```
use bank_hex;
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');

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);

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

select c.*, a.account_type
from customer c join account a
ON c.id = a.customer_id;
/*
1          harry     potter   2002-03-21       savings
```

1 harry potter 2002-03-21 current

2 ronald weasley 2001-02-10 current

3 hermione granger 2002-11-15 zero_balance

3 hermione granger 2002-11-15 savings

*/

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

```
select c.*, t.*
from customer c join account a  ON c.id = a.customer_id
join transaction t ON a.id = t.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

*/

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

```
update account
SET balance = 7000
where id = 1;
```

```
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;
/*
harry potter
ronald weasley
hermione granger
*/

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

delete
from account
where account_type = 'savings' and balance = '0';

-- 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.

select balance
from account
where id = '1';
/*
7000
*/

-- 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';
```

```
/*
2	current 120000 2
4	current 150000 1

*/

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

select a.account_type, t.transaction_type, t.amount, t.transaction_date
from account a join transaction t
ON a.id = t.account_id
where account_type = 'savings';

/*
savings deposit 10000   2024-02-01
savings withdrawal      5000    2024-02-02
savings transfer7000    2024-02-05
*/

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

select id , balance* (12/100) as interest_accrued
from account where account_type='savings';

/*
+----+------------------+
| id | interest_accrued |
+----+------------------+
|  1 |        840.0000 |
```

| 5 | 3600.0000 |

+----+------------------+

*/

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

select *

from account

where balance < '10000';

/*

1 savings 7000 1

*/

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

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

/* projection : customer

criteria : account

*/

select c.first_name , AVG(a.balance) as avg_account_balance

from customer c join account a

ON c.id = a.customer_id

group by c.first_name;

/*

harry 78500.0000

ronald 120000.0000

```
hermione        65000.0000
   */

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

   select c.id,c.first_name,a.balance
   from customer c join account a
   ON c.id = a.customer_id
   order by balance desc
   limit 10
   ;

   /*
   1       harry    150000
2          ronald   120000
3          hermione         100000
3          hermione         30000
1          harry    7000
   */

   -- 3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

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

   /*
```

```
1	harry	potter	2002-03-21	1	savings	7000	1	1	deposit	10000	2024-02-01	1

*/


/*

4. Write a SQL query to Find the Oldest and Newest Customers.

*/


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


 select *

from transaction t

	JOIN account a ON t.account_id = a.id

	JOIN customer c ON a.customer_id = c.id

where t.account_id = 1;


/*

1	deposit	10000	2024-02-01	1	1	savings	7000	1	1	harry	potter	2002-03-21

2	withdrawal	5000	2024-02-02	1	1	savings	7000	1	1	harry	potter	2002-03-21

*/


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


select c.first_name

from customer c

JOIN account a ON a.customer_id = c.id

group by c.id

having count(a.id)>1;
```

```
/*
harry
hermione
*/

-- 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 *, balance*(12/100) as interest_accrued
from account
where account_type='savings';

/*
+----+--------------+---------+-------------+------------------+
| id | account_type | balance | customer_id | interest_accrued |
+----+--------------+---------+-------------+------------------+
|  1 | savings      |    7000 |           1 |        840.0000 |
|  5 | savings      |   30000 |           3 |       3600.0000 |
+----+--------------+---------+-------------+------------------+
*/

-- 11. 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;

/*
        account_type    Total_Balance
        savings 37000
```

```
        current 270000
        zero_balance    100000
*/

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

select a.*, count(a.id) as no_of_transaction
from account a JOIN  customer c
ON a.customer_id = c.id
JOIN transaction t
ON a.id = t.account_id
group by a.id
order by no_of_transaction desc;

/*
1       savings 7000    1       2
4       current 150000 1        1
2       current 120000 2        1
3       zero_balance    100000 3        1
5       savings 30000   3       1
*/


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


select c.first_name, sum(a.balance) as account_aggregate_balance
from account a join customer c on a.customer_id=c.id
group by c.first_name
having account_aggregate_balance>100000;
```

```
/*
harry    157000
ronald   120000
hermione         130000
*/

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



select *
from transaction
where transaction_type in (select transaction_type
                    from transaction
                    group by transaction_type having
                      count(*)>1 )
                      and amount in (select amount
                                  from transaction
                                  group by amount
                                  having count(*)>1 );

/*
3         deposit 20000   2024-02-02       2
5         transfer20000   2024-02-01       4

*/

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

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

```
select max(balance),c.first_name,c.id
from customer c join account a on c.id=a.customer_id
group by a.balance,c.id,c.first_name
having balance=(select max(balance) from account)
order by balance desc limit 1;

/*
150000 harry    1
*/
```

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

```
select c.first_name, avg(balance)
from account a join customer c on c.id=a.customer_id
group by customer_id
having (select count(id) from customer)>1;

/*
harry    78500.0000
ronald   120000.0000
hermione         65000.0000
*/
```

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

```
SELECT a.id, a.account_type,a.balance,t.id,t.amount as transaction_amount
```

```
FROM account a
JOIN transaction t ON a.id = t.account_id
WHERE t.amount > (SELECT AVG(amount) FROM transaction);

/*
2          current 120000 3          20000
4          current 150000 5          20000
*/


-- 4. Identify customers who have no recorded transactions.


 select c.*
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(balance) as total_balance
from account
where id not in(select account_id from transaction);


-- 6. Retrieve transactions for accounts with the lowest balance.

select t.id,t.transaction_type,a.balance,a.id
from account a join transaction t
ON a.id=t.account_id
```

```
group by t.id,a.balance,t.transaction_type,a.id
order by balance asc limit 1;

/*
1	deposit 7000	1
*/

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

select c.*
from customer c join account a
ON a.customer_id=c.id
group by c.id
having count(distinct a.account_type)>1;

/*
1	harry	potter	2002-03-21
3	hermione	granger	2002-11-15
*/

-- 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.
select c.first_name,c.last_name,t.*
from customer c
join account a on a.customer_id=c.id
join transaction t on t.account_id=a.id
where c.id=3;

/*
```

hermione granger 4 withdrawal 8000 2024-02-02 3

hermione granger 6 transfer 7000 2024-02-05 5

*/

-- 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;
```

/*

37000 savings

270000 current

100000 zero_balance

*/