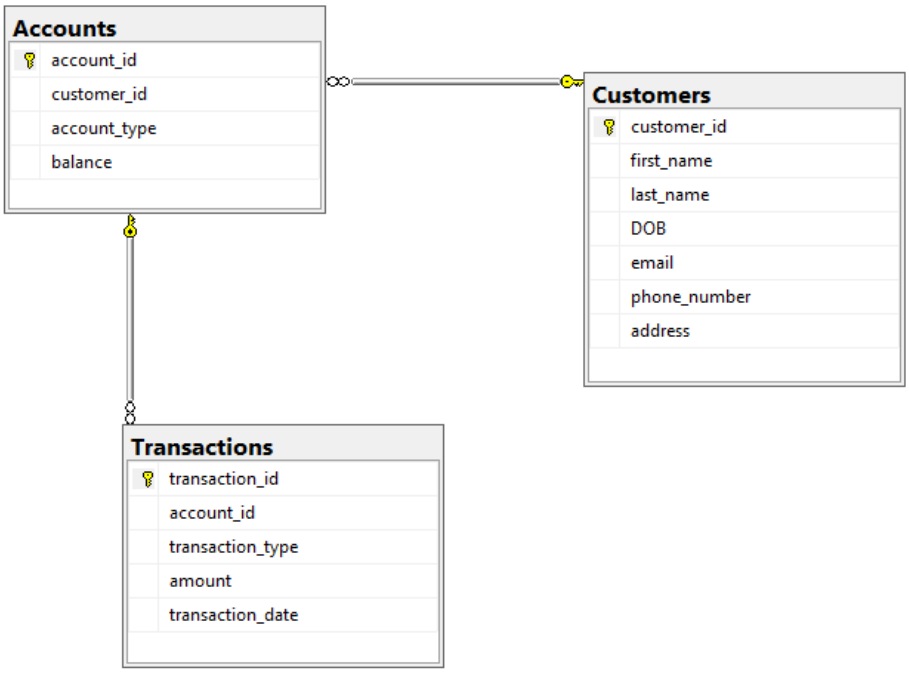
ASSIGNMENT 3 – BANKING SYSTEM – SNEHA C

```
/*Tasks 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
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(30),
    last_name VARCHAR(30),
    DOB DATE,
    email VARCHAR(50),
    phone_number VARCHAR(20),
    address VARCHAR(255));

CREATE TABLE Accounts (
    account_id INT PRIMARY KEY,
    customer_id INT,
    account_type VARCHAR(50),
    balance DECIMAL(15, 2),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id));

CREATE TABLE Transactions (
    transaction_id INT PRIMARY KEY,
    account_id INT,
    transaction_type VARCHAR(50),
    amount DECIMAL(15, 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.
/*

CUSTOMER TABLE:
--'customer_id' is the primary key of the Customers table.

ACCOUNTS TABLE:
--'account_id' is the primary key of the Accounts table.
--'customer_id' is a foreign key referencing the customer_id in the Customers table.

TRANSACTIONS TABLE:
--'transaction_id' is the primary key of the Transactions table.
--'account_id' is a foreign key referencing the account_id in the Accounts table

*/

/*5. Write SQL scripts to create the mentioned tables with appropriate data types,
constraints,
and relationships.
• Customers
• Accounts
• Transactions

RELATIONSHIPS:
--Each customer can have multiple accounts
--Each account is associated with a single customer
--Each transaction is associated with a single account
--Accounts can have multiple transactions

*/


/*Tasks 2: Select, Where, Between, AND, LIKE:*/

/*
1. Insert at least 10 sample records into each of the following tables.
• Customers
• Accounts
• Transactions
*/

INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number,
address)
VALUES
    (1, 'Rahul', 'Kumar', '1990-05-15', 'rahul.kumar@email.com', '1234567890', '123
Main Street, Connaught Place, Delhi'),
    (2, 'Priya', 'Singh', '1985-08-22', 'priya.singh@email.com', '9876543210', '456
Oak Street, Bandra, Mumbai'),
    (3, 'Amit', 'Sharma', '1992-12-10', 'amit.sharma@email.com', '5551234567', '789
Pine Street, Salt Lake, Kolkata'),
    (4, 'Neha', 'Patil', '1980-02-28', 'neha.patil@email.com', '3335559999', '101
Maple Street, Indiranagar, Bangalore'),
    (5, 'Raj', 'Gupta', '1995-07-18', 'raj.gupta@email.com', '7778881234', '202 Cedar
Street, Jubilee Hills, Hyderabad'),
    (6, 'Sneha', 'Sekar', '1988-09-05', 'sneha.sekar@email.com', '4447772222', '303
Elm Street, T Nagar, Chennai'),
    (7, 'Vikas', 'Rao', '1998-03-20', 'vikas.rao@email.com', '1112223333', '404 Birch
Street, Koregaon Park, Pune'),
    (8, 'Meera', 'Malhotra', '1983-11-12', 'meera.malhotra@email.com', '9990001111',
'505 Oak Street, Navrangpura, Ahmedabad'),
    (9, 'Aniket', 'Nair', '1991-06-25', 'aniket.nair@email.com', '2224446666', '606
Pine Street, Malviya Nagar, Jaipur'),
```

```
    (10, 'Pooja', 'Rajput', '1987-04-03', 'pooja.rajput@email.com', '8889994444', '707
Maple Street, Andheri West, Mumbai');

SELECT * FROM Customers;
```

Results | Messages

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	1	Rahul	Kumar	1990-05-15	rahul.kumar@email.com	1234567890	123 Main Street, Connaught Place, Delhi
2	2	Priya	Singh	1985-08-22	priya.singh@email.com	9876543210	456 Oak Street, Bandra, Mumbai
3	3	Amit	Sharma	1992-12-10	amit.sharma@email.com	5551234567	789 Pine Street, Salt Lake, Kolkata
4	4	Neha	Patil	1980-02-28	neha.patil@email.com	3335559999	101 Maple Street, Indiranagar, Bangalore
5	5	Raj	Gupta	1995-07-18	raj.gupta@email.com	7778881234	202 Cedar Street, Jubilee Hills, Hyderabad
6	6	Sneha	Sekar	1988-09-05	sneha.sekar@email.com	4447772222	303 Elm Street, T Nagar, Chennai
7	7	Vikas	Rao	1998-03-20	vikas.rao@email.com	1112223333	404 Birch Street, Koregaon Park, Pune
8	8	Meera	Malhotra	1983-11-12	meera.malhotra@email.com	9990001111	505 Oak Street, Navrangpura, Ahmedabad
9	9	Aniket	Nair	1991-06-25	aniket.nair@email.com	2224446666	606 Pine Street, Malviya Nagar, Jaipur
10	10	Pooja	Rajput	1987-04-03	pooja.rajput@email.com	8889994444	707 Maple Street, Andheri West, Mumbai

```
INSERT INTO Accounts (account_id, customer_id, account_type, balance)
VALUES
    (101, 1, 'savings', 5000.00),
    (102, 1, 'current', 2000.00),
    (103, 2, 'savings', 8000.00),
    (104, 2, 'current', 3000.00),
    (105, 3, 'savings', 10000.00),
    (106, 3, 'current', 1500.00),
    (107, 4, 'zero_balance', 0.00),
    (108, 4, 'current', 7000.00),
    (109, 5, 'savings', 6000.00),
    (110, 5, 'zero_balance', 0.00);

SELECT * FROM Accounts;
```

Results | Messages

	account_id	customer_id	account_type	balance
1	101	1	savings	5000.00
2	102	1	current	2000.00
3	103	2	savings	8000.00
4	104	2	current	3000.00
5	105	3	savings	10000.00
6	106	3	current	1500.00
7	107	4	zero_balance	0.00
8	108	4	current	7000.00
9	109	5	savings	6000.00
10	110	5	zero_balance	0.00

```
INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount,
transaction_date)
VALUES
    (1001, 101, 'deposit', 1000.00, '2024-03-01'),
    (1002, 101, 'withdrawal', 500.00, '2024-03-02'),
    (1003, 102, 'deposit', 1500.00, '2024-03-03'),
    (1004, 103, 'withdrawal', 2000.00, '2024-03-04'),
    (1005, 104, 'deposit', 800.00, '2024-03-05'),
    (1006, 105, 'transfer', 1200.00, '2024-03-06'),
    (1007, 106, 'transfer', 200.00, '2024-03-07'),
    (1008, 107, 'deposit', 0.00, '2024-03-08'),
    (1009, 108, 'deposit', 1000.00, '2024-03-09'),
    (1010, 109, 'transfer', 300.00, '2024-03-10');
```

```
SELECT * FROM Transactions;
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1001	101	deposit	1000.00	2024-03-01
2	1002	101	withdrawal	500.00	2024-03-02
3	1003	102	deposit	1500.00	2024-03-03
4	1004	103	deposit	2000.00	2024-03-04
5	1005	104	deposit	800.00	2024-03-05
6	1006	105	transfer	1200.00	2024-03-06
7	1007	106	transfer	200.00	2024-03-07
8	1008	107	deposit	0.00	2024-03-08
9	1009	108	deposit	1000.00	2024-03-09
10	1010	109	transfer	300.00	2024-03-10

```
/*2. Write SQL queries for the following tasks:*/

--1. Write a SQL query to retrieve the name, account type and email of all customers.
select c.first_name,c.last_name,a.account_type,c.email from Customers c join Accounts
a on c.customer_id=a.customer_id;
```

Results | Messages

	first_name	last_name	account_type	email
1	Rahul	Kumar	savings	rahul.kumar@email.com
2	Rahul	Kumar	current	rahul.kumar@email.com
3	Priya	Singh	savings	priya.singh@email.com
4	Priya	Singh	current	priya.singh@email.com
5	Amit	Sharma	savings	amit.sharma@email.com
6	Amit	Sharma	current	amit.sharma@email.com
7	Neha	Patil	zero_balance	neha.patil@email.com
8	Neha	Patil	current	neha.patil@email.com
9	Raj	Gupta	savings	raj.gupta@email.com
10	Raj	Gupta	zero_balance	raj.gupta@email.com

```
--2. Write a SQL query to list all transaction corresponding customer.
declare @customer int=4;
select t.* from Transactions t join Accounts a on t.account_id=a.account_id where
a.customer_id=@customer;
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1008	107	deposit	0.00	2024-03-08
2	1009	108	deposit	1000.00	2024-03-09

```
--3. Write a SQL query to increase the balance of a specific account by a certain
amount.
declare @account int = 101;
declare @increment decimal(15,2) = 1000.00;
update Accounts set balance=balance+@increment where account_id=@account;
```

Messages

```
(1 row affected)

Completion time: 2024-03-06T10:46:31.2285196+05:30
```

Results Messages

	account_id	customer_id	account_type	balance
1	101	1	savings	6000.00
2	102	1	current	2000.00
3	103	2	savings	8000.00
4	104	2	current	3000.00
5	105	3	savings	10000.00
6	106	3	current	1500.00
7	107	4	zero_balance	0.00
8	108	4	current	7000.00
9	109	5	savings	6000.00
10	110	5	zero_balance	0.00

```
--4. Write a SQL query to Combine first and last names of customers as a full_name.
select concat(first_name,' ',last_name) as full_name from Customers;
```

Results Messages

	full_name
1	Rahul Kumar
2	Priya Singh
3	Amit Sharma
4	Neha Patil
5	Raj Gupta
6	Sneha Sekar
7	Vikas Rao
8	Meera Malhotra
9	Aniket Nair
10	Pooja Rajput

```
--5. Write a SQL query to remove accounts with a balance of zero where the account
--type is savings.
delete from Accounts where balance=0.00 and account_type='savings';
```

Messages

```
(0 rows affected)

Completion time: 2024-03-06T10:54:42.8690540+05:30
```

```
--6. Write a SQL query to Find customers living in a specific city.
select * from Customers where address like '%mumbai%';
```

Results Messages

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	2	Priya	Singh	1985-08-22	priya.singh@email.com	9876543210	456 Oak Street, Bandra, Mumbai
2	10	Pooja	Rajput	1987-04-03	pooja.rajput@email.com	8889994444	707 Maple Street, Andheri West, Mumbai

```
--7. Write a SQL query to Get the account balance for a specific account.
declare @account int = 102;
```

```
select balance from Accounts where account_id=@account;
```

Results | Messages

	account_id	customer_id	account_type	balance
1	102	1	current	2000.00
2	104	2	current	3000.00
3	106	3	current	1500.00
4	108	4	current	7000.00

```
--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>1000.00;
```

Results | Messages

	account_id	customer_id	account_type	balance
1	102	1	current	2000.00
2	104	2	current	3000.00
3	106	3	current	1500.00
4	108	4	current	7000.00

```
--9. Write a SQL query to Retrieve all transactions for a specific account.
declare @account int =101;
select * from Transactions where account_id=@account;
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1001	101	deposit	1000.00	2024-03-01
2	1002	101	withdrawal	500.00	2024-03-02

```
--10. Write a SQL query to Calculate the interest accrued on savings accounts based on
a
--given interest rate.
declare @interest int = 5;
select (balance*1* @interest)/100 as interest_accrued from Accounts where
account_type='savings';
```

Results | Messages

	interest_accrued
1	350.000000
2	400.000000
3	500.000000
4	300.000000

```
--11. Write a SQL query to Identify accounts where the balance is less than a
specified
--overdraft limit.
declare @overdraft decimal(15,2)=100;
select * from Accounts where balance between 0 and @overdraft;
```

Results | Messages

	account_id	customer_id	account_type	balance
1	107	4	zero_balance	0.00
2	110	5	zero_balance	0.00

```sql
--12. Write a SQL query to Find customers not living in a specific city.
select * from Customers where address not like '%chennai%';
```

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	1	Rahul	Kumar	1990-05-15	rahul.kumar@email.com	1234567890	123 Main Street, Connaught Place, Delhi
2	2	Priya	Singh	1985-08-22	priya.singh@email.com	9876543210	456 Oak Street, Bandra, Mumbai
3	3	Amit	Sharma	1992-12-10	amit.sharma@email.com	5551234567	789 Pine Street, Salt Lake, Kolkata
4	4	Neha	Patil	1980-02-28	neha.patil@email.com	3335559999	101 Maple Street, Indiranagar, Bangalore
5	5	Raj	Gupta	1995-07-18	raj.gupta@email.com	7778881234	202 Cedar Street, Jubilee Hills, Hyderabad
6	7	Vikas	Rao	1998-03-20	vikas.rao@email.com	1112223333	404 Birch Street, Koregaon Park, Pune
7	8	Meera	Malhotra	1983-11-12	meera.malhotra@email.com	9990001111	505 Oak Street, Navrangpura, Ahmedabad
8	9	Aniket	Nair	1991-06-25	aniket.nair@email.com	2224446666	606 Pine Street, Malviya Nagar, Jaipur
9	10	Pooja	Rajput	1987-04-03	pooja.rajput@email.com	8889994444	707 Maple Street, Andheri West, Mumbai

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

```sql
--1. Write a SQL query to Find the average account balance for all customers.
select customer_id,avg(balance) as Average_Balance from Accounts group by customer_id;
```

Results Messages

	customer_id	Average_Balance
1	1	4500.000000
2	2	5500.000000
3	3	5750.000000
4	4	3500.000000
5	5	3000.000000

```sql
--2. Write a SQL query to Retrieve the top 10 highest account balances.
select top 10 balance from Accounts order by balance desc;
```

Results Messages

	balance
1	10000.00
2	8000.00
3	7000.00
4	7000.00
5	6000.00
6	3000.00
7	2000.00
8	1500.00
9	0.00
10	0.00

```sql
--3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.
select sum(amount) as Total_Deposits from Transactions where
transaction_type='deposit' and transaction_date='2024-03-01';
```

Results Messages

	Total_Deposits
1	1000.00

```sql
--4. Write a SQL query to Find the Oldest and Newest Customers.
select min(customer_id) as Oldest_Customer from Customers;
```

Results Messages

	Oldest_Customer
1	1

```
select max(customer_id) as Newest_Customer from Customers;
```

Results Messages

	Newest_Customer
1	10

```
--5. Write a SQL query to Retrieve transaction details along with the account type.
select t.*,a.account_type from Transactions t join Accounts a on
t.account_id=a.account_id;
```

Results Messages

	transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1	1001	101	deposit	1000.00	2024-03-01	savings
2	1002	101	withdrawal	500.00	2024-03-02	savings
3	1003	102	deposit	1500.00	2024-03-03	current
4	1004	103	deposit	2000.00	2024-03-04	savings
5	1005	104	deposit	800.00	2024-03-05	current
6	1006	105	transfer	1200.00	2024-03-06	savings
7	1007	106	transfer	200.00	2024-03-07	current
8	1008	107	deposit	0.00	2024-03-08	zero_balance
9	1009	108	deposit	1000.00	2024-03-09	current
10	1010	109	transfer	300.00	2024-03-10	savings

```
--6. Write a SQL query to Get a list of customers along with their account details.
select c.customer_id,c.first_name,c.last_name,a.* from Customers c join Accounts a on
c.customer_id=a.customer_id;
```

Results Messages

	transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1	1001	101	deposit	1000.00	2024-03-01	savings
2	1002	101	withdrawal	500.00	2024-03-02	savings
3	1003	102	deposit	1500.00	2024-03-03	current
4	1004	103	deposit	2000.00	2024-03-04	savings
5	1005	104	deposit	800.00	2024-03-05	current
6	1006	105	transfer	1200.00	2024-03-06	savings
7	1007	106	transfer	200.00	2024-03-07	current
8	1008	107	deposit	0.00	2024-03-08	zero_balance
9	1009	108	deposit	1000.00	2024-03-09	current
10	1010	109	transfer	300.00	2024-03-10	savings

```
--7. Write a SQL query to Retrieve transaction details along with customer information
for a
--specific account.
declare @account int = 101;
select t.*,c.* from Transactions t join Accounts a on t.account_id=a.account_id
join Customers c on a.customer_id=c.customer_id where a.account_id=101;
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date	customer_id	first_name	last_name	DOB	email	phone_number	address
1	1001	101	deposit	1000.00	2024-03-01	1	Rahul	Kumar	1990-05-15	rahul.kumar@email.com	1234567890	123 Main Street, Connaught Place, Delhi
2	1002	101	withdrawal	500.00	2024-03-02	1	Rahul	Kumar	1990-05-15	rahul.kumar@email.com	1234567890	123 Main Street, Connaught Place, Delhi

```
--8. Write a SQL query to Identify customers who have more than one account.
select customer_id,count(account_id) as Account_count from Accounts group by
customer_id having count(account_id)>1;
```

	customer_id	Account_count
1	1	2
2	2	2
3	3	2
4	4	2
5	5	2

```
--9. Write a SQL query to Calculate the difference in transaction amounts between
deposits and
--withdrawals
select account_id, sum(case when transaction_type='deposit' then amount else 0 end)-
sum(case when transaction_type='withdrawal' then amount else 0 end)
as amount_difference from Transactions group by account_id;
```

Results | Messages

	account_id	amount_difference
1	101	500.00
2	102	1500.00
3	103	2000.00
4	104	800.00
5	105	0.00
6	106	0.00
7	107	0.00
8	108	1000.00
9	109	0.00

```
--10. Write a SQL query to Calculate the average daily balance for each account over a
specified
--period
declare @start DATE = '2024-03-01';
declare @end DATE = '2024-03-31';


select t.account_id,avg(a.balance) as average_daily_balance from Transactions t
join Accounts a on t.account_id = a.account_id where
t.transaction_date between @start and @end group by t.account_id;
```

Results Messages

	account_id	average_daily_balance
1	101	7000.000000
2	102	2000.000000
3	103	8000.000000
4	104	3000.000000
5	105	10000.000000
6	106	1500.000000
7	107	0.000000
8	108	7000.000000
9	109	6000.000000

```
--11. Calculate the total balance for each account type.
select account_type,sum(balance) as Total_balance from Accounts group by account_type;
```

Results Messages

	account_type	Total_balance
1	current	13500.00
2	savings	31000.00
3	zero_balance	0.00

```
--12. Identify accounts with the highest number of transactions order by descending
order.
select account_id,count(*) as transaction_count from Transactions group by account_id
order by transaction_count desc;
```

Results Messages

	account_id	transaction_count
1	101	2
2	102	1
3	103	1
4	104	1
5	105	1
6	106	1
7	107	1
8	108	1
9	109	1

```
--13. List customers with high aggregate account balances, along with their account
types.
select Customers.customer_id,account_type,sum(balance) as aggregate_balance
from Customers join Accounts on Customers.customer_id = Accounts.customer_id
group by Customers.customer_id, first_name, last_name, account_type
order by aggregate_balance desc;
```

Results | Messages

	customer_id	account_type	aggregate_balance
1	3	savings	10000.00
2	2	savings	8000.00
3	4	current	7000.00
4	1	savings	7000.00
5	5	savings	6000.00
6	2	current	3000.00
7	1	current	2000.00
8	3	current	1500.00
9	4	zero_balance	0.00
10	5	zero_balance	0.00

```
--14. Identify and list duplicate transactions based on transaction amount, date, and
account.
select transaction_id,amount, transaction_date, account_id, count(*) as
Transaction_Count
from Transactions group by transaction_id, amount, transaction_date, account_id
having count(*) > 1;
```

Results | Messages

	transaction_id	amount	transaction_date	account_id	Transaction_Count

```
/*Tasks 4: Subquery and its type:*/

--1. Retrieve the customer(s) with the highest account balance.
select * from Customers where customer_id=(select customer_id from Accounts
where balance= (select max(balance) from Accounts));
```

Results | Messages

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	3	Amit	Sharma	1992-12-10	amit.sharma@email.com	5551234567	789 Pine Street, Salt Lake, Kolkata

```
--2. Calculate the average account balance for customers who have more than one
account.
select customer_id,avg(balance) as average_account_balance from Accounts group by
customer_id having count(*)>1;
```

Results | Messages

	customer_id	average_account_balance
1	1	4500.000000
2	2	5500.000000
3	3	5750.000000
4	4	3500.000000
5	5	3000.000000

```sql
--3. Retrieve accounts with transactions whose amounts exceed the average transaction
amount.
select * from Transactions where amount> (select avg(amount) from Transactions);
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1001	101	deposit	1000.00	2024-03-01
2	1003	102	deposit	1500.00	2024-03-03
3	1004	103	deposit	2000.00	2024-03-04
4	1006	105	transfer	1200.00	2024-03-06
5	1009	108	deposit	1000.00	2024-03-09

```sql
--4. Identify customers who have no recorded transactions.
select * from Customers where customer_id not in (select customer_id from Accounts
where account_id in (select distinct account_id from Transactions));
```

Results | Messages

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	6	Sneha	Sekar	1988-09-05	sneha.sekar@email.com	4447772222	303 Elm Street, T Nagar, Chennai
2	7	Vikas	Rao	1998-03-20	vikas.rao@email.com	1112223333	404 Birch Street, Koregaon Park, Pune
3	8	Meera	Malhotra	1983-11-12	meera.malhotra@email.com	9990001111	505 Oak Street, Navrangpura, Ahmedabad
4	9	Aniket	Nair	1991-06-25	aniket.nair@email.com	2224446666	606 Pine Street, Malviya Nagar, Jaipur
5	10	Pooja	Rajput	1987-04-03	pooja.rajput@email.com	8889994444	707 Maple Street, Andheri West, Mumbai

```sql
--5. Calculate the total balance of accounts with no recorded transactions.
select sum(balance) as total_balance from Accounts where account_id not in (select
distinct account_id from Transactions);
```

Results | Messages

	total_balance
1	0.00

```sql
--6. Retrieve transactions for accounts with the lowest balance
select * from Transactions where account_id in (select account_id from Accounts where
balance = (select min(balance) from Accounts));
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1008	107	deposit	0.00	2024-03-08

```sql
--7. Identify customers who have accounts of multiple types.
select * from Customers where customer_id in (select customer_id from Accounts
group by customer_id having count(account_type)>1);
```

Results | Messages

	customer_id	first_name	last_name	DOB	email	phone_number	address
1	1	Rahul	Kumar	1990-05-15	rahul.kumar@email.com	1234567890	123 Main Street, Connaught Place, Delhi
2	2	Priya	Singh	1985-08-22	priya.singh@email.com	9876543210	456 Oak Street, Bandra, Mumbai
3	3	Amit	Sharma	1992-12-10	amit.sharma@email.com	5551234567	789 Pine Street, Salt Lake, Kolkata
4	4	Neha	Patil	1980-02-28	neha.patil@email.com	3335559999	101 Maple Street, Indiranagar, Bangalore
5	5	Raj	Gupta	1995-07-18	raj.gupta@email.com	7778881234	202 Cedar Street, Jubilee Hills, Hyderabad

```sql
--8. Calculate the percentage of each account type out of the total number of accounts
SELECT account_type, count(*) AS account_count, count(*) * 100.0 / (select count(*)
from Accounts)
```

```
as percentage from Accounts group by account_type;
```

Results | Messages

	account_type	account_count	percentage
1	current	4	40.000000000000
2	savings	4	40.000000000000
3	zero_balance	2	20.000000000000

```
--9. Retrieve all transactions for a customer with a given customer_id.
declare @customer int = 5;
select * from Transactions where account_id in (select account_id from Accounts where
customer_id=@customer);
```

Results | Messages

	transaction_id	account_id	transaction_type	amount	transaction_date
1	1010	109	transfer	300.00	2024-03-10

```
--10. Calculate the total balance for each account type, including a subquery within
the SELECT
--clause.
select account_type, sum(balance)
as Total_Balance from Accounts group by account_type;
```

Results | Messages

	account_type	Total_Balance
1	current	13500.00
2	savings	31000.00
3	zero_balance	0.00