

Hexaware Assignment – SQL Banking System

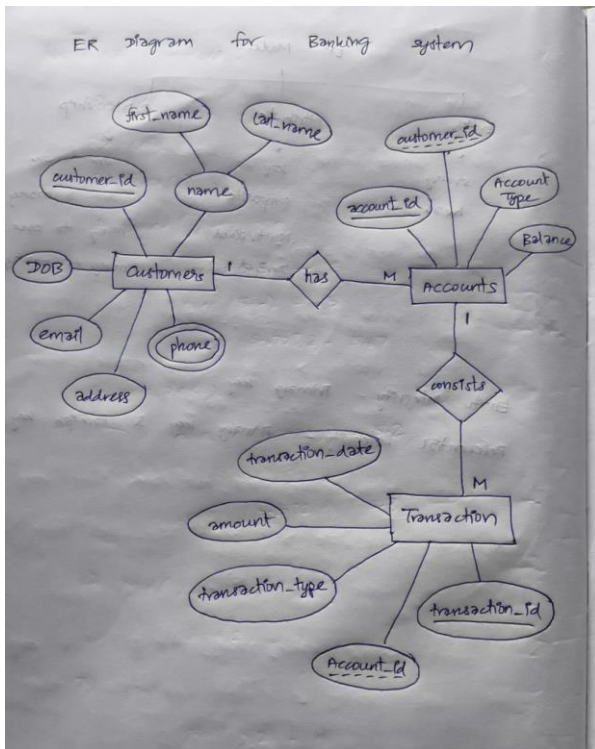
TASK 1

1. Create the database named "HMBank"

```
mysql> create database hmbank;
```

```
mysql> use hmbank;
```

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



3. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships. • Customers • Accounts • Transactions

```
mysql> create table Customers(cust_id int primary key, first_name varchar(20), last_name
varchar(20), dob date, email text, phone_num varchar(15), address text);
```

```
mysql> create table Accounts(acc_id int primary key, cust_id int, acc_type varchar(15), balance
decimal(15,2) default 0.00, foreign key(cust_id) references Customers(cust_id) on delete cascade);
```

```
mysql> create table Transactions(trans_id int primary key, acc_id int, trans_type varchar(15), amount
decimal(15,2) check(amount>0), trans_date date, foreign key(acc_id) references Accounts(acc_id) on
delete cascade);
```

TASK 2

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

```
mysql> insert into Customers values (1, "Rahul", "Sharma", "1994-10-04", "rahul@gmail.com",  
"5656565656", "13, nehru street, chennai"), (2, "Akash", "Gupta", "1990-04-10",  
"akash@yahoo.com", "1234567890", "10, gandhi street, chennai"), (3, "Suresh", "Raina", "1993-10-  
03", "raina@gmail.com", "3333333333", "22, marina, chennai"), (4, "Virat", "Kohli", "1985-11-07",  
"kohli@gmail.com", "4949494949", "18, chinnaswamy, bangalore"), (5, "David", "Miller", "1999-06-  
01", "miller@yahoo.com", "1010101010", "17, nehru street, chennai"), (6, "Gopal", "Varma", "2000-  
01-08", "gopal@gmail.com", "2323232323", "23, main street, chennai"), (7, "MS", "Dhoni", "1980-07-  
07", "dhoni@gmail.com", "7777777777", "7, marina, chennai"), (8, "Ravindra", "Jadeja", "1990-11-  
24", "jadeja@gmail.com", "8888888888", "8, marina, chennai"), (9, "DJ", "Bravo", "1993-10-12",  
"bravo@yahoo.com", "9999999999", "9, nehru st, chennai"), (10, "Shivam", "Dube", "1984-02-19",  
"dube@gmail.com", "2525252525", "58, cross street, chennai");
```

```
mysql> insert into accounts (acc_id, cust_id, acc_type, balance) values (1, 5, "current", 5000.00), (2,  
3, "zero_balance", 0.00), (3, 7, "savings", 75000.90), (4, 1, "savings", 15000.50), (5, 10,  
"zero_balance", 0.00), (6, 6, "zero_balance", 0.00), (7, 2, "current", 22000.75), (8, 9, "savings",  
9800.10), (9, 4, "savings", 45000.25), (10, 8, "current", 12000.30), (11, 1, "current", 8000.00), (12, 2,  
"savings", 16000.40);
```

```
mysql> insert into transactions (trans_id, acc_id, trans_type, amount, trans_date) values (1, 1,  
"deposit", 5000.00, "2024-03-01"), (2, 2, "withdrawal", 2000.00, "2024-03-02"), (3, 3, "deposit",  
10000.50, "2024-03-03"), (4, 4, "transfer", 2500.00, "2024-03-04"), (5, 5, "deposit", 7500.00, "2024-  
03-05"), (6, 6, "withdrawal", 1500.00, "2024-03-06"), (7, 7, "transfer", 3000.00, "2024-03-07"), (8, 8,  
"deposit", 8000.75, "2024-03-08"), (9, 9, "withdrawal", 4000.00, "2024-03-09"), (10, 10, "deposit",  
6000.00, "2024-03-10"), (11, 11, "transfer", 2000.50, "2024-03-11"), (12, 12, "deposit", 9000.00,  
"2024-03-12");
```

II. Write SQL queries for the following tasks:

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

```
mysql> select concat(first_name, ' ', last_name) as name, acc_type, email from Customers, Accounts  
where Customers.cust_id = Accounts.cust_id;
```

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

```
mysql> select  
Customers.cust_id, first_name, Accounts.acc_id, acc_type, trans_id, trans_type, amount, trans_date  
from Customers, Accounts, Transactions where Customers.cust_id=Accounts.cust_id and  
Accounts.acc_id=Transactions.acc_id;
```

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

```
mysql> update accounts set balance=balance+2000 where acc_id=5;
```

4. Write a SQL query to Combine first and last names of customers as a full_name.

```
mysql> select concat(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.

```
mysql> delete from accounts where balance=0 and acc_type="savings";
```

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

```
mysql> select * from Customers where address like"%chennai%";
```

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

```
mysql> select balance from Accounts where acc_id=5;
```

8. Write a SQL query to List all current accounts with a balance greater than \$1,000.

```
mysql> select * from Accounts where acc_type="current" and balance>1000;
```

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

```
mysql> select * from transactions where acc_id=5;
```

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

```
mysql> select acc_id,cust_id,balance,balance*(0.1) as "interest_amt" from accounts where acc_type="savings";
```

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

```
mysql> select * from accounts where balance<2000;
```

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

```
mysql> select * from Customers where address not like "%chennai%";
```

TASK 3

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

```
mysql> select avg(balance) as "average balance" from accounts;
```

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

```
mysql> select balance from accounts order by balance desc limit 10;
```

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

```
mysql> select sum(amount) as "Total Deposits" from transactions where trans_type="deposit" and trans_date="2024-03-18";
```

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

```
mysql> select c.cust_id, concat(first_name, ' ', last_name) as "Oldest Customer" from Customers c join Accounts a on c.cust_id=a.cust_id join Transactions t on a.acc_id=t.acc_id order by trans_date desc limit 1;
```

```
mysql> select c.cust_id, concat(first_name, ' ', last_name) as "Newest Customer" from Customers c join Accounts a on c.cust_id=a.cust_id join Transactions t on a.acc_id=t.acc_id order by trans_date limit 1;
```

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

```
mysql> select t.*,a.acc_type from accounts a right join transactions t on a.acc_id=t.acc_id;
```

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

```
mysql> select c.cust_id,first_name,last_name,dob,email,phone_num,address,acc_id,acc_type,balance from customers c left join accounts a on c.cust_id=a.cust_id;
```

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

```
mysql> select t.*,c.cust_id,concat(c.first_name,' ',c.last_name) as "Cust Name",phone_num from transactions t left join accounts a on t.acc_id=a.acc_id join customers c on a.cust_id=c.cust_id where t.acc_id=5;
```

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

```
mysql> select c.cust_id,c.first_name,c.last_name,count(a.acc_id) from customers c join accounts a on c.cust_id=a.cust_id group by c.cust_id having count(a.acc_id)>1;
```

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

```
mysql> select (sum(case when trans_type="deposit" then amount end )-sum(case when trans_type="withdrawal" then amount end)) as "Difference" from transactions;
```

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

```
mysql> select acc_id, sum(amount)/count(distinct trans_date) from transactions where trans_date between "2024-03-03" and "2024-03-17" group by acc_id;
```

11. Calculate the total balance for each account type.

```
mysql> select acc_type, sum(balance) from accounts group by acc_type;
```

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

```
mysql> select acc_id, count(*) as "No_of_Transactions" from transactions group by acc_id order by No_of_Transactions desc;
```

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

```
mysql> select c.cust_id,c.first_name,a.acc_type,sum(a.balance) from customers c join accounts a on c.cust_id = a.cust_id group by c.cust_id,a.acc_type having sum(balance)>10000 ;
```

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

```
mysql> select acc_id,amount,trans_date,count(*) from transactions group by acc_id,amount,trans_date having count(*) >1;
```

TASK 4

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

```
mysql> select c.cust_id,c.first_name,balance from customers c join accounts a on c.cust_id=a.cust_id
where balance = (select max(balance) from accounts);
```

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

```
mysql> select cust_id, avg(balance) from accounts group by cust_id having count(acc_id)>1;
```

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

```
mysql> select * from transactions where amount > (select avg(amount) from transactions);
```

4. Identify customers who have no recorded transactions.

```
mysql> select cust_id from customers where cust_id not in (select cust_id from accounts where
acc_id in (select acc_id from transactions));
```

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

```
mysql> select sum(balance) from accounts where acc_id not in (select acc_id from transactions);
```

6. Retrieve transactions for accounts with the lowest balance.

```
mysql> select * from transactions where acc_id in (select acc_id from accounts where balance in
(select min(balance) from accounts));
```

7. Identify customers who have accounts of multiple types.

```
mysql> select cust_id,count(acc_type) from accounts group by cust_id having count(acc_type)>1;
```

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

```
mysql> select acc_type, (count(*)/(select count(*) from accounts))*100.0 from accounts group by
acc_type;
```

9. Retrieve all transactions for a customer with a given customer_id.

```
mysql> select * from transactions where acc_id in (select acc_id from accounts where cust_id=7);
```

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

```
mysql> select distinct acc_type, (select sum(balance) from accounts a where a.acc_type =  
b.acc_type) as total_balance from accounts b;
```