ASSIGNMENT 2

BANKINGSYSTEM

INSERTIONS :

```
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;
```

QUERIES:

```
/*TASK 2*/


/* 1. query to retrieve the name,account type and email of all customers*/
select distinct c.first_name,c.last_name,a.account_type
from customer c left join account a on c.id = a.customer_id;
```

OUTPUT:

first_name	last_name	account_type
harry	potter	savings
harry	potter	current
ronald	weasley	current
hermione	granger	zero_balance
hermione	granger	savings

```
/*2.query to list all transaction corresponding customer*/
select c.first_name,c.last_name,t.transaction_type
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
harry	potter	deposit
harry	potter	withdrawal
harry	potter	transfer
ronald	weasley	deposit
hermione	granger	withdrawal
hermione	granger	transfer

```
/*3.query to increase the balance of a specific account by a certain amount */
update account
```

```
set balance = '130000'

where id = 2;
```

```
/*4.query to combine firt and last name of customers as full name */

select id, concat(first_name,last_name) as full_name

from customer;
```

OUTPUT:

```
id   full_name
 1  harrypotter
 2  ronaldweasley
 3  hermionegranger
```

```
/*5.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.query to find customer living in a specific city.*/

select *

from customer

where city = 'mumbai';
```

```
/*7.query to get account balance for a specific account*/

select *

from account

where account_type = 'savings';
```

OUTPUT:

```
id account_type balance customer_id
1  savings      50000   1
5  savings      30000   3
```

```
/*8.query to list all current accounts with a balance greater than 1000 */

select *

from account

where account_type = 'current' and balance > 1000;
```

OUTPUT:

```
id account_type balance customer_id
2  current      130000  2
4  current      150000  1
```

```
/*9.query to retrieve all transactions for a specific account */

select t.*,a.balance

from account a join transaction t on a.id = t.account_id

where a.account_type = 'savings';
```

OUTPUT:

```
id transaction_type amount transaction_date account_id balance
1  deposit          10000  2024-02-01       1          50000
2  withdrawal       5000   2024-02-02       1          50000
6  transfer         7000   2024-02-05       5          30000
```

```
/*10.query to calculate the inetrest accrued on savings accounts based on a
given interest rate*/

select *,(balance * 1.2/100) as interest_accrued

from account

where account_type = 'savings';
```

OUTPUT:

```
id account_type balance customer_id interest_accrued
1  savings      50000   1           600
5  savings      30000   3           360
```

```
/*11.query to identify accounts where the balance is less than the specified
overdraft limit */

select *

from account

where balance < 100000;
```

OUTPUT:

```
id account_type balance customer_id
1  savings      50000   1
5  savings      30000   3
```

```
/*12.query to find customers not living in a specific city */

select *

from customer

where city not like 'mumbai';
```

```
/*TASK 3*/
```

```
/*1.query to find average account balance for all customers */

select c.*,avg(a.balance) as average_balance

from customer c join account a on c.id = a.customer_id

group by c.first_name;
```

OUTPUT:

id	first_name	last_name	dob	average_balance
1	harry	potter	2002-03-21	100000
3	hermione	granger	2002-11-15	65000
2	ronald	weasley	2001-02-10	130000

```
/*2.query to retrieve top 10 highest account balances */
select c.first_name,c.last_name,a.balance
from customer c join account a on c.id = a.customer_id
group by a.balance
order by a.balance desc
limit 2;
```

OUTPUT:

first_name	last_name	balance
harry	potter	150000
ronald	weasley	130000

```
/*3.query to calculate total deposits for all customers in specific date */
select c.*,t.*
from customer c left join account a on c.id = a.customer_id
                join transaction t on a.id = t.account_id
where t.transaction_type = 'deposit' and transaction_date = '2024-02-02';
```

OUTPUT:

id	first_name	last_name	dob	id	transaction_type	amount	transaction_date	account_id
2	ronald	weasley	2001-02-10	3	deposit	20000	2024-02-02	2

```
 /*4.query to Find the Oldest and Newest Customers.*/

select *

from customer

where dob = (select min(dob) from customer);
```

OUTPUT:

```
id first_name last_name      dob
 2 ronald      weasley    2001-02-10
```

```
select *

from customer

where dob = (select max(dob) from customer);
```

OUTPUT:

```
id first_name last_name      dob
 3 hermione    granger    2002-11-15
```

```
/*5.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.query to Get a list of customers along with their account details.*/

select distinct c.*,a.*

from customer c join account a on c.id = a.customer_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
1	harry	potter	2002-03-21	4	current	150000	1
3	hermione	granger	2002-11-15	5	savings	30000	3

```
/*7.query to Retrieve transaction details along with customer information for a specific account.*/

select t.*,c.*,a.account_type

from transaction t join account a on a.id = t.account_id

                                             join customer c on c.id = a.customer_id

where account_type = 'savings';
```

OUTPUT:

id	transaction_type	amount	transaction_date	account_id	id	first_name	last_name	dob	account_type
1	deposit	10000	2024-02-01	1	1	harry	potter	2002-03-21	savings
2	withdrawal	5000	2024-02-02	1	1	harry	potter	2002-03-21	savings
6	transfer	7000	2024-02-05	5	3	hermione	granger	2002-11-15	savings

```
/*8.query to Identify customers who have more than one account.*/

select c.*, count(a.id)

from customer c join account a on c.id = a.customer_id

group by c.id

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

OUTPUT:

```
id first_name last_name    dob      count(a.id)
1  harry      potter     2002-03-21 2
3  hermione   granger    2002-11-15 2
```

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

projection : balance account

criteria: account type

*/

select sum(balance),account_type

from account

group by account_type;
```

OUTPUT:

```
sum(balance) account_type
280000       current
80000        savings
100000       zero_balance
```

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

projection : accounts

criteria: transactions
```

```
*/

select a.account_type , count(t.id)

from account a join transaction t on a.id = t.account_id

group by a.account_type

order by count(t.id) desc

limit 1;
```

OUTPUT:

account_type	count(t.id)
savings	3

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

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;
```

OUTPUT:

id	first_name	last_name	dob	id	account_type	balance	customer_id	sum(a.balance)
1	harry	potter	2002-03-21	1	savings	50000	1	200000
3	hermione	granger	2002-11-15	3	zero_balance	100000	3	130000
2	ronald	weasley	2001-02-10	2	current	130000	2	130000

```
-- TASK 4--


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

select c.first_name,c.last_name,a.account_type,a.balance
```

from customer c join account a on c.id = a.customer_id

where a.balance in (select max(balance) from account;

OUTPUT:

first_name	last_name	account_type	balance
harry	potter	current	150000

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

-- projection: account

-- criteria : customer

select *,avg(a.balance),count(*)

from customer c join account a on c.id = a.customer_id

group by c.id

having count(*) > 1;

OUTPUT:

id	first_name	last_name	dob	id	account_type	balance	customer_id	avg(a.balance)	count(*)
1	harry	potter	2002-03-21	1	savings	50000	1	100000	2
3	hermione	granger	2002-11-15	3	zero_balance	100000	3	65000	2

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

select *

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	id	transaction_type	amount	transaction_date	account_id
2	current	130000	2	3	deposit	20000	2024-02-02	2
4	current	150000	1	5	transfer	20000	2024-02-01	4

```
-- 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);
```

OUTPUT:

id	first_name	last_name	dob

```
-- 5. Calculate the total balance of accounts with no recorded transactions.
select sum(balance)
from account
where id not in (select a.id from account a join transaction t on a.id = t.account_id);
```

OUTPUT:

sum(balance)

```
-- 6. Retrieve transactions for accounts with the lowest balance.
select t.*,a.account_type,a.balance
```

```
from transaction t join account a on a.id = t.account_id

where a.balance in (select min(balance) from account);
```

OUTPUT:

id	transaction_type	amount	transaction_date	account_id	account_type	balance
6	transfer	7000	2024-02-05	5	savings	30000

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

select *,count(distinct account_type)

from account

group by customer_id

having count(distinct account_type) > 1;
```

OUTPUT:

id	account_type	balance	customer_id	count(distinct account_type)
1	savings	50000	1	2
3	zero_balance	100000	3	2

```
-- 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;
```

OUTPUT:

account_type	percentage
current	40.0000
savings	40.0000
zero_balance	20.0000

```sql
-- 9. Retrieve all transactions for a customer with a given customer_id.

select *

from transaction

where id in ( select id from account where customer_id = 1);
```

OUTPUT:

```
id transaction_type amount transaction_date account_id
1  deposit          10000  2024-02-01       1
4  withdrawal       8000   2024-02-02       3
```

```sql
-- 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;
```

OUTPUT:

```
account_type sum(balance)
current      280000
savings      80000
zero_balance 100000
```