# Library Management System DataBase (DB)

# **TEAM MEMBERS:**

1.	ABDALLAH WALEED KAMAL MOUSA	ID: 20230192
2.	SAIF SAMEH FATHY ELSAWY	ID: 20230142
3.	ABDELRAHMAN FATHY ELBAHRAWEY	ID: 20230176
4.	ABDELRAHMAN ALI GHONEMI AYAD	ID: 20230174
5.	MARCO WAEL EBRAHIM	ID: 20230250

# Create Book Table:

```
1 CREATE TABLE Book (
2 Book_ID NUMBER PRIMARY KEY,
3 Title VARCHAR2(100) NOT NULL,
4 Author VARCHAR2(100),
5 ISBN VARCHAR2(20) UNIQUE,
6 Genre VARCHAR2(50),
7 Publication_Year NUMBER
8 );
```

#### Output:

**Book Table is Successfully Created** 

# Create Member Table:

```
CREATE TABLE Member (
       Member_ID NUMBER PRIMARY KEY,
Name VARCHAR2(100) NOT N
2
3
       Name
                      VARCHAR2(100) NOT NULL,
       Address VARCHAR2(200),
4
      Phone_Number VARCHAR2(20),
5
             VARCHAR2(100) UNIQUE,
       Email
6
7
       Membership_Type VARCHAR2(50)
  );
```

#### Output:

**Member Table is Successfully Created** 

Create Librarian Table:

```
CREATE TABLE Librarian (
      Librarian_ID NUMBER PRIMARY KEY,
3
                      VARCHAR2(100) NOT NULL,
      Name
      Position
4
                     VARCHAR2(50),
5
      Email
                     VARCHAR2(100) UNIQUE,
      Shift
                     VARCHAR2(20)
6
7
  );
```

Output:

**Librarian Table is Successfully Created** 

# Create Transaction Table:

```
CREATE TABLE Transaction (
Transaction_ID NUMBER PRIMARY KEY,
Book_ID NUMBER,
Member_ID NUMBER,
Issue_Date DATE NOT NULL,
Return_Date DATE,
Status VARCHAR2(20),
CONSTRAINT fk_book FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID),
CONSTRAINT fk_member FOREIGN KEY (Member_ID) REFERENCES Member(Member_ID)

One of the content of the content
```

#### **Output:**

**Transaction Table is Successfully Created** 

# INSERT INTO Book Statements and Output:

```
INSERT INTO Book VALUES (1, '1984', 'George Orwell', '9780451524935', 'Dystopian', 1949);
INSERT INTO Book VALUES (2, 'To Kill a Mockingbird', 'Harper Lee', '9780061120084', 'Classic', 1960);
INSERT INTO Book VALUES (3, 'The Great Gatsby', 'F. Scott Fitzgerald', '9780743273565', 'Novel', 1925);
INSERT INTO Book VALUES (4, 'The Hobbit', 'J.R.R. Tolkien', '9780547928227', 'Fantasy', 1937);
INSERT INTO Book VALUES (5, 'Sapiens', 'Yuval Noah Harari', '9780062316110', 'History', 2011);
```

# **Output:**

Book_ID	Title	Author	ISBN	Genre	Publication_Year
1	1984	George Orwell	9780451524935	Dystopian	1949
2	To Kill a Mockingbird	Harper Lee	9780061120084	Classic	1960
3	The Great Gatsby	F. Scott Fitzgerald	9780743273565	Novel	1925
4	The Hobbit	J.R.R. Tolkien	9780547928227	Fantasy	1937
5	Sapiens	Yuval Noah Harari	9780062316110	History	2011

# **♣** INSERT INTO Member Statements and Output:

```
INSERT INTO Member VALUES (101, 'Alice Smith', '123 Main St', '555-1234', 'alice@example.com', 'Premium');
INSERT INTO Member VALUES (102, 'Bob Johnson', '456 Park Ave', '555-5678', 'bob@example.com', 'Standard');
INSERT INTO Member VALUES (103, 'Charlie Davis', '789 Elm Rd', '555-9012', 'charlie@example.com', 'Premium');
INSERT INTO Member VALUES (104, 'Dana Lee', '321 Oak Dr', '555-3456', 'dana@example.com', 'Standard');
INSERT INTO Member VALUES (105, 'Eve Adams', '654 Pine Ln', '555-7890', 'eve@example.com', 'Premium');
```

#### > Output:

Member_ID	Name	Address	Phone_Number	Email	Membership_Type
101	Alice Smith	123 Main St	555-1234	alice@example.com	Premium
102	Bob Johnson	456 Park Ave	555-5678	bob@example.com	Standard
103	Charlie Davis	789 Elm Rd	555-9012	charlie@example.com	Premium
104	Dana Lee	321 Oak Dr	555-3456	dana@example.com	Standard

105 Eve Adams	654 Pine Ln	555-7890	eve@example.com	Premium
---------------	-------------	----------	-----------------	---------

#### **♣** INSERT INTO Librarian Statements and Output:

```
1 INSERT INTO Librarian VALUES (201, 'John Miller', 'Head Librarian', 'john@library.com', 'Morning');
2 INSERT INTO Librarian VALUES (202, 'Sarah Green', 'Assistant', 'sarah@library.com', 'Evening');
3 INSERT INTO Librarian VALUES (203, 'Mark White', 'Clerk', 'mark@library.com', 'Afternoon');
4 INSERT INTO Librarian VALUES (204, 'Laura Brown', 'Technician', 'laura@library.com', 'Wight');
5 INSERT INTO Librarian VALUES (205, 'Tom Black', 'Volunteer', 'tom@library.com', 'Weekend');
```

# Output:

Librarian_ID	Name	Position	Email	Shift
201	John Miller	Head Librarian	john@library.com	Morning
202	Sarah Green	Assistant	sarah@library.com	Evening
203	Mark White	Clerk	mark@library.com	Afternoon
204	Laura Brown	Technician	laura@library.com	Night
205	Tom Black	Volunteer	tom@library.com	Weekend

#### **♣** INSERT INTO Transaction Statements and Output:

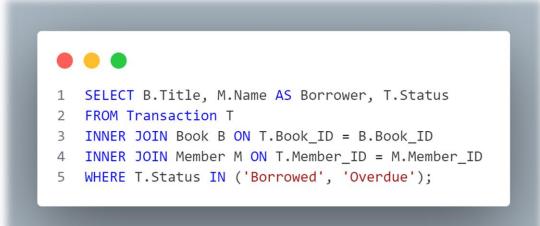
```
1 INSERT INTO Transaction VALUES (1001, 1, 101, SYSDATE-10, SYSDATE, 'Returned');
2 INSERT INTO Transaction VALUES (1002, 2, 102, SYSDATE-5, NULL, 'Borrowed');
3 INSERT INTO Transaction VALUES (1003, 3, 103, SYSDATE-8, SYSDATE-2, 'Returned');
4 INSERT INTO Transaction VALUES (1004, 4, 104, SYSDATE, NULL, 'Overdue');
5 INSERT INTO Transaction VALUES (1005, 5, 105, SYSDATE-3, SYSDATE+5, 'Returned');
```

#### Output:

Transaction_ID	Book_ID	Member_ID	Issue_Date	Return_Date	Status
1001	1	101	2025-04-01	2025-04-10	Returned
1002	2	102	2025-04-05	NULL	Borrowed
1003	3	103	2025-04-02	2025-04-12	Returned

1004	4	104	2025-04-10	NULL	Overdue
1005	5	105	025-04-07	2025-04-15	Returned

#### 1. WHERE Clause + INNER JOIN:



# Output:

Title	Borrower	Status
To Kill a Mockingbird	Bob Johnson	Borrowed
The Hobbit	Dana Lee	Overdue
1984	Alice Smith	Borrowed

#### **4** 2. GROUP BY + HAVING:

```
SELECT M.Name, COUNT(T.Transaction_ID) AS Total_Transactions
FROM Member M
JOIN Transaction T ON M.Member_ID = T.Member_ID
GROUP BY M.Name
HAVING COUNT(T.Transaction_ID) > 0;
```

#### > Output:

Name	Total_Transactions
Charlie Davis	1
Dana Lee	1
Eve Adams	1
Alice Smith	2
Bob Johnson	1

# **4** 3. INNER JOIN + Multiple Columns:

```
SELECT T.Transaction_ID, B.Title, M.Name, T.Issue_Date, T.Status
FROM Transaction T
INNER JOIN Book B ON T.Book_ID = B.Book_ID
INNER JOIN Member M ON T.Member_ID = M.Member_ID;
```

# Output:

Transaction_ID	Title	Name	Issue_Date	Status
1001	1984	Alice Smith	27-APR-25	Returned
1006	1984	Alice Smith	07-MAY-25	Borrowed
1002	To Kill a Mockingbird	Bob Johnson	02-MAY-25	Borrowed
1003	The Great Gatsby	Charlie Davis	29-APR-25	Returned
1004	The Hobbit	Dana Lee	07-MAY-25	Overdue
1005	Sapiens	Eve Adams	04-MAY-25	Returned

#### **4. LEFT JOIN + WHERE Clause:**

```
1 SELECT B.Title, T.Status
2 FROM Book B
3 LEFT JOIN Transaction T ON B.Book_ID = T.Book_ID
4 WHERE T.Status = 'Borrowed';
```

# Output:

Title	STATUS
To Kill a Mockingbird	Borrowed
1984	Borrowed

# **4** 5. Subquery:

Query: List members who borrowed books written by George Orwell.

```
SELECT DISTINCT M.Name
FROM Member M
WHERE M.Member_ID IN (
SELECT T.Member_ID
FROM Transaction T
JOIN Book B ON T.Book_ID = B.Book_ID
WHERE B.Author = 'George Orwell'
);
9
```

# Output:

Name Alice Smith

# Advanced SQL (self-study):

> Sequence:

```
CREATE SEQUENCE transaction_seq

START WITH 1006

INCREMENT BY 1

NOCACHE;

--- Example insert using sequence

--- INSERT INTO Transaction (Transaction_ID, Book_ID, Member_ID, Issue_Date, Return_Date, Status)

--- VALUES (transaction_seq.NEXTVAL, 1, 101, SYSDATE, NULL, 'Borrowed');
```

# **ERD**

