```
show databases;
use banking_system;
show tables;

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

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
/*
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
where c.id=a.customer_id;
/*
Output
      first_name last_name   account_type
      harry potter      savings
      ronald     weasley     current
      hermione   granger     zero_balance
      harry potter      current
      hermione   granger     savings
*/
```

```
-- 2. Write a SQL query to list all transaction corresponding customer.
select
c.first_name,c.last_name,t.transaction_type,t.amount,t.transaction_date
from customer c
JOIN account a ON c.id = a.customer_id
JOIN transaction t ON a.id = t.account_id;
/*
Output
      first_name last_name  transaction_type amount      transaction_date
      harry potter     deposit    10000 2024-02-01
      harry potter     withdrawal 5000  2024-02-02
      harry potter     transfer   20000 2024-02-01
      ronald     weasley    deposit     20000 2024-02-02
      hermione   granger    withdrawal  8000  2024-02-02
      hermione   granger    transfer    7000  2024-02-05
*/
```

```
-- 3. Write a SQL query to increase the balance of a specific account by a
certain amount.
update account
set balance = balance + 10000
where id=2;
select * from account where id=2;
/*
Output
      id   account_type     balance     customer_id
      2    current     130000     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;
/*
Output
      full_name
      harry potter
      ronald weasley
      hermione granger
*/
```

```sql
-- 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';
/*
Output
No output
*/
```

```sql
-- 6. Write a SQL query to Find customers living in a specific city.
select *
from customer
where city='mumbai';
/*
Output
No ouput
*/
```

```sql
-- 7. Write a SQL query to Get the account balance for a specific account.
select balance
from account
where id=1;
/*
Output
      balance
      50000
*/
```

```sql
-- 8. Write a SQL query to List all current accounts with a balance
greater than $1,000.
select *
from account
where balance>1000 and account_type='current';
/*
Output
      id    account_type     balance    customer_id
      2     current    130000     2
      4     current    150000     1

*/
```

```
-- 9. Write a SQL query to Retrieve all transactions for a specific
account.
select *
from transaction
where id=1;
/*
Output
      id    transaction_type amount      transaction_date account_id
      1     deposit    10000 2024-02-01  1

*/
```

```
-- 10. Write a SQL query to Calculate the interest accrued on savings
accounts based on a given interest rate.
select id, balance * 0.10 AS interest
from account
where account_type = 'savings';
/*
Output
      id    interest
      1     5000
      5     3000
*/
```

```
-- 11. Write a SQL query to Identify accounts where the balance is less
than a specified overdraft limit.
select *
from account
where balance <=50000;
/*
Output
      id    account_type     balance     customer_id
      1     savings    50000 1
      5     savings    30000 3

*/
```

```
-- 12. Write a SQL query to Find customers not living in a specific city.
select *
from customer
where city<>'delhi';
/*
```

```
Output
No output
*/




-- Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:
-- 1. Write a SQL query to Find the average account balance for all
customers.
select c.id,avg(a.balance)
from customer c JOIN account a ON c.id=a.customer_id
group by c.id;
/*
Output
      id    avg(a.balance)
      1     100000
      2     130000
      3     65000
*/




-- 2. Write a SQL query to Retrieve the top 10 highest account balances.
select balance
from account
order by balance desc
limit 10;
/*
Output
      balance
      150000
      130000
      100000
      50000
      30000
*/




-- 3. Write a SQL query to Calculate Total Deposits for All Customers in
specific date.
select c.*,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='deposit';
/*
Output
      id    first_name  last_name   dob   transaction_type  amount
      transaction_date
      2     ronald      weasley     2001-02-10  deposit     20000 2024-02-02
*/
```

```
-- 4. Write a SQL query to Find the Oldest and Newest Customers.
select first_name,MIN(dob) AS oldest_customer,
                 MAX(dob) AS newest_customer
from customer;
/*
Output
      first_name  oldest_customer  newest_customer
      harry 2001-02-10  2002-11-15
*/
```

```
-- 5. Write a SQL query to Retrieve transaction details along with the
account type.
select t.*,a.account_type
from account a JOIN transaction t ON a.id=t.account_id ;
/*
Output
      id    transaction_type amount     transaction_date account_id
      account_type
      1     deposit    10000 2024-02-01  1     savings
      2     withdrawal  5000  2024-02-02  1     savings
      3     deposit    20000 2024-02-02  2     current
      4     withdrawal  8000  2024-02-02  3     zero_balance
      5     transfer   20000 2024-02-01  4     current
      6     transfer   7000  2024-02-05  5     savings
*/
```

```
-- 6. Write a SQL query to Get a list of customers along with their
account details.
select c.*,a.*
from customer c JOIN account a ON c.id=a.customer_id
group by c.id;
/*
Output
      id    first_name  last_name   dob   id    account_type     balance
      customer_id
      1     harry potter      2002-03-21  1     savings     50000 1
      2     ronald      weasley     2001-02-10  2     current    130000
      2
      3     hermione    granger     2002-11-15  3     zero_balance
      100000     3
*/
```

```
-- 7. Write a SQL query to Retrieve transaction details along with
customer information for a specific account.
select c.*,t.*
from customer c JOIN account a ON c.id=a.customer_id
                                  JOIN transaction t ON
a.id=t.account_id;
/*
Output
      id    first_name  last_name   dob   id    transaction_type  amount
      transaction_date  account_id
      1     harry potter      2002-03-21  1     deposit     10000 2024-02-01
      1
      1     harry potter      2002-03-21  2     withdrawal  5000  2024-02-02
      1
      1     harry potter      2002-03-21  5     transfer    20000 2024-02-01
      4
      2     ronald      weasley     2001-02-10  3     deposit     20000 2024-
02-02 2
      3     hermione    granger     2002-11-15  4     withdrawal  8000  2024-
02-02 3
      3     hermione    granger     2002-11-15  6     transfer    7000  2024-
02-05 5
*/
```

```
-- 8. Write a SQL query to Identify customers who have more than one
account.
select c.first_name,c.last_name,count(a.account_type) as count
from customer c JOIN account a ON c.id=a.customer_id
group by a.account_type
having count>1;
/*
Output
      first_name  last_name   count
      ronald      weasley     2
      harry potter      2
*/
```

```
-- 11. Calculate the total balance for each account type.
select account_type,sum(balance) as total_balance
from account
group by account_type;
/*
Output
      account_id  transaction_count
      5     1
      4     1
      3     1
      2     1
      1     2
*/
```

```
-- 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 account_id desc;
/*
Output
      account_id  transaction_count
      5     1
      4     1
      3     1
      2     1
      1     2
*/
```

```
-- 13. List customers with high aggregate account balances, along with
their account types.
select c.first_name,c.last_name,a.account_type,sum(a.balance) as
total_balance
from customer c
join account a on c.id = a.customer_id
group by c.id, a.account_type
order by total_balance desc;
/*
Output
      first_name  last_name   account_type     total_balance
      harry potter      current    150000
      ronald      weasley     current    130000
      hermione    granger     zero_balance     100000
      harry potter      savings    50000
      hermione    granger     savings    30000
*/
```

```
-- Task 4
-- 1. Retrieve the customer(s) with the highest account balance.
select c.id,c.first_name,c.last_name,max(a.balance) as highest_balance
from customer c
join ccount a on c.id = a.customer_id
```

```
group by c.id, c.first_name, c.last_name
order by highest_balance desc
limit 1;
/*
Output
      id   first_name last_name   highest_balance
      1    harry potter     150000
*/



-- 2. Calculate the average account balance for customers who have more
than one account.
select customer_id
from account
group by customer_id
having count(id) > 1;
/*
Output
      customer_id
      1
      3
*/


-- 3. Retrieve accounts with transactions whose amounts exceed the average
transaction amount.
select a.*
from account a
join transaction t on a.id = t.account_id
where t.amount > (select avg(amount) from transaction);
/*
Output
      id   account_type     balance    customer_id
      2    current    130000     2
      4    current    150000     1
*/




-- 4. Identify customers who have no recorded transactions.
select c.*
from customer c
left join transaction t on c.id = t.id
where t.id is null;
/*
Output
No output
*/
```

```
-- 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.account_id is null;
/*
Output
No Output
*/
```

```
-- 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);
/*
Output
      id    transaction_type amount      transaction_date account_id
      6     transfer    7000  2024-02-05  5
*/
```

```
-- 7. Identify customers who have accounts of multiple types.
select c.*
from customer c
join account a on c.id = a.customer_id
group by c.id
having count( a.account_type) > 1;
/*
Output
      id    first_name  last_name   dob
      1     harry potter      2002-03-21
      3     hermione    granger    2002-11-15
*/
```

```
-- 9. Retrieve all transactions for a customer with a given customer_id.
select t.*
from transaction t
JOIN account a ON t.account_id = a.id
JOIN customer c ON a.customer_id = c.id
where c.id = 3;
/*
Output
      id    transaction_type amount      transaction_date account_id
      4     withdrawal  8000  2024-02-02  3
      6     transfer    7000  2024-02-05  5
*/
```

```
-- 10. Calculate the total balance for each account type.
select a.account_type,sum(a.balance) as total_balance
from account a
group by a.account_type;
/*
Output
      account_type     total_balance
      current    280000
      savings    80000
      zero_balance     100000
*/
```