ADAJAR, MILVI M. BSCpE 2A1 Database Management System

Chapter 2: Structured Query Language (SQL) Laboratory Activity 3:

Laboratory Title: Structured Query Language (SQL) - Basic Queries Chapter No. and Topic: Chapter 2 - Structured Query Language (SQL) Discussions:

This activity covers the basics of querying data from a table using SQL.

Activity Description:

Learn how to retrieve data using SELECT, filter with WHERE clauses, and sort results using ORDER BY.

Objectives:

- Write basic SQL queries using SELECT.
- Apply filters using WHERE clauses.
- Sort results using ORDER BY.

Materials:

• MySQL Workbench or SQL client

Procedure:

- 1. Open MySQL Workbench and connect to the LibraryManagement database.
- 2. Retrieve all columns from the Books table:

sql
Copy code
SELECT * FROM Books;

27 • SELECT * FROM Books;

1. Retrieve books with the genre 'Fiction':

Copy code SELECT * FROM Books WHERE Genre = 'Fiction'; SELECT * FROM Books WHERE Genre = 'Fiction'; 26 • INSERT INTO Books (Title, Author, ISBN, Genre) VALUES 27 • ('The Great Gatsby', 'F. Scott Fitzgerald', '9780743273565', 'Classic'), 28 29 Edit: 🚄 🖶 🖶 Export/Import: 识 👸 Wrap C tesult Grid 🔢 🔷 Filter Rows: BookID Title Author ISBN Genre To Kill a Mockingbird Harper Lee 9780061120084 Fiction 8 The Catcher in the Rye J.D. Salinger 9780316769488 Fiction 79 To Kill a Mockingbird Harper Lee 9780061120084 Fiction 85 The Catcher in the Rye J.D. Salinger 9780316769488 Fiction 156 To Kill a Mockingbird Harper Lee 9780061120084 Fiction 162 The Catcher in the Rye J.D. Salinger 9780316769488 Fiction To Kill a Mockingbird 233 Harper Lee 9780061120084 Fiction The Catcher in the Rye J.D. Salinger 9780316769488 Fiction 239 310 To Kill a Mockingbird Harper Lee 9780061120084 Fiction The Catcher in the Rye J.D. Salinger 9780316769488 Fiction 316 387 To Kill a Mockingbird Harper Lee 9780061120084 Fiction 393 The Catcher in the Rye J.D. Salinger 9780316769488 Fiction NULL NULL

1. Sort the books by Title in ascending order:

```
copy code

SELECT * FROM Books ORDER BY Title ASC;

SELECT * FROM Books ORDER BY Title ASC;

INSERT INTO Members (FirstName, LastName, Email) VALUES
('John', 'Doe', 'john.doe@example.com'),

('Jane', 'Smith', 'jane.smith@example.com'),

('Michael', 'Johnson', 'michael.johnson@example.com'),

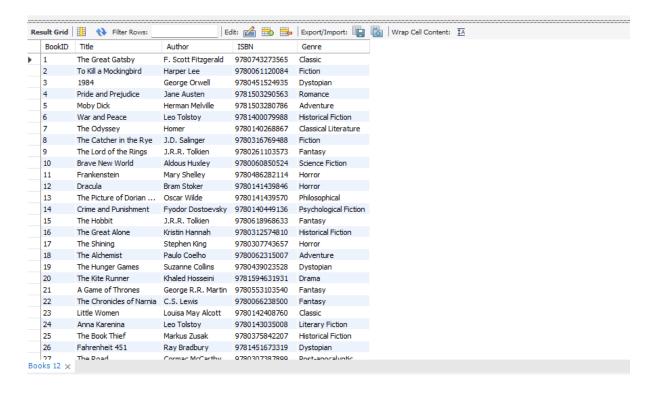
('Emily', 'Davis', 'emily.davis@example.com'),

('David', 'Brown', 'david.brown@example.com'),

('Sarah', 'Williams', 'sarah.williams@example.com'),

('James', 'Jones', 'james.jones@example.com'),
```

Result: Basic queries to retrieve and filter data from the Books table.



Additional Questions/Discussions:

- How do WHERE and ORDER BY improve the functionality of SQL queries?
 - The WHERE clause improves SQL queries by allowing users to filter and retrieve specific rows from a database table based on conditions. This helps in narrowing down large datasets and extracting only relevant information. The ORDER BY clause enhances functionality by sorting the retrieved data in ascending (ASC) or descending (DESC) order based on one or more columns, making it easier to analyze and present structured data.

Conclusions:

In this lab activity, we explored the fundamentals of SQL queries, including data retrieval using SELECT, filtering results using the WHERE clause, and sorting data using ORDER BY. These basic query operations form the foundation for interacting with relational databases, allowing users to efficiently extract and organize information based on specific requirements.