



IKIGAI



BARBERSHOP

Article on a Database Case Study by:

ABHAY SINGH THAKUR

0

[HTTPS://GITHUB.COM/ABHAYZAP/DATA-SCIENCE](https://github.com/abhayzap/data-science)

TABLE OF CONTENT

- Introduction
- Mission
- Objectives
- Identify Entities
- Database Design and Development
- Relationship between Table
- Implementation and Feature
- Views
- Key Benefits
- Future Enhancements
- Conclusion

Introduction

The IKIGAI Barbershop Database is designed to create an efficient and organized data management system for barbershops. It enables seamless tracking of appointments, inventory, financial transactions, and customer management. This system ensures smooth operations, reducing manual work while improving efficiency and accuracy.

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 – Ensures easy booking, reduces conflicts.
- Enhance Inventory Management – Tracks product usage and restocking needs.
- Automate Revenue & Expense Tracking – Helps monitor income and expenses in real time.

Identify Entities

The primary entities identified for the database are:

- Customers – Stores client information and preferences.
- Appointments – Manages scheduling and bookings.
- Employees – Keeps track of barbers and staff availability.
- Services – Catalogs various grooming options.
- Products – Maintains inventory details.
- Payments – Oversees financial transactions.

Database Design and Development

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
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

<u>Field name</u>	Attribute	Description
<u>appointment_id</u>	Int, Auto-Increment, Primary Key	Unique ID for each appointment
customer_id	int, Foreign key	Reference to the customer
employee_id	int, Foreign key	Reference to the employee
<u>product_id</u>	int , Primary Key	Reference to the product
<u>service_id</u>	int , Primary Key	Reference to the service
date_time	Datetime	Appointment date and time
status	varchar(50)	Status of the appointment (e.g., Completed, Scheduled)
preferences	Text	Customer's preferences
offer	int, Foreign key	Customer's discount , promotion_id from promotion table
amount	decimal(10,2)	Payment amount
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)
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

Relationship between Table:

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	Payment_mode	One-to-Many	appointment_id → Payment_mode.appointment_id	An appointment has many corresponding payment.
Appointments	Reviews	One-to-One	appointment_id → Reviews.appointment_id	Each appointment may have one review.

Implementation and Features

To make the database more user-friendly and efficient, several important features were implemented:

ER Diagram (Entity-Relationship Diagram)

The ER Diagram is a visual representation of how different parts of the database are connected. It helps in understanding the structure and relationships between various elements.

The main components of the database include:

- **Customers** – Stores information about clients and their preferences.
- **Employees** – Keeps track of barbers and staff details.
- **Services** – Lists all available grooming options.
- **Appointments** – Manages booking schedules for customers.
- **Products** – Maintains inventory records of grooming supplies.
- **Payments** – Handles financial transactions.

Each record in the database is uniquely identified using a **Primary Key**, ensuring that no two records are the same. **Foreign Keys** are used to link related tables, allowing data to be efficiently organized and retrieved. This structured approach improves the accuracy, consistency, and speed of data management.



Views:

VIEW 1 - APPOINTMENTS

Purpose

The purpose of this view is to analyze and count the number of appointments scheduled for each service. This information is essential for identifying popular services, tracking resource utilization, and making data-driven decisions.

Implementation

Creating the View

The SQL statement below demonstrates how the Appointment Summary view is created by joining relevant tables:

```
356 -- View for appointment details including customer, employee, and service
357 CREATE VIEW AppointmentDetails AS
358 SELECT
359     a.appointment_id,
360     c.name AS customer_name,
361     e.name AS employee_name,
362     s.name AS service_name,
363     a.date_time AS appointment_date,
364     a.preferences as preferences,
365     a.status AS appointment_status
366 FROM
367     Appointments a
368 LEFT JOIN
369     Customers c ON a.customer_id = c.customer_id
370 LEFT JOIN
371     Employees e ON a.employee_id = e.employee_id
372 LEFT JOIN
373     Services s ON a.service_id = s.service_id;
```

Example Query

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

```
410 -- Query to show appointment details for a specific customer
411 SELECT *
412 FROM AppointmentDetails
413 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

Purpose

This view's objective is to compile supplier and inventory facts. It helps companies to

- Track stock levels for improved inventory control
- Keep an eye on supplier performance and relationships.

Implementation

Creating the View

The SQL statement below demonstrates how the ProductInventory view is created:

```

375  -- View for product inventory along with supplier information
376  CREATE VIEW ProductInventory AS
377  SELECT
378      p.product_id,
379      p.name AS product_name,
380      p.quantity,
381      p.price,
382      s.name AS supplier_name,
383      s.contact_info AS supplier_contact
384  FROM
385      Products p
386  JOIN
387      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:

```

426  -- Query to show payment details for a specific customer
427  SELECT *
428  FROM PaymentDetails
429  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

Purpose:

To analyze payments, applied promotions, and appointment details for financial tracking and customer discount usage.

Creating the View

The SQL statement below demonstrates how the PaymentDetails view is created:

```
389  -- View for payment details with associated
390  -- promotions and appointment information
391  CREATE VIEW PaymentDetails AS
392  SELECT
393      p.payment_id,
394      c.name AS customer_name,
395      p.amount,
396      p.payment_method,
397      p.date AS payment_date,
398      pr.name AS promotion_name,
399      a.date_time AS appointment_date,
400      a.status AS appointment_status
401  FROM
402      Payments p
403  LEFT JOIN
404      Appointments a ON p.appointment_id = a.appointment_id
405  LEFT JOIN
406      Promotions pr ON a.offer = pr.promotion_id
407  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:

```

426 -- Query to show payment details for a specific customer
427 SELECT *
428 FROM PaymentDetails
429 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

Key Benefits:

- **Real-Time Access to Critical Information:** The integration of various tables and views ensures that managers and employees can retrieve essential business data instantly. This enhances decision-making, improves operational efficiency, and fosters a more responsive service environment.
- **Streamlined Workflow and Coordination:** With a centralized system, teams can seamlessly coordinate their tasks, reducing redundancies and improving communication across departments.
- **Data-Driven Insights for Better Performance:** By consolidating key business metrics, organizations can analyze trends, monitor performance, and optimize strategies to enhance productivity and customer satisfaction.

Future Enhancements:

- **Loyalty Programs for Customer Retention:** Implementing a structured loyalty program will help businesses reward repeat customers, increase engagement, and foster long-term relationships.
- **Advanced Customer Preference Tracking:** By leveraging data analytics, businesses can personalize services based on individual customer preferences, enhancing user experience and satisfaction.
- **AI-Powered Service Recommendations:** Integrating artificial intelligence will enable businesses to provide smart recommendations, anticipate customer needs, and offer tailored solutions, ultimately boosting sales and customer loyalty.

Conclusion:

In today's data-driven world, managing information effectively is key to success. SQL views help businesses organize and analyze complex data in a simple way, making decision-making easier. For example, the Product Inventory View helps maintain stock levels and choose reliable suppliers, while the Payment Details View provides insights into sales, promotions, and customer preferences. These views improve efficiency, security, and data accessibility. However, they must be well-designed to avoid performance issues. As technology advances, businesses that use structured data management will stay competitive and unlock new growth opportunities.