**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;

1. Retrieve books with the genre 'Fiction':

sql

Copy code

SELECT \* FROM Books WHERE Genre = 'Fiction';

1. Sort the books by Title in ascending order:

sql

Copy code

SELECT \* FROM Books ORDER BY Title ASC;

A screenshot of a computer

Description automatically generated**Result:**  
Basic queries to retrieve and filter data from the Books table.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Additional Questions/Discussions:**

* How do WHERE and ORDER BY improve the functionality of SQL queries?

The WHERE clause in SQL filters records based on specified conditions, allowing you to retrieve only the data that meets certain criteria. This improves query efficiency by narrowing down the result set. The ORDER BY clause, on the other hand, sorts the results in ascending or descending order, enhancing the readability and organization of data. Together, WHERE and ORDER BY optimize SQL queries by refining data selection and presenting it in a desired order.

**Conclusions:**

In conclusion, the WHERE and ORDER BY clauses significantly enhance the functionality of SQL queries by filtering data based on specific conditions and organizing the result set in a structured order, improving both query efficiency and data presentation.

**Laboratory Activity 4:**

**Laboratory Title:** SQL - JOIN Operation  
**Chapter No. and Topic:** Chapter 2 - Structured Query Language (SQL)  
**Discussions:**  
This activity introduces students to SQL JOIN operations for combining data from multiple tables.

**Activity Description:**  
Learn how to use INNER JOIN, LEFT JOIN, and RIGHT JOIN to combine tables.

**Objectives:**

* Write SQL JOIN queries to retrieve data from multiple tables.
* Use INNER JOIN, LEFT JOIN, and RIGHT JOIN.

**Materials:**

* MySQL Workbench or SQL client

**Procedure:**

1. Retrieve a list of all transactions, including book title and member name:

sql

Copy code

SELECT Books.Title, Members.FirstName, Members.LastName

FROM Transactions

INNER JOIN Books ON Transactions.BookID = Books.BookID

INNER JOIN Members ON Transactions.MemberID = Members.MemberID;

1. Retrieve a list of all books with transaction details, even those without transactions (LEFT JOIN):

sql

Copy code

SELECT Books.Title, Members.FirstName, Members.LastName

FROM Books

LEFT JOIN Transactions ON Books.BookID = Transactions.BookID

LEFT JOIN Members ON Transactions.MemberID = Members.MemberID;

**Result:**  
JOIN operations linking tables to retrieve combined data.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with text

Description automatically generated

**Additional Questions/Discussions:**

* How does the LEFT JOIN differ from the INNER JOIN?

The key difference between LEFT JOIN and INNER JOIN is that INNER JOIN only returns matching rows from both tables, while LEFT JOIN returns all rows from the left table and matches them with rows from the right table, filling in NULLs when no match is found.

**Conclusions:**

In conclusion, INNER JOIN is used when only matching records from both tables are needed, while LEFT JOIN is useful when all records from the left table should be included, regardless of whether there is a match in the right table, with unmatched rows displaying NULL values.