BANKING SYSTEM

```
use bankingsystem;
describe customer;
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);
-- 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. --
select c.first_name,c.last_name, a.account_type from customer c join account a on c.id=a.customer_id ;
/*harry potter savings
harry potter current
ronald weasleycurrent
```

```
hermione grangerzero_balance
hermione grangersavings*/
select * from customer;
-- 2. Write a SQL query to list all transaction corresponding customer.--
select c.id,c.first_name,t.id,t.transaction_date,t.amount,t.transaction_type
from customer c join account a ON c.id = a.customer_id
join transaction t ON a.id = t.account_id;
/*1 harry 1 2024-02-01 10000 deposit
1 harry 2 2024-02-02 5000 withdrawal
1 harry 5 2024-02-01 20000 transfer
2 ronald 3 2024-02-02 20000 deposit
3 hermione 4 2024-02-02 8000 withdrawal
3 hermione 6 2024-02-05 7000 transfer*/
-- 3. Write a SQL query to increase the balance of a specific account by a certain amount.--
update account set balance=50000 where id=1;
-- updated
-- 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 balance=0 and account_type='savings';
-- 6. Write a SQL query to Find customers living in a specific city.--
select * from customer where city='chennai';
-- no city column
-- 7. Write a SQL query to Get the account balance for a specific account.--
select * from account where account_type='savings';
/*1 savings 50000 1
```

5 savings 30000 3

*/

-- 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';

/*2 current 120000 2

4 current 150000 1

*/

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

select t.* from transaction t join account a on a.id=t.account_id where a.id=3;

/*4 withdrawal 8000 2024-02-02 3*/

-- 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';

/*1 6000.0000

5 3600.0000*/

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

select id,account_type,balance from account where balance<80000;

/*1 savings 50000

5 savings 30000*/

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

select * from customer where city<>'pune' ;

-- no city column present

-- 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,c.first_name,avg(balance) from customer c join account a on c.id=a.customer_id group by c.id;

/* 1 harry 100000.0000

2 ronald 120000.0000

3 hermione 65000.0000*/

-- 2. Write a SQL query to Retrieve the top 3 highest account balances.--

```
select balance from account order by balance desc limit 3;
/*150000
120000
100000*/
-- 3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.--
select sum(amount)
from transaction
where transaction_type='deposit' and transaction_date='2024-02-01';
/*10000*/
-- 4. Write a SQL query to Find the Oldest and Newest Customers.--
select first_name,last_name, min(dob) as newest_customer,max(dob) as oldest_customer
from customer group by first_name,last_name;
/*harry potter 2002-03-21 2002-03-21
ronald weasley2001-02-10 2001-02-10
hermione granger 2002-11-15 2002-11-15*/
-- Write a SQL query to Retrieve transaction details along with the account type.--
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 transfer20000 2024-02-01 4 current
6 transfer7000 2024-02-05 5 savings*/
-- 6. Write a SQL query to Get a list of customers along with their account details.--
select * from customer c join account a on c.id=a.customer_id;
/*1 harry potter 2002-03-21 1 savings 50000 1
1 harry potter 2002-03-21 4 current 150000 1
2 ronald weasley2001-02-10 2 current 120000 2
3 hermione granger 2002-11-15 3 zero_balance 100000 3
```

3 hermione granger 2002-11-15 5 savings 30000 3*/

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

select t.*,c.* from customer c

join account a on c.id=a.customer_id

join transaction t on t.account_id=a.id where a.id=3;

/*4 withdrawal 8000 2024-02-02 3 3 hermione granger 2002-11-15*/

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

select c.* , count(c.id) as number_of_accounts from customer c join account a on c.id=a.customer_id

group by c.id having count(c.id)>1 ;

/*1 harry potter 2002-03-21 2

3 hermione granger 2002-11-15 2*/

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

select ( (select sum(amount) from transaction where transaction_type='deposit')-

(select sum(amount) from transaction where transaction_type='withdrawal')) as difference;

/*17000*/

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

select a.id, avg(balance)

from account a

join transaction t on a.id=t.account_id

where transaction_date

between '2024-02-01' and '2024-02-03'

group by a.id ;

/*1 50000.0000

2 120000.0000

3 100000.0000

4 150000.0000*/

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

```
select account_type ,sum(balance) as total_balance from account group by account_type;
/*savings 80000
current 270000
zero_balance 100000*/
-- 12. Identify accounts with the highest number of transactions order by descending order.--
select a.id,count(t.id) as num_transactions from account a join transaction t on a.id=t.account_id group
by a.id order by num_transactions desc;
/*1 2
2 1
3 1
4 1
5 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,a.account_type from
account a join customer c on a.customer_id=c.id group by c.first_name,a.account_type
having account_aggregate_balance>100000;
/*harry 150000 current
ronald 120000 current*/
-- 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 t group by transaction_type having
count(*)>1 )
and amount in (select amount from transaction t 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 100000.0000
ronald 120000.0000
hermione 65000.0000*/
-- 3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.
SELECT a.id,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 120000 3 20000
4 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);
 /* NO OUTPUT
*/
-- 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);
```

```
/* NO OUTPUT
*/
-- 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;
/*6 transfer30000 5*/
-- 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.
select account_type,count(*) as num_accounts,(count(*)* 100/(select count(*) from account)) as
percentage from account group by account_type;
/*savings 2 40.0000
current 2 40.0000
zero_balance 1 20.0000*/
-- 9. Retrieve all transactions for a customer with a given customer_id.
select concat(first_name ,' ',last_name) as full_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 transfer7000 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;
/*80000 savings
```

270000 current

100000 zero_balance*/