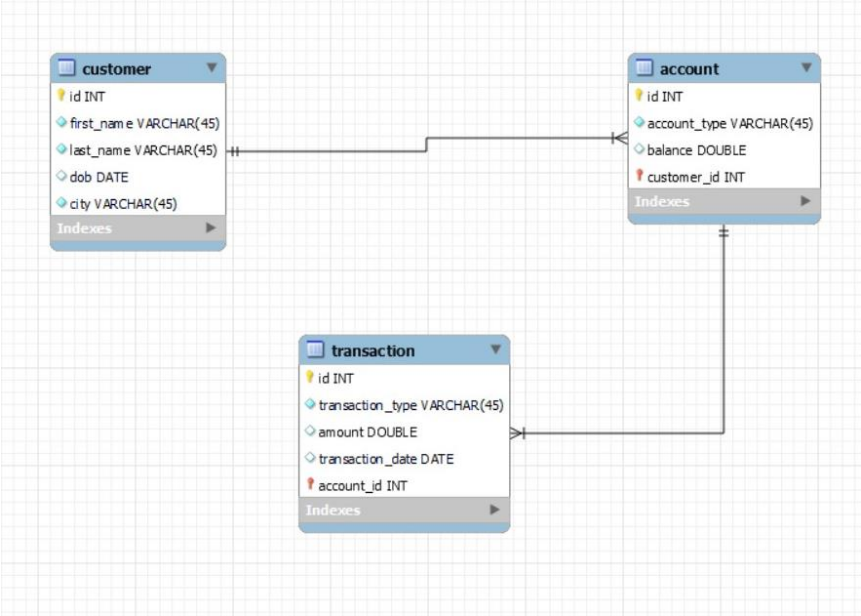
Bank System



use bank;

show tables;

describe customer;

insert into customer(first_name,last_name,dob,city) values

('harry','potter','2002-03-21',"chennai"),

('ronald','weasley','2001-02-10','Mumbai'),

('hermione','granger','2002-11-15','bangalore');

select * from customer;

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);
select * from account;

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);
select * from transaction;
-- Task 2
/*
1. Write a SQL query to retrieve the name, account type and email of all customers. //
since we have not collected email we will not be displaying them.
*/
select c.first_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 *
FROM customer c
JOIN transaction t ON t.account_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 + 6000

WHERE customer_id = 2;

select * from account;

/*

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'; #doubt

/*

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

*/

SELECT * from customer

Where city in("bangalore");

/*

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

*/

SELECT balance

FROM account

```
WHERE customer_id = 2;

/*

8. 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;

/*

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

*/

SELECT *

FROM transaction t

WHERE t.account_id = 1;

/*

10. Write a SQL query to Calculate the interest acquired on savings accounts based on a

given interest rate.

*/

select (sum(balance)*0.1) as Interest_collected

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 c.first_name,a.account_type,a.balance

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

```
where a.balance <150000;
/*
12. Write a SQL query to Find customers not living in a specific city.
*/
select *
from customer
where city not in ("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.amount,t.transaction_date,a.account_type
from transaction t , account a
where 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,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 c.first_name,t.transaction_type,t.amount
from customer c ,account a, transaction t
where c.id=a.customer_id and a.id=t.account_id and a.id = 2;
```

```
/*
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;

/*
We find deposit amount using 1 query
and withdrawal amount using another query.
```

then we bring the result togather in a Derived table called as T.

T

--

30K

13K

from this T(which is a table in itself) we compute MAX and MIN and do the arithmetic and get the result

*/

-- alternatively

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 c.id,avg(a.balance)

from account a

join customer c on c.id=a.customer_id

group by a.customer_id;

```
/*
11. Calculate the total balance for each account type.
*/
select a.account_type,sum(a.balance)
from account a
group by a.account_type;
/*
12. Identify accounts with the highest number of transactions order by descending order.
*/
select c.first_name,a.account_type,t.account_id,count(t.id) as Count_transactions
from account a
join transaction t on a.id=t.account_id
join customer c on c.id=a.customer_id
group by t.account_id
order by no_of_transaction desc
limit 1;
/*
13. List customers with high aggregate account balances, along with their account types.
*/
select c.id,c.first_name,a.account_type,a.balance
from customer c
join account a on c.id=a.customer_id
order by a.balance desc
limit 1;
/*
14. Identify and list duplicate transactions based on transaction amount, date, and account
```

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

-- Task 4
/*
1. Retrieve the customer(s) with the highest account balance.
*/
select * from account
where 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 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));
/*

5. Calculate the total balance of accounts with no recorded transactions.
*/
select sum(balance) from account where id NOT IN(select
account_id from transaction);

/*
6. Retrieve transactions for accounts with the lowest balance.
*/
select t.* from transaction t join account a on t.account_id=a.id
where a.balance=(select
min(balance) from account);

/*
7. Identify customers who have accounts of multiple types.
```

```
*/
select * from customer where id in (select
a.customer_id from account a group by a.customer_id
having count(distinct a.account_type)>1);
use banking;

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