Banking System

Submitted by: Anirudh Suryawanshi

anirudhsuryawanshi759@gmail.com

```
-- -------------------------Task 1-----------------------------------
use bankingsystem;
show tables;
describe customer;

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(type,balance,customer_id) values
('savings',50000,1) ,
('current',120000,2) ,
('zero_balance',100000,3),
('current',150000,1) ,
('savings',30000,3);

insert into transaction(type,amount,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 customer;
select * from account;
select * from transaction;

-- --------------------------Task 2------------------------------------

-- Q1 Write a SQL query to retrieve the name, account type and email of all customers.
select c.first_name, a.type
from customer c JOIN account a ON c.id=a.customer_id;

/* Output
+------------+--------------+
| first_name | type         |
+------------+--------------+
| harry      | savings      |
| ronald     | current      |
| hermione   | zero_balance |
| harry      | current      |
| hermione   | savings      |
+------------+--------------+
*/

-- Q2 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;

/* Output
+----+------------+-----------+------------+----+------------+--------+------------+------------+
| id | first_name | last_name | dob        | id | type       | amount | 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 |
+----+------------+-----------+------------+----+------------+--------+------------+------------+
*/

-- Q3 Write a SQL query to increase the balance of a specific account by a certain amount.
update account
set balance=75000
where customer_id=1;

-- Q4 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
+----+------------+-----------+------------+------------------+
| id | first_name | last_name | dob        | full_name        |
+----+------------+-----------+------------+------------------+
|  1 | harry      | potter    | 2002-03-21 | harry potter     |
|  2 | ronald     | weasley   | 2001-02-10 | ronald weasley   |
|  3 | hermione   | granger   | 2002-11-15 | hermione granger |
+----+------------+-----------+------------+------------------+
*/
-- Q5 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 type='savings';

-- Q6 Write a SQL query to Find customers living in a specific city.
select *
from customer
where dob='2002-03-21';

/* Output
+----+------------+-----------+------------+
| id | first_name | last_name | dob        |
+----+------------+-----------+------------+
|  1 | harry      | potter    | 2002-03-21 |
+----+------------+-----------+------------+
*/

-- Q7 Write a SQL query to Get the account balance for a specific account.
select balance
from account
where id=1;

/* Output
+---------+
| balance |
+---------+
| 75000   |
+---------+
*/

-- Q8 Write a SQL query to List all current accounts with a balance greater than $1,000.
select *
```

```sql
from account
where type='current' AND balance>1000;

/*Output
+----+---------+---------+-------------+
| id | type    | balance | customer_id |
+----+---------+---------+-------------+
|  2 | current | 120000  |           2 |
|  4 | current | 75000   |           1 |
+----+---------+---------+-------------+
*/

-- Q9 Write a SQL query to Retrieve all transactions for a specific account.
select *
from transaction
where account_id=1;

/* Output
+----+------------+--------+------------+------------+
| id | type       | amount | date       | account_id |
+----+------------+--------+------------+------------+
|  1 | deposit    |  10000 | 2024-02-01 |          1 |
|  2 | withdrawal |   5000 | 2024-02-02 |          1 |
+----+------------+--------+------------+------------+
*/

-- Q10 Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.
select first_name, (2*3*5)
from customer c JOIN account a ON c.id=a.customer_id
where type='savings';
```

```
/* Output

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

| first_name | (2*3*5) |

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

| harry      |      30 |

| hermione   |      30 |

+------------+---------+*/
```

-- Q11 Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

```
select *

from account

where balance<100000;
```

```
/*Output

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

| id | type    | balance | customer_id |

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

|  1 | savings | 75000   |           1 |

|  4 | current | 75000   |           1 |

|  5 | savings | 30000   |           3 |

+----+---------+---------+-------------+*/
```

-- Q12 Write a SQL query to Find customers not living in a specific city.

```
select *

from customer

where dob != '2002-03-21';
```

```
/*Output

+----+------------+-----------+------------+
| id | first_name | last_name | dob        |
+----+------------+-----------+------------+
|  2 | ronald     | weasley   | 2001-02-10 |
|  3 | hermione   | granger   | 2002-11-15 |
+----+------------+-----------+------------+*/

-- -------------------------Task 3---------------------------------------------

-- Q1 Write a SQL query to Find the average account balance for all customers.
select c.first_name,c.last_name, avg(a.balance) as average
from customer c JOIN account a ON c.id=a.customer_id
group by c.id;

/*
+------------+-----------+---------+
| first_name | last_name | average |
+------------+-----------+---------+
| harry      | potter    |   75000 |
| ronald     | weasley   |  120000 |
| hermione   | granger   |   65000 |
+------------+-----------+---------+*/

-- Q2 Write a SQL query to Retrieve the top 10 highest account balances
select balance
from account
order by balance
limit 2;

/*
+---------+
| balance |
+---------+
| 100000  |
| 120000  |
+---------+*/

-- Q3 Write a SQL query to Calculate Total Deposits for All Customers in specific date.
select type, sum(amount)
from transaction
where type='deposit' AND date='2024-02-02'
group by type;

/*
+---------+-------------+
| type    | sum(amount) |
+---------+-------------+
| deposit |       20000 |
+---------+-------------+*/

-- Q4 Write a SQL query to Find the Oldest and Newest Customers
(select first_name,dob from customer order by dob ASC limit 1)
UNION
(select first_name,dob from customer order by dob DESC limit 1);

/*
+------------+------------+
| first_name | dob        |
+------------+------------+
| ronald     | 2001-02-10 |
| hermione   | 2002-11-15 |
+------------+------------+*/
```

```
-- Q5 Write a SQL query to Retrieve transaction details along with the account type.
select a.type,t.*
from account a JOIN transaction t on a.id=t.account_id;
/*
+--------------+----+------------+--------+------------+------------+
| type         | id | type       | amount | date       | account_id |
+--------------+----+------------+--------+------------+------------+
| savings      |  1 | deposit    |  10000 | 2024-02-01 |          1 |
| savings      |  2 | withdrawal |   5000 | 2024-02-02 |          1 |
| current      |  3 | deposit    |  20000 | 2024-02-02 |          2 |
| zero_balance |  4 | withdrawal |   8000 | 2024-02-02 |          3 |
| current      |  5 | transfer   |  20000 | 2024-02-01 |          4 |
| savings      |  6 | transfer   |   7000 | 2024-02-05 |          5 |
+--------------+----+------------+--------+------------+------------+*/
```

```
-- Q6 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;
/*
+----+------------+-----------+------------+----+--------------+---------+-------------+
| id | first_name | last_name | dob        | id | type         | balance | customer_id |
+----+------------+-----------+------------+----+--------------+---------+-------------+
|  1 | harry      | potter    | 2002-03-21 |  1 | savings      | 75000   |           1 |
|  2 | ronald     | weasley   | 2001-02-10 |  2 | current      | 120000  |           2 |
|  3 | hermione   | granger   | 2002-11-15 |  3 | zero_balance | 100000  |           3 |
|  1 | harry      | potter    | 2002-03-21 |  4 | current      | 75000   |           1 |
|  3 | hermione   | granger   | 2002-11-15 |  5 | savings      | 30000   |           3 |
+----+------------+-----------+------------+----+--------------+---------+-------------+*/
```

```
-- Q7 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
where t.account_id=2;
/*
+----+------------+-----------+------------+----+---------+--------+------------+------------+
| id | first_name | last_name | dob        | id | type    | amount | date       | account_id |
+----+------------+-----------+------------+----+---------+--------+------------+------------+
|  2 | ronald     | weasley   | 2001-02-10 |  3 | deposit |  20000 | 2024-02-02 |          2 |
+----+------------+-----------+------------+----+---------+--------+------------+------------+*/

-- Q8 Write a SQL query to Identify customers who have more than one account.
select *
from customer c JOIN account a ON c.id=a.customer_id
group by c.id
having count(c.id)>1;
/*
+----+------------+-----------+------------+----+--------------+---------+-------------+
| id | first_name | last_name | dob        | id | type         | balance | customer_id |
+----+------------+-----------+------------+----+--------------+---------+-------------+
|  1 | harry      | potter    | 2002-03-21 |  1 | savings      | 75000   |           1 |
|  3 | hermione   | granger   | 2002-11-15 |  3 | zero_balance | 100000  |           3 |
+----+------------+-----------+------------+----+--------------+---------+-------------+*/

-- Q9 Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.
(select sum(amount)
from transaction
where type='deposit')
UNION
```

```
(select sum(amount)
from transaction
where type='withdrawal') ;
/*
+-------------+
| sum(amount) |
+-------------+
|       30000 |
|       13000 |
+-------------+*/

-- Q10 Write a SQL query to Calculate the average daily balance for each account over a specified period.
select a.type,avg(a.balance)
from account a JOIN transaction t ON a.id=t.account_id
where t.date BETWEEN '2024-02-02' AND '2024-02-05'
group by a.id;
/*
+--------------+----------------+
| type         | avg(a.balance) |
+--------------+----------------+
| savings      |          75000 |
| current      |         120000 |
| zero_balance |         100000 |
| savings      |          30000 |
+--------------+----------------+*/

-- Q11 Calculate the total balance for each account type.
select type, sum(balance) as total_balance
from account
group by type;
```

```
/*
+--------------+---------------+
| type         | total_balance |
+--------------+---------------+
| current      |        195000 |
| savings      |        105000 |
| zero_balance |        100000 |
+--------------+---------------+*/

-- Q12 Identify accounts with the highest number of transactions order by descending order.
select type, count(balance) as number_of_transactions
from account
group by type
order by number_of_transactions DESC;
/*
+--------------+------------------------+
| type         | number_of_transactions |
+--------------+------------------------+
| current      |                      2 |
| savings      |                      2 |
| zero_balance |                      1 |
+--------------+------------------------+*/

-- Q13 List customers with high aggregate account balances, along with their account types.
select c.first_name, a.type, sum(a.balance) as total_balance
from customer c JOIN account a ON c.id=a.customer_id
group by c.id
order by total_balance DESC;
/*
+------------+--------------+---------------+
| first_name | type         | total_balance |
+------------+--------------+---------------+
| harry      | savings      |        150000 |
| hermione   | zero_balance |        130000 |
| ronald     | current      |        120000 |
+------------+--------------+---------------+*/

-- --------------------------Task 4-----------------------------------

-- 1. Retrieve the customer(s) with the highest account balance.
select c.*,a.balance
from customer c JOIN account a ON c.id=a.customer_id
order by a.balance
limit 1;
/*
+----+------------+-----------+------------+---------+
| id | first_name | last_name | dob        | balance |
+----+------------+-----------+------------+---------+
|  3 | hermione   | granger   | 2002-11-15 | 100000  |
+----+------------+-----------+------------+---------+*/

-- 2. Calculate the average account balance for customers who have more than one account.
select c.*,avg(a.balance) as average_balance
from customer c JOIN account a ON c.id=a.customer_id
group by c.id
having count(a.id)>1;
/*
+----+------------+-----------+------------+-----------------+
| id | first_name | last_name | dob        | average_balance |
+----+------------+-----------+------------+-----------------+
|  1 | harry      | potter    | 2002-03-21 |           75000 |
|  2 | ronald     | weasley   | 2001-02-10 |           72500 |
|  3 | hermione   | granger   | 2002-11-15 |          65000 |
+----+------------+-----------+------------+-----------------+*/

-- 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
group by t.account_id
having avg(t.amount)>(select avg(amount) from transaction);
/*
+----+---------+---------+-------------+
| id | type    | balance | customer_id |
+----+---------+---------+-------------+
|  2 | current | 120000  |           2 |
|  4 | current | 75000   |           1 |
+----+---------+---------+-------------+*/

-- 4. Identify customers who have no recorded transactions.
select *
from customer
where id NOT IN(select c.id
                                        from customer c
                                        JOIN account a ON c.id=a.customer_id
                                        JOIN transaction t ON a.id=t.account_id);
/*
+----+------------+-----------+------------+
| id | first_name | last_name | dob        |
+----+------------+-----------+------------+
|  4 | tom        | holland   | 2003-05-16 |
+----+------------+-----------+------------+*/

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

```
select *
from account
where id NOT IN(select a.id
                                        from account a JOIN transaction t ON a.id=t.account_id);
/*
+----+--------------+---------+-------------+
| id | type         | balance | customer_id |
+----+--------------+---------+-------------+
|  6 | zero_balance | 25000   |           2 |
+----+--------------+---------+-------------+*/

-- 6. Retrieve transactions for accounts with the lowest balance.
select t.*,a.balance
from account a JOIN transaction t ON a.id=t.account_id
order by a.balance DESC
limit 1;
/*
+----+---------+--------+------------+------------+---------+
| id | type    | amount | date       | account_id | balance |
+----+---------+--------+------------+------------+---------+
|  1 | deposit |  10000 | 2024-02-01 |          1 | 75000   |
+----+---------+--------+------------+------------+---------+*/

-- 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.id)>1;
/*
+----+------------+-----------+------------+
| id | first_name | last_name | dob        |
+----+------------+-----------+------------+
|  1 | harry      | potter    | 2002-03-21 |
|  2 | ronald     | weasley   | 2001-02-10 |
|  3 | hermione   | granger   | 2002-11-15 |
+----+------------+-----------+------------+*/

-- 8. Calculate the percentage of each account type out of the total number of accounts.
select *, count(type)/(select count(id) from account)*100 as percentage
from account
group by type;
/*
+----+--------------+---------+-------------+------------+
| id | type         | balance | customer_id | percentage |
+----+--------------+---------+-------------+------------+
|  2 | current      | 120000  |           2 |    33.3333 |
|  1 | savings      | 75000   |           1 |    33.3333 |
|  3 | zero_balance | 100000  |           3 |    33.3333 |
+----+--------------+---------+-------------+------------+*/

-- 9. Retrieve all transactions for a customer with a given customer_id.
select *
from customer c
JOIN account a ON c.id=a.customer_id
JOIN transaction t ON a.id=t.account_id
where c.id=1;
/*
+----+------------+-----------+------------+----+---------+---------+-------------+----+------------+--------+------------+--
----------+
| id | first_name | last_name | dob        | id | type    | balance | customer_id | id | type       | amount
| date       | account_id |
+----+------------+-----------+------------+----+---------+---------+-------------+----+------------+--------+------------+--
----------+
|  1 | harry      | potter    | 2002-03-21 |  1 | savings | 75000   |          1 |  1 | deposit    |  10000 | 2024-02-01 |          1 |
|  1 | harry      | potter    | 2002-03-21 |  1 | savings | 75000   |          1 |  2 | withdrawal |   5000 | 2024-02-02 |          1 |
|  1 | harry      | potter    | 2002-03-21 |  4 | current | 75000   |          1 |  5 | transfer   |  20000 | 2024-02-01 |          4 |
+----+------------+-----------+------------+----+---------+---------+-------------+----+------------+--------+------------+------------+*/


-- 10. Calculate the total balance for each account type, including a subquery within the SELECT clause.
select type, sum(balance) as total_balance
from account
group by type;
/*
+--------------+---------------+
| type         | total_balance |
+--------------+---------------+
| current      |        195000 |
| savings      |        105000 |
| zero_balance |        125000 |
+--------------+---------------+*/
```