TASK:1

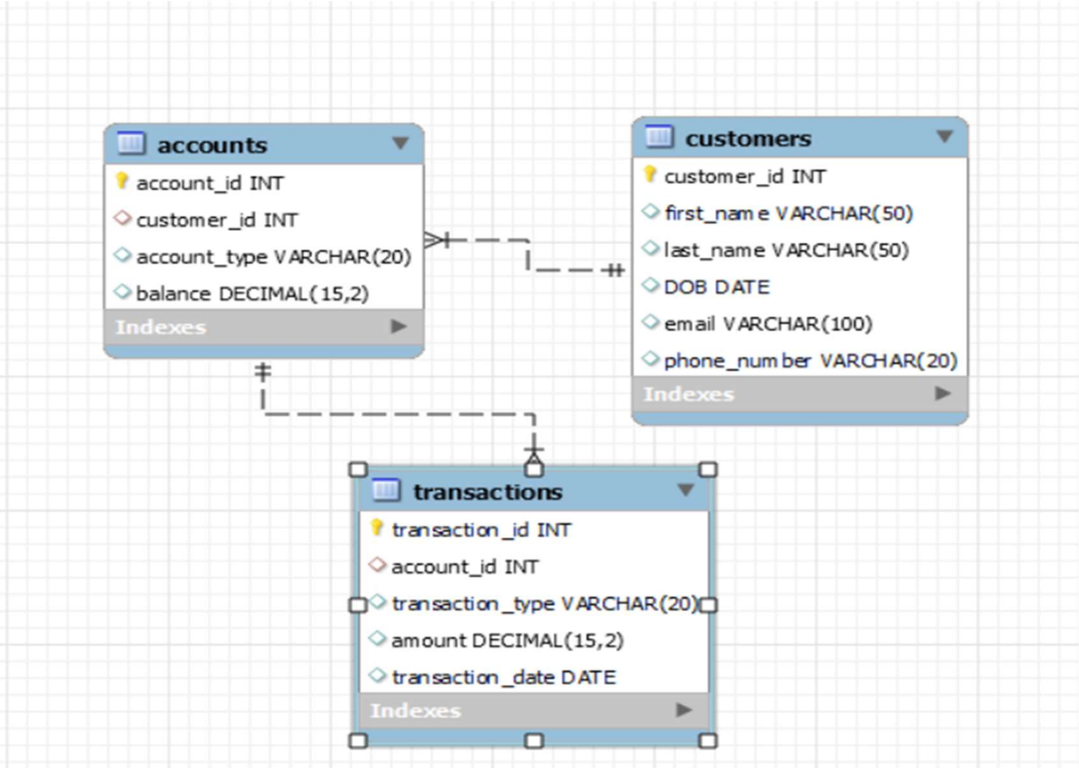
Query1;

```
create database HMBank;
```

Query2;

```
CREATE TABLE Customers (
   customer_id INT PRIMARY KEY,
   first_name VARCHAR(50) ,
   last_name VARCHAR(50) ,
   DOB date,
   email VARCHAR(100) ,
   phone_number VARCHAR(20)

);



CREATE TABLE Accounts (
   account_id INT PRIMARY KEY,
   customer_id INT,
   account_type VARCHAR(20),
   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(20) ,
   amount DECIMAL(15, 2) ,
    transaction_date date,
    FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
);
```

Query3;



accounts
account_id INT
customer_id INT
account_type VARCHAR(20)
balance DECIMAL(15,2)
Indexes

customers
customer_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
DOB DATE
email VARCHAR(100)
phone_number VARCHAR(20)
Indexes

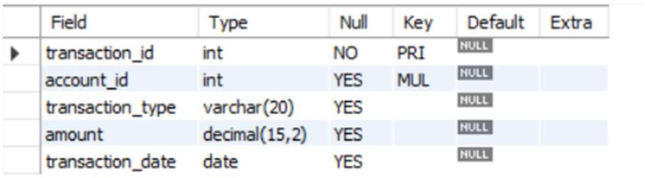
transactions
transaction_id INT
account_id INT
transaction_type VARCHAR(20)
amount DECIMAL(15,2)
transaction_date DATE
Indexes

Query4;

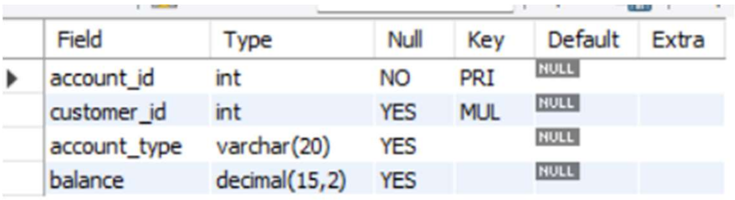
desc Customers;

desc Accounts;

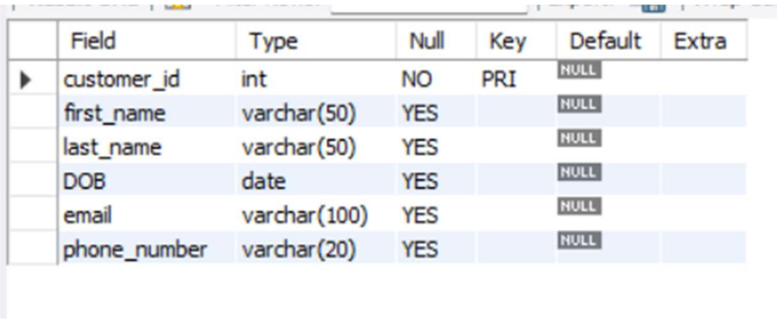
desc Transactions;



	Field	Type	Null	Key	Default	Extra
▶	transaction_id	int	NO	PRI	NULL	
	account_id	int	YES	MUL	NULL	
	transaction_type	varchar(20)	YES		NULL	
	amount	decimal(15,2)	YES		NULL	
	transaction_date	date	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	decimal(15,2)	YES		NULL	

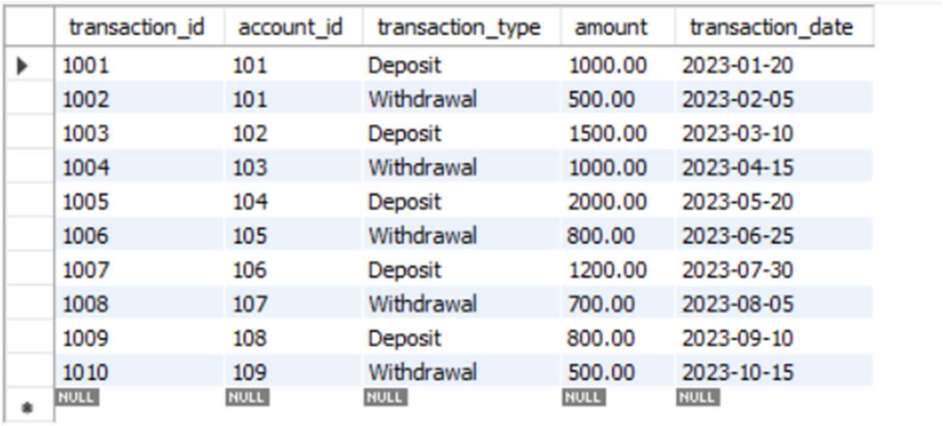


	Field	Type	Null	Key	Default	Extra
▶	customer_id	int	NO	PRI	NULL	
	first_name	varchar(50)	YES		NULL	
	last_name	varchar(50)	YES		NULL	
	DOB	date	YES		NULL	
	email	varchar(100)	YES		NULL	
	phone_number	varchar(20)	YES		NULL	

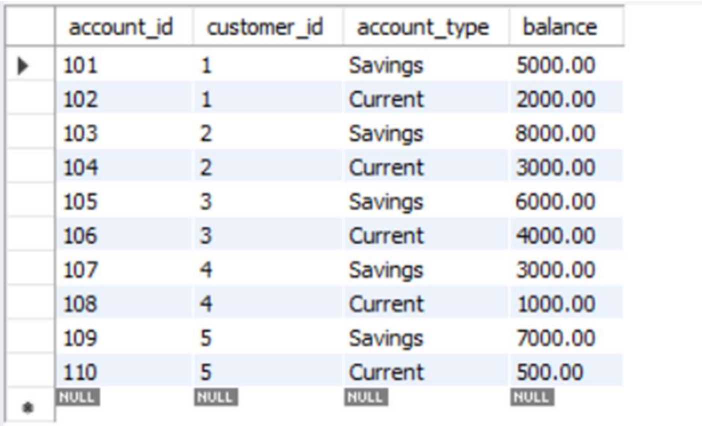
TASK 2

-- Query 1;

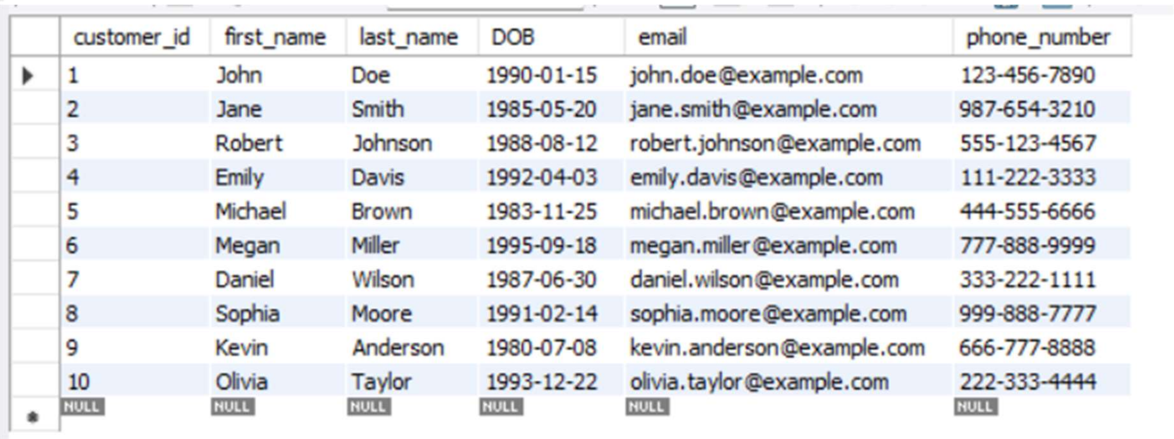
```
INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number)
VALUES
    (1, 'John', 'Doe', '1990-01-15', 'john.doe@example.com', '123-456-7890'),
    (2, 'Jane', 'Smith', '1985-05-20', 'jane.smith@example.com', '987-654-3210'),
    (3, 'Robert', 'Johnson', '1988-08-12', 'robert.johnson@example.com', '555-123-4567'),
    (4, 'Emily', 'Davis', '1992-04-03', 'emily.davis@example.com', '111-222-3333'),
    (5, 'Michael', 'Brown', '1983-11-25', 'michael.brown@example.com', '444-555-6666'),
    (6, 'Megan', 'Miller', '1995-09-18', 'megan.miller@example.com', '777-888-9999'),
    (7, 'Daniel', 'Wilson', '1987-06-30', 'daniel.wilson@example.com', '333-222-1111'),
    (8, 'Sophia', 'Moore', '1991-02-14', 'sophia.moore@example.com', '999-888-7777'),
    (9, 'Kevin', 'Anderson', '1980-07-08', 'kevin.anderson@example.com', '666-777-8888'),
    (10, 'Olivia', 'Taylor', '1993-12-22', 'olivia.taylor@example.com', '222-333-4444');

-- Insert 10 sample records into Accounts
INSERT INTO Accounts (account_id, customer_id, account_type, balance)
VALUES
    (101, 1, 'Savings', 5000.00),
    (102, 1, 'Checking', 2000.00),
    (103, 2, 'Savings', 8000.00),
    (104, 2, 'Checking', 3000.00),
    (105, 3, 'Savings', 6000.00),
    (106, 3, 'Checking', 4000.00),
    (107, 4, 'Savings', 3000.00),
    (108, 4, 'Checking', 1000.00),
    (109, 5, 'Savings', 7000.00),
    (110, 5, 'Checking', 500.00);

-- Insert 10 sample records into Transactions
INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
VALUES
    (1001, 101, 'Deposit', 1000.00, '2023-01-20'),
    (1002, 101, 'Withdrawal', 500.00, '2023-02-05'),
    (1003, 102, 'Deposit', 1500.00, '2023-03-10'),
    (1004, 103, 'Withdrawal', 1000.00, '2023-04-15'),
    (1005, 104, 'Deposit', 2000.00, '2023-05-20'),
    (1006, 105, 'Withdrawal', 800.00, '2023-06-25'),
    (1007, 106, 'Deposit', 1200.00, '2023-07-30'),
    (1008, 107, 'Withdrawal', 700.00, '2023-08-05'),
    (1009, 108, 'Deposit', 800.00, '2023-09-10'),
    (1010, 109, 'Withdrawal', 500.00, '2023-10-15');
   -- Update sample data for the Accounts table
UPDATE Accounts
SET account_type = 'Savings'
WHERE account_id IN (101, 103, 105, 107, 109);

UPDATE Accounts
SET account_type = 'Current'
WHERE account_id IN (102, 104, 106, 108, 110);

UPDATE Accounts
SET account_type = 'Zero_Balance'
WHERE account_id = 111;

-- Update sample data for the Transactions table
UPDATE Transactions
SET transaction_type = 'Deposit'
WHERE transaction_id IN (1001, 1003, 1005, 1007, 1009);

UPDATE Transactions
SET transaction_type = 'Withdrawal'
WHERE transaction_id IN (1002, 1004, 1006, 1008, 1010);

UPDATE Transactions
SET transaction_type = 'Transfer'
WHERE transaction_id = 1011;

   select * from  Customers;
    select * from  Accounts;
     select * from  Transactions;
```



transaction_id	account_id	transaction_type	amount	transaction_date
1001	101	Deposit	1000.00	2023-01-20
1002	101	Withdrawal	500.00	2023-02-05
1003	102	Deposit	1500.00	2023-03-10
1004	103	Withdrawal	1000.00	2023-04-15
1005	104	Deposit	2000.00	2023-05-20
1006	105	Withdrawal	800.00	2023-06-25
1007	106	Deposit	1200.00	2023-07-30
1008	107	Withdrawal	700.00	2023-08-05
1009	108	Deposit	800.00	2023-09-10
1010	109	Withdrawal	500.00	2023-10-15
NULL	NULL	NULL	NULL	NULL



account_id	customer_id	account_type	balance
101	1	Savings	5000.00
102	1	Current	2000.00
103	2	Savings	8000.00
104	2	Current	3000.00
105	3	Savings	6000.00
106	3	Current	4000.00
107	4	Savings	3000.00
108	4	Current	1000.00
109	5	Savings	7000.00
110	5	Current	500.00
NULL	NULL	NULL	NULL



customer_id	first_name	last_name	DOB	email	phone_number
1	John	Doe	1990-01-15	john.doe@example.com	123-456-7890
2	Jane	Smith	1985-05-20	jane.smith@example.com	987-654-3210
3	Robert	Johnson	1988-08-12	robert.johnson@example.com	555-123-4567
4	Emily	Davis	1992-04-03	emily.davis@example.com	111-222-3333
5	Michael	Brown	1983-11-25	michael.brown@example.com	444-555-6666
6	Megan	Miller	1995-09-18	megan.miller@example.com	777-888-9999
7	Daniel	Wilson	1987-06-30	daniel.wilson@example.com	333-222-1111
8	Sophia	Moore	1991-02-14	sophia.moore@example.com	999-888-7777
9	Kevin	Anderson	1980-07-08	kevin.anderson@example.com	666-777-8888
10	Olivia	Taylor	1993-12-22	olivia.taylor@example.com	222-333-4444
NULL	NULL	NULL	NULL	NULL	NULL

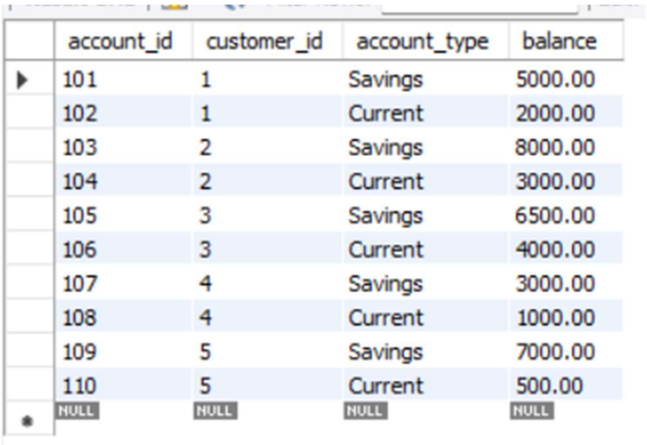
```
-- Query 2-1;

SELECT

   CONCAT(first_name, ' ', last_name) AS full_name,

   account_type,

   email

FROM Customers

JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
```

-- Query 2-2;

```
SELECT
   c.customer_id,
   CONCAT(c.first_name, ' ', c.last_name) AS full_name,
   t.transaction_id,
   t.transaction_type,
   t.amount,
   t.transaction_date
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id
JOIN Transactions t ON a.account_id = t.account_id;
```

-- Query 2-3;

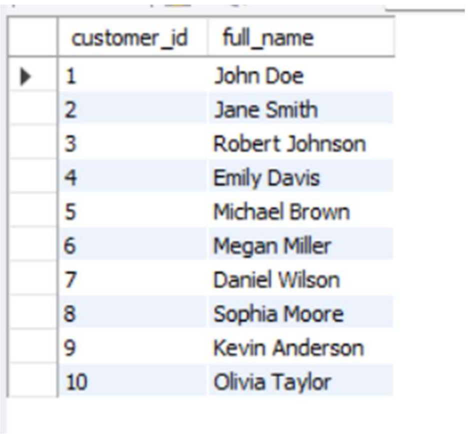
```
UPDATE Accounts
SET balance = balance + 500
WHERE account_id = 105;
select * from Accounts;
```



account_id	customer_id	account_type	balance
101	1	Savings	5000.00
102	1	Current	2000.00
103	2	Savings	8000.00
104	2	Current	3000.00
105	3	Savings	6500.00
106	3	Current	4000.00
107	4	Savings	3000.00
108	4	Current	1000.00
109	5	Savings	7000.00
110	5	Current	500.00
NULL	NULL	NULL	NULL

-- Query 2- 4;

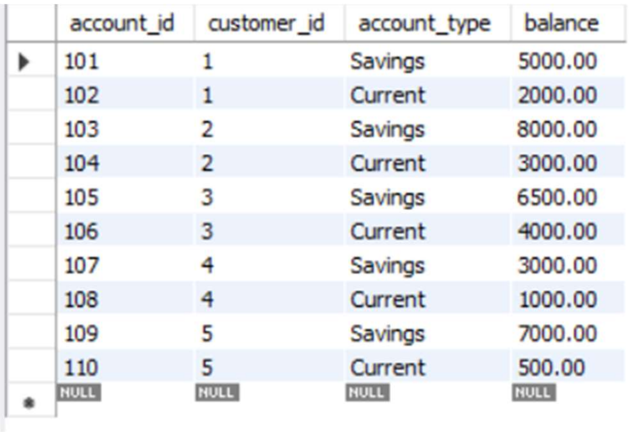
```
SELECT
    customer_id,
    CONCAT(first_name, ' ', last_name) AS full_name
FROM Customers;
```



customer_id	full_name
1	John Doe
2	Jane Smith
3	Robert Johnson
4	Emily Davis
5	Michael Brown
6	Megan Miller
7	Daniel Wilson
8	Sophia Moore
9	Kevin Anderson
10	Olivia Taylor

-- Query 2- 5;

```
DELETE FROM Accounts
WHERE balance = 0 AND account_type = 'Savings'AND account_id > 0;
select * from Accounts;
```



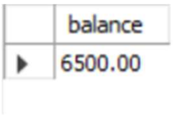
account_id	customer_id	account_type	balance
101	1	Savings	5000.00
102	1	Current	2000.00
103	2	Savings	8000.00
104	2	Current	3000.00
105	3	Savings	6500.00
106	3	Current	4000.00
107	4	Savings	3000.00
108	4	Current	1000.00
109	5	Savings	7000.00
110	5	Current	500.00
NULL	NULL	NULL	NULL

-- Query 2-6;

```
SELECT *
FROM Customers
WHERE city = 'YourCity';
```

-- Query 2-7;

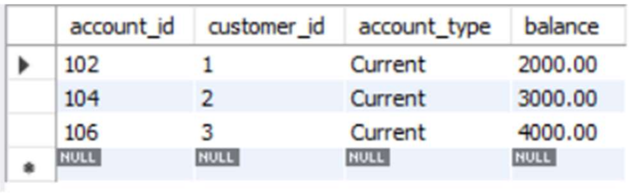
```
SELECT balance

FROM Accounts

WHERE account_id = 105;
```



balance
6500.00

-- Query 2-8;

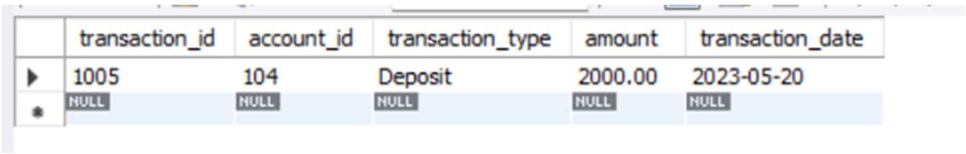
```
SELECT *

FROM Accounts

WHERE account_type = 'Current' AND balance > 1000;
```



account_id	customer_id	account_type	balance
102	1	Current	2000.00
104	2	Current	3000.00
106	3	Current	4000.00
NULL	NULL	NULL	NULL

-- Query 2-9;

```
SELECT *

FROM Transactions

WHERE account_id = 104;
```

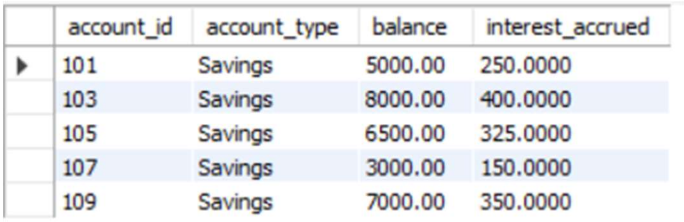


transaction_id	account_id	transaction_type	amount	transaction_date
1005	104	Deposit	2000.00	2023-05-20
NULL	NULL	NULL	NULL	NULL

-- Query 2-10;

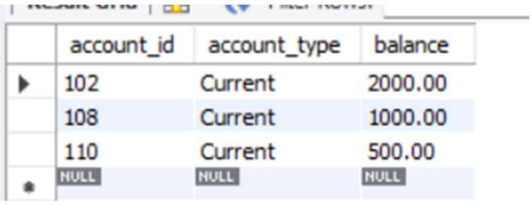
```
-- Assuming interest_rate is given as a decimal (e.g., 0.05 for 5%)

SELECT

    account_id,

    account_type,

    balance,

    balance * 0.05 AS interest_accrued
```

```
FROM

  Accounts

WHERE

  account_type = 'Savings';
```



	account_id	account_type	balance	interest_accrued
▶	101	Savings	5000.00	250.0000
	103	Savings	8000.00	400.0000
	105	Savings	6500.00	325.0000
	107	Savings	3000.00	150.0000
	109	Savings	7000.00	350.0000

```
-- Query 2-11;

-- Assuming overdraft_limit is the specified limit

SELECT

  account_id,

  account_type,

  balance

FROM

  Accounts

WHERE

  balance < 3000;
```



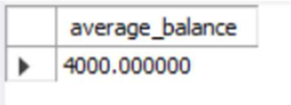
	account_id	account_type	balance
▶	102	Current	2000.00
	108	Current	1000.00
	110	Current	500.00
*	NULL	NULL	NULL

```
-- Query 2-12;

-- Assuming 'YourCity' is the city to be excluded

SELECT

  customer_id,

  first_name,

  last_name,

  email,
  phone_number,
  address
FROM
  Customers
WHERE
  city <> 'YourCity' OR city IS NULL;
```

Task 3:

-- 1. Find the average account balance for all customers:

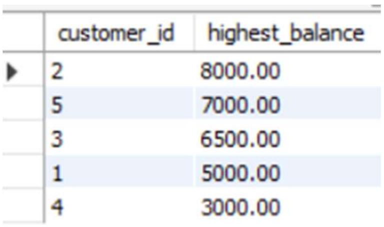
```
SELECT AVG(balance) AS average_balance
FROM Accounts;
```



	average_balance
▶	4000.000000

-- 2. Retrieve the top 10 highest account balances:

```
SELECT customer_id, MAX(balance) AS highest_balance
FROM Accounts
GROUP BY customer_id
ORDER BY highest_balance DESC
LIMIT 10;
```

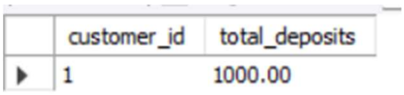


	customer_id	highest_balance
▶	2	8000.00
	5	7000.00
	3	6500.00
	1	5000.00
	4	3000.00

-- 3. Calculate Total Deposits for All Customers on a specific date:

```
SELECT customer_id, SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) AS
total_deposits
```

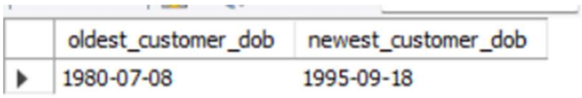
```
FROM Transactions t

JOIN Accounts a ON t.account_id = a.account_id

WHERE transaction_date = '2023-01-20'

GROUP BY customer_id;
```



	customer_id	total_deposits
▶	1	1000.00

-- 4. Find the Oldest and Newest Customers:

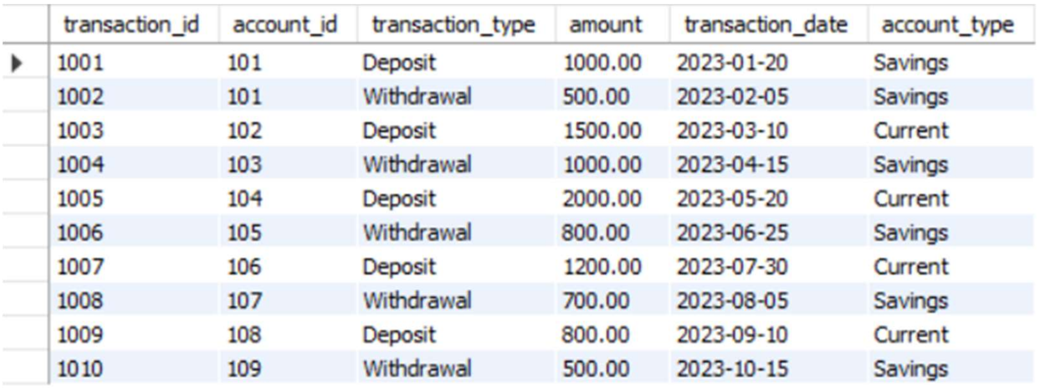
```
SELECT MIN(DOB) AS oldest_customer_dob, MAX(DOB) AS newest_customer_dob

FROM Customers;
```



	oldest_customer_dob	newest_customer_dob
▶	1980-07-08	1995-09-18

-- 5. Retrieve transaction details along with the account type:

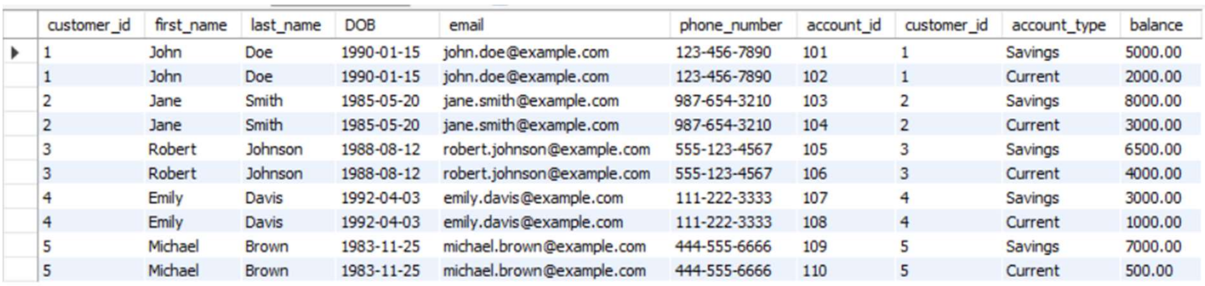
```
SELECT t.*, a.account_type

FROM Transactions t

JOIN Accounts a ON t.account_id = a.account_id;
```



	transaction_id	account_id	transaction_type	amount	transaction_date	account_type
▶	1001	101	Deposit	1000.00	2023-01-20	Savings
	1002	101	Withdrawal	500.00	2023-02-05	Savings
	1003	102	Deposit	1500.00	2023-03-10	Current
	1004	103	Withdrawal	1000.00	2023-04-15	Savings
	1005	104	Deposit	2000.00	2023-05-20	Current
	1006	105	Withdrawal	800.00	2023-06-25	Savings
	1007	106	Deposit	1200.00	2023-07-30	Current
	1008	107	Withdrawal	700.00	2023-08-05	Savings
	1009	108	Deposit	800.00	2023-09-10	Current
	1010	109	Withdrawal	500.00	2023-10-15	Savings

-- 6. Get a list of customers along with their account details:

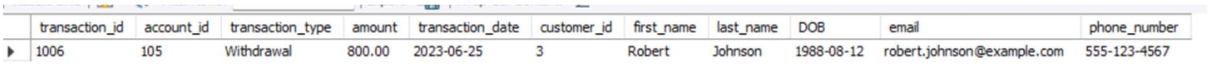
```
SELECT c.*, a.*

FROM Customers c

JOIN Accounts a ON c.customer_id = a.customer_id;
```



customer_id	first_name	last_name	DOB	email	phone_number	account_id	customer_id	account_type	balance
1	John	Doe	1990-01-15	john.doe@example.com	123-456-7890	101	1	Savings	5000.00
1	John	Doe	1990-01-15	john.doe@example.com	123-456-7890	102	1	Current	2000.00
2	Jane	Smith	1985-05-20	jane.smith@example.com	987-654-3210	103	2	Savings	8000.00
2	Jane	Smith	1985-05-20	jane.smith@example.com	987-654-3210	104	2	Current	3000.00
3	Robert	Johnson	1988-08-12	robert.johnson@example.com	555-123-4567	105	3	Savings	6500.00
3	Robert	Johnson	1988-08-12	robert.johnson@example.com	555-123-4567	106	3	Current	4000.00
4	Emily	Davis	1992-04-03	emily.davis@example.com	111-222-3333	107	4	Savings	3000.00
4	Emily	Davis	1992-04-03	emily.davis@example.com	111-222-3333	108	4	Current	1000.00
5	Michael	Brown	1983-11-25	michael.brown@example.com	444-555-6666	109	5	Savings	7000.00
5	Michael	Brown	1983-11-25	michael.brown@example.com	444-555-6666	110	5	Current	500.00

-- 7. Retrieve transaction details along with customer information for a specific account:

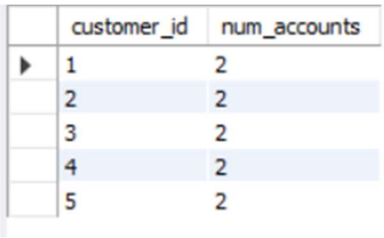
```
SELECT t.*, c.*

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 = 105
```



transaction_id	account_id	transaction_type	amount	transaction_date	customer_id	first_name	last_name	DOB	email	phone_number
1006	105	Withdrawal	800.00	2023-06-25	3	Robert	Johnson	1988-08-12	robert.johnson@example.com	555-123-4567

-- 8. Identify customers who have more than one account:

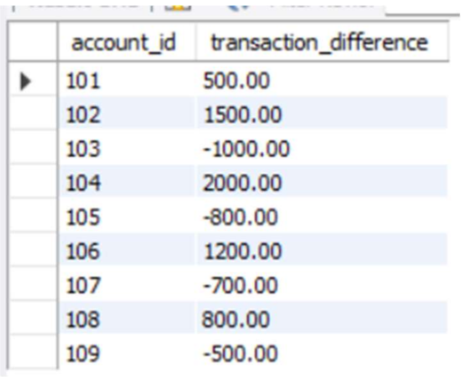
```
SELECT customer_id, COUNT(account_id) AS num_accounts

FROM Accounts

GROUP BY customer_id

HAVING num_accounts > 1;
```



customer_id	num_accounts
1	2
2	2
3	2
4	2
5	2

-- 9. Calculate the difference in transaction amounts between deposits and withdrawals:

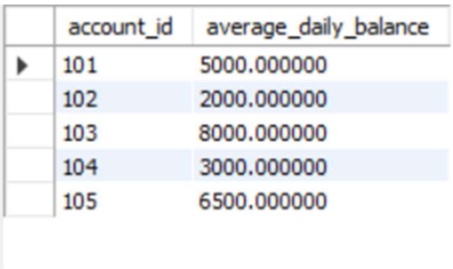
```
SELECT account_id, SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END)
AS transaction_difference

FROM Transactions

GROUP BY account_id;
```



account_id	transaction_difference
101	500.00
102	1500.00
103	-1000.00
104	2000.00
105	-800.00
106	1200.00
107	-700.00
108	800.00
109	-500.00

-- 10. Calculate the average daily balance for each account over a specified period:

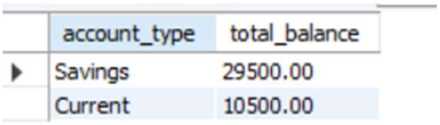
```
SELECT A.account_id, AVG(A.balance) AS average_daily_balance

FROM Accounts A

JOIN Transactions T ON A.account_id = T.account_id

WHERE DATE(T.transaction_date) BETWEEN '2023-01-20' AND '2023-06-25'

GROUP BY A.account_id

LIMIT 0, 1000;
```



account_id	average_daily_balance
101	5000.000000
102	2000.000000
103	8000.000000
104	3000.000000
105	6500.000000

-- 11. Calculate the total balance for each account type:

```
SELECT account_type, SUM(balance) AS total_balance

FROM Accounts

GROUP BY account_type;
```

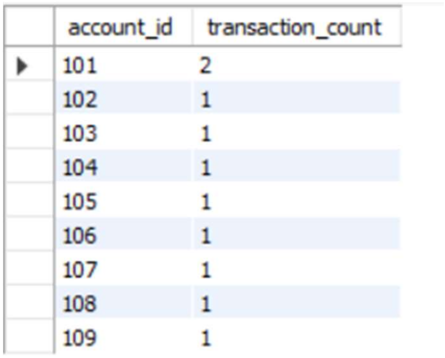


account_type	total_balance
Savings	29500.00
Current	10500.00

-- 12. Identify accounts with the highest number of transactions in descending order:

```
SELECT account_id, COUNT(transaction_id) AS transaction_count

FROM Transactions
```

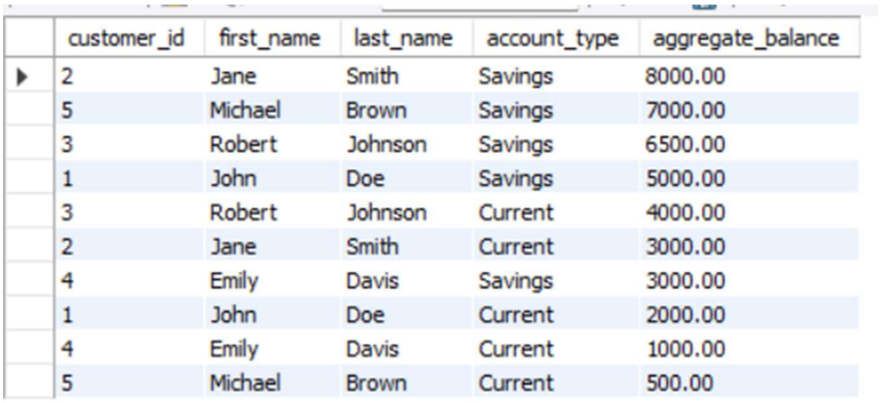
```
GROUP BY account_id

ORDER BY transaction_count DESC;
```



account_id	transaction_count
101	2
102	1
103	1
104	1
105	1
106	1
107	1
108	1
109	1

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

```
SELECT c.customer_id, c.first_name, c.last_name, a.account_type, SUM(a.balance) AS
aggregate_balance

FROM Customers c

JOIN Accounts a ON c.customer_id = a.customer_id

GROUP BY c.customer_id, c.first_name, c.last_name, a.account_type

ORDER BY aggregate_balance DESC;
```

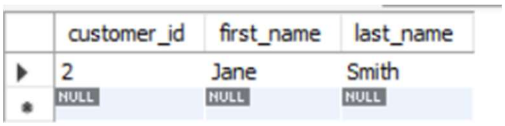


customer_id	first_name	last_name	account_type	aggregate_balance
2	Jane	Smith	Savings	8000.00
5	Michael	Brown	Savings	7000.00
3	Robert	Johnson	Savings	6500.00
1	John	Doe	Savings	5000.00
3	Robert	Johnson	Current	4000.00
2	Jane	Smith	Current	3000.00
4	Emily	Davis	Savings	3000.00
1	John	Doe	Current	2000.00
4	Emily	Davis	Current	1000.00
5	Michael	Brown	Current	500.00

Task 4

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

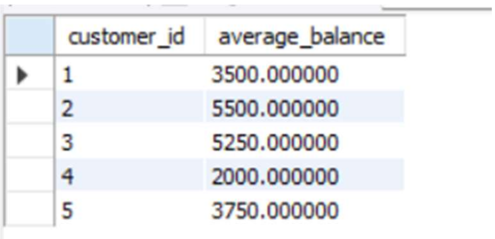
```
SELECT customer_id, first_name, last_name

FROM Customers

WHERE customer_id = (SELECT customer_id FROM Accounts ORDER BY balance DESC LIMIT 1);
```



customer_id	first_name	last_name
2	Jane	Smith
NULL	NULL	NULL

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

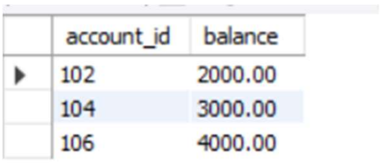
```
SELECT customer_id, AVG(balance) AS average_balance

FROM Accounts

GROUP BY customer_id

HAVING COUNT(account_id) > 1;
```



customer_id	average_balance
1	3500.000000
2	5500.000000
3	5250.000000
4	2000.000000
5	3750.000000

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

```
SELECT a.account_id, a.balance

FROM Accounts a

JOIN Transactions t ON a.account_id = t.account_id

WHERE t.amount > (SELECT AVG(amount) FROM Transactions);
```



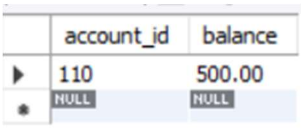
account_id	balance
102	2000.00
104	3000.00
106	4000.00

-- 4. Identify customers who have no recorded transactions:

```
SELECT customer_id, first_name, last_name

FROM Customers

WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Transactions);
```

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

```
SELECT account_id, balance

FROM Accounts

WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);
```



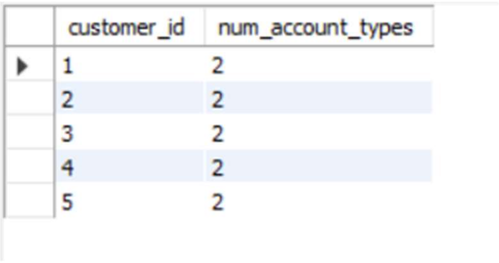
account_id	balance
110	500.00
NULL	NULL

-- 6. Retrieve transactions for accounts with the lowest balance:

```
SELECT t.*

FROM Transactions t

JOIN Accounts a ON t.account_id = a.account_id

WHERE a.balance = (SELECT MIN(balance) FROM Accounts);
```

-- 7. Identify customers who have accounts of multiple types:

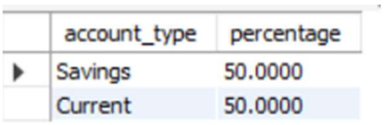
```
SELECT customer_id, COUNT(DISTINCT account_type) AS num_account_types

FROM Accounts

GROUP BY customer_id

HAVING num_account_types > 1;
```



customer_id	num_account_types
1	2
2	2
3	2
4	2
5	2

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

```
SELECT account_type, COUNT(account_id) * 100 / (SELECT COUNT(account_id) FROM Accounts) AS percentage

FROM Accounts

GROUP BY account_type;
```



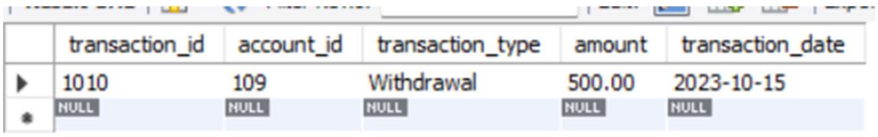
account_type	percentage
Savings	50.0000
Current	50.0000

-- 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 = <given_customer_id>);



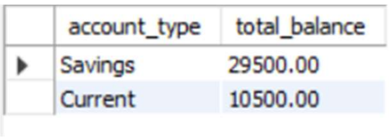
transaction_id	account_id	transaction_type	amount	transaction_date
1010	109	Withdrawal	500.00	2023-10-15
NULL	NULL	NULL	NULL	NULL

-- 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 Accounts a

GROUP BY account_type;



account_type	total_balance
Savings	29500.00
Current	10500.00