

Mini-project Part-2 - Database Implementation

1. Write SQL statements to create all tables in the final normalized ERD in Task 4.2C.
 - Use appropriate data types and required constraints for attributes.
 - Ensure all relationships and relationship constraints are maintained. Note that the order of table creation can be important when tables have Foreign Keys.
 Provide SQL statements to create all tables in your report.

<p>QUERY 1: -</p> <p>-- Create the Member table</p> <pre>CREATE TABLE Member (MemberID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Address VARCHAR(255), Email VARCHAR(100), PhoneNumber VARCHAR(20), DateOfBirth DATE, MembershipType VARCHAR(50), MembershipExpiryDate DATE);</pre>	<p>QUERY 4: -</p> <p>-- Create the LoanTransaction table</p> <pre>CREATE TABLE LoanTransaction (TransactionID INT PRIMARY KEY, LoanDate DATE, DueDate DATE, ReturnDate DATE, MemberID INT, ItemID INT, FOREIGN KEY (MemberID) REFERENCES Member(MemberID), FOREIGN KEY (ItemID) REFERENCES LibraryItem(ItemID));</pre>
<p>QUERY 2: -</p> <p>-- Create the LibraryItem table</p> <pre>CREATE TABLE LibraryItem (ItemID INT PRIMARY KEY, Title VARCHAR(100), Author VARCHAR(100), ISBN VARCHAR(20), Category VARCHAR(50), Format VARCHAR(20), PublicationYear INT, AvailabilityStatus VARCHAR(50));</pre>	<p>QUERY 5: -</p> <p>-- Create the LibraryProgram table</p> <pre>CREATE TABLE LibraryProgram (ProgramID INT PRIMARY KEY, ProgramName VARCHAR(100), Description TEXT, ProgramDate DATE, Location VARCHAR(255), AttendeeCount INT, BranchID INT, FOREIGN KEY (BranchID) REFERENCES LibraryBranch(BranchID));</pre>
<p>QUERY 3: -</p> <p>-- Create the LibraryBranch table</p> <pre>CREATE TABLE LibraryBranch (BranchID INT PRIMARY KEY, BranchName VARCHAR(100), Address VARCHAR(255), PhoneNumber VARCHAR(20), OperatingHours VARCHAR(255));</pre>	<p>QUERY 6: -</p> <p>-- Create the Staff table</p> <pre>CREATE TABLE Staff (StaffID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), PhoneNumber VARCHAR(20), BranchID INT, FOREIGN KEY (BranchID) REFERENCES LibraryBranch(BranchID));</pre>

2. Insert at least FIVE records in each table using SQL.

Provide all SQL statements used to insert data and screenshots of results of 'SELECT * FROM ;' for each table in your report.

QUERY 7: -

-- Insert data into the Member table

INSERT INTO Member (MemberID, FirstName, LastName, Address, Email, PhoneNumber, DateOfBirth, MembershipType, MembershipExpiryDate)

VALUES

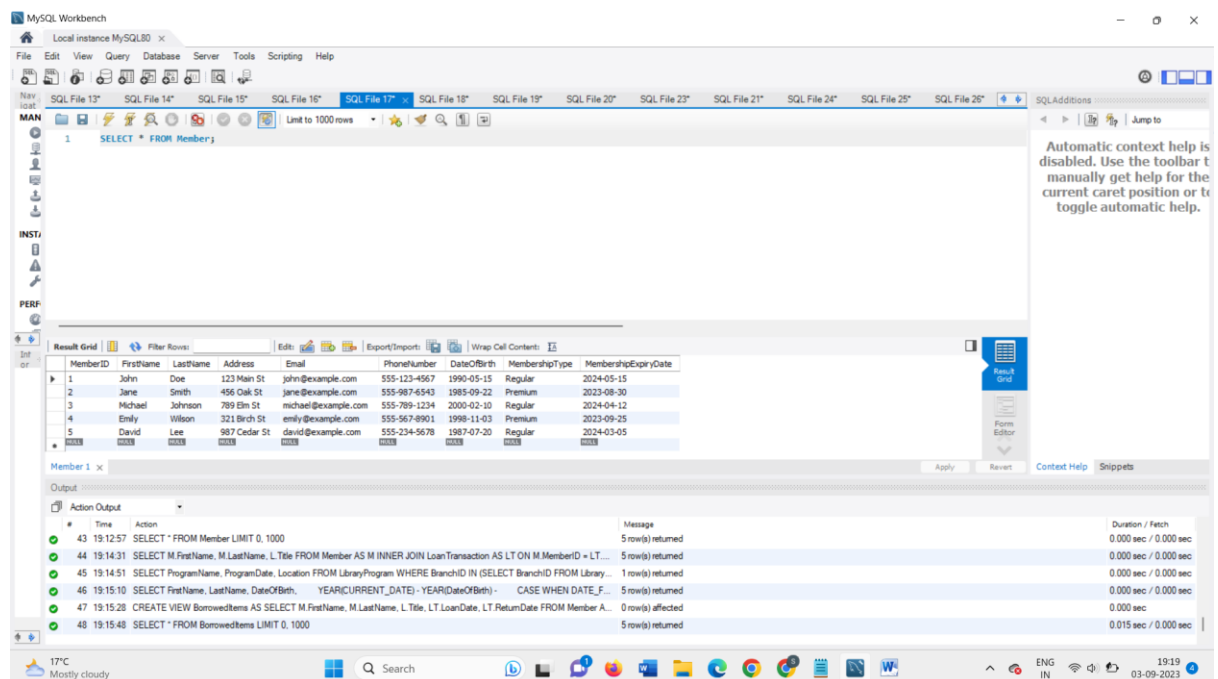
(1, 'John', 'Doe', '123 Main St', 'john@example.com', '555-123-4567', '1990-05-15', 'Regular', '2024-05-15'),

(2, 'Jane', 'Smith', '456 Oak St', 'jane@example.com', '555-987-6543', '1985-09-22', 'Premium', '2023-08-30'),

(3, 'Michael', 'Johnson', '789 Elm St', 'michael@example.com', '555-789-1234', '2000-02-10', 'Regular', '2024-04-12'),

(4, 'Emily', 'Wilson', '321 Birch St', 'emily@example.com', '555-567-8901', '1998-11-03', 'Premium', '2023-09-25'),

(5, 'David', 'Lee', '987 Cedar St', 'david@example.com', '555-234-5678', '1987-07-20', 'Regular', '2024-03-05');



QUERY 8: -

-- Insert data into the LibraryItem table

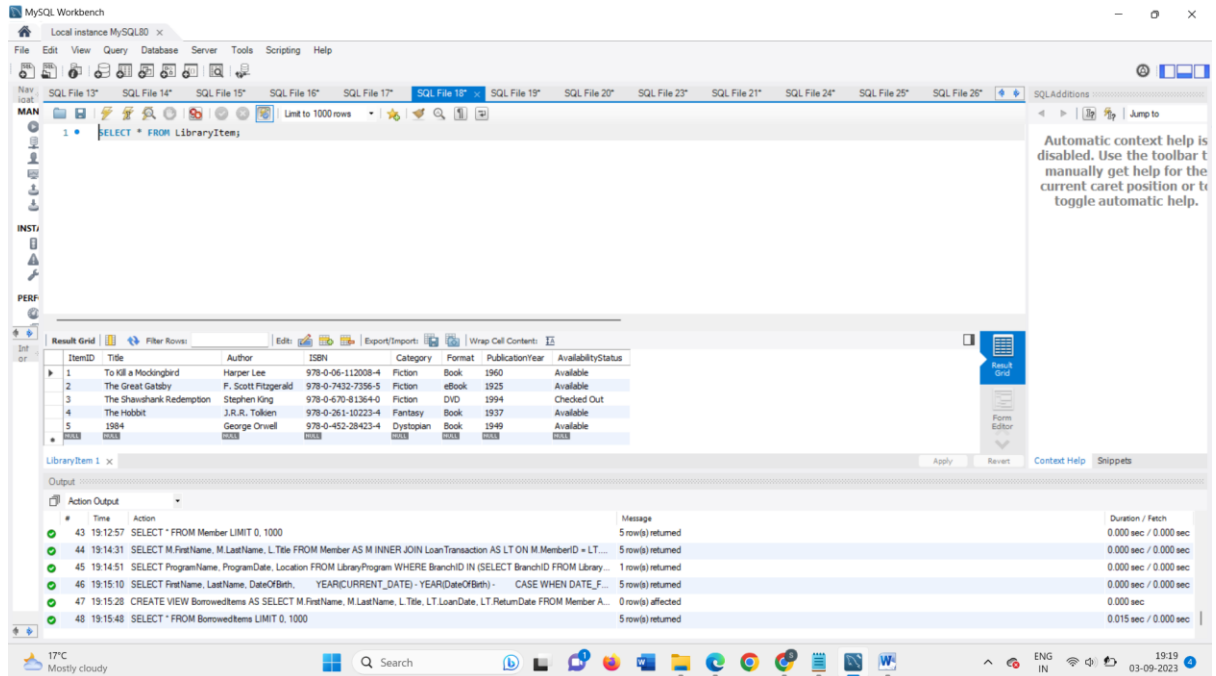
INSERT INTO LibraryItem (ItemID, Title, Author, ISBN, Category, Format, PublicationYear, AvailabilityStatus)

VALUES

(1, 'To Kill a Mockingbird', 'Harper Lee', '978-0-06-112008-4', 'Fiction', 'Book', 1960, 'Available'),

(2, 'The Great Gatsby', 'F. Scott Fitzgerald', '978-0-7432-7356-5', 'Fiction', 'eBook', 1925, 'Available'),

(3, 'The Shawshank Redemption', 'Stephen King', '978-0-670-81364-0', 'Fiction', 'DVD', 1994, 'Checked Out'),
 (4, 'The Hobbit', 'J.R.R. Tolkien', '978-0-261-10223-4', 'Fantasy', 'Book', 1937, 'Available'),
 (5, '1984', 'George Orwell', '978-0-452-28423-4', 'Dystopian', 'Book', 1949, 'Available');

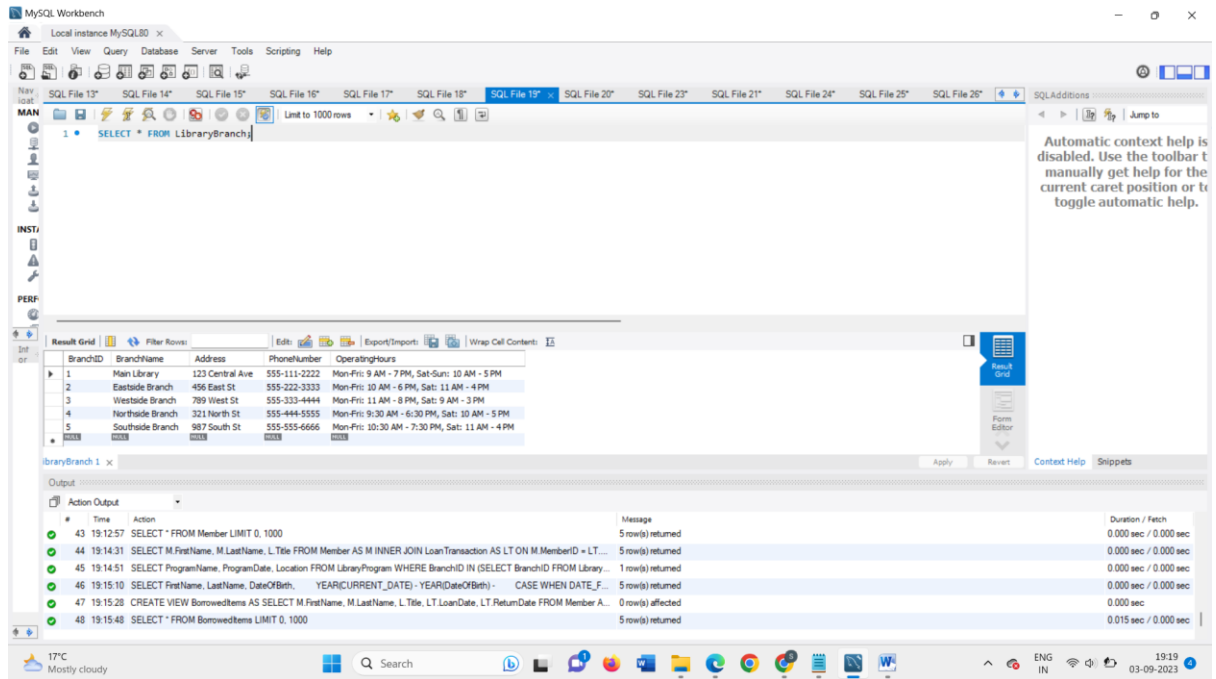


QUERY 9: -

-- Insert data into the LibraryBranch table

INSERT INTO LibraryBranch (BranchID, BranchName, Address, PhoneNumber, OperatingHours)
 VALUES

(1, 'Main Library', '123 Central Ave', '555-111-2222', 'Mon-Fri: 9 AM - 7 PM, Sat-Sun: 10 AM - 5 PM'),
 (2, 'Eastside Branch', '456 East St', '555-222-3333', 'Mon-Fri: 10 AM - 6 PM, Sat: 11 AM - 4 PM'),
 (3, 'Westside Branch', '789 West St', '555-333-4444', 'Mon-Fri: 11 AM - 8 PM, Sat: 9 AM - 3 PM'),
 (4, 'Northside Branch', '321 North St', '555-444-5555', 'Mon-Fri: 9:30 AM - 6:30 PM, Sat: 10 AM - 5 PM'),
 (5, 'Southside Branch', '987 South St', '555-555-6666', 'Mon-Fri: 10:30 AM - 7:30 PM, Sat: 11 AM - 4 PM');



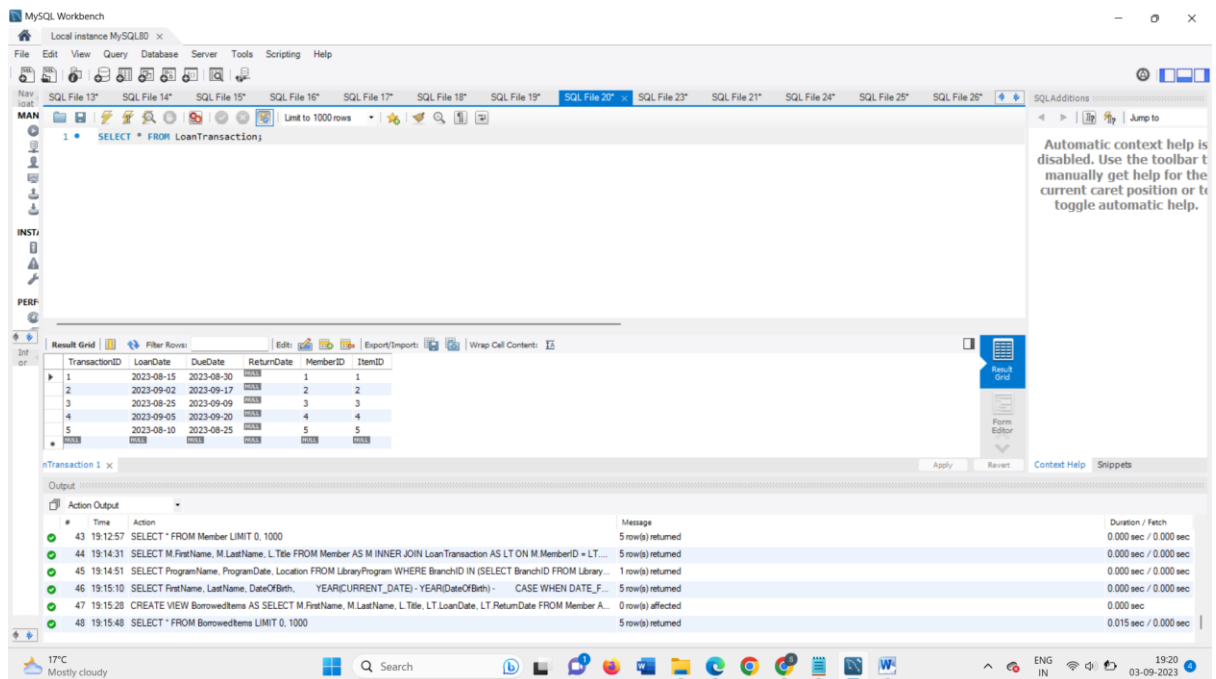
QUERY 10: -

-- Insert data into the LoanTransaction table

INSERT INTO LoanTransaction (TransactionID, LoanDate, DueDate, ReturnDate, MemberID, ItemID)

VALUES

- (1, '2023-08-15', '2023-08-30', NULL, 1, 1),
- (2, '2023-09-02', '2023-09-17', NULL, 2, 2),
- (3, '2023-08-25', '2023-09-09', NULL, 3, 3),
- (4, '2023-09-05', '2023-09-20', NULL, 4, 4),
- (5, '2023-08-10', '2023-08-25', NULL, 5, 5);



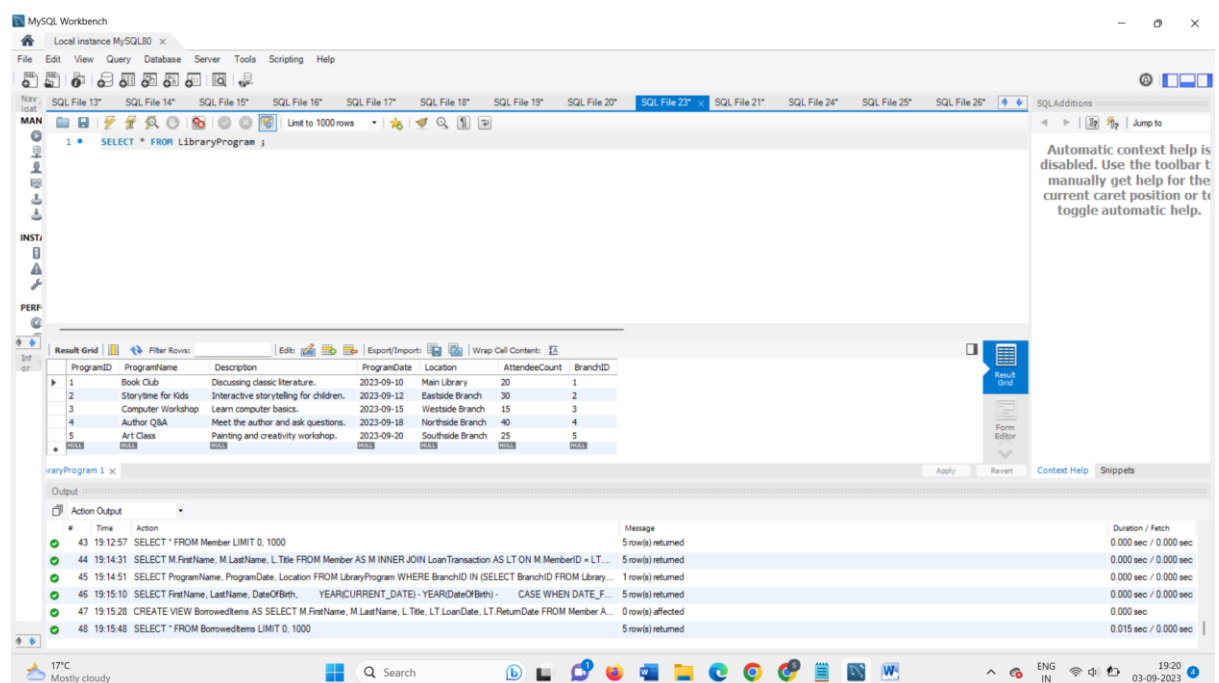
QUERY 11: -

-- Insert data into the LibraryProgram table

```
INSERT INTO LibraryProgram (ProgramID, ProgramName, Description, ProgramDate, Location, AttendeeCount, BranchID)
```

VALUES

```
(1, 'Book Club', 'Discussing classic literature.', '2023-09-10', 'Main Library', 20, 1),
(2, 'Storytime for Kids', 'Interactive storytelling for children.', '2023-09-12', 'Eastside Branch', 30, 2),
(3, 'Computer Workshop', 'Learn computer basics.', '2023-09-15', 'Westside Branch', 15, 3),
(4, 'Author Q&A', 'Meet the author and ask questions.', '2023-09-18', 'Northside Branch', 40, 4),
(5, 'Art Class', 'Painting and creativity workshop.', '2023-09-20', 'Southside Branch', 25, 5);
```



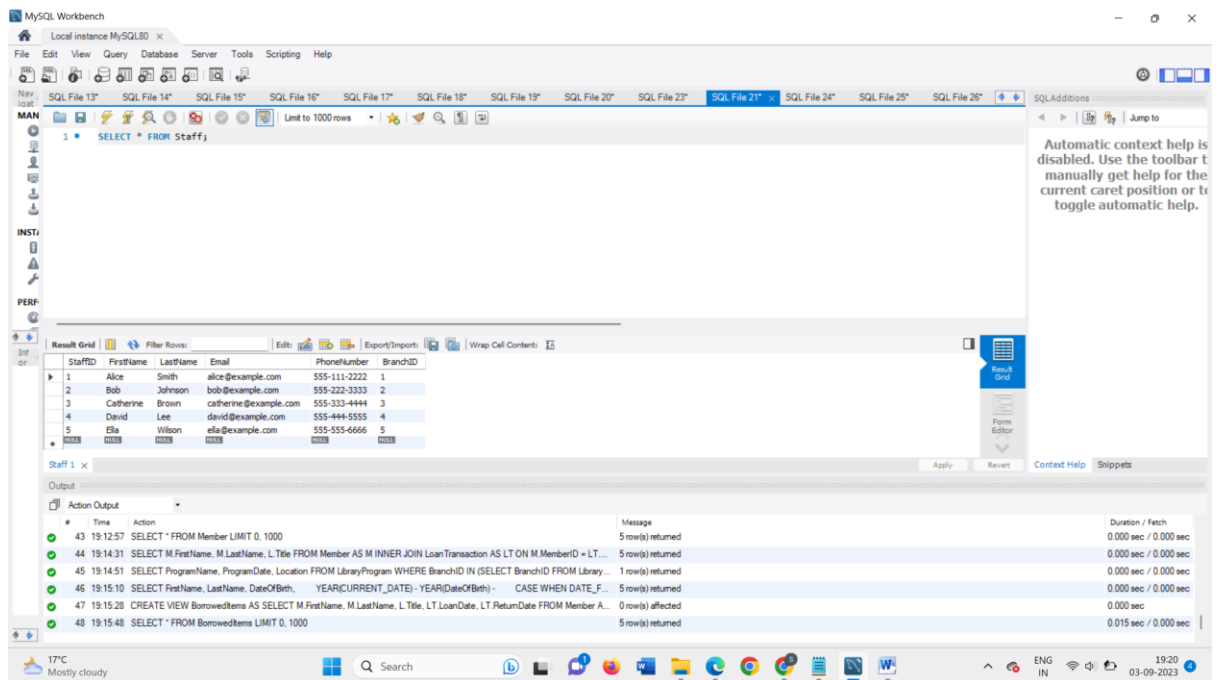
QUERY 12: -

-- Insert data into the Staff table

```
INSERT INTO Staff (StaffID, FirstName, LastName, Email, PhoneNumber, BranchID)
```

VALUES

```
(1, 'Alice', 'Smith', 'alice@example.com', '555-111-2222', 1),
(2, 'Bob', 'Johnson', 'bob@example.com', '555-222-3333', 2),
(3, 'Catherine', 'Brown', 'catherine@example.com', '555-333-4444', 3),
(4, 'David', 'Lee', 'david@example.com', '555-444-5555', 4),
(5, 'Ella', 'Wilson', 'ella@example.com', '555-555-6666', 5);
```



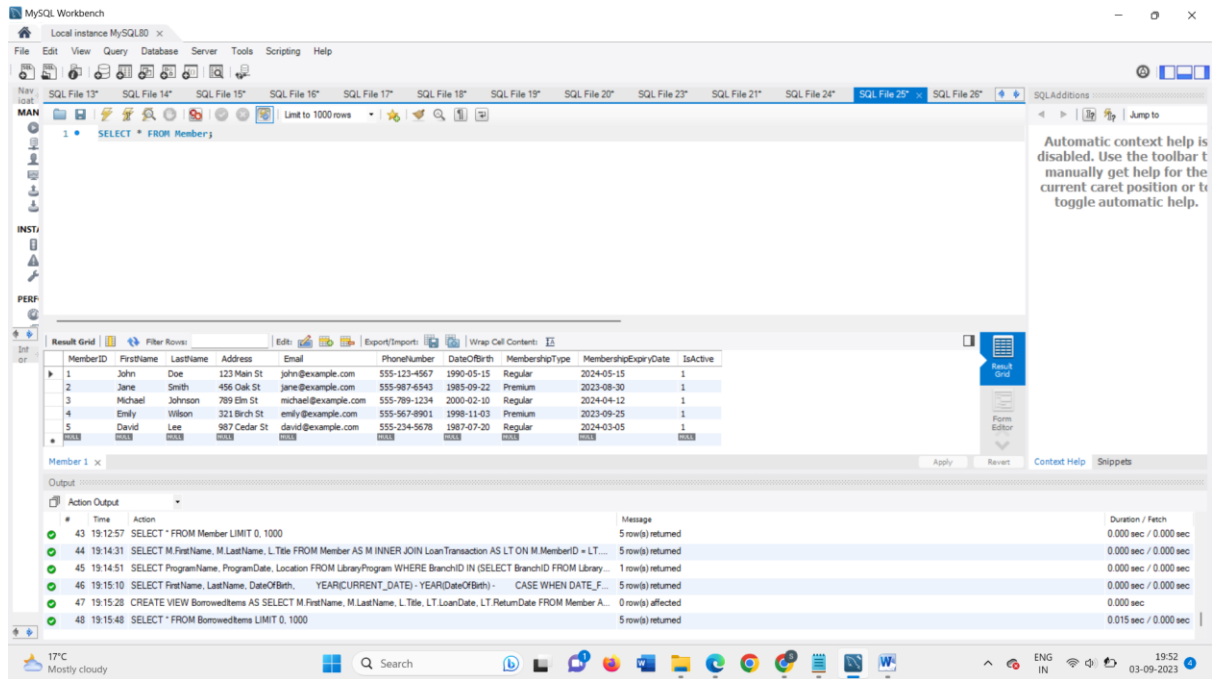
- Write and run an SQL statement to add one additional column (attribute) in any one of the existing tables with a default value. First, provide the following information:
 - Name of table you want to add a new column
 - Name, data type and default value of the new column you want to add in the table
 Then write an SQL statement to do it. Your SQL statement must do what you said you want to do in the first part of this question. When you run the SQL, it populates the default value in the newly added column for all existing records in the table.
- Provide your SQL statement

Query 13: -

ALTER TABLE Member

ADD COLUMN IsActive BOOLEAN DEFAULT TRUE;

a screenshot of records in the table using 'SELECT * FROM ' to show that existing records have default values for the newly added column in your report.



4. Write and run an SQL statement to update the default value of the newly added column to a different value for certain rows based on a condition using any other column. NOTE: You are NOT changing the values of the new column for all records; you are ONLY changing for the records that match your condition. First, provide the following information:
 - What is the condition you want to use to filter the rows? You can use any comparison operator.
 - What is the new value of the newly added column you want to set for those selected rows?
 Then, write an SQL statement to do it. Your SQL command must do what you said you want to do in the first part of this question. When you run the SQL, it changes the value of the newly added column for the records matching the condition from the default value to the new value. Provide your SQL statement

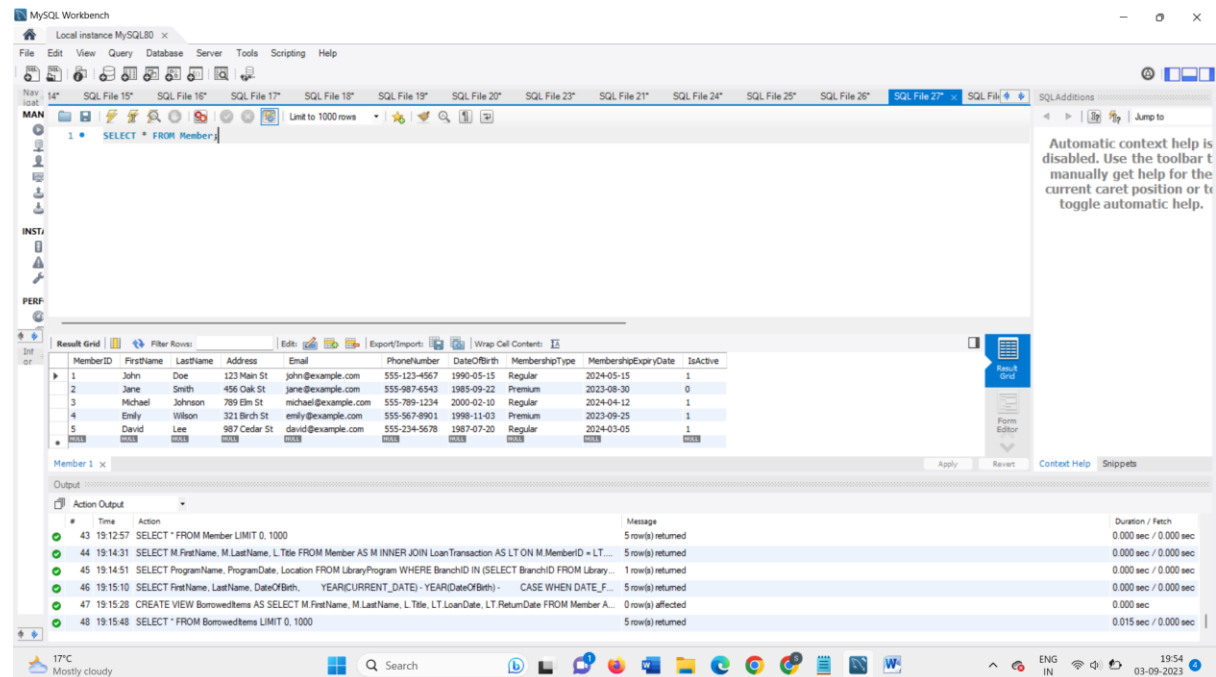
Query 14: -

```
-- Disable safe update mode for the current session
SET SQL_SAFE_UPDATES = 0;

-- UPDATE statement
UPDATE Member
SET IsActive = FALSE
WHERE MembershipExpiryDate < CURRENT_DATE;

-- Re-enable safe update mode if desired
SET SQL_SAFE_UPDATES = 1;
```

a screenshot of records in the table using 'SELECT * FROM ;' to show that the value of newly added column is successfully updated for records matching the condition in your report.

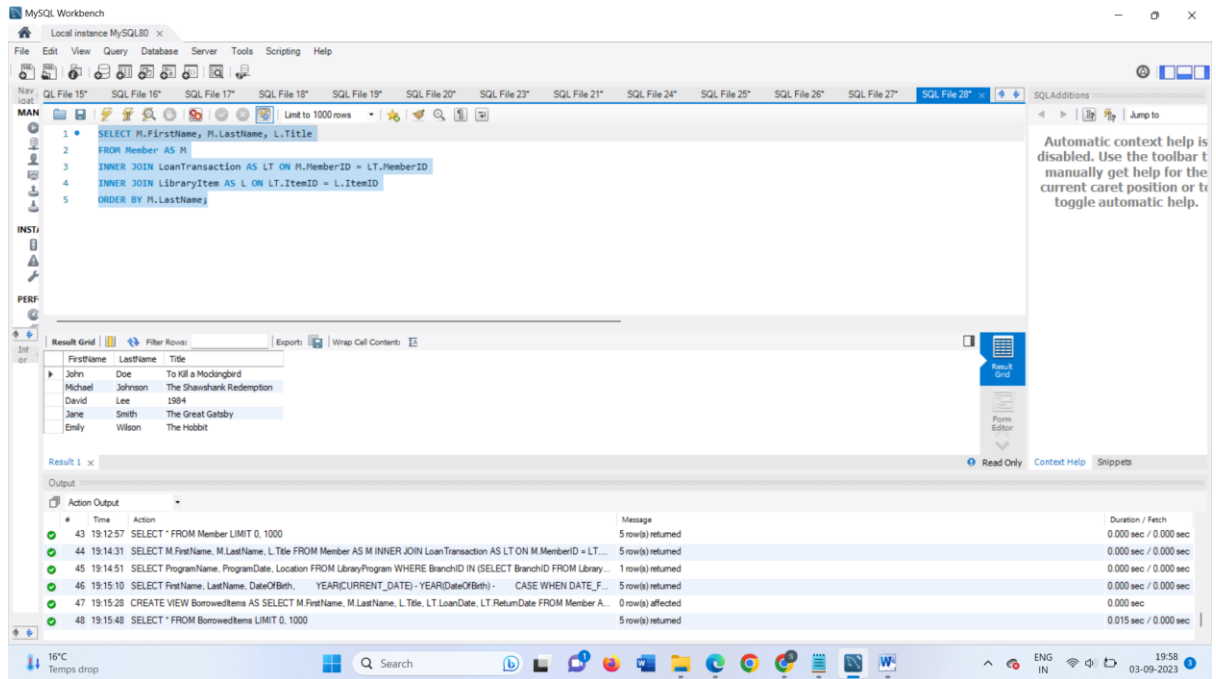


5. In this section, you are required to write SQL queries to interact with the database you implemented. Answer each SQL question in this section with the following:
 - First you provide what you want to do
 - Provide an SQL statement to do what you want to do and provide a screenshot of results after successful execution of the SQL command

- a) Write an SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY.

Query 15: -

```
SELECT M.FirstName, M.LastName, L.Title
FROM Member AS M
INNER JOIN LoanTransaction AS LT ON M.MemberID = LT.MemberID
INNER JOIN LibraryItem AS L ON LT.ItemID = L.ItemID
ORDER BY M.LastName;
```

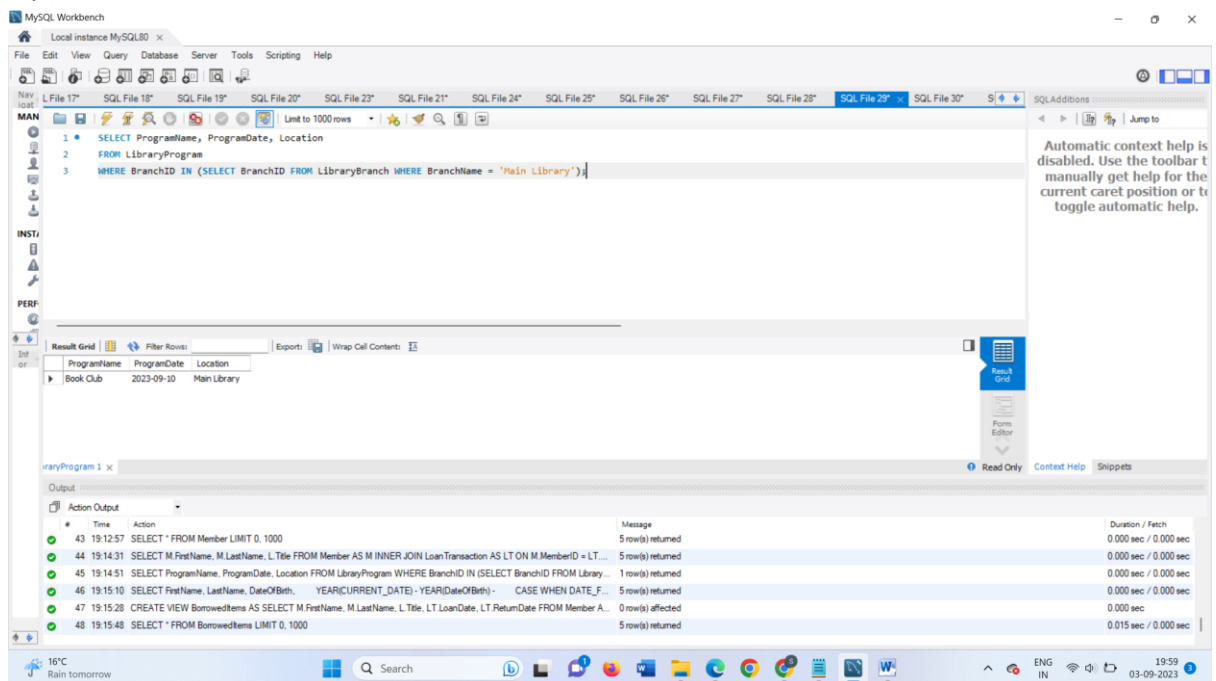



b) Write an SQL query to demonstrate the use of SELECT with WHERE and IN.

Query 16: -

SELECT ProgramName, ProgramDate, Location FROM LibraryProgram

WHERE BranchID IN (SELECT BranchID FROM LibraryBranch WHERE BranchName = 'Main Library');

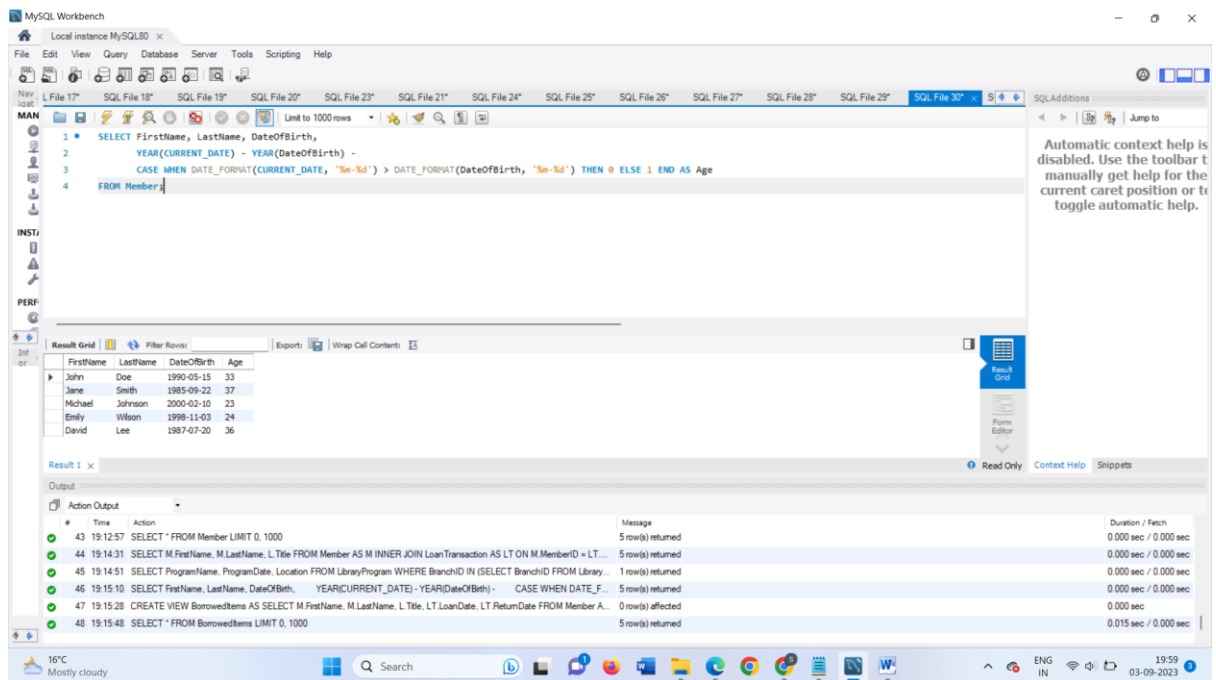


c) Write an SQL query to demonstrate the use of at least one DATE function.

Query 17:-

SELECT FirstName, LastName, DateOfBirth, YEAR(CURRENT_DATE) - YEAR(DateOfBirth) -

CASE WHEN DATE_FORMAT(CURRENT_DATE, '%m-%d') > DATE_FORMAT(DateOfBirth, '%m-%d') THEN 0 ELSE 1 END AS Age FROM Member;



- d) Write an SQL statement to create a VIEW using a SELECT statement with a JOIN. Provide the statement to create the VIEW you want and demonstrate the output of the VIEW using 'SELECT * FROM <viewname>.

Query 18: -

CREATE VIEW BorrowedItems AS

SELECT M.FirstName, M.LastName, L.Title, LT.LoanDate, LT.ReturnDate

FROM Member AS M

INNER JOIN LoanTransaction AS LT ON M.MemberID = LT.MemberID

INNER JOIN LibraryItem AS L ON LT.ItemID = L.ItemID;

