Assignment - 3 :

Task-1:

1.

```
create database HMBank;
```

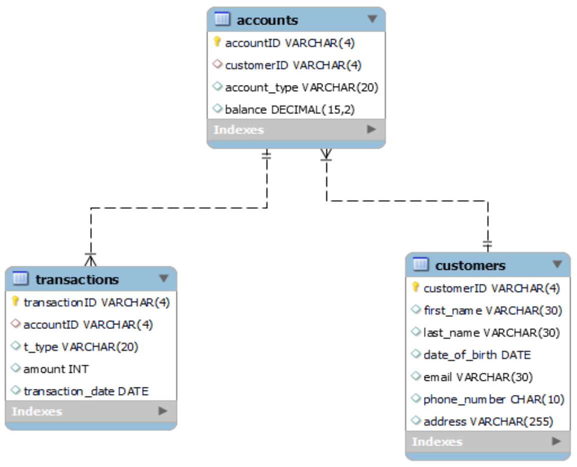
```
1  17:48:05  create database HMBank                                   1 row(s) affected                    0.000 sec
```

2.

```
create table customers(
    customerID varchar(4) primary key,
    first_name varchar(30),
    last_name varchar(30),
    date_of_birth date,
    email varchar(30) unique,
    phone_number char(10) unique,
    address varchar(255)
);

create table accounts(
    accountID varchar(4) primary key,
    customerID varchar(4),
    account_type varchar(20),
    balance decimal(15, 2),
    foreign key(customerID) references customers(customerID) on delete cascade
);

create table Transactions(
    transactionID varchar(4) primary key,
    accountID varchar(4),
    t_type varchar(20),
    amount int,
    transaction_date date,
    foreign key(accountID) references accounts(accountID) on delete cascade
);
```

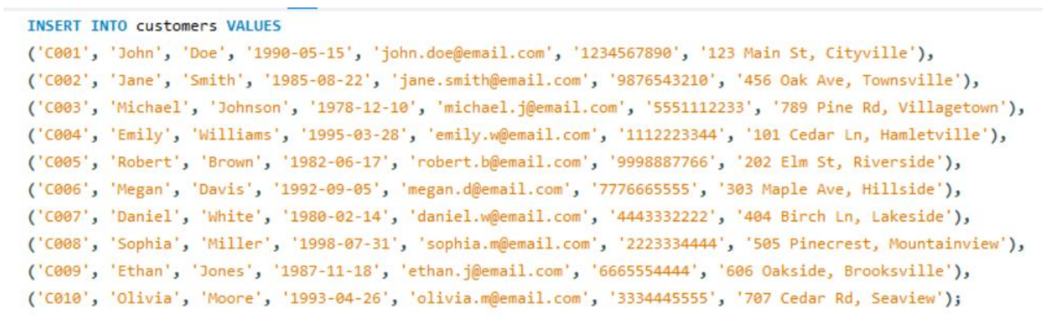
```
3  17:55:54  create table customers( customerID varchar(4) primary key,   first_name varchar(30),   last_name varchar(30),...  0 row(s) affected   0.047 sec
4  17:56:11  create table accounts( accountID varchar(4) primary key,   customerID varchar(4),   account_type varchar(2...   0 row(s) affected   0.031 sec
5  17:56:15  create table Transactions(transactionID varchar(4) primary key,   accountID varchar(4),   t_type varchar(20),  ...  0 row(s) affected   0.032 sec
```

3.

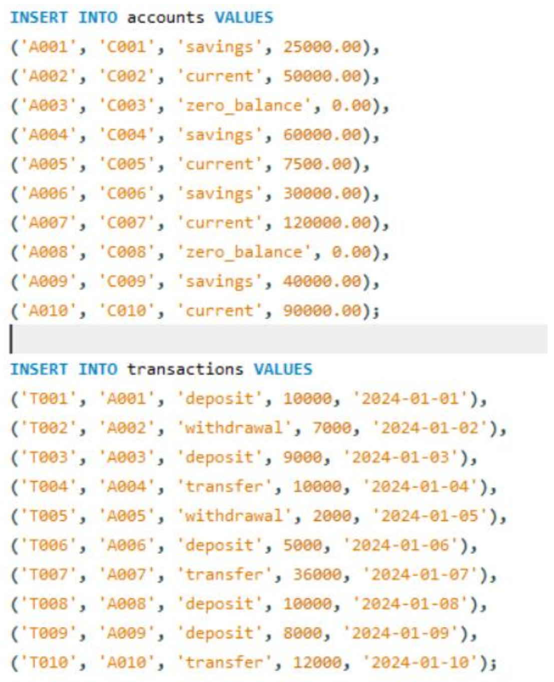


Task-2:

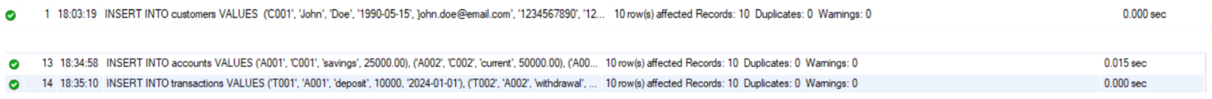
1.



```sql
INSERT INTO customers VALUES
('C001', 'John', 'Doe', '1990-05-15', 'john.doe@email.com', '1234567890', '123 Main St, Cityville'),
('C002', 'Jane', 'Smith', '1985-08-22', 'jane.smith@email.com', '9876543210', '456 Oak Ave, Townsville'),
('C003', 'Michael', 'Johnson', '1978-12-10', 'michael.j@email.com', '5551112233', '789 Pine Rd, Villagetown'),
('C004', 'Emily', 'Williams', '1995-03-28', 'emily.w@email.com', '1112223344', '101 Cedar Ln, Hamletville'),
('C005', 'Robert', 'Brown', '1982-06-17', 'robert.b@email.com', '9998887766', '202 Elm St, Riverside'),
('C006', 'Megan', 'Davis', '1992-09-05', 'megan.d@email.com', '7776665555', '303 Maple Ave, Hillside'),
('C007', 'Daniel', 'White', '1980-02-14', 'daniel.w@email.com', '4443332222', '404 Birch Ln, Lakeside'),
('C008', 'Sophia', 'Miller', '1998-07-31', 'sophia.m@email.com', '2223334444', '505 Pinecrest, Mountainview'),
('C009', 'Ethan', 'Jones', '1987-11-18', 'ethan.j@email.com', '6665554444', '606 Oakside, Brooksville'),
('C010', 'Olivia', 'Moore', '1993-04-26', 'olivia.m@email.com', '3334445555', '707 Cedar Rd, Seaview');
```

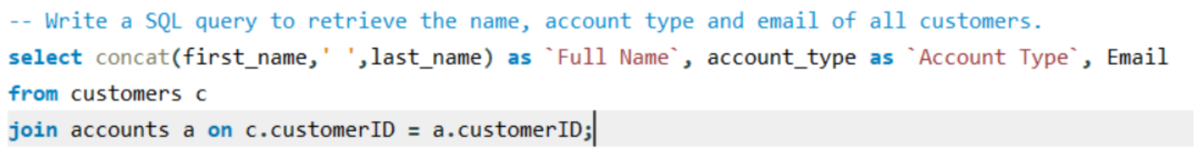


```sql
INSERT INTO accounts VALUES
('A001', 'C001', 'savings', 25000.00),
('A002', 'C002', 'current', 50000.00),
('A003', 'C003', 'zero_balance', 0.00),
('A004', 'C004', 'savings', 60000.00),
('A005', 'C005', 'current', 7500.00),
('A006', 'C006', 'savings', 30000.00),
('A007', 'C007', 'current', 120000.00),
('A008', 'C008', 'zero_balance', 0.00),
('A009', 'C009', 'savings', 40000.00),
('A010', 'C010', 'current', 90000.00);

INSERT INTO transactions VALUES
('T001', 'A001', 'deposit', 10000, '2024-01-01'),
('T002', 'A002', 'withdrawal', 7000, '2024-01-02'),
('T003', 'A003', 'deposit', 9000, '2024-01-03'),
('T004', 'A004', 'transfer', 10000, '2024-01-04'),
('T005', 'A005', 'withdrawal', 2000, '2024-01-05'),
('T006', 'A006', 'deposit', 5000, '2024-01-06'),
('T007', 'A007', 'transfer', 36000, '2024-01-07'),
('T008', 'A008', 'deposit', 10000, '2024-01-08'),
('T009', 'A009', 'deposit', 8000, '2024-01-09'),
('T010', 'A010', 'transfer', 12000, '2024-01-10');
```

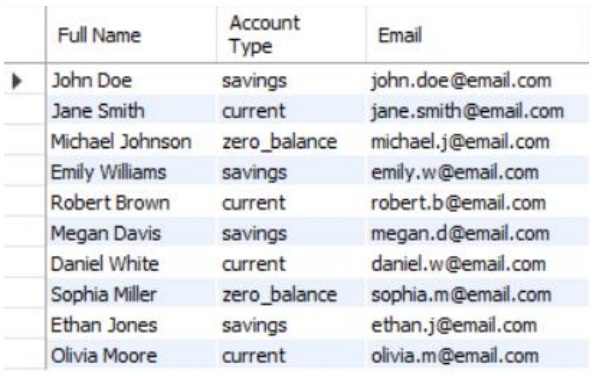


```
1   18:03:19  INSERT INTO customers VALUES  ('C001', 'John', 'Doe', '1990-05-15', 'john.doe@email.com', '1234567890', '12...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0    0.000 sec

13  18:34:58  INSERT INTO accounts VALUES ('A001', 'C001', 'savings', 25000.00), ('A002', 'C002', 'current', 50000.00), ('A00...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0    0.015 sec
14  18:35:10  INSERT INTO transactions VALUES ('T001', 'A001', 'deposit', 10000, '2024-01-01'), ('T002', 'A002', 'withdrawal', ...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0    0.000 sec
```

2.

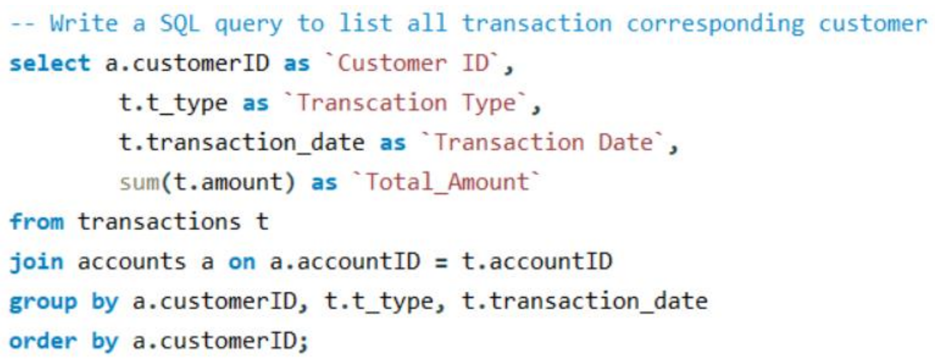


```sql
-- Write a SQL query to retrieve the name, account type and email of all customers.
select concat(first_name,' ',last_name) as `Full Name`, account_type as `Account Type`, Email
from customers c
join accounts a on c.customerID = a.customerID;
```

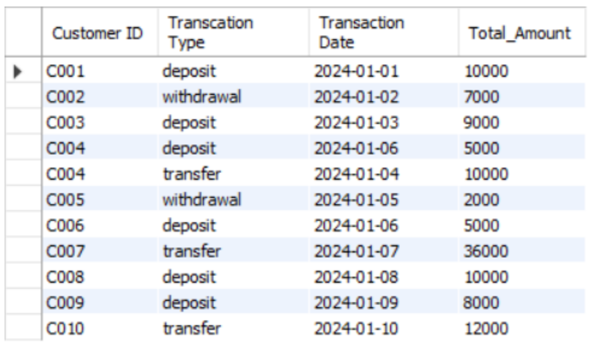


Full Name	Account Type	Email
John Doe	savings	john.doe@email.com
Jane Smith	current	jane.smith@email.com
Michael Johnson	zero_balance	michael.j@email.com
Emily Williams	savings	emily.w@email.com
Robert Brown	current	robert.b@email.com
Megan Davis	savings	megan.d@email.com
Daniel White	current	daniel.w@email.com
Sophia Miller	zero_balance	sophia.m@email.com
Ethan Jones	savings	ethan.j@email.com
Olivia Moore	current	olivia.m@email.com

3.

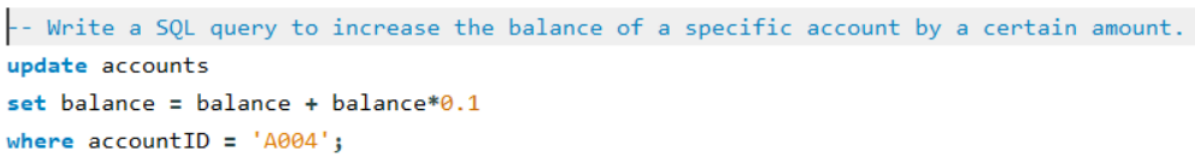


```sql
-- Write a SQL query to list all transaction corresponding customer
select a.customerID as `Customer ID`,
        t.t_type as `Transcation Type`,
        t.transaction_date as `Transaction Date`,
        sum(t.amount) as `Total_Amount`
from transactions t
join accounts a on a.accountID = t.accountID
group by a.customerID, t.t_type, t.transaction_date
order by a.customerID;
```

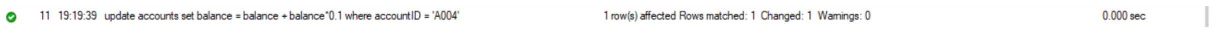


Customer ID	Transcation Type	Transaction Date	Total_Amount
C001	deposit	2024-01-01	10000
C002	withdrawal	2024-01-02	7000
C003	deposit	2024-01-03	9000
C004	deposit	2024-01-06	5000
C004	transfer	2024-01-04	10000
C005	withdrawal	2024-01-05	2000
C006	deposit	2024-01-06	5000
C007	transfer	2024-01-07	36000
C008	deposit	2024-01-08	10000
C009	deposit	2024-01-09	8000
C010	transfer	2024-01-10	12000

4.

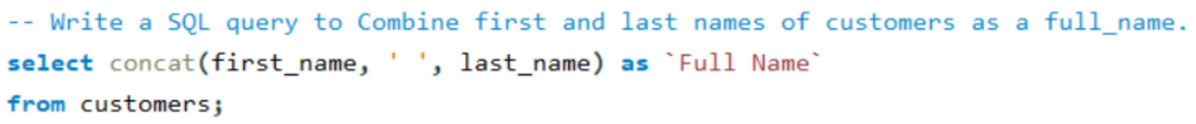


```sql
-- Write a SQL query to increase the balance of a specific account by a certain amount.
update accounts
set balance = balance + balance*0.1
where accountID = 'A004';
```

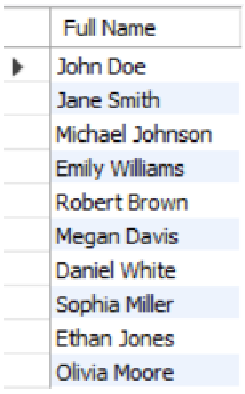


```
11  19:19:39  update accounts set balance = balance + balance*0.1 where accountID = 'A004'    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
```

5.



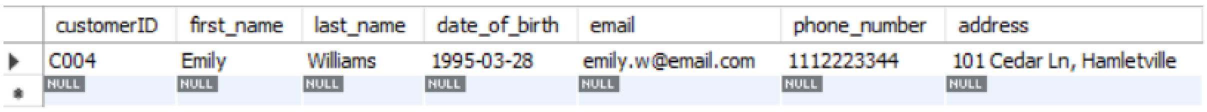
```sql
-- Write a SQL query to Combine first and last names of customers as a full_name.
select concat(first_name, ' ', last_name) as `Full Name`
from customers;
```



Full Name
John Doe
Jane Smith
Michael Johnson
Emily Williams
Robert Brown
Megan Davis
Daniel White
Sophia Miller
Ethan Jones
Olivia Moore

6.

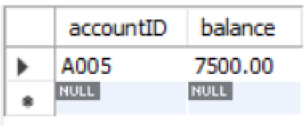
```
-- Write a SQL query to Find customers living in a specific city.
  select * from customers
  where address LIKE '%Hamletville%';
```



customerID	first_name	last_name	date_of_birth	email	phone_number	address
C004	Emily	Williams	1995-03-28	emily.w@email.com	1112223344	101 Cedar Ln, Hamletville
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7.

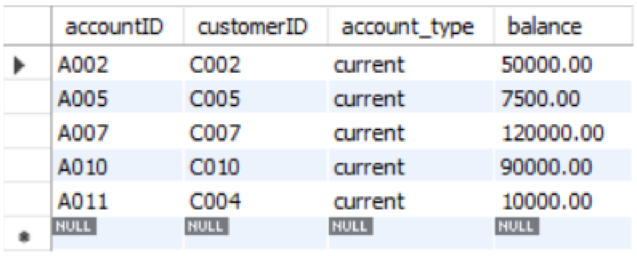
```
-- Write a SQL query to Get the account balance for a specific account.
  select accountID, balance
  from accounts
  where accountID = 'A005';
```



accountID	balance
A005	7500.00
NULL	NULL

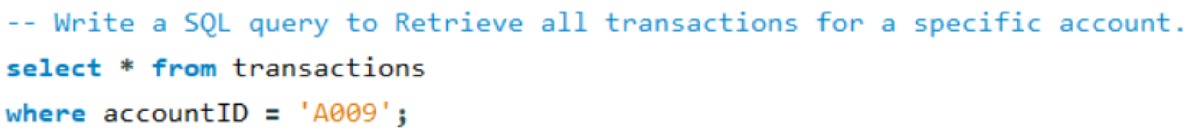
8.

```
-- 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;
```

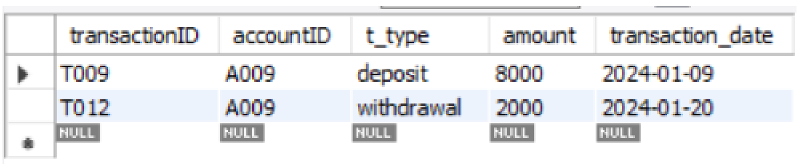


accountID	customerID	account_type	balance
A002	C002	current	50000.00
A005	C005	current	7500.00
A007	C007	current	120000.00
A010	C010	current	90000.00
A011	C004	current	10000.00
NULL	NULL	NULL	NULL

9.

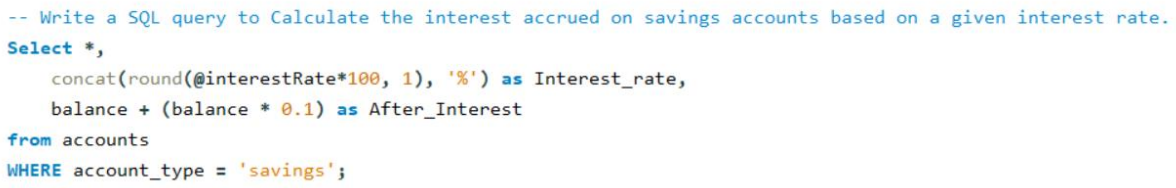


```sql
-- Write a SQL query to Retrieve all transactions for a specific account.
select * from transactions
where accountID = 'A009';
```

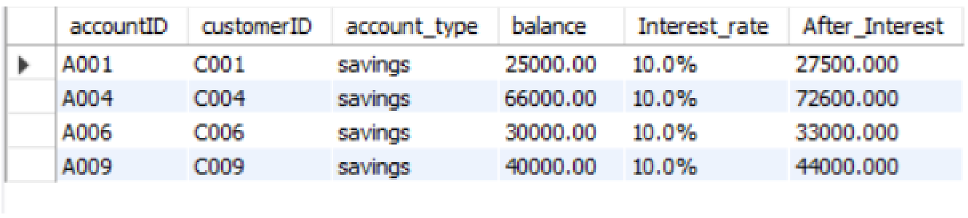


transactionID	accountID	t_type	amount	transaction_date
T009	A009	deposit	8000	2024-01-09
T012	A009	withdrawal	2000	2024-01-20
NULL	NULL	NULL	NULL	NULL

10.

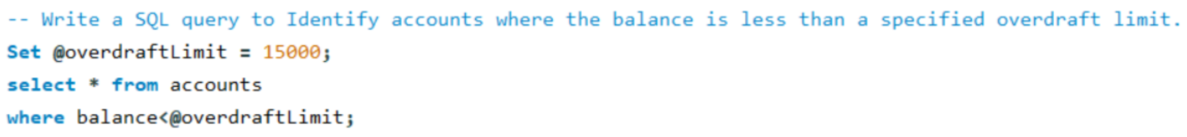


```sql
-- Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.
Select *,
    concat(round(@interestRate*100, 1), '%') as Interest_rate,
    balance + (balance * 0.1) as After_Interest
from accounts
WHERE account_type = 'savings';
```

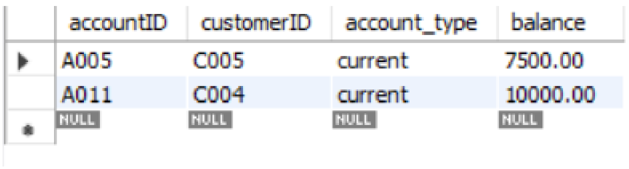


accountID	customerID	account_type	balance	Interest_rate	After_Interest
A001	C001	savings	25000.00	10.0%	27500.000
A004	C004	savings	66000.00	10.0%	72600.000
A006	C006	savings	30000.00	10.0%	33000.000
A009	C009	savings	40000.00	10.0%	44000.000

11.

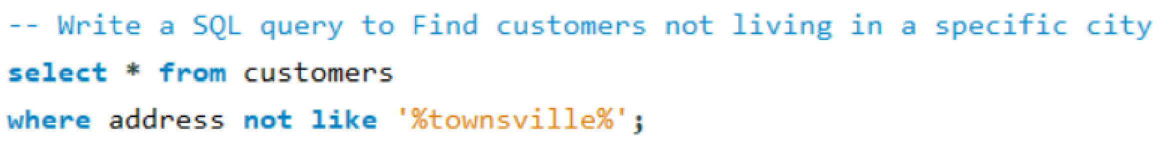


```sql
-- Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.
Set @overdraftLimit = 15000;
select * from accounts
where balance<@overdraftLimit;
```

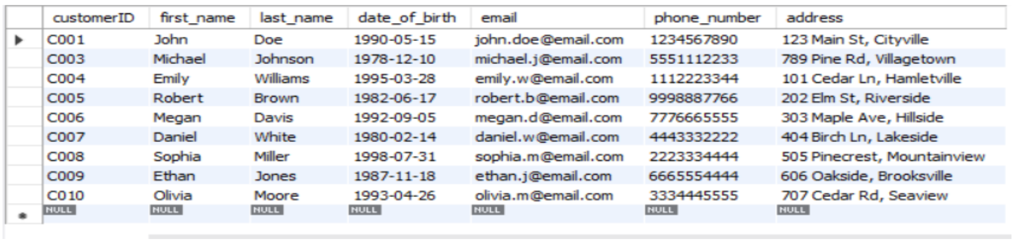


accountID	customerID	account_type	balance
A005	C005	current	7500.00
A011	C004	current	10000.00
NULL	NULL	NULL	NULL

12.



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

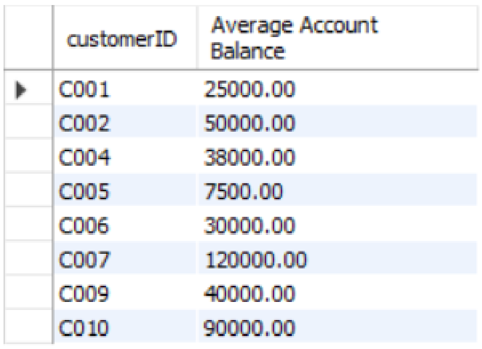


customerID	first_name	last_name	date_of_birth	email	phone_number	address
C001	John	Doe	1990-05-15	john.doe@email.com	1234567890	123 Main St, Cityville
C003	Michael	Johnson	1978-12-10	michael.j@email.com	5551112233	789 Pine Rd, Villagetown
C004	Emily	Williams	1995-03-28	emily.w@email.com	1112223344	101 Cedar Ln, Hamletville
C005	Robert	Brown	1982-06-17	robert.b@email.com	9998887766	202 Elm St, Riverside
C006	Megan	Davis	1992-09-05	megan.d@email.com	7776665555	303 Maple Ave, Hillside
C007	Daniel	White	1980-02-14	daniel.w@email.com	4443332222	404 Birch Ln, Lakeside
C008	Sophia	Miller	1998-07-31	sophia.m@email.com	2223334444	505 Pinecrest, Mountainview
C009	Ethan	Jones	1987-11-18	ethan.j@email.com	6665554444	606 Oakside, Brooksville
C010	Olivia	Moore	1993-04-26	olivia.m@email.com	3334445555	707 Cedar Rd, Seaview
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task-3:

1.

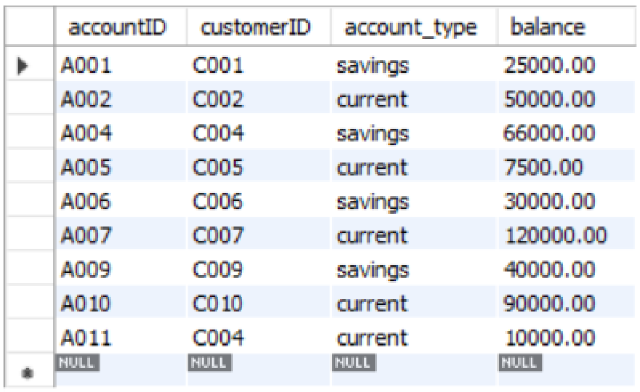
```
-- Write a SQL query to Find the average account balance for all customers.
 select customerID, round(avg(balance), 2) as `Average Account Balance`
 from accounts
 group by customerID;
```



customerID	Average Account Balance
C001	25000.00
C002	50000.00
C004	38000.00
C005	7500.00
C006	30000.00
C007	120000.00
C009	40000.00
C010	90000.00

2.

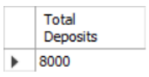
```
-- Write a SQL query to Retrieve the top 10 highest account balances.
 select *
 from accounts
 order by balance desc
 limit 10;
```



accountID	customerID	account_type	balance
A001	C001	savings	25000.00
A002	C002	current	50000.00
A004	C004	savings	66000.00
A005	C005	current	7500.00
A006	C006	savings	30000.00
A007	C007	current	120000.00
A009	C009	savings	40000.00
A010	C010	current	90000.00
A011	C004	current	10000.00
NULL	NULL	NULL	NULL

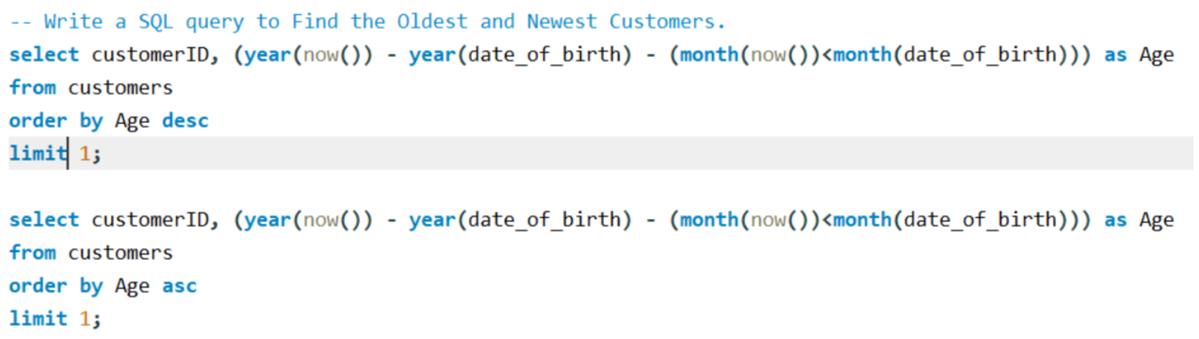
3.

```
-- Write a SQL query to Calculate Total Deposits for All Customers in specific date.
 select sum(amount)
 from transactions
 where t_type = 'deposit' and transaction_date = '2024-01-09';
```

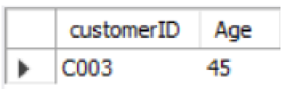


Total Deposits
8000

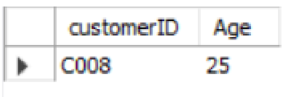
4.



```
-- Write a SQL query to Find the Oldest and Newest Customers.
select customerID, (year(now()) - year(date_of_birth) - (month(now())<month(date_of_birth))) as Age
from customers
order by Age desc
limit 1;

select customerID, (year(now()) - year(date_of_birth) - (month(now())<month(date_of_birth))) as Age
from customers
order by Age asc
limit 1;
```

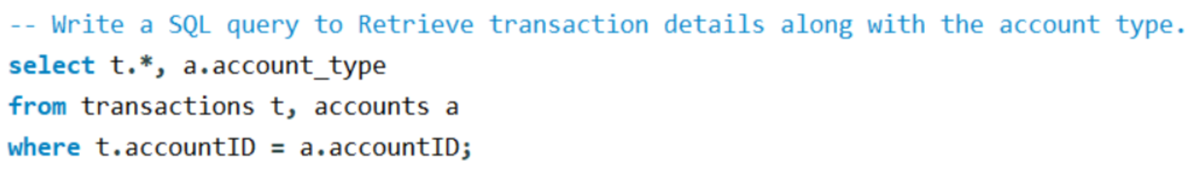


customerID	Age
C003	45

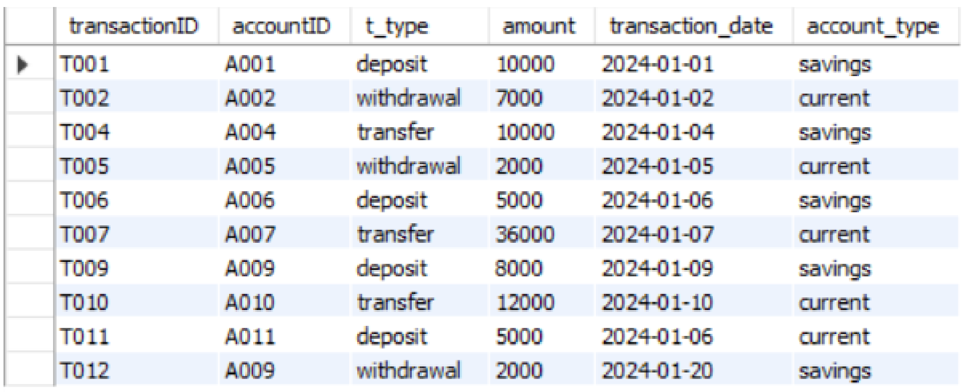


customerID	Age
C008	25

5.

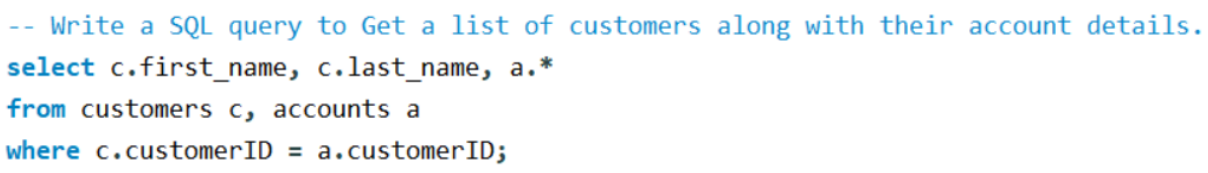


```
-- Write a SQL query to Retrieve transaction details along with the account type.
select t.*, a.account_type
from transactions t, accounts a
where t.accountID = a.accountID;
```

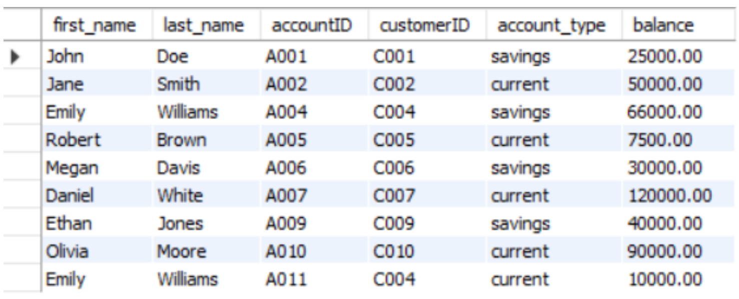


transactionID	accountID	t_type	amount	transaction_date	account_type
T001	A001	deposit	10000	2024-01-01	savings
T002	A002	withdrawal	7000	2024-01-02	current
T004	A004	transfer	10000	2024-01-04	savings
T005	A005	withdrawal	2000	2024-01-05	current
T006	A006	deposit	5000	2024-01-06	savings
T007	A007	transfer	36000	2024-01-07	current
T009	A009	deposit	8000	2024-01-09	savings
T010	A010	transfer	12000	2024-01-10	current
T011	A011	deposit	5000	2024-01-06	current
T012	A009	withdrawal	2000	2024-01-20	savings

6.



```
-- Write a SQL query to Get a list of customers along with their account details.
select c.first_name, c.last_name, a.*
from customers c, accounts a
where c.customerID = a.customerID;
```



first_name	last_name	accountID	customerID	account_type	balance
John	Doe	A001	C001	savings	25000.00
Jane	Smith	A002	C002	current	50000.00
Emily	Williams	A004	C004	savings	66000.00
Robert	Brown	A005	C005	current	7500.00
Megan	Davis	A006	C006	savings	30000.00
Daniel	White	A007	C007	current	120000.00
Ethan	Jones	A009	C009	savings	40000.00
Olivia	Moore	A010	C010	current	90000.00
Emily	Williams	A011	C004	current	10000.00

7.

```sql
-- Write a SQL query to Retrieve transaction details along with customer information for a specific account.
select c.*, t.accountID, t.t_type, t.amount
from customers c
join accounts a on c.customerID = a.customerID
join transactions t on a.accountID = t.accountID
where c.customerID = 'C009';
```

customerID	first_name	last_name	date_of_birth	email	phone_number	address	accountID	t_type	amount
C009	Ethan	Jones	1987-11-18	ethan.j@email.com	6665554444	606 Oakside, Brooksville	A009	deposit	8000
C009	Ethan	Jones	1987-11-18	ethan.j@email.com	6665554444	606 Oakside, Brooksville	A009	withdrawal	2000

8.

```sql
-- Write a SQL query to Identify customers who have more than one account.
select customerID
from accounts
group by customerID
having count(customerID)>1;
```

customerID
C004

9.

```sql
-- Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.
select SUM(case
            when t_type='deposit' then amount else (-1 * amount)
        end) as Difference
from transactions
where t_type in ('deposit', 'withdrawal');
```

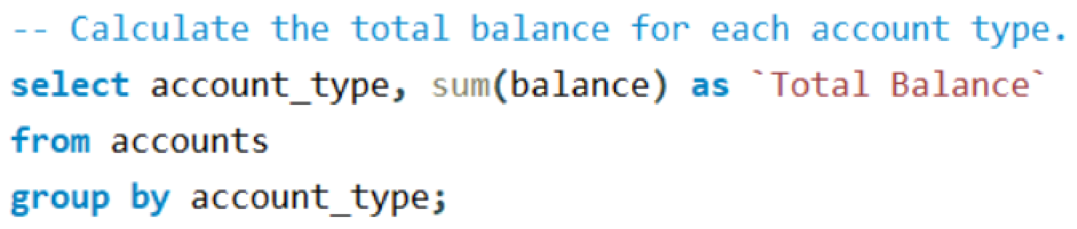
Difference
17000

10.

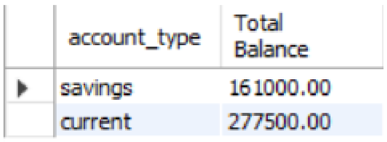
```sql
-- Write a SQL query to Calculate the average daily balance for each account over a specified period.
select a.accountID, round(avg(a.balance), 2) AS Average_Balance
from accounts a
left join transactions t on t.accountID = a.accountID and t.transaction_date between '2024-01-06' and '2024-01-20'
group by a.accountID;
```

accountID	Average_Balance
A001	25000.00
A002	50000.00
A004	66000.00
A005	7500.00
A006	30000.00
A007	120000.00
A009	40000.00
A010	90000.00
A011	10000.00

11.

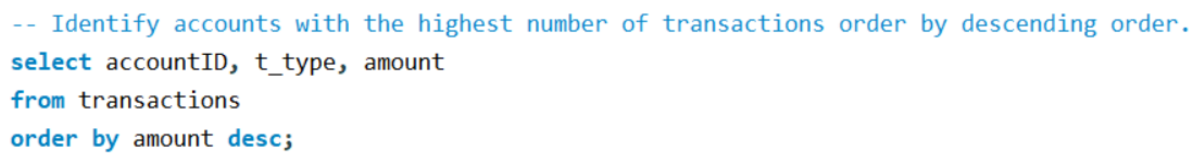


```
-- Calculate the total balance for each account type.
select account_type, sum(balance) as `Total Balance`
from accounts
group by account_type;
```

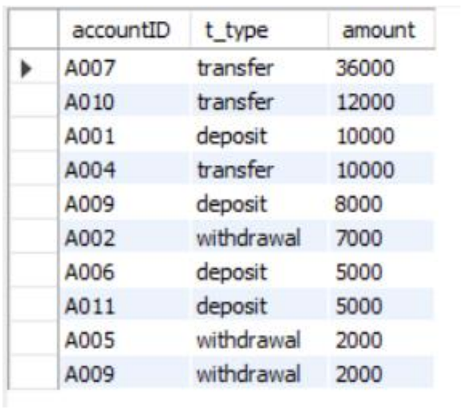


account_type	Total Balance
savings	161000.00
current	277500.00

12.

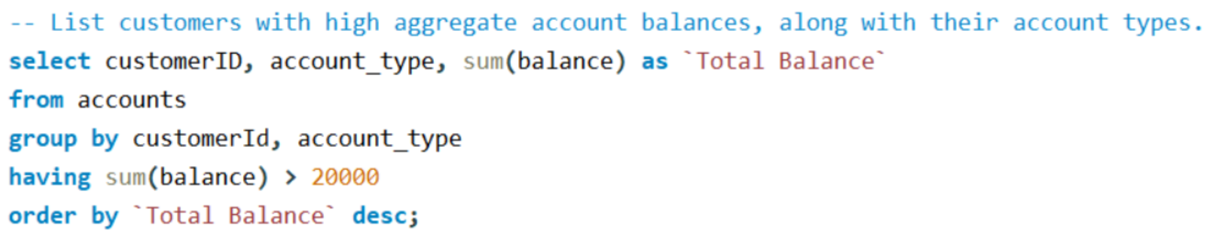


```
-- Identify accounts with the highest number of transactions order by descending order.
select accountID, t_type, amount
from transactions
order by amount desc;
```

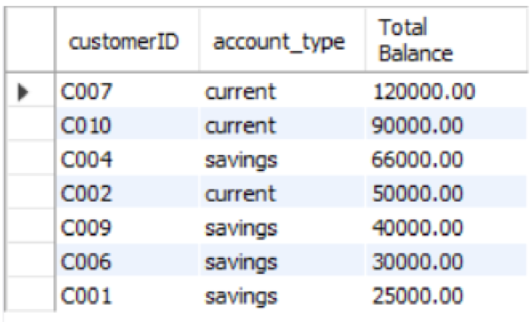


accountID	t_type	amount
A007	transfer	36000
A010	transfer	12000
A001	deposit	10000
A004	transfer	10000
A009	deposit	8000
A002	withdrawal	7000
A006	deposit	5000
A011	deposit	5000
A005	withdrawal	2000
A009	withdrawal	2000

13.

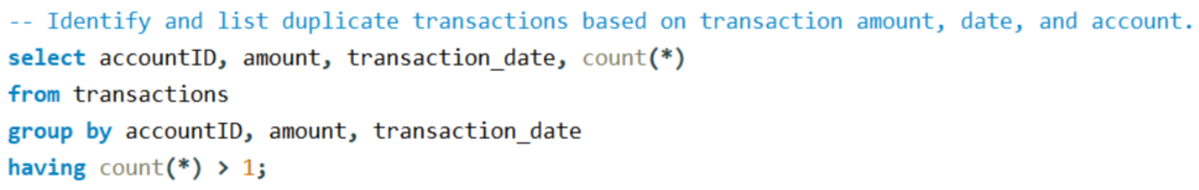


```
-- List customers with high aggregate account balances, along with their account types.
select customerID, account_type, sum(balance) as `Total Balance`
from accounts
group by customerId, account_type
having sum(balance) > 20000
order by `Total Balance` desc;
```

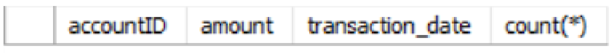


customerID	account_type	Total Balance
C007	current	120000.00
C010	current	90000.00
C004	savings	66000.00
C002	current	50000.00
C009	savings	40000.00
C006	savings	30000.00
C001	savings	25000.00

14.



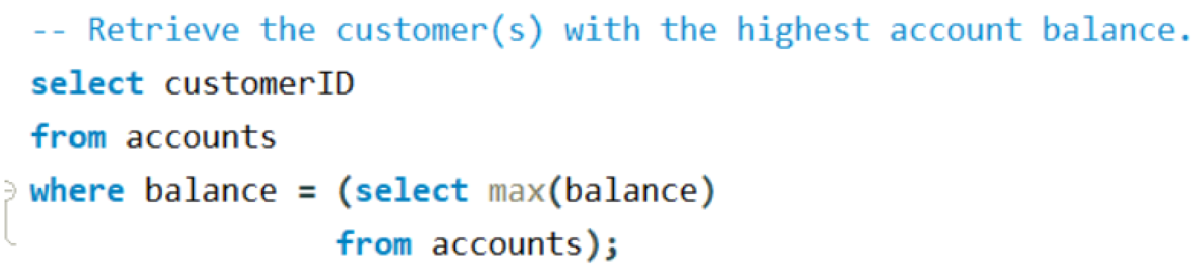
```sql
-- Identify and list duplicate transactions based on transaction amount, date, and account.
select accountID, amount, transaction_date, count(*)
from transactions
group by accountID, amount, transaction_date
having count(*) > 1;
```



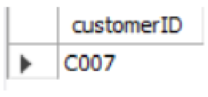
accountID	amount	transaction_date	count(*)

Task-4:

1.

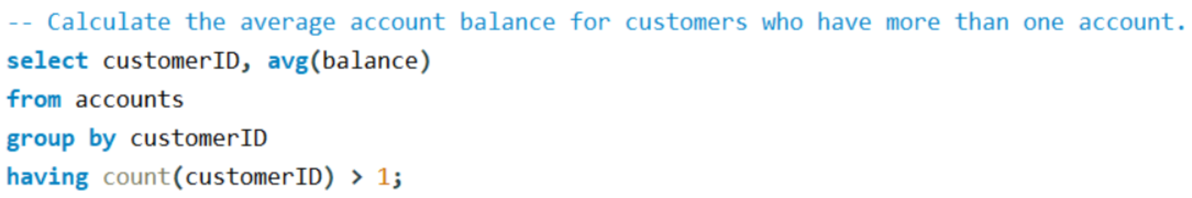


```sql
-- Retrieve the customer(s) with the highest account balance.
select customerID
from accounts
where balance = (select max(balance)
                 from accounts);
```

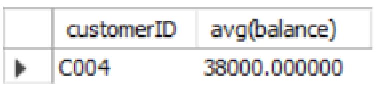


customerID
C007

2.

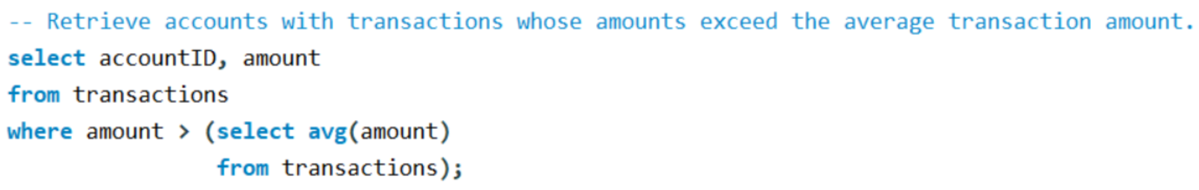


```sql
-- Calculate the average account balance for customers who have more than one account.
select customerID, avg(balance)
from accounts
group by customerID
having count(customerID) > 1;
```

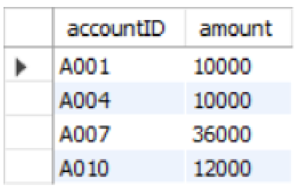


customerID	avg(balance)
C004	38000.000000

3.

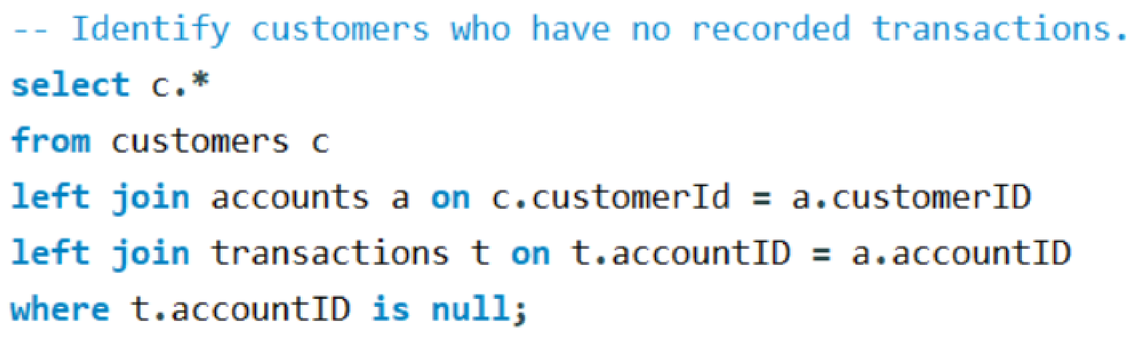


```sql
-- Retrieve accounts with transactions whose amounts exceed the average transaction amount.
select accountID, amount
from transactions
where amount > (select avg(amount)
                from transactions);
```

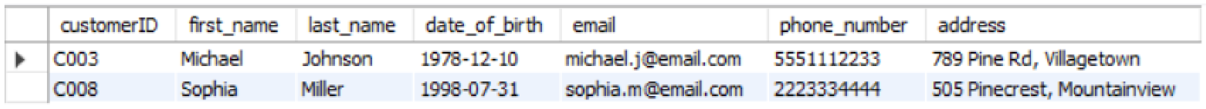


accountID	amount
A001	10000
A004	10000
A007	36000
A010	12000

4.

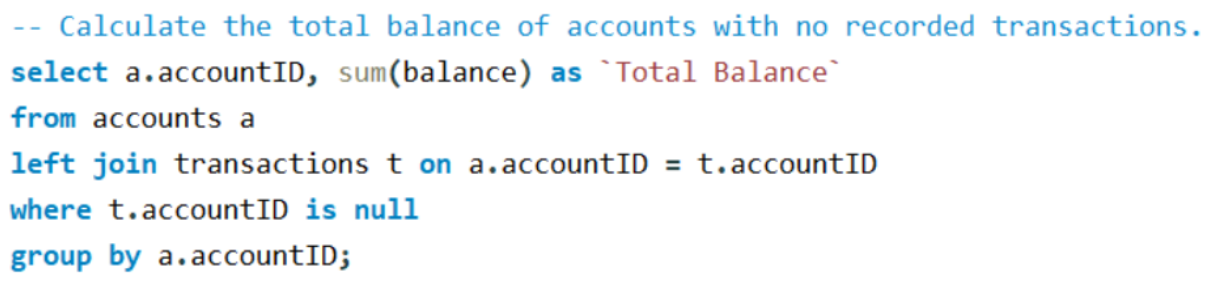


```
-- Identify customers who have no recorded transactions.
select c.*
from customers c
left join accounts a on c.customerId = a.customerID
left join transactions t on t.accountID = a.accountID
where t.accountID is null;
```

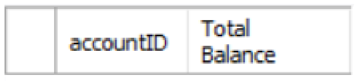


	customerID	first_name	last_name	date_of_birth	email	phone_number	address
▶	C003	Michael	Johnson	1978-12-10	michael.j@email.com	5551112233	789 Pine Rd, Villagetown
	C008	Sophia	Miller	1998-07-31	sophia.m@email.com	2223334444	505 Pinecrest, Mountainview

5.

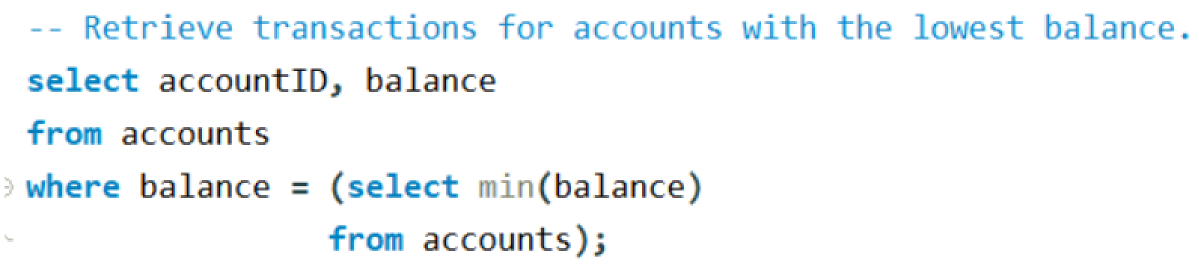


```
-- Calculate the total balance of accounts with no recorded transactions.
select a.accountID, sum(balance) as `Total Balance`
from accounts a
left join transactions t on a.accountID = t.accountID
where t.accountID is null
group by a.accountID;
```

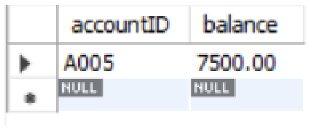


	accountID	Total Balance

6.

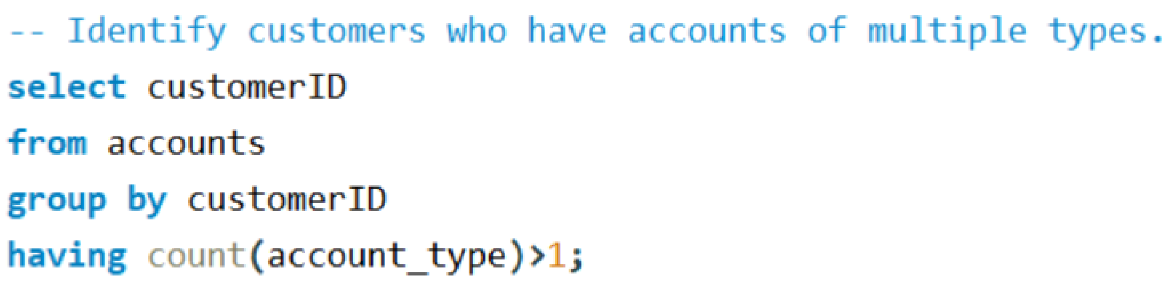


```
-- Retrieve transactions for accounts with the lowest balance.
select accountID, balance
from accounts
where balance = (select min(balance)
                 from accounts);
```

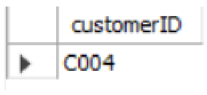


	accountID	balance
▶	A005	7500.00
*	NULL	NULL

7.

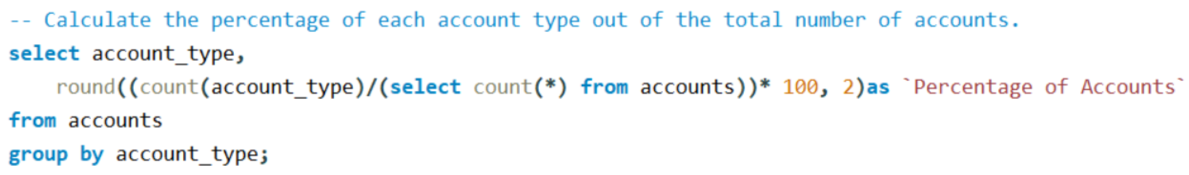


```sql
-- Identify customers who have accounts of multiple types.
select customerID
from accounts
group by customerID
having count(account_type)>1;
```

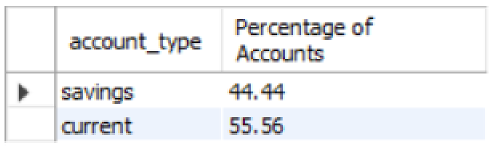


	customerID
▶	C004

8.

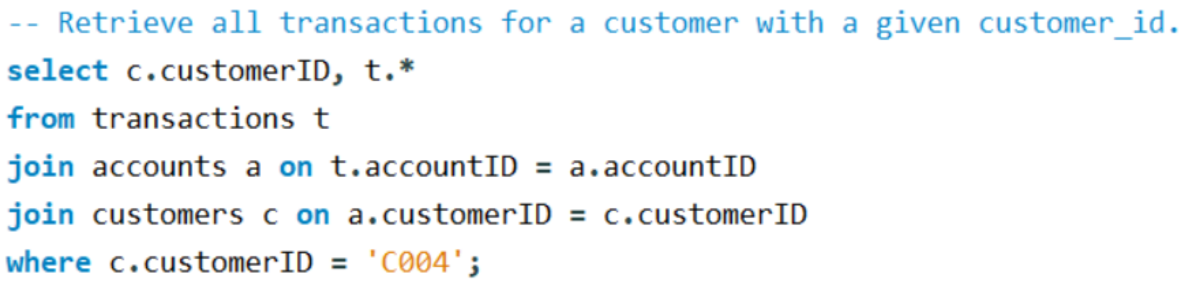


```sql
-- Calculate the percentage of each account type out of the total number of accounts.
select account_type,
    round((count(account_type)/(select count(*) from accounts))* 100, 2)as `Percentage of Accounts`
from accounts
group by account_type;
```

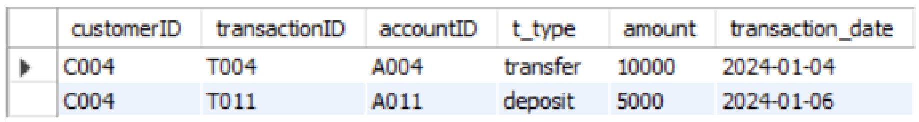


	account_type	Percentage of Accounts
▶	savings	44.44
	current	55.56

9.

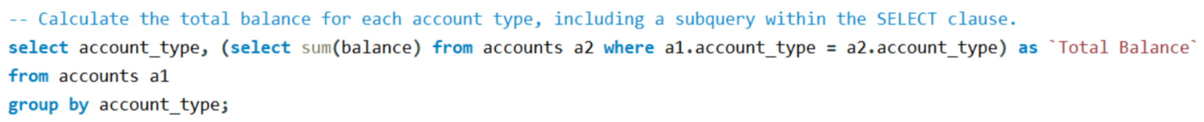


```sql
-- Retrieve all transactions for a customer with a given customer_id.
select c.customerID, t.*
from transactions t
join accounts a on t.accountID = a.accountID
join customers c on a.customerID = c.customerID
where c.customerID = 'C004';
```

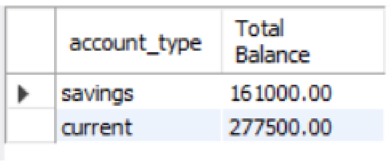


	customerID	transactionID	accountID	t_type	amount	transaction_date
▶	C004	T004	A004	transfer	10000	2024-01-04
	C004	T011	A011	deposit	5000	2024-01-06

10.



```sql
-- Calculate the total balance for each account type, including a subquery within the SELECT clause.
select account_type, (select sum(balance) from accounts a2 where a1.account_type = a2.account_type) as `Total Balance`
from accounts a1
group by account_type;
```



	account_type	Total Balance
▶	savings	161000.00
	current	277500.00