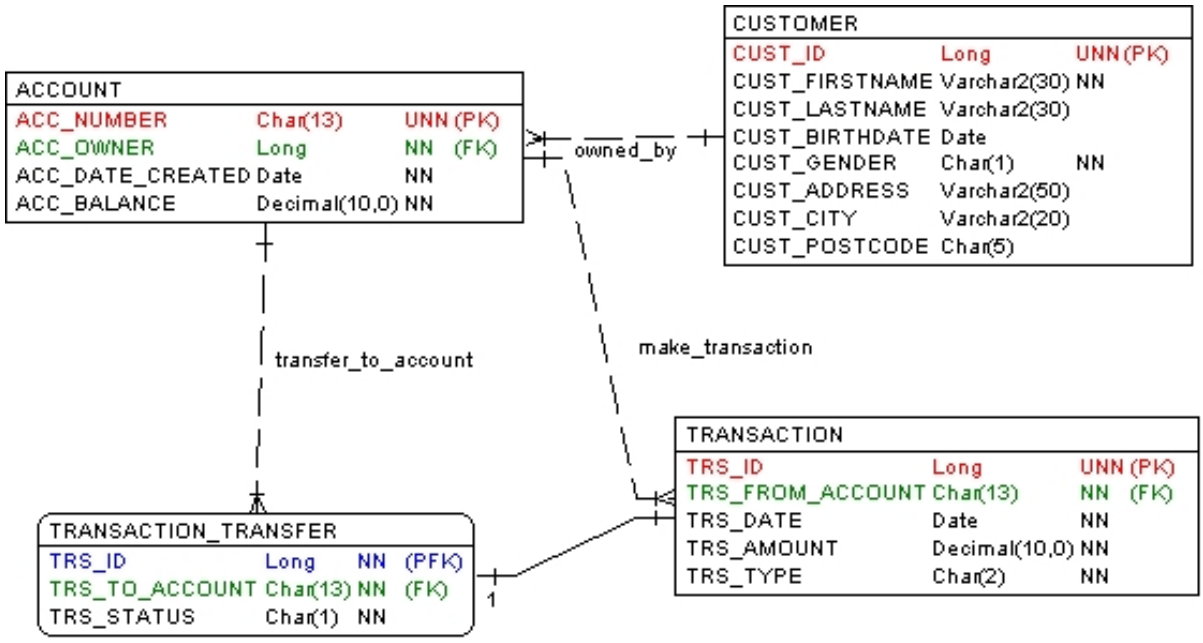
Structured Query Language (SQL)



Note:
UNN : Unique, Not Null
PK : Primary Key
FK : Foreign Key
NN : Not Null

Table description:

1. CUSTOMER: contains all bank customer data
   The column CUST_GENDER is set to 1 for male and 2 for female.
2. ACCOUNT: contains all customer accounts

The column ACC_OWNER is the Foreign Key which refers to the column CUST_ID of the table CUSTOMER.

3. TRANSACTION: records all transactions.
   The column TRS_TYPE contains the transaction type with possible values:
   a. DB: for debit transaction.
   b. CR: for credit transaction.
   c. TF: for money transfer. The column TRS_FROM_ACCOUNT contains the source account.
4. TRANSACTION_TRANSFER: contains additional information if the transaction is money transfer.
   The column TRS_STATUS contains one of the following values:
   a. 0: if transaction is not executed yet
   b. 1: if transaction is executed successfully
   c. -1: if transaction is failed (e.g. the balance of source account (TRS_FROM_ACCOUNT) is not available)

Questions

Create SQL queries for the following requirements:

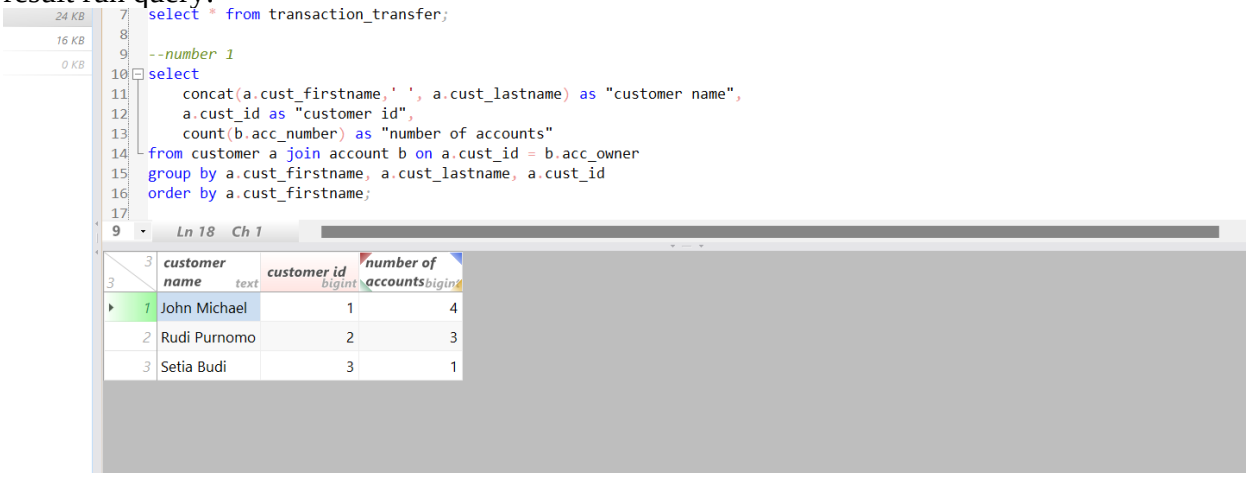
Data dummy insert sql customer : on folder Document -> dummy_customer.sql
Data dummy insert sql account : on folder Document -> dummy_account.sql
Data dummy insert sql transaction: on folder Document -> dummy_transaction.sql
Data dummy insert sql transaction_transfer : on folder Document -> dummy_transaction_transfer.sql

1. Recapitulation of number of accounts owned by every customer.

Answer:

```
select
   concat(a.cust_firstname,' ', a.cust_lastname) as "customer name",
   a.cust_id as "customer id",
   count(b.acc_number) as "number of accounts"
from customer a join account b on a.cust_id = b.acc_owner
group by a.cust_firstname, a.cust_lastname, a.cust_id
order by a.cust_firstname;
```

result run query:



2. All transactions created by John Michael sorted by account number and transaction date

Answer:

```
select
   concat(c.cust_firstname,' ', c.cust_lastname) as "customer name",
   c.cust_id as "customer id",
   b.acc_number as "account number",
   a.*
from transaction a
   join account b on a.trs_from_account = b.acc_number
   join customer c on b.acc_owner = c.cust_id
where c.cust_id = 1
order by b.acc_number, a.trs_date desc;
```

result run query:

```
19
20  --number 2
21  select
22      concat(c.cust_firstname,' ', c.cust_lastname) as "customer name",
23      c.cust_id as "customer id",
24      b.acc_number as "account number",
25      a.*
26  from transaction a
27      join account b on a.trs_from_account = b.acc_number
28      join customer c on b.acc_owner = c.cust_id
29  where c.cust_id = 1
30  order by b.acc_number, a.trs_date desc;
```

Ln 30 Ch 40

	customer name text	customer id bigint	account number	trs_id bigint	trs_amount numeric	trs_date timestamp	trs_from_account character vary...	trs_type character vary...
1	John Michael	1	1	2	60000.00	2023-01-15 00:00:00	1	CR
2	John Michael	1	1	4	70000.00	2022-02-11 00:00:00	1	TF
3	John Michael	1	2	1	20000.00	2023-02-10 00:00:00	2	DB
4	John Michael	1	3	3	10000.00	2021-02-11 00:00:00	3	TF