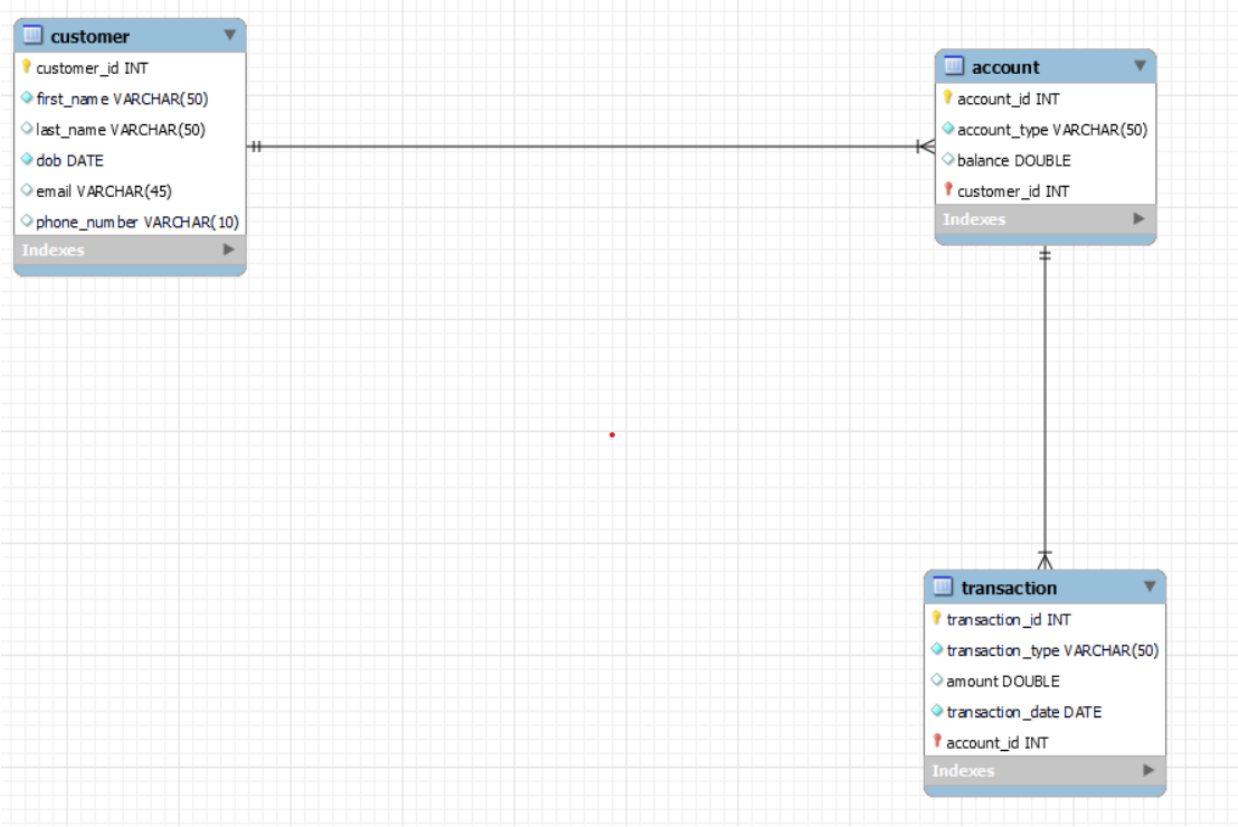
ASSIGNMENT 1:

BANKING SYSTEM

ER DIAGRAM:



QUERIES:

```
use hexfeb;
show tables;
#TASK 1:
create table customer1(customer_id int primary key auto_increment,first_name varchar(50),
last_name varchar(50),dob date,email varchar(50),phone_number varchar(10));
describe customer1;
insert into customer1(first_name,last_name,dob,email,phone_number)
values('aadhya','parvin','1989-04-02','aadhya@gmail.com','4567540'),
('zoya',null,'1986-03-20','zoya@gmail.com','4237540'),
('pankaj','pratap','1983-11-02','pankaj@gmail.com','4245490'),
```

```
('rachel','sebastin','1999-10-15','sebastin@gmail.com','47568480'),
('goyal',null,'2001-07-02','goyal@gmail.com','4567891'),
('rayan','parish','1980-05-18','rayan@gmail.com','4567426');
select *from customer1;


create table account1(account_id int primary key auto_increment,
account_type varchar(50),balance double,customer_id int);

insert into account1(account_type,balance,customer_id)
values('joint account',80000,5),
('savings account',200000,1),
('savings account',350000,4),
('salary account',90000,6),
('nri account',178000,3),
('salary account',70000,2);
create table transaction1(transaction_id int primary key auto_increment,transaction_type
varchar(45),
amount double,transaction_date date,account_id int);

insert into transaction1(transaction_type,
amount,transaction_date,account_id)
values('payment',6000,'2022-12-24',2),
('deposit',80000,'2012-10-02',4),
('transfer',12000,'2000-07-20',3),
('check',40000,'2023-11-12',5),
('payment',2500,'2024-02-20',1),
('withdrawals',17300,'2013-05-27',6);

#TASK 2:
-- 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,a.account_type
from customer1 c,account1 a
where c.customer_id=a.customer_id;

-- 2. Write a SQL query to list all transaction corresponding customer.
select t.transaction_type,t.transaction_date,t.amount,c.first_name
from transaction1 t,account1 a,customer1 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+500) as new_amt
from account1
where account_id=5;

-- 4. Write a SQL query to Combine first and last names of customers as a full_name.
select concat(first_name,' ',last_name) as full_name
from customer1;
-- 5. Write a SQL query to remove accounts with a balance of zero where the account
-- type is savings.
delete from account1
where  balance=0;
select* from account1;

-- 6. Write a SQL query to Find customers living in a specific city.
select first_name
from customer1
where customer_id=5;
-- 7. Write a SQL query to Get the account balance for a specific account.
select balance
from account1
where account_type='savings account';
-- 8. Write a SQL query to List all current accounts with a balance greater than $1,000.
select *
from account1
where balance>1000;
-- 9. Write a SQL query to Retrieve all transactions for a specific account.
select *
from transaction1 t,account1 a
where a.account_id=t.transaction_id and a.account_id=3;

-- 10. Write a SQL query to Calculate the interest accrued on savings accounts based on a
-- given interest rate.
select account_id,account_type,
case
when balance>100000 then (balance-(balance/2)*3)
when balance>200000 then (balance-(balance/4)*2)
else 'no interest'
end as bal_int
```

```
from account1;
/*11. Write a SQL query to Identify accounts where the balance is less than a specified
overdraft limit.*/
select *
from account1
where balance<200000;
-- 12. Write a SQL query to Find customers not living in a specific city.
select *
from customer1
where first_name not in('aadhya','rachel','goyal','pankaj');

/*TASK 3:
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 c.customer_id,avg(a.balance) as bal
from account1 a, customer1 c
where 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 balance
from account1
order by balance desc
limit 10;
-- 3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.
select sum(amount),transaction_date
from transaction1
where transaction_type='deposit'
group by(transaction_date);
-- 4. Write a SQL query to Find the Oldest and Newest Customers.
select c.customer_id,c.first_name,t.transaction_type,t.transaction_date
from customer1 c,transaction1 t,account1 a
where c.customer_id=a.customer_id and a.account_id=t.account_id
order by(t.transaction_date);

-- 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 account1 a,transaction1 t
where 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,c.last_name,a.account_id,a.account_type,a.balance
```

```
from customer1 c join account1 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_type
from customer1 c ,account1 a,
transaction1 t where c.customer_id=a.customer_id and a.account_id=t.account_id;
-- 8. Write a SQL query to Identify customers who have more than one account.
select count(c.customer_id) as id,c.first_name,a.account_type
from customer1 c,account1 a
where c.customer_id=a.customer_id
group by(a.account_type)
having id>1;

-- 9. Write a SQL query to Calculate the difference in transaction amounts between deposits and
-- withdrawals.
select
((select sum(amount)
from transaction1
where transaction_type='deposit')-(select sum(amount)
from transaction1
where transaction_type='withdrawal')) as diff;


-- 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 account1 a,transaction1 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 sum(balance) as bal
from account1
group by(account_type);
-- 12. Identify accounts with the highest number of transactions order by descending order.

select a.account_id,t.transaction_type
from transaction1 t,account1 a
where a.account_id=t.account_id
order by amount desc;
```

```
-- 13. List customers with high aggregate account balances, along with their account types.
select c.customer_id,c.first_name,a.account_type,a.balance
from account1 a,customer1 c
where c.customer_id=a.customer_id
order by a.balance desc;
-- 14. Identify and list duplicate transactions based on transaction amount, date,
-- and account.
select account_id,count(*) as dup
from transaction
group by account_id
having dup>1;
#04.03.2024
/*Tasks 4: Subquery and its type:
1. Retrieve the customer(s) with the highest account balance.*/
select c.first_name,a.balance
from customer1 c,account1 a
where c.customer_id=a.customer_id
order by a.balance desc
limit 0,1;
/*2. Calculate the average account balance for customers who have more than one account.*/
select avg(balance) as bal
from account1
where customer_id in(select customer_id
                     from account1
                     group by customer_id
                     having count(account_id)>1);
/*3. Retrieve accounts with transactions whose amounts exceed the average
transaction amount.*/
select account_id
from transaction
where amount > (select avg(amount) as amt from transaction);
-- 4. Identify customers who have no recorded transactions.
select id,first_name
from customer
where id IN (select customer_id from account where id NOT IN
(select account_id from transaction));
-- 5. Calculate the total balance of accounts with no recorded transactions.
select sum(balance)
from account1
where amount in(select amount
```

```
        from transaction1
        where amount=0);
-- 6. Retrieve transactions for accounts with the lowest balance.
select *
from transaction1 where account_id in(select account_id from account1
where balance=(select min(balance)from account1));
/*7. Identify customers who have accounts of multiple types.*/
select first_name
from customer1
where account_type in(select account_type
                      from account1
                      group by account_type
                      having count(customer_id>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=1);
/*10. Calculate the total balance for each account type, including a subquery within the SELECT
clause.*/
select account_type,sum(balance) as bal
from account1
group by account_type;
```