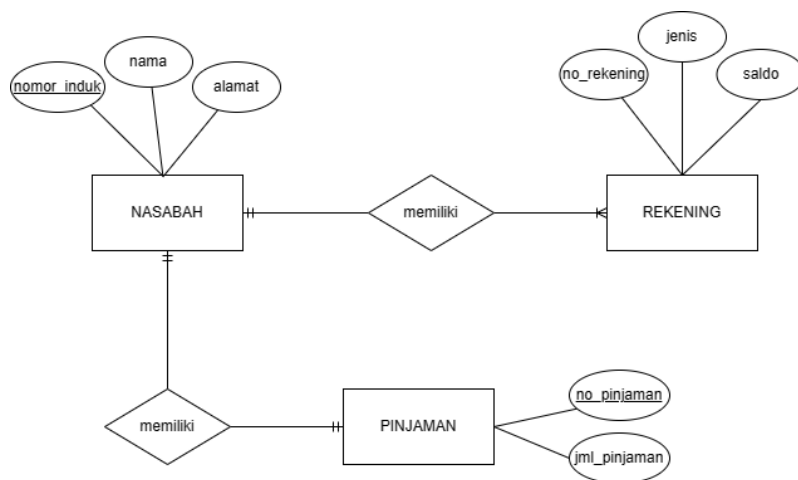


Nama : Paramadina Evita Pertiwi
NIM : 212103014
Mata Kuliah : Pemrograman Basis Data
Dosen Pengampu : Ahmad Hanafi, S.T., M.Eng.

- Sebuah sistem informasi bank terdapat entitas NASABAH (nomor_induk, nama, alamat), REKENING (no_rekening, jenis, saldo), dan PINJAMAN (no_pinjaman, jumlah pinjaman). Nasabah bisa memiliki 1 atau lebih rekening dalam bank tersebut, namun rekening hanya boleh dimiliki oleh satu nasabah. Nasabah hanya boleh memiliki hanya satu pinjaman, pinjaman hanya boleh dimiliki oleh satu nasabah. Buatlah Diagram Hubungan Entitasnya (ERD).



- Buatlah rancangan tabel fisik untuk kasus tersebut di atas.

Tabel Nasabah(Nomor_induk, Nama, Alamat)

Nama Field	Tipe Data	Panjang Data	Ket
Nomor_induk	int	4	PK
Nama	var	30	
Alamat	var	30	

Tabel Rekening(No_rekening, Jenis, Saldo)

Nama Field	Tipe Data	Panjang Data	Ket
No_rekening	int	10	PK
Jenis	var	30	
Saldo	int	4	
Nomor_induk	int	4	FK

Tabel Pinjaman(No pinjaman, Jml_pinjaman)

Nama Field	Tipe Data	Panjang Data	Ket
No pinjaman	int	24	PK
Jml pinjaman	int	30	
Nomor induk	int	4	FK

3. Dari rancangan tabel fisik tersebut, silakan lanjutkan dengan membuat database sesuai dengan rancangan tersebut. Gunakan nama database: nama belakang kalian.

```
Command Prompt - mysql.exe  X  +  v

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| akademik |
| flaskdb  |
| information_schema |
| menagerie |
| mysql    |
| paramadina |
| penggajian |
| performance_schema |
| phpmyadmin |
| test     |
+-----+
10 rows in set (0.045 sec)

MariaDB [(none)]> create database pertiwi;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| akademik |
| flaskdb  |
| information_schema |
| menagerie |
| mysql    |
| paramadina |
| penggajian |
| performance_schema |
| pertiwi  |
| phpmyadmin |
| test     |
+-----+
11 rows in set (0.001 sec)

MariaDB [(none)]> use pertiwi;
```

A. Tabel Nasabah

```
MariaDB [pertiwi]> create table nasabah(nomor_induk int(4) primary key, nama varchar(30), alamat varchar(30));
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [pertiwi]> desc nasabah;
```

Field	Type	Null	Key	Default	Extra
nomor_induk	int(4)	NO	PRI	NULL	
nama	varchar(30)	YES		NULL	
alamat	varchar(30)	YES		NULL	

3 rows in set (0.012 sec)

create table nasabah(nomor_induk int(4) primary key, nama varchar(30), alamat varchar(30));

B. Tabel Rekening

```
MariaDB [pertiwi]> create table rekening(no_rekening int(4) primary key, jenis varchar(30), saldo int(4), nomor_induk int, foreign key(nomor_induk) references nasabah(nomor_induk));
Query OK, 0 rows affected (0.019 sec)
```

```
MariaDB [pertiwi]> desc rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(4)	NO	PRI	NULL	
jenis	varchar(30)	YES		NULL	
saldo	int(4)	YES		NULL	
nomor_induk	int(11)	YES	MUL	NULL	

4 rows in set (0.013 sec)

create table rekening(no_rekening int(4) primary key, jenis varchar(30), saldo int(4), nomor_induk int, foreign key(nomor_induk) references nasabah(nomor_induk));

C. Tabel Pinjaman

```
MariaDB [pertiwi]> create table pinjaman(no_pinjaman int(4) primary key, jml_pinjaman int(30), nomor_induk int, foreign key(nomor_induk) references nasabah(nomor_induk));
```

```
Query OK, 0 rows affected (0.016 sec)
```

```
MariaDB [pertiwi]> desc pinjaman;
```

Field	Type	Null	Key	Default	Extra
no_pinjaman	int(4)	NO	PRI	NULL	
jml_pinjaman	int(30)	YES		NULL	
nomor_induk	int(11)	YES	MUL	NULL	

```
3 rows in set (0.015 sec)
```

```
create table pinjaman(no_pinjaman int(4) primary key, jml_pinjaman int(30),  
nomor_induk int, foreign key(nomor_induk) references nasabah(nomor_induk));
```

4. Isikan minimal 3 buah data pada masing-masing tabel

A. Tabel Nasabah

```
MariaDB [pertiwi]> insert into nasabah values('1','Paramadina','Yogyakarta'), ('2','Evita','Surakarta'), ('3','Dinda','Malang');
```

```
Query OK, 3 rows affected (0.004 sec)
```

```
Records: 3 Duplicates: 0 Warnings: 0
```

```
MariaDB [pertiwi]> select*from nasabah;
```

nomor_induk	nama	alamat
1	Paramadina	Yogyakarta
2	Evita	Surakarta
3	Dinda	Malang

```
3 rows in set (0.001 sec)
```

B. Tabel Rekening

```
MariaDB [pertiwi]> insert into rekening values('2222', 'Silver', 200000000, '1'),('2322', 'Gold', 100000000, '2'),('2422', 'Platinum', 150000000, '3');
Query OK, 3 rows affected (0.004 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [pertiwi]> select*from rekening;
+-----+-----+-----+-----+
| no_rekening | jenis   | saldo      | nomor_induk |
+-----+-----+-----+-----+
| 2222        | Silver  | 200000000  | 1           |
| 2322        | Gold    | 100000000  | 2           |
| 2422        | Platinum| 150000000  | 3           |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

```
insert into rekening values('2222', 'Silver', 200000000, '1'),('2322', 'Gold',
100000000, '2'),('2422', 'Platinum', 150000000, '3');
```

C. Tabel Pinjaman

```
MariaDB [pertiwi]> insert into pinjaman values('1001', 5000000, '1'),('1002'
-> , 65000000, '2'), ('1003', 68000000, '3');
Query OK, 3 rows affected (0.003 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [pertiwi]> select*from pinjaman;
+-----+-----+-----+
| no_pinjaman | jml_pinjaman | nomor_induk |
+-----+-----+-----+
| 1001        | 5000000      | 1           |
| 1002        | 65000000     | 2           |
| 1003        | 68000000     | 3           |
+-----+-----+-----+
3 rows in set (0.000 sec)
```

```
insert into pinjaman values('1001', 5000000, '1'),('1002', 65000000, '2'), ('1003',
68000000, '3');
```

5. Sajikan informasi berikut :

- a. Tampilkan nama dan alamat para nasabah

```
MariaDB [pertiwi]> SELECT nama, alamat FROM nasabah;
+-----+-----+
| nama      | alamat    |
+-----+-----+
| Paramadina| Yogyakarta|
| Evita     | Surakarta |
| Dinda     | Malang    |
+-----+-----+
3 rows in set (0.001 sec)
```

```
SELECT nama, alamat FROM nasabah;
```

- b. Tampilkan jumlah dana yang berhasil dihimpun oleh bank

```
MariaDB [pertiwi]> SELECT SUM(saldo) AS dana_terhimpun FROM rekening;
+-----+
| dana_terhimpun |
+-----+
|      450000000 |
+-----+
1 row in set (0.000 sec)
```

```
SELECT SUM(saldo) AS dana_terhimpun FROM rekening;
```

Query ini bertujuan untuk menghitung jumlah total dari kolom "saldo" dalam tabel "rekening" dan memberikan alias "dana_terhimpun" pada hasilnya.

1. Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi: Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel.
2. Fungsi agregat "SUM" digunakan untuk menjumlahkan nilai-nilai dalam kolom "saldo".
3. "saldo" adalah nama kolom dalam tabel "rekening" yang berisi nilai saldo rekening.
4. Kata kunci "AS" digunakan untuk memberikan alias pada hasil dari fungsi agregat. Dalam hal ini, hasilnya akan diberi alias "dana_terhimpun".
5. "dana_terhimpun" adalah alias yang diberikan pada hasil jumlah saldo yang terkumpul.
6. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query.

Dalam hal ini, tabel yang digunakan adalah "rekening". Jadi, query ini akan menghitung jumlah total dari kolom "saldo" dalam tabel "rekening" dan hasilnya akan ditampilkan dengan nama alias "dana_terhimpun".

- c. Tampilkan jumlah dana yang dipinjamkan oleh bank

```
MariaDB [pertiwi]> SELECT SUM(jml_pinjaman) AS dana_dipinjam FROM pinjaman;
+-----+
| dana_dipinjam |
+-----+
|      138000000 |
+-----+
1 row in set (0.001 sec)
```

```
SELECT SUM(jml_pinjaman) AS dana_dipinjam FROM pinjaman;
```

- d. Tampilkan siapa nama pemilik rekening yang memiliki nilai saldo terbesar!

```
MariaDB [pertiwi]> SELECT nasabah.nama FROM nasabah JOIN rekening ON nasabah.nomor_induk=rekening.nomor_induk WHERE rekening.saldo=(SELECT MAX(saldo) FROM rekening);
+-----+
| nama      |
+-----+
| Paramadina |
+-----+
1 row in set (0.001 sec)
```

```
SELECT nasabah.nama FROM nasabah JOIN rekening ON
nasabah.nomor_induk=rekening.nomor_induk WHERE rekening.saldo=(SELECT
MAX(saldo) FROM rekening);
```

Query ini bertujuan untuk mengambil nama nasabah yang memiliki saldo tertinggi dalam tabel "rekening" dan menghubungkannya dengan tabel "nasabah" menggunakan kolom "nomor_induk".

Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "nama" dari tabel "nasabah".
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "nasabah".
3. Kata kunci "JOIN" digunakan untuk menggabungkan (join) tabel "nasabah" dengan tabel "rekening" berdasarkan kondisi yang ditentukan.
4. "nasabah.nomor_induk" dan "rekening.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.
5. Kata kunci "WHERE" digunakan untuk menentukan kriteria filter untuk membatasi hasil yang dihasilkan. Dalam hal ini, kita membatasi hasil hanya pada baris yang memiliki saldo yang sama dengan nilai maksimum dalam kolom "saldo" dari tabel "rekening".
6. "rekening.saldo" adalah kolom yang digunakan untuk memeriksa kondisi kesetaraan dengan nilai maksimum.
7. Kata kunci "SELECT MAX(saldo) FROM rekening" digunakan untuk mengambil nilai maksimum dari kolom "saldo" dalam tabel "rekening" menggunakan fungsi agregat "MAX".

Jadi, query ini akan mengambil nama nasabah yang memiliki saldo tertinggi dalam tabel "rekening" dengan menggabungkan tabel "nasabah" dan "rekening" berdasarkan kolom "nomor_induk" dan membandingkan saldo dengan nilai maksimum yang diambil dari tabel "rekening".

```

MariaDB [pertiwi]> SELECT nasabah.nama, rekening.saldo FROM nasabah JOIN rekening ON
nasabah.nomor_induk=rekening.nomor_induk WHERE rekening.saldo=(SELECT MAX(saldo) FROM
rekening);
+-----+-----+
| nama      | saldo      |
+-----+-----+
| Paramadina | 2000000000 |
+-----+-----+
1 row in set (0.001 sec)

```

```

SELECT nasabah.nama, rekening.saldo FROM nasabah JOIN rekening ON
nasabah.nomor_induk=rekening.nomor_induk WHERE rekening.saldo=(SELECT
MAX(saldo) FROM rekening);

```

Query ini bertujuan untuk mengambil nama nasabah dan saldo dari nasabah yang memiliki saldo tertinggi dalam tabel "rekening" dan menghubungkannya dengan tabel "nasabah" menggunakan kolom "nomor_induk".

Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "nama" dari tabel "nasabah" dan kolom "saldo" dari tabel "rekening".
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "nasabah".
3. Kata kunci "JOIN" digunakan untuk menggabungkan (join) tabel "nasabah" dengan tabel "rekening" berdasarkan kondisi yang ditentukan.
4. "nasabah.nomor_induk" dan "rekening.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.
5. Kata kunci "WHERE" digunakan untuk menentukan kriteria filter untuk membatasi hasil yang dihasilkan. Dalam hal ini, kita membatasi hasil hanya pada baris yang memiliki saldo yang sama dengan nilai maksimum dalam kolom "saldo" dari tabel "rekening".
6. "rekening.saldo" adalah kolom yang digunakan untuk memeriksa kondisi kesetaraan dengan nilai maksimum.
7. Kata kunci "SELECT MAX(saldo) FROM rekening" digunakan untuk mengambil nilai maksimum dari kolom "saldo" dalam tabel "rekening" menggunakan fungsi agregat "MAX".

Jadi, query ini akan mengambil nama nasabah dan saldo dari nasabah yang memiliki saldo tertinggi dalam tabel "rekening" dengan menggabungkan tabel "nasabah" dan "rekening" berdasarkan kolom "nomor_induk" dan membandingkan saldo dengan nilai maksimum yang diambil dari tabel "rekening".

- e. Tampilkan jumlah terkecil dari pinjaman!

```
MariaDB [pertiwi]> SELECT MIN(jml_pinjaman) AS minimal_pinjaman FROM pinjaman;
+-----+
| minimal_pinjaman |
+-----+
|          5000000 |
+-----+
1 row in set (0.001 sec)
```

```
SELECT MIN(jml_pinjaman) AS minimal_pinjaman FROM pinjaman;
```

- f. Tampilkan jumlah dari rekening yang dimiliki masing-masing nasabah

```
MariaDB [pertiwi]> SELECT nasabah.nama, COUNT(rekening.no_rekening) AS jumlah_rekening
FROM nasabah JOIN rekening ON nasabah.nomor_induk=rekening.nomor_induk GROUP BY nasabah.nama;
+-----+-----+
| nama      | jumlah_rekening |
+-----+-----+
| Dinda     | 1               |
| Evita     | 1               |
| Paramadina | 1               |
+-----+-----+
3 rows in set (0.001 sec)
```

```
SELECT nasabah.nama, COUNT(rekening.no_rekening) AS jumlah_rekening FROM
nasabah JOIN rekening ON nasabah.nomor_induk=rekening.nomor_induk GROUP BY
nasabah.nama;
```

Query ini bertujuan untuk mengambil nama nasabah dan menghitung jumlah rekening yang dimiliki oleh setiap nasabah dalam tabel "nasabah" dan "rekening" yang dihubungkan menggunakan kolom "nomor_induk". Hasilnya akan dikelompokkan berdasarkan nama nasabah menggunakan klausa "GROUP BY".

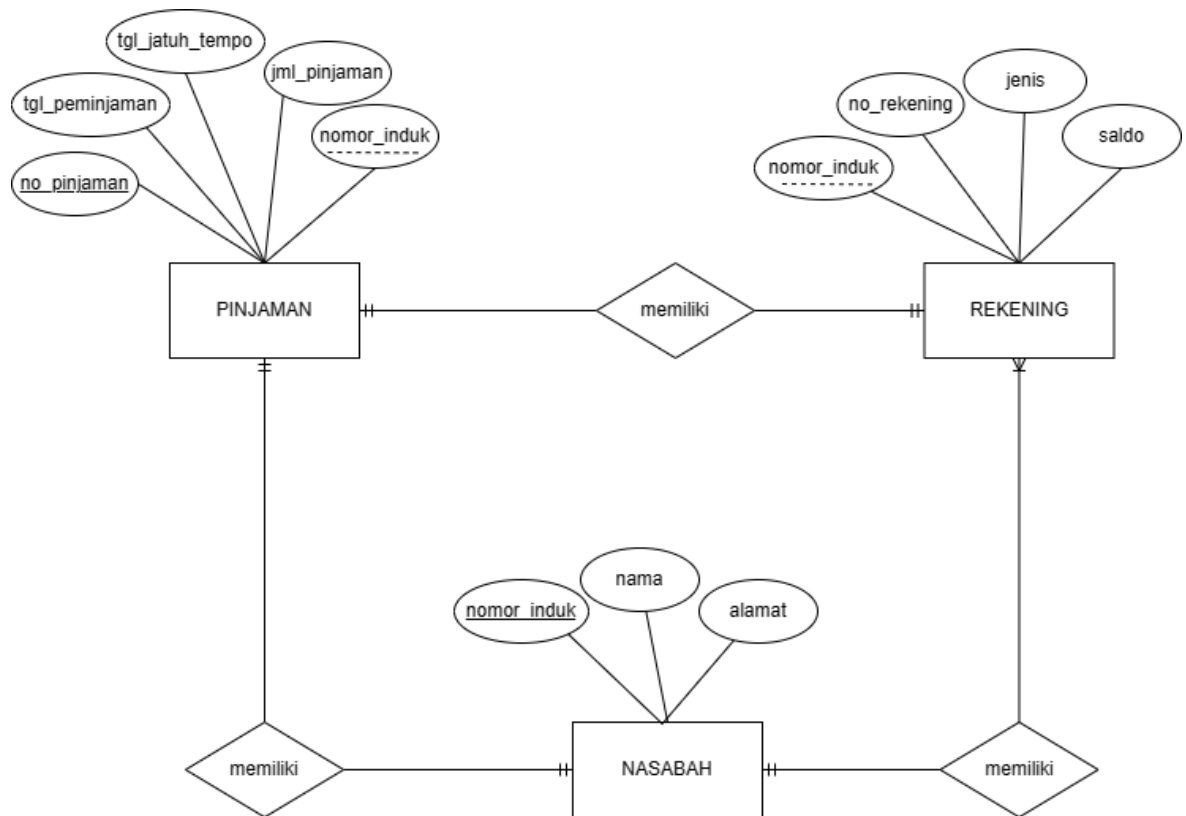
Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "nama" dari tabel "nasabah" dan menggunakan fungsi agregat "COUNT" untuk menghitung jumlah rekening.
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "nasabah".
3. Kata kunci "JOIN" digunakan untuk menggabungkan (join) tabel "nasabah" dengan tabel "rekening" berdasarkan kondisi yang ditentukan.
4. "nasabah.nomor_induk" dan "rekening.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.
5. Kata kunci "GROUP BY" digunakan untuk mengelompokkan hasil berdasarkan kolom yang ditentukan, dalam hal ini, "nasabah.nama".
6. Fungsi agregat "COUNT(rekening.no_rekening)" digunakan untuk menghitung jumlah rekening dengan menghitung jumlah baris yang memiliki nilai tidak null pada kolom "no_rekening" dalam tabel "rekening".

7. Alias "jumlah_rekening" diberikan pada hasil fungsi agregat "COUNT(rekening.no_rekening)" untuk memberikan nama yang lebih deskriptif pada kolom tersebut.

Jadi, query ini akan mengambil nama nasabah dan menghitung jumlah rekening yang dimiliki oleh setiap nasabah dalam tabel "nasabah" dan "rekening" dengan menggabungkan tabel berdasarkan kolom "nomor_induk" dan mengelompokkan hasil berdasarkan nama nasabah.

6. Terdapat update proses bisnis demikian :
- setiap peminjam harus memiliki rekening di bank tersebut, rancangan ERD apakah yang harus disesuaikan? (petunjuk: ingat tentang FK)



- b. peminjaman memiliki tanggal peminjaman dan tanggal jatuh tempo.

```
MariaDB [pertiwi]> SELECT nasabah.nama AS nama_peminjam, pinjaman.tgl_pinjaman, pinjaman.tgl_jatuh_tempo FROM pinjaman INNER JOIN nasabah ON pinjaman.nomor_induk=nasabah.nomor_induk;
```

nama_peminjam	tgl_pinjaman	tgl_jatuh_tempo
Paramadina	2022-11-20	2023-01-20
Evita	2023-02-24	2023-03-24
Dinda	2023-04-25	2023-05-25

3 rows in set (0.001 sec)

```
SELECT nasabah.nama AS nama_peminjam,  
pinjaman.tgl_pinjaman, pinjaman.tgl_jatuh_tempo FROM pinjaman INNER JOIN nasabah  
ON pinjaman.nomor_induk=nasabah.nomor_induk;
```

Query ini bertujuan untuk mengambil nama peminjam, tanggal pinjaman, dan tanggal jatuh tempo dari tabel "pinjaman" dengan menggabungkannya dengan tabel "nasabah" menggunakan kolom "nomor_induk".

Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "nama" dari tabel "nasabah" dan kolom "tgl_pinjaman" serta "tgl_jatuh_tempo" dari tabel "pinjaman".
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "pinjaman".
3. Kata kunci "INNER JOIN" digunakan untuk menggabungkan (join) tabel "pinjaman" dengan tabel "nasabah" berdasarkan kondisi yang ditentukan.
4. "pinjaman.nomor_induk" dan "nasabah.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.
5. Alias "nama_peminjam" diberikan pada kolom "nasabah.nama" untuk memberikan nama yang lebih deskriptif pada kolom tersebut.

Jadi, query ini akan mengambil nama peminjam, tanggal pinjaman, dan tanggal jatuh tempo dari tabel "pinjaman" dengan menggabungkannya dengan tabel "nasabah" berdasarkan kolom "nomor_induk". Alias "nama_peminjam" digunakan untuk memberikan nama yang lebih deskriptif pada kolom "nasabah.nama".

```
MariaDB [pertiwi]> SELECT pinjaman.no_pinjaman, pinjaman.jml_pinjaman, pinjaman.tgl_pinjaman AS tanggal_peminjaman, pinjaman.tgl_jatuh_tempo AS tanggal_jatuh_tempo, nasabah.nama AS nama_nasabah FROM pinjaman INNER JOIN nasabah ON pinjaman.nomor_induk=nasabah.nomor_induk;
```

no_pinjaman	jml_pinjaman	tanggal_peminjaman	tanggal_jatuh_tempo	nama_nasabah
1001	5000000	2022-11-20	2023-01-20	Paramadina
1002	65000000	2023-02-24	2023-03-24	Evita
1003	68000000	2023-04-25	2023-05-25	Dinda

3 rows in set (0.001 sec)

```
SELECT pinjaman.no_pinjaman, pinjaman.jml_pinjaman, pinjaman.tgl_pinjaman AS tanggal_peminjaman, pinjaman.tgl_jatuh_tempo AS tanggal_jatuh_tempo, nasabah.nama AS nama_nasabah FROM pinjaman INNER JOIN nasabah ON pinjaman.nomor_induk=nasabah.nomor_induk;
```

Query ini bertujuan untuk mengambil nomor pinjaman, jumlah pinjaman, tanggal pinjaman (dengan alias "tanggal_peminjaman"), tanggal jatuh tempo (dengan alias "tanggal_jatuh_tempo"), dan nama nasabah (dengan alias "nama_nasabah") dari tabel "pinjaman" dengan menggabungkannya dengan tabel "nasabah" menggunakan kolom "nomor_induk".

Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "no_pinjaman", "jml_pinjaman", "tgl_pinjaman" (dengan alias "tanggal_peminjaman"), "tgl_jatuh_tempo" (dengan alias "tanggal_jatuh_tempo"), dan "nama" (dengan alias "nama_nasabah").
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "pinjaman".
3. Kata kunci "INNER JOIN" digunakan untuk menggabungkan (join) tabel "pinjaman" dengan tabel "nasabah" berdasarkan kondisi yang ditentukan.
4. "pinjaman.nomor_induk" dan "nasabah.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.
5. Alias "tanggal_peminjaman" diberikan pada kolom "tgl_pinjaman" untuk memberikan nama yang lebih deskriptif pada kolom tersebut.
6. Alias "tanggal_jatuh_tempo" diberikan pada kolom "tgl_jatuh_tempo" untuk memberikan nama yang lebih deskriptif pada kolom tersebut.
7. Alias "nama_nasabah" diberikan pada kolom "nasabah.nama" untuk memberikan nama yang lebih deskriptif pada kolom tersebut.

Jadi, query ini akan mengambil nomor pinjaman, jumlah pinjaman, tanggal pinjaman (dengan alias "tanggal_peminjaman"), tanggal jatuh tempo (dengan alias "tanggal_jatuh_tempo"), dan nama nasabah (dengan alias "nama_nasabah") dari tabel "pinjaman" dengan menggabungkannya dengan tabel "nasabah" berdasarkan kolom "nomor_induk".

- c. setiap peminjam diperbolehkan meminjam selama memiliki rekening(hint: gunakan join)

```
MariaDB [pertiwi]> SELECT nasabah.nama
-> FROM nasabah
-> JOIN rekening ON nasabah.nomor_induk = rekening.nomor_induk;

+-----+
| nama   |
+-----+
| Paramadina |
| Evita    |
| Dinda    |
+-----+
3 rows in set (0.001 sec)
```

SELECT nasabah.nama FROM nasabah JOIN rekening ON nasabah.nomor_induk = rekening.nomor_induk;

Query ini bertujuan untuk mengambil nama-nama nasabah dari tabel "nasabah" dengan melakukan operasi join dengan tabel "rekening" menggunakan kolom "nomor_induk".

Berikut adalah penjelasan langkah demi langkah tentang bagaimana query ini berfungsi:

1. Kata kunci "SELECT" digunakan untuk memilih kolom atau ekspresi yang akan diambil dari tabel. Dalam hal ini, kita memilih kolom "nama" dari tabel "nasabah".
2. Kata kunci "FROM" digunakan untuk menentukan tabel mana yang akan digunakan dalam query. Dalam hal ini, kita menggunakan tabel "nasabah".
3. Kata kunci "JOIN" digunakan untuk menggabungkan (join) tabel "nasabah" dengan tabel "rekening" berdasarkan kondisi yang ditentukan.
4. "nasabah.nomor_induk" dan "rekening.nomor_induk" adalah kolom yang digunakan sebagai kunci untuk menggabungkan kedua tabel.

Jadi, query ini akan mengambil nama-nama nasabah dari tabel "nasabah" yang memiliki kesamaan pada kolom "nomor_induk" dengan tabel "rekening". Dengan melakukan operasi join ini, Anda akan mendapatkan hasil yang memasangkan nasabah dengan rekening yang memiliki nomor induk yang sama.

- Lakukan penyesuaian tersebut dalam rancangan tabel fisik, sesuaikan database anda, lalu capture tiap tahap. (ingat tentang mengubah constrain dan membuat table, boleh menggunakan data baru/menghapus data lama)

Tabel Pinjaman(no_pinjaman, jml_pinjaman, nomor_induk, tgl_peminjaman, tgl_jatuh_tempo)

Nama Field	Tipe Data	Panjang Data	Ket
No pinjaman	int	24	PK
Jml pinjaman	int	30	
Nomor induk	int	4	FK
Tgl peminjaman	date		
Tgl jatuh tempo	date		

```
MariaDB [pertiwi]> CREATE TABLE pinjaman(no_pinjaman INT(4) PRIMARY KEY, jml_pinjaman INT(30), tgl_pinjaman DATE, tgl_jatuh_tempo DATE, nomor_induk INT, FOREIGN KEY(nomor_induk) REFERENCES nasabah(nomor_induk));
Query OK, 0 rows affected (0.019 sec)
```

```
MariaDB [pertiwi]> desc pinjaman;
```

Field	Type	Null	Key	Default	Extra
no_pinjaman	int(4)	NO	PRI	NULL	
jml_pinjaman	int(30)	YES		NULL	
tgl_pinjaman	date	YES		NULL	
tgl_jatuh_tempo	date	YES		NULL	
nomor_induk	int(11)	YES	MUL	NULL	

5 rows in set (0.012 sec)

```
CREATE TABLE pinjaman(no_pinjaman INT(4) PRIMARY KEY, jml_pinjaman INT(30), tgl_pinjaman DATE, tgl_jatuh_tempo DATE, nomor_induk INT, FOREIGN KEY(nomor_induk) REFERENCES nasabah(nomor_induk));
```

```
MariaDB [pertiwi]> INSERT INTO pinjaman VALUES ('1001', 5000000, '2022-11-20', '2023-01-20', '1'), ('1002', 65000000, '2023-02-24', '2023-03-24', '2'), ('1003', 68000000, '2023-04-25', '2023-05-25', '3');
Query OK, 3 rows affected (0.003 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
MariaDB [pertiwi]> SELECT*FROM pinjaman;
```

no_pinjaman	jml_pinjaman	tgl_pinjaman	tgl_jatuh_tempo	nomor_induk
1001	5000000	2022-11-20	2023-01-20	1
1002	65000000	2023-02-24	2023-03-24	2
1003	68000000	2023-04-25	2023-05-25	3

3 rows in set (0.001 sec)

```
INSERT INTO pinjaman VALUES ('1001', 5000000, '2022-11-20', '2023-01-20', '1'), ('1002', 65000000, '2023-02-24', '2023-03-24', '2'), ('1003', 68000000, '2023-04-25', '2023-05-25', '3');
```

8. Lakukan transaksi peminjaman oleh salah satu nasabah. Capture query dan hasilnya.

```
MariaDB [pertiwi]> SELECT nasabah.nomor_induk, nasabah.nama, rekening.no_rekening, pinjaman.jml_pinjaman, nasabah.alamat, pinjaman.tgl_pinjaman, pinjaman.tgl_jatuh_tempo
-> FROM nasabah
-> JOIN pinjaman ON nasabah.nomor_induk = pinjaman.nomor_induk
-> JOIN rekening ON nasabah.nomor_induk = rekening.nomor_induk
-> WHERE nasabah.nomor_induk =1;
+-----+-----+-----+-----+-----+-----+-----+
| nomor_induk | nama      | no_rekening | jml_pinjaman | alamat    | tgl_pinjaman | tgl_jatuh_tempo |
+-----+-----+-----+-----+-----+-----+-----+
| 1           | Paramadina | 2222        | 50000000     | Yogyakarta | 2022-11-20   | 2023-01-20       |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.002 sec)
```

```
SELECT nasabah.nomor_induk, nasabah.nama, rekening.no_rekening, pinjaman.jml_pinjaman,
nasabah.alamat, pinjaman.tgl_pinjaman, pinjaman.tgl_jatuh_tempo
FROM nasabah
JOIN pinjaman ON nasabah.nomor_induk = pinjaman.nomor_induk
JOIN rekening ON nasabah.nomor_induk = rekening.nomor_induk
WHERE nasabah.nomor_induk =1;
```

Query ini bertujuan untuk mengambil informasi berikut dari tabel "nasabah", "pinjaman", dan "rekening" dengan melakukan join antara ketiga tabel tersebut berdasarkan kolom "nomor_induk":

- nasabah.nomor_induk
- nasabah.nama
- rekening.no_rekening
- pinjaman.jml_pinjaman
- nasabah.alamat
- pinjaman.tgl_pinjaman
- pinjaman.tgl_jatuh_tempo

Klausula "WHERE nasabah.nomor_induk = 1" digunakan untuk membatasi hasil query hanya untuk nasabah dengan nomor_induk = 1.

Jadi, query ini akan mengambil informasi spesifik tentang nasabah, rekening, dan pinjaman yang terhubung melalui kolom "nomor_induk" untuk nasabah dengan nomor_induk = 1.