

HOMEWORK 6

Join

Nama : Bagus Cipta Pratama

NIM : 23/516539/PA/22097

Kelas : KOMC

Buatlah suatu database *library* yang terdiri dari tiga tabel yaitu *book*, *user*, dan *flow* dengan kolom sebagai berikut:

- Books: **bookID**, bookTitle, authorName, borrowedStatus
- Flow: **flowID**, userIDBorrowing, bookIDBorrowed, borrowDate, returnDate
- User: **userID**, userName, numberOfBorrowing, numberOfReturning

Sebelum memulai menulis query untuk mengeksplorasi join , kita harus terlebih dahulu membuat database dengan konten yang ada dan sama persis dengan yang ada diatas .

Berikut adalah query yang saya tulis :

```
mysql> create database library;
Query OK, 1 row affected (0.01 sec)

mysql> use library;
Database changed
mysql> create table books (
    -> bookid int primary key,
    -> booktitle varchar(255),
    -> authorname varchar(255),
    -> borrowedstatus enum('dipinjam','tersedia')
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table flow (  
-> flowid int primary key,  
-> useridborrowing int,  
-> bookidborrowed int,  
-> borrowdate date,  
-> returndate date,  
-> foreign key (useridborrowing) references user(userid),  
-> foreign key (bookidborrowed) references books(bookid)  
-> );
```

```
mysql> create table flow (  
-> flowid int primary key,  
-> useridborrowing int,  
-> bookidborrowed int,  
-> borrowdate date,  
-> returndate date,  
-> foreign key (useridborrowing) references user(userid),  
-> foreign key (bookidborrowed) references books(bookid)  
-> );  
Query OK, 0 rows affected (0.04 sec)
```

Kemudian , saya mengisi database tersebut dengan beberapa data sehingga kita bisa melakukan operasi join pada tabel tabel yang ada didatabase tersebut .

```

mysql> INSERT INTO Books (bookID, bookTitle, authorName, borrowedStatus)
    -> VALUES
    -> (1, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Tersedia'),
    -> (2, 'To Kill a Mockingbird', 'Harper Lee', 'Dipinjam'),
    -> (3, '1984', 'George Orwell', 'Tersedia'),
    -> (4, 'Pride and Prejudice', 'Jane Austen', 'Dipinjam'),
    -> (5, 'Moby-Dick', 'Herman Melville', 'Tersedia');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO User (userID, userName, numberOfBorrowing, numberOfReturning)
    -> VALUES
    -> (101, 'Alice', 5, 3),
    -> (102, 'Bob', 2, 2),
    -> (103, 'Charlie', 4, 3),
    -> (104, 'David', 1, 1),
    -> (105, 'Eve', 0, 0);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Flow (flowID, userIDBorrowing, bookIDBorrowed, borrowDate, returnDate)
    -> VALUES
    -> (1001, 101, 2, '2024-10-21', NULL), -- Alice meminjam 'To Kill a Mockingbird'
    -> (1002, 103, 4, '2024-10-20', NULL), -- Charlie meminjam 'Pride and Prejudice'
    -> (1003, 102, 1, '2024-10-15', '2024-10-20'), -- Bob meminjam 'The Great Gatsby', sudah dikembalikan
    -> (1004, 101, 3, '2024-10-10', '2024-10-17'); -- Alice meminjam '1984', sudah dikembalikan
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

Gunakan klausa *JOIN* untuk

- Menampilkan semua judul buku yang memiliki status dipinjam dan tanggal peminjamannya kemarin

Sebelum menulis query, saya menambahkan data dengan menulis query data baru agar hari ini Ketika saya mengerjakan mendapat hasil.

```
mysql> use library;
Database changed
mysql> INSERT INTO Flow (flowID, userIDBorrowing, bookIDBorrowed
, borrowDate, returnDate)
-> VALUES
-> (1005, 102, 5, '2024-10-22', NULL); -- Bob meminjam 'Moby
-Dick' pada tanggal 22 Oktober 2024
Query OK, 1 row affected (0.03 sec)
```

```
mysql> UPDATE Books
-> SET borrowedStatus = 'Dipinjam'
-> WHERE bookID = 5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Setelah itu , saya menuliskan query sebagai berikut dengan hasil seperti dibawah ini :

```
mysql> SELECT b.bookTitle
-> FROM Books b
-> JOIN Flow f ON b.bookID = f.bookIDBorrowed
-> WHERE b.borrowedStatus = 'Dipinjam'
-> AND f.borrowDate = CURRENT_DATE - INTERVAL 1 DAY;
+-----+
| bookTitle |
+-----+
| Moby-Dick |
+-----+
1 row in set (0.00 sec)
```

- Menampilkan semua judul buku, termasuk buku yang tidak dipinjam dan userID peminjam yang meminjam buku tersebut

```
mysql> select b.booktitle , f.useridborrowing
-> from books b
-> left join flow f on b.bookid = f.bookidborrowed;
+-----+-----+
| booktitle | useridborrowing |
+-----+-----+
| The Great Gatsby | 102 |
| To Kill a Mockingbird | 101 |
| 1984 | 101 |
| Pride and Prejudice | 103 |
| Moby-Dick | 102 |
+-----+-----+
5 rows in set (0.00 sec)
```

Disini saya menggunakan left join antara books dan flow , sehingga semua buku akan ditampilkan , termasuk yang tidak dipinjam

- Menampilkan semua buku yang dipinjam dan semua userID, baik dia meminjam atau tidak

```
mysql> select booktitle,userid
-> from user
-> right join flow on user.userid = flow.useridborrowing join
books on books.bookid = flow.bookidborrowed where books.borrow
edstatus = true;
+-----+-----+
| booktitle          | userid |
+-----+-----+
| To Kill a Mockingbird |    101 |
| Pride and Prejudice  |    103 |
| Moby-Dick           |    102 |
+-----+-----+
3 rows in set (0.01 sec)
```

- Menggunakan satu *query*, buatlah daftar semua judul buku dan nama *user* yang meminjam buku tersebut dan *user* tersebut telah meminjam lebih dari 3 buku.

```
SELECT b.bookTitle, u.userName
-> FROM Books b
-> JOIN Flow f ON b.bookID = f.bookIDBorrowed
-> JOIN User u ON f.userIDBorrowing = u.userID
-> WHERE u.numberOfBorrowing > 3;
+-----+-----+
| bookTitle          | userName |
+-----+-----+
| To Kill a Mockingbird | Alice   |
| 1984                | Alice   |
| Pride and Prejudice  | Charlie |
+-----+-----+
3 rows in set (0.00 sec)
```