**Banking System**

**Tasks 1: Database Design:**

1. Create the database named "HMBank"

create database HMBank

use HMBank

2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

create table Customers (

customer\_id int primary key,

first\_name varchar(100),

last\_name varchar(100),

DOB date,

email varchar(255),

phone\_number nvarchar(100),

address varchar(255))

create table Accounts(

account\_id int primary key,

customer\_id int,

account\_type varchar(255),

balance decimal(18,0),

foreign key(customer\_id) references Customers(customer\_id))

create table Transactions(

transaction\_id int primary key,

account\_id int,

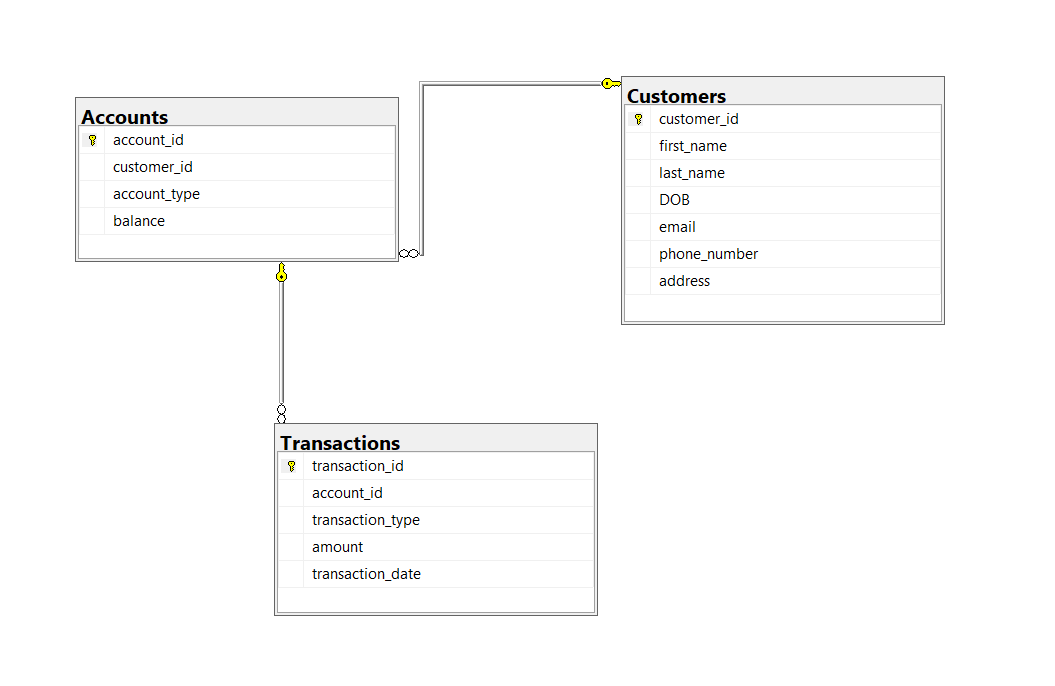
transaction\_type varchar(100),

amount decimal(18,2),

transaction\_date date,

foreign key(account\_id) references Accounts(account\_id))

4. Create an ERD (Entity Relationship Diagram) for the database.



5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

create table Accounts(

account\_id int primary key,

customer\_id int,

account\_type varchar(255),

balance decimal(18,0),

foreign key(customer\_id) references Customers(customer\_id))

create table Transactions(

transaction\_id int primary key,

account\_id int,

transaction\_type varchar(100),

amount decimal(18,2),

transaction\_date date,

foreign key(account\_id) references Accounts(account\_id))

* 1. 6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.
  + Customers
  + Accounts
  + Transactions

1. create table Customers (
2. customer\_id int primary key,
3. first\_name varchar(100),
4. last\_name varchar(100),
5. DOB date,
6. email varchar(255),
7. phone\_number nvarchar(100),
8. address varchar(255))
9. create table Accounts(
10. account\_id int primary key,
11. customer\_id int,
12. account\_type varchar(255),
13. balance decimal(18,0),
14. foreign key(customer\_id) references Customers(customer\_id))
15. create table Transactions(
16. transaction\_id int primary key,
17. account\_id int,
18. transaction\_type varchar(100),
19. amount decimal(18,2),
20. transaction\_date date,
21. foreign key(account\_id) references Accounts(account\_id))

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

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

Accounts

Transactions

insert into Customers(customer\_id,first\_name,last\_name,DOB,email,phone\_number,address)

values(1,'Rakhul','Varma','2004-06-17','rakhul@gmail.com','8772475942','Mumbai'),

(2,'Sneha','Iyer','2004-06-17','sneha@gmail.com','9725725564','Lucknow'),

(3,'Anil','Kumar','2005-06-17','anil@gmail.com','9674274682','Delhi'),

(4,'Kavya','Menon','2003-06-17','kavya@gmail.com','8846205434','Bangalore'),

(5,'Riya','Das','2018-06-17','riya@gmail.com','8268394245','Bangalore'),

(6,'Bill','Weekends','2002-06-17','bill@gmail.com','8294583024','Lucknow'),

(7,'Arjun','Reddy','2009-06-17','arjun@gmail.com','7890346829','Delhi'),

(8,'Nisha','Shankar','2015-06-17','nisha@gmail.com','9735926893','Hyderabad'),

(9,'Riya','Das','2018-06-17','riya@gmail.com','8268394245','Bangalore'),

(10,'Vikas','Patel','2023-06-17','vikas@gmail.com','9876543027','Chennai')

insert into Accounts(account\_id,customer\_id,account\_type,balance)

values(101,1,'savings',6000.00),

(102,1,'current',9000.00),

(103,3,'zero\_balance',450.89),

(104,4,'current',2000.00),

(105,5,'savings',0),

(106,7,'zero\_balance',54300.60),

(107,7,'current',10340.00),

(108,8,'savings',0),

(109,9,'zero\_balance',63400.00),

(110,10,'zero\_balance',35500.56)

insert into Transactions(transaction\_id,account\_id,transaction\_type,amount,transaction\_date)

values(3001,101,'deposit',5000,'2024-06-01'),

(3002,102,'transfer',3000,'2024-07-04'),

(3003,103,'deposit',2000,'2024-06-23'),

(3004,104,'withdrawal',0,'2024-03-09'),

(3005,101,'transfer',7000,'2024-12-21'),

(3006,106,'transfer',0,'2024-03-18'),

(3007,107,'deposit',1000,'2024-09-16'),

(3008,104,'withdrawal',500,'2024-06-05'),

(3009,101,'deposit',5000,'2024-06-01'),

(3010,110,'deposit',5500,'2024-05-19')

2. Write SQL queries for the following tasks:

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

select c.first\_name+' '+c.last\_name as name,a.account\_type,c.email

from Customers c join Accounts a

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

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

select c.first\_name as name,t.transaction\_id,t.account\_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 t.account\_id=a.account\_id

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

update Accounts set balance=balance+5000

where account\_id between 105 and 107

select \* from accounts

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

select Customers.first\_name+' '+Customers.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' and account\_id

not in ( select distinct account\_id from transactions)

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

select \* from Customers

where address='Lucknow'

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

select account\_id,balance from accounts

where balance between 1000 and 10000

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 amount <3000

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

select account\_id,customer\_id,account\_type,balance, balance\*0.0025 as interest

from accounts where account\_type='savings'--25% interest

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

select \* from accounts where balance <2000—overdraft limit is 2000

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

select \* from customers where address!='delhi'

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

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

select customer\_id ,avg(balance) as average\_acc\_balance

from accounts

group by customer\_id

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

select top(10) balance ,account\_id from accounts

order by balance desc

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

select c.first\_name as name ,sum(t.amount) as total\_deposits

from Transactions t join accounts a on t.account\_id=a.account\_id join customers c

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

where Transaction\_type='deposit' and transaction\_date between '2023-01-01' and '2025-01-01'

group by c.first\_name

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

select top(1) first\_name as oldest\_customers,customer\_id from customers

order by dob

select top(1) first\_name as newest\_customers ,customer\_id from customers

order by dob desc

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

Select t.transaction\_id,t.account\_id,t.transaction\_type,t.amount,t.transaction\_date,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.first\_name,a.account\_id,a.customer\_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 c.\*,t.\* from customers c join accounts a

on c.customer\_id=a.customer\_id join Transactions t

on a.account\_id=t.account\_id where a.account\_id =101

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

select c.customer\_id,c.first\_name as name ,count (a.account\_id) as account\_count

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

group by c.customer\_id,c.first\_name

having count(a.account\_id)>1

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

select(select sum(amount) as total\_deposit from transactions

where transaction\_type='deposit')-

(select sum(amount) as total\_withdrawal from transactions

where transaction\_type='withdrawal')as difference

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

select account\_id ,account\_type,round(balance/datediff(day,'2024-06-01','2024-06-10')+1,2)as avg\_daily\_balance from accounts

11. Calculate the total balance for each account type.

select account\_type,sum(balance) as total\_balance

from Accounts

group by account\_type

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

select account\_id,count(\*) as transaction\_count

from Transactions group by account\_id

order by transaction\_count desc

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

select c.customer\_id,c.first\_name as 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 a.account\_type,c.customer\_id, c.first\_name

order by total\_balance desc

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

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

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

select c.customer\_id,c.first\_name as name,a.balance

from customers c join accounts a on a.customer\_id=c.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 c.customer\_id,c.first\_name as name ,count (a.account\_id) as account\_count,avg(a.balance) as average\_balance

from accounts a join customers c

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

group by c.customer\_id,c.first\_name having count(a.account\_id)>1

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

select distinct account\_id from Transactions

where amount>(select avg(amount) from Transactions)

4. Identify customers who have no recorded transactions.

select \* from customers c where not exists (select \* from accounts a join Transactions t

on a.account\_id=t.account\_id where a.customer\_id=c.customer\_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 distinct account\_id from Transactions)

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 c.customer\_id,a.account\_type from accounts a join customers c

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

group by c.customer\_id,account\_type

having count(distinct account\_type)>1

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

select account\_type, count(\*) as count\_type, round(count(\*)\*100/(select count(\*) from accounts),2) as percentage

from Accounts group by account\_type

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=4)

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

select account\_type,(select sum(balance) from accounts ) as total\_balance from accounts group by account\_type