**Banking System**

**Task 1**

CREATE DATABASE HMBank

USE HMBank

create table Customers

(customer\_id int primary key,

first\_name varchar(50) not null,

last\_name varchar(50) not null,

DOB date not null,

email varchar(100) unique not null,

phone\_number varchar(15) unique not null,

customer\_Address text not null)

create table Accounts

(account\_id int primary key,

customer\_id int not null,

account\_type varchar(20) not null check (account\_type IN ('savings', 'current', 'zero\_balance')),

balance decimal(15,2) not null default 0.00,

foreign key (customer\_id) references Customers(customer\_id) ON delete cascade)

create table Transactions

(transaction\_id int primary key,

account\_id int not null,

transaction\_type varchar(20) not null check (transaction\_type IN ('deposit', 'withdrawal', 'transfer')),

amount decimal(15,2) not null,

transaction\_date datetime default getdate(),

foreign key (account\_id) references Accounts(account\_id) ON delete cascade)

**Task 2**

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

• Customers

• Accounts

• Transactions

insert into Customers values

(1, 'John', 'Doe', '1990-05-12', 'john.doe@example.com', '1234567890', '123 Main St, NY'),

(2, 'Alice', 'Smith', '1985-09-23', 'alice.smith@example.com', '2345678901', '456 Oak St, CA'),

(3, 'Bob', 'Johnson', '1992-07-15', 'bob.johnson@example.com', '3456789012', '789 Pine St, TX'),

(4, 'Emma', 'Williams', '1988-11-30', 'emma.williams@example.com', '4567890123', '101 Maple St, FL'),

(5, 'David', 'Brown', '1995-03-17', 'david.brown@example.com', '5678901234', '202 Birch St, WA'),

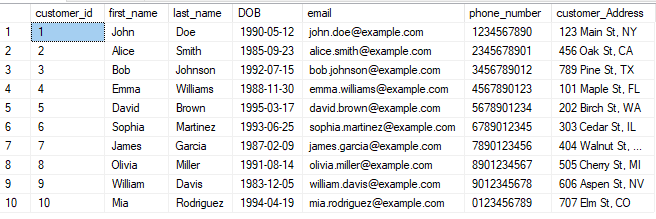
(6, 'Sophia', 'Martinez', '1993-06-25', 'sophia.martinez@example.com', '6789012345', '303 Cedar St, IL'),

(7, 'James', 'Garcia', '1987-02-09', 'james.garcia@example.com', '7890123456', '404 Walnut St, OH'),

(8, 'Olivia', 'Miller', '1991-08-14', 'olivia.miller@example.com', '8901234567', '505 Cherry St, MI'),

(9, 'William', 'Davis', '1983-12-05', 'william.davis@example.com', '9012345678', '606 Aspen St, NV'),

(10, 'Mia', 'Rodriguez', '1994-04-19', 'mia.rodriguez@example.com', '0123456789', '707 Elm St, CO')



Insert into Accounts (customer\_id, account\_type, balance) values

(1, 'savings', 5000.00),

(2, 'current', 12000.50),

(3, 'zero\_balance', 0.00),

(4, 'savings', 7500.75),

(5, 'current', 22000.00),

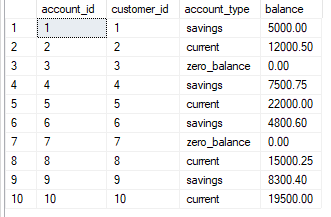
(6, 'savings', 4800.60),

(7, 'zero\_balance', 0.00),

(8, 'current', 15000.25),

(9, 'savings', 8300.40),

(10, 'current', 19500.00)



insert into Transactions values

(1, 1, 'deposit', 1000.00, '2025-03-10 10:30:00'),

(2, 2, 'withdrawal', 500.00, '2025-03-11 12:45:00'),

(3, 3, 'deposit', 2000.00, '2025-03-12 15:20:00'),

(4, 4, 'transfer', 300.00, '2025-03-13 18:10:00'),

(5, 5, 'withdrawal', 2500.00, '2025-03-14 09:00:00'),

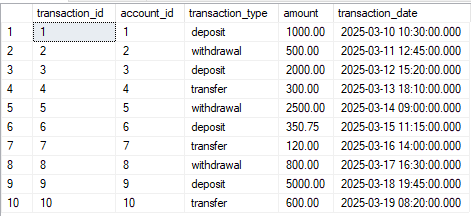
(6, 6, 'deposit', 350.75, '2025-03-15 11:15:00'),

(7, 7, 'transfer', 120.00, '2025-03-16 14:00:00'),

(8, 8, 'withdrawal', 800.00, '2025-03-17 16:30:00'),

(9, 9, 'deposit', 5000.00, '2025-03-18 19:45:00'),

(10, 10, 'transfer', 600.00, '2025-03-19 08:20:00')

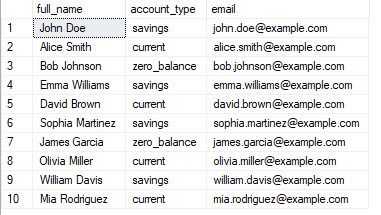


1. Write SQL queries for the following tasks:

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

select first\_name + ' ' + last\_name as full\_name, account\_type, email from customers

join accounts on customers.customer\_id = accounts.customer\_id

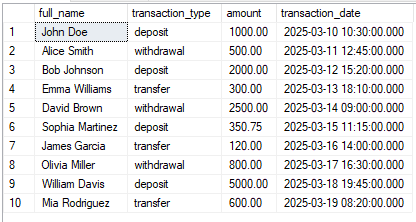


--2--Write a SQL query to list all transaction corresponding customer.

select first\_name + ' ' + last\_name as full\_name, transaction\_type, amount, transaction\_date from customers

join accounts on customers.customer\_id = accounts.customer\_id

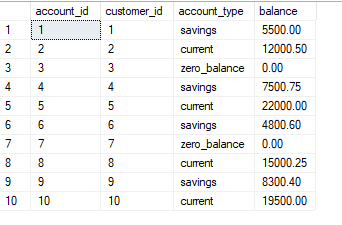
join transactions on accounts.account\_id = transactions.account\_id



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

update accounts set balance = balance + 500 where account\_id = 1

select \* from Accounts



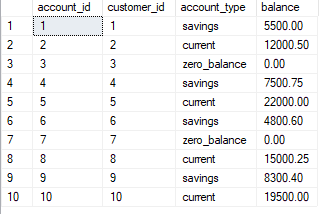
--4--Write a SQL query to Combine first and last names of customers as a full\_name.

select first\_name + ' ' + last\_name as full\_name from customers

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

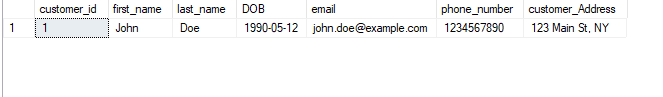
delete from accounts where balance = 0 and account\_type = 'savings'

select \* from Accounts



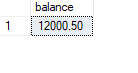
--6--Write a SQL query to Find customers living in a specific city.

select \* from customers where customer\_address like '%123 Main St, NY%'



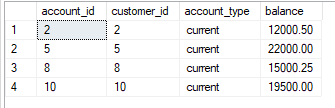
--7--Write a SQL query to Get the account balance for a specific account.

select balance from accounts where account\_id=2



--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



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

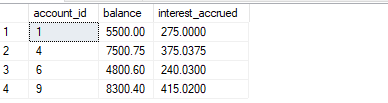
select \* from transactions where account\_id=2

C:\Users\Admin\Pictures\Screenshots\Screenshot (159).png

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

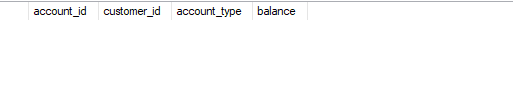
select account\_id, balance, balance \* 0.05 as interest\_accrued from accounts

where account\_type = 'savings'



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

select \* from accounts where balance<-300



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

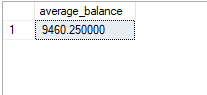
select \* from customers where customer\_Address not like '%123 Main St, NY%'



**Task 3**

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

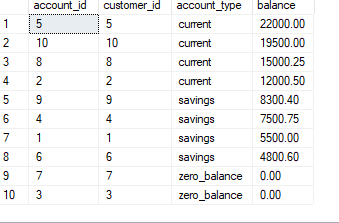
select avg(balance) as average\_balance from accounts



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

select top 10 \* from accounts

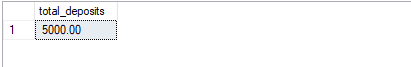
order by balance desc



--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'

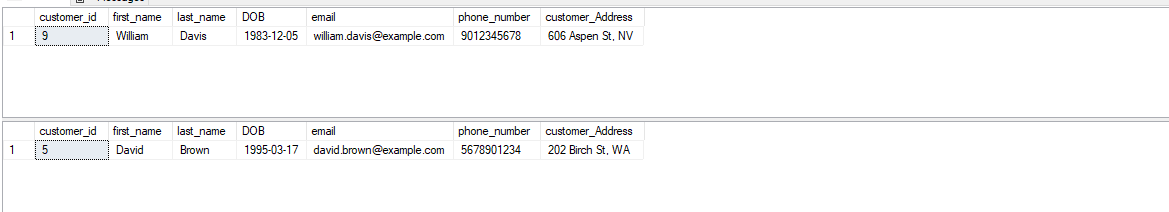
and transaction\_date = '2025-03-18 19:45:00'



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

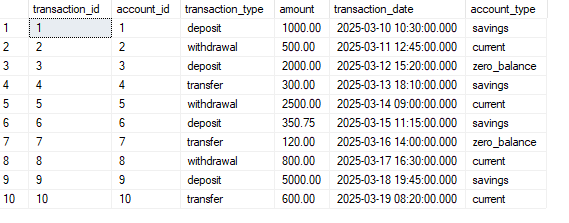
select top 1 \* from customers order by DOB

select top 1 \* from customers order by DOB desc



--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 a.account\_id=t.account\_id



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

select c.\*, a.account\_id, a.account\_type, a.balance from customers c

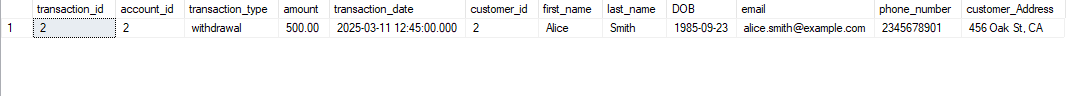
join accounts a on c.customer\_id = a.customer\_id



--7--Write a SQL query to 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

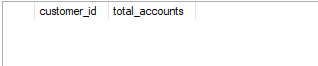
join customers c on c.customer\_id=a.customer\_id where t.account\_id=2



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

select customer\_id, count(account\_id) as total\_accounts from accounts group by customer\_id

having count(account\_id) > 1



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

select account\_id,

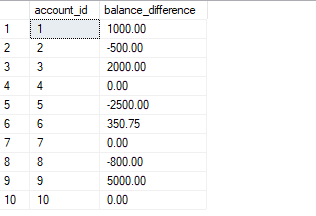
sum(case when transaction\_type = 'deposit' then amount else 0 end) -

sum(case when transaction\_type = 'withdrawal' then amount else 0 end)

as balance\_difference

from transactions

group by account\_id

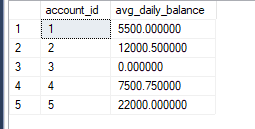


--10--Write a SQL query to Calculate the average daily balance for each account over a specified period.

select account\_id, avg(balance) as avg\_daily\_balance from accounts

where account\_id IN (select account\_id from transactions where transaction\_date between '2025-03-01' and '2025-03-15')

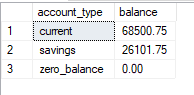
group by account\_id



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

select account\_type, sum(balance) from accounts

group by account\_type

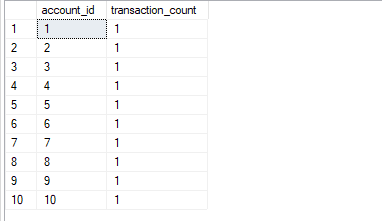


--12--Identify accounts with the highest number of transactions order by descending order.

select account\_id, count(transaction\_id) as transaction\_count from transactions

group by account\_id

order by transaction\_count desc



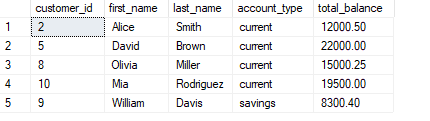
--13-- List customers with high aggregate account balances, along with their account types.

select c.customer\_id, c.first\_name, c.last\_name, a.account\_type, sum(a.balance) as total\_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

having sum(a.balance) > 8000

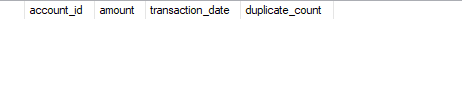


--14--Identify and list duplicate transactions based on transaction amount, date, and account.

select account\_id, amount, transaction\_date, count(\*) as duplicate\_count from transactions

group by account\_id, amount, transaction\_date

having count(\*) > 1



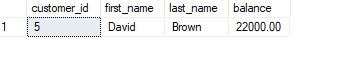
**Task 4**

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

select c.customer\_id, c.first\_name, c.last\_name, a.balance from customers c

join accounts a on c.customer\_id = a.customer\_id

where a.balance = (select max(balance) from accounts)



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

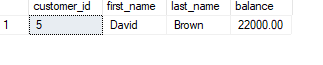
select avg(balance) as avg\_balance

from accounts

where customer\_id IN (select customer\_id from accounts

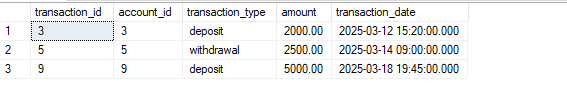
group by customer\_id

having count(account\_id) > 1)



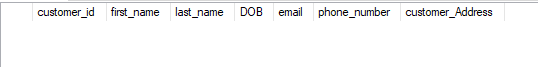
--3--Retrieve accounts with transactions whose amounts exceed the average transaction amount.

select \*from transactions where amount>(select avg(amount) from transactions)



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

select \* from customers where customer\_id not in (select customer\_id from accounts a join transactions t on a.account\_id = t.account\_id)



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

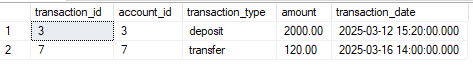
select sum(balance) as total\_balance from accounts

where account\_id not in (select account\_id from transactions)

C:\Users\Admin\Pictures\Screenshots\Screenshot (182).png

--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))



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

select customer\_id from accounts

group by customer\_id

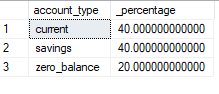
having count(account\_type) > 1

C:\Users\Admin\Pictures\Screenshots\Screenshot (183).png

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

select account\_type, count(\*) \* 100.0 / (select count(\*) from accounts) as \_percentage from accounts

group by account\_type



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

select t.\* from transactions t join accounts a on t.account\_id = a.account\_id

WHERE a.customer\_id = 1

C:\Users\Admin\Pictures\Screenshots\Screenshot (185).png

--10--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

