```
use banking_db;
show tables;
select * from transaction;
select* from account;
select * from customer;
```

task 3 9,10

```
-- Task 2
/*
 Write SQL queries for the following tasks:

1. Write a SQL query to retrieve the name, account type and email of all customers.

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

3. Write a SQL query to increase the balance of a specific account by a certain amount.

4. Write a SQL query to Combine first and last names of customers as a full_name.

5. Write a SQL query to remove accounts with a balance of zero where the account
type is savings.

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

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

.

8. Write a SQL query to List all current accounts with a balance greater than $1,000.

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

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

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

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

*/

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

select c.id,c.first_name,t.transaction_type,t.amount
 from transaction t JOIN account a ON t.account_id = a.id
 JOIN customer c ON c.id=a.customer_id;

update account set balance = '56000' where id=3;

/* CONCAT */
```

```
select concat(first_name,' ',last_name) as full_name from customer;



delete from account
where account_type = 'saving' and balance = 50000;




select concat(first_name,' ',last_name) as full_name from customer where city='chennai';

select balance from account where id=1;

select id,balance as current from account where balance > 50000;

select * from transaction where account_id=2;

select * from customer where city !='chennai';


SELECT id, balance * 10 AS interest_accrued
FROM account;

SELECT id, balance
FROM account
WHERE balance < 1000;
```

```
/* task 3 */
/* 1. Write a SQL query to Find the average account balance for all customers.  */
select avg(balance),customer_id from account
group by customer_id;


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

select balance,customer_id from account group by balance 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.transaction_date,t.amount
from transaction t JOIN account a ON t.account_id=a.id JOIN customer c ON c.id=a.customer_id
where transaction_date='2024-02-01' and transaction_type = 'deposit';


/* 4. Write a SQL query to Find the Oldest and Newest Customers. */
(select first_name,last_name,dob,'oldest' from customer order by dob limit 0,1)
UNION
(select first_name,last_name,dob,'young' 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.

*/

/*

6. Write a SQL query to Get a list of customers along with their account details.

*/

/*

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

*/

```
select t.transaction_type,t.transaction_date,t.amount,a.account_type
from account a JOIN transaction t ON t.account_id = a.id;

select c.first_name,c.last_name,a.account_type,a.balance from
customer c JOIN account a ON c.id= a.customer_id
group by customer_id;

select c.first_name,c.last_name,
t.transaction_type,t.amount,t.transaction_date,a.account_type,a.balance from
customer c  JOIN account a ON c.id = a.customer_id JOIN transaction t ON t.account_id = a.id
where a.account_type='savings';
```

/*

8. Write a SQL query to Identify customers who have more than one account.

*/

```
select c.first_name,c.last_name,count(c.id) from
customer c JOIN account a On c.id= a.customer_id
group by a.customer_id
having count(c.id)>1;
```

/*

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

*/

/*

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

```
SELECT a.id,t.transaction_date,
AVG(a.balance) AS average_daily_balance
FROM account a JOIN transaction t On t.account_id=a.id having t.transaction_date
between '2024-02-01' and '2024-02-05';
```

/*11. Calculate the total balance for each account type.

*/

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

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

*/

```
select id,count(id) as highest_transaction from account group by id order by highest_transaction
desc;
```

/*

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

*/

```
select c.first_name,a.account_type,a.balance from account a JOIN customer c ON
c.id=a.customer_id;


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


select id,transaction_type,transaction_date,amount,count(account_id) as duplicate from transaction
group by account_id;









/* SUB QUERY */


/*Task 4: Sub query*/




/*
        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);



select * from account;


/*
10. Calculate the total balance for each account type, including a subquery within the SELECT
clause.
```

```
*/

select sum(balance),account_type 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);


 /*insert into customer(first_name,last_name,dob) values ('draco','malfoy','2000-05-06');

*/

 /*
4. Identify customers who have no recorded transactions.
*/

select customer_id from account where id NOT IN (1,2,3,4,5);


select * from account;


/*
Tasks 4: Subquery and its type:
```

1. Retrieve the customer(s) with the highest account balance.

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.

4. Identify customers who have no recorded transactions.

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

6. Retrieve transactions for accounts with the lowest balance.

7. Identify customers who have accounts of multiple types.

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.

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

*/

```
select * from customer;
select * from account;
```

```
select id from customer where id IN
(select balance from account order by balance desc);
```

```
select avg(balance) from account where customer_id in
(select customer_id from account group by customer_id having count(customer_id)>1);
```

/* 3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

4. Identify customers who have no recorded transactions.*/

```
select * from transaction;
select * from customer;


select id from transaction where amount >( select avg(amount) from transaction);


 select id,first_name
 from customer
 where NOT EXISTS (select distinct c.id
                                    from customer c join account a ON c.id=a.customer_id);


 /*
5. Calculate the total balance of accounts with no recorded transactions.
6. Retrieve transactions for accounts with the lowest balance.
7. Identify customers who have accounts of multiple types.*/


select sum(balance) from account where id not in(select distinct a.id from account a
JOIN transaction t
ON t.account_id=a.id);

select * from transaction where id =
(select id from account order by balance limit 1 );

select * from customer where id in(select id from account
 where account_type in('savings','current','zero balance'));


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

```
 */
 SELECT
    account_type,
    COUNT(*) * 100.0 / (SELECT COUNT(*) FROM account) AS percentage
FROM
    account
GROUP BY
    account_type;

 /* doubt*/

 .
/*9. Retrieve all transactions for a customer with a given customer_id. /
*/
select * from transaction where 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) from account group by account_type;
```