**TASK 1**1. create database HMBank;

6. use HMBank;

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),

address varchar(255)

);

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)

);

**TASK 2**

1. INSERT INTO customers (customer\_id, first\_name, last\_name, dob, email, phone\_number, address) VALUES

(1, 'John', 'Doe', '1985-04-12', 'john.doe@example.com', '555-1234', '123 Elm St, Springfield'),

(2, 'Jane', 'Smith', '1990-08-05', 'jane.smith@example.com', '555-5678', '456 Oak St, Ridgemont'),

(3, 'Alice', 'Johnson', '1978-11-23', 'alice.johnson@example.com', '555-8765', '789 Pine St, Lakeshore'),

(4, 'Bob', 'Lee', '1982-01-16', 'bob.lee@example.com', '555-4321', '321 Maple St, Hillvalley'),

(5, 'Mary', 'Taylor', '1995-07-30', 'mary.taylor@example.com', '555-9823', '654 Cedar St, Brookwood'),

(6, 'Mike', 'Brown', '1965-05-17', 'mike.brown@example.com', '555-2938', '987 Spruce St, Eastwood'),

(7, 'Nancy', 'Davis', '1989-12-06', 'nancy.davis@example.com', '555-7384', '213 Willow St, Westgate'),

(8, 'Chris', 'Wilson', '1972-03-29', 'chris.wilson@example.com', '555-6729', '546 Birch St, Greenfield'),

(9, 'Patricia', 'Lopez', '1984-10-28', 'patricia.lopez@example.com', '555-9812', '879 Elm St, Southbank'),

(10, 'Luis', 'Martinez', '1993-02-19', 'luis.martinez@example.com', '555-6345', '162 Oak St, Northridge');

--VALUESFORACCOUNTTABLE:

INSERT INTO accounts (account\_id, customer\_id, account\_type, balance) VALUES

(1, 1, 'Checking', 1500.00),

(2, 1, 'Savings', 5500.50),

(3, 2, 'Checking', 850.75),

(4, 3, 'Savings', 12250.00),

(5, 4, 'Checking', 2150.00),

(6, 5, 'Savings', 8000.00),

(7, 6, 'Checking', 480.00),

(8, 7, 'Savings', 300.00),

(9, 8, 'Checking', 5000.00),

(10,9,'Savings',7500.25);  
  
--INSERTstatementsfortransactionstable:  
  
INSERT INTO transactions (transaction\_id, account\_id, transaction\_type, amount, transaction\_date) VALUES

(1, 1, 'Deposit', 200.00, '2024-04-30'),

(2, 1, 'Withdrawal', 50.00, '2024-04-30'),

(3, 2, 'Deposit', 300.00, '2024-04-29'),

(4, 2, 'Withdrawal', 20.00, '2024-04-29'),

(5, 3, 'Deposit', 450.00, '2024-04-28'),

(6, 4, 'Withdrawal', 30.00, '2024-04-27'),

(7, 5, 'Deposit', 250.00, '2024-04-26'),

(8, 6, 'Withdrawal', 60.00, '2024-04-25'),

(9, 7, 'Deposit', 780.00, '2024-04-24'),

(10, 8, 'Withdrawal', 90.00, '2024-04-23');

2.  
  
i. select c.first\_name, c.last\_name, a.account\_type, c.email from customers c

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

ii. select c.first\_name, c.last\_name, t\* from customers c inner join accounts a on c.customer\_id = a.customer\_id inner join transactions t ona.account\_id = t.account\_id;

iii. update accounts set balance = balance + 5000 where account\_id = 101;

iv. select concat(c.first\_name, ' ', c.last\_name) as full\_name from customers c;

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

vi. select \*from customers where address like '%Delhi%';

vii. select balance from accounts where account\_id = 102;

viii. select \* from accounts where account\_type = 'current' and balance > 1000;

ix. select \* from transactions where account\_id = 103;

x. select account\_id, balance \* 0.05 as interest\_accrued from accounts where account\_type = 'savings';

xi. select \* from accounts where balance < -500;

xii. select \* from customers where address not like '%Mumbai%';

**TASK 3**

1.select avg(balance) as average\_balance from accounts;

2. select account\_id, balance from accounts order by balance desc limit 10;

3.select sum(amount) as total\_deposits from transactions where transaction\_type = 'deposit'and transaction\_date = '2024-04-01';

4. select c1.first\_name, c2.first\_name from customers c1 join customers c2 on c1.customer\_id=(select min(customer\_id) from customers) and c2.customer\_id=(select max(customer\_id) from customers);

5.select t.\*, a.account\_type from transactions t inner join accounts a on t.account\_id = a.account\_id;

6.select c.\*, a.account\_type, a.balance from customers c inner join accounts a on c.customer\_id = a.customer\_id;

7. select c.\*, t.\* from customers c inner join accounts a on c.customer\_id = a.customer\_id inner join transactions t ona.account\_id = t.account\_id where a.account\_id = 101;

8.select c.customer\_id, c.first\_name, c.last\_name, count(a.account\_id) as num\_accounts from customers c inner join accounts a on c.customer\_id = a.customer\_id group by c.customer\_id having count(a.account\_id) > 1;

9.select account\_id, sum(amount) - sum(amount \* (transaction\_type<> 'deposit')) as transaction\_difference from transactions group by account\_id;

10. select account\_id, avg(balance) from accounts group by account\_id;

11. select account\_type, sum(balance) from accounts group by account\_type;

12. select account\_id, count(transaction\_id) as num\_transactions from transactions

group by account\_id order by num\_transactions desc;

13. select c.customer\_id, c.first\_name, c.last\_name, sum(a.balance) as total\_balance from customers c inner join accounts a on c.customer\_id = a.customer\_id group by c.customer\_id order by total\_balancedesc;

14. select transaction\_id, account\_id, amount, transaction\_date from transactions group by account\_id, amount, transaction\_date having count(\*) > 1;

**TASK 4**

1.select c.customer\_id, c.first\_name, c.last\_name, (select max(balance) from accounts where customer\_id = c.customer\_id) as highest\_balance from customers c order by highest\_balancedesc;

2.select avg(balance) from ( select c.customer\_id, count(a.account\_id) as num\_accounts, sum(a.balance) from customers c inner join accounts a on c.customer\_id = a.customer\_id group by c.customer\_id having num\_accounts> 1);

3. select a.\*from accounts a inner join transactions t ona.account\_id = t.account\_id where t.amount> (select avg(amount) from transactions);

4. select customer\_id, first\_name, last\_name from customers where customer\_id not in ( select distinct c.customer\_id from customers c inner join accounts a on c.customer\_id = a.customer\_id inner join transactions t ona.account\_id = t.account\_id);

5.select sum(balance) from accounts where account\_id not in (select account\_id from transactions);

6. select t.\*from transactions t inner join accounts a on t.account\_id = a.account\_id where a.balance = (select min(balance) from accounts);

7. select c.customer\_id, c.first\_name, c.last\_name from customers c inner join (select customer\_id from accounts group by customer\_id having count(distinct account\_type) > 1) as multi\_account\_customers on c.customer\_id = multi\_account\_customers.customer\_id;

8.select account\_type, count(\*) \* 100/ (select count(\*) from accounts) from accounts group by account\_type;

9. select t.\*from transactions t where t.account\_id in (select account\_id from accounts where customer\_id = 101);

10. select account\_type, (select sum(balance) from accounts where account\_type = a.account\_type) from accounts a group by account\_type;