data-driven approach that empowers the CEO with critical insights into key areas of the business, from profitability and customer behavior to operational trends. By leveraging the power of our relational database, we've created a foundation for continuous optimization and growth. The ability to analyze menu performance, customer preferences, and year-over-year trends equips the company to make informed decisions, improve efficiency, and enhance the customer experience. Ultimately, this project supports SushiGo in achieving its goals of profitability, customer satisfaction, and sustainable growth in a competitive market.