Documentation: Library Management System Database Project Progress

---

Introduction

This document details the step-by-step progress of the Library Management System (LMS) database project, covering team roles, decisions, tasks completed, and challenges faced. It serves as a comprehensive record of our efforts, from platform selection to documentation and testing.

---

1. Team Roles and Responsibilities

November 24th, 2024

Roles were assigned to team members based on individual strengths and expertise:

- **Talha**: Platform selection, filling gaps in any task/role, managing GitHub repositories.
- **Deborah**: Documentation of the entire process.
- **Atharva**: Database schema creation (DDL scripts).
- **Conner**: SQL query writing and functionality testing.
- **Abdullah**: Testing and validating the database.
- **Alec**: Populating the database with sample data.

---

2. Platform Selection

Initial Platform: PostgreSQL

- Selected due to team familiarity and its robust tools like **pgAdmin** and **CLI utilities** for schema management, data import, and query testing.

Platform Change: MySQL (November 30th, 2024)

- Decided during a group meeting to switch to MySQL due to its simplicity, beginner-friendly features, and better alignment with the team's skill set.
- Talha managed the creation of GitHub repositories for code sharing and collaboration.

3. Project Task Timeline

A timeline was established during the meeting on **November 24th**:

Task	Description	Deadline
Choose platform	Finalize PostgreSQL as the database platform.	Nov 24
Create schema	Write and test DDL scripts to set up the database.	Nov 26
Populate data	Source/generate realistic data and import it.	Nov 28
Functionality testing	Write and execute SQL queries to test functionality.	Nov 30
Test and validate	Validate results, handle edge cases, and optimize.	Dec 2
Documentation	Compile a detailed report with setup steps, challenges, and outputs.	Dec 3
Final review and demo	Ensure all deliverables are complete and ready.	Dec 3

---

4. Key Progress Updates

November 25th, 2024

- Talha and Abdullah had a **9-minute call** to update Abdullah on the group meeting he missed the previous day.

November 30th, 2024

- A group meeting was held to review progress.
- **Decision to switch to MySQL**: Based on team skills and simplicity, MySQL was deemed a better fit.
- **Testing the DDL Script**:
    - Abdullah verified that the **DDL script worked locally on MySQL**.
    - He confirmed that all tables were created correctly and tested them with random values.

December 1st, 2024

- Atharva updated the DDL script based on grader feedback from Project Part 4.
- **New Attribute**: Atharva proposed adding a genre attribute to the Resource table to improve usability. The team approved the addition.

December 2nd, 2024

- For sample data display, the team chose **.md or .csv formats** over .sql for better readability.

- **Key Schema Enhancements**:
    - **Composite Key**: PRIMARY KEY (userID, resourceID) was implemented.
    - **Granularity Improvement**: Added isAvailable BOOLEAN NOT NULL to the Resource table.
    - **ENUM Usage**: Extended the ENUM attribute from User to FeePolicy for consistency.
- Conner developed additional queries for functional requirements but faced challenges in testing non-functional requirements.

December 3, 2024

- **Alec** completed the task of populating tables with realistic sample data using Python's **Faker** library. This ensured diverse and realistic entries for testing purposes.
- Conner and Abdullah collaborated to test the SQL queries created for the project's functional requirements.
- They focused on validating query accuracy and functionality, troubleshooting minor issues that arose during testing.
- **Team Meeting:**
    - The team discovered that schema changes from December 2 hadn’t committed to GitHub.

- To minimize troubleshooting, Conner and Alec decided to proceed with the previous schema version for data population and query creation.

---

5. Functionality Testing and Results

SQL Queries Developed:

- Retrieve resources by title, author, or genre.
- Generate borrowing reports for users.
- Calculate late fees dynamically based on FeePolicy.

Key Features:

- **Aggregate Functions**: For summaries like total items borrowed by a user.
- **Stored Procedures**: Automate repetitive tasks like calculating late fees.
- **Views**: Simplified complex queries for frequent use cases.

Below are a few snippets of the queries, testing and results:

```
mysql> SELECT b.borrowID, u.name AS userName, r.title, b.dueDate, DATEDIFF(CURRENT_DATE, b.dueDate) AS daysOverdue
    -> FROM Borrowing b
    -> JOIN User u ON b.userID = u.userID
    -> JOIN Resource r ON b.resourceID = r.resourceID
    -> WHERE b.returnDate IS NULL AND b.dueDate < CURRENT_DATE
    ->   LIMIT 10; --Only prints the first 10 results for testing
+----------+------------------+----------------------------------------+------------+-------------+
| borrowID | userName         | title                                  | dueDate    | daysOverdue |
+----------+------------------+----------------------------------------+------------+-------------+
|        3 | Steven Kerr      | His purpose this.                      | 2024-08-10 |         115 |
|        8 | Russell Benton   | Under author throw design onto.        | 2024-03-17 |         261 |
|       13 | Paul Perez DDS   | Religious perform language receive nice. | 2024-06-06 |         180 |
|       19 | Melanie Patel    | Garden market place.                   | 2024-10-07 |          57 |
|       24 | Jeanette Young   | Most cover.                            | 2024-01-31 |         307 |
|       26 | Jasmine Campbell | Citizen agent.                         | 2024-07-09 |         147 |
|       32 | Jasmine Walker   | Wrong discuss low thought.             | 2024-10-15 |          49 |
|       37 | George Hale      | Box theory again onto.                 | 2024-07-15 |         141 |
|       48 | Amanda Choi      | Quickly trip ask.                      | 2024-11-03 |          30 |
|       51 | Crystal Dickerson | Think ask now language.                | 2024-08-18 |         107 |
+----------+------------------+----------------------------------------+------------+-------------+
10 rows in set (0.01 sec)

    ->
```

```
mysql>
mysql> SELECT resourceID, title, type, availableCopies, totalCopies
    -> FROM Resource
    -> WHERE availableCopies > 0
    -> LIMIT 10;
+------------+---------------------------------+--------------+-----------------+-------------+
| resourceID | title                           | type         | availableCopies | totalCopies |
+------------+---------------------------------+--------------+-----------------+-------------+
|          1 | Learn SQL                       | Book         |               5 |          10 |
|          2 | Database Design                 | Book         |               3 |           5 |
|          3 | Fictional Tales                 | Book         |               2 |           4 |
|          4 | Environment yourself hot pattern. | Digital Media |             101 |         116 |
|          5 | Like throw guess.               | Book         |              37 |         155 |
|          6 | Economic budget degree then.    | Magazine     |             170 |         178 |
|          7 | Expert TV light low.            | Digital Media |              93 |         184 |
|          8 | Mother ago.                     | Magazine     |              10 |          80 |
|          9 | Team science they day window.   | Magazine     |             120 |         135 |
|         10 | Heart believe.                  | Digital Media |              62 |         151 |
+------------+---------------------------------+--------------+-----------------+-------------+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
```

```
mysql> SELECT r.title, COUNT(b.borrowID) AS timesBorrowed
    -> FROM Borrowing b
    -> JOIN Resource r ON b.resourceID = r.resourceID
    -> GROUP BY r.title
    -> ORDER BY timesBorrowed DESC
    -> LIMIT 10;
+---------------------------------+---------------+
| title                           | timesBorrowed |
+---------------------------------+---------------+
| Measure care throw toward.      |             6 |
| Owner then seven physical base. |             6 |
| Baby have really.               |             5 |
| Increase chance.                |             5 |
| Hear rest.                      |             4 |
| Sit piece large.                |             4 |
| Push fill look sense.           |             4 |
| Safe up begin media.            |             4 |
| Field gun agency military.      |             4 |
| Matter lose appear open.        |             4 |
+---------------------------------+---------------+
10 rows in set (0.03 sec)

mysql>
```

```
mysql> SELECT b.borrowID, r.title, b.borrowDate, b.dueDate, b.returnDate, b.lateFee
    -> FROM Borrowing b
    -> JOIN Resource r ON b.resourceID = r.resourceID
    -> WHERE b.userID = 1;
+-----------+------------------------+------------+------------+------------+---------+
| borrowID  | title                  | borrowDate | dueDate    | returnDate | lateFee |
+-----------+------------------------+------------+------------+------------+---------+
|      6001 | Western our sometimes. | 2024-11-24 | 2024-12-01 | NULL       |    0.00 |
|     10000 | Western our sometimes. | 2024-11-24 | 2024-12-01 | NULL       |    0.00 |
| 988873723 | Western our sometimes. | 2024-11-24 | 2024-12-01 | NULL       |    0.00 |
+-----------+------------------------+------------+------------+------------+---------+
3 rows in set (0.01 sec)

mysql>
```

SELECT b.borrowID, r.title, b.borrowDate, b.dueDate, b.returnDate, b.lateFee

Done

```
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3888, 1552, 5249, '2023-12-05', '2024-01-10', '2024-01-03', 2
3.14, 21);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3889, 574, 9849, '2024-10-06', '2024-10-23', '2024-10-16', 3.
64, 25);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3890, 4467, 4183, '2024-07-26', '2024-08-10', '2024-08-04', 1
3.92, 6);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3891, 1727, 719, '2024-05-10', '2024-05-16', '2024-05-29', NU
LL, 20);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3892, 3707, 951, '2024-07-11', '2024-08-09', '2024-07-26', 20
.71, 15);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3893, 577, 1784, '2023-12-29', '2024-01-03', '2024-01-08', NU
LL, 22);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3894, 576, 4449, '2024-07-04', '2024-07-07', '2024-07-23', NU
LL, 6);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3895, 2317, 4397, '2024-09-16', 'NULL', '2024-10-01', NULL, 1
9);
ERROR 1292 (22007): Incorrect date value: 'NULL' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3896, 4700, 6698, '2024-07-11', '2024-07-12', '2024-08-01', N
ULL, 6);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3897, 3420, 5509, '2024-11-21', '2024-11-22', '2024-11-30', N
ULL, 25);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3898, 113, 4309, '2024-04-07', '2024-04-27', '2024-04-19', 31
.74, 15);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3899, 1067, 6739, '2024-09-09', '2024-09-18', '2024-10-04', N
ULL, 10);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3900, 3866, 5294, '2024-01-03', '2024-01-10', '2024-01-16', N
ULL, 24);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (3901, 4132, 4878, '2024-10-22', '2024-10-27', '2024-10-31', N
ULL, 3);
```

```
10001 rows in set (0.01 sec)

mysql> DELETE FROM Resource
    -> WHERE resourceID = 1001;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`librarydb`.`borrowing`, CONSTRAINT `borrowing_ibfk_2` FOREIGN KEY (`resourceID`) REFERENCE
S `resource` (`resourceID`))
mysql>
mysql> DELETE FROM Resource
    -> WHERE resourceID = 7977;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`librarydb`.`borrowing`, CONSTRAINT `borrowing_ibfk_2` FOREIGN KEY (`resourceID`) REFERENCE
S `resource` (`resourceID`))
mysql>
mysql> UPDATE Resource
    -> SET availableCopies = availableCopies - 1
    -> WHERE resourceID = 1001 AND availableCopies > 0;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

```
ULL, 15);
ERROR 1292 (22007): Incorrect date value: '1991' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1250, 4802, 4461, '2024-05-02', NULL, '2024-05-30', NULL, 2
);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`librarydb`.`borrowing`, CONSTRAINT `borrowing_ibfk_2` FOREIGN KEY (`resourceID`) REFERENCES
`resource` (`resourceID`))
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1251, 1178, 5863, '2024-06-21', 2024-06-24, '2024-06-28', N
ULL, 17);
ERROR 1292 (22007): Incorrect date value: '1994' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1252, 2140, 5049, '2024-02-09', 2024-03-01, '2024-02-18', 2
6.7, 13);
ERROR 1292 (22007): Incorrect date value: '2020' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1253, 4132, 1387, '2023-12-29', 2024-01-29, '2024-01-26', 3
.01, 2);
ERROR 1292 (22007): Incorrect date value: '1994' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1254, 3465, 5929, '2024-03-31', 2024-04-29, '2024-04-14', 6
8.81, 11);
ERROR 1292 (22007): Incorrect date value: '1991' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1255, 530, 7677, '2024-04-28', 2024-05-04, '2024-05-11', NU
LL, 6);
ERROR 1292 (22007): Incorrect date value: '2015' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1256, 2210, 4988, '2024-01-04', 2024-02-14, '2024-01-31', 1
7.92, 11);
ERROR 1292 (22007): Incorrect date value: '2008' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1257, 2343, 3659, '2024-03-16', 2024-03-31, '2024-03-24', 2
7.83, 22);
ERROR 1292 (22007): Incorrect date value: '1990' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1258, 2761, 2061, '2023-12-12', 2024-01-02, '2023-12-27', 2
0.08, 1);
ERROR 1292 (22007): Incorrect date value: '2021' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1259, 242, 4062, '2023-12-16', 2023-12-18, '2023-12-24', NU
LL, 20);
ERROR 1292 (22007): Incorrect date value: '1993' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1260, 2738, 7598, '2024-04-03', 2024-04-17, '2024-04-12', 1
1.59, 23);
ERROR 1292 (22007): Incorrect date value: '2003' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1261, 1236, 6968, '2024-04-03', 2024-04-24, '2024-04-25', N
ULL, 22);
ERROR 1292 (22007): Incorrect date value: '1996' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1262, 4093, 8439, '2024-09-15', 2024-10-16, '2024-10-14', 3
.93, 22);
ERROR 1292 (22007): Incorrect date value: '1998' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1263, 2498, 3325, '2024-11-21', 2024-12-14, '2024-12-03', 1
6.8, 7);
ERROR 1292 (22007): Incorrect date value: '1998' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1264, 899, 9694, '2024-02-20', 2024-03-03, '2024-03-18', NU
LL, 12);
ERROR 1292 (22007): Incorrect date value: '2018' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1265, 1620, 2533, '2024-05-30', NULL, '2024-06-25', NULL, 1
8);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`librarydb`.`borrowing`, CONSTRAINT `borrowing_ibfk_2` FOREIGN KEY (`resourceID`) REFERENCES
`resource` (`resourceID`))
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1266, 4456, 8739, '2024-02-14', 2024-03-02, '2024-02-28', 3
.88, 23);
ERROR 1292 (22007): Incorrect date value: '2019' for column 'returnDate' at row 1
mysql> INSERT INTO Borrowing (borrowID, userID, resourceID, borrowDate, returnDate, dueDate, lateFee, policyID) VALUES (1267, 2539, 1335, '2023-12-09', 2024-01-02, '2024-01-07', N
```

---

6. Challenges and Solutions

1. **Platform Migration**
    - **Challenge:** The transition from PostgreSQL to MySQL required significant re-testing to ensure schema compatibility and functionality.
    - **Solution:** Atharva adapted the DDL scripts for MySQL, while Abdullah thoroughly tested them to confirm that all schema components worked as intended. The team collaboratively ensured the transition did not disrupt progress.

2. **Schema Updates**
    - **Challenge:** Integrating grader feedback and implementing new attributes like genre and isAvailable necessitated iterative modifications to the schema.
    - **Solution:** Atharva updated the schema based on feedback and team discussions, ensuring alignment with project requirements. The updates were validated through rigorous team testing.
3. **Data Consistency**
    - **Challenge:** Maintaining data integrity during bulk imports was critical, especially given the volume of generated data.
    - **Solution:** Python scripts automated data insertion, and **transaction control** was employed to ensure rollback in case of errors. This approach ensured sample data accuracy and prevented inconsistencies in the database.
4. **Commit Issues**
    - **Challenge:** Schema changes made on December 2 did not successfully commit to GitHub, leading to discrepancies between schema versions.
    - **Solution:** To minimize troubleshooting, the team agreed to proceed with the previous schema version for data population and query creation. This decision helped maintain project momentum while avoiding further delays.

---

7. Database Setup and Data Import

Database Setup

- The initial database schema was created by designing SQL tables based on the project's functional and technical requirements.
- The schema underwent iterative refinements, incorporating feedback from the TA and input from group members to align with the functional goals and improve usability.

- Modifications included the addition of attributes such as genre and isAvailable for better granularity, as well as adjustments to accommodate feedback from earlier project parts.

Data Generation and Import

- The team utilized Python's **Faker** library to generate realistic sample data. This tool provided diverse entries, such as names, company names, and other fields crucial for the system's functionality.
- **Data Import Process:**
    - Python scripts were used to programmatically generate **SQL INSERT** statements.
    - The generated statements populated the database tables efficiently and consistently, reducing the risk of manual data entry errors.
    - This automated approach also allowed for quick scalability, enabling the team to generate and import additional data as needed during testing.

Functionality Testing

- Queries were developed to meet and validate the functional requirements outlined in the documentation:
    - **Borrowing Management:** Queries for issuing, returning, and managing borrowed resources.
    - **Resource Management:** Queries to manage resources, including availability and attributes.
    - **Search and Reports:** Queries to search for resources by title, author, or genre and generate user borrowing reports.
    - **User Management:** Queries for creating, updating, and managing user records.

Key Notes:

- Foreign key constraints occasionally restricted deletions from parent tables. In such cases, dependent rows in child tables were also deleted to maintain referential integrity.
- Functional testing confirmed that the queries performed as expected, producing accurate and correct results.
- The search and report queries primarily provide general examples for displaying data, while queries for other functionalities demonstrate SQL operations to modify respective tables effectively.

Conclusion

The Library Management System (LMS) database project exemplified collaborative problem-solving, technical expertise, and adaptability. From initial planning to final testing, the team worked cohesively, leveraging individual strengths to overcome challenges such as platform migration, schema refinement, and data consistency.

The transition from PostgreSQL to MySQL, though unplanned, demonstrated the team's ability to pivot and adapt while maintaining project momentum. Continuous schema updates and the use of tools like Python's Faker library ensured a robust, realistic database capable of supporting the project's functional requirements. Additionally, functionality testing validated the effectiveness of the queries and their alignment with the system's goals, emphasizing data integrity and operational efficiency.

Through iterative improvements, transparent communication, and meticulous documentation, the team successfully delivered a functional, well-documented database system. This project not only met its technical objectives but also fostered valuable teamwork and learning experiences, ensuring readiness for future challenges in database design and management.