Mata Kuliah: Manajemen Basis Data
Nama: Mohammad Hanif Furqan Aufa Putra
Kelas: F NRP: 5025221161

Tugas 3: SQL Function

Exercise

Books(BookID, Title, AuthorID, PublicationYear, Genre)
Authors(AuthorID, Name, BirthYear, Nationality)
Members(MemberID, Name, MembershipStartDate, Email)
Borrowings(BorrowingID, BookID, MemberID, BorrowDate, ReturnDate)

Create a SQL function to …

1. return the number of books borrowed by member with the given MemberID
2. find books of a specific genre
3. calculate the length of time to borrow a book in days

Sebelum mengerjakan no 1-3, mari kita buat terlebih dahulu database

```sql
CREATE DATABASE library;
```

```sql
CREATE TABLE Authors (
    AuthorID INT PRIMARY KEY,
    Name VARCHAR(100),
    BirthYear INT,
    Nationality VARCHAR(100)
);

CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(255),
    AuthorID INT,
    PublicationYear INT,
    Genre VARCHAR(100),
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);
```

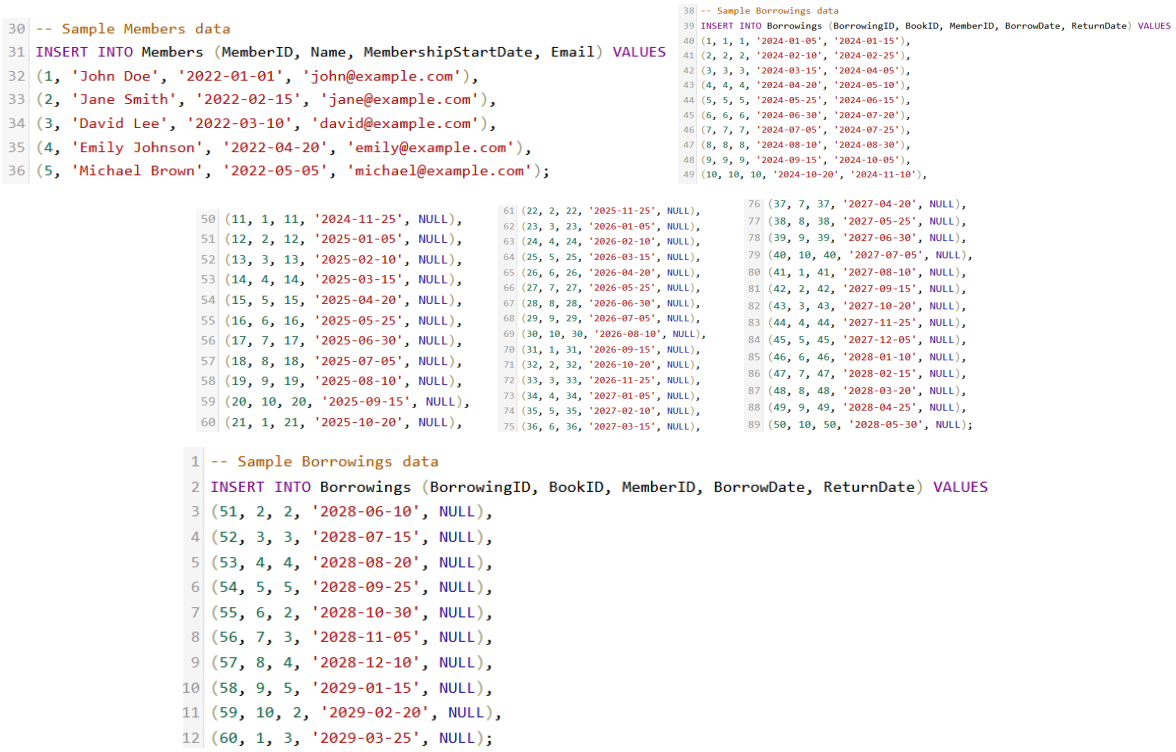
```sql
CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    Name VARCHAR(100),
    MembershipStartDate DATE,
    Email VARCHAR(255)
);

CREATE TABLE Borrowings (
    BorrowingID INT PRIMARY KEY,
    BookID INT,
    MemberID INT,
    BorrowDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```

Lalu Insert Data Sample untuk nantinya dimunculkan menggunakan fungsi SQL

```sql
-- Sample Authors data
INSERT INTO Authors (AuthorID, Name, BirthYear, Nationality) VALUES
(1, 'Jane Austen', 1775, 'British'),
(2, 'George Orwell', 1903, 'British'),
(3, 'Fyodor Dostoevsky', 1821, 'Russian'),
(4, 'Haruki Murakami', 1949, 'Japanese'),
(5, 'Gabriel Garcia Marquez', 1927, 'Colombian'),
(6, 'J.K. Rowling', 1965, 'British'),
(7, 'Tolkien', 1892, 'British'),
(8, 'Leo Tolstoy', 1828, 'Russian'),
(9, 'Ernest Hemingway', 1899, 'American'),
(10, 'Emily Bronte', 1818, 'British');
```

```sql
-- Sample Books data
INSERT INTO Books (BookID, Title, AuthorID, PublicationYear, Genre) VALUES
(1, 'Pride and Prejudice', 1, 1813, 'Romance'),
(2, '1984', 2, 1949, 'Dystopian'),
(3, 'Crime and Punishment', 3, 1866, 'Psychological Fiction'),
(4, 'Norwegian Wood', 4, 1987, 'Fiction'),
(5, 'One Hundred Years of Solitude', 5, 1967, 'Magical Realism'),
(6, 'Harry Potter and the Philosopher''s Stone', 6, 1997, 'Fantasy'),
(7, 'The Lord of the Rings', 7, 1954, 'Fantasy'),
(8, 'War and Peace', 8, 1869, 'Historical Fiction'),
(9, 'The Old Man and the Sea', 9, 1952, 'Novella'),
(10, 'Wuthering Heights', 10, 1847, 'Gothic Fiction');
```



```sql
-- Sample Members data
INSERT INTO Members (MemberID, Name, MembershipStartDate, Email) VALUES
(1, 'John Doe', '2022-01-01', 'john@example.com'),
(2, 'Jane Smith', '2022-02-15', 'jane@example.com'),
(3, 'David Lee', '2022-03-10', 'david@example.com'),
(4, 'Emily Johnson', '2022-04-20', 'emily@example.com'),
(5, 'Michael Brown', '2022-05-05', 'michael@example.com');

-- Sample Borrowings data
INSERT INTO Borrowings (BorrowingID, BookID, MemberID, BorrowDate, ReturnDate) VALUES
(1, 1, 1, '2024-01-05', '2024-01-15'),
(2, 2, 2, '2024-02-10', '2024-02-25'),
(3, 3, 3, '2024-03-15', '2024-04-05'),
(4, 4, 4, '2024-04-20', '2024-05-10'),
(5, 5, 5, '2024-05-25', '2024-06-15'),
(6, 6, 6, '2024-06-30', '2024-07-20'),
(7, 7, 7, '2024-07-05', '2024-07-25'),
(8, 8, 8, '2024-08-10', '2024-08-30'),
(9, 9, 9, '2024-09-15', '2024-10-05'),
(10, 10, 10, '2024-10-20', '2024-11-10'),
(11, 1, 11, '2024-11-25', NULL),
(12, 2, 12, '2025-01-05', NULL),
(13, 3, 13, '2025-02-10', NULL),
(14, 4, 14, '2025-03-15', NULL),
(15, 5, 15, '2025-04-20', NULL),
(16, 6, 16, '2025-05-25', NULL),
(17, 7, 17, '2025-06-30', NULL),
(18, 8, 18, '2025-07-05', NULL),
(19, 9, 19, '2025-08-10', NULL),
(20, 10, 20, '2025-09-15', NULL),
(21, 1, 21, '2025-10-20', NULL),
(22, 2, 22, '2025-11-25', NULL),
(23, 3, 23, '2026-01-05', NULL),
(24, 4, 24, '2026-02-10', NULL),
(25, 5, 25, '2026-03-15', NULL),
(26, 6, 26, '2026-04-20', NULL),
(27, 7, 27, '2026-05-25', NULL),
(28, 8, 28, '2026-06-30', NULL),
(29, 9, 29, '2026-07-05', NULL),
(30, 10, 30, '2026-08-10', NULL),
(31, 1, 31, '2026-09-15', NULL),
(32, 2, 32, '2026-10-20', NULL),
(33, 3, 33, '2026-11-25', NULL),
(34, 4, 34, '2027-01-05', NULL),
(35, 5, 35, '2027-02-10', NULL),
(36, 6, 36, '2027-03-15', NULL),
(37, 7, 37, '2027-04-20', NULL),
(38, 8, 38, '2027-05-25', NULL),
(39, 9, 39, '2027-06-30', NULL),
(40, 10, 40, '2027-07-05', NULL),
(41, 1, 41, '2027-08-10', NULL),
(42, 2, 42, '2027-09-15', NULL),
(43, 3, 43, '2027-10-20', NULL),
(44, 4, 44, '2027-11-25', NULL),
(45, 5, 45, '2027-12-05', NULL),
(46, 6, 46, '2028-01-10', NULL),
(47, 7, 47, '2028-02-15', NULL),
(48, 8, 48, '2028-03-20', NULL),
(49, 9, 49, '2028-04-25', NULL),
(50, 10, 50, '2028-05-30', NULL);
```

```sql
-- Sample Borrowings data
INSERT INTO Borrowings (BorrowingID, BookID, MemberID, BorrowDate, ReturnDate) VALUES
(51, 2, 2, '2028-06-10', NULL),
(52, 3, 3, '2028-07-15', NULL),
(53, 4, 4, '2028-08-20', NULL),
(54, 5, 5, '2028-09-25', NULL),
(55, 6, 2, '2028-10-30', NULL),
(56, 7, 3, '2028-11-05', NULL),
(57, 8, 4, '2028-12-10', NULL),
(58, 9, 5, '2029-01-15', NULL),
(59, 10, 2, '2029-02-20', NULL),
(60, 1, 3, '2029-03-25', NULL);
```

Disini saya menggunakan ‘`DELIMITER //`‘ untuk Mengubah delimiter dari `;` menjadi ‘`//`‘ agar MySQL memahami akhir pernyataan fungsi dan ‘`DELIMITER ;`’ untuk mengembalikan delimiter ke nilai default, yaitu ‘`;`’, Disini `BEGIN ... END` adalah Blok yang berisi pernyataan yang akan dieksekusi oleh fungsi. Tidak hanya itu, `END //` juga digunakan untuk menutup dari blok fungsi dengan menggunakan delimiter `//`.

1.
```sql
DELIMITER //
CREATE FUNCTION GetBorrowedBookCountForMember(MemberID INT) RETURNS INT
BEGIN
    DECLARE BookCount INT;

    SELECT COUNT(*) INTO BookCount
    FROM Borrowings
    WHERE Borrowings.MemberID = MemberID;

    RETURN BookCount;
END //
DELIMITER ;
```

a. `CREATE FUNCTION GetBorrowedBookCountForMember(MemberID INT) RETURNS INT` digunakan untuk membuat fungsi bernama GetBorrowedBookCountForMember yang menerima parameter MemberID dan mengembalikan nilai bertipe INT.
b. `DECLARE BookCount INT;`berfungsi untuk Mendeklarasikan variabel lokal bernama BookCount dengan tipe data INT.
c. `SELECT COUNT(*) INTO BookCount FROM Borrowings WHERE Borrowings.MemberID = MemberID;`digunakan untuk menghitung jumlah baris di tabel Borrowings yang memiliki MemberID sesuai dengan nilai parameter yang diberikan, dan menyimpan hasilnya ke dalam variabel BookCount.
d. `RETURN BookCount;`akan mengembalikan nilai BookCount sebagai hasil dari fungsi.

Ini adalah hasil dari fungsi ketika dipanggil

```sql
SELECT GetBorrowedBookCountForMember(1);
```

GetBorrowedBookCountForMember(1)
10

```sql
SELECT GetBorrowedBookCountForMember(3);
```

GetBorrowedBookCountForMember(3)
13

2.

```
DELIMITER //
CREATE PROCEDURE FindBooksByGenre(IN input_genre VARCHAR(100))
BEGIN
    SELECT BookID, Title, AuthorID, PublicationYear, Genre
    FROM Books
    WHERE Genre = input_genre;
END //
DELIMITER ;
```

a. `CREATE PROCEDURE FindBooksByGenre(IN input_genre VARCHAR(100))` digunakan untuk membuat prosedur bernama FindBooksByGenre yang menerima parameter input_genre dengan tipe data VARCHAR(100).

b. `SELECT BookID, Title, AuthorID, PublicationYear, Genre FROM Books WHERE Genre = input_genre;`digunakan untuk memilih kolom BookID, Title, AuthorID, PublicationYear, dan Genre dari tabel Books yang memiliki Genre sesuai dengan nilai parameter input_genre.

Ini adalah hasil dari Prosedur ketika dipanggil

```
CALL FindBooksByGenre('Romance');
```

BookID	Title	AuthorID	PublicationYear	Genre
1	Pride and Prejudice	1	1813	Romance

```
CALL FindBooksByGenre('Fantasy');
```

BookID	Title	AuthorID	PublicationYear	Genre
6	Harry Potter and the Philosopher's Stone	6	1997	Fantasy
7	The Lord of the Rings	7	1954	Fantasy

3.
```
DELIMITER //
CREATE FUNCTION CalculateBorrowingDuration(BorrowDate DATE, ReturnDate DATE) RETURNS INT
BEGIN
    DECLARE Duration INT;

    IF ReturnDate IS NOT NULL THEN
        SET Duration = DATEDIFF(ReturnDate, BorrowDate);
    ELSE
        SET Duration = DATEDIFF(CURRENT_DATE(), BorrowDate);
    END IF;

    RETURN Duration;
END //
DELIMITER ;
```

a. `CREATE FUNCTION CalculateBorrowingDuration(BorrowDate DATE, ReturnDate DATE) RETURNS INT` digunakan untuk membuat fungsi bernama CalculateBorrowingDuration yang menerima dua parameter bertipe DATE dan mengembalikan nilai bertipe INT.
b. `DECLARE Duration INT;`digunakan untuk mendeklarasikan variabel lokal bernama Duration dengan tipe data INT.
c. `IF ReturnDate IS NOT NULL THEN ... END IF;`akan menjalankan pernyataan IF ELSE untuk menentukan nilai variabel Duration tergantung pada ketersediaan ReturnDate.
d. `SET Duration = DATEDIFF(ReturnDate, BorrowDate);`akan menghitung selisih hari antara ReturnDate dan BorrowDate jika ReturnDate tidak null.
e. `SET Duration = DATEDIFF(CURRENT_DATE(), BorrowDate);` digunakan untuk menghitung selisih hari antara tanggal saat ini dan BorrowDate jika ReturnDate null.
f. `RETURN Duration;` Mengembalikan nilai Duration sebagai hasil dari fungsi.

Ini adalah hasil dari Prosedur ketika dipanggil

```
SELECT CalculateBorrowingDuration('2024-05-01', '2024-06-01');
```

CalculateBorrowingDuration('2024-05-01', '2024-06-01')
31

```
SELECT CalculateBorrowingDuration('2024-03-01', NULL);
```

CalculateBorrowingDuration('2024-03-01', NULL)
36