BookBazaar Project Documentation

Prepared by:

Team 2

Matthew Wael
George Raafat
Zinab Elamir
Omar Hussien
Ahmed Elkady
Yasmine Sherif
Noureldin Salaheldin

July 25, 2025

Contents

List of Figures

List of Tables

Listings

Chapter 1

Project Overview & Goals

This document provides comprehensive documentation for the **BookBazaar Database System**, detailing its design, implementation, and operational procedures. The project aims to establish a robust and scalable database solution for managing books, authors, users, and a separate system for handling book reviews.

Key Goals

- To demonstrate **relational database design** and implementation using MySQL.
- To illustrate fundamental **CRUD (Create, Read, Update, Delete) operations** in SQL.
- To showcase the integration of a **NoSQL database (MongoDB)** for flexible data storage, specifically for book reviews.
- To provide a clear, reproducible setup and operational guide for both database systems.

Technologies Used

- **MySQL 8.x Community Server**: The relational database management system.
- **MongoDB 6.x**: The NoSQL database for flexible review storage.
- **MySQL CLI / Workbench**: Tools for interacting with MySQL.
- **Mongo Shell / Compass** (Optional for users): Tools for interacting with MongoDB.
- **SQL**: For defining and manipulating data in MySQL.
- **JavaScript**: For scripting CRUD operations in the Mongo Shell.

Chapter 2

Setup Instructions

This chapter provides detailed, OS-agnostic instructions for installing and initializing both MySQL and MongoDB. While specific installation steps may vary slightly depending on your operating system, the general process remains consistent.

2.1 MySQL Setup

1. Install MySQL 8.x Community Server and MySQL CLI / Workbench

To begin, download the appropriate installer for your operating system from the official MySQL website: `https://dev.mysql.com/downloads/mysql/`.

- **Windows/macOS**: Use the official installer package.
- **Linux**: Utilize your distribution's package manager (e.g., `apt` for Debian/Ubuntu, `yum` for CentOS/RHEL) or download the official archives.

During the installation, be sure to note the **root password** you set, as it will be required for initial setup. We recommend also installing the **MySQL Command Line Client** or **MySQL Workbench** for easier interaction.

2. Create the `bookbazaar` Schema

After installing MySQL, log in as the `root` user (or any user with schema creation privileges) to your MySQL server:

```
mysql -u root -p
```

Listing 2.1: Log in as the root user from PowerShell

Execute the following SQL command to create the dedicated schema:

```
CREATE DATABASE bookbazaar DEFAULT CHARSET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

Listing 2.2: Creating the bookbazaar database

This command creates a database named `bookbazaar` with full **Unicode (UTF-8mb4)** support, ensuring proper handling of various characters, including emojis.

3. Add `bookadmin` User with Specific Permissions

For security best practices, we will create a dedicated user named `bookadmin` with full rights *only* on the `bookbazaar` schema. Replace `your_secure_password` with a strong, unique password.

```
CREATE USER 'bookadmin'@'localhost' IDENTIFIED BY 'bookadmin';
GRANT ALL PRIVILEGES ON bookbazaar.* TO 'bookadmin'@'localhost';
```

Listing 2.3: Creating bookadmin user and granting privileges

Important: If you need to access MySQL from a different machine, change `'localhost'` to `'%'`. However, this is generally less secure and should be done with caution.

4. Test the MySQL Connection

To verify your MySQL setup, open your terminal or command prompt and execute the following command:

```
mysql -u bookadmin -p bookbazaar
```

Listing 2.4: Testing MySQL connection

You will be prompted to enter the `bookadmin` user's password. A successful connection will display the MySQL command prompt, indicating that you're connected to the `bookbazaar` database.

2.2 MongoDB Setup

1. Install MongoDB 6.x and Start the Daemon

Download the MongoDB Community Server for your operating system from the official MongoDB website: https://www.mongodb.com/try/download/community. Follow their installation instructions.

Then, in a new terminal, open the MongoDB Shell:

```
mongosh
```

On some operating systems, MongoDB might be set up to run automatically as a background service.

2. Create the `bookbazaar_reviews` Database

MongoDB databases are created implicitly when you first insert data into a collection within them. To prepare, open the Mongo Shell by typing `mongo` in your terminal:

```
use bookbazaar_reviews;
```

Listing 2.5: Switching to the reviews database

This command will switch you to the `bookbazaar_reviews` database context. If the database doesn't exist, MongoDB will create it upon the first data insertion.

3. (Optional) Add a User with `readWrite` on `bookbazaar_reviews`

For enhanced security, it's recommended to create a dedicated user for your reviews database. First, switch to the `admin` database, then create the user. Replace `your_secure_mongo_password` with a strong password.

```
use admin;
db.createUser(
   {
     user: "review_user",
     pwd: "your_secure_mongo_password",
     roles: [ { role: "readWrite", db: "bookbazaar_reviews" } ]
   }
);
```

Listing 2.6: Creating a MongoDB user

4. Verify MongoDB Connection

To verify your MongoDB connection, open your terminal and use one of the following commands:

- **Without a dedicated user**:

  ```
  mongo
  ```

- **With the dedicated user** (`review_user`): ¡— Check this line carefully

  ```
  mongo --username review_user --password your_secure_mongo_password --
      authenticationDatabase admin
  ```

A successful connection will bring you to the Mongo Shell prompt.

Chapter 3

How to Run Scripts

This chapter details how to execute the provided SQL and JavaScript scripts to set up your relational schema, perform CRUD operations, and interact with the MongoDB reviews collection.

3.1 Running `schema.sql`

The ‘schema.sql‘ file contains the Data Definition Language (DDL) for your ‘bookbazaar‘ MySQL database. It creates the ‘authors‘, ‘books‘, and ‘users‘ tables with appropriate constraints and populates them with initial sample data.

1. Ensure your MySQL server is running.
2. Open your terminal or command prompt.
3. Navigate to the directory where you’ve saved ‘schema.sql‘.
4. Execute the script using the ‘bookadmin‘ user for the ‘bookbazaar‘ database:

```
mysql -u bookadmin -p bookbazaar < schema.sql
```

Listing 3.1: Executing schema.sql

5. You will be prompted to enter the ‘bookadmin‘ password. Upon successful execution, your MySQL tables will be created and populated. You can verify this by logging into MySQL and running ‘SHOW TABLES;‘ or ‘SELECT * FROM authors;‘.

3.2 Running `crud_demo.sql`

The `crud_demo.sql` script showcases basic Create, Read, Update, and Delete (CRUD) operations on your `bookbazaar` MySQL database. Each operation is carefully wrapped within a transaction for easy review and and potential rollback.

1. Ensure your MySQL server is running and the `bookbazaar` schema (from `schema.sql`) is already set up.
2. Open your terminal or command prompt.
3. Navigate to the directory where you’ve saved `crud_demo.sql` .
4. Execute the script using the `bookadmin` user for the `bookbazaar` database:

```
        mysql -u bookadmin -p bookbazaar < crud_demo.sql
```

Listing 3.2: Executing crud_demo.sql

5. You will be prompted to enter the `bookadmin` password. The script will execute, demonstrating each CRUD operation.

Important Note on Transactions: Each operation within `crud_demo.sql` is enclosed by `START TRANSACTION;` and `COMMIT;`. This allows reviewers to easily roll back the changes if they wish to re-run the script from a clean state.

3.3 Running `crud_reviews.js`

The `crud_reviews.js` script is a JavaScript file designed to be run in the Mongo Shell. It demonstrates CRUD operations on the `bookbazaar_reviews` MongoDB database.

1. Ensure the MongoDB daemon (`mongod`) is running.

2. Open your terminal or command prompt.

3. Navigate to the directory where you've saved `crud_reviews.js` .

4. Execute the script using the Mongo Shell. Choose the appropriate command based on whether you set up a dedicated user:

   - **Without a dedicated user**:

   ```
               mongo < crud_reviews.js
   ```

   Listing 3.3: Executing crud_reviews.js (no user)

   - **With the dedicated user** `review_user`:

   ```
               mongo --username review_user --password
                   your_secure_mongo_password --authenticationDatabase admin <
                   crud_reviews.js
   ```

   Listing 3.4: Executing crud$_{r}eviews.js(withuser)$

5. The `printjson` statements within the script will output the results of each MongoDB operation directly to your terminal.

Chapter 4

ER Diagram & Reviews Collection Structure

This chapter provides a visual representation of the relational schema for the ‘bookbazaar‘ database and details the structure of the ‘reviews‘ collection in MongoDB.

4.1 Relational Database ER Diagram

The following Entity-Relationship (ER) diagram illustrates the design of the ‘bookbazaar‘ relational database. It shows the entities (**authors**, **users**, **books**) and their relationships, along with primary and foreign keys.

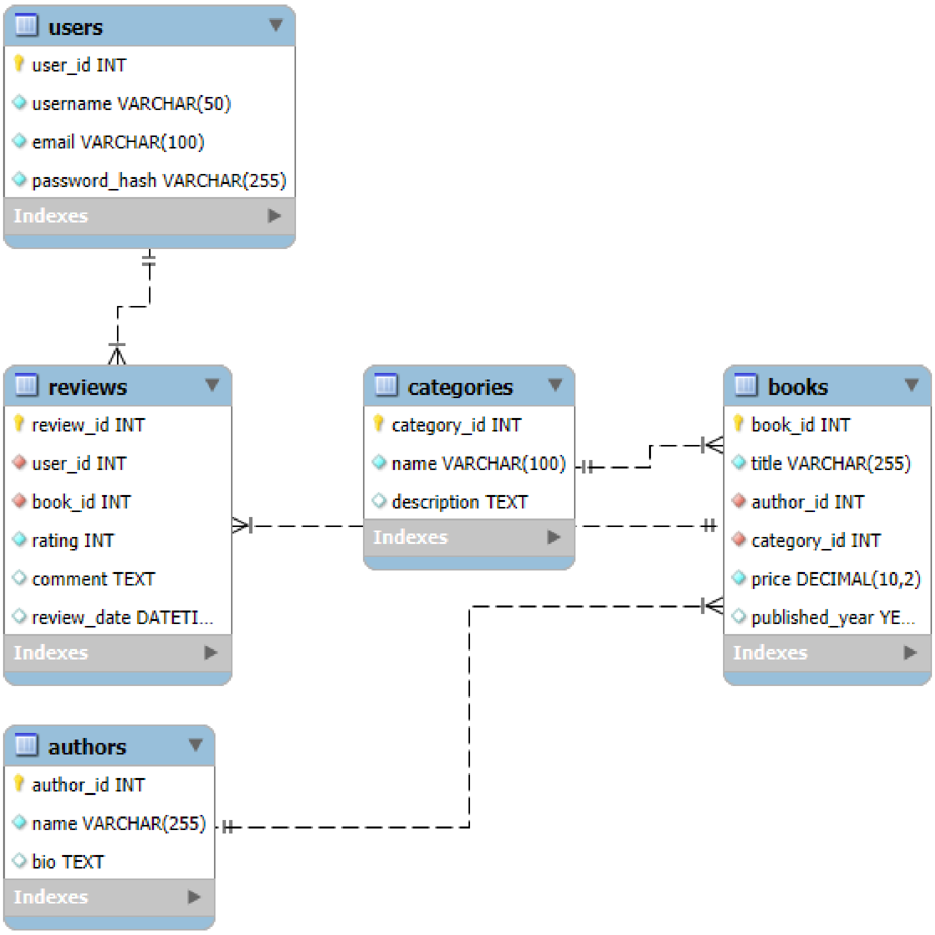

Figure 4.1: BookBazaar Relational Database ER Diagram

4.2 Reviews Collection Structure

The ‘reviews‘ collection in MongoDB is designed to store flexible and detailed review documents for books. Below is a description of its fields:

Field	**Type**	**Required?**	**Description**
˙id	`ObjectId`	✓	Unique identifier for the review document.
book˙id	`int`	✓	Foreign key referencing the `book_id` from the MySQL database. This links a review to a specific book.
reviewer	`string`	✓	Name or identifier of the user who wrote the review.
rating	`int (1-5)`	✓	Rating given to the book, ranging from 1 (lowest) to 5 (highest).
comment	`string`		Optional textual review providing details about the rating.
created˙at	`ISODate`	✓	Timestamp indicating when the review document was created.

Table 4.1: Reviews Collection Fields

Chapter 5

Troubleshooting

This chapter addresses common issues you might encounter during the setup and operation of your BookBazaar database systems, along with their potential solutions.

5.1 Common MySQL Issues

Authentication Errors (`Access Denied`)

- **Symptom**: You receive an error message like `ERROR 1045 (28000): Access denied for user...`.
- **Solution**:
  - Double-check the username and password you are using.
  - Ensure the user (‘bookadmin‘ in this case) has the correct host specified (e.g., ‘’bookadmin’@’localhost’‘). If you’re trying to connect from a different machine, and the user is only ‘localhost‘, it will fail.
  - If you are using the ‘root‘ user, ensure its password is correct. If forgotten, you may need to follow MySQL’s specific steps for resetting the root password.

Port Conflicts

- **Symptom**: MySQL server fails to start, or you cannot connect, often indicating that the default port (3306) is already in use.
- **Solution**:
  - Verify if another MySQL instance or another application is currently using port 3306.
  - Stop the conflicting service if possible.
  - Alternatively, you can configure your MySQL server to use a different port by editing its configuration file (‘my.cnf‘ on Linux/macOS or ‘my.ini‘ on Windows).

Charset Issues

- **Symptom**: Unicode characters (e.g., special symbols, non-English text, emojis) display incorrectly as question marks or strange characters.
- **Solution**:
  - Ensure that your ‘bookbazaar‘ database, as well as individual tables and columns, are configured with the ‘utf8mb4‘ character set and ‘utf8mb4_unicode_ci‘ collation. This was included in the ‘CREATE DATABASE‘ statement, but verify if you created tables manually.
  - Confirm that your MySQL client (CLI or Workbench) is also set to use UTF-8 encoding.

5.2 Common MongoDB Issues

Daemon Not Running

- **Symptom**: When trying to connect to MongoDB, you receive a ‘MongoDB connection error: connection refused‘ or similar message.
- **Solution**: This indicates that the MongoDB server process (‘mongod‘) is not running.
  - Open your terminal and start the ‘mongod‘ process.
  - On systems where MongoDB runs as a service, ensure the service is active.

Port Conflicts

- **Symptom**: MongoDB fails to start, indicating that its default port (27017) is already in use.
- **Solution**:
  - Identify and stop any other process that might be using port 27017.
  - You can configure MongoDB to run on a different port by editing its configuration file (e.g., ‘mongod.conf‘ or a custom ‘–port‘ argument when starting ‘mongod‘).

Authentication Errors

- **Symptom**: When trying to connect to MongoDB with a user, you get an ‘Authentication failed‘ error.
- **Solution**:
  - Verify the username, password, and the ‘authenticationDatabase‘ provided in your connection command.
  - Ensure that the user you are trying to connect with has the correct roles (‘readWrite‘ in this case) assigned to the specific database (‘bookbazaar_reviews‘) you wish to access.
  - Remember that authentication is often disabled by default in MongoDB. If you’ve created users, you might need to enable ‘auth‘ in your ‘mongod.conf‘ file and restart the ‘mongod‘ service for authentication to take effect.