Assignment 6 Join
Nama : Kosmas Rio Legowo
NIM : 23/512012/PA/21863
Buatlah suatu database *library* yang terdiri dari tiga tabel yaitu *book, user,* dan *flow* dengan kolom sebagai berikut:

```
create database if not exists library;
use library
```

Books: **bookID**, bookTitle, authorName, borrowedStatus

```
create table if not exists Books(
-> bookID int not null auto_increment,
-> bookTitle varchar(255) not null,
-> authorName varchar(255),
-> borrowedStatus boolean not null,
-> primary key(bookID)
-> );
insert into books(bookTitle, authorName, borrowedStatus)
-> values
-> ('Algorithm and Data Structures', 'Thomas H. Cormen', true),
-> ('Clean Code', 'Robert Cecil Martin', true),
-> ('Designing Data-Intensive Applications', 'Martin Kleppmann', true),
-> ('Design Patterns: Elements of Reusable Objects', 'Erich Gamma', false),
-> ('Hands-On Machine Learning with Scikit-Learn', 'Geron Aurelien', true),
-> ('The Art of Computer Programming', 'Donald Knuth', false);
```

```
mysql> select * from books;
+--------+-----------------------------------------+--------------------+----------------+
| bookID | bookTitle                               | authorName         | borrowedStatus |
+--------+-----------------------------------------+--------------------+----------------+
|      1 | Algorithm and Data Structures           | Thomas H. Cormen   |              1 |
|      2 | Clean Code                              | Robert Cecil Martin |              1 |
|      3 | Designing Data-Intensive Applications   | Martin Kleppmann   |              1 |
|      4 | Design Patterns: Elements of Reusable Objects | Erich Gamma |              0 |
|      5 | Hands-On Machine Learning with Scikit-Learn | Geron Aurelien |              1 |
|      6 | The Art of Computer Programming         | Donald Knuth       |              0 |
+--------+-----------------------------------------+--------------------+----------------+
6 rows in set (0.00 sec)
```

User: **userID**, userName, numberOfBorrowing, numberOfReturning

```
create table if not exists User(
-> userID int not null auto_increment,
-> userName varchar(255) not null,
-> numberOfBorrowing int,
-> numberOfReturning int,
-> primary key(userID)
-> );
insert into user(userName, numberOfBorrowing, numberOfReturning)
   -> values
   -> ('John', 5, 3),
```

-> ('Kenny', 2, 1),
-> ('Tom', 4, 4),
-> ('David', 10,7),
-> ('Charles', 1, 1),
-> ('Bill', 3, 0);

```
mysql> select * from user;
+--------+----------+--------------------+--------------------+
| userID | userName | numberOfBorrowing | numberOfReturning |
+--------+----------+--------------------+--------------------+
|      1 | John     |                  5 |                  3 |
|      2 | Kenny    |                  2 |                  1 |
|      3 | Tom      |                  4 |                  4 |
|      4 | David    |                 10 |                  7 |
|      5 | Charles  |                  1 |                  1 |
|      6 | Bill     |                  3 |                  0 |
+--------+----------+--------------------+--------------------+
6 rows in set (0.00 sec)
```

Flow: **flowID**, *userIDBorrowing*, *bookIDBorrowed*, borrowDate, returnDate

```
create table if not exists Flow(
-> flowID int not null auto_Increment,
-> userIDBorrowing int not null,
-> bookIDBorrowed int not null,
-> borrowDate date,
-> returnDate date,
-> primary key(flowID),
-> foreign key (userIDBorrowing) references User(userID),
-> foreign key (bookIDBorrowed) references Books(bookID)
-> );
insert into flow(userIDBorrowing, bookIDBorrowed, borrowDate, returnDate)
    -> values
    -> (1, 3, '2024-10-21', '2024-10-23'),
    -> (2, 5, '2024-10-21', '2024-10-22'),
    -> (5, 1, '2024-10-20', '2024-10-22'),
    -> (3, 2, '2024-10-20', '2024-10-23'),
    -> (4, 3, '2024-10-20', '2024-10-21');
```

```
mysql> select * from flow
    -> ;
+--------+-----------------+----------------+------------+------------+
| flowID | userIDBorrowing | bookIDBorrowed | borrowDate | returnDate |
+--------+-----------------+----------------+------------+------------+
|      1 |               1 |              3 | 2024-10-21 | 2024-10-23 |
|      2 |               2 |              5 | 2024-10-21 | 2024-10-22 |
|      3 |               5 |              1 | 2024-10-20 | 2024-10-22 |
|      4 |               3 |              2 | 2024-10-20 | 2024-10-23 |
|      5 |               4 |              3 | 2024-10-20 | 2024-10-21 |
+--------+-----------------+----------------+------------+------------+
5 rows in set (0.00 sec)
```

- Gunakan klausa *JOIN* untuk
- Menampilkan semua judul buku yang memiliki status dipinjam dan tanggal peminjamannya kemarin

```
select b.bookTitle, f.borrowDate
    -> from books b join flow f on b.bookid=f.bookidborrowed
    -> where b.borrowedstatus=1 and f.borrowdate = curdate()-interval 1 day;
```

```
mysql> select b.bookTitle, f.borrowDate
    -> from books b join flow f on b.bookid=f.bookidborrowed
    -> where b.borrowedstatus=1 and f.borrowdate = curdate()-interval 1 day;
+-------------------------------------------+------------+
| bookTitle                                 | borrowDate |
+-------------------------------------------+------------+
| Designing Data-Intensive Applications     | 2024-10-21 |
| Hands-On Machine Learning with Scikit-Learn | 2024-10-21 |
+-------------------------------------------+------------+
2 rows in set (0.00 sec)
```

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

```
select b.booktitle, f.userIDBorrowing
    -> from books b left join flow f on b.bookid = f.bookidborrowed;
```

```
mysql> select b.booktitle, f.userIDBorrowing
    -> from books b left join flow f on b.bookid = f.bookidborrowed;
+-----------------------------------------------+-----------------+
| booktitle                                     | userIDBorrowing |
+-----------------------------------------------+-----------------+
| Algorithm and Data Structures                 |               5 |
| Clean Code                                    |               3 |
| Designing Data-Intensive Applications         |               1 |
| Designing Data-Intensive Applications         |               4 |
| Design Patterns: Elements of Reusable Objects |            NULL |
| Hands-On Machine Learning with Scikit-Learn   |               2 |
| The Art of Computer Programming               |            NULL |
+-----------------------------------------------+-----------------+
7 rows in set (0.00 sec)
```

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

```
select b.booktitle, u.userid
    -> from user u left join flow f on u.userid=f.useridborrowing
    -> left join books b on f.bookidborrowed=b.bookid;
```

```
mysql> select b.booktitle, u.userid
    -> from user u left join flow f on u.userid=f.useridborrowing
    -> left join books b on f.bookidborrowed=b.bookid;
+-------------------------------------------+--------+
| booktitle                                 | userid |
+-------------------------------------------+--------+
| Designing Data-Intensive Applications     |      1 |
| Hands-On Machine Learning with Scikit-Learn |      2 |
| Clean Code                                |      3 |
| Designing Data-Intensive Applications     |      4 |
| Algorithm and Data Structures             |      5 |
| NULL                                      |      6 |
+-------------------------------------------+--------+
6 rows in set (0.00 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 user u join flow f on u.userid=f.useridborrowing
- -> join books b on f.bookidborrowed=b.bookid
- -> where u.numberofborrowing > 3;

```
mysql> select b.booktitle, u.username
    -> from user u join flow f on u.userid=f.useridborrowing
    -> join books b on f.bookidborrowed=b.bookid
    -> where u.numberofborrowing > 3;
+---------------------------------------+----------+
| booktitle                             | username |
+---------------------------------------+----------+
| Designing Data-Intensive Applications | John     |
| Clean Code                            | Tom      |
| Designing Data-Intensive Applications | David    |
+---------------------------------------+----------+
3 rows in set (0.00 sec)
```