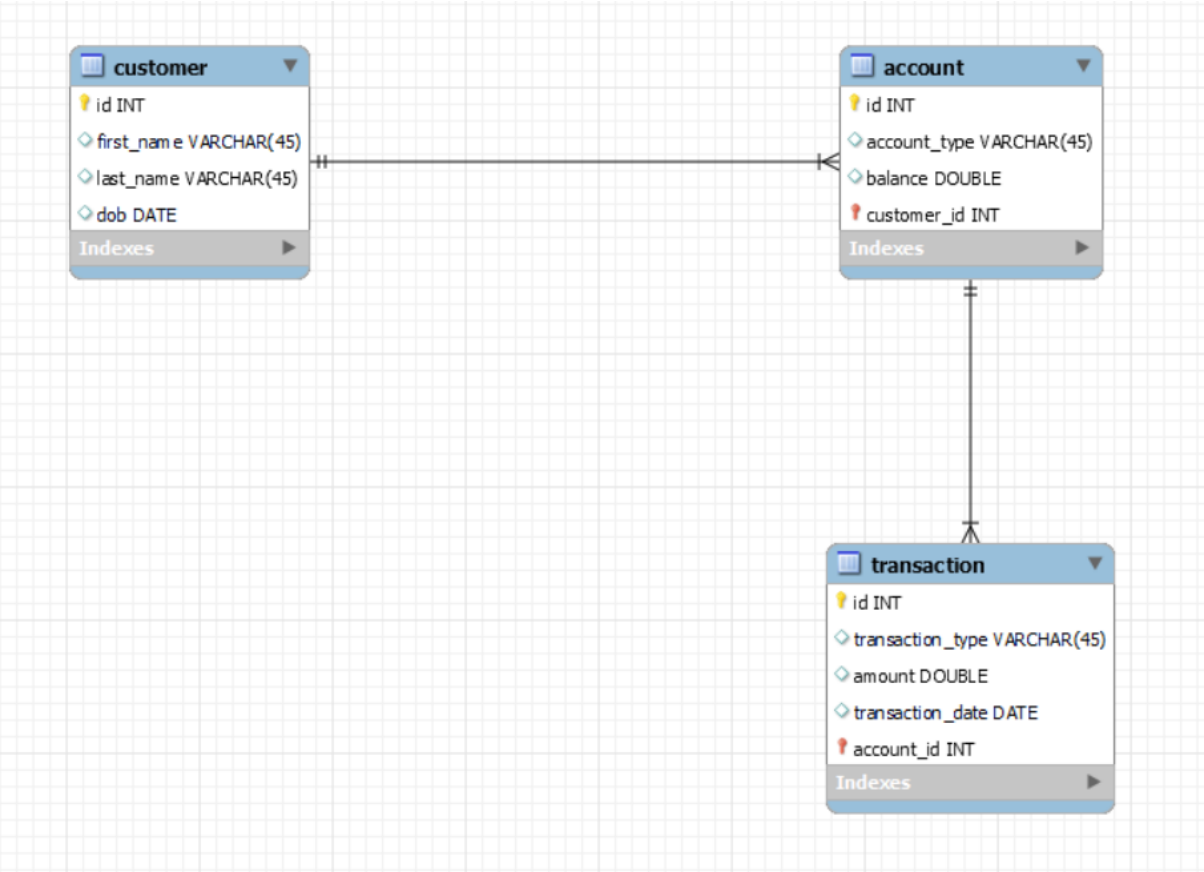
BANKING SYSTEM



CODE:

```
use bank_hex_feb_24;

insert into customer(first_name,last_name,dob) values
('harry','potter','2002-03-21'),
('ronald','weasley','2001-02-10'),
('hermione','granger','2002-11-15');

insert into account(account_type,balance,customer_id) values
('savings',50000,1) ,
('current',120000,2) ,
('zero_balance',100000,3),
('current',150000,1) ,
('savings',30000,3);
```

```
insert into transaction(transaction_type,amount,transaction_date,account_id)
values
('deposit', 10000, '2024-02-01',1),
('withdrawal', 5000, '2024-02-02',1),
('deposit', 20000, '2024-02-02',2),
('withdrawal', 8000, '2024-02-02',3),
('transfer', 20000, '2024-02-01',4),
('transfer', 7000, '2024-02-05',5);
```

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, a.account_type
from customer c, account a
where c.id = a.customer_id;

-- 2. Write a SQL query to list all transaction corresponding customer.
select t.id, t.transaction_type, t.amount, t.transaction_date, c.first_name, c.last_name
from transaction t, customer c, account a
where t.account_id = a.id AND a.customer_id = c.id;

-- 3. Write a SQL query to increase the balance of a specific account by a certain amount.
update account
SET balance = balance + 100
where id = 2;

-- 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 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';
select * from account;

-- 6. Write a SQL query to Find customers living in a specific city.
select *
from customer
where address = 'chennai';  # city is not present in database

-- 7. Write a SQL query to Get the account balance for a specific account.
select id, balance
from account
where id = '3';

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

-- 9. Write a SQL query to Retrieve all transactions for a specific account.
select *
from transaction
where account_id =1;

-- 10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.
select id, (balance * 0.5) as interest_accrued
from account
where account_type='savings';
```

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

```
select *
from account
where balance <4000;
```

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

```
select CONCAT(first_name, ' ', last_name) as name
from customer
where address != 'chennai';    # city is not present in database
```

Task 3

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

```
select customer_id, AVG(balance)
from account
group by customer_id;
```

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

```
select balance
from account
order by balance DESC
limit 0,3;
```

-- 3. Write a SQL query to Calculate Total Deposits for All Customers in specific date. Also display name of the customer

```
select c.first_name,c.last_name,t.transaction_type, t.amount, t.transaction_date
from transaction t JOIN account a ON a.id = t.account_id JOIN customer c ON c.id = a.customer_id
where t.transaction_date = '2024-02-02' AND t.transaction_type='withdrawal';
```

```
-- 4. Write a SQL query to Find the Oldest and Newest Customers.
(select first_name,dob,'oldest' as status from customer order by dob limit 0,1)
UNION
(select first_name,dob,'youngest' as status from customer order by dob DESC limit 0,1);

-- 5. Write a SQL query to Retrieve transaction details along with the account type.
select distinct a.id, t.transaction_type, t.amount, t.transaction_date, a.account_type
from transaction t
JOIN account a ON t.account_id = a.id;

-- 6. Write a SQL query to Get a list of customers along with their account details.
 select CONCAT(c.first_name,' ', c.last_name) as name, a.account_type, a.balance
 from customer c, account a
 where c.id= a.customer_id;

-- 7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.
 select distinct t.account_id, t.transaction_type, t.amount, t.transaction_date, c.first_name, c.last_name, a.account_type, a.balance
from customer c
JOIN account a on a.customer_id = c.id
inner join transaction t on t.account_id = a.id
where a.id=3;

-- 8. Write a SQL query to Identify customers who have more than one account.
select c.first_name,count(c.id) as Number_of_accounts
from customer c JOIN account a ON c.id = a.customer_id
-- where count(c.id) > 1 - 0          Invalid use of group function
group by a.customer_id
having Number_of_accounts>1;
```

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

```
select MAX(amount) - MIN(amount) as difference

from

((select transaction_type ,SUM(amount) as amount, 'deposit' as op

from transaction

where transaction_type ='deposit' )

union

(select transaction_type , SUM(amount) as amount, 'withdrawal' as op

from transaction

where transaction_type ='withdrawal')) AS T;


select

((select SUM(amount)

from transaction

where transaction_type ='deposit' ) -  (select  SUM(amount)

from transaction

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 a.account_type, AVG(a.balance) as Average_balance

from account a join transaction t on a.id=t.account_id

where t.transaction_date between '2024-02-01' AND '2024-02-05'

group by a.account_type;
```

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

```
select account_type, SUM(balance) as total_balance

from account

group by account_type;
```

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

```
select a.id, a.account_type, a.balance, COUNT(t.id) as transaction_count
from account a
JOIN transaction t on a.id = t.account_id
group by a.id
order by transaction_count DESC;

-- 13. List customers with high aggregate account balances, along with their account types.
select c.first_name, a.account_type , sum(a.balance) as account_balance
from customer c join account a on c.id=a.customer_id
group by a.account_type
order by account_balance desc
limit 0 ,1;

-- 14. Identify and list duplicate transactions based on transaction amount, date, and account
select amount , transaction_date , account_id , count(*) as duplicates
from transaction
group by account_id
having duplicates>1;


# Task 4

-- 1. Retrieve the customer(s) with the highest account balance.
select id, first_name, last_name
from customer
where id = (select customer_id
        from account
        order by balance DESC
        limit 0,1);

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

```
-- 3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.
select id, account_type, balance
from account
where id IN (select account_id
        from transaction
        where amount > (select avg(amount)
                                                    from transaction));

-- 4. Identify customers who have no recorded transactions.
select distinct c.id , c.first_name
from customer c
JOIN account a on c.id = a.customer_id
where a.id NOT IN (select account_id
                                    from transaction);

-- 5. Calculate the total balance of accounts with no recorded transactions.
select SUM(a.balance) as total_balance
from account a
left join transaction t on a.id = t.account_id
where t.id is null;

-- 6. Retrieve transactions for accounts with the lowest balance.
select *
from transaction
where account_id = (select id
            from account
            order by balance ASC
            limit 0,1);

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

```
select id, first_name, last_name
from customer
where id IN (select distinct customer_id
        from account
        group by customer_id
        having COUNT(distinct account_type) > 1);

-- 8. Calculate the percentage of each account type out of the total number of accounts.
select account_type,
    count(id) * 100.0 / (select count(*) from account) as percentage
from account
group by account_type;

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

-- 10. Calculate the total balance for each account type, including a subquery within the SELECT clause
select a.account_type,
    (select sum(balance)
                from account
                where account_type = a.account_type) as total_balance
from account a
group by a.account_type;
```