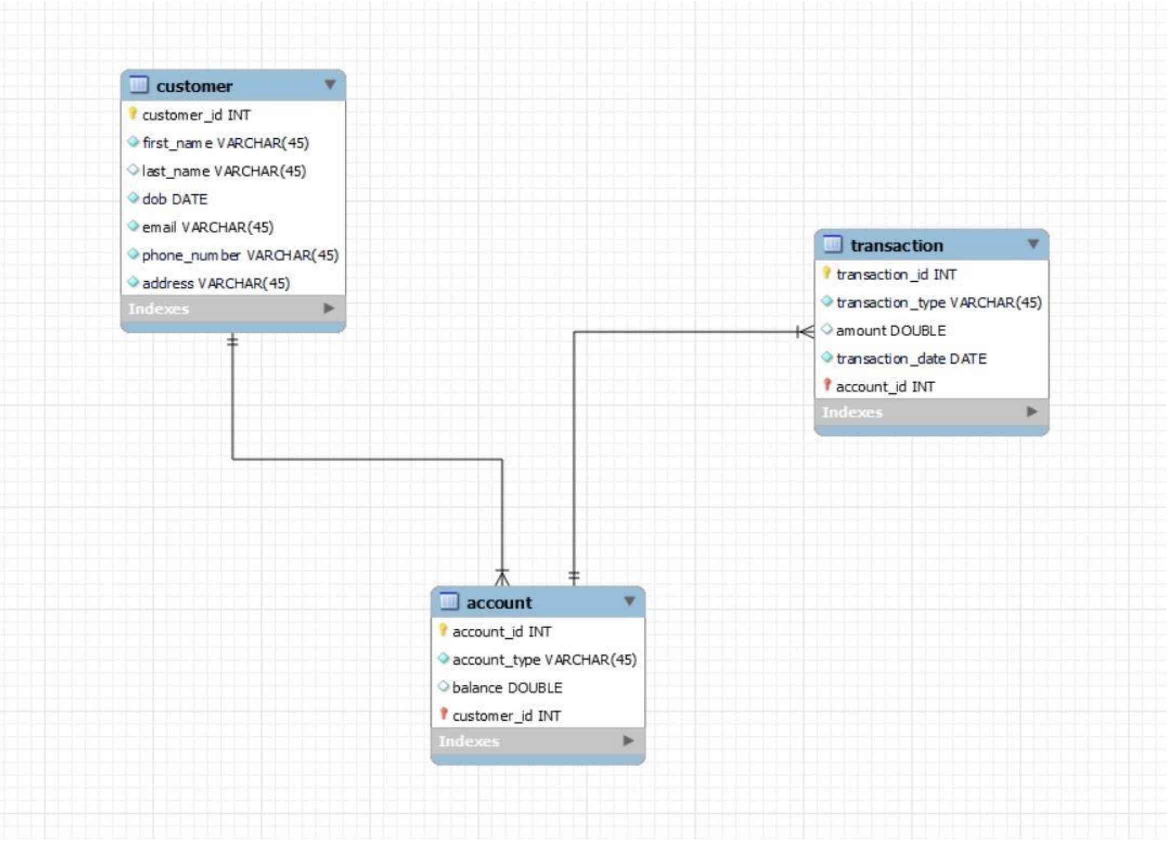
BANKING SYSTEM

ER DIAGRAM



```
use bank;

show tables;

describe customer;

describe transaction;

describe account;

insert into
customer(first_name,last_name,dob,email,phone_number,address)values('nams','mugen','2001-02-03','nams9@gmail.com',9865342341,'Chennai'),
('hari','brues','2002-01-01','hari@gmail.com',9865743277,'Bangalore'),

('suvetha','vel','2000-09-08','vel@gmail.com',8886754320,'Mumbai'),

('ram','sabari','2001-12-12','ram1@gmail.com',8964729888,'Chennai'),

('pavi','balu','2002-09-23','pavi@gmail.com',9876541000,'Bangalore'),

('uma','murugan','2000-11-29','uma@gmail.com',9000541000,'Mumbai'),

('ajith','muthu','2002-08-08','ajith@gmail.com',9986141000,'Bangalore'),

('shriya','gopal','2001-01-23','gopi@gmail.com',9234541000,'Pune'),

('madhu','vijay','2000-12-20','madhu@gmail.com',8909876543,'Mumbai'),

('gayu','chan','2001-09-13','pavi@gmail.com',8876541000,'Pune');

select * from customer;

insert into
account(account_type,balance,customer_id)values("joint",800000,2),("savings",700000,1),

("salary",8000000,2),("joint",90000000,3),("current",6700000,4),("joint",805000,6),

("savings",800000,5),("salary",900000,6),("current",1000000,7),("current",100500,1);

select * from account;

insert into transaction(transaction_type,amount,transaction_date,account_id)values('online
payment',8000,'2024-08-09',6),('current savings',9000,'2023-08-23',7),

('withdrawal',10000,'2023-11-12',8),('online payment',300000,'2023-01-
10',7),('deposit',4040000,'2023-01-01',9),('withdrawal',8500,'2024-03-07',5),

('online payment',9500,'2024-10-09',6),('deposit',10500,'2024-01-25',9),('current
savings',7000,'2024-02-09',8),('online payment',6500,'2024-02-10',8);

select * from transaction;

#TASK 2

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

select c.first_name,c.last_name,c.email,a.account_type from customer c,account a where
c.customer_id=a.customer_id;
```

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

select t.transaction_id,t.transaction_type,t.transaction_date,t.amount from transaction t,account a,customer c where c.customer_id=a.customer_id and a.account_id=t.account_id;

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

select account_id,account_type,(balance+2000) as new_amount from account where account_id=3;

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

select customer_id,concat(first_name,' ',last_name) as full_name from customer;

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

delete from account where balance = 0 and account_type = 'savings';

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

select first_name,last_name from customer where address like '%chennai%';

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

select * from account where account_id=10;

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

select * from account where balance>1000;

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

select * from transaction t join account a on a.account_id=t.account_id where a.account_id=3;

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

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

select * from account where balance<1000000;

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

select * from customer where address NOT LIKE '%chennai%';

-- Tasks 3:
```

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

select c.customer_id,c.first_name,avg(balance) from customer c join account a on
c.customer_id=a.customer_id group by c.customer_id;

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

select account_id,balance from account order by balance DESC limit 0,10;

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

select sum(amount),transaction_date from transaction where transaction_type='deposit' and
transaction_date='2024-01-25' group by transaction_date;

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

(select first_name,dob,'youngest' as age from customer order by dob desc limit 0,1) UNION

(select first_name,dob,'oldest' as age from customer order by dob limit 0,1);

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

select t.transaction_id,t.transaction_type,t.transaction_date,a.account_type from transaction t join
account 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.customer_id,c.first_name,c.last_name,a.account_id,a.account_type,a.balance from
customer c join account 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.transaction_id,t.transaction_type,t.transaction_date,

c.customer_id,c.first_name,c.dob,a.account_id from customer c join account a on
c.customer_id=a.customer_id join transaction t on

t.account_id=a.account_id where a.account_id=5;

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

select c.customer_id,c.first_name,count(a.account_id) from customer c join account a on
c.customer_id=a.customer_id group by c.customer_id 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) from transaction where transaction_type='deposit')-(select
sum(amount) from transaction 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 avg(a.balance),t.transaction_date from account a,transaction t where
a.account_id=t.account_id and t.transaction_date between '2020-01-01' and '2022-12-30'

group by(t.transaction_date);
```

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

```
select account_type,sum(balance)as balance from account group by account_type;
```

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

```
select account_id, COUNT(transaction_id) from transaction group by account_id order by
transaction_id desc;
```

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

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

```
select amount,transaction_date,account_id,count() as duplicate_transactions from transaction
group by amount, transaction_date,account_id HAVING COUNT() >1;
```

#Task-4

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

```
select customer_id,balance from account where balance=(select max(balance) from account);
```

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

```
select account_id,avg(balance) from account where customer_id in (select customer_id from
account group by

customer_id having count(customer_id) > 1);
```

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

```
select account_id, amount from transaction where amount >( select avg(amount) from transaction);
```

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

select customer_id,first_name from customer where customer_id IN (select customer_id from
account where customer_id not in(select account_id from transaction));

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

select account_id,sum(balance) as total_balance from account where account_id not in (select
account_id from transaction);

-- 6. Retrieve transactions for accounts with the lowest balance.

select * from transaction where account_id in(select account_id from account where balance =
(select min(balance)from account));

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

select customer_id from account group by customer_id having count(account_type) > 1;

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

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

select * from transaction where account_id IN (select account_id from account where
customer_id=8);

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

select account_type,sum(balance) as total_balance from account group by account_type;m account
group by account_type;
```