ASSIGNMENT:

BANKING SYSTEM

TASK 1

1. Create the database named "HMBank"

```
CREATE DATABASE HMBANK;
```

✔	13	20:34:43	CREATE DATABASE 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 text, last_name text, DOB date, email varchar(50), phone_number varchar(10), address varchar(150));

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

create table Transactions( transaction_id int primary key, account_id int, transaction_type varchar(25), amount bigint,transaction_date date, foreign key(account_id) references Accounts(account_id));
```

accounts
customers
transactions

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
DOB	date	YES		NULL	
email	varchar(50)	YES		NULL	
phone_number	varchar(10)	YES		NULL	
address	varchar(150)	YES		NULL	

Field	Type	Null	Key	Default	Extra
account_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
account_type	varchar(20)	YES		NULL	
balance	bigint	YES		NULL	

Field	Type	Null	Key	Default	Extra
transaction_id	int	NO	PRI	NULL	
account_id	int	YES	MUL	NULL	
transaction_type	varchar(25)	YES		NULL	
amount	bigint	YES		NULL	
transaction_date	date	YES		NULL	

TASK 2:

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

Customers

```
INSERT INTO Customers VALUES
(1, 'John', 'Doe', '1990-05-15', 'john.doe@example.com', 1234567890, '123 Main St, City, Country'),
(2, 'Jane', 'Smith', '1985-09-20', 'jane.smith@example.com', 9876543210, '456 Elm St, City, Country'),
(3, 'Michael', 'Johnson', '1978-11-12', 'michael.johnson@example.com', 5551234567, '789 Oak St, City, Country'),
(4, 'Emily', 'Brown', '1995-03-25', 'emily.brown@example.com', 9998887776, '321 Maple St, City, Country'),
(5, 'David', 'Wilson', '1980-07-08', 'david.wilson@example.com', 4445556666, '654 Pine St, City, Country'),
(6, 'Sarah', 'Taylor', '1992-12-10', 'sarah.taylor@example.com', 3332221111, '876 Cedar St, City, Country'),
(7, 'Christopher', 'Martinez', '1983-04-18', 'chris.martinez@example.com', 1112223333, '432 Birch St, City, Country'),
(8, 'Jessica', 'Anderson', '1975-08-22', 'jessica.anderson@example.com', 7778889999, '567 Walnut St, City, Country'),
(9, 'Daniel', 'Garcia', '1998-01-05', 'daniel.garcia@example.com', 6667778888, '890 Cherry St, City, Country'),
(10, 'Amanda', 'Thomas', '1987-06-30', 'amanda.thomas@example.com', 2223334444, '789 Pineapple St, City, Country'),
(11, 'Matthew', 'Hernandez', '1993-09-28', 'matthew.hernandez@example.com', 5556667777, '345 Mango St, City, Country'),
(12, 'Lauren', 'King', '1981-02-14', 'lauren.king@example.com', 9990001111, '234 Peach St, City, Country'),
(13, 'Ryan', 'Young', '1976-07-19', 'ryan.young@example.com', 4443332222, '678 Banana St, City, Country'),
(14, 'Stephanie', 'Scott', '1990-11-03', 'stephanie.scott@example.com', 8889990000, '543 Grape St, City, Country'),
(15, 'Justin', 'Ramirez', '1984-05-08', 'justin.ramirez@example.com', 7776665555, '876 Orange St, City, Country');
```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Michael	Smith	1990-05-15	john.doe@example.com	1234567890	123 Main St, City, Country
2	Jane	Smith	1985-09-20	jane.smith@example.com	9876543210	456 Elm St, City, Country
3	Michael	Johnson	1978-11-12	michael.johnson@example.com	5551234567	789 Oak St, City, Country
4	Emily	Brown	1995-03-25	emily.brown@example.com	9998887776	321 Maple St, City, Country
5	David	Wilson	1980-07-08	david.wilson@example.com	4445556666	654 Pine St, City, Country
6	Sarah	Taylor	1992-12-10	sarah.taylor@example.com	3332221111	876 Cedar St, City, Country
7	Christopher	Martinez	1983-04-18	chris.martinez@example.com	1112223333	432 Birch St, City, Country
8	Jessica	Anderson	1975-08-22	jessica.anderson@example.com	7778889999	567 Walnut St, City, Country
9	Daniel	Garcia	1998-01-05	daniel.garcia@example.com	6667778888	890 Cherry St, City, Country
10	Amanda	Thomas	1987-06-30	amanda.thomas@example.com	2223334444	789 Pineapple St, City, Country
11	Matthew	Hernandez	1993-09-28	matthew.hernandez@example....	5556667777	345 Mango St, City, Country
12	Lauren	King	1981-02-14	lauren.king@example.com	9990001111	234 Peach St, City, Country
13	Ryan	Young	1976-07-19	ryan.young@example.com	4443332222	678 Banana St, City, Country
14	Stephanie	Scott	1990-11-03	stephanie.scott@example.com	8889990000	543 Grape St, City, Country
15	Justin	Ramirez	1984-05-08	justin.ramirez@example.com	7776665555	876 Orange St, City, Country
16	Emily	Brown	1995-03-25	emily.brown@example.com	9998887776	321 Maple St, City, Country

Accounts

```
INSERT INTO Accounts VALUES
(101, 1, 'Savings', 50000),
(102, 2, 'Checking', 10000),
(103, 3, 'Savings', 75000),
(104, 4, 'Checking', 25000),
(105, 5, 'Savings', 100000),
(106, 6, 'Savings', 0),
(107, 7, 'Checking', 20000),
(108, 8, 'Savings', 90000),
(109, 9, 'Checking', 30000),
(110, 10, 'Savings', 0),
(111, 11, 'Checking', 40000),
(112, 12, 'Savings', 95000),
(113, 13, 'Checking', 35000),
(114, 14, 'Savings', 70000),
(115, 15, 'Checking', 50000)
(100, 2, 'Savings', 10000);
```

account_id	customer_id	account_type	balance
101	1	Savings	50000
102	2	Checking	10000
103	3	Savings	75000
104	4	Checking	25000
105	5	Savings	100000
106	6	Savings	60000
107	7	Checking	20000
108	8	Savings	990000
109	9	Checking	30000
110	10	Savings	80000
111	11	Chec	40000
112	12	Savings	95000
113	13	Checking	35000
114	14	Savings	70000
115	15	Checking	50000
NULL	NULL	NULL	NULL

Transactions:

```
INSERT INTO Transactions VALUES
(1001, 101, 'Deposit', 10000, '2024-04-01'),
(1002, 102, 'Withdrawal', 500, '2024-04-02'),
(1003, 103, 'Deposit', 20000, '2024-04-03'),
(1004, 104, 'Withdrawal', 1000, '2024-04-04'),
(1005, 105, 'Deposit', 5000, '2024-04-05'),
(1006, 106, 'Withdrawal', 2000, '2024-04-06'),
(1007, 107, 'Deposit', 1500, '2024-04-07'),
(1008, 108, 'Withdrawal', 3000, '2024-04-08'),
(1009, 109, 'Deposit', 4000, '2024-04-09'),
(1010, 110, 'Withdrawal', 6000, '2024-04-10'),
(1011, 111, 'Deposit', 8000, '2024-04-11'),
(1012, 112, 'Withdrawal', 2000, '2024-04-12'),
(1013, 113, 'Deposit', 10000, '2024-04-13'),
(1014, 114, 'Withdrawal', 3000, '2024-04-14'),
(1015, 115, 'Deposit', 7000, '2024-04-15');
```

transaction_id	account_id	transaction_type	amount	transaction_date
1001	101	Deposit	10000	2024-04-01
1002	102	Withdrawal	500	2024-04-02
1003	103	Deposit	20000	2024-04-03
1004	104	Withdr Withdrawal	000	2024-04-04
1005	105	Deposit	5000	2024-04-05
1006	106	Withdrawal	2000	2024-04-06
1007	107	Deposit	1500	2024-04-07
1008	108	Withdrawal	3000	2024-04-08
1009	109	Deposit	4000	2024-04-09
1010	110	Withdrawal	6000	2024-04-10
1011	111	Deposit	8000	2024-04-11
1012	112	Withdrawal	2000	2024-04-12
1013	113	Deposit	10000	2024-04-13
1014	114	Withdrawal	3000	2024-04-14
1015	115	Deposit	7000	2024-04-15
NULL	NULL	NULL	NULL	NULL

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

```
select concat(first_name," ",last_name) as Name,(select account_type from Accounts where Accounts.customer_id=Customers.customer_id )as type ,email from Customers;
```

Result Grid | Filter Rows: | Export: | Wr

Name	type	email
John Doe	Savings	john.doe@example.com
Jane Smith	Checking	jane.smith@example.com
Michael Johnson	Savings	michael.johnson@example.com
Emily Brown	Checking	emily.brown@example.com
David Wilson	Savings	david.wilson@example.com
Sarah Taylor	Savings	sarah.taylor@example.com
Christopher Martinez	Checking	chris.martinez@example.com
Jessica Anderson	Savings	jessica.anderson@example.com
Daniel Garcia	Checking	daniel.garcia@example.com
Amanda Thomas	Savings	amanda.thomas@example.com
Matthew Hernandez	Checking	matthew.hernandez@example....
Lauren King	Savings	lauren.king@example.com
Ryan Young	Checking	ryan.young@example.com
Stephanie Scott	Savings	stephanie.scott@example.com
Justin Ramirez	Checking	justin.ramirez@example.com

3. Write a SQL query to list all transaction corresponding customer.

```
select concat(c.first_name," ",c.last_name) as Name, t.transaction_type,t.amount from Customers c join Accounts a on c.customer_id=a.customer_id join Transactions t on a.account_id=t.account_id;
```

Result Grid | Filter Rows:

Name	transaction_type	amount
John Doe	Deposit	10000
Jane Smith	Withdrawal	500
Michael Johnson	Deposit	20000
Emily Brown	Withdrawal	1000
David Wilson	Deposit	5000
Sarah Taylor	Withdrawal	2000
Christopher Martinez	Deposit	1500
Jessica Anderson	Withdrawal	3000
Daniel Garcia	Deposit	4000
Amanda Thomas	Withdrawal	6000
Matthew Hernandez	Deposit	8000
Lauren King	Withdrawal	2000
Ryan Young	Deposit	10000
Stephanie Scott	Withdrawal	3000
Justin Ramirez	Deposit	7000

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

```
update Accounts set balance=(balance+150000) where customer_id=8;

select concat(first_name," ",last_name) as Name,(select balance from Accounts where Accounts.customer_id=Customers.customer_id) from Customers;
```

	Name	(select balance from Accounts where Accounts.customer_id=Customers.customer_id)
▶	John Doe	50000
	Jane Smith	10000
	Michael Johnson	75000
	Emily Brown	25000
	David Wilson	100000
	Sarah Taylor	60000
	Christopher Martinez	20000
	Jessica Anderson	540000
	Daniel Garcia	30000
	Amanda Thomas	80000
	Matthew Hernandez	40000
	Lauren King	95000
	Ryan Young	35000
	Stephanie Scott	70000
	Justin Ramirez	50000

60000

5. 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;
```

Result Grid | Filter Rows:

	full_name
▶	John Doe
	Jane Smith
	Michael Johnson
	Emily Brown
	David Wilson
	Sarah Taylor
	Christopher Martinez
	Jessica Anderson
	Daniel Garcia
	Amanda Thomas
	Matthew Hernandez
	Lauren King
	Ryan Young
	Stephanie Scott
	Justin Ramirez

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

```
delete from Transactions where account_id in( Select account_id from Accounts where Balance=0 and account_type="Savings");
delete from Accounts where balance=0 and account_type="Savings";
```

Result Grid | Filter Rows:

full_name	account_type	balance
John Doe	Savings	50000
Jane Smith	Checking	10000
Michael Johnson	Savings	75000
Emily Brown	Checking	25000
David Wilson	Savings	100000
Sarah Taylor	Savings	60000
Christopher Martinez	Checking	20000
Jessica Anderson	Savings	840000
Daniel Garcia	Checking	30000
Amanda Thomas	Savings	80000
Matthew Hernandez	Checking	40000
Lauren King	Savings	95000
Ryan Young	Checking	35000
Stephanie Scott	Savings	70000
Justin Ramirez	Checking	50000

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

```
select concat(first_name," ",last_name) as Name from Customers where address like "%%%%Banana%%%%" or
                address like "%%%%Cherry%%%%" ;
```

Result Grid | Filter

Name
Daniel Garcia
Ryan Young

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

```
select first_name , balance from Customers join Accounts on
Customers.customer_id=Accounts.customer_id where account_id=105;
```

Result Grid | Filter R

first_name	balance
David	100000

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

```
select first_name , account_type , balance from Customers join Accounts on
Customers.customer_id=Accounts.customer_id where account_type="Checking" and balance>1000;
```

Result Grid | Filter Rows:

first_name	account_type	balance
Jane	Checking	10000
Emily	Checking	25000
Christopher	Checking	20000
Daniel	Checking	30000
Matthew	Checking	40000
Ryan	Checking	35000
Justin	Checking	50000

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

```
select first_name , transaction_type , amount , transaction_date from Customers join Accounts on Customers.customer_id=Accounts.customer_id join Transactions on Accounts.account_id=Transactions.account_id where first_name="john";
```

Result Grid | Filter Rows: | Export:

first_name	transaction_type	amount	transaction_date
John	Deposit	10000	2024-04-01

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

```
select concat(first_name," ",last_name) as Name, account_type, balance, balance*0.05 as Interest from Customers join Accounts on Customers.customer_id=Accounts.customer_id;
```

Result Grid | Filter Rows: | Export:

Name	account_type	balance	Interest
John Doe	Savings	50000	2500.00
Jane Smith	Checking	10000	500.00
Michael Johnson	Savings	75000	3750.00
Emily Brown	Checking	25000	1250.00
David Wilson	Savings	100000	5000.00
Sarah Taylor	Savings	60000	3000.00
Christopher Martinez	Checking	20000	1000.00
Jessica Anderson	Savings	990000	49500.00
Daniel Garcia	Checking	30000	1500.00
Amanda Thomas	Savings	80000	4000.00
Matthew Hernandez	Checking	40000	2000.00
Lauren King	Savings	95000	4750.00
Ryan Young	Checking	35000	1750.00
Stephanie Scott	Savings	70000	3500.00
Justin Ramirez	Checking	50000	2500.00

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

```
select concat(first_name," ",last_name) as Name, balance from Customers join Accounts on Customers.customer_id=Accounts.customer_id where balance<50000;
```

	Name	balance
▶	Jane Smith	10000
	Emily Brown	25000
	Christopher Martinez	20000
	Daniel Garcia	30000
	Matthew Hernandez	40000
	Ryan Young	35000

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

```
select concat(first_name," ",last_name) as Name, address from customers where address not like "%Banana%";
```

Result Grid | Filter Rows: | Expo

	Name	address
▶	John Doe	123 Main St, City, Country
	Jane Smith	456 Elm St, City, Country
	Michael Johnson	789 Oak St, City, Country
	Emily Brown	321 Maple St, City, Country
	David Wilson	654 Pine St, City, Country
	Sarah Taylor	876 Cedar St, City, Country
	Christopher Martinez	432 Birch St, City, Country
	Jessica Anderson	567 Walnut St, City, Country
	Daniel Garcia	890 Cherry St, City, Country
	Amanda Thomas	789 Pineapple St, City, Country
	Matthew Hernandez	345 Mango St, City, Country
	Lauren King	234 Peach St, City, Country
	Stephanie Scott	543 Grape St, City, Country
	Justin Ramirez	876 Orange St, City, Country

TASK 3:

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

```
select avg(balance) As Average from Accounts;
```

	Average
▶	115333.3333

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

```
select balance as Top_10 from Accounts order by(balance) desc limit 10 ;
```

Top_10
990000
100000
95000
80000
75000
70000
60000
50000
50000
40000

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

```
select concat(first_name," ",last_name) as Name,transaction_type, amount from Customers join
Accounts on Customers.customer_id=Accounts.customer_id join Transactions on
Accounts.account_id=Transactions.account_id where transaction_date="2024-04-10";
```

Result Grid | Filter Rows:

Name	transaction_type	amount
Amanda Thomas	Withdrawal	6000

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

```
SELECT MAX(CASE WHEN customer_id = min_id THEN CONCAT(first_name, ' ', last_name) END) AS
NEW_CUST, MAX(CASE WHEN customer_id = max_id THEN CONCAT(first_name, ' ', last_name) END) AS
OLD_Cust FROM Customers
CROSS JOIN
  (SELECT MIN(customer_id) AS min_id, MAX(customer_id) AS max_id FROM Customers) AS subquery;
```

NEW_CUST	OLD_Cust
John Doe	Emily Brown

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

```
select concat(first_name," ",last_name) as Name,
transaction_id,transaction_type,amount,transaction_date, account_type from Customers join Accounts
on Customers.customer_id=Accounts.customer_id join Transactions on
Accounts.account_id=Transactions.account_id;
```

Name	transaction_id	transaction_type	amount	transaction_date	account_type
John Doe	1001	Deposit	10000	2024-04-01	Savings
Jane Smith	1002	Withdrawal	500	2024-04-02	Checking
Michael Johnson	1003	Deposit	20000	2024-04-03	Savings
Emily Brown	1004	Withdrawal	1000	2024-04-04	Checking
David Wilson	1005	Deposit	5000	2024-04-05	Savings
Sarah Taylor	1006	Withdrawal	2000	2024-04-06	Savings
Christopher Martinez	1007	Deposit	1500	2024-04-07	Checking
Jessica Anderson	1008	Withdrawal	3000	2024-04-08	Savings
Daniel Garcia	1009	Deposit	4000	2024-04-09	Checking
Amanda Thomas	1010	Withdrawal	6000	2024-04-10	Savings
Matthew Hernandez	1011	Deposit	8000	2024-04-11	Checking
Lauren King	1012	Withdrawal	2000	2024-04-12	Savings
Ryan Young	1013	Deposit	10000	2024-04-13	Checking
Stephanie Scott	1014	Withdrawal	3000	2024-04-14	Savings

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

```
select concat(first_name," ",last_name) as Name, account_id,account_type,balance from Customers
join Accounts on Customers.customer_id=Accounts.customer_id;
```

Result Grid | Filter Rows: | Export:

Name	account_id	account_type	balance
John Doe	101	Savings	50000
Jane Smith	102	Checking	10000
Michael Johnson	103	Savings	75000
Emily Brown	104	Checking	25000
David Wilson	105	Savings	100000
Sarah Taylor	106	Savings	60000
Christopher Martinez	107	Checking	20000
Jessica Anderson	108	Savings	990000
Daniel Garcia	109	Checking	30000
Amanda Thomas	110	Savings	80000
Matthew Hernandez	111	Checking	40000
Lauren King	112	Savings	95000
Ryan Young	113	Checking	35000
Stephanie Scott	114	Savings	70000

Result 14

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

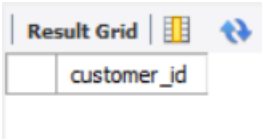
```
select distinct concat(first_name," ",last_name) as Name, Accounts.account_id,
transaction_id,transaction_type,amount,transaction_date from Customers join Accounts on
Customers.customer_id=Accounts.customer_id join Transactions on
Accounts.account_id=Transactions.account_id where Accounts.account_id="108";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	account_id	transaction_id	transaction_type	amount	transaction_date
Jessica Anderson	108	1008	Withdrawal	3000	2024-04-08

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

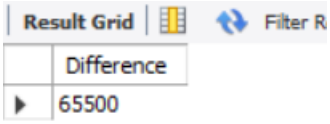
```
select customer_id from Accounts group by customer_id having count(*)>1;
```



customer_id

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

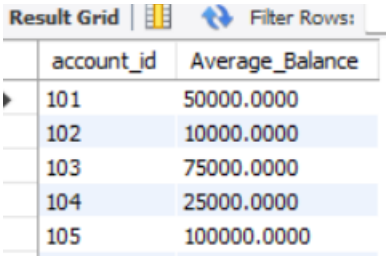
```
SELECT SUM(CASE WHEN transaction_type = 'Deposit' THEN amount
        WHEN transaction_type = 'Withdrawl' THEN -amount
        ELSE 0 END) AS Difference
FROM Transactions;
```



Difference
65500

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

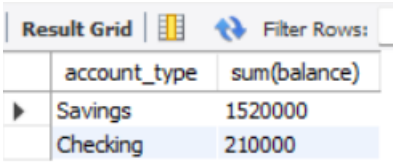
```
select account_id , avg(balance) as Average_Balance from Accounts group by account_id;
```



account_id	Average_Balance
101	50000.0000
102	10000.0000
103	75000.0000
104	25000.0000
105	100000.0000

11. Calculate the total balance for each account type.

```
select account_type , sum(balance) from Accounts group by account_type;
```



account_type	sum(balance)
Savings	1520000
Checking	210000

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

```
select account_id,count(account_id) as no_of_Transactions from Transactions2 group by
account_id order by no_of_Transactions desc;
```

Result Grid | Filter Rows:

account_id	no_of_Transactions
1	4
2	2
3	2
4	2

2

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

```
select concat(first_name," ",last_name), balance , account_type from Customers join Accounts
on Customers.customer_id=Accounts.customer_id group by first_name order by balance desc ;
```

Result Grid | Filter Rows: | Export:

concat(first_name," ",last_name)	balance	account_type
Jessica Anderson	990000	Savings
David Wilson	100000	Savings
Lauren King	95000	Savings
Amanda Thomas	80000	Savings
Stephanie Scott	70000	Savings
Sarah Taylor	60000	Savings

Result 30 × Sarah Taylor

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

```
select account_id, amount,transaction_date, count(*) from Transactions2 group by account_id,
amount,transaction_date having count(*)>1;
```

Result Grid | Filter Rows: | Export

account_id	amount	transaction_date	count(*)

TASK 4

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

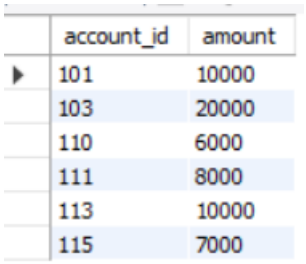
```
select * from Customers where customer_id=(select customer_id from Accounts where balance=(select
max(balance) from Accounts));
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	DOB	email	phone_number	address
8	Jessica	Anderson	1975-08-22	jessica.anderson@example.com	7778889999	567 Walnut St, City, Country
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

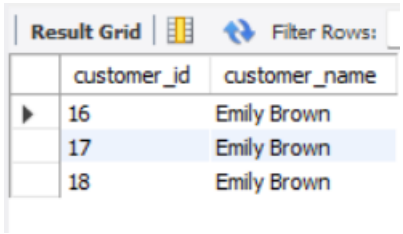
```
select account_id,amount from transactions
where  amount >(select  avg(amount) from transactions);
```



account_id	amount
101	10000
103	20000
110	6000
111	8000
113	10000
115	7000

4. Identify customers who have no recorded transactions.

```
SELECT c.customer_id,CONCAT(c.first_name, ' ', c.last_name) AS customer_name FROM Customers c
LEFT JOIN Accounts a ON c.customer_id = a.customer_id LEFT JOIN Transactions t ON a.account_id =
t.account_id WHERE t.account_id IS NULL;
```

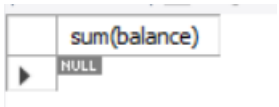


Result Grid | Filter Rows:

customer_id	customer_name
16	Emily Brown
17	Emily Brown
18	Emily Brown

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

```
select sum(balance) from accounts
where account_id not in (select account_id from transactions);
```



sum(balance)
NULL

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));
```

transaction_id	account_id	transaction_type	amount	transaction_date
1002	102	Withdrawal	500	2024-04-02
NULL	NULL	NULL	NULL	NULL

7. Identify customers who have accounts of multiple types.

```
SELECT customer_id,COUNT(DISTINCT account_type) AS num_account_types
FROM Accounts
GROUP BY customer_id
HAVING COUNT(DISTINCT account_type) > 1;
```

Result Grid | Filter Rows:

customer_id	num_account_types

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

```
SELECT account_type,COUNT(*) AS num_accounts,
ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts), 2) AS percentage
FROM Accounts
GROUP BY account_type;
```

account_type	num_accounts	percentage
Savings	8	53.33
Checking	7	46.67

9. Retrieve all transactions for a customer with a given customer_id.

```
SELECT * FROM Transactions WHERE  account_id IN (SELECT account_id FROM Accounts WHERE
customer_id = 3 );
```

transaction_id	account_id	transaction_type	amount	transaction_date
1003	103	Deposit	20000	2024-04-03
NULL	NULL	NULL	NULL	NULL

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

```
SELECT account_type,
(SELECT SUM(balance) FROM Accounts WHERE account_type = a.account_type) AS total_balance
FROM (SELECT DISTINCT account_type FROM Accounts) AS a;
```

account_type	total_balance
Savings	1520000
Checking	210000