Cover

# acknowledgement

I would like to express my sincere gratitude to Mr. Abishek Bimali, our web development teacher, for his invaluable guidance, support, and encouragement throughout this project. His expert teaching, patience, and dedication to helping students understand both the fundamentals and the advanced aspects of web technologies have played a vital role in shaping my knowledge and skills in this field.

Thank you, sir, for being a source of inspiration and for constantly motivating me to strive for excellence.

# Executive Summary

Table of Contents

[acknowledgement 2](#_Toc203345821)

[introduction 5](#_Toc203345822)

[1. Database Design 6](#_Toc203345823)

[1.1 Entity Relationship Diagram (ERd) 6](#_Toc203345824)

[1.2 Database Normalization 7](#_Toc203345825)

[2.Database Queries 8](#_Toc203345826)

[2.1 Createing all normalized tables in SQL. 8](#_Toc203345827)

[2.2 Entering data for all tables 9](#_Toc203345828)

[2.3 query to get all customer who haven’t paid 10](#_Toc203345829)

[2.4 Books available in the "Fiction" genre in descending order along with their number of copies 11](#_Toc203345830)

[2.5 Query to count the number of books available in the "Mystery" genre. 12](#_Toc203345831)

[2.6 Query to retrieve all customers who have made reservations for events named as Book Reading 13](#_Toc203345832)

[2.7 Update the book genre of any uploaded book from "Classic Literature" to "Modern Classics”. 13](#_Toc203345833)

[2.8  Query to find the top 3 customers who have spent the highest total amount onorders 14](#_Toc203345834)

[2.9 query to retrieve all orders placed in the last 7 days, including the customer’s full name and order total. 15](#_Toc203345835)

[3. critical reflection 16](#_Toc203345836)

[Reflecting on Requirements and Database Technology Enhancements 16](#_Toc203345837)

[Refrences 19](#_Toc203345838)

# Table Of Figures

[Figure 1:Introduction 6](#_Toc203862916)

[Figure 2: ERD OF CAFE 7](#_Toc203862917)

[Figure 3:Creating Table 9](#_Toc203862918)

[Figure 4:Inserting Data 10](#_Toc203862919)

[Figure 5:who\_havent\_paid 11](#_Toc203862920)

[Figure 6:Result\_who\_haven't\_paid 11](#_Toc203862921)

[Figure 7:BooksOnFiction 12](#_Toc203862922)

[Figure 8:ResulfOfFictionBooks 12](#_Toc203862923)

[Figure 9:AvailableBooksOnMystery 13](#_Toc203862924)

[Figure 10:ResultOfBooksAvailableOnMysteryGenre 13](#_Toc203862925)

[Figure 11:ReservationASBookReading 14](#_Toc203862926)

[Figure 12:ResultReservationAsBookReading 14](#_Toc203862927)

[Figure 13:Update 14](#_Toc203862928)

[Figure 14:ResultUpdate 14](#_Toc203862929)

[Figure 15:TopSpent 15](#_Toc203862930)

[Figure 16:ResultOfTopSpent 15](#_Toc203862931)

[Figure 17:OrderPlacedInLast7Days 16](#_Toc203862932)

[Figure 18:ResultOrderPlacedInLast7Days 16](#_Toc203862933)

[Figure 19:Idea 17](#_Toc203862934)

[Figure 20:Normalization 17](#_Toc203862935)

[Figure 21:BenefitsOfCloudDatabase 18](#_Toc203862936)

[Figure 22: Flexibility in Data Management 18](#_Toc203862937)

[Figure 23:IntegratingMobileApp 19](#_Toc203862938)

# introduction

Cozy Book Cafe is a warm venue that mixes book affection and coffee comfort. Clients come in casual or become members and enjoy the benefits of book access, book and reading reservations, membership in programs, and order-in meal service. For them to work smoothly, there should be an efficiently managed database system to manage them effectively.

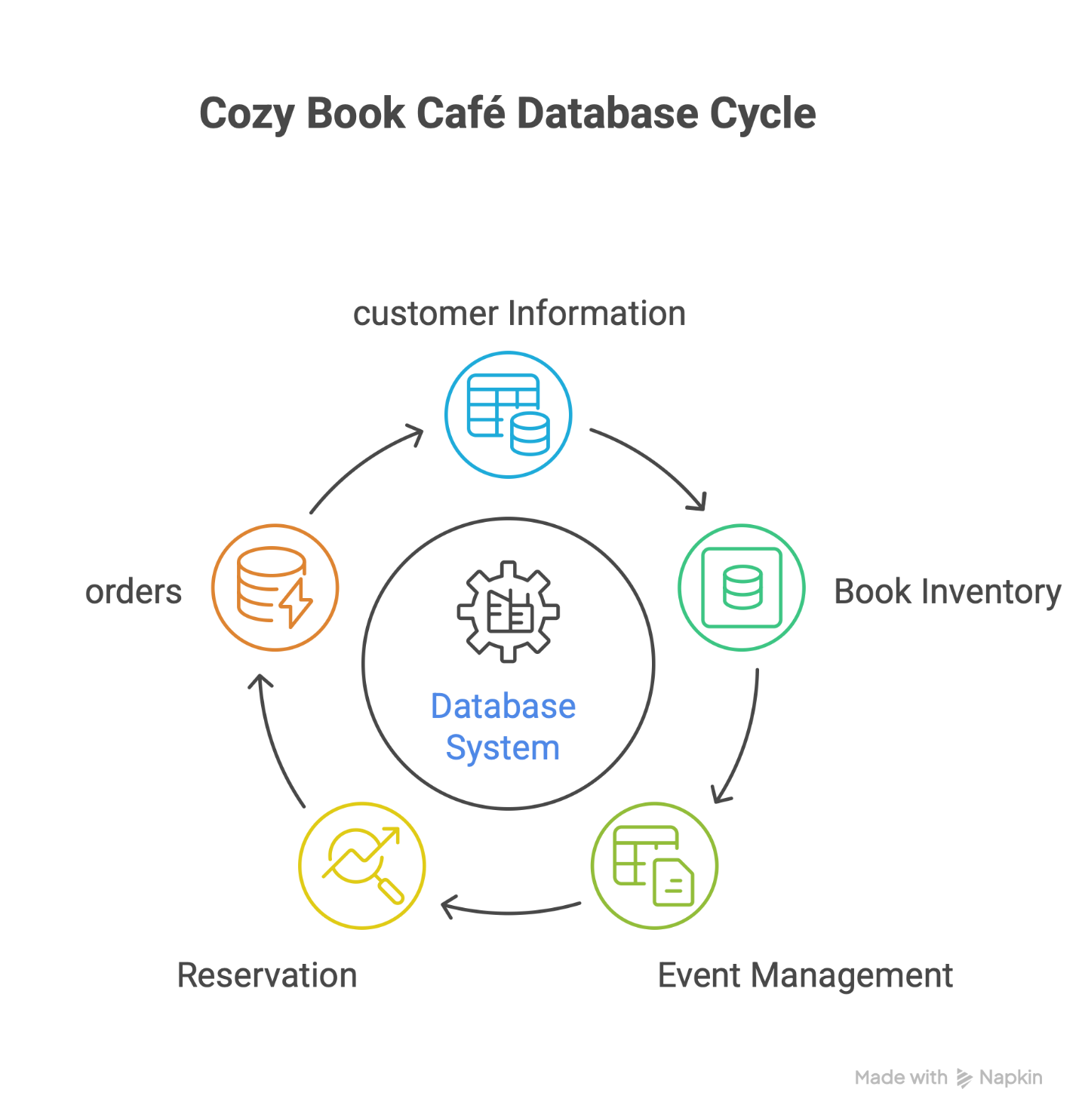


Figure 1:Introduction

The report shows the design and implementation of a relational database to manage key operations such as customer handling, book stock, reservations, function bookings, and payment handling. The report includes an ERD, SQL scripts, example data, queries, and a brief reflection. The report additionally mentions how existing database technologies such as NoSQL and cloud infrastructure can further improve the operations of the cafe.

# 1. Database Design

## 1.1 Entity Relationship Diagram (ERd)

Entity-Relationship Diagram (ERD) of Cozy Book Cafe seeks to organize and manage essential information related to customers, books, events, reservations, orders, and payments. The Customer table keeps basic information like customer ID, name, email, phone, and membership status. The Books table accommodates each book's ID, book title, author, category, and number of copies available. The Events table includes event ID, name, date, time, and entry fee. The Reservations table connects customers to planned events they book, and the status of the booking. The Orders table details food and beverage orders, and cost amount, which sums up the cost order.

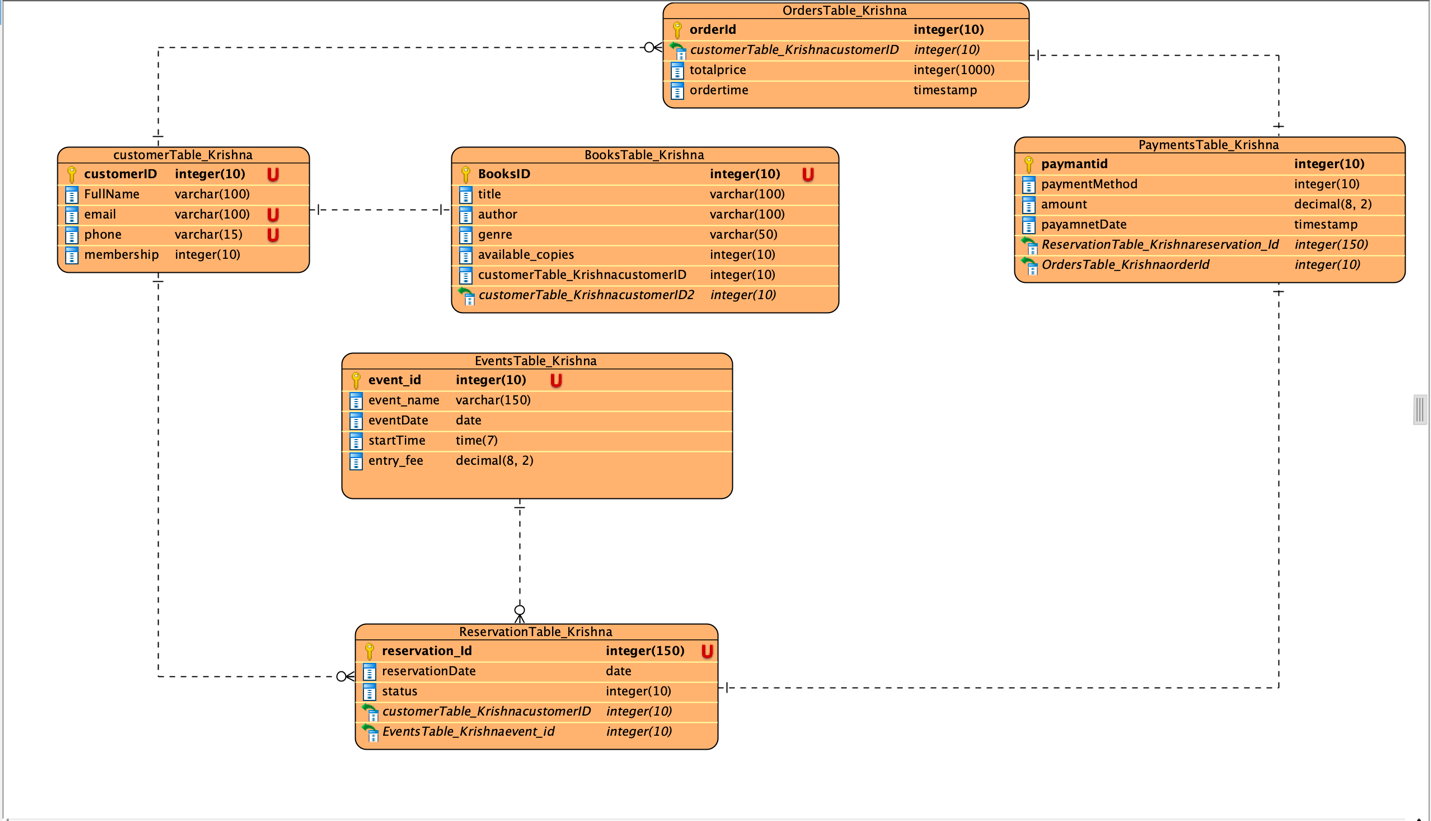


Figure 2: ERD OF CAFE

The Payments table details payment information like payment method, amount, and date, either an order or an event payment for its reservation. These tables interconnect such that they connect customers to their orders, reservations, and payment, and events to reservations and payment, too. Overall, the ERD provides an understandable and systematic plan that helps the coffee house smoothly operate and keep its information accurate and uniform.

## 1.2 Database Normalization

The given data structure benefits from normalization to reduce redundancy and improve data integrity. Here's how normalization principles are applied:

* **First Normal Form (1NF):** All tables have a primary key, and each column contains atomic values. This is already satisfied in the provided structure.
* **Second Normal Form (2NF):** All non-key attributes are fully functionally dependent on the primary key. This is also satisfied. For example, in the Orders table, total\_price is dependent on order\_id and customer\_id.
* **Third Normal Form (3NF):** All non-key attributes are non-transitively dependent on the primary key. This is also satisfied. There are no non-key attributes that depend on other non-key attributes.

Normalization improves the database structure by:

* **Reducing Data Redundancy:** Information about customers is stored only once in the Customers table, avoiding duplication across multiple tables.
* **Improving Data Integrity:** Changes to customer information only need to be made in one place, ensuring consistency.
* **Simplifying Queries:** Normalized tables make it easier to write queries that retrieve related data from multiple tables.
* **Enhancing Data Modification:** Updates and deletions are simplified, reducing the risk of inconsistencies.

# 2.Database Queries

## 2.1 Createing all normalized tables in SQL.

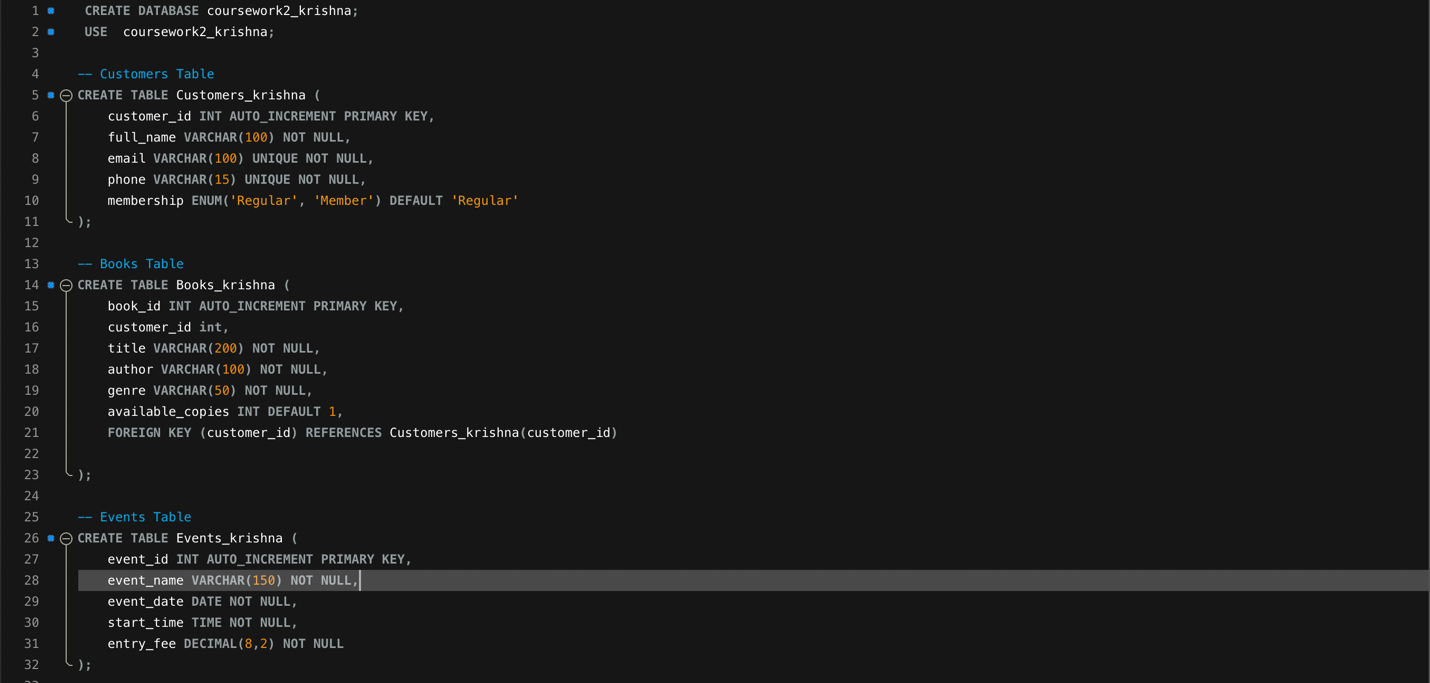
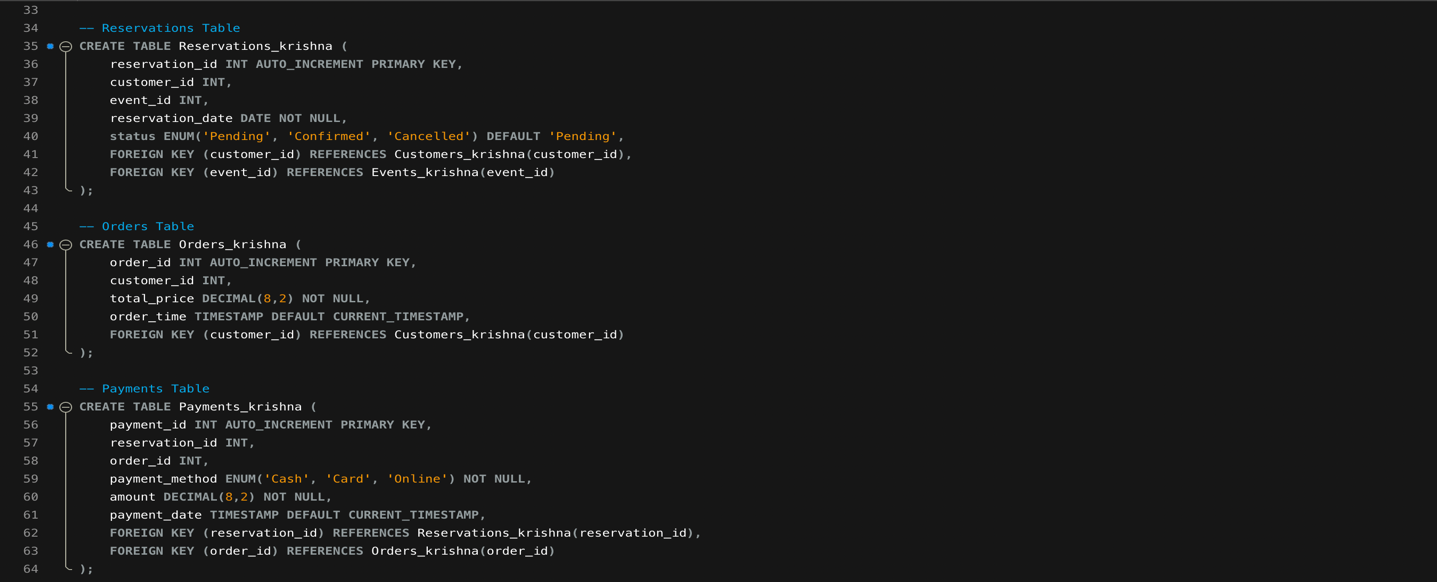


Figure 3:Creating Table

This SQL script creates the core tables for the Cozy Book Cafe's database system. It begins with the Customers\_krishnatable, storing basic customer details including membership type. The Books\_krishna table holds information about available books and it also includes a foreign key referencing the customer, though this relationship may not be necessary unless books are being linked to specific users. The Events\_krishna table defines scheduled events with their names, dates, times, and entry fees. Reservations\_krishna links customers to events they have booked, tracking the reservation status. The Orders\_krishna table records food and drink orders placed by customers along with timestamps. Lastly, the Payments\_krishna table connects both reservations and orders to their corresponding payments, including method, amount, and payment date. Altogether, these tables form a normalized structure for effectively managing the cafe's operations.

## 2.2 Entering data for all tables

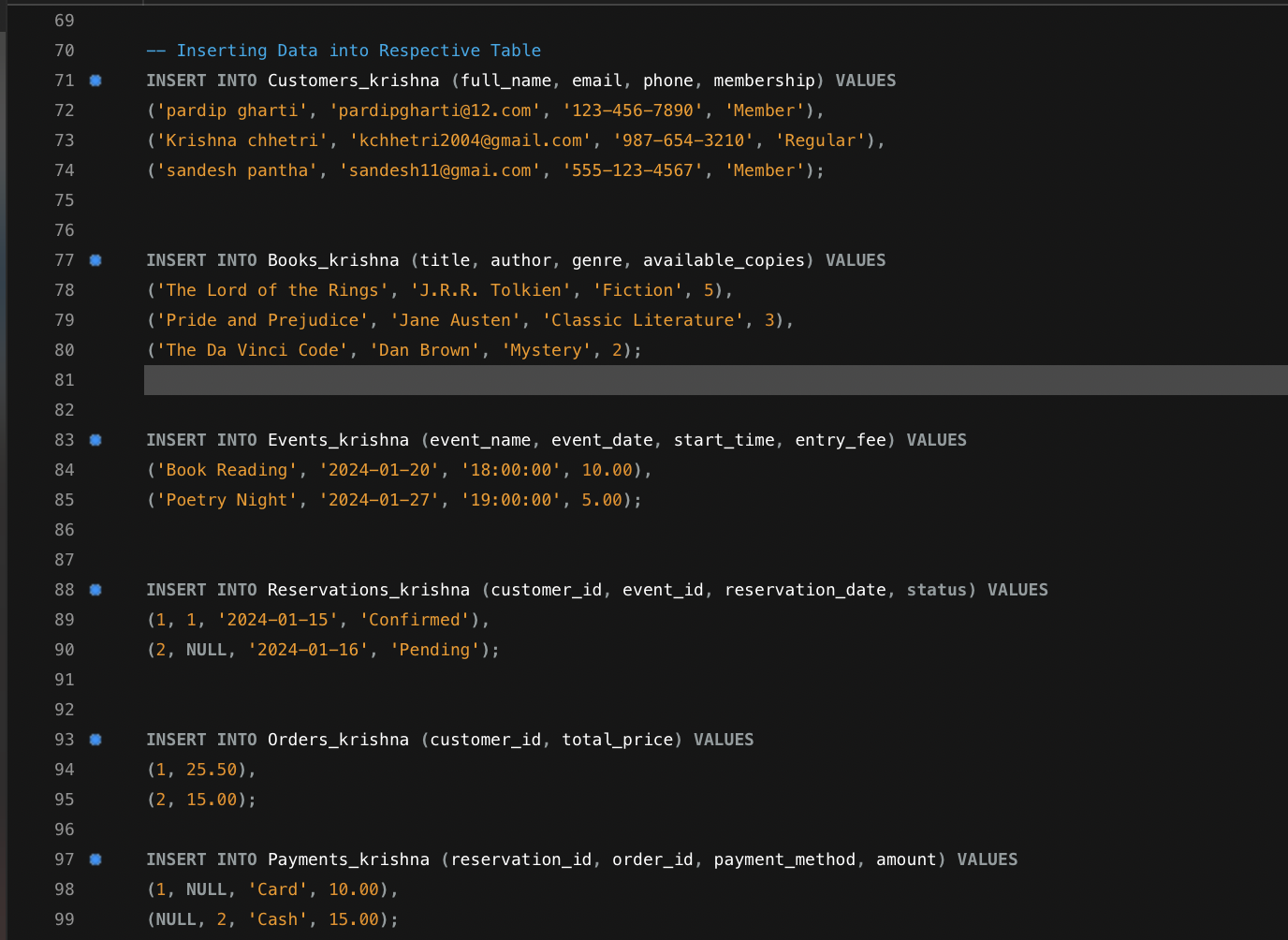


Figure 4:Inserting Data

This SQL script inserts sample data into the Cozy Book Cafe database, populating all key tables to simulate real-world operations. The Customers\_krishna table is filled with three customer records, each with a unique name, email, phone number, and membership type. The Books\_krishna table adds four books from various genres, showing their availability. Events such as "Book Reading" and "Poetry Night" are inserted into the Events\_krishna table with scheduled dates, times, and entry fees. The Reservations\_krishna table links customers to events, including one confirmed and one pending reservation. The Orders\_krishna table records food or drink purchases by customers along with the total price. Finally, the Payments\_krishna table logs payments, connecting them either to reservations or orders, and specifies the method of payment. This dataset provides a functional base for running and testing queries within the system.

## 2.3 query to get all customer who haven’t paid

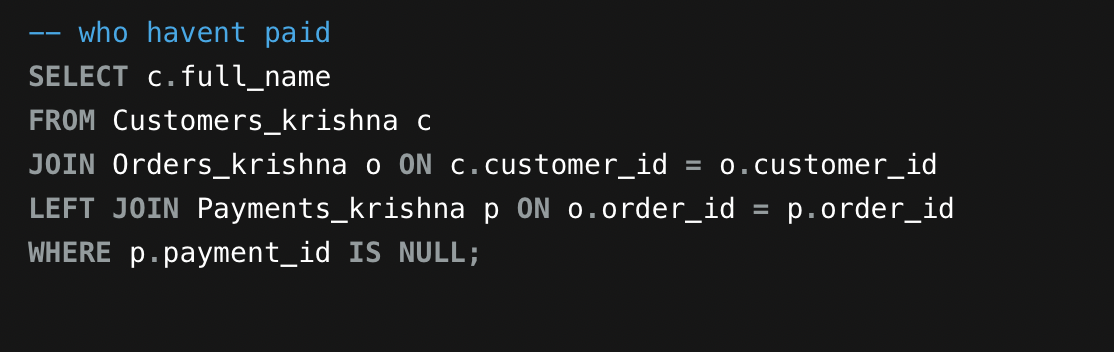


Figure 5:who\_havent\_paid

The SQL query retrieves the customer names that have placed orders but have yet to pay. It joins the `Customers\_krishna` and `Orders\_krishna` tables to select customers that have placed orders, and it performs an inner or left join with `Payments\_krishna` to select all the orders, including unpaid orders. The `WHERE` statement removes the ones that have payments, and it displays the customers that have orders with no corresponding records in the payments table.

A black and white striped background

AI-generated content may be incorrect.

Figure 6:Result\_who\_haven't\_paid

## 2.4 Books available in the "Fiction" genre in descending order along with their number of copies

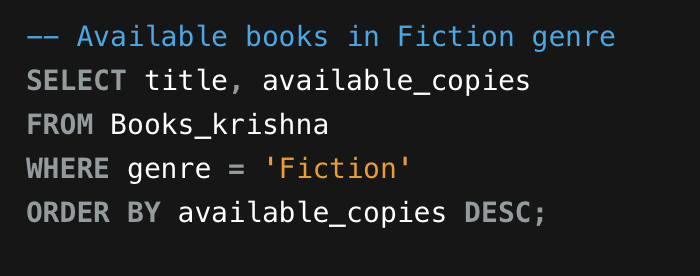


Figure 7:BooksOnFiction

This SQL query retrieves the book titles and available book copies from the `Books\_krishna` table which are of the "Fiction" genre. It uses the `WHERE` clause to limit the results and the `ORDER BY` clause to list the results in descending order based on available book copies, displaying the books with the most available copies first.

A black and white line

AI-generated content may be incorrect.

Figure 8:ResulfOfFictionBooks

## 2.5 Query to count the number of books available in the "Mystery" genre.

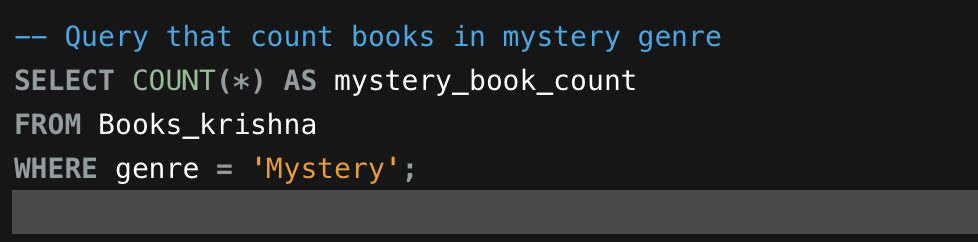


Figure 9:AvailableBooksOnMystery

This query counts how many books belong to the "Mystery" genre in the Books\_krishna table using the COUNT(\*)function with a WHERE filter.

A black and white striped background

AI-generated content may be incorrect.

Figure 10:ResultOfBooksAvailableOnMysteryGenre

## 2.6 Query to retrieve all customers who have made reservations for events named as Book Reading

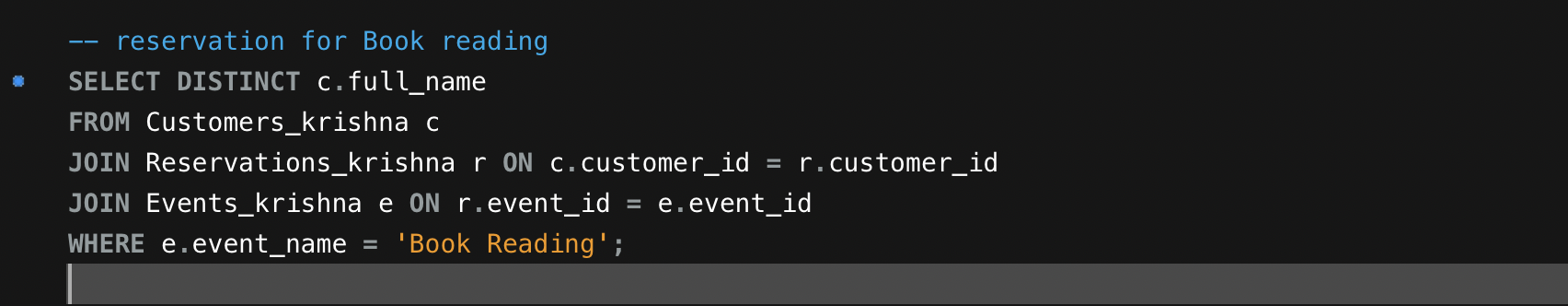


Figure 11:ReservationASBookReading

This query retrieves the names of customers who reserved the event titled as "Book Reading" by joining the Customers\_krishna, Reservations\_krishna, and Events\_krishna tables and filtering by the event name.

A black rectangular object with white text

AI-generated content may be incorrect.

Figure 12:ResultReservationAsBookReading

## 2.7 Update the book genre of any uploaded book from "Classic Literature" to "Modern Classics”.

A black screen with yellow text

AI-generated content may be incorrect.

Figure 13:Update

This query temporarily disables safe update mode to change the genre from "Classic Literature" to "Modern Classics" in the Books\_krishna table, then re-enables safe mode.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14:ResultUpdate

## 2.8  Query to find the top 3 customers who have spent the highest total amount onorders

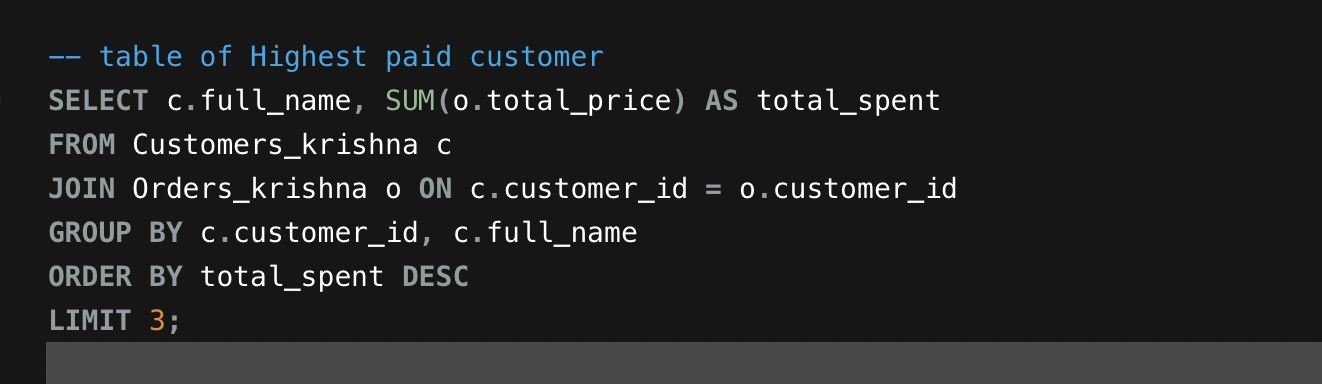


Figure :TopSpent

This query joins customers with their orders, sums the total spending per customer, sorts them by amount spent in descending order, and returns the top 3 spenders.

A black and white line

AI-generated content may be incorrect.

Figure 16:ResultOfTopSpent

## 2.9 query to retrieve all orders placed in the last 7 days, including the customer’s full name and order total.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 17:OrderPlacedInLast7Days

This query fetches all orders from the last 7 days by comparing order\_time with the current date using NOW() - INTERVAL 7 DAY, and joins it with customer details to show their names and order totals.

A screenshot of a black and white screen

AI-generated content may be incorrect.

Figure 18:ResultOrderPlacedInLast7Days

# 3. critical reflection

## Reflecting on Requirements & Database Technology Enhancements

Designing the database for the Cozy Book Cafe wasn't just a technical challenge; it was also about creating the digital core of livable, cozy space where coffee meets stories. A relational database was created to create a cozy, interactive space where coffee and stories are seamlessly integrated into everyday life. Creating a relational database helps to keep the cafe's charm intact, by managing reading sessions, booking events, and tracking every cappuccino and copy of Pride and Prejudice.

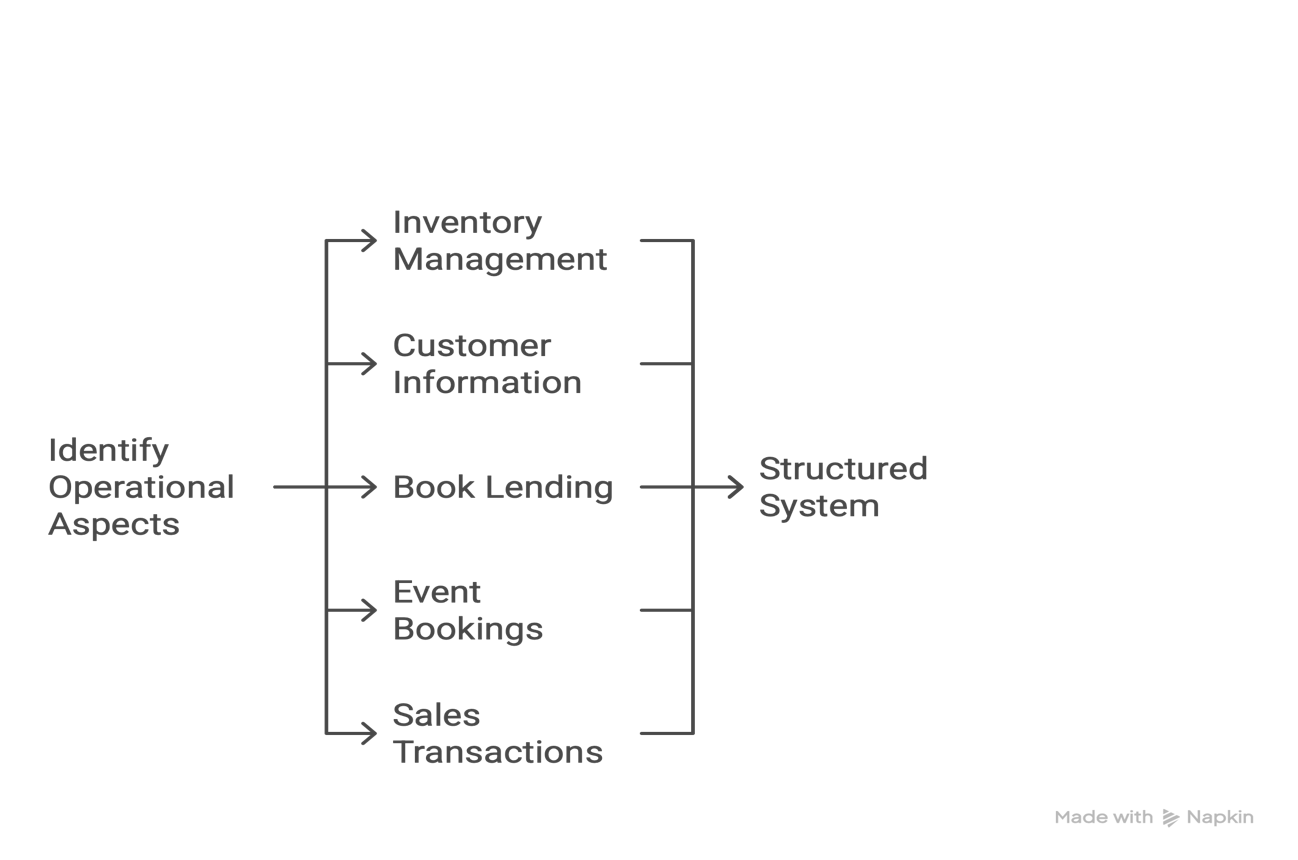


Figure 19:Idea

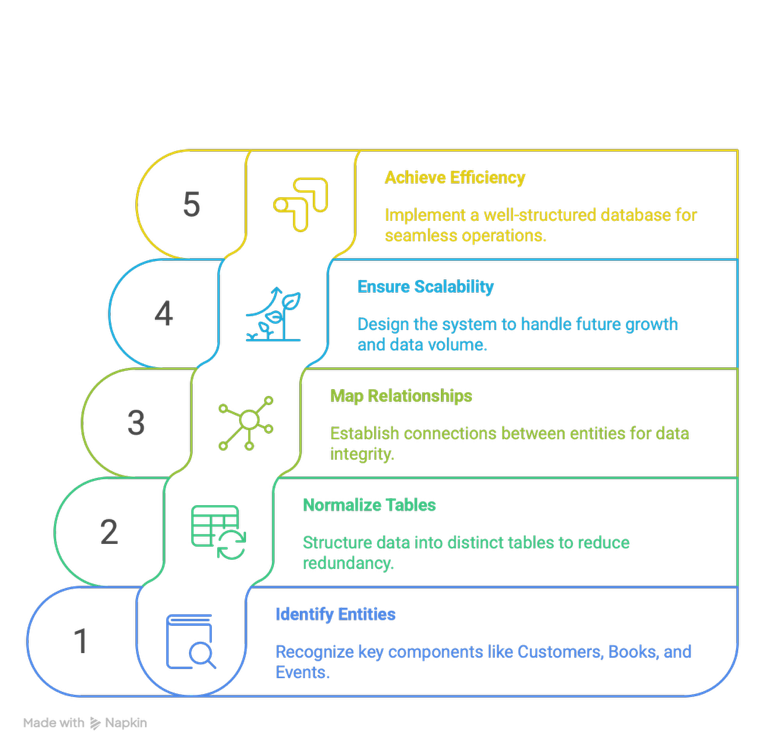
The system's tables were arranged in an orderly manner: Customers, Books and Events are the key words used to normalize this function. The arrangement prevents duplicate data, enhanced its security, and facilitates efficient functioning.

Figure :Normalization

For example, When a member book tickets to stumbling upon the library and orders ice cream, it ensures that their interactions are recorded in real time. For instance.

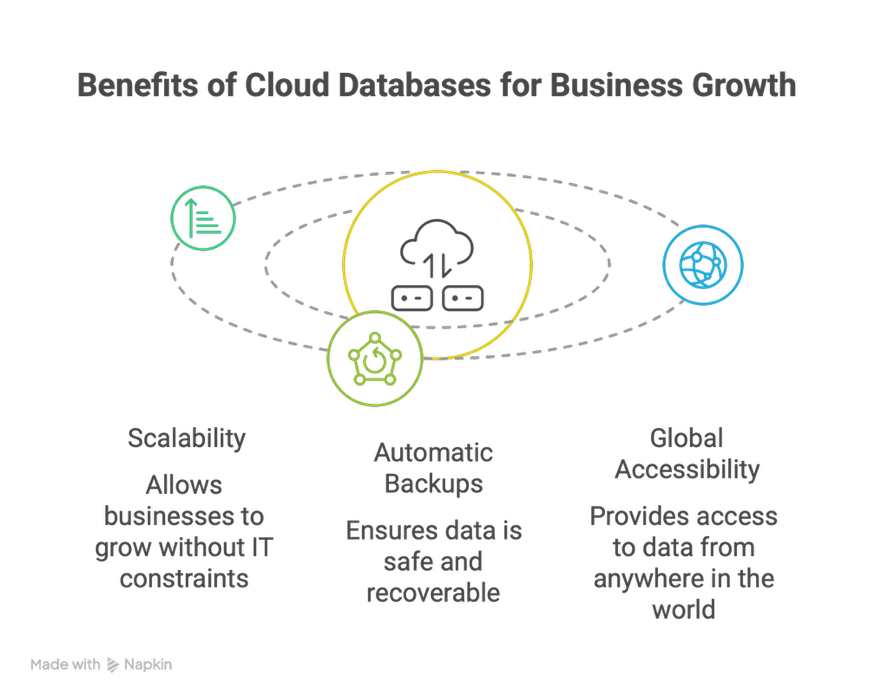


Figure 21:BenefitsOfCloudDatabase

Take advantage of cloud databases such as Amazon RDS or Google Cloud SQL. These platforms provide more than just storage; they also offer flexibility, automatic backups, and global accessibility (Kleppmann, 2017).

**Data can be stored without any schema in NoSQL databases like MongoDB. Granitebraker, 2010 highlights the ease of storing real-time customer feedback during events or on a digital bookshelf with user reviews, which can be achieved through NoSQL's processing of data.**

A screenshot of a device

AI-generated content may be incorrect.

Figure 22: Flexibility in Data Management

It complements the current relational setup, creating a hybrid system that’s both rigid where needed and flexible where possible.

Additionally, by coordination this database with versatile apps or web dashboards utilizing APIs, the cafe can modernize client encounters letting clients save seats, see occasion calendars, or pre-order coffee from their phones. Backend frameworks like Firebase or cloud-hosted GraphQL can bring this to life (Padhy et al., 2011).

A diagram of a customer experience

AI-generated content may be incorrect.

Figure 23:IntegratingMobileApp

In pith, the social database lays the foundational but blending it with cutting edge cloud and NoSQL arrangements changes it into a computerized environment. This advancement bolsters not as it were productive administration but too opens data-driven decision-making, personalized client encounters, and future development.

# Refrences