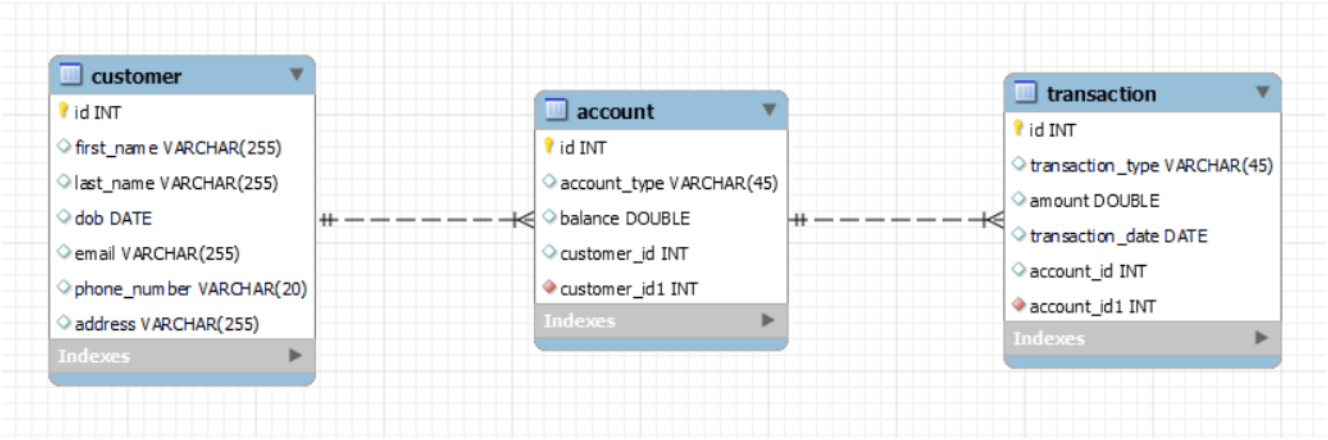
ASSIGNMENT 3 – BANKING SYSTEM

ER DIAGRAM:



Queries:

```
create database bankingsystem;
use bankingsystem;

-- Task 1:
Q6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

-- creation of tables.....
create table customer(id int primary key not null,
 first_name varchar(255), last_name varchar(255),
 dob date, email varchar(255),
 phone_number varchar(20), address varchar(255));

create table account(id int primary key not null, account_type varchar(255),
 balance double, customer_id int, foreign key(customer_id) references customer(id));

create table transaction(id int primary key not null, transaction_type varchar(255),
 amount double, transaction_date date, account_id int,
 foreign key(account_id) references account(id));

-- Task 2:
-- Q1. Insert at least 10 sample records into each of the following tables. Customer, Account, Transaction

-- insertion of values in the table.....
insert into customer (id, first_name, last_name, dob, email, phone_number, address)
values
(1, 'Arjun', 'Kumar', '1990-05-15', 'arjun.kumar@gmail.com', '1234567890', '123 Main St'),
(2, 'Deepika', 'Nair', '1985-08-21', 'deepika.nair@gmail.com', '9876543210', '456 Oak Ave'),
(3, 'Siddharth', 'Menon', '1992-03-10', 'siddharth.menon@gmail.com', '5551234567', '789 Pine St'),
(4, 'Aishwarya', 'Balan', '1988-12-05', 'aishwarya.balan@gmail.com', '3332221111', '321 Elm St'),
```

```
(5, 'Pranav', 'Rajan', '1995-06-30', 'pranav.rajan@gmail.com', '7778889999', '654 Birch Ave'),
(6, 'Nithya', 'Suresh', '1982-09-18', 'nithya.suresh@gmail.com', '1115556666', '987 Cedar St'),
(7, 'Aditya', 'Krishnan', '1998-02-22', 'aditya.krishnan@gmail.com', '9990001111', '234 Maple Ave'),
(8, 'Ananya', 'Pillai', '1980-07-12', 'ananya.pillai@gmail.com', '4447778888', '567 Pine St'),
(9, 'Karthik', 'Raman', '1993-11-08', 'karthik.raman@gmail.com', '2223334444', '876 Oak Ave'),
(10, 'Meera', 'Chandran', '1987-04-14', 'meera.chandran@gmail.com', '6669990000', '765 Birch St');

insert into account (id, account_type, balance, customer_id)
values
(101, 'savings', 5000.00, 1),
(102, 'current', 2000.00, 2),
(103, 'savings', 8000.00, 3),
(104, 'current', 3000.00, 4),
(105, 'zero_balance', 6000.00, 5),
(106, 'current', 7000.00, 6),
(107, 'savings', 4000.00, 7),
(108, 'current', 9000.00, 8),
(109, 'savings', 3500.00, 9),
(110, 'zero_balance', 4500.00, 10);

insert into transaction (id, transaction_type, amount, transaction_date, account_id)
values
(1001, 'deposit', 1000.00, '2024-02-01', 101),
(1002, 'withdrawal', 500.00, '2024-02-05', 102),
(1003, 'transfer', 2000.00, '2024-02-10', 103),
(1004, 'withdrawal', 300.00, '2024-02-15', 104),
(1005, 'deposit', 1500.00, '2024-02-20', 105),
(1006, 'withdrawal', 700.00, '2024-02-25', 106),
(1007, 'deposit', 800.00, '2024-03-01', 107),
(1008, 'withdrawal', 1000.00, '2024-03-05', 108),
(1009, 'deposit', 1200.00, '2024-03-10', 109),
(1010, 'transfer', 600.00, '2024-03-15', 110);

-- SQL Queries

-- Q1. 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.id=a.customer_id;

-- Q2. Write a SQL query to list all transaction corresponding customer.
select c.first_name, c.last_name, t.id as Transaction_id, t.transaction_type, t.amount
   from customer c, transaction t, account a
   where t.account_id=a.id and
   a.customer_id=c.id;
```

```sql
-- Q3. Write a SQL query to increase the balance of a specific account by a certain amount.
update account
set balance = balance+500;

-- Q4. 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 customer;

-- Q5. 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';

-- Q6. Write a SQL query to Find customers living in a specific city.
select * from customer where address like '%main st%';

-- Q7. Write a SQL query to Get the account balance for a specific account
select id, balance
    from account
    where id=101;

-- Q8. Write a SQL query to List all current accounts with a balance greater than $1,000.
select * from account
where account_type='current' and
balance >=1000;

-- Q9. Write a SQL query to Retrieve all transactions for a specific account.
select * from transaction
where account_id=101;

-- Q10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.
select id , account_type,
   balance, '2%' as interest_rate,
   balance * (2 / 100) AS interest_accrued
   from account
         where account_type = 'savings';
-- Note: not sure with the formula used

-- Q11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.
select * from account where balance<0;
-- considering 0 as overdraft limit

-- Q12. Write a SQL query to Find customers not living in a specific city.
select * from customer where
address not like '%elm st%';
```

```
-- Task-3 :

-- Q1. Write a SQL query to Find the average account balance for all customers.
select avg(balance) as avg_balance
from account;

-- Q2. Write a SQL query to Retrieve the top 10 highest account balances.
 select * from account
 order by balance desc
 limit 10;

 -- Q3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.
 update transaction                    -- updated the table to get multiple transaction on same date
   set transaction_date = '2024-02-01'
   where id=1003 and id=1010 and id=1005;

select sum(amount) as total_deposit
from transaction where transaction_type='deposit'
and transaction_date='2024-02-01';

-- Q4. Write a SQL query to Find the Oldest and Newest Customers.
 select min(dob) from customer;
 select concat(first_name,' ',last_name) as customer_name, 'oldest' as cus_type, dob from customer
where dob= (select min(dob) from customer)
 union
 select concat(first_name,' ',last_name) as customer_name, 'newest' as cus_type, dob from customer
where dob= (select max(dob) from customer);

 -- Q5. Write a SQL query to Retrieve transaction details along with the account type.
 select transaction.*, account.account_type from transaction
 join account where account.id=transaction.account_id;

 -- Q6. Write a SQL query to Get a list of customers along with their account details.
 select c.first_name as customer_name, a.*
 from customer c join account a
 where c.id=a.customer_id;

 -- Q7. Write a SQL query to Retrieve transaction details along with customer information for a specific
account
 select t.*, c.first_name from transaction t join account a on a.id=t.account_id
   join customer c on c.id=a.customer_id;

-- Q8. Write a SQL query to Identify customers who have more than one account
 update account              -- updated for getting more than one account for same customer
   set customer_id=3
   where id=106;
```

```
select c.id, c.first_name as customer_name from customer c
join account a on c.id=a.customer_id
group by customer_id
having count(a.customer_id)>1;
```

-- Q9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals

```
select sum(case when transaction_type='deposit'then amount else 0 end) as total_deposit,
sum(case when transaction_type='withdrawal'then amount else 0 end) as total_withdrawal,
(sum(case when transaction_type='deposit' then amount else 0 end)) - (sum(case when transaction_type='withdrawal' then amount else 0 end)) as difference
from transaction;
```

-- refered internet for syntax

-- Q10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

-- not sure about the query

-- Q11. Calculate the total balance for each account type.

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

-- Q12. Identify accounts with the highest number of transactions order by descending order.

update transaction set account_id=101 where id=1006; -- updated for getting same account with multiple transaction

```
select a.id, count(t.account_id) as total_transactiom
   from account a left join
   transaction t on a.id=t.account_id
   group by a.id
   order by count(t.account_id) desc;
```

-- Q13. List customers with high aggregate account balances, along with their account types.

```
select
   c.id ,
   c.first_name,
   a.account_type,
   sum(a.balance)/count(a.customer_id) as aggregate_balance
from customer c
left join account a ON c.id = a.customer_id
group by a.customer_id
order by aggregate_balance desc;
```

-- Q14. Identify and list duplicate transactions based on transaction amount, date, and account.

-- not sure about the query

```
-- Task 4:
-- Q1. Retrieve the customer(s) with the highest account balance.
select c.id, c.first_name, a.balance
   from customer c join
   account a on c.id=a.customer_id and a.balance=(select max(balance) from account);

-- Q2. Calculate the average account balance for customers who have more than one account.
select c.id, c.first_name, sum(a.balance)/count(a.customer_id) as aggregate_balance,
count(a.customer_id) as no_of_accounts
from customer c join account a
on c.id=a.customer_id group by a.customer_id
having count(a.customer_id)>1;

-- Q3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.
select c.id as customer_id, c.first_name, a.id as account_id,
   t.id as transaction_id, t.amount
   from transaction t join account a
   on t.account_id=a.id
   join customer c
   on a.customer_id=c.id
   having t.amount>(select avg(amount) from transaction);

-- Q4. Identify customers who have no recorded transactions
select c.id, c.first_name
from customer c
join account a on c.id=a.customer_id
join transaction t on a.id=t.account_id
where t.id is null;

-- Q5. Calculate the total balance of accounts with no recorded transactions.
select sum(balance) as total_balance
   from account
   where id not in
   (select account_id from transaction);

-- Q6. Retrieve transactions for accounts with the lowest balance.
select * from transaction
    where account_id=
   (select id from account where balance=(select min(balance) from account));

-- Q7. Identify customers who have accounts of multiple types.
select c.id AS customer_id, c.first_name
from customer c
join account a ON c.id = a.customer_id
group by c.id, c.first_name
having count(distinct a.account_type) > 1;
```

```
-- Q8. Calculate the percentage of each account type out of the total number of accounts
select account_type, count(*) as total_accounts,
   (count(*) / (select count(*) from account)) * 100 as percentage
   from account
         group by account_type;

-- Q9. Retrieve all transactions for a customer with a given customer_id.
select t.* from transaction t
join account a on a.id=t.account_id join
customer c on c.id=a.customer_id
where c.id=1;

-- Q10. 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;
```