Chuks Azubuike

Alisa Crowe

Rita Herfi

Samantha Keppler

Online Bookstore Database Implementation

Introduction

For the term project, we decided to visualize and implement an online bookstore. Our online bookstore database is a comprehensive management system, designed to handle the demands and complexities associated with operating a virtual bookstore. This includes tracking book sales and returns (i.e., customer transactions), inventory management, customer reviews, and maintaining records of supplier, publisher, and author information. The database consists of 12 different entities, namely: Book, Author, Inventory, InventorySupplier, Publisher, Supplier, OrderItem, Payment, Order, Customer, Return and Review.

Additionally, the online bookstore database can be utilized by various businesses who are either actively operating in the digital space, or seeking to venture into e-commerce. This includes self-publishing authors, physical/online retail bookstores, educational institutions, and online learning platforms such as Coursera or Udemy. Thus, our database system serves as a flexible and versatile solution for these businesses.

Business Rules

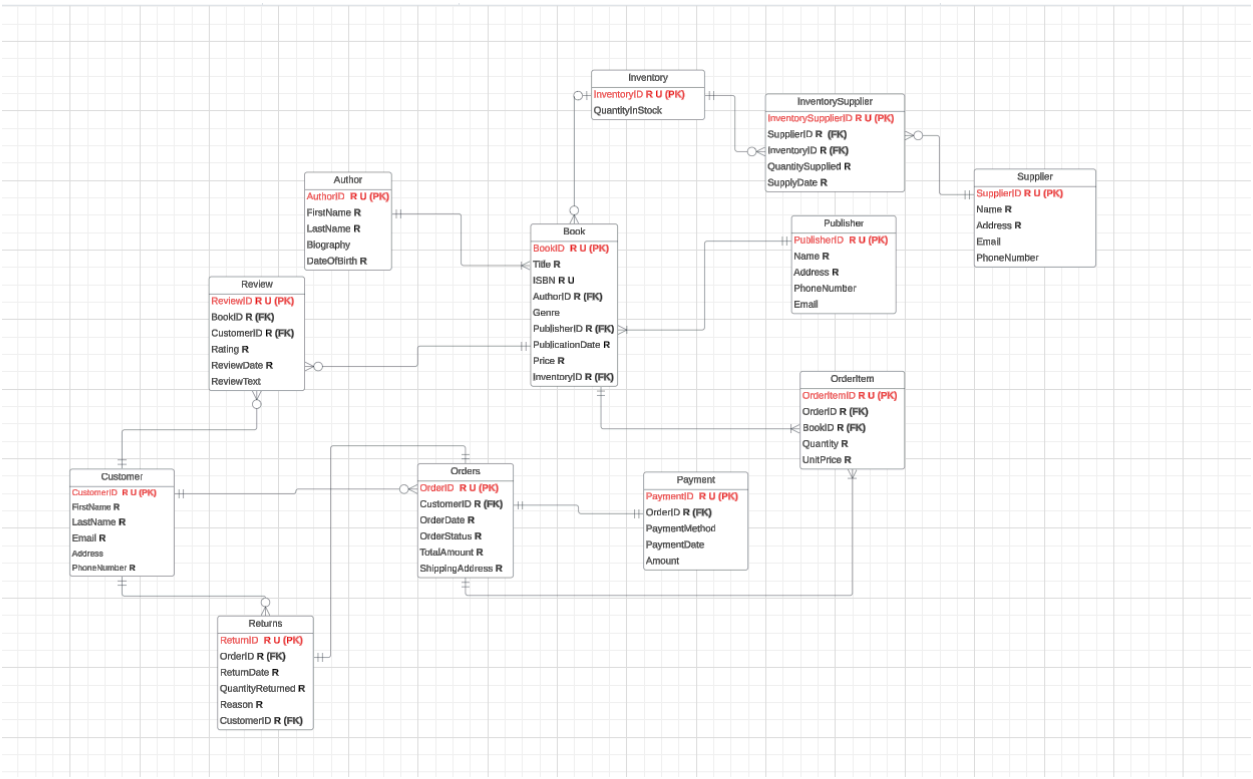
Below are the business rules that govern the inner workings and structure of our database. They have been implemented to ensure consistency across entity relationships and are based on real-world business practices.

- An Author can write one or more Books and each Book is written by exactly one Author.

- A Supplier can supply one or more Books and a Book can be supplied by one or more Suppliers.
- Inventory can store one or more Books and each Book is stored in either zero or one Inventory.
- A registered Customer can place zero, one, or multiple orders but each Order has exactly one Customer associated with it.
- Each Payment is associated with one Order, and an Order is completed in exactly one Payment (no installments, and payment method can either be debit, credit, or paypal).
- Each Payment must be worth more than $0, and all Payments must be in USD.
- Each Order can contain one or more OrderItems but each OrderItem is associated with exactly one Order.
- Each Book can represent multiple OrderItems but each OrderItem is associated with one Book.
- An Order must consist of at least one OrderItem, and each OrderItem must cost more than $0.
- A Publisher can publish one or more Books but a Book is published by exactly one Publisher.
- A Review has to have at least a rating or a written comment. It can have both, but not neither. Each rating must be between 0 and 5 stars.
- Each Return must have at least one Book in it, or each Return is associated with exactly one Order.
- A Customer can make zero or multiple Returns, but each Return is associated with exactly one Customer.

- An Order can have different statuses (pending, shipped, delivered, canceled), but only one status at a time.
- A Customer can return a Book within one week of their purchase.
- Customers will get a two percent discount on an Order if the total amount is greater than $100.

ERD/ Relational Model



The ERD diagram shown here depicts the relationship between different entities in our online database system. Since books are the main commodity for this online business, we have positioned the Book table in the center of the diagram to emphasize its role as the primary product and then show how other entities relate to it, either directly or indirectly. For example, a direct one-to-many relationship exists between the Book and Author tables, showing that each author whose information is in our database has authored at least one book, or has written multiple books.

Moreover, Publisher and Book also have a similar relationship, whereby a publisher publishes at least one book, and at most multiple books. The Inventory table on the other hand, shares a one-to-many relationship with the Book table, however, in this case inventory can contain either zero or multiple books.This relationship was implemented to signify a real-life situation where the inventory is empty or does not contain a particular book. Other indirect relationships in the diagram include a one-to-many relationship between the Customer and Review tables, Customer and Returns, Inventory and InventorySupplier, InventorySupplier and Supplier, and lastly, a one-to-one relationship between Orders and Payment tables, as well as the Orders and Returns tables.

Database Implementation

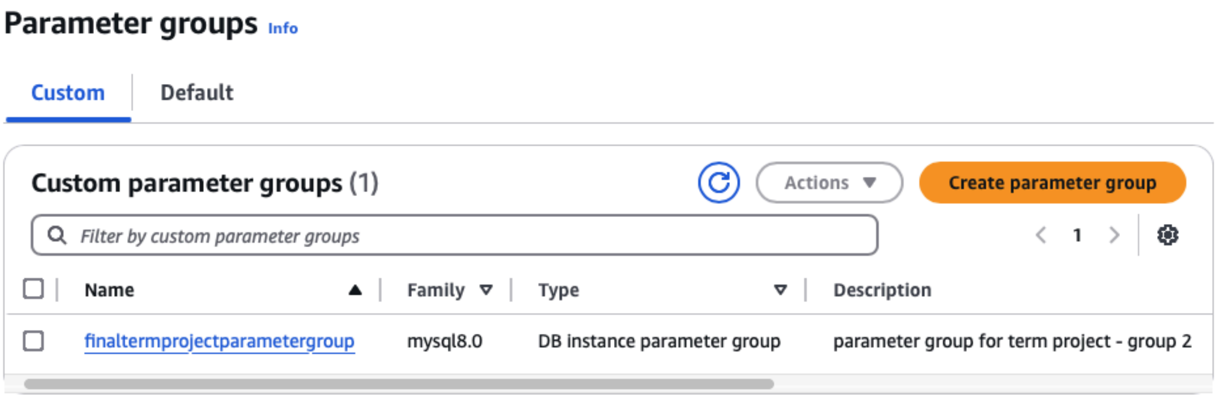
This section illustrates our SQL code, and the process of translating our database model into a working database system on MySQL. Included in the code is the insertion of 25 rows of dummy data, which was generated using Microsoft Copilot.

SQL Code

In the code, we also implemented several indexes, views, triggers, and stored procedures to make our database more efficient and user-friendly. We created indexes for larger tables and tables we assumed would be queried frequently by a database user. The purpose of the indexes were to improve the speed and efficiency of query performances, join operations, and quick searches. As for views, we created hybrid views of tables that we felt would often be viewed together, such as Customer and Orders. The views eliminate the need to write join queries each time you want to access related data in multiple tables. Then, we created Procedures for functions that the database would have to perform frequently, such as adding new orders into the Orders table. These procedures can be called to efficiently update and add on to tables with new information. Finally, we created triggers to match with automatically enforce our business rules when a particular event occurs. For example, the ApplyDiscount trigger automatically enforces a 2% discount when an order costs more than $100.

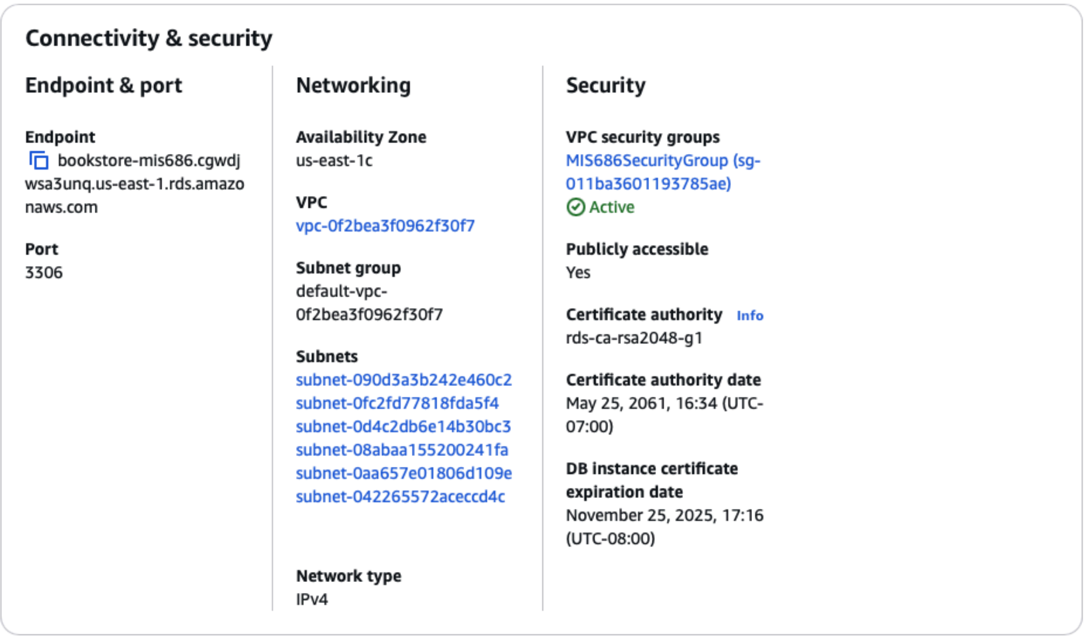
Database Deployment

We deployed our database using Amazon AWS RDS (Relational Database Service). We first created a parameter group named “finalprojectparametergroup” using MySQL 8.0, as shown in the screenshot below. This parameter group was configured to customize the database settings according to the requirements of our project.



☐	Name	Family	Type	Description
☐	finaltermprojectparametergroup	mysql8.0	DB instance parameter group	parameter group for term project - group 2

Next, we created an empty database titled “bookstore-mis686” using the MySQL 8.0 engine. The configuration settings are provided in the screenshot below. For the user “admin”, we used the password “passwordMIS686” to connect to the database. Public access was enabled, and we created a new security group called “MIS686SecurityGroup”. To allow connections from any location, we modified the inbound rules of this security group to permit connections from all IP addresses.



Connectivity & security

Endpoint & port

Endpoint
bookstore-mis686.cgwdjwsa3unq.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1c

VPC
vpc-0f2bea3f0962f30f7

Subnet group
default-vpc-0f2bea3f0962f30f7

Subnets
subnet-090d3a3b242e460c2
subnet-0fc2fd77818fda5f4
subnet-0d4c2db6e14b30bc3
subnet-08abaa155200241fa
subnet-0aa657e01806d109e
subnet-042265572aceccd4c

Network type
IPv4

Security

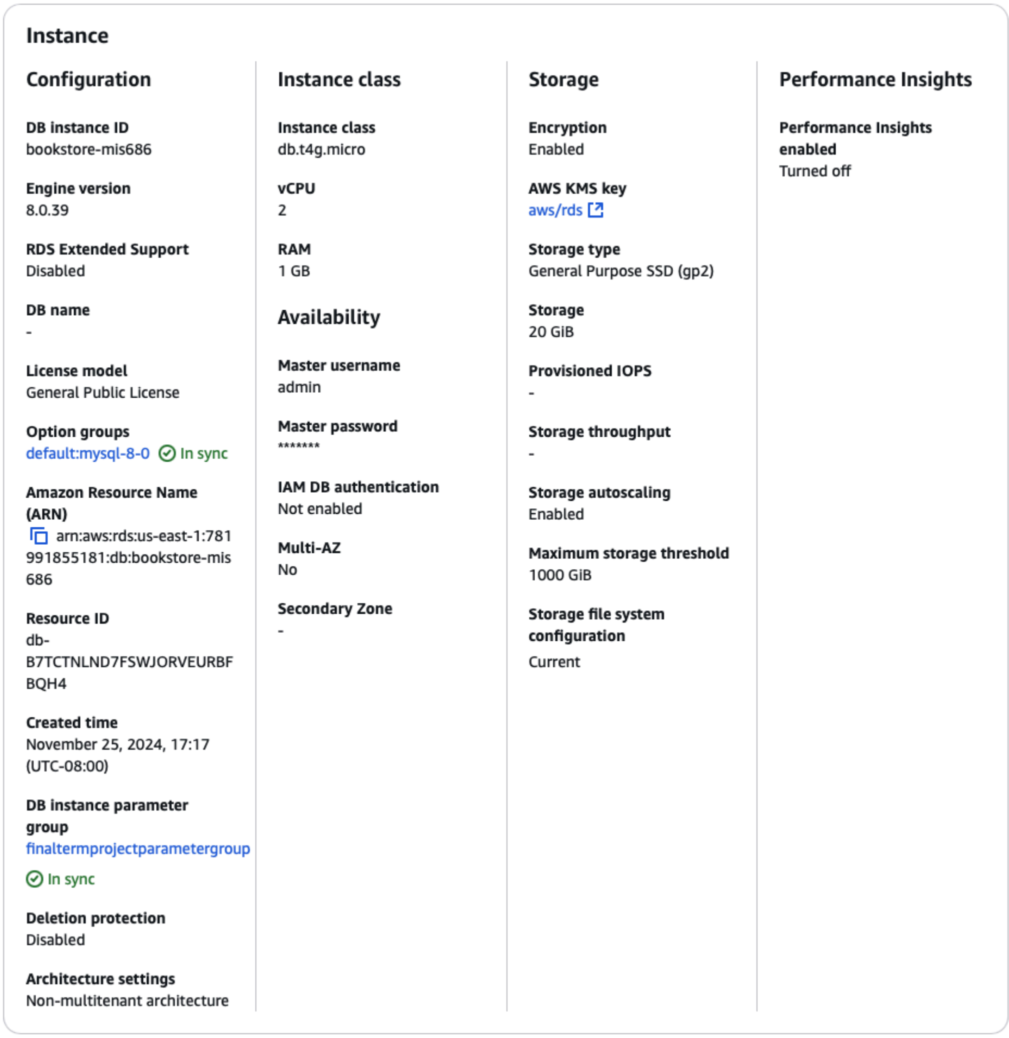
VPC security groups
MIS686SecurityGroup (sg-011ba3601193785ae)
✓ Active

Publicly accessible
Yes

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 25, 2061, 16:34 (UTC-07:00)

DB instance certificate expiration date
November 25, 2025, 17:16 (UTC-08:00)



Instance

Configuration

DB instance ID
bookstore-mis686

Engine version
8.0.39

RDS Extended Support
Disabled

DB name
-

License model
General Public License

Option groups
default:mysql-8-0 ✓ In sync

Amazon Resource Name (ARN)
arn:aws:rds:us-east-1:781991855181:db:bookstore-mis686

Resource ID
db-B7TCTNLND7FSWJORVEURBFBQH4

Created time
November 25, 2024, 17:17 (UTC-08:00)

DB instance parameter group
finaltermprojectparametergroup
✓ In sync

Deletion protection
Disabled

Architecture settings
Non-multitenant architecture

Instance class

Instance class
db.t4g.micro

vCPU
2

RAM
1 GB

Availability

Master username
admin

Master password
****

IAM DB authentication
Not enabled

Multi-AZ
No

Secondary Zone
-

Storage

Encryption
Enabled

AWS KMS key
aws/rds

Storage type
General Purpose SSD (gp2)

Storage
20 GiB

Provisioned IOPS
-

Storage throughput
-

Storage autoscaling
Enabled

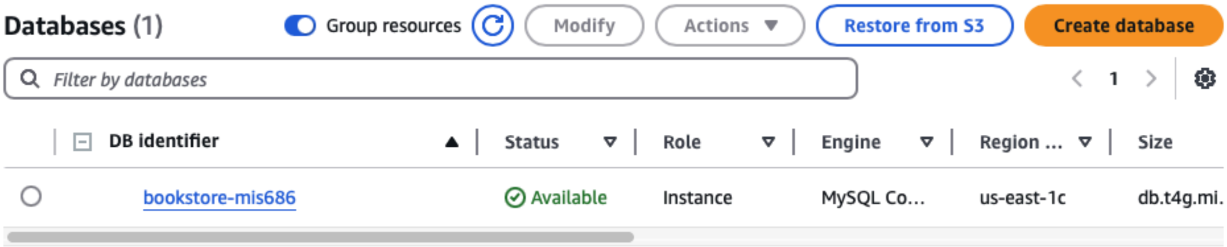
Maximum storage threshold
1000 GiB

Storage file system configuration
Current

Performance Insights

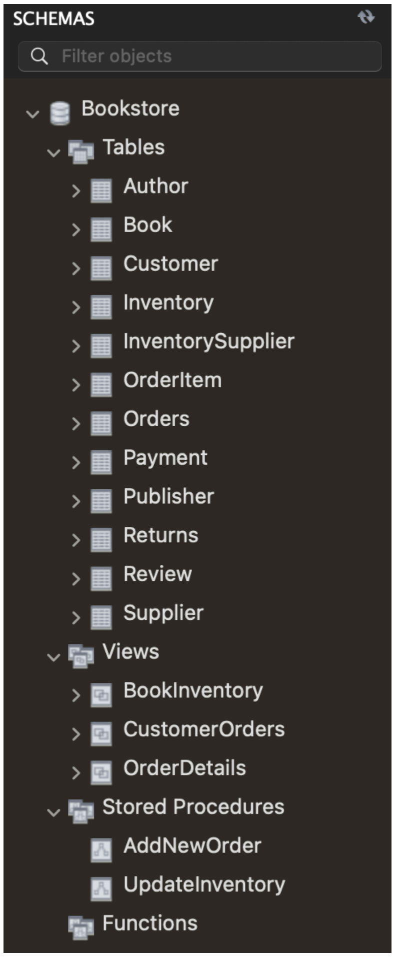
Performance Insights enabled
Turned off

Once the database status changed to “Available”, we copied the endpoint address. Using this endpoint as the Hostname, along with the username “admin” and the password configured during the database creation, we successfully connected to the database through MySQL.



	DB identifier ▲	Status ▽	Role ▽	Engine ▽	Region ... ▽	Size
○	bookstore-mis686	✓ Available	Instance	MySQL Co...	us-east-1c	db.t4g.mi.

When the database was successfully set up, we utilized the DDL Statements from the Database Implementation section to initialize the database and create the tables shown in the ERD. The schema from the deployed database in MySQL is shown in the screenshot below.



SCHEMAS

Filter objects

- Bookstore
  - Tables
    - Author
    - Book
    - Customer
    - Inventory
    - InventorySupplier
    - OrderItem
    - Orders
    - Payment
    - Publisher
    - Returns
    - Review
    - Supplier
  - Views
    - BookInventory
    - CustomerOrders
    - OrderDetails
  - Stored Procedures
    - AddNewOrder
    - UpdateInventory
  - Functions

In our database deployment, we also implemented User Access Management to enable secure and controlled access to the database. We created two users – “employee” and “trainee”, each with different permission settings. The “employee” user has full data manipulation privileges to view, add, update, or delete records. The “trainee” user can view and update data, but they cannot add new rows or delete existing ones. Once these permissions were granted, changes were applied using the “FLUSH PRIVILEGES” command to ensure the user configurations were applied immediately.

Analytical Questions

The list of simple and complex analytical questions below allow us to derive insights and understand trends in the bookstore’s business operations. These questions focus on inventory, revenue, sales performance, and customer behavior, providing a comprehensive view of the business.

Simple Questions:

1. How many books does the bookstore have in stock for each genre?
2. How much revenue has the bookstore made from orders based on order status (Delivered, Processing, etc.)?
3. What are the top ten best-selling authors based on the number of books sold?
4. What are the top ten best-selling books based on the number of copies sold?
5. How have sales trends changed over time?

Complex Questions:

6. What is the average rating given by each customer that has completed at least one review?
7. What are the top five authors whose books have generated the most revenue?
8. Which genres should the bookstore consider expanding its inventory based on revenue and customer interest?
9. Which books have above-average ratings?

Python-SQL Dashboard

The dashboard was designed to provide answers and visual diagrams to the analytical questions listed above. By integrating SQL queries with Python for data visualization, we were able to extract relevant data from the database. Python libraries, like Pandas for data manipulation and Matplotlib for visualization, were used to create charts and graphs to answer the questions. The dashboard creation followed the approach of the given template by using Datapane to put text, figures, and tables together in a comprehensive report. We first tested each of the SQL queries in MySQL to ensure the desired outcome was obtained. Then, we established a connection to the database as the admin user and ran the SQL commands inside of Python code.

The dashboard has three numbers at the top that display important metrics and is subsequently divided into four parts: Revenue, Inventory, Popularity, and Ratings. Each of these parts contains answers to the analytical questions listed above. Using Matplotlib, several types of visualizations were created, including bar charts, pie charts, and box plots.

Click here to view the dashboard creation notebook

Click here to view the dashboard

Explanation of Analytical Insights

In the Revenue Analysis, we identified which authors generated the most revenue for the bookstore to help capitalize on their popularity. Of the 25 authors in the database, only 11 contributed to the bookstore's revenue. The bar graph highlights Daniel Allen, Thomas Clark, and Robert Moore as the top three revenue-generating authors. The pie chart reveals that Science Fiction was the leading genre, accounting for approximately 46% of the bookstore’s revenue. Additionally, most revenue came from orders marked as Processing, suggesting a significant portion of revenue stems from recently placed orders. A line graph of sales over time shows consistent spikes every five weeks, followed by gradual declines until the next spike occurs.

For the Inventory Analysis, the bookstore has the highest inventory count for Science Fiction books, which aligns with the findings from the Revenue Analysis. However, while Mystery books have the second highest inventory count, they generated the second least amount of revenue, suggesting that the bookstore should consider purchasing less of this genre.

The Popularity Analysis provided valuable insights into customer behavior. The first bar graph shows that the favorite genre among the bookstore's customers is Science Fiction, which is consistent with findings from both the Revenue and Inventory analyses. Thomas Clark was the best-selling author based on the number of books sold, with *The Enchanted Forest* by Thomas Clark being the best-selling book. Interestingly, while Thomas Clark was the top seller by volume, he was only the second highest revenue-generating author. This suggests that the highest revenue-generating author likely has more expensive books or a wider variety of well-selling books.

Lastly, the Ratings Analysis focused on identifying books with above-average ratings to understand customer preferences. As expected, many of the highly rated books align with the best-selling books, though this alignment is not perfect due to the use of dummy data. The average rating given by each customer was calculated and visualized using a box plot, which revealed that the overall average of customer ratings was 3.0, with a minimum of 1 and a maximum of 5. The data is perfectly symmetrical, which is a result of dummy data generation and does not accurately reflect real-world scenarios.