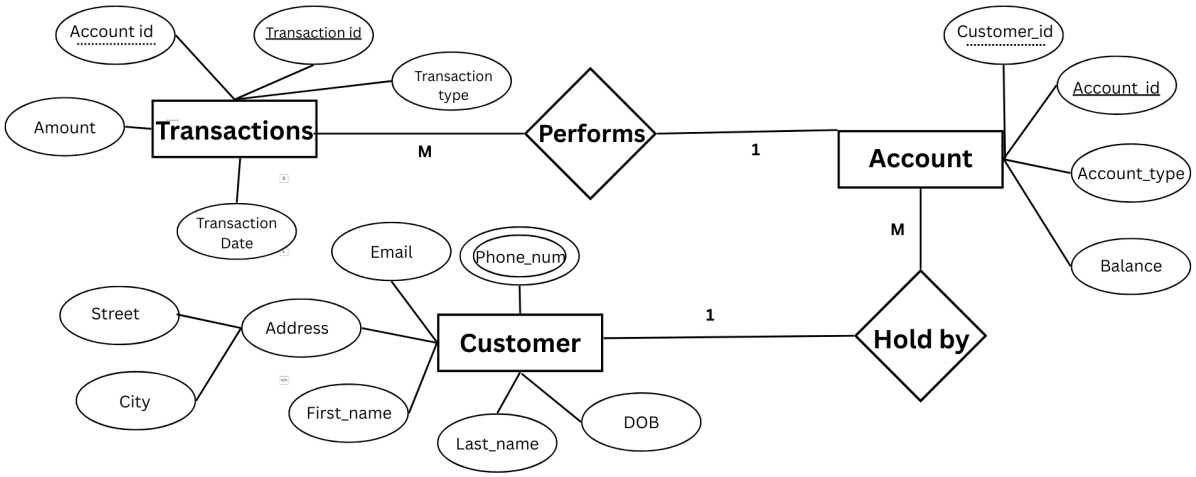
BANKING SYSTEM

ER Diagram:



SQL Queries:

1. create database hmbank;

2. use hmbank;

3. create table customers (
   customer_id int auto_increment 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,
   address text not null
);

4. create table accounts (
   account_id int auto_increment primary key,
   customer_id int,
   account_type enum('savings', 'current', 'zero_balance') not null,
   balance decimal(12,2) not null check (balance >= 0),

```
    foreign key (customer_id) references customers(customer_id) on delete cascade
);

5. create table transactions (
    transaction_id int auto_increment primary key,
    account_id int,
    transaction_type enum('deposit', 'withdrawal', 'transfer') not null,
    amount decimal(12,2) not null check (amount > 0),
    transaction_date timestamp default current_timestamp,
    foreign key (account_id) references accounts(account_id) on delete cascade
);

6. insert into customers (first_name, last_name, dob, email, phone_number, address) values
('john', 'doe', '1990-05-15', 'john.doe@example.com', '9876543210', '123 main st, citya'),
('alice', 'smith', '1985-08-22', 'alice.smith@example.com', '9876543211', '456 park ave,
cityb'),
('bob', 'brown', '1992-11-10', 'bob.brown@example.com', '9876543212', '789 elm st, cityc'),
('emily', 'davis', '1988-03-25', 'emily.davis@example.com', '9876543213', '321 maple st,
cityd'),
('michael', 'johnson', '1995-07-30', 'michael.johnson@example.com', '9876543214', '654 oak
st, citye'),
('sarah', 'williams', '1993-02-17', 'sarah.williams@example.com', '9876543215', '987 birch st,
cityf'),
('david', 'wilson', '1980-09-05', 'david.wilson@example.com', '9876543216', '741 pine st,
cityg'),
('laura', 'martinez', '1997-06-12', 'laura.martinez@example.com', '9876543217', '852 cedar st,
cityh'),
('chris', 'anderson', '1989-12-01', 'chris.anderson@example.com', '9876543218', '963 walnut
st, cityi'),
('sophia', 'thomas', '1991-04-29', 'sophia.thomas@example.com', '9876543219', '147 spruce
st, cityj');

7. insert into accounts (customer_id, account_type, balance) values
(1, 'savings', 5000.00),
(2, 'current', 12000.00),
(3, 'zero_balance', 0.00),
(4, 'savings', 7500.00),
(5, 'current', 20000.00),
(6, 'savings', 8300.00),
(7, 'zero_balance', 0.00),
(8, 'savings', 4100.00),
(9, 'current', 15000.00),
(10, 'savings', 6200.00);

8. insert into transactions (account_id, transaction_type, amount) values
(1, 'deposit', 2000.00),
(2, 'withdrawal', 500.00),
(3, 'deposit', 1500.00),
```

(4, 'transfer', 1000.00),
(5, 'deposit', 3000.00),
(6, 'withdrawal', 700.00),
(7, 'deposit', 500.00),
(8, 'transfer', 200.00),
(9, 'deposit', 2500.00),
(10, 'withdrawal', 400.00);

9. select c.first_name, c.last_name, a.account_type, c.email from customers c join accounts a on c.customer_id = a.customer_id;

10. select c.first_name, c.last_name, t.transaction_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 a.account_id = t.account_id;

11. update accounts set balance = balance + 50 where account_id = 101;

12. select concat(first_name, ' ', last_name) as full_name from customers;

13. delete from accounts where balance = 0 and account_type = 'savings' and account_id is not null;

14. select * from customers where address like '%citya';

15. select balance from accounts where account_id = 5;

16. select * from accounts where account_type = 'current' and balance > 1000;

17. select * from transactions where account_id = 2;

18. select account_id, balance, balance * 0.05 as interest_accrued
from accounts
where account_type = 'savings';

19. select * from accounts where balance < 500;

20. select * from customers
where address not like '%cityc';

21. select avg(balance) as average_balance from accounts;

22. select * from accounts order by balance desc limit 10;

23. select sum(amount) as total_deposits
from transactions where transaction_type = 'deposit' and transaction_date = '1985-08-22';

24. select * from customers order by dob asc limit 1;

25. select * from customers order by dob desc limit 1;

26. 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 t.account_id = a.account_id;

27. select c.customer_id, c.first_name, c.last_name, c.email, a.account_id, a.account_type, a.balance from customers c join accounts a on c.customer_id = a.customer_id;

28. select c.customer_id, c.first_name, c.last_name, t.transaction_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 a.account_id = t.account_id
where t.account_id = 105;

29. select customer_id, count(account_id) as account_count from accounts group by customer_id having count(account_id) > 1;

30. 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;

31. select account_id, avg(balance) as avg_daily_balance from accounts where account_id in (select distinct account_id from transactions where transaction_date between '2024-01-01' and '2024-03-01') group by account_id;

32. select account_type, sum(balance) as total_balance from accounts group by account_type;

33. select account_id, count(transaction_id) as transaction_count from transactions group by account_id order by transaction_count desc;

34. 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, a.account_type having sum(a.balance) > 10000;

35. select account_id, transaction_date, amount, count(*) as duplicate_count from transactions group by account_id, transaction_date, amount having count(*) > 1;

36. select * from customers where customer_id in (select customer_id from accounts where balance = (select max(balance) from accounts));

37. select avg(balance) as average_balance from accounts where customer_id in (select customer_id from accounts group by customer_id having count(account_id) > 1);

38. select * from transactions where amount > (select avg(amount) from transactions);

39. select * from customers where customer_id not in (select distinct customer_id from accounts join transactions on accounts.account_id = transactions.account_id);

40.select sum(balance) as total_balance from accounts where account_id not in (select distinct account_id from transactions);

41.select * from transactions where account_id in (select account_id from accounts where balance = (select min(balance) from accounts));

42.select * from customers where customer_id in (select customer_id from accounts group by customer_id having count(distinct account_type) > 1);

43. select account_type, (count(*) * 100.0) / (select count(*) from accounts) as percentage from accounts group by account_type;

44. select * from transactions where account_id in (select account_id from accounts where customer_id = given_customer_id);

45.select account_type, (select sum(balance) from accounts a2 where a2.account_type = a1.account_type) as total_balance from accounts a1 group by account_type;