Assignment - 2

```
use banksystem;
show tables;
describe account;
describe customer;
describe transaction;


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;
/*
1	harry	potter	2002-03-21
2	ronald	weasley	2001-02-10
3	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);

select * from account;
/*
1	savings	100	1
```

2 current 120000 2

3 zero_balance 101000 3

4 current 150000 1

5 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);

select * from transaction;
```

/*

1 deposit 10000 2024-02-01 1

2 withdrawal 5000 2024-02-02 1

3 deposit 20000 2024-02-02 2

4 withdrawal 8000 2024-02-02 3

5 transfer 20000 2024-02-01 4

6 transfer 7000 2024-02-05 5

*/

-- 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.

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.

*/

-- 1

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

/*

```
harry      potter savings

ronald     weasley          current

hermione  granger        zero_balance

harry      potter current

hermione  granger        savings
```

*/

-- 2

```
select * from customer c , account a, transaction t where c.id=a.customer_id and a.id = t.account_id;
```

/*

```
1          harry       potter       2002-03-21          1          savings       100          1          1
           deposit    10000 2024-02-01       1

1          harry       potter              2002-03-21 1          savings       100             1       2
           withdrawal       5000     2024-02-02       1

1          harry       potter       2002-03-21          4          current       150000       1          5
           transfer 20000    2024-02-01       4

2          ronald      weasley             2001-02-10 2          current       120000       2          3
           deposit     20000          2024-02-02       2
```

3 hermione granger 2002-11-15 3 zero_balance 101000 3 4 withdrawal 8000 2024-02-02 3

3 hermione granger 2002-11-15 5 savings 30000 3 6 transfer 7000 2024-02-05 5

*/

-- 3

update account set balance = balance +100 where customer_id = 1 and account_type = 'savings';

update account set balance=balance+1000 where account_type='zero_balance' and id=3 ;

-- 4

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

/*

1 harrypotter

2 ronaldweasley

3 hermionegranger

*/

-- 5

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

-- 6

-- 7

select * from account where account_type ='zero_balance' and customer_id=3;

/*

3 zero_balance 101000 3

*/

-- 8

select * from account where account_type = 'current' and balance>1000;

/*

2 current 120000 2

4 current 150000 1

*/

-- 9

```
select a.account_type , t.* from account a , transaction t where a.id = t.account_id and
a.account_type = 'current' and a.customer_id= 1 ;

/*

current 5          transfer 20000   2024-02-01       4

*/

-- 10 no interest given

-- 11

select * from account where balance <50000;

/*

1          savings  100       1

5          savings  30000   3

*/

-- 12



/*

Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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

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

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

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

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.

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

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.

11. Calculate the total balance for each account type.
```

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

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

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

*/

-- 1

```
select AVG(balance) as average_balance from account;
```

/*

80220

*/

-- 2

```
select * from account order by balance DESC limit 10;
```

/*

4 current 150000 1

2 current 120000 2

3 zero_balance 101000 3

5 savings 30000 3

1 savings 100 1

*/

-- 3

```
select SUM(amount) as total_deposits from transaction where transaction_type = 'deposit' and
transaction_date = '2024-02-01';
```

/*

10000

*/

-- 4

```
select * from customer order by dob ASC limit 1; -- Oldest
```

/*

2 ronald weasley2001-02-10

```
*/
select * from customer order by dob desc limit 1; -- Newest
/*
3	hermione	granger	2002-11-15
*/

-- 5
select t.*, a.account_type from transaction t join account a ON t.account_id = a.id;

/*
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
select c.*, a.* from customer c join account a ON c.id = a.customer_id;
/*
1	harry	potter	2002-03-21	1	savings	100	1
2	ronald	weasley	2001-02-10	2	current	120000	2
3	hermione	granger	2002-11-15	3	zero_balance	101000	3
1	harry	potter	2002-03-21	4	current	150000	1
3	hermione	granger	2002-11-15	5	savings	30000	3
*/
-- 7
select t.*, c.* from transaction t join account a ON t.account_id = a.id join customer c ON
a.customer_id = c.id where a.id = 1;
/*
```

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

*/

-- 8

select c.*, COUNT(a.id) AS num_accounts from customer c join account a ON c.id = a.customer_id group by c.id having num_accounts > 1;

/*

1 harry potter 2002-03-21 2

3 hermione granger 2002-11-15 2

*/

-- 9

-- 10

-- 11

select account_type, SUM(balance) AS total_balance from account group by account_type;

/*

current 270000

savings 30100

zero_balance 101000

*/

-- 12

select account_id, COUNT(*) AS num_transactions from transaction group by account_id order by num_transactions DESC;

/*

1 2

2 1

3 1

4 1

5 1

*/

-- 13

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

/*

harry potter 150100 current,savings

ronald weasley 120000 current

hermione granger 131000 zero_balance,savings

*/

-- 14

select transaction_date, account_id, amount, COUNT(*) AS num_duplicates from transaction group by transaction_date, account_id, amount having num_duplicates > 1;

/* null output */

-- ----------task 4 ----------

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

select * from customer where id = (SELECT customer_id from account order by balance DESC limit 1);

/*

1 harry potter 2002-03-21

*/

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

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

```
/*
75050   2
65500   2
*/
```

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

```
select * from account where id IN (SELECT account_id from transaction group by account_id having AVG(amount) > (select AVG(amount) from transaction));
```

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

-- 4 Identify customers who have no recorded transactions.

```
select * from customer where id not in(select DISTINCT customer_id from account);
```

```
/* null output */
```

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

```
select SUM(balance) from account where customer_id IN (select id from customer where id not in (select DISTINCT customer_id from transaction));
```

```
/* null output */
```

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

```
select * from transaction t where t.account_id IN (select a.id from account a order by balance ASC ) limit 1;
```

/*

1 deposit 10000 2024-02-01 1

*/

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

select * from customer where id IN (SELECT customer_id from account group by customer_id having COUNT(DISTINCT 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.

select account_type,(count(*) * 100/(select count(*) from account)) as percentage from account group by account_type;

/*

current 2 40.00000

savings 2 40.00000

zero_balance 1 20.00000

*/

-- 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 = 2);

/*

3 deposit 20000 2024-02-02 2

*/

-- 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;

/*

current        270000

savings            30100

zero_balance    101000

*/
```