

# ASSIGNMENT

## BANKING SYSTEM

Name: Maddaka Tejaswini

ASSIGNMENT: BANKING SYSTEM

---

### TASK 1: Database Design

#### 1. Create the database named "HMBank"

```
create database HMBank;  
use HMBank;
```

#### 2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

**/\* Database Tables \*/**

**/\* 1. Customers:**

- customer\_id (Primary Key)
- first\_name
- last\_name
- DOB (Date of Birth)
- email
- phone\_number
- address \*/

```
create table Customers(  
customer_id int primary key identity(001,1),  
first_name varchar(30) ,  
last_name varchar(30) ,  
DOB date ,  
email varchar(100) ,  
phone_number varchar(100) ,  
address varchar(50)  
);
```

**/\*2. Accounts:**

- account\_id (Primary Key)
- customer\_id (Foreign Key)
- account\_type (e.g., savings, current, zero\_balance)
- balance \*/

```

create table Accounts(
account_id varchar(20) primary key,
customer_id int,
account_type varchar(20),
balance decimal(20,2),
foreign key(customer_id) references Customers(customer_id)
);

```

### /\*3. Transactions:

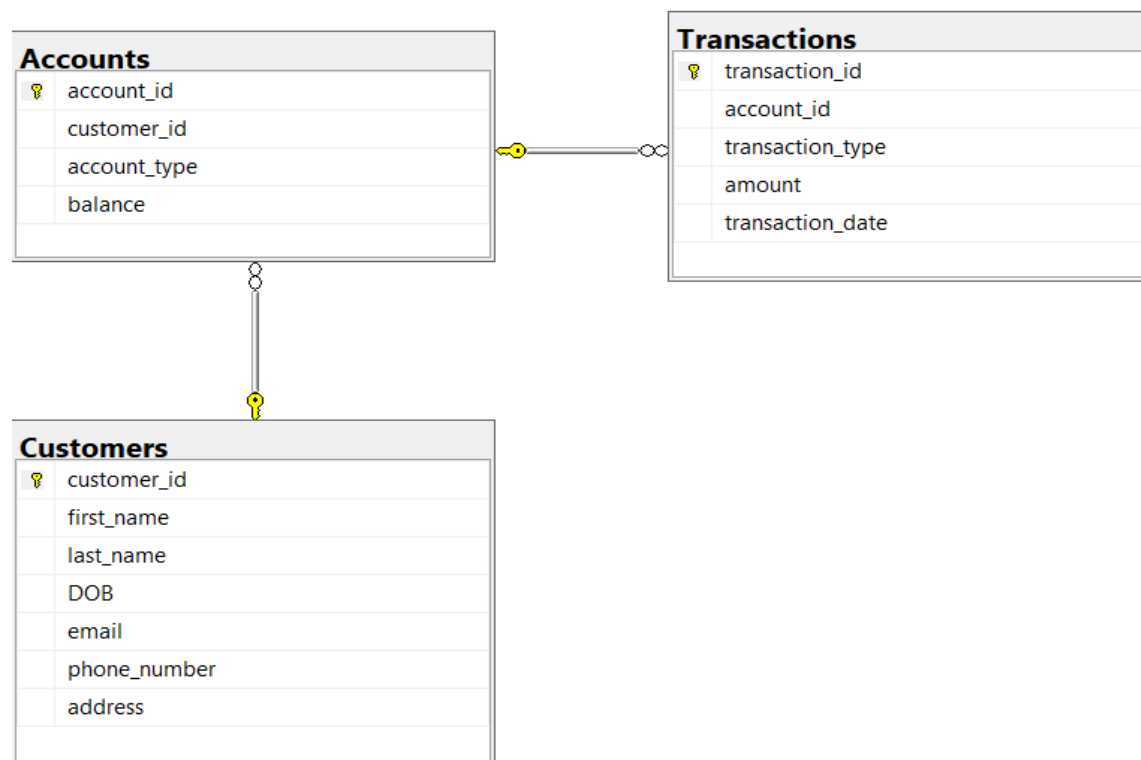
- transaction\_id (Primary Key)
- account\_id (Foreign Key)
- transaction\_type (e.g., deposit, withdrawal, transfer)
- amount
- transaction\_date \*/

```

create table Transactions(
transaction_id varchar(100) primary key, account_id varchar(20),
transaction_type varchar(20),
amount decimal(8,2), transaction_date date,
foreign key (account_id) references Accounts(account_id)
);

```

### 3. Create an ERD (Entity Relationship Diagram) for the database.



#### 4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

All primary keys and foreign keys are considered while creating the tables.

#### TASK 2: Select, Where, Between, AND, LIKE

##### 1. Insert at least 10 sample records into each of the following tables.

- Customers
- Accounts
- Transactions

##### /\* RECORDS OF CUSTOMER TABLE\*/

```
insert into Customers values('Anne','John','2001-10-12',
'annejohn@gmail.com','9852654753','14/480,Church street,Miami');
insert into Customers values('Emma','Thomas','1998-01-08',
'emma@gmail.com','8695756984','1C-10, Lakeview,Portland');
insert into Customers values('Noah','Olivia','2000-09-04',
'olivia12@gmail.com','789654357','12-B,Grifender street,New York');
insert into Customers values('David','Son','1999-02-05',
'david8@gmail.com','7895651423','63/1,Johnson street,San Jose');
insert into Customers values('Martin','Rich','2002-04-06',
'martinz@gmail.com','9563285412','56/9,Wainut,Tucson');
insert into Customers values('Blue','Harris','1997-10-03',
'blue97@gmail.com','6859352946','35-D,Main street,Fort Worth');
insert into Customers values('Kevin','Jose','2003-07-12',
'kevinjose@gmail.com','8534976581','89/7,Cedar,Honolulu');
insert into Customers values('Pat','Carol','2001-04-09',
'patcarol@gmail.com','7689572612','475,Maple,Omaha');
insert into Customers values('Amy','Mathew','2004-10-12',
'amymathew7@gmail.com','7654892642','165/1B,Kingston,Las Vegas');
insert into Customers values('Laura','James','1998-03-05',
'laurajames9@gmail.com','9556411791','164,Second street,Phoenix');
```

##### /\*RECORDS OF ACCOUNTS TABLE\*/

```
insert into Accounts values(4568794568,1,'savings',0.00);
insert into Accounts values(2563597841,2,'current',1900.00);
insert into Accounts values(8659145286,3,'current',7856.00);
insert into Accounts values(7568246648,4,'savings',-1500.00);
insert into Accounts values(2487965441,5,'zero_balance',5600.00);
insert into Accounts values(3774662889,6,'savings',47080.90);
insert into Accounts values(4757678441,7,'zero_balance',148300.00);
insert into Accounts values(5896423598,8,'savings',165000.00);
insert into Accounts values(5221440003,9,'zero_balance',2000.00);
insert into Accounts values(2336640078,10,'current',38250.00);
```

##### /\*RECORDS OF TRANSACTIONS TABLE\*/

```
insert into Transactions
values('T7609182336333033272666',4568794568,'withdrawal',10000.00,'2023-12-30');
insert into Transactions
values('T1235682336333033272456',2563597841,'deposit',250000.90,'2022-06-12');
insert into Transactions
values('T9409145897333033272898',8659145286,'transfer',11060.00,'2021-11-03');
insert into Transactions
values('T270918233699993272753',4568794568,'deposit',120050.50,'2024-06-22');
```

```

insert into Transactions
values('T2129182336333035654159',2487965441,'transfer',1600.00,'2020-03-13');
insert into Transactions
values('T9809182336347326272369',4568794568,'withdrawal',75000.00,'2023-02-07');
insert into Transactions
values('T2409182336333033272666',4757678441,'deposit',29500.00,'2024-07-11');
insert into Transactions
values('T5409182336396458272147',4568794568,'deposit',50000.00,'2024-06-22');
insert into Transactions
values('T3209182336362351272852',5221440003,'withdrawal',90800.00,'2020-10-28');
insert into Transactions
values('T4809182336316598272862',2487965441,'deposit',346120.00,'2024-12-01');

```

## 2.SQL queries for the following tasks:

### 1. Write a SQL query to retrieve the name, account type and email of all customers.

```

select Customers.first_name,Customers.last_name,Accounts.account_type,Customers.email
from Customers join Accounts on Customers.customer_id=Accounts.customer_id;

```

	first_name	last_name	account_type	email
1	Anne	John	current	annejohn@gmail.com
2	Martin	Rich	zero_balance	martinz@gmail.com
3	Emma	Thomas	current	emma@gmail.com
4	Blue	Harris	savings	blue97@gmail.com
5	Anne	John	savings	annejohn@gmail.com
6	Kevin	Jose	zero_balance	kevinjose@gmail.com
7	Amy	Mathew	zero_balance	amymathew7@gmail.com
8	Pat	Carol	savings	patcarol@gmail.com
9	David	Son	savings	david8@gmail.com
10	Noah	Olivia	current	olivia12@gmail.com

### 2. Write a SQL query to list all transaction corresponding customer.

```

select Customers.first_name,Customers.last_name,Transactions.*
from Customers
join Accounts on Customers.customer_id=Accounts.customer_id
join Transactions on Accounts.account_id=Transactions.account_id;

```

	first_name	last_name	transaction_id	account_id	transaction_type	amount	transaction_date
1	Emma	Thomas	T1235682336333033272456	2563597841	deposit	250000.90	2022-06-12
2	Martin	Rich	T2129182336333035654159	2487965441	transfer	1600.00	2020-03-13
3	Kevin	Jose	T2409182336333033272666	4757678441	deposit	29500.00	2024-07-11
4	Anne	John	T270918233699993272753	4568794568	deposit	120050.50	2024-06-22
5	Amy	Mathew	T3209182336362351272852	5221440003	withdrawal	90800.00	2020-10-28
6	Martin	Rich	T4809182336316598272862	2487965441	deposit	346120.00	2024-12-01
7	Pat	Carol	T5409182336396458272147	5896423598	deposit	50000.00	2024-06-22
8	Anne	John	T7609182336333033272666	4568794568	withdrawal	10000.00	2023-12-30
9	Noah	Olivia	T9409145897333033272898	8659145286	transfer	11060.00	2021-11-03
10	Anne	John	T9809182336347326272369	4568794568	withdrawal	75000.00	2023-02-07

### 3. Write a SQL query to increase the balance of a specific account by a certain amount.

```
update Accounts set balance=balance+5000 where balance=10000;
select * from Accounts;
```

	account_id	customer_id	account_type	balance
1	2336640078	1	current	38250.00
2	2487965441	5	zero_balance	5600.00
3	2563597841	2	current	1900.00
4	3774662889	6	savings	47080.90
5	4568794568	1	savings	0.00
6	4757678441	7	zero_balance	148300.00
7	5221440003	9	zero_balance	2000.00
8	5896423598	8	savings	165000.00
9	7568246648	4	savings	-1500.00
10	8659145286	3	current	7856.00

4. Write a SQL query to Combine first and last names of customers as a full\_name.

```
select first_name+' '+last_name as full_name from Customers;
```

	full_name
1	Anne John
2	Emma Thomas
3	Noah Olivia
4	David Son
5	Martin Rich
6	Blue Harris
7	Kevin Jose
8	Pat Carol
9	Amy Mathew
10	Anne John

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.

```
/*deleting transaction record*/
delete t
from Transactions t
join Accounts a on t.account_id = a.account_id
where a.Balance = 0 and a.Account_Type = 'savings';
/* deleting record from account*/
delete from Accounts
where Balance = 0 and Account_Type = 'savings';
```

## Messages

(1 row affected)

Completion time: 2024-09-25T00:12:27.7288651+05:30

### 6. Write a SQL query to Find customers living in a specific city.

```
select * from customers where address like '%Miami%';
```

Results		Messages					
	customer_id	first_name	last_name	DOB	email	phone_number	address
1	1	Anne	John	2001-10-12	annejohn@gmail.com	9852654753	14/480,Church street,Miami

### 7. Write a SQL query to Get the account balance for a specific account.

```
select balance from Accounts where account_id=2487965441;
```

Results		Messages	
	balance		
1	5600.00		

### 8. Write a SQL query to List all current accounts with a balance greater than \$1,000.

```
select * from Accounts where account_type='current' and balance>1000;
```

Results		Messages		
	account_id	customer_id	account_type	balance
1	2336640078	1	current	38250.00
2	2563597841	2	current	1900.00
3	8659145286	3	current	7856.00

### 9. Write a SQL query to Retrieve all transactions for a specific account.

```
select * from Transactions where account_id=2487965441;
```

Results		Messages			
	transaction_id	account_id	transaction_type	amount	transaction_date
1	T2129182336333035654159	2487965441	transfer	1600.00	2020-03-13
2	T4809182336316598272862	2487965441	deposit	346120.00	2024-12-01

### 10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

```
select balance, balance * 0.1 as interest from Accounts where account_type='savings';
```

Results Messages		
	balance	interest
1	47080.90	4708.090
2	0.00	0.000
3	165000.00	16500.000
4	-1500.00	-150.000

**11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.**

```
select * from Accounts where balance < -1000;
```

Results Messages				
	account_id	customer_id	account_type	balance
1	7568246648	4	savings	-1500.00

**12. Write a SQL query to Find customers not living in a specific city.**

```
select * from Customers where address not like '%Miami%';
```

Results Messages							
	customer_id	first_name	last_name	DOB	email	phone_number	address
1	2	Emma	Thomas	1998-01-08	emma@gmail.com	8695756984	1C-10, Lakeview,Portland
2	3	Noah	Olivia	2000-09-04	olivia12@gmail.com	789654357	12-B,Grifender street,New York
3	4	David	Son	1999-02-05	david8@gmail.com	7895651423	63/1,Johnson street,San Jose
4	5	Martin	Rich	2002-04-06	martinz@gmail.com	9563285412	56/9,Wainut,Tucson
5	6	Blue	Harris	1997-10-03	blue97@gmail.com	6859352946	35-D,Main street,Fort Worth
6	7	Kevin	Jose	2003-07-12	kevinjose@gmail.com	8534976581	89/7,Cedar,Honolulu
7	8	Pat	Carol	2001-04-09	patcarol@gmail.com	7689572612	475,Maple,Omaha
8	9	Amy	Mathew	2004-10-12	amymathew7@gmail.com	7654892642	165/1B,Kingston,Las Vegas
9	10	Anne	John	1998-03-05	laurajames9@gmail.com	9556411791	164,Second street,Phoenix

**Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:**

**1. Write a SQL query to Find the average account balance for all customers.**

```
select avg(balance) as average_balance from Accounts ;
```

Results Messages	
	average_balance
1	41448.690000

**2. Write a SQL query to Retrieve the top 10 highest account balances.**

```
select balance from Accounts order by balance desc offset 0 rows fetch first 10 rows only;
```

Results		Messages
	balance	
1	165000.00	
2	148300.00	
3	47080.90	
4	38250.00	
5	7856.00	
6	5600.00	
7	2000.00	
8	1900.00	
9	0.00	
10	-1500.00	

### 3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

```
select count(transaction_type) Total_deposits
from Transactions
where transaction_type='deposit' and transaction_date='2024-06-22';
```

Results		Messages
	Total_deposits	
1	2	

### 4. Write a SQL query to Find the Oldest and Newest Customers.

```
select min(transaction_date) as Oldest_customer, max(transaction_date) as
Newest_customer
from Transactions;
```

Results			Messages
	Oldest_customer	Newest_customer	
1	2020-03-13	2024-12-01	

### 5. Write a SQL query to Retrieve transaction details along with the account type.

```
select Transactions.*,account_type from Transactions
join Accounts on Transactions.account_id=Accounts.account_id;
```

Results							Messages
	transaction_id	account_id	transaction_type	amount	transaction_date	account_type	
1	T1235682336333033272456	2563597841	deposit	250000.90	2022-06-12	current	
2	T2129182336333035654159	2487965441	transfer	1600.00	2020-03-13	zero_balance	
3	T2409182336333033272666	4757678441	deposit	29500.00	2024-07-11	zero_balance	
4	T2709182336999993272753	4568794568	deposit	120050.50	2024-06-22	savings	
5	T3209182336362351272852	5221440003	withdrawal	90800.00	2020-10-28	zero_balance	
6	T4809182336316598272862	2487965441	deposit	346120.00	2024-12-01	zero_balance	
7	T5409182336396458272147	5896423598	deposit	50000.00	2024-06-22	savings	
8	T7609182336333033272666	4568794568	withdrawal	10000.00	2023-12-30	savings	
9	T9409145897333033272898	8659145286	transfer	11060.00	2021-11-03	current	
10	T9809182336347326272369	4568794568	withdrawal	75000.00	2023-02-07	savings	



## 6. Write a SQL query to Get a list of customers along with their account details.

```
select Customers.*,Accounts.* from Customers
join Accounts on Customers.customer_id=Accounts.customer_id;
```

	customer_id	first_name	last_name	DOB	email	phone_number	address	account_id	customer_id	account_type	balance
1	1	Anne	John	2001-10-12	annejohn@gmail.com	9852654753	14/480,Church street,Miami	2336640078	1	current	38250.00
2	5	Martin	Rich	2002-04-06	martinz@gmail.com	9563285412	56/9,Wainut,Tucson	2487965441	5	zero_balance	5600.00
3	2	Emma	Thomas	1998-01-08	emma@gmail.com	8695756984	1C-10, Lakeview,Portland	2563597841	2	current	1900.00
4	6	Blue	Harris	1997-10-03	blue97@gmail.com	6859352946	35-D,Main street,Fort Worth	3774662889	6	savings	47080.90
5	1	Anne	John	2001-10-12	annejohn@gmail.com	9852654753	14/480,Church street,Miami	4568794568	1	savings	0.00
6	7	Kevin	Jose	2003-07-12	kevinjose@gmail.com	8534976581	89/7,Cedar,Honolulu	4757678441	7	zero_balance	148300.00
7	9	Amy	Mathew	2004-10-12	amymathew7@gmail.com	7654892642	165/1B,Kingston,Las Vegas	5221440003	9	zero_balance	2000.00
8	8	Pat	Carol	2001-04-09	patcarol@gmail.com	7689572612	475,Maple,Omaha	5896423598	8	savings	165000.00
9	4	David	Son	1999-02-05	david8@gmail.com	7895651423	63/1,Johnson street,San Jose	7568246648	4	savings	-1500.00
10	3	Noah	Olivia	2000-09-04	olivia12@gmail.com	789654357	12-B,Grifender street,New York	8659145286	3	current	7856.00

## 7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

```
select Transactions.*,Customers.* from Customers
join Accounts on Customers.customer_id=Accounts.customer_id
join Transactions on Accounts.account_id=Transactions.account_id where
Accounts.account_id=8659145286;
```

	transaction_id	account_id	transaction_type	amount	transaction_date	customer_id	first_name	last_name	DOB	email	phone_number	address
1	T940914589733033272898	8659145286	transfer	11060.00	2021-11-03	3	Noah	Olivia	2000-09-04	olivia12@gmail.com	789654357	12-B,Grifender street,New York

## 8. Write a SQL query to Identify customers who have more than one account.

```
select Customers.*
from Customers
group by Customers.customer_id,Customers.first_name,Customers.last_name,
Customers.DOB,Customers.email,Customers.phone_number,Customers.address
having count(Customers.customer_id)>1 ;
```

✓ Query executed successfully.

## 9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

```
select (select sum(amount) from Transactions where transaction_type = 'deposit') -
(select sum(amount) from Transactions where transaction_type = 'withdrawal')
as Difference_in_amount;
```

	Difference_in_amount
1	619871.40

## 10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

```
select Transactions.account_id, avg(balance) as Avg_daily_balance
from Transactions
join Accounts on Transactions.account_id=Accounts.account_id
where transaction_date between '2021-11-03' and '2024-07-11'
group by Transactions.account_id;
```

Results Messages		
	account_id	Avg_daily_balance
1	2563597841	1900.000000
2	4568794568	0.000000
3	4757678441	148300.000000
4	5896423598	165000.000000
5	8659145286	7856.000000

### 11. Calculate the total balance for each account type.

```
select Accounts.account_type, sum(Accounts.balance) as Total_balance
from Accounts
join Customers on Accounts.customer_id=Customers.customer_id
group by Accounts.account_type;
```

Results Messages		
	account_type	Total_balance
1	current	48006.00
2	savings	210580.90
3	zero_balance	155900.00

### 12. Identify accounts with the highest number of transactions order by descending order.

```
select Accounts.*, count(Transactions.account_id) as no_of_transactions
from Accounts
join Transactions on Accounts.account_id=Transactions.account_id
group by Transactions.account_id, Accounts.account_id, Accounts.customer_id,
Accounts.account_type, Accounts.balance
order by no_of_transactions desc;
```

Results Messages					
	account_id	customer_id	account_type	balance	no_of_transactions
1	4568794568	1	savings	0.00	3
2	2487965441	5	zero_balance	5600.00	2
3	2563597841	2	current	1900.00	1
4	4757678441	7	zero_balance	148300.00	1
5	5221440003	9	zero_balance	2000.00	1
6	5896423598	8	savings	165000.00	1
7	8659145286	3	current	7856.00	1

### 13. List customers with high aggregate account balances, along with their account types.

```
select account_type, account_id, sum(balance) as agg_acc_bal
from Accounts
group by account_id, account_type
order by agg_acc_bal desc offset 0 rows fetch first 5 rows only ;
```

	account_type	account_id	agg_acc_bal
1	savings	5896423598	165000.00
2	zero_balance	4757678441	148300.00
3	savings	3774662889	47080.90
4	current	2336640078	38250.00
5	current	8659145286	7856.00

#### 14. Identify and list duplicate transactions based on transaction amount, date, and account.

```
select amount, transaction_date, account_id, count(*) as Duplicate_count
from Transactions
group by amount, transaction_date, account_id
having count(*) > 1;
```

✓ Query executed successfully.

#### Tasks 4: Subquery and its type:

##### 1. Retrieve the customer(s) with the highest account balance.

```
select c.*, (select a.balance from Accounts a
where a.customer_id = c.customer_id and a.balance =
(select max(balance) from Accounts)) as balance from Customers c
where (select max(balance) from Accounts a
where a.customer_id = c.customer_id) = (select max(balance) from Accounts);
```

	customer_id	first_name	last_name	DOB	email	phone_number	address	balance
1	8	Pat	Carol	2001-04-09	patcarol@gmail.com	7689572612	475,Maple,Omaha	165000.00

##### 2. Calculate the average account balance for customers who have more than one account.

```
select avg(balance) as Avg_acc_balance
from Accounts
where customer_id in (select customer_id from Accounts
group by customer_id
having count(account_id) > 1);
```

	Avg_acc_balance
1	19125.000000

##### 3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

```
select account_id, amount, (select avg(amount) from Transactions) as
Average_transaction_amount
from Transactions
where amount > (select avg(amount)
from Transactions);
```

	account_id	amount	Average_transaction_amount
1	2563597841	250000.90	98413.140000
2	4568794568	120050.50	98413.140000
3	2487965441	346120.00	98413.140000

#### 4. Identify customers who have no recorded transactions.

```
select * from Customers
where customer_id NOT IN (select distinct customer_id
from Accounts
where account_id IN (select distinct account_id
from Transactions));
```

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	4	David	Son	1999-02-05	david8@gmail.com	7895651423	63/1,Johnson street,San Jose
2	6	Blue	Harris	1997-10-03	blue97@gmail.com	6859352946	35-D,Main street,Fort Worth
3	10	Anne	John	1998-03-05	laurajames9@gmail.com	9556411791	164,Second street,Phoenix

#### 5. Calculate the total balance of accounts with no recorded transactions.

```
select sum(a.balance) as Total_balance
from Accounts a
where a.account_id NOT IN (select t.account_id
from Transactions t);
```

	Total_balance
1	83830.90

#### 6. Retrieve transactions for accounts with the lowest balance.

```
select Transactions.*,Accounts.balance
from Transactions
right join Accounts on Transactions.account_id=Accounts.account_id
where Accounts.balance=
(select min(balance) from Accounts );
```

	transaction_id	account_id	transaction_type	amount	transaction_date	balance
1	NULL	NULL	NULL	NULL	NULL	-1500.00

#### 7. Identify customers who have accounts of multiple types.

```
select c.customer_id, c.first_name, c.last_name
from Customers c
where(select count(distinct a.account_type)
from Accounts a
where a.customer_id = c.customer_id)>1;
```

	customer_id	first_name	last_name
1	1	Anne	John

**8. Calculate the percentage of each account type out of the total number of accounts.**

```
select account_type, count(*) * 100.0 / (select count(*) from Accounts) as Percentage
from Accounts
group by account_type;
```

	account_type	Percentage
1	current	30.0000000000000
2	savings	40.0000000000000
3	zero_balance	30.0000000000000

**9. Retrieve all transactions for a customer with a given customer\_id.**

```
select *
from Transactions t
where t.account_id = (select a.account_id
from Accounts a
where a.customer_id = '9');
```

	transaction_id	account_id	transaction_type	amount	transaction_date
1	T3209182336362351272852	5221440003	withdrawal	90800.00	2020-10-28

**10. Calculate the total balance for each account type, including a subquery within the SELECT clause.**

```
select a.account_type, (select sum(balance)
from Accounts
where account_type = a.account_type) as Total_balance
from Accounts a
group by a.account_type;
```

	account_type	Total_balance
1	current	48006.00
2	savings	210580.90
3	zero_balance	155900.00