Ikigai Barbershop: A Comprehensive Database Case Study

Article on a Database Case Study by

Husna Banu Nayeem

Table of Contents:

Abstract

Effective business management in the modern era requires the integration of data-driven solutions. The Ikigai Barbershop database project aims to design and implement a centralized, robust database system to streamline operations in a barbershop environment. This case study explores the objectives, design, implementation, key features, and future enhancements of the database system.

1. Introduction

In today's fast-paced world, efficient data management is crucial for service-based businesses like barbershops. The "Ikigai Barbershop" project aims to design and implement a robust, centralized database system tailored for a barbershop. This system ensures seamless integration of customer management, appointment scheduling, inventory tracking, and financial operations. By leveraging a well-structured database, barbershops can enhance their operational efficiency, provide better customer service, and ultimately drive business growth.

2. Mission and Objectives

Mission:

To design and implement a robust, centralized database system tailored for a barbershop, ensuring seamless integration of customer management, appointment scheduling, inventory tracking, and financial operations.

Objectives:

Optimize Appointment Scheduling: Streamline the process of booking and managing appointments to reduce wait times and improve customer satisfaction.

Enhance Inventory Management: Keep track of product stock levels, manage supplier information, and ensure that essential items are always available.

Automate Revenue & Expense Tracking: Accurately monitor financial transactions, including payments for services and product sales, to maintain a clear picture of the barbershop's financial health.

Identify Entities

Key entities identified for the database include:

- Customers – Stores customer details and preferences.
- Appointments – Manages booking schedules.
- Employees – Tracks barbers and staff schedules.
- Services – Lists different grooming services offered.

- Products – Maintains inventory records.
- Payments – Handles financial transactions.

3. Database Design and Development

Final List of Tables

The finalized tables include:

PRELIMINARY LIST OF TABLES:

- BRANCH
- Payments
- Customers
- Employees
- Appointments
- Services
- Products
- Suppliers
- Promotions
- Customer Promotions
- Appointment Products

FINAL LIST OF TABLES:

BRANCH

Payment

Customers

Employees

Appointments

Services

Products

Suppliers

Promotions

1. Customers Table

Column Name	Data Type	Description
customer_id	INT (PK)	Unique ID for each customer
name	VARCHAR (100)	Customer's full name
phone	VARCHAR (12)	Contact number

Column Name	Data Type	Description
email	VARCHAR (100)	Email address
preferences	TEXT	Customer preferences
auth	TEXT	Authentication details

2. Employees Table

Column Name	Data Type	Description
employee_id	INT (PK)	Unique ID for each employee
name	VARCHAR (100)	Employee’s full name
role	VARCHAR (50)	Job position (e.g., Barber)
skills	TEXT	Special skills or expertise
availability	TEXT	Work schedule
resign_date	DATE	Resignation date (if any)
salary	DECIMAL	Salary of the employee
branch_id	INT (FK)	Branch where they work

3. Branches Table

Column Name	Data Type	Description
branch_id	INT (PK)	Unique ID for each branch
name	VARCHAR (100)	Branch name
location	VARCHAR (100)	Address of the branch

4. Services Table

Column Name	Data Type	Description
service_id	INT (PK)	Unique ID for each service
name	VARCHAR (100)	Service name (e.g., Haircut)
price	DECIMAL (10,2)	Cost of the service
duration	INT	Duration in minutes

5. Products Table

Column Name	Data Type	Description
product_id	INT (PK)	Unique ID for each product
name	VARCHAR(100)	Product name (e.g., Shampoo)
quantity	INT	Available stock
price	DECIMAL(10,2)	Product price
supplier_id	INT (FK)	Supplier providing the product

6. Suppliers Table

Column Name	Data Type	Description
supplier_id	INT (PK)	Unique ID for each supplier
name	VARCHAR(100)	Supplier name
contact_info	TEXT	Supplier contact details

7. Appointments Table

Column Name	Data Type	Description
appointment_id	INT (PK)	Unique ID for each appointment
customer_id	INT (FK)	Customer booking the service
employee_id	INT (FK)	Barber handling the appointment
service_id	INT (FK)	Service being availed
product_id	INT (FK)	Products used (if any)
date_time	DATETIME	Appointment date and time
status	VARCHAR (50)	Status (e.g., Completed, Pending)

8. Payments Table

Column Name	Data Type	Description
payment_id	INT (PK)	Unique ID for each payment
appointment_id	INT (FK)	Related appointment ID
amount	DECIMAL (10,2)	Total payment amount
payment_method	VARCHAR (50)	Payment mode (Cash/Card)
date	DATETIME	Payment date
created at	TIMESTAMP	Timestamp of transaction

9. Reviews Table

Column Name	Data Type	Description
review_id	INT (PK)	Unique ID for each review
appointment_id	INT (FK)	Related appointment ID
rating	INT	Customer rating (1-5)

Column Name	Data Type	Description
comments	TEXT	Customer feedback

10. Promotions Table

Column Name	Data Type	Description
promotion_id	INT (PK)	Unique ID for each promotion
name	VARCHAR (100)	Promotion name
discount_percentage	DECIMAL (5,2)	Discount applied
valid_from	DATE	Promotion start date
valid_to	DATE	Promotion end date

4. Implementation and Features

To enhance accessibility and usability, the following key features were implemented:

ER Diagram:

The Entity-Relationship Diagram visually represents the database structure. The key entities in the database include Customers, Employees, Services, Appointments, Products, and Payments. Primary keys uniquely identify each record, while foreign keys establish relationships between different tables. This structured approach ensures efficient data organization and retrieval.

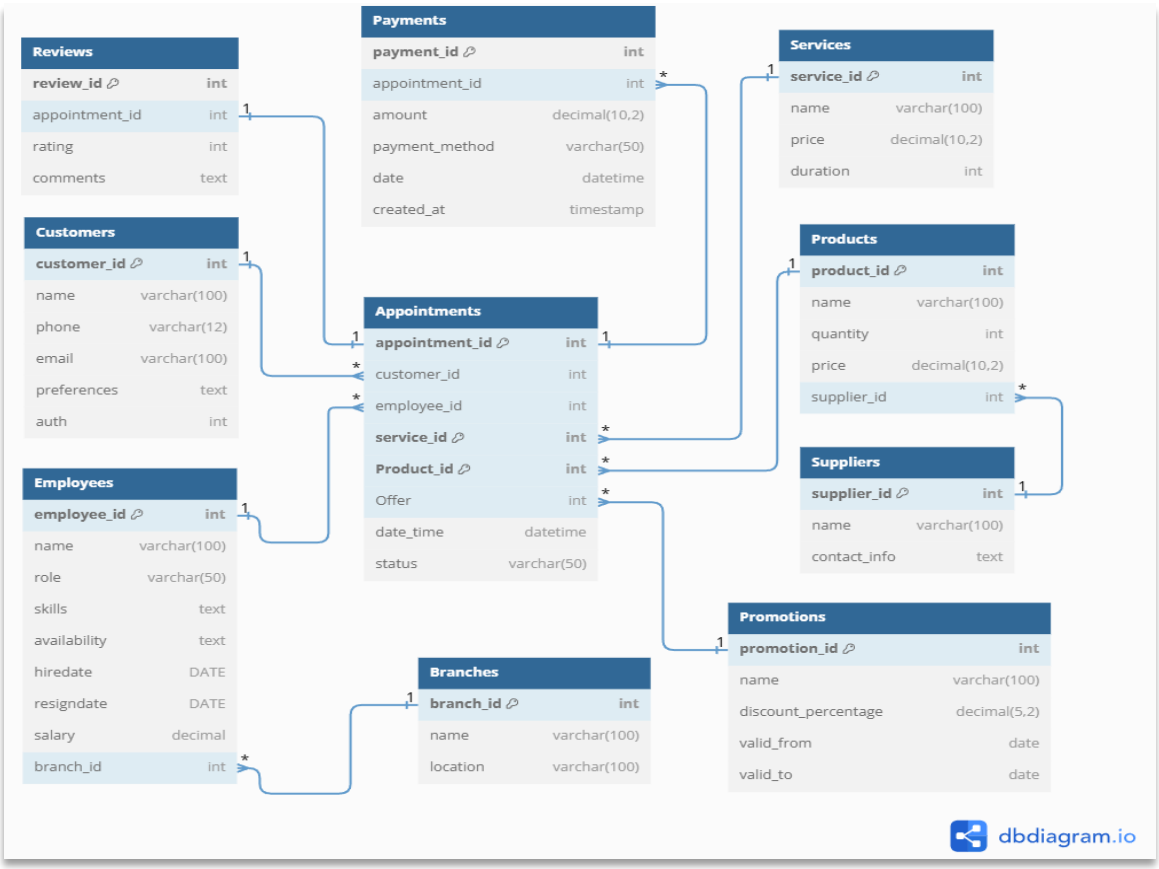


Table: Relationship Between Tables

Table Name	Related Table	Relationship Type	Foreign Key Description
Customers	Appointments	One-to-Many	customer_id → Appointments.customer_id (A customer can have multiple appointments.)
Employees	Appointments	One-to-Many	employee_id → Appointments.employee_id (An employee can be assigned multiple appointments.)
Branches	Employees	One-to-Many	branch_id → Employees.branch_id (A branch can have multiple employees.)
Services	Appointments	One-to-Many	service_id → Appointments.service_id (An appointment is associated with one service.)
Products	Appointments	Many-to-Many	product_id → Appointments.product_id (A product may be used in multiple appointments.)

Promotions	Appointments	One-to-Many	promotion_id → Appointments.offer (A promotion can be applied to multiple appointments.)
Products	Suppliers	Many-to-One	supplier_id → Suppliers.supplier_id (A product is supplied by a single supplier.)
Appointments	Payments	One-to-Many	appointment_id → Payments.appointment_id (An appointment can have multiple corresponding payments.)
Appointments	Reviews	One-to-One	appointment_id → Reviews.appointment_id (Each appointment may have one review.)

5. Views on Data Retrieval

Database Views

To make data retrieval easier, several database views were designed. Below are three key use cases that enhance operational efficiency:

View 1: Appointment Summary

- **Scenario:** Want to know how many appointments have been made for each service.
- **Tables Used:** Appointments, Services, Customers, Employees
- **Fields Used:** Appointment_id, customer_name, Employee_name, Service_name, appointment_date, preferences, appointment_status
- **Purpose:** To gather information for counting the total number of appointments for each service
- **View Name:** Appointment Summary

Creating view:

```sql
-- View for appointment details including customer, employee, and service
CREATE VIEW AppointmentDetails AS
SELECT
    a.appointment_id,
    c.name AS customer_name,
    e.name AS employee_name,
    s.name AS service_name,
    a.date_time AS appointment_date,
    a.preferences as preferences,
    a.status AS appointment_status
FROM
    Appointments a
LEFT JOIN
    Customers c ON a.customer_id = c.customer_id
LEFT JOIN
    Employees e ON a.employee_id = e.employee_id
LEFT JOIN
    Services s ON a.service_id = s.service_id;
```

Example Query

The following query retrieves information for a specific customer, filtering by appointment date:

```sql
-- Query to show appointment details for a specific customer
SELECT *
FROM AppointmentDetails
WHERE customer_name = 'John Doe' and  DATE(appointment_date) = "2025-01-28";
```

Output:

	appointment_id	customer_name	employee_name	service_name	appointment_date	preferences	appointment_status
1	1	John Doe	Alice Johnson	Haircut	2025-01-28 10:00:00	Quiet environment, natural light	Scheduled

View 2: Product Inventory

- **Scenario:** Want to track product inventory along with supplier details to monitor stock levels and supplier information.
- **Tables Used:** Products, Suppliers
- **Fields Used:** product_id, product_name, product_quantity, Product_price, supplier_name, supplier_contact
- **Purpose:** To gather inventory details along with supplier information for better stock and supplier management.
- **View Name:** Product Inventory

Creating the View:

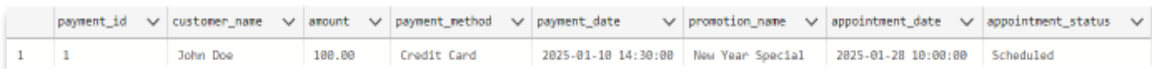
```sql
-- View for product inventory along with supplier information
CREATE VIEW ProductInventory AS
SELECT
    p.product_id,
    p.name AS product_name,
    p.quantity,
    p.price,
    s.name AS supplier_name,
    s.contact_info AS supplier_contact
FROM
    Products p
JOIN
    Suppliers s ON p.supplier_id = s.supplier_id;
```

Finding the Most Popular Product

The following query identifies the most popular product based on usage:

```sql
-- Query to show payment details for a specific customer
SELECT *
FROM PaymentDetails
WHERE customer_name = 'John Doe' and  DATE(appointment_date) = "2025-01-28";
```

Output:



	payment_id	customer_name	amount	payment_method	payment_date	promotion_name	appointment_date	appointment_status
1	1	John Doe	100.00	Credit Card	2025-01-10 14:30:00	New Year Special	2025-01-28 10:00:00	Scheduled

View 3: Payment Details

- **Scenario:** Want to track payment details along with associated promotions and appointment information to analyse revenue and discounts used.
- **Tables Used:** Payments, Appointments, Promotions, Customers
- **Fields Used:** payment id, customer name, amount, payment method, payment date, promotion name, appointment date, appointment status
- **Purpose:** To analyse payments, applied promotions, and appointment details for financial tracking and customer discount usage.
- **View Name:** Payment Details

Creating the view:

```sql
-- View for payment details with associated
-- promotions and appointment information
CREATE VIEW PaymentDetails AS
SELECT
    p.payment_id,
    c.name as customer_name,
    p.amount,
    p.payment_method,
    p.date AS payment_date,
    pr.name AS promotion_name,
    a.date_time AS appointment_date,
    a.status AS appointment_status
FROM
    Payments p
LEFT JOIN
    Appointments a ON p.appointment_id = a.appointment_id
LEFT JOIN
    Promotions pr ON a.offer = pr.promotion_id
RIGHT  JOIN Customers c ON a.customer_id = c.customer_id;
```

Printing receipt of specific customer

The following query show payment detail of specific customer:

```sql
-- Query to show payment details for a specific customer
SELECT *
FROM PaymentDetails
WHERE customer_name = 'John Doe' and  DATE(appointment_date) = "2025-01-28";
```

Output:

	payment_id	customer_name	amount	payment_method	payment_date	promotion_name	appointment_date	appointment_status
1	1	John Doe	100.00	Credit Card	2025-01-10 14:30:00	New Year Special	2025-01-28 10:00:00	Scheduled

6. Benefits of the Database System

The implementation of the Ikigai Barbershop database system offers several advantages:

Improved Efficiency

- Automated processes reduce manual work and errors.
- Appointments are easily scheduled and tracked.

Enhanced Data Accuracy
- Centralized data reduces inconsistencies.
- Customer preferences and histories are maintained.

Better Decision-Making
- Data-driven insights enable business growth.
- Inventory and revenue trends can be analyzed for improvements.

7. Future Enhancements
To further improve the system, the following enhancements are proposed:

Loyalty Programs
A points-based system to reward repeat customers and encourage loyalty.

AI-Powered Service Recommendations
Leveraging artificial intelligence to suggest services based on customer history and trends.

8. Conclusion
The Ikigai Barbershop database successfully integrates customer management, appointment scheduling, inventory tracking, and financial operations into a centralized system. By leveraging data-driven insights, the business can enhance service efficiency and customer satisfaction. With potential future enhancements such as AI-based recommendations and loyalty programs, this system sets a strong foundation for further technological advancements in barbershop management.