Final Report - Sakila Database in MySQL

Bryan Alexander Morantes Cardenas
Course: Database Management
Date: 10 marzo 2025

Table of Contents

---

1. Introduction

In this assignment, I worked with the **Sakila** database in MySQL to perform various data management tasks. The objective was to enhance my understanding of SQL statements and MySQL Workbench tools, such as the SQL Editor, Schema Inspector, and Query Builder. Below are the details and results of each task as requested.

---

2. Task 1: Identifying Tools and Statements for Modifying Database Content

SQL Statements Used:

- **INSERT:** Inserts new records into tables.
- **UPDATE:** Modifies existing records in tables.
- **DELETE:** Removes records from tables.
- **ALTER:** Modifies the structure of an existing table, such as adding or removing columns.

MySQL Workbench Tools:

- **SQL Editor:** Allows writing and executing SQL queries.
- **Schema Inspector:** Displays the detailed structure of tables and their relationships.
- **Query Builder:** A visual tool to construct SQL queries without writing the code manually.

[Insert screenshots or examples of these tools if needed]

---

3. Task 2: Data Insertion, Deletion, and Update

- **Insertion:** A new fictitious actor was inserted into the actor table using the INSERT statement.
- **Update:** The last name of an existing actor was updated using the UPDATE statement.
- **Deletion:** An actor was deleted from the table using the DELETE statement.

Executed SQL:

- SQL File: 02_modify_actor.sql
- Screenshots: 02_modify_actor_screenshots/

---

4. Task 3: Creating a Table from a Query Result

- I executed a query to retrieve all films released after 2005 from the film table.
- The results were stored in a new table called recent_films.

SQL Script:

```
sql
CopiarEditar
SELECT * INTO recent_films
FROM film
WHERE release_year > 2005;
```

Executed SQL:

- SQL File: 03_create_recent_films.sql
- Screenshot: 03_recent_films_screenshot.png

---

5. Task 4: Designing Complex SQL Scripts

The following SQL script was written to:

- List all customers who rented a film in the last 30 days.
- Identify the most rented film in the database.
- Display the total revenue generated per store.

SQL Script:

```
sql
CopiarEditar
-- List customers who rented films in the last 30 days
SELECT customer_id, first_name, last_name
FROM customer
WHERE customer_id IN (
   SELECT DISTINCT customer_id
   FROM rental
```

```
    WHERE rental_date > CURDATE() - INTERVAL 30 DAY
);

-- Most rented film
SELECT title, COUNT(*) AS rental_count
FROM film
JOIN rental ON film.film_id = rental.film_id
GROUP BY title
ORDER BY rental_count DESC
LIMIT 1;

-- Total revenue per store
SELECT store_id, SUM(amount) AS total_revenue
FROM payment
GROUP BY store_id;
```

SQL File: 04_complex_queries.sql

---

6. Task 5: Understanding Transactions

A transaction was performed that:

- Inserted a new rental record.
- Updated the inventory to reflect the rental.
- Committed the transaction.

SQL File:

- SQL File: 05_transaction_example.sql
- Section in the final report covering transactions.

---

7. Task 6: Rolling Back Transactions

A scenario was demonstrated where a transaction was partially executed but later rolled back due to an error (e.g., an attempt to rent a movie that is out of stock).

SQL File:

- SQL File: 06_rollback_example.sql
- Section in the final report covering rollback transactions.

---

8. Task 7: Understanding Record Locking Policies

I researched different types of record-locking mechanisms (pessimistic vs. optimistic locking) and tested a scenario where two users attempted to update the same record simultaneously.

Section in the final report covering record locking policies.

---

9. Task 8: Ensuring Data Integrity and Consistency

I identified potential data integrity issues in the Sakila database and implemented foreign key constraints and triggers to maintain data consistency.

SQL File:

- SQL File: 08_data_integrity.sql
- Section in the final report covering data integrity.

---

10. Conclusion

This assignment helped me develop a deeper understanding of SQL statements, transactions, and the importance of ensuring data integrity in a relational database. By applying theoretical knowledge through practical tasks, I was able to work on real-world database management issues, gaining insights into MySQL Workbench tools and the Sakila database structure.

Challenges and Reflections:

Throughout this assignment, I faced a few challenges, primarily related to understanding foreign key constraints, triggers, and handling transactions. Implementing the trigger to prevent rentals with out-of-stock inventory was tricky due to syntax issues and error handling in MySQL. Additionally, learning to structure transactions and properly manage rollbacks took some time, especially when simulating real-world errors like out-of-stock rentals.

Designing complex SQL queries was also a challenge, particularly when joining multiple tables and aggregating data for reports. However, with practice and reference to documentation, I was able to refine my approach and successfully complete the tasks.

Overall, the assignment helped me better understand MySQL features and SQL scripting, and it allowed me to overcome the technical difficulties with persistence and problem-solving.

References:

1. MySQL Documentation: https://dev.mysql.com/doc/
2. "Learning MySQL" by Seyed M.M. (2020).
3. MySQL Workbench Manual: https://dev.mysql.com/doc/workbench/en/
4. ChatGPT: https://chatgpt.com/