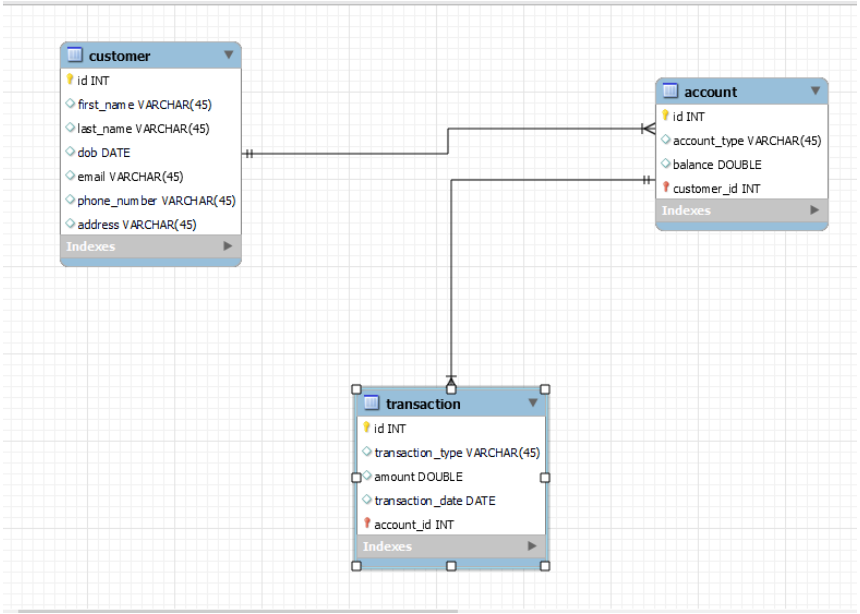
BANKING SYSTEM ASSIGNMENT

ER DIAGRAM:



QUERIES:

use bank_db;

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 customer(first_name,last_name,dob) values

('malar','vizhi','2000-03-31'),

('Sheela','prakash','1998-07-21'),

('prakash','ram','1995-03-09');

----------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),

('savings',300000,4),

('current',110000,5),

('zero_balance',800000,6);

---

insert into transaction(transaction_type,amount,transaction_date,account_id)

values

('deposit', 10000, '2024-02-01',1),

('withdrawal', 5000, '2024-02-02',6),

('deposit', 20000, '2024-02-02',2),

('withdrawal', 8000, '2024-02-02',3),

('transfer', 20000, '2024-02-01',4),

('transfer', 7000, '2024-02-05',5),

('deposit', 90000, '2024-02-01',7);

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 first_name,last_name,account_type
from customer c join account a ON c.id=a.customer_id;
```

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

```
select t.id,t.amount,t.transaction_date,c.id, c.first_name
from  transaction t
join customer c on t.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+10000
```

where account.id=3;

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';

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 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 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 t.account_id = a.id;

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

select id,balance * (interest_rate / 100) 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 < overdraft_limit;

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

select first_name,last_name

from customer

where address NOT LIKE '%bangalore%';

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 t.id,t.transaction_type, t.transaction_date,t.amount,

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 c.first_name,c.last_name,a.id,a.account_type,a.balance

from customer c join account a on c.id = a.customer_id;

7.Write a SQL query to Retrieve transaction details along with customer information for a specific account.

select t.id,t.transaction_date, t.amount,c.id,c.first_name

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

```
join customer c on a.customer_id = c.id

 where a.id = 4;
```

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;
```

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

11. Calculate the total balance for each account type.

```
select account_type, sum(balance) as total_amount

from account

group by account_type;
```

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

```
  select   account_id, count(*) as transaction_count

from transaction

group by account_id
```

```
order by transaction_count desc;
```

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

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

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

```
select amount, transaction_date, account_id, count(*) as duplicate_count
from transaction
group by amount, transaction_date,account_id
having  count(*) > 1;
```

Tasks 4: Subquery and its type:

1. 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
where a.balance = (select max(balance) from account);
```

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

```
select avg(balance)
from account
where customer_id IN (select customer_id
                      from account
                      group by customer_id
                      having count(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 );
```

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));
-- troubleshooting
select distinct account_id from transaction; -- (1,2,3,4,5)
select customer_id from account where id NOT IN (1,2,3,4,5); -- (6)
select * from customer where id IN (4);
```

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

```
select sum(balance) as total_balance
from account
where id not in (
   select distinct account_id
   from transaction );
```

6. Retrieve transactions for accounts with the lowest balance.

```
select *
from transaction
where account_id in (select 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(distinct account_type) > 1;
```

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

```
Select account_type, count(*) as num_accounts,
```

```
    (count(*) * 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
where account_id IN (select 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 total_balance
from account
group by account_type;
```