Jangu Okyekole Bistro Management System

Mary Nansikombi and James Bray

Feedback from Teaching Assistant

“Your current amount of tables will require much more work than you should be giving yourselves. Try to focus on fewer things, and see if you can bring it down to no more than 7 tables, if possible. For example, focusing on employees, customers, orders, the meals on their order, and the relations between those. Obviously, you don't have to do exactly that. Let me know if you have any questions. “

Feedback from Ed Discussion Post

optimize the menu. I’d be interested to know specifically what metrics are planned to reach this goal, and how the design of the database is oriented to those priorities. Is it to streamline ingredients used, so bulk ordering discounts can be leveraged? Is it to better forecast demand and ordering patterns, to minimize food waste? Is it to track labor cost and find ways to minimize it? Knowing the specific approach to increase profitability would be helpful to ensure the database is designed to that end, and any future reporting desires are taken into account up front. I’d recommend choosing a couple primary metrics to design around at most.One way that could happen is by streamlining ingredients: the restaurant currently orders 60 different ingredients for their dishes. They hope to use this system to find the most popular dishes and replace meals with lesser-used ingredients with variations of popular dishes. At the end of 6 months, the goal is to reduce the number of ordered ingredients to 40 total (a 33% reduction).

- Split name attribute in customers into first and last name and set both to notNULL
- Same thing in employees entity, also change hours_worked to shift
- Use an enum in the discount category instead of varchar
- Make customer optional in sales in case a customer hasn’t eaten there yet but made a reservation
- Removing ingredients entirely, don’t need to pay attention to those metrics

Actions based on the feedback

We also decided to take one main primary focus of monitoring which meals are bought by customers the most so we optimize them on the menu in order to make more sales and design our database based on this primary goal.
Created a first and last name attribute in the customers and employee table so we personalize their experience.We went ahead and made customer optional in sales since some may reserve but haven't made a sale and if they don't show up we have them saved up in the database. We removed ingredients since its difficult to quantify how much of an ingredient is in each meal. We made the database less complicated and tracked down only the metrics we need for simplicity. We decided to focus primarily on one goal of increasing sales through optimizing our menu and design around that aspect. We also made the discounts to sales (1:M) optional since a customer can choose on whether to take the discount or not on a certain sale

Upgrades to draft version

- Before we had: customers, employees, meals, ingredients, sales, suppliers, supplier purchases, and discounts
- Suggested and implemented: employees, customers, orders(sales).This enabled us to get rid of some tables for simplicity
- Simplified entities to customers, employees, meals, sales (orders), and discounts.
- Focused on one goal which is to increase sales through optimizing the user menu.
- Added first and last name attributes to employees for personalisation.
- Changed the type in discounts category to enum so we have our discounts fixed for the customers.

Project Overview

Jangu Okyekole Bistro is a new Ugandan restaurant quickly gaining popularity in its community so they decided to come up with an advanced system for tracking all of their metrics. The goal of the project is to centralize all of the data that wasn't tracked as meticulously. All aspects of the business will be tracked: customers, sales, employees, meals, ingredients, suppliers, supplier purchases, and discounts. The restaurant has quickly grown to serve 1000 customers a week, so this system will help to manage those customer relations in addition to tracking expenses. Since the restaurant only serves authentic and locally sourced ingredients in their meals, their expenses need to be tracked to ensure they're able to maintain their goal of providing excellent food and

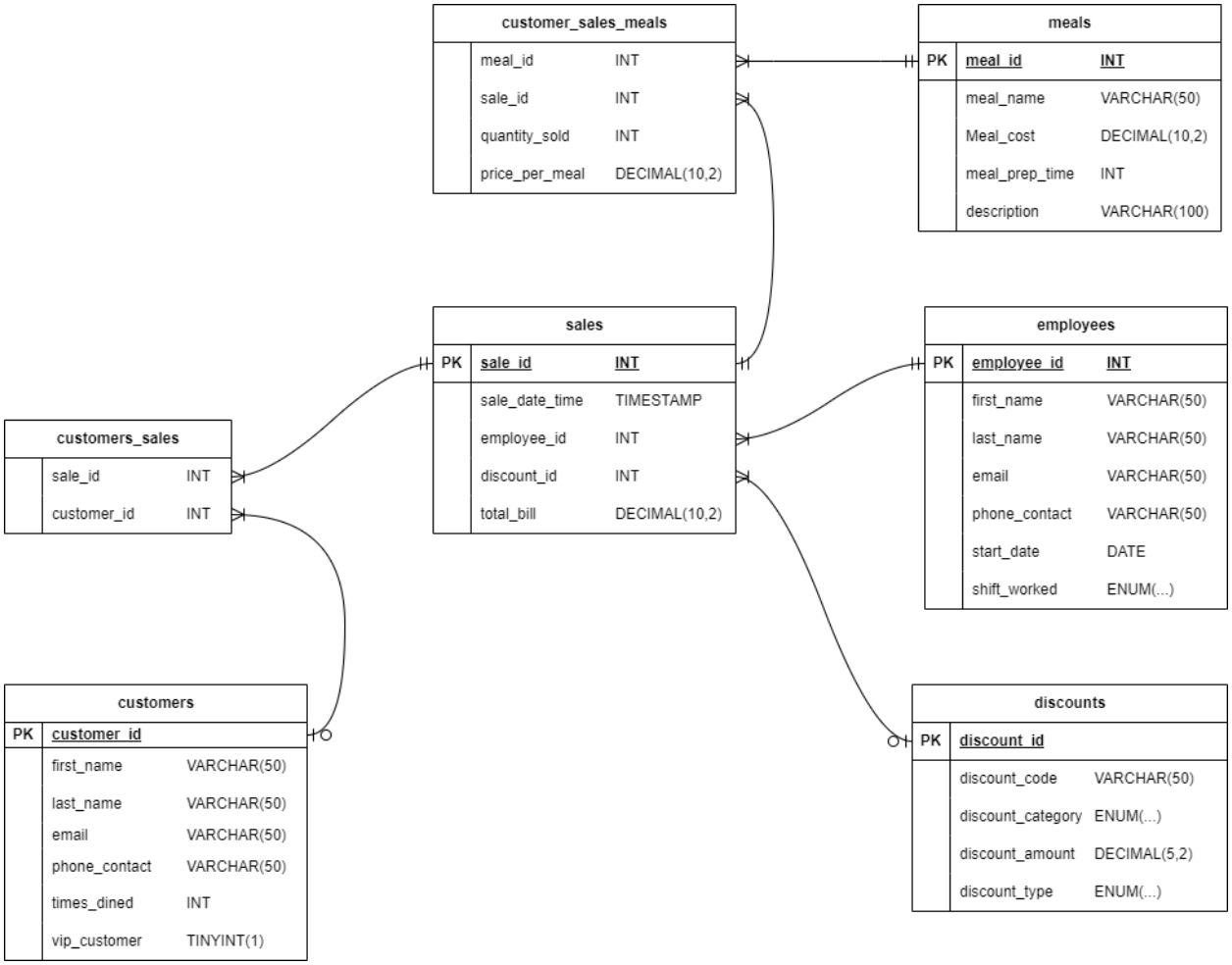
service while maintaining profitable margins. They currently make about $30,000 in revenue per week, but their expenses are ranging from $20,000 to $25,000, and they're looking to improve their margins by tracking how often meals are ordered to optimize their menu. Over the next 3 months, Jangu Okyekole will track which meals are ordered the most and aims to increase their menu size by 20%. The system will also allow the owners to track their employees, when they work, and how long they've been there. They hope that by utilizing this new system, they will be able to retain loyal customers as well as raise profit margins through the optimization of the menu based on meals popular to customers.

Database Outline

- **Customers: Holds details about the customers of the restaurant.**
    - customer_id: int, auto_increment, NOT NULL, PK
    - first_name: varchar(50), NOT NULL
    - last_name: varchar(50), NOT NULL
    - email: varchar(50), NOT NULL, unique
    - phone_contact: varchar(15), NOT NULL, unique
    - times_dined: int, NULL, –calculated
    - vip_customer: tinyint(1), NOT NULL
    - relationship: A M:M relationship with sales.
- **Customers_sales (Intersection Table): Facilitates the M:M relationship between customers and sales.**
    - customer_id: int, NOT NULL, FK
    - sale_id: int, NOT NULL, FK
- **Employees: Details about the employees of the restaurant**
    - employee_id: int, auto_increment, NOT NULL, PK
    - first_name: varchar(50), NOT NULL
    - last_name: varchar(50), NOT NULL
    - email: varchar(50), NOT NULL, unique
    - phone_contact: varchar(15), NOT NULL
    - start_date: date, NOT NULL,
    - shift_worked: ENUM('morning', 'afternoon','evening'), NOT NULL
    - Relationship: A 1:M relationship between Employees and sales with the PK of the employee in a customer
- **Meals: Details of the meals served at the restaurant**
    - meal_ID: int, auto_increment, NOT NULL, PK
    - meal_name: varchar(50), NOT NULL
    - meal_price: int, NOT NULL
    - meal_prep_time: int, NOT NULL

    - description: varchar(100), NOT NULL
    - Relationships: A M:M relationship between meals and sales.
- **Sales: Holds details of the sales made to customers**
    - sale_id: int, auto_increment, NOT NULL, PK
    - sale_date_time: datetime, NOT NULL
    - discount_ID: int
    - total_bill: decimal(10, 2), NOT NULL
    - Relationships: A M:1 relationship with employees, a M:1 relationship with discounts, a M:M relationship between sales and customers, and a M:M relationship between sales and meals.
- **Customer_Sales_Meals (Intersection table):** Facilitates the M:M relationship between sales and meals.
    - meal_id: int, NOT NULL, FK
    - sale_id: int, NOT NULL, FK
    - quantity_sold: int, NOT NULL
    - price_per_meal: decimal (10,2), NOT NULL
- **Discounts: Represents types of discounts available to the customers.**
    - discount_id: int, NOT NULL, PK,
    - discount_code: varchar(50), NOT NULL
    - discount_category: enum('Customer Birthday', 'Special Holiday', 'New Parent'), NULL
    - discount_amount: decimal(5,2), NOT NULL
    - discount_type: enum('Percentage', 'Fixed)
    - Relationship: 1:M relationship between Discounts and customer sales, since each sale can only have one discount applied.

Entity relationship diagram


customer_sales_meals
meal_id INT
sale_id INT
quantity_sold INT
price_per_meal DECIMAL(10,2)
meals
PK meal_id INT
meal_name VARCHAR(50)
Meal_cost DECIMAL(10,2)
meal_prep_time INT
description VARCHAR(100)
sales
PK sale_id INT
sale_date_time TIMESTAMP
employee_id INT
discount_id INT
total_bill DECIMAL(10,2)
employees
PK employee_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
email VARCHAR(50)
phone_contact VARCHAR(50)
start_date DATE
shift_worked ENUM(...)
customers_sales
sale_id INT
customer_id INT
customers
PK customer_id
first_name VARCHAR(50)
last_name VARCHAR(50)
email VARCHAR(50)
phone_contact VARCHAR(50)
times_dined INT
vip_customer TINYINT(1)
discounts
PK discount_id
discount_code VARCHAR(50)
discount_category ENUM(...)
discount_amount DECIMAL(5,2)
discount_type ENUM(...)