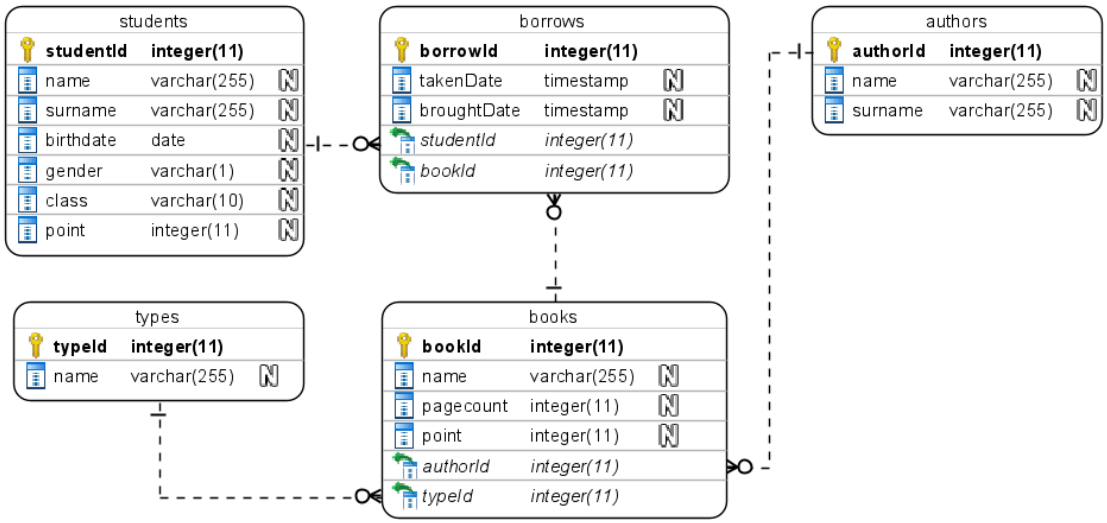
KAWAH EDUKASI
BACKEND BATCH VI

Tugas : Test Minggu Ke-2 (ERD & SQL Database)
Nama : Isep Lutpi Nur
Email : iseplutpinur7@gmail.com

1. Membuat ERD yang terdiri minimal 3 table yang saling memiliki relasi



2. Menjelaskan ERD yang sudah dibuat

ERD diatas merupakan rancangan dari database sederhana dari sebuah perpustakaan untuk peminjaman sebuah buku.

3. Buat SQL create table berdasarkan ERD yang dibuat

Dibawah ini merupakan sintaks DDL untuk membuat table sesuai dengan ERD yang telah dibuat sebelumnya

a. Tabel Student

```sql
CREATE TABLE students (
  studentId INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL,
  surname VARCHAR(255) DEFAULT NULL,
  birthday DATETIME DEFAULT NULL,
  gender ENUM('M','F') DEFAULT NULL,
  class VARCHAR(10) DEFAULT NULL,
  point integer(11) DEFAULT 0
);
```

b. Tabel Book Author

```
CREATE TABLE authors (
  authorId INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL,
  surname VARCHAR(255) DEFAULT NULL
);
```

c. Tabel Book Type

```
CREATE TABLE types (
  typeId INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL
);
```

d. Tabel Borrow

```
CREATE TABLE borrows (
  borrowId INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  takenDate DATETIME DEFAULT NULL,
  broughtDate DATETIME DEFAULT NULL,
  studentId INT(11) DEFAULT NULL,
  bookId INT(11) DEFAULT NULL
);
```

e. Tabel Book

```
CREATE TABLE books (
  bookId INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL,
  pagecount INT(11) DEFAULT 0,
  point INT(11) DEFAULT 0,
  authorId INT(11) DEFAULT NULL,
  typeId INT(11) DEFAULT NULL
);
```

f. Relasi Tabel Borrow

```
ALTER TABLE borrows
  ADD CONSTRAINT borrows_1 FOREIGN KEY (studentId) REFERENCES students
(studentId) ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT borrows_2 FOREIGN KEY (bookId) REFERENCES books
(bookId) ON DELETE SET NULL ON UPDATE CASCADE;
```

g. Relasi Tabel Book

```
ALTER TABLE books
```

```
  ADD CONSTRAINT books_1 FOREIGN KEY (authorId) REFERENCES authors
(authorId) ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT books_2 FOREIGN KEY (typeId) REFERENCES types (typeId)
ON DELETE SET NULL ON UPDATE CASCADE;
```

4. Buat SQL insert data ke dalam masing table-table.

Dibawah ini merupakan sintaks DML mengisi separuh data yang ada ke dalam masing-masing table yang sudah di buat sebelumnya.

a. Tabel Student

```
INSERT INTO students
  (studentId, name, surname, birthday, gender, class, point)
  VALUES
  (1, 'Hazel', 'Gree', '1999-05-15', 'F', '9B', 916),
  (2, 'Ashley', 'Marshall', '1999-10-28', 'F', '12D', 215);
```

b. Tabel Book Author

```
INSERT INTO authors
  (authorId, name, surname)
  VALUES
  (1, 'William Dean', 'Howells'),
  (2, 'Frederic', 'Brown');
```

c. Tabel Book Type

```
INSERT INTO types
  (typeId, name)
  VALUES
  (1, 'Science fiction'),
  (2, 'Satire');
```

d. Tabel Borrow

```
INSERT INTO borrows
  (borrowId, studentId, bookId, takenDate, broughtDate)
  VALUES
  (1, 360, 142, '2015-08-09 13:26:00.000', '2015-08-20 06:59:00.000'),
  (2, 308, 131, '2015-08-10 19:44:00.000', '2015-08-15 10:46:00.000');
```

e. Tabel Book

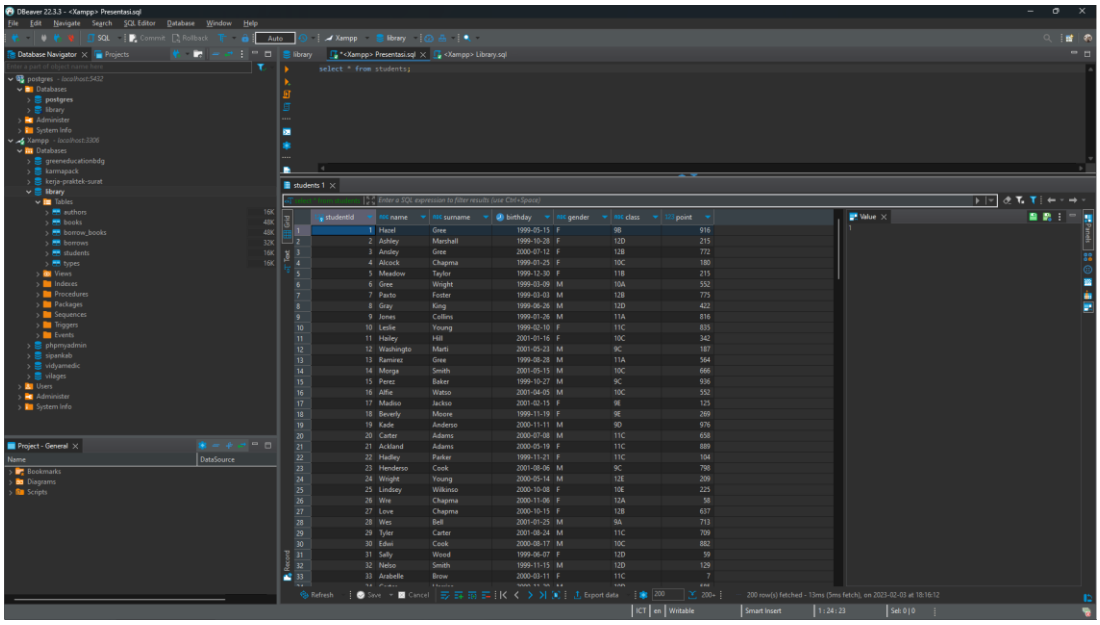
```
INSERT INTO books
  (bookId, name, pagecount, point, authorId, typeId)
  VALUES
```

```
  (1, 'A Daughter of the Snows', 199, 84, 3, 9),
  (2, 'The Near East: 10,000 Years of History', 298, 52, 9, 13);
```

5. Buat SQL select data dari satu table

Mengambil data dari table students

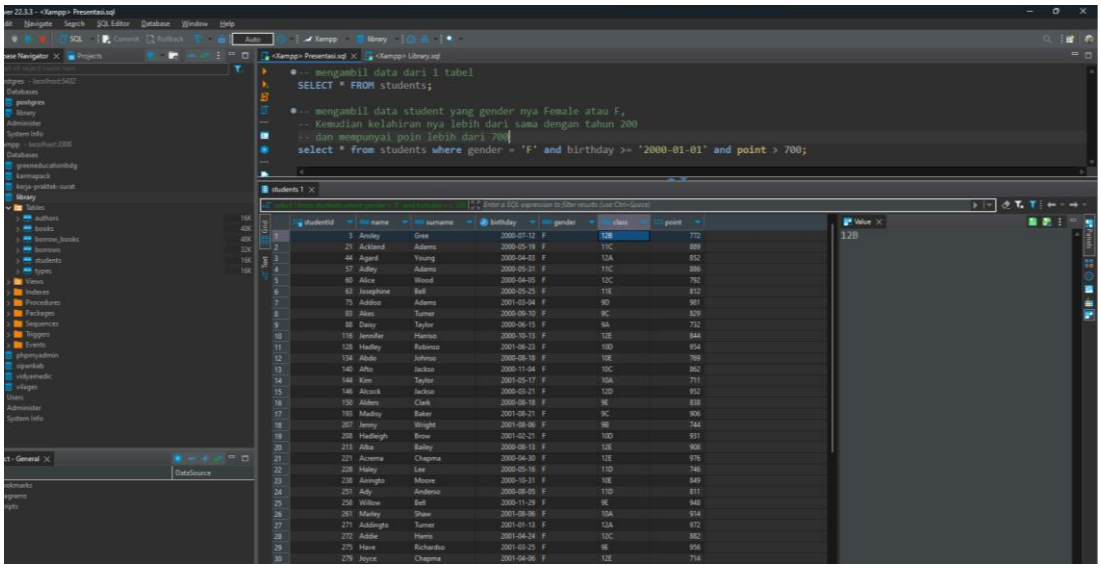
```
SELECT * FROM students;
```



6. Buat SQL select data dari satu table dengan minimal 2 kondisi dan jelaskan maksud dari SQL yang dibuat

Mengambil data student yang gender nya Female atau F, Kemudian kelahiran nya lebih dari sama dengan tahun 200 dan mempunyai poin lebih dari 700.

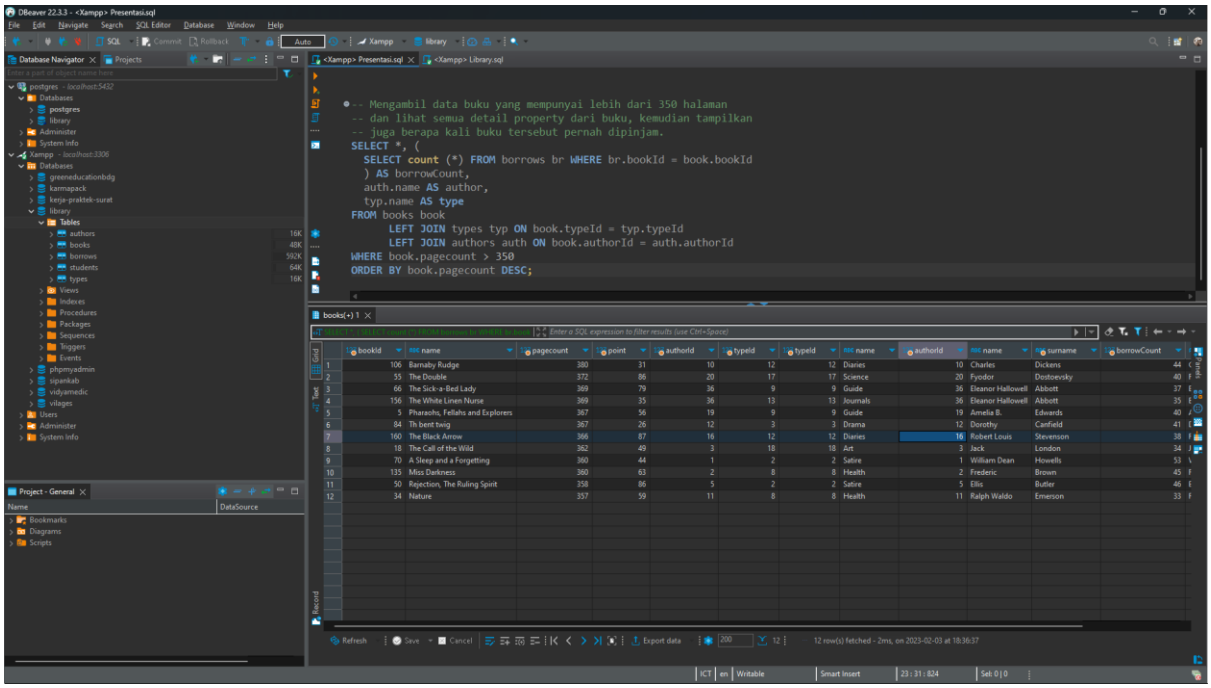
```
SELECT * FROM students WHERE gender = 'F' AND birthday >= '2000-01-01' AND
POINT > 700;
```



7. Buat SQL select yang mengimplementasikan join/sub query/cte dan jelaskan maksud dari SQL yang dibuat

Mengambil data buku yang mempunyai lebih dari 350 halaman dan lihat semua detail property dari buku, kemudian tampilkan juga berapa kali buku tersebut pernah dipinjam.

```
SELECT *, (
  SELECT count (*) FROM borrows br WHERE br.bookId = book.bookId
  ) AS borrowCount,
  auth.name AS author,
  typ.name AS type
FROM books book
    LEFT JOIN types typ ON book.typeId = typ.typeId
    LEFT JOIN authors auth ON book.authorId = auth.authorId
WHERE book.pagecount > 350
ORDER BY book.pagecount DESC;
```



Link Database SQL:
https://github.com/upi20/be_kawah_edukasi_batch_6/raw/master/MInggu%202/Library.sql