HEXAWARE ASSIGNMENT – BANKING SYSTEM

Tasks 1: Database Design:

1. Create the database named "HMBank"

create database hmbank;

```
mysql> create database hmbank;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| hexa               |
| hexa2              |
| hmbank             |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
9 rows in set (0.00 sec)
```

2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

Customers:

create table customers

(customer_id int primary key,

first_name varchar(20),

last_name varchar(20),

dob date,

email varchar(30),

phone_number varchar(15),

address text);

```
mysql> desc customers;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| customer_id  | int         | NO   | PRI | NULL    |       |
| first_name   | varchar(20) | YES  |     | NULL    |       |
| last_name    | varchar(20) | YES  |     | NULL    |       |
| dob          | date        | YES  |     | NULL    |       |
| email        | varchar(30) | YES  |     | NULL    |       |
| phone_number | varchar(15) | YES  |     | NULL    |       |
| address      | text        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Accounts:

create table accounts

(account_id int primary key,

customer_id int,

account_type enum ('savings','current','zero_balance'),

balance decimal (15,2) default 0.00,

foreign key(customer_id) references customers(customer_id));

```
mysql> desc accounts;
+--------------+-----------------------------------------+------+-----+---------+-------+
| Field        | Type                                    | Null | Key | Default | Extra |
+--------------+-----------------------------------------+------+-----+---------+-------+
| account_id   | int                                     | NO   | PRI | NULL    |       |
| customer_id  | int                                     | YES  | MUL | NULL    |       |
| account_type | enum('savings','current','zero_balance') | YES  |     | NULL    |       |
| balance      | decimal(15,2)                           | YES  |     | 0.00    |       |
+--------------+-----------------------------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Transactions:

create table transactions

(transaction_id int primary key auto_increment,

account_id int,

transaction_type enum ('deposit','withdrawal','transfer'),

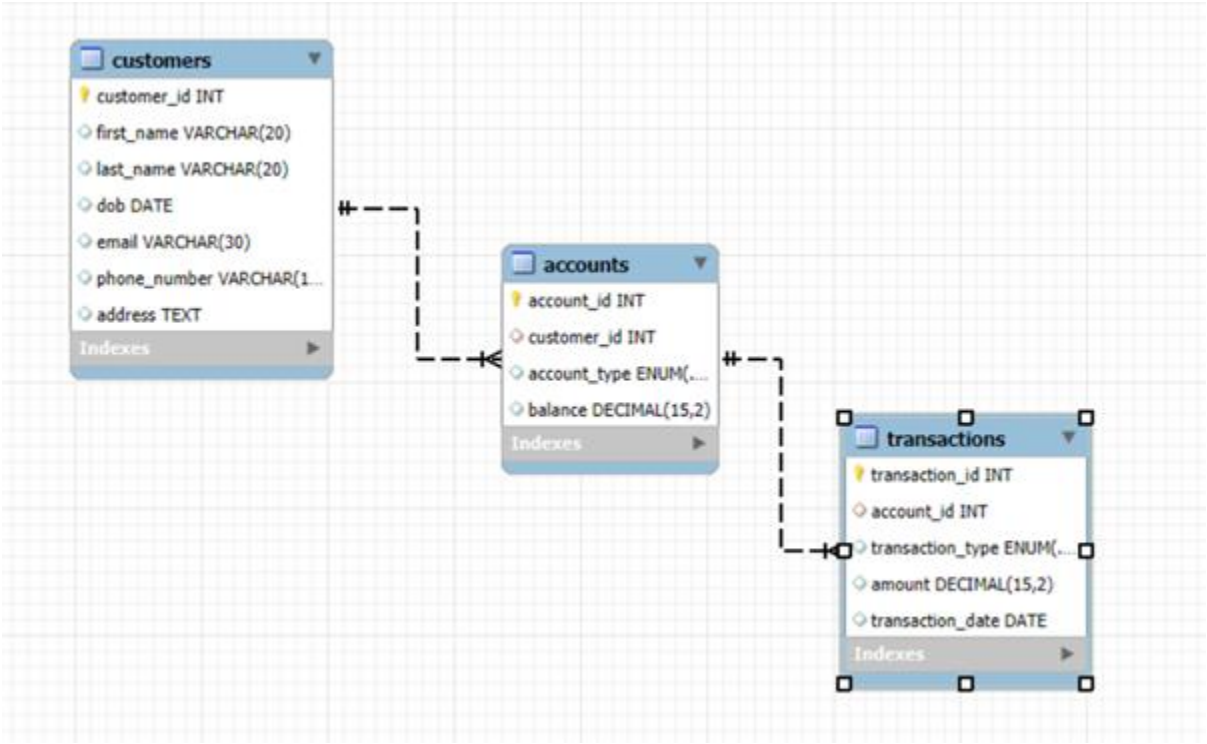
amount decimal (15,2) default 0.00,

transaction_date timestamp default current_timestamp,

foreign key(account_id) references accounts(account_id));

```
mysql> desc transactions;
+------------------+---------------------------------------------+------+-----+-------------------+-------------------+
| Field            | Type                                        | Null | Key | Default           | Extra             |
+------------------+---------------------------------------------+------+-----+-------------------+-------------------+
| transaction_id   | int                                         | NO   | PRI | NULL              | auto_increment    |
| account_id       | int                                         | YES  | MUL | NULL              |                   |
| transaction_type | enum('deposit','withdrawal','transfer')     | YES  |     | NULL              |                   |
| amount           | decimal(15,2)                               | YES  |     | 0.00              |                   |
| transaction_date | datetime                                    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+------------------+---------------------------------------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```

3.Create an ERD (Entity Relationship Diagram) for the database.



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

All primary keys and foreign keys were properly defined in the SQL provided.

5. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

- **Customers**
- **Accounts**
- **Transactions**

Tables created already following the constraints.

Tasks 2: Select, Where, Between, AND, LIKE:

1.Insert at least 10 sample records into each of the following tables.

Customers :

insert into customers values

(1,'Rajesh', 'Sharma', '1985-05-12', 'rajesh.sharma@example.com', '9876543210', 'Chennai, Tamil Nadu'),

(2,'Priya', 'Kumar', '1990-08-22', 'priya.kumar@example.com', '9867543211', 'Mumbai, Maharashtra'),

(3,'Anita', 'Verma', '1978-03-15', 'anita.verma@example.com', '9871234567', 'Delhi, Delhi'),

(4,'Vikram', 'Singh', '1989-12-05', 'vikram.singh@example.com', '9876549870', 'Jaipur, Rajasthan'),

(5,'Neha', 'Patel', '1993-06-30', 'neha.patel@example.com', '9812345678', 'Ahmedabad, Gujarat'),

(6,'Ravi', 'Nair', '1982-04-18', 'ravi.nair@example.com', '9823456789', 'Kochi, Kerala'),

(7,'Pooja', 'Mehta', '1995-11-09', 'pooja.mehta@example.com', '9834567890', 'Pune, Maharashtra'),

(8,'Suresh', 'Yadav', '1988-09-19', 'suresh.yadav@example.com', '9845678901', 'Lucknow, Uttar Pradesh'),

(9,'Lakshmi', 'Reddy', '1991-01-25', 'lakshmi.reddy@example.com', '9856789012', 'Hyderabad, Telangana'),

(10,'Manoj', 'Gupta', '1975-02-10', 'manoj.gupta@example.com', '9867890123', 'Kolkata, West Bengal');

```
mysql> select * from customers;
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
| customer_id | first_name | last_name | dob        | email                    | phone_number | address                |
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
|           1 | Rajesh     | Sharma    | 1985-05-12 | rajesh.sharma@example.com | 9876543210   | Chennai, Tamil Nadu    |
|           2 | Priya      | Kumar     | 1990-08-22 | priya.kumar@example.com   | 9867543211   | Mumbai, Maharashtra    |
|           3 | Anita      | Verma     | 1978-03-15 | anita.verma@example.com   | 9871234567   | Delhi, Delhi           |
|           4 | Vikram     | Singh     | 1989-12-05 | vikram.singh@example.com  | 9876549870   | Jaipur, Rajasthan      |
|           5 | Neha       | Patel     | 1993-06-30 | neha.patel@example.com    | 9812345678   | Ahmedabad, Gujarat     |
|           6 | Ravi       | Nair      | 1982-04-18 | ravi.nair@example.com     | 9823456789   | Kochi, Kerala          |
|           7 | Pooja      | Mehta     | 1995-11-09 | pooja.mehta@example.com   | 9834567890   | Pune, Maharashtra      |
|           8 | Suresh     | Yadav     | 1988-09-19 | suresh.yadav@example.com  | 9845678901   | Lucknow, Uttar Pradesh |
|           9 | Lakshmi    | Reddy     | 1991-01-25 | lakshmi.reddy@example.com | 9856789012   | Hyderabad, Telangana   |
|          10 | Manoj      | Gupta     | 1975-02-10 | manoj.gupta@example.com   | 9867890123   | Kolkata, West Bengal   |
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
10 rows in set (0.00 sec)
```

Accounts:

insert into accounts values

(1, 1,'savings', 10000.50),

(2, 2,'current', 25000.00),

(3, 3,'zero_balance', 0.00),

(4, 4,'savings', 5000.75),

(5, 5,'current', 15000.20),

(6, 6,'savings', 2000.00),

(7, 7,'zero_balance', 0.00),

(8, 8,'current', 18000.35),

(9, 9,'savings', 12000.80),

(10, 10,'current', 30000.10);

```
mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      | 10000.50 |
|          2 |           2 | current      | 25000.00 |
|          3 |           3 | zero_balance |     0.00 |
|          4 |           4 | savings      |  5000.75 |
|          5 |           5 | current      | 15000.20 |
|          6 |           6 | savings      |  2000.00 |
|          7 |           7 | zero_balance |     0.00 |
|          8 |           8 | current      | 18000.35 |
|          9 |           9 | savings      | 12000.80 |
|         10 |          10 | current      | 30000.10 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

Transactions:

insert into transactions (account_id, transaction_type, amount) values

(1, 'deposit', 2000.00),

(2, 'withdrawal', 500.00),

(3, 'deposit', 10000.00),

(4, 'transfer', 1500.50),

(5, 'withdrawal', 750.00),

(6, 'deposit', 3000.75),

(7, 'withdrawal', 1200.00),

(8, 'transfer', 4500.50),

(9, 'deposit', 3500.00),

(10, 'withdrawal', 250.75);

```
mysql> select * from transactions;
+----------------+------------+------------------+----------+---------------------+
| transaction_id | account_id | transaction_type | amount   | transaction_date    |
+----------------+------------+------------------+----------+---------------------+
|              1 |          1 | deposit          |  2000.00 | 2025-03-21 08:22:44 |
|              2 |          2 | withdrawal       |   500.00 | 2025-03-21 08:22:44 |
|              3 |          3 | deposit          | 10000.00 | 2025-03-21 08:22:44 |
|              4 |          4 | transfer         |  1500.50 | 2025-03-21 08:22:44 |
|              5 |          5 | withdrawal       |   750.00 | 2025-03-21 08:22:44 |
|              6 |          6 | deposit          |  3000.75 | 2025-03-21 08:22:44 |
|              7 |          7 | withdrawal       |  1200.00 | 2025-03-21 08:22:44 |
|              8 |          8 | transfer         |  4500.50 | 2025-03-21 08:22:44 |
|              9 |          9 | deposit          |  3500.00 | 2025-03-21 08:22:44 |
|             10 |         10 | withdrawal       |   250.75 | 2025-03-21 08:22:44 |
+----------------+------------+------------------+----------+---------------------+
10 rows in set (0.00 sec)
```

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 concat(first_name, ' ', last_name) as customer_name,

accounts.account_type,

customers.email

from customers

join accounts on customers.customer_id = accounts.customer_id;

customer_name	account_type	email
Rajesh Sharma	savings	rajesh.sharma@example.com
Priya Kumar	current	priya.kumar@example.com
Anita Verma	zero_balance	anita.verma@example.com
Vikram Singh	savings	vikram.singh@example.com
Neha Patel	current	neha.patel@example.com
Ravi Nair	savings	ravi.nair@example.com
Pooja Mehta	zero_balance	pooja.mehta@example.com
Suresh Yadav	current	suresh.yadav@example.com
Lakshmi Reddy	savings	lakshmi.reddy@example.com
Manoj Gupta	current	manoj.gupta@example.com

10 rows in set (0.00 sec)

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

```
select concat(c.first_name, ' ', c.last_name) as customer_name,
c.email,
t.transaction_id,
t.transaction_type,
t.amount,
t.transaction_date
from customers c
join accounts a on c.customer_id = a.customer_id
join transactions t on a.account_id = t.account_id;
```

customer_name	email	transaction_id	transaction_type	amount	transaction_date
Rajesh Sharma	rajesh.sharma@example.com	1	deposit	2000.00	2025-03-21 08:22:44
Priya Kumar	priya.kumar@example.com	2	withdrawal	500.00	2025-03-21 08:22:44
Anita Verma	anita.verma@example.com	3	deposit	10000.00	2025-03-21 08:22:44
Vikram Singh	vikram.singh@example.com	4	transfer	1500.50	2025-03-21 08:22:44
Neha Patel	neha.patel@example.com	5	withdrawal	750.00	2025-03-21 08:22:44
Ravi Nair	ravi.nair@example.com	6	deposit	3000.75	2025-03-21 08:22:44
Pooja Mehta	pooja.mehta@example.com	7	withdrawal	1200.00	2025-03-21 08:22:44
Suresh Yadav	suresh.yadav@example.com	8	transfer	4500.50	2025-03-21 08:22:44
Lakshmi Reddy	lakshmi.reddy@example.com	9	deposit	3500.00	2025-03-21 08:22:44
Manoj Gupta	manoj.gupta@example.com	10	withdrawal	250.75	2025-03-21 08:22:44

10 rows in set (0.00 sec)

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

```
update accounts
set balance = balance + 1000.00
```

where account_id = 1;

```
mysql> update accounts
    -> set balance = balance + 1000.00
    -> where account_id = 1;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      | 11000.50 |
|          2 |           2 | current      | 25000.00 |
|          3 |           3 | zero_balance |     0.00 |
|          4 |           4 | savings      |  5000.75 |
|          5 |           5 | current      | 15000.20 |
|          6 |           6 | savings      |  2000.00 |
|          7 |           7 | zero_balance |     0.00 |
|          8 |           8 | current      | 18000.35 |
|          9 |           9 | savings      | 12000.80 |
|         10 |          10 | current      | 30000.10 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

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 customers;

```
+---------------+
| full_name     |
+---------------+
| Rajesh Sharma |
| Priya Kumar   |
| Anita Verma   |
| Vikram Singh  |
| Neha Patel    |
| Ravi Nair     |
| Pooja Mehta   |
| Suresh Yadav  |
| Lakshmi Reddy |
| Manoj Gupta   |
+---------------+
10 rows in set (0.00 sec)
```

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

```
delete from accounts

where balance = 0 and account_type = 'savings';
```

```
mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      | 11000.50 |
|          2 |           2 | current      | 25000.00 |
|          3 |           3 | zero_balance |     0.00 |
|          4 |           4 | savings      |  5000.75 |
|          5 |           5 | current      | 15000.20 |
|          6 |           6 | savings      |  2000.00 |
|          7 |           7 | zero_balance |     0.00 |
|          8 |           8 | current      | 18000.35 |
|          9 |           9 | savings      | 12000.80 |
|         10 |          10 | current      | 30000.10 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

6. Write a SQL query to Find customers living in a specific city.

```
select concat(first_name, ' ', last_name) as full_name,

email,

phone_number

from customers

where address like '%chennai%';
```

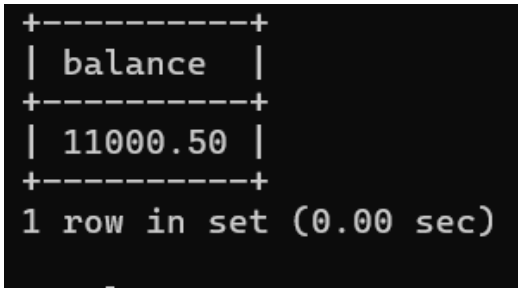
```
+---------------+---------------------------+--------------+
| full_name     | email                     | phone_number |
+---------------+---------------------------+--------------+
| Rajesh Sharma | rajesh.sharma@example.com | 9876543210   |
+---------------+---------------------------+--------------+
1 row in set (0.02 sec)
```

7. Write a SQL query to Get the account balance for a specific account.

```
select balance

from accounts

where account_id = 1;
```



balance
11000.50

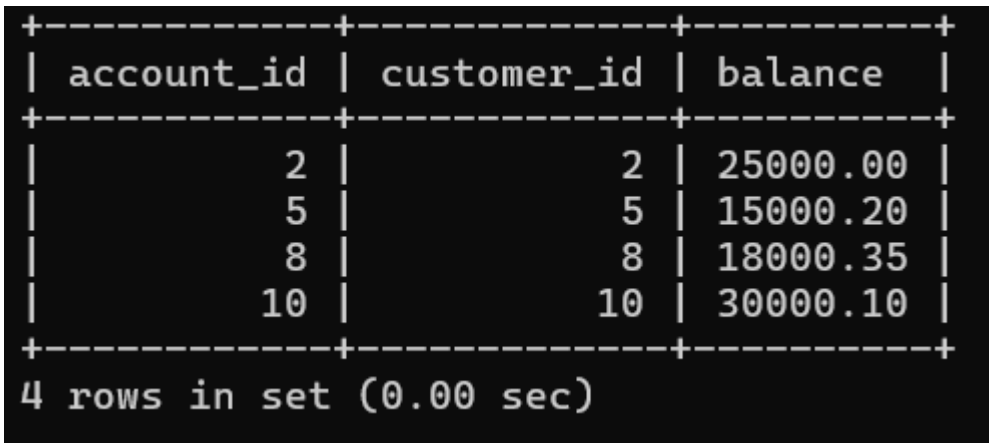
1 row in set (0.00 sec)

8. Write a SQL query to List all current accounts with a balance greater than $1,000.

select account_id, customer_id, balance

from accounts

where account_type = 'current' and balance > 1000;



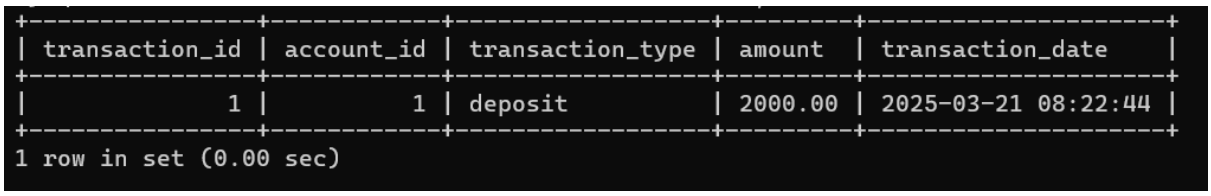
account_id	customer_id	balance
2	2	25000.00
5	5	15000.20
8	8	18000.35
10	10	30000.10

4 rows in set (0.00 sec)

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

select * from transactions

where account_id=1;



transaction_id	account_id	transaction_type	amount	transaction_date
1	1	deposit	2000.00	2025-03-21 08:22:44

1 row in set (0.00 sec)

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

Let me take rate of interest as 10.

select account_id, balance,

balance * 0.1 as interest_accrued

from accounts

where account_type = 'savings';

```
+------------+----------+------------------+
| account_id | balance  | interest_accrued |
+------------+----------+------------------+
|          1 | 11000.50 |         1100.050 |
|          4 |  5000.75 |          500.075 |
|          6 |  2000.00 |          200.000 |
|          9 | 12000.80 |         1200.080 |
+------------+----------+------------------+
4 rows in set (0.00 sec)
```

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

select account_id, customer_id, account_type, balance

from accounts

where balance < -1000;

```
mysql> select account_id, customer_id, account_type, balance
    -> from accounts
    -> where balance < -1000;
Empty set (0.00 sec)
```

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

select * from customers

where address not like '%chennai%';

```
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
| customer_id | first_name | last_name | dob        | email                    | phone_number | address                |
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
|           2 | Priya      | Kumar     | 1990-08-22 | priya.kumar@example.com  | 9867543211   | Mumbai, Maharashtra    |
|           3 | Anita      | Verma     | 1978-03-15 | anita.verma@example.com  | 9871234567   | Delhi, Delhi           |
|           4 | Vikram     | Singh     | 1989-12-05 | vikram.singh@example.com | 9876549870   | Jaipur, Rajasthan      |
|           5 | Neha       | Patel     | 1993-06-30 | neha.patel@example.com   | 9812345678   | Ahmedabad, Gujarat     |
|           6 | Ravi       | Nair      | 1982-04-18 | ravi.nair@example.com    | 9823456789   | Kochi, Kerala          |
|           7 | Pooja      | Mehta     | 1995-11-09 | pooja.mehta@example.com  | 9834567890   | Pune, Maharashtra      |
|           8 | Suresh     | Yadav     | 1988-09-19 | suresh.yadav@example.com | 9845678901   | Lucknow, Uttar Pradesh |
|           9 | Lakshmi    | Reddy     | 1991-01-25 | lakshmi.reddy@example.com| 9856789012   | Hyderabad, Telangana   |
|          10 | Manoj      | Gupta     | 1975-02-10 | manoj.gupta@example.com  | 9867890123   | Kolkata, West Bengal   |
+-------------+------------+-----------+------------+--------------------------+--------------+------------------------+
9 rows in set (0.00 sec)
```

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 avg(balance) as average_balance
from accounts;
```

```
+-----------------+
| average_balance |
+-----------------+
|    11800.270000 |
+-----------------+
1 row in set (0.00 sec)
```

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

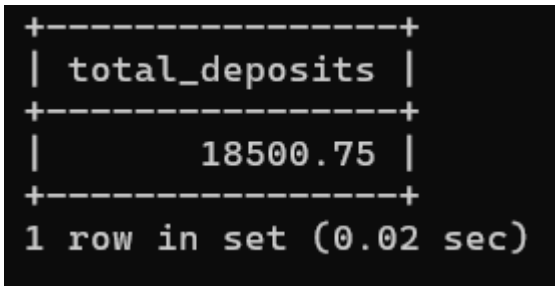
```
select account_id, customer_id, account_type, balance
from accounts
order by balance desc;
```

```
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|         10 |          10 | current      | 30000.10 |
|          2 |           2 | current      | 25000.00 |
|          8 |           8 | current      | 18000.35 |
|          5 |           5 | current      | 15000.20 |
|          9 |           9 | savings      | 12000.80 |
|          1 |           1 | savings      | 11000.50 |
|          4 |           4 | savings      |  5000.75 |
|          6 |           6 | savings      |  2000.00 |
|          3 |           3 | zero_balance |     0.00 |
|          7 |           7 | zero_balance |     0.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

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

```
select sum(amount) as total_deposits
from transactions
```

where transaction_type = 'deposit' and date(transaction_date) = '2025-03-21';

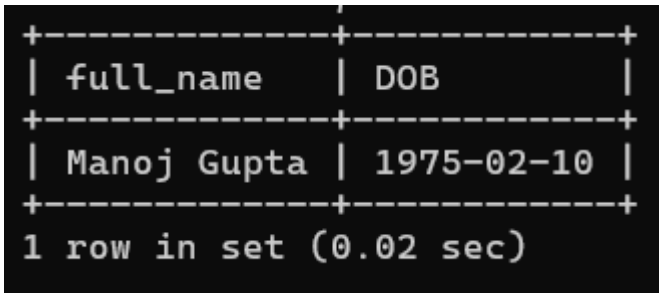


```
+----------------+
| total_deposits |
+----------------+
|       18500.75 |
+----------------+
1 row in set (0.02 sec)
```

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

BY AGE(oldest):

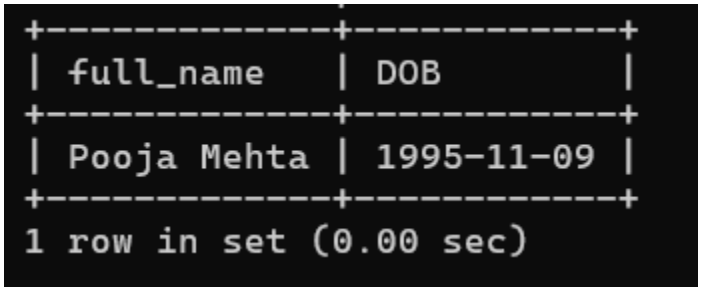
```
select concat(first_name, ' ', last_name) as full_name, dob
from customers
order by dob asc
limit 1;
```



```
+-------------+------------+
| full_name   | DOB        |
+-------------+------------+
| Manoj Gupta | 1975-02-10 |
+-------------+------------+
1 row in set (0.02 sec)
```

BY AGE(Newest):

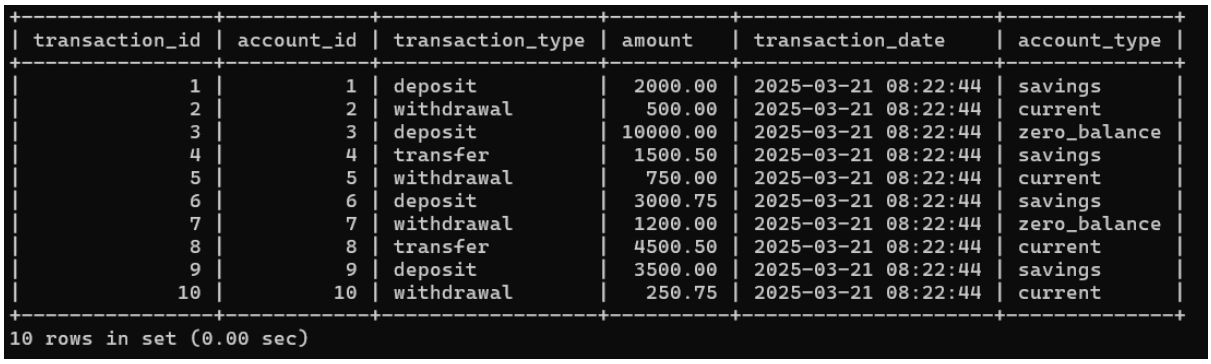
```
select concat(first_name, ' ', last_name) as full_name, dob
from customers
order by dob desc
limit 1;
```



```
+-------------+------------+
| full_name   | DOB        |
+-------------+------------+
| Pooja Mehta | 1995-11-09 |
+-------------+------------+
1 row in set (0.00 sec)
```

5. Write a SQL query to Retrieve transaction details along with the account type.

```
select t.transaction_id,
t.account_id,
t.transaction_type,
t.amount,
t.transaction_date,
a.account_type
from transactions t
join accounts a on t.account_id = a.account_id;
```



transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1	1	deposit	2000.00	2025-03-21 08:22:44	savings
2	2	withdrawal	500.00	2025-03-21 08:22:44	current
3	3	deposit	10000.00	2025-03-21 08:22:44	zero_balance
4	4	transfer	1500.50	2025-03-21 08:22:44	savings
5	5	withdrawal	750.00	2025-03-21 08:22:44	current
6	6	deposit	3000.75	2025-03-21 08:22:44	savings
7	7	withdrawal	1200.00	2025-03-21 08:22:44	zero_balance
8	8	transfer	4500.50	2025-03-21 08:22:44	current
9	9	deposit	3500.00	2025-03-21 08:22:44	savings
10	10	withdrawal	250.75	2025-03-21 08:22:44	current

10 rows in set (0.00 sec)

6. Write a SQL query to Get a list of customers along with their account details.

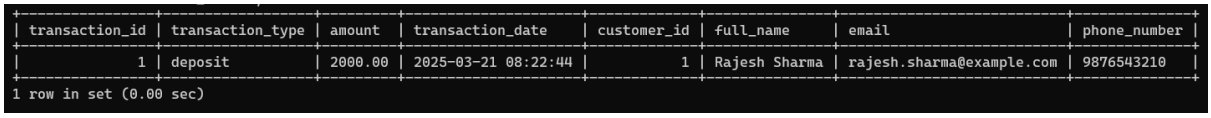
```
select *
from customers c
join accounts a on c.customer_id = a.customer_id;
```

customer_id	first_name	last_name	dob	email	phone_number	address	account_id	customer_id	account_type	balance
1	Rajesh	Sharma	1985-05-12	rajesh.sharma@example.com	9876543210	Chennai, Tamil Nadu	1	1	savings	11000.50
2	Priya	Kumar	1990-08-22	priya.kumar@example.com	9867543211	Mumbai, Maharashtra	2	2	current	25000.00
3	Anita	Verma	1978-03-15	anita.verma@example.com	9871234567	Delhi, Delhi	3	3	zero_balance	0.00
4	Vikram	Singh	1989-12-05	vikram.singh@example.com	9876549870	Jaipur, Rajasthan	4	4	savings	5000.75
5	Neha	Patel	1993-06-30	neha.patel@example.com	9812345678	Ahmedabad, Gujarat	5	5	current	15000.20
6	Ravi	Nair	1982-04-18	ravi.nair@example.com	9823456789	Kochi, Kerala	6	6	savings	2000.00
7	Pooja	Mehta	1995-11-09	pooja.mehta@example.com	98345678901	Pune, Maharashtra	7	7	zero_balance	0.00
8	Suresh	Yadav	1988-09-19	suresh.yadav@example.com	9845678901	Lucknow, Uttar Pradesh	8	8	current	18000.35
9	Lakshmi	Reddy	1991-01-25	lakshmi.reddy@example.com	9856789012	Hyderabad, Telangana	9	9	savings	12000.80
10	Manoj	Gupta	1975-02-10	manoj.gupta@example.com	9867890123	Kolkata, West Bengal	10	10	current	30000.10

10 rows in set (0.00 sec)

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

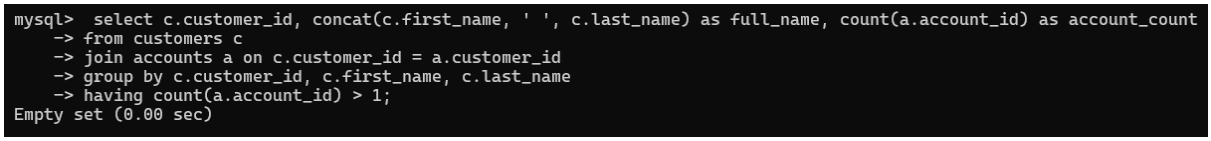
```
select t.transaction_id,t.transaction_type, t.amount, t.transaction_date,
c.customer_id, concat(c.first_name, ' ', c.last_name) as full_name, c.email,
c.phone_number
from transactions t
join accounts a on t.account_id = a.account_id
join customers c on a.customer_id = c.customer_id
where t.account_id = 1;
```



```
+----------------+------------------+---------+---------------------+-------------+--------------+-------------------------+--------------+
| transaction_id | transaction_type | amount  | transaction_date    | customer_id | full_name    | email                   | phone_number |
+----------------+------------------+---------+---------------------+-------------+--------------+-------------------------+--------------+
|              1 | deposit          | 2000.00 | 2025-03-21 08:22:44 |           1 | Rajesh Sharma | rajesh.sharma@example.com | 9876543210   |
+----------------+------------------+---------+---------------------+-------------+--------------+-------------------------+--------------+
1 row in set (0.00 sec)
```

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

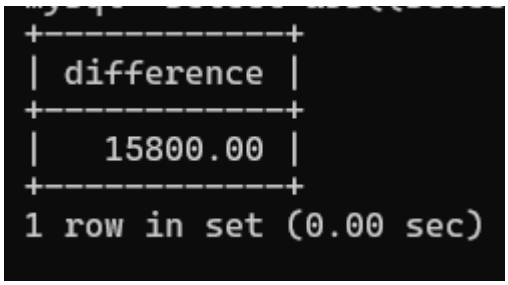
```
select c.customer_id, concat(c.first_name, ' ', c.last_name) as full_name,
count(a.account_id) as account_count
from customers c
join accounts a on c.customer_id = a.customer_id
group by c.customer_id, c.first_name, c.last_name
having count(a.account_id) > 1;
```



```
mysql>  select c.customer_id, concat(c.first_name, ' ', c.last_name) as full_name, count(a.account_id) as account_count
    -> from customers c
    -> join accounts a on c.customer_id = a.customer_id
    -> group by c.customer_id, c.first_name, c.last_name
    -> having count(a.account_id) > 1;
Empty set (0.00 sec)
```

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

```
select abs((select sum(amount) from transactions where
transaction_type='deposit')-(select sum(amount) from transactions where
transaction_type='withdrawal')) as difference;
```



```
+------------+
| difference |
+------------+
|   15800.00 |
+------------+
1 row in set (0.00 sec)
```

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

```
select account_id, avg(daily_balance) as average_daily_balance
from (select account_id, transaction_date, sum(amount) over (partition by
account_id order by transaction_date) as daily_balance
from transaction
where transaction_date between '2025-03-01' and '2025-03-31' ) as
daily_balances
group by account_id;
```

account_id	average_daily_balance
1	2000.000000
2	500.000000
3	10000.000000
4	1500.500000
5	750.000000
6	3000.750000
7	1200.000000
8	4500.500000
9	3500.000000
10	250.750000

10 rows in set (0.00 sec)

11. Calculate the total balance for each account type.

```
select sum(balance) as tot_balance,account_type
from accounts
group by account_type;
```

tot_balance	account_type
30002.05	savings
88000.65	current
0.00	zero_balance

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

```
select count(transaction_id) as transaction_count,account_id from transactions
group by account_id
order by transaction_count desc;
```

transaction_count	account_id
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10

10 rows in set (0.00 sec)

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

```
select c.first_name, c.last_name, a.account_type, sum(a.balance) as
total_balance
from customers c
join accounts a on c.customer_id = a.customer_id
group by c.customer_id, a.account_type
order by total_balance desc;
```

first_name	last_name	account_type	total_balance
Manoj	Gupta	current	30000.10
Priya	Kumar	current	25000.00
Suresh	Yadav	current	18000.35
Neha	Patel	current	15000.20
Lakshmi	Reddy	savings	12000.80
Rajesh	Sharma	savings	11000.50
Vikram	Singh	savings	5000.75
Ravi	Nair	savings	2000.00
Anita	Verma	zero_balance	0.00
Pooja	Mehta	zero_balance	0.00

10 rows in set (0.00 sec)

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

select account_id, amount, transaction_date, count(*) as duplicate_count
from transactions
group by account_id, amount, transaction_date
having count(*) > 1;

```
mysql> select account_id, amount, transaction_date, count(*) as duplicate_count
    -> from transactions
    -> group by account_id, amount, transaction_date
    -> having count(*) > 1;
Empty set (0.00 sec)
```

Tasks 4: Subquery and its type:

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

```
select customer_id,first_name,last_name from customers
where customer_id = (select customer_id from accounts
where balance = (select max(balance) from accounts));
```

```
+-------------+------------+-----------+
| customer_id | first_name | last_name |
+-------------+------------+-----------+
|          10 | Manoj      | Gupta     |
+-------------+------------+-----------+
1 row in set (0.00 sec)
```

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

```
select avg(balance) as average_balance
from accounts
where customer_id in (
select customer_id
from accounts
group by customer_id
having count(*) > 1
);
```

```
+-----------------+
| average_balance |
+-----------------+
|            NULL |
+-----------------+
1 row in set (0.00 sec)
```

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

```
select account_id, transaction_id, transaction_type, amount, transaction_date
from transactions
```

where amount > (select avg(amount) from transactions);

account_id	transaction_id	transaction_type	amount	transaction_date
3	3	deposit	10000.00	2025-03-21 08:22:44
6	6	deposit	3000.75	2025-03-21 08:22:44
8	8	transfer	4500.50	2025-03-21 08:22:44
9	9	deposit	3500.00	2025-03-21 08:22:44

4 rows in set (0.00 sec)

4. Identify customers who have no recorded transactions.

select customer_id,first_name,last_name from customers

where customer_id in(select customer_id from accounts where

account_id not in (select account_id from transactions));

```
mysql> select customer_id,first_name,last_name from customers
    -> where customer_id in(select customer_id from accounts where
    -> account_id not in (select account_id from transactions));
Empty set (0.00 sec)
```

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

select sum(balance) from accounts

where account_id not in(select account_id from transactions);

sum(balance)
NULL

1 row in set (0.00 sec)

6. Retrieve transactions for accounts with the lowest balance.

select * from transactions

where account_id in (select account_id from accounts

where balance=(select min(balance) from accounts));

transaction_id	account_id	transaction_type	amount	transaction_date
3	3	deposit	10000.00	2025-03-21 08:22:44
7	7	withdrawal	1200.00	2025-03-21 08:22:44

2 rows in set (0.00 sec)

7. Identify customers who have accounts of multiple types.

```
select c.customer_id, c.first_name, c.last_name
from customers c
join accounts a on c.customer_id = a.customer_id
group by c.customer_id, c.first_name, c.last_name
having count(distinct a.account_type) > 1;
```

```
mysql> select c.customer_id, c.first_name, c.last_name
    -> from customers c
    -> join accounts a on c.customer_id = a.customer_id
    -> group by c.customer_id, c.first_name, c.last_name
    -> having count(distinct a.account_type) > 1;
Empty set (0.00 sec)
```

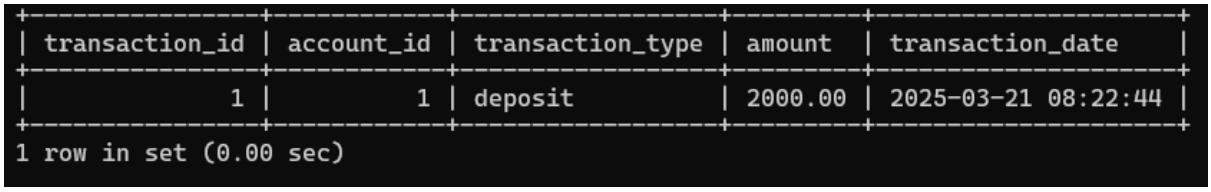
8. Calculate the percentage of each account type out of the total number of accounts.

```
select account_type,
count(*) as count,
round((count(*) * 100.0 / (select count(*) from accounts)), 2) as percentage
from accounts
group by account_type;
```

```
+--------------+-------+------------+
| account_type | count | percentage |
+--------------+-------+------------+
| savings      |     4 |      40.00 |
| current      |     4 |      40.00 |
| zero_balance |     2 |      20.00 |
+--------------+-------+------------+
3 rows in set (0.02 sec)
```

9. Retrieve all transactions for a customer with a given customer_id.

```
select * from transactions
where account_id in (select account_id from accounts
where customer_id=1);
```

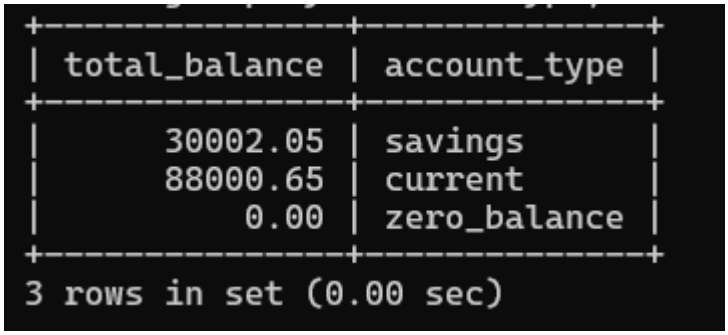


transaction_id	account_id	transaction_type	amount	transaction_date
1	1	deposit	2000.00	2025-03-21 08:22:44

1 row in set (0.00 sec)

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

```
select (select sum(balance) from accounts as a2
where a2.account_type = a1.account_type) as total_balance,account_type
from accounts as a1
group by account_type;
```



total_balance	account_type
30002.05	savings
88000.65	current
0.00	zero_balance

3 rows in set (0.00 sec)