UNIVERSITY
LIBRARY MANAGEMENT SYSTEM
Assignment # 3

GROUP MEMBERS

Ahmed Abasimel

Efrata Bayle

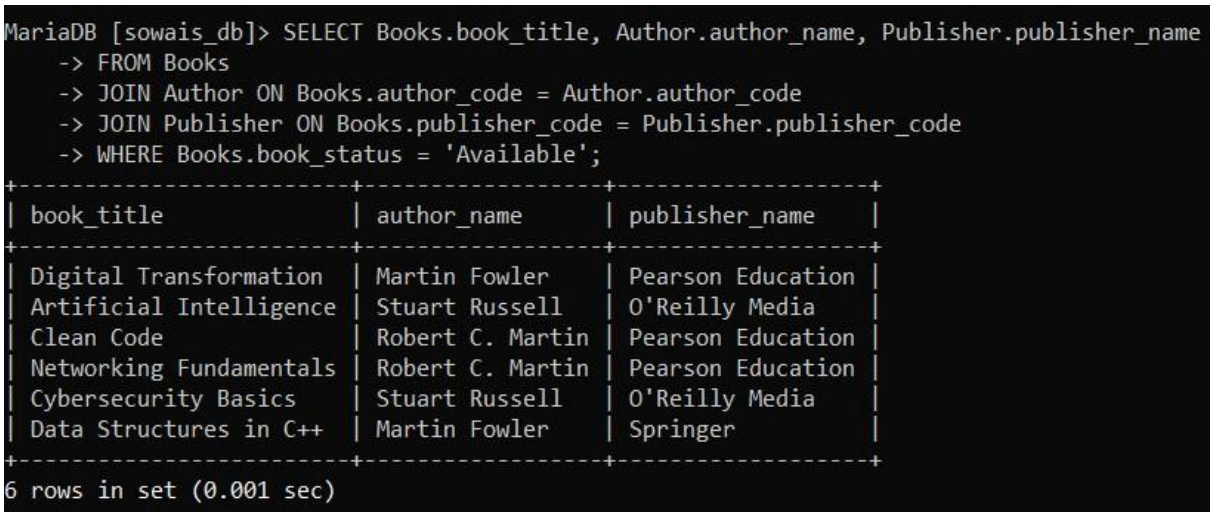
Muhammad Owais Suhail

Database used:sowais_db

In this project, SQL queries are used to manage and retrieve data efficiently from the library management system. These queries involve joining tables, applying conditions, and grouping data to extract meaningful insights. Below is an example of such queries that help perform various operations like finding available books, retrieving author and publisher details, and aggregating data across the library system.

1) SELECT Books.book_title, Author.author_name, Publisher.publisher_name
   FROM Books
   JOIN Author ON Books.author_code = Author.author_code
   JOIN Publisher ON Books.publisher_code = Publisher.publisher_code
   WHERE Books.book_status = 'Available';

   This query retrieves the titles of available books, along with their authors and publishers.



```
MariaDB [sowais_db]> SELECT Books.book_title, Author.author_name, Publisher.publisher_name
    -> FROM Books
    -> JOIN Author ON Books.author_code = Author.author_code
    -> JOIN Publisher ON Books.publisher_code = Publisher.publisher_code
    -> WHERE Books.book_status = 'Available';
+------------------------+------------------+-------------------+
| book_title             | author_name      | publisher_name    |
+------------------------+------------------+-------------------+
| Digital Transformation | Martin Fowler    | Pearson Education |
| Artificial Intelligence | Stuart Russell   | O'Reilly Media    |
| Clean Code             | Robert C. Martin | Pearson Education |
| Networking Fundamentals | Robert C. Martin | Pearson Education |
| Cybersecurity Basics   | Stuart Russell   | O'Reilly Media    |
| Data Structures in C++ | Martin Fowler    | Springer          |
+------------------------+------------------+-------------------+
6 rows in set (0.001 sec)
```

2) SELECT Vendor.vendor_code, Vendor.contact_no, SUM(Books.book_price) AS total_sales
FROM Vendor
JOIN Books ON Vendor.vendor_code = Books.vendor_code
GROUP BY Vendor.vendor_code, Vendor.contact_no

This query calculates the total sales (sum of book prices) for each vendor, grouped by vendor code and contact number.

```
MariaDB [sowais_db]> SELECT Vendor.vendor_code, Vendor.contact_no, SUM(Books.book_price) AS total_sales
    -> FROM Vendor
    -> JOIN Books ON Vendor.vendor_code = Books.vendor_code
    -> GROUP BY Vendor.vendor_code, Vendor.contact_no;
+-------------+------------+-------------+
| vendor_code | contact_no | total_sales |
+-------------+------------+-------------+
|         301 | 555-2345   |       77.48 |
|         302 | 555-6789   |       89.38 |
|         303 | 555-9876   |       48.99 |
+-------------+------------+-------------+
3 rows in set (0.001 sec)
```

3) SELECT
(SELECT COUNT(*) FROM E_books) AS total_ebooks,
(SELECT COUNT(*) FROM Physical_books) AS total_physical_books;

This query counts the total number of eBooks and physical books in the system by using subqueries to count entries from both tables while also using an alias.

```
MariaDB [sowais_db]> SELECT
    ->   (SELECT COUNT(*) FROM E_books) AS total_ebooks,
    ->   (SELECT COUNT(*) FROM Physical_books) AS total_physical_books;
+--------------+----------------------+
| total_ebooks | total_physical_books |
+--------------+----------------------+
|            3 |                    2 |
+--------------+----------------------+
1 row in set (0.000 sec)
```

4) SELECT Publisher.publisher_name, AVG(Books.book_price) AS avg_price
FROM Publisher
JOIN Books ON Publisher.publisher_code = Books.publisher_code
GROUP BY Publisher.publisher_name;

This query calculates the average book price for each publisher by grouping the results by publisher name.

```
MariaDB [sowais_db]> SELECT Publisher.publisher_name, AVG(Books.book_price) AS avg_price
    -> FROM Publisher
    -> JOIN Books ON Publisher.publisher_code = Books.publisher_code
    -> GROUP BY Publisher.publisher_name;
+-------------------+-----------+
| publisher_name    | avg_price |
+-------------------+-----------+
| O'Reilly Media    | 29.793333 |
| Pearson Education | 25.826667 |
| Springer          | 24.495000 |
+-------------------+-----------+
3 rows in set (0.001 sec)
```

5) SELECT Members.first_name, Members.last_name, Books.book_title, Borrowing_history.Issue_date
FROM Members
JOIN Borrowing_history ON Members.member_id = Borrowing_history.member_id
JOIN Books ON Borrowing_history.book_id = Books.book_id;

This query retrieves the first name, last name of members, book titles, and browsing history (browse date) by joining members, browsing history, and books tables.

```
MariaDB [sowais_db]> SELECT Members.first_name, Members.last_name, Books.book_title, Borrowing_history.Issue_date FROM Members JOIN Borrowing_history ON Members.member_id = Borrowing_history.member_id JOIN Books
 ON Borrowing_history.book_id = Books.book_id;
+------------+-----------+---------------------------------------+------------+
| first_name | last_name | book_title                            | Issue_date |
+------------+-----------+---------------------------------------+------------+
| John       | Doe       | Digital Transformation                | 2024-09-01 |
| Jane       | Smith     | Artificial Intelligence               | 2024-09-10 |
| Alice      | Johnson   | Clean Code                            | 2024-09-05 |
| Robert     | Brown     | Empowering Machines to Learn and Evolve | 2024-09-08 |
| Mary       | Williams  | Master C++ Programming Language       | 2024-09-15 |
| James      | Anderson  | Networking Fundamentals               | 2024-10-01 |
| Emily      | Davis     | Cybersecurity Basics                  | 2024-10-02 |
| Michael    | Brown     | Data Structures in C++                | 2024-10-03 |
+------------+-----------+---------------------------------------+------------+
8 rows in set (0.001 sec)
```

6) SELECT book_title, book_price
   FROM Books
   ORDER BY book_price DESC
   LIMIT 3;

This query lists the top 3 most expensive books by ordering the books by price in descending order and limiting the result to the first 3.

```
MariaDB [sowais_db]> SELECT book_title, book_price
    -> FROM Books
    -> ORDER BY book_price DESC
    -> LIMIT 3;
+-------------------------+------------+
| book_title              | book_price |
+-------------------------+------------+
| Cybersecurity Basics    |      42.50 |
| Networking Fundamentals |      35.99 |
| Clean Code              |      30.50 |
+-------------------------+------------+
3 rows in set (0.001 sec)
```

7) SELECT Books.book_title
   FROM Books
   JOIN Publisher ON Books.publisher_code = Publisher.publisher_code
   WHERE Publisher.publisher_name = 'Pearson Education';

This query lists the titles of all books published by “Pearson Education” using a where clause to filter based on the publisher’s name.

```
MariaDB [sowais_db]> SELECT Books.book_title
    -> FROM Books
    -> JOIN Publisher ON Books.publisher_code = Publisher.publisher_code
    -> WHERE Publisher.publisher_name = 'Pearson Education';
+-------------------------+
| book_title              |
+-------------------------+
| Digital Transformation  |
| Clean Code              |
| Networking Fundamentals |
+-------------------------+
3 rows in set (0.001 sec)
```

8) SELECT FacultyMember.member_id, Members.first_name, Members.last_name, COUNT(Borrowing_history.book_id) AS total_borrowed
FROM FacultyMember
JOIN Members ON FacultyMember.member_id = Members.member_id
JOIN Borrowing_history ON Members.member_id = Borrowing_history.member_id
GROUP BY FacultyMember.member_id
ORDER BY total_borrowed DESC
LIMIT 3;

This query lists the top 3 faculty members who have borrowed the most books, grouping by their member IDs and ordering by total books borrowed.

```
MariaDB [sowais_db]> SELECT FacultyMember.member_id, Members.first_name, Members.last_name, COUNT(Borrowing_history.book_id) AS total_borrowed
    -> FROM FacultyMember
    -> JOIN Members ON FacultyMember.member_id = Members.member_id
    -> JOIN Borrowing_history ON Members.member_id = Borrowing_history.member_id
    -> GROUP BY FacultyMember.member_id
    -> ORDER BY total_borrowed DESC
    -> LIMIT 3;
+-----------+------------+-----------+----------------+
| member_id | first_name | last_name | total_borrowed |
+-----------+------------+-----------+----------------+
|         2 | Jane       | Smith     |              1 |
|         4 | Robert     | Brown     |              1 |
|         6 | James      | Anderson  |              1 |
+-----------+------------+-----------+----------------+
3 rows in set (0.001 sec)
```

9) SELECT Members.first_name, Members.last_name, Books.book_title, Borrowing_history.Duration
FROM Borrowing_history
JOIN Members ON Borrowing_history.member_id = Members.member_id
JOIN Books ON Borrowing_history.book_id = Books.book_id
WHERE Borrowing_history.Duration > '00:30:00';

This query retrieves the books that members have browsed for more than 30 minutes, along with member details and duration spent.

```
MariaDB [sowais_db]> SELECT Members.first_name, Members.last_name, Books.book_title, Borrowing_history.Duration
    -> FROM Borrowing_history
    -> JOIN Members ON Borrowing_history.member_id = Members.member_id
    -> JOIN Books ON Borrowing_history.book_id = Books.book_id
    -> WHERE Borrowing_history.Duration > '00:30:00';
+------------+-----------+----------------------------------------+----------+
| first_name | last_name | book_title                             | Duration |
+------------+-----------+----------------------------------------+----------+
| John       | Doe       | Digital Transformation                 | 14:00:00 |
| Jane       | Smith     | Artificial Intelligence                | 10:00:00 |
| Alice      | Johnson   | Clean Code                             | 07:00:00 |
| Robert     | Brown     | Empowering Machines to Learn and Evolve | 05:00:00 |
| Mary       | Williams  | Master C++ Programming Language        | 15:00:00 |
| James      | Anderson  | Networking Fundamentals                | 06:00:00 |
| Emily      | Davis     | Cybersecurity Basics                   | 07:00:00 |
| Michael    | Brown     | Data Structures in C++                 | 09:00:00 |
+------------+-----------+----------------------------------------+----------+
8 rows in set (0.001 sec)
```