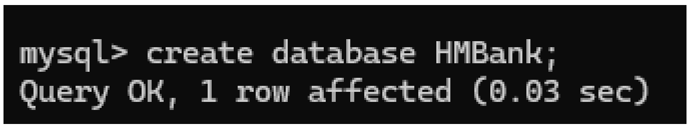
Mohammed Ibrahim Sheriff U
Batch_4

ASSIGNMENT 3 – SQL & OOPS BANKING SYSTEM

Tasks 1: Database Design:

1. Create the database named "HMBank"

create database HMBank;



```
mysql> create database HMBank;
Query OK, 1 row affected (0.03 sec)
```

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

Customers:

- customer_id (Primary Key)
- first_name
- last_name
- DOB (Date of Birth)
- email
- phone_number
- address

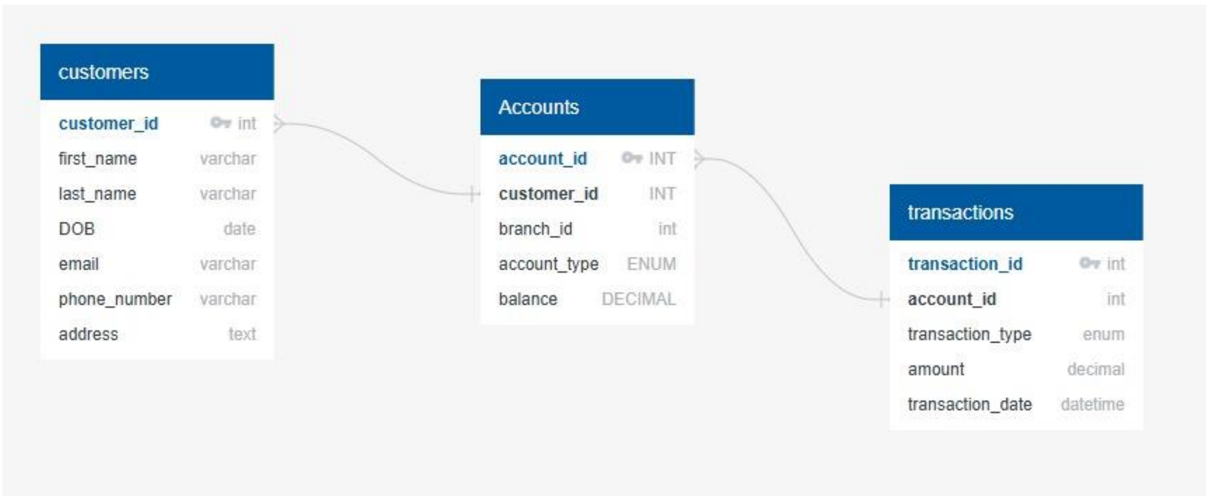
Accounts:

- account_id (Primary Key)
- customer_id (Foreign Key referencing Customers)
- account_type (e.g., savings, current, zero_balance)
- balance

Transactions:

- transaction_id (Primary Key)
- account_id (Foreign Key referencing Accounts)
- transaction_type (e.g., deposit, withdrawal, transfer)
- amount
- transaction_date

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



Entities and Attributes:

- **Customers**:
    - Attributes: customer_id (PK), first_name, last_name, DOB, email, phone_number, address
- **Accounts**:
    - Attributes: account_id (PK), customer_id (FK), account_type, balance
- **Transactions**:
    - Attributes: transaction_id (PK), account_id (FK), transaction_type, amount, transaction_date

Relationships:

- **Customers to Accounts**: One-to-Many
    - One customer can have multiple accounts.

    - customer_id in Accounts is a foreign key referencing customer_id in Customers.

- **Accounts to Transactions**: One-to-Many
    - One account can have multiple transactions.
    - account_id in Transactions is a foreign key referencing account_id in Accounts.

Customers: Identified by customer_id, contains customer details.

Accounts: Identified by account_id, linked to Customers via customer_id, specifies account type and balance.

Transactions: Identified by transaction_id, linked to Accounts via account_id, records transaction details.

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

Primary Keys:

- customer_id in Customers
- account_id in Accounts
- transaction_id in Transactions

Foreign Keys:

- customer_id in Accounts references customer_id in Customers.
- account_id in Transactions references account_id in Accounts.

Additional Constraints:

- email in Customers should be unique to prevent duplicate customer accounts.
- first_name and last_name in Customers are typically required (NOT NULL).
- account_type in Accounts should be restricted to valid types (e.g., 'savings', 'current', 'zero_balance').
- balance in Accounts should be non-negative (enforced via CHECK or application logic).
- transaction_type in Transactions should be restricted to valid types (e.g., 'deposit', 'withdrawal', 'transfer').

- amount in Transactions should be positive (enforced via CHECK or application logic).

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

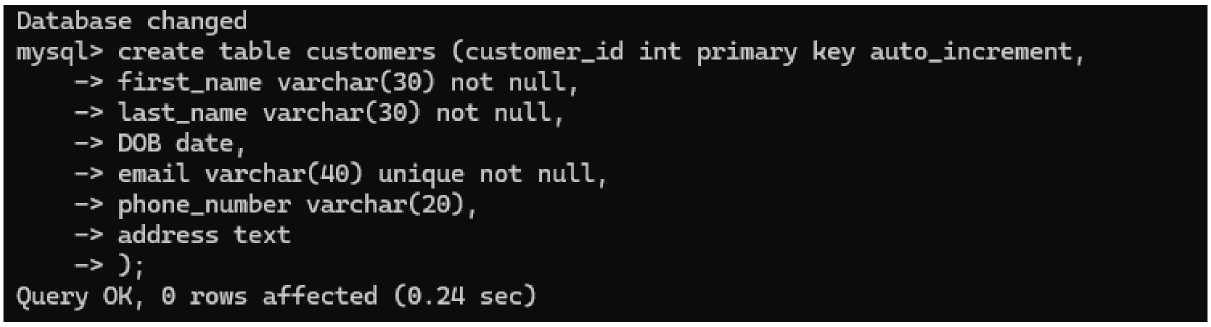
• Customers

• Accounts

• Transactions

• Customers :

```
create table customers (customer_id int primary key auto_increment,
    first_name varchar(30) not null,
    last_name varchar(30) not null,
     DOB date,
     email varchar(40) unique not null,
     phone_number varchar(20),
     address text
     );
```



```
Database changed
mysql> create table customers (customer_id int primary key auto_increment,
    -> first_name varchar(30) not null,
    -> last_name varchar(30) not null,
    -> DOB date,
    -> email varchar(40) unique not null,
    -> phone_number varchar(20),
    -> address text
    -> );
Query OK, 0 rows affected (0.24 sec)
```

• Accounts:

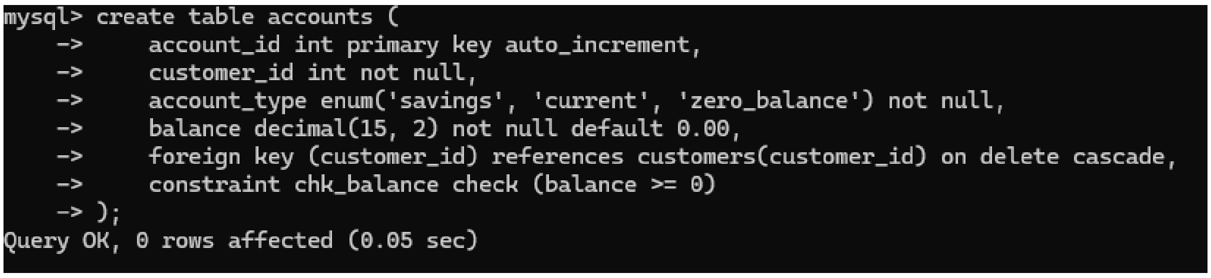
```
create table accounts (
    account_id int primary key auto_increment,
    customer_id int not null,
    account_type enum('savings', 'current', 'zero_balance') not null,
```

balance decimal(15, 2) not null default 0.00,

foreign key (customer_id) references customers(customer_id) on delete cascade,

constraint chk_balance check (balance >= 0)

);



```
mysql> create table accounts (
    ->     account_id int primary key auto_increment,
    ->     customer_id int not null,
    ->     account_type enum('savings', 'current', 'zero_balance') not null,
    ->     balance decimal(15, 2) not null default 0.00,
    ->     foreign key (customer_id) references customers(customer_id) on delete cascade,
    ->     constraint chk_balance check (balance >= 0)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

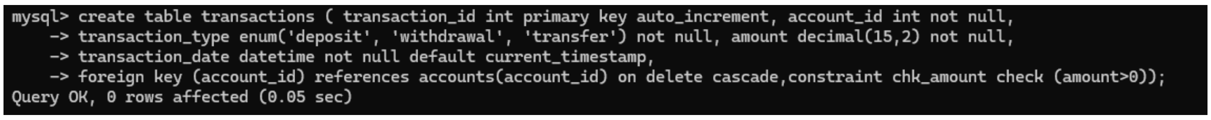
• Transactions

create table transactions (transaction_id int primary key auto_increment, account_id int not null,

-> transaction_type enum('deposit', 'withdrawal', 'transfer') not null, amount decimal(15,2) not null,

-> transaction_date datetime not null default current_timestamp,

-> foreign key (account_id) references accounts(account_id) on delete cascade,constraint chk_amount check (amount>0));



```
mysql> create table transactions ( transaction_id int primary key auto_increment, account_id int not null,
    -> transaction_type enum('deposit', 'withdrawal', 'transfer') not null, amount decimal(15,2) not null,
    -> transaction_date datetime not null default current_timestamp,
    -> foreign key (account_id) references accounts(account_id) on delete cascade,constraint chk_amount check (amount>0));
Query OK, 0 rows affected (0.05 sec)
```

Mohammed Ibrahim Sheriff U
Batch_4

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

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

• Customers

• Accounts

• Transactions

• Customers

insert into customers (first_name, last_name, DOB, email, phone_number, address) values('Ibrahim','Sheriff','2000-01-01','sheriff@gmail.com','+91-1234567890','123 main madurai'),

('Hari','Sudhan','2002-06-07','haris@gmail.com','+91-7865458965','12 west madurai'),

('Sheryl','Madina','1999-04-23','madinasheryl@gmail.com','+91-7854278546','123 main coimbatore'),

('Umar','Sheriff','1979-04-23','umar@gmail.com','+91-8627496874','74 east madurai'),

('David','Westly','1981-09-12','wdavid@gmail.com','+91-9834765430','11 cross street chennai'),

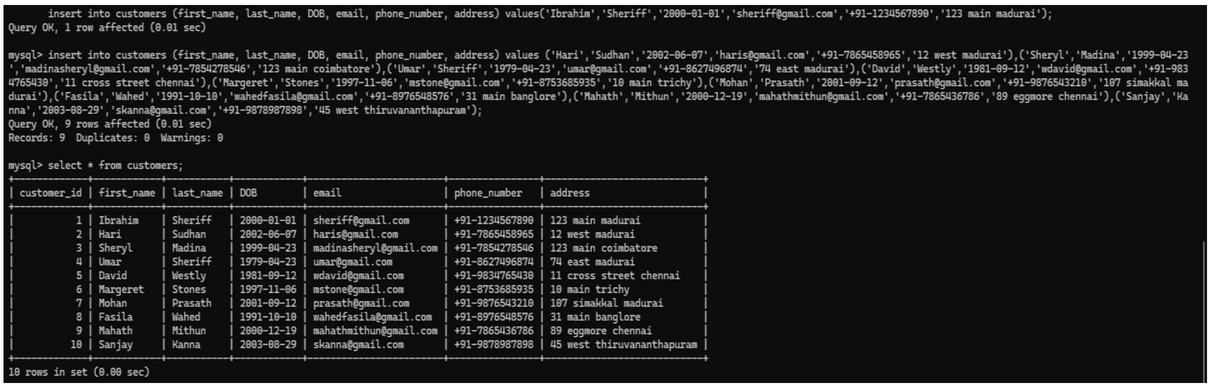
('Margeret','Stones','1997-11-06','mstone@gmail.com','+91-8753685935','10 main trichy'),

('Mohan','Prasath','2001-09-12','prasath@gmail.com','+91-9876543210','107 simakkal madurai'),

('Fasila','Wahed','1991-10-10','wahedfasila@gmail.com','+91-8976548576','31 main banglore'),

('Mahath','Mithun','2000-12-19','mahathmithun@gmail.com','+91-7865436786','89 eggmore chennai'),

('Sanjay','Kanna','2003-08-29','skanna@gmail.com','+91-9878987898','45 west thiruvananthapuram');



```
      insert into customers (first_name, last_name, DOB, email, phone_number, address) values('Ibrahim','Sheriff','2000-01-01','sheriff@gmail.com','+91-1234567890','123 main madurai');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers (first_name, last_name, DOB, email, phone_number, address) values ('Hari','Sudhan','2002-06-07','haris@gmail.com','+91-7865458965','12 west madurai'),('Sheryl','Madina','1999-04-23
','madinasheryl@gmail.com','+91-7854278546','123 main coimbatore'),('Umar','Sheriff','1979-04-23','umar@gmail.com','+91-8627496874','74 east madurai'),('David','Westly','1981-09-12','wdavid@gmail.com','+91-983
4765430','11 cross street chennai'),('Margeret','Stones','1997-11-06','mstone@gmail.com','+91-8753685935','10 main trichy'),('Mohan','Prasath','2001-09-12','prasath@gmail.com','+91-9876543210','107 simakkal ma
durai'),('Fasila','Wahed','1991-10-10','wahedfasila@gmail.com','+91-8976548576','31 main banglore'),('Mahath','Mithun','2000-12-19','mahathmithun@gmail.com','+91-7865436786','89 eggmore chennai'),('Sanjay','Ka
nna','2003-08-29','skanna@gmail.com','+91-9878987898','45 west thiruvananthapuram');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> select * from customers;
+-------------+------------+-----------+------------+------------------------+----------------+----------------------------+
| customer_id | first_name | last_name | DOB        | email                  | phone_number   | address                    |
+-------------+------------+-----------+------------+------------------------+----------------+----------------------------+
|           1 | Ibrahim    | Sheriff   | 2000-01-01 | sheriff@gmail.com      | +91-1234567890 | 123 main madurai           |
|           2 | Hari       | Sudhan    | 2002-06-07 | haris@gmail.com        | +91-7865458965 | 12 west madurai            |
|           3 | Sheryl     | Madina    | 1999-04-23 | madinasheryl@gmail.com | +91-7854278546 | 123 main coimbatore        |
|           4 | Umar       | Sheriff   | 1979-04-23 | umar@gmail.com         | +91-8627496874 | 74 east madurai            |
|           5 | David      | Westly    | 1981-09-12 | wdavid@gmail.com       | +91-9834765430 | 11 cross street chennai    |
|           6 | Margeret   | Stones    | 1997-11-06 | mstone@gmail.com       | +91-8753685935 | 10 main trichy             |
|           7 | Mohan      | Prasath   | 2001-09-12 | prasath@gmail.com      | +91-9876543210 | 107 simakkal madurai       |
|           8 | Fasila     | Wahed     | 1991-10-10 | wahedfasila@gmail.com  | +91-8976548576 | 31 main banglore           |
|           9 | Mahath     | Mithun    | 2000-12-19 | mahathmithun@gmail.com | +91-7865436786 | 89 eggmore chennai         |
|          10 | Sanjay     | Kanna     | 2003-08-29 | skanna@gmail.com       | +91-9878987898 | 45 west thiruvananthapuram |
+-------------+------------+-----------+------------+------------------------+----------------+----------------------------+
10 rows in set (0.00 sec)
```

• Accounts

insert into accounts (customer_id, account_type, balance) values

(1,'savings',8000.00),

(1,'current',2000),

(2,'savings',6000),

(3,'zero_balance',0.00),

(4,'current',3000),

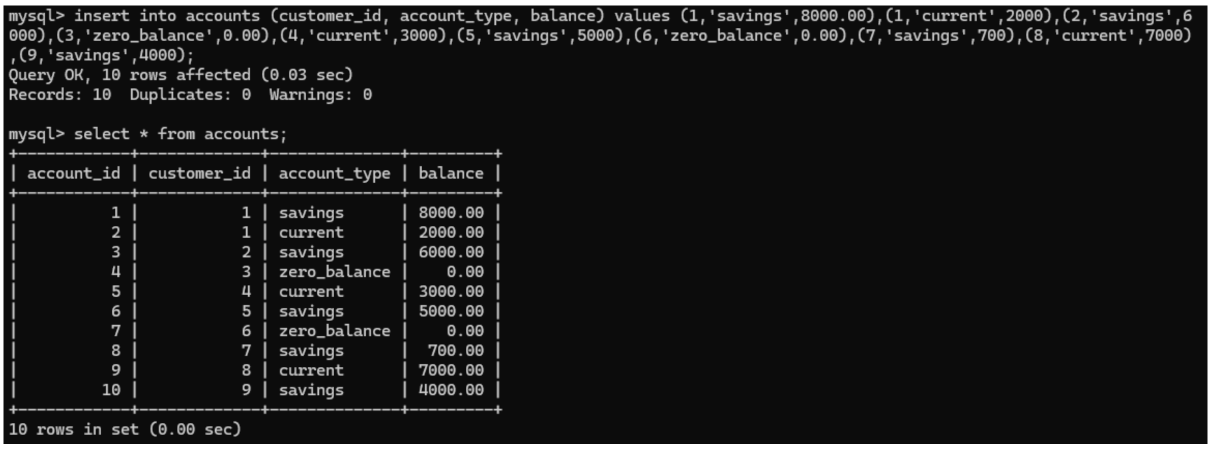
(5,'savings',5000),

(6,'zero_balance',0.00),

(7,'savings',700),

(8,'current',7000),

(9,'savings',4000);



```
mysql> insert into accounts (customer_id, account_type, balance) values (1,'savings',8000.00),(1,'current',2000),(2,'savings',6
000),(3,'zero_balance',0.00),(4,'current',3000),(5,'savings',5000),(6,'zero_balance',0.00),(7,'savings',700),(8,'current',7000)
,(9,'savings',4000);
Query OK, 10 rows affected (0.03 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from accounts;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|          1 |           1 | savings      | 8000.00 |
|          2 |           1 | current      | 2000.00 |
|          3 |           2 | savings      | 6000.00 |
|          4 |           3 | zero_balance |    0.00 |
|          5 |           4 | current      | 3000.00 |
|          6 |           5 | savings      | 5000.00 |
|          7 |           6 | zero_balance |    0.00 |
|          8 |           7 | savings      |  700.00 |
|          9 |           8 | current      | 7000.00 |
|         10 |           9 | savings      | 4000.00 |
+------------+-------------+--------------+---------+
10 rows in set (0.00 sec)
```

• Transactions

insert into transactions (account_id, transaction_type, amount, transaction_date) values (1,'withdrawal',2000,'2025-01-01 10:00:00'),

-> (2,'deposit',3000,'2025-01-02 11:00:00'),

-> (3,'deposit',3000,'2025-01-03 12:00:00'),

-> (4,'deposit',7000,'2025-01-04 13:00:00'),

-> (5,'withdrawal',1000,'2025-01-05 14:00:00'),

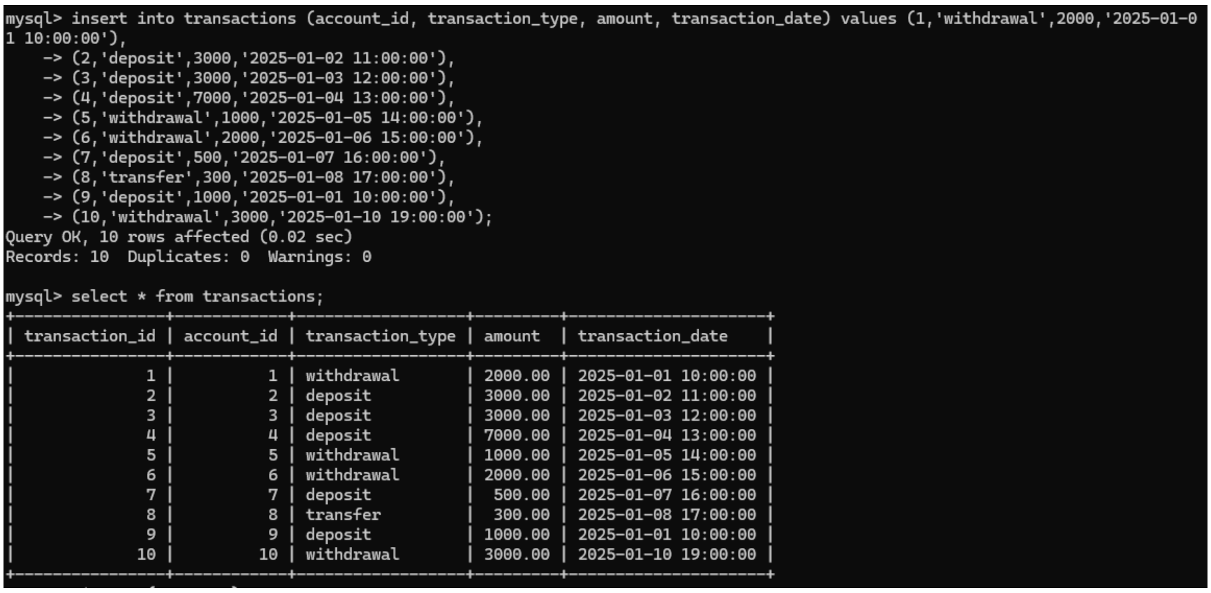
-> (6,'withdrawal',2000,'2025-01-06 15:00:00'),

-> (7,'deposit',500,'2025-01-07 16:00:00'),

-> (8,'transfer',300,'2025-01-08 17:00:00'),

-> (9,'deposit',1000,'2025-01-01 10:00:00'),

-> (10,'withdrawal',3000,'2025-01-10 19:00:00');



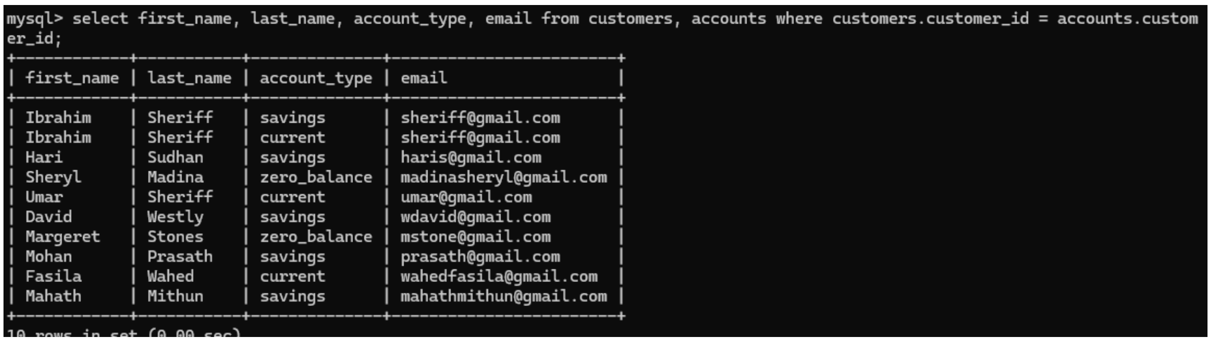
```
mysql> insert into transactions (account_id, transaction_type, amount, transaction_date) values (1,'withdrawal',2000,'2025-01-0
1 10:00:00'),
    -> (2,'deposit',3000,'2025-01-02 11:00:00'),
    -> (3,'deposit',3000,'2025-01-03 12:00:00'),
    -> (4,'deposit',7000,'2025-01-04 13:00:00'),
    -> (5,'withdrawal',1000,'2025-01-05 14:00:00'),
    -> (6,'withdrawal',2000,'2025-01-06 15:00:00'),
    -> (7,'deposit',500,'2025-01-07 16:00:00'),
    -> (8,'transfer',300,'2025-01-08 17:00:00'),
    -> (9,'deposit',1000,'2025-01-01 10:00:00'),
    -> (10,'withdrawal',3000,'2025-01-10 19:00:00');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from transactions;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|              1 |          1 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 |
|              2 |          2 | deposit          | 3000.00 | 2025-01-02 11:00:00 |
|              3 |          3 | deposit          | 3000.00 | 2025-01-03 12:00:00 |
|              4 |          4 | deposit          | 7000.00 | 2025-01-04 13:00:00 |
|              5 |          5 | withdrawal       | 1000.00 | 2025-01-05 14:00:00 |
|              6 |          6 | withdrawal       | 2000.00 | 2025-01-06 15:00:00 |
|              7 |          7 | deposit          |  500.00 | 2025-01-07 16:00:00 |
|              8 |          8 | transfer         |  300.00 | 2025-01-08 17:00:00 |
|              9 |          9 | deposit          | 1000.00 | 2025-01-01 10:00:00 |
|             10 |         10 | withdrawal       | 3000.00 | 2025-01-10 19:00:00 |
+----------------+------------+------------------+---------+---------------------+
```

Mohammed Ibrahim Sheriff U
Batch_4

2. Write SQL queries for the following tasks:

1. Write a SQL query to retrieve the name, account type and email of all customers.

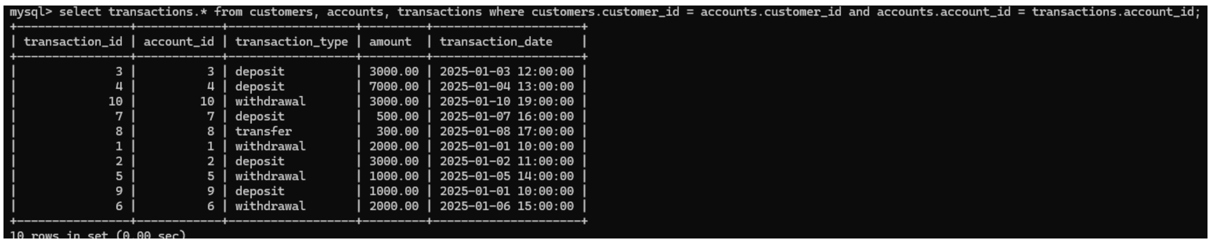
select first_name, last_name, account_type, email from customers, accounts where customers.customer_id = accounts.customer_id;



```
mysql> select first_name, last_name, account_type, email from customers, accounts where customers.customer_id = accounts.custom
er_id;
+------------+-----------+--------------+------------------------+
| first_name | last_name | account_type | email                  |
+------------+-----------+--------------+------------------------+
| Ibrahim    | Sheriff   | savings      | sheriff@gmail.com      |
| Ibrahim    | Sheriff   | current      | sheriff@gmail.com      |
| Hari       | Sudhan    | savings      | haris@gmail.com        |
| Sheryl     | Madina    | zero_balance | madinasheryl@gmail.com |
| Umar       | Sheriff   | current      | umar@gmail.com         |
| David      | Westly    | savings      | wdavid@gmail.com       |
| Margeret   | Stones    | zero_balance | mstone@gmail.com       |
| Mohan      | Prasath   | savings      | prasath@gmail.com      |
| Fasila     | Wahed     | current      | wahedfasila@gmail.com  |
| Mahath     | Mithun    | savings      | mahathmithun@gmail.com |
+------------+-----------+--------------+------------------------+
10 rows in set (0.00 sec)
```

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

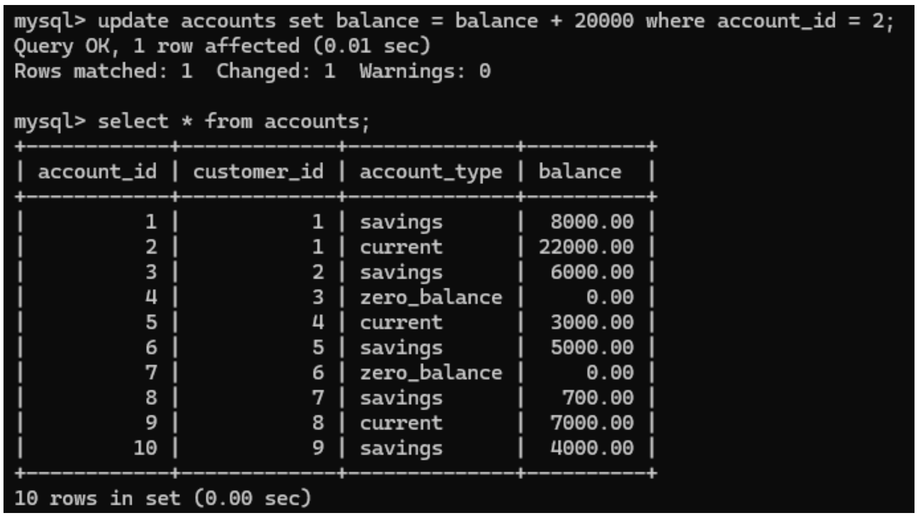
select transactions.* from customers, accounts, transactions where customers.customer_id = accounts.customer_id and accounts.account_id = transactions.account_id;



```
mysql> select transactions.* from customers, accounts, transactions where customers.customer_id = accounts.customer_id and accounts.account_id = transactions.account_id;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|              3 |          3 | deposit          | 3000.00 | 2025-01-03 12:00:00 |
|              4 |          4 | deposit          | 7000.00 | 2025-01-04 13:00:00 |
|             10 |         10 | withdrawal       | 3000.00 | 2025-01-10 19:00:00 |
|              7 |          7 | deposit          |  500.00 | 2025-01-07 16:00:00 |
|              8 |          8 | transfer         |  300.00 | 2025-01-08 17:00:00 |
|              1 |          1 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 |
|              2 |          2 | deposit          | 3000.00 | 2025-01-02 11:00:00 |
|              5 |          5 | withdrawal       | 1000.00 | 2025-01-05 14:00:00 |
|              9 |          9 | deposit          | 1000.00 | 2025-01-01 10:00:00 |
|              6 |          6 | withdrawal       | 2000.00 | 2025-01-06 15:00:00 |
+----------------+------------+------------------+---------+---------------------+
10 rows in set (0.00 sec)
```

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

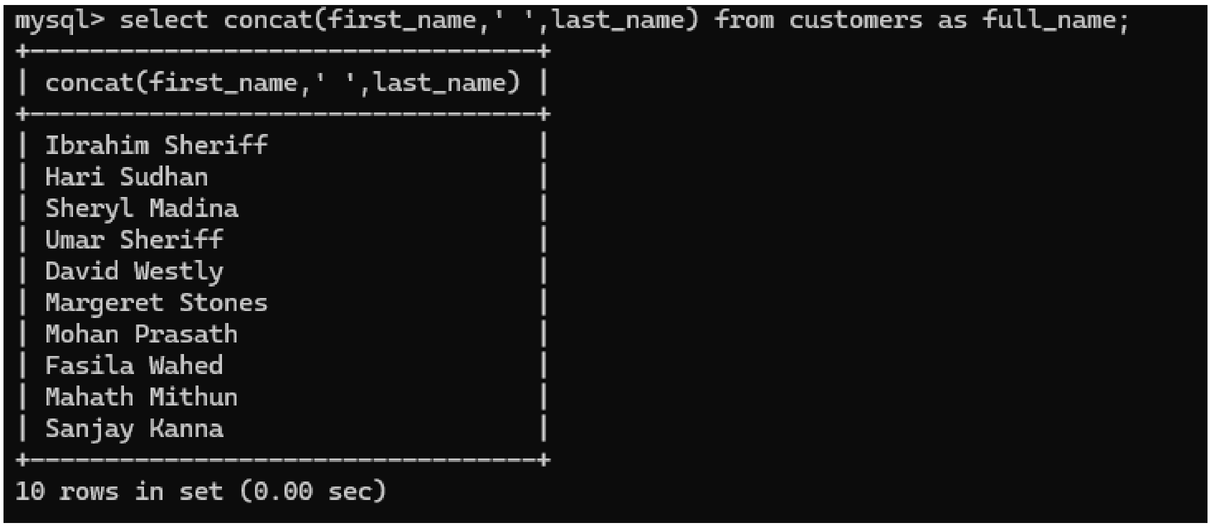
Update accounts set balance = balance + 20000 where account_id = 2;



```
mysql> update accounts set balance = balance + 20000 where account_id = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      |  8000.00 |
|          2 |           1 | current      | 22000.00 |
|          3 |           2 | savings      |  6000.00 |
|          4 |           3 | zero_balance |     0.00 |
|          5 |           4 | current      |  3000.00 |
|          6 |           5 | savings      |  5000.00 |
|          7 |           6 | zero_balance |     0.00 |
|          8 |           7 | savings      |   700.00 |
|          9 |           8 | current      |  7000.00 |
|         10 |           9 | savings      |  4000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

4. Write a SQL query to Combine first and last names of customers as a full_name.

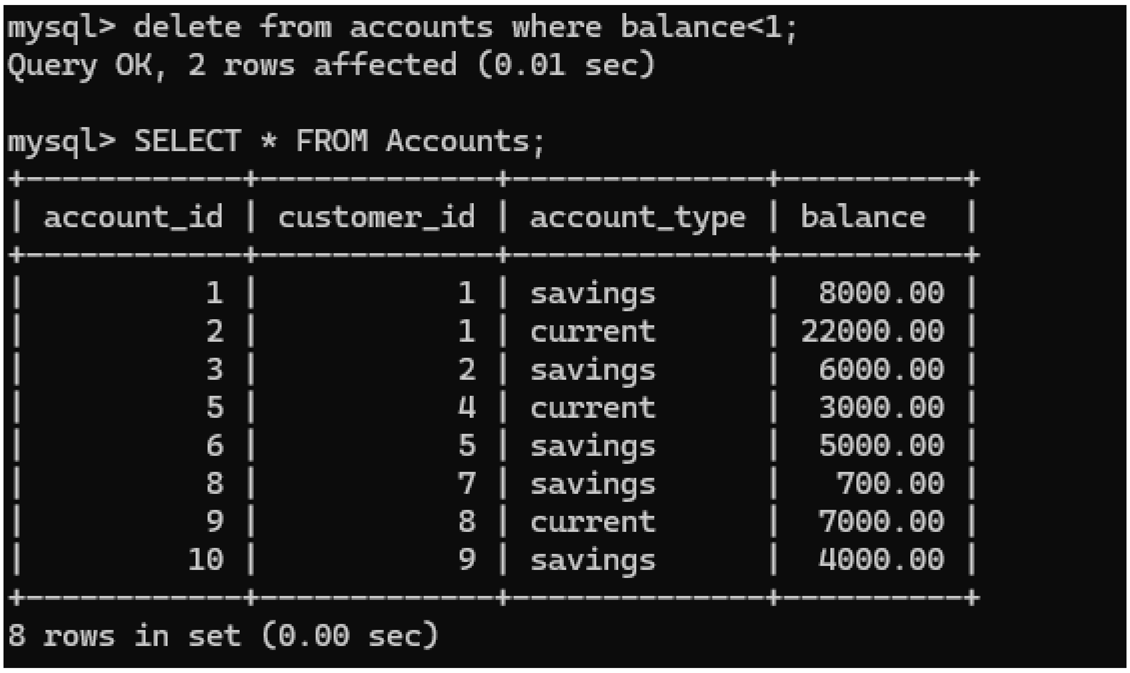
select concat(first_name,' ',last_name) from customers as full_name;



```
mysql> select concat(first_name,' ',last_name) from customers as full_name;
+----------------------------------+
| concat(first_name,' ',last_name) |
+----------------------------------+
| Ibrahim Sheriff                  |
| Hari Sudhan                      |
| Sheryl Madina                    |
| Umar Sheriff                     |
| David Westly                     |
| Margeret Stones                  |
| Mohan Prasath                    |
| Fasila Wahed                     |
| Mahath Mithun                    |
| Sanjay Kanna                     |
+----------------------------------+
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.

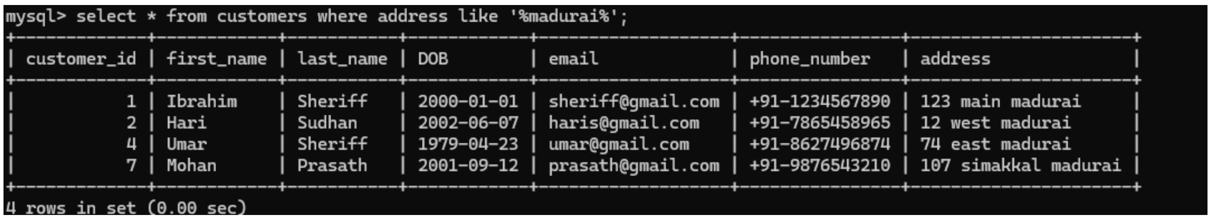
delete from accounts where balance<1;



```
mysql> delete from accounts where balance<1;
Query OK, 2 rows affected (0.01 sec)

mysql> SELECT * FROM Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      |  8000.00 |
|          2 |           1 | current      | 22000.00 |
|          3 |           2 | savings      |  6000.00 |
|          5 |           4 | current      |  3000.00 |
|          6 |           5 | savings      |  5000.00 |
|          8 |           7 | savings      |   700.00 |
|          9 |           8 | current      |  7000.00 |
|         10 |           9 | savings      |  4000.00 |
+------------+-------------+--------------+----------+
8 rows in set (0.00 sec)
```

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

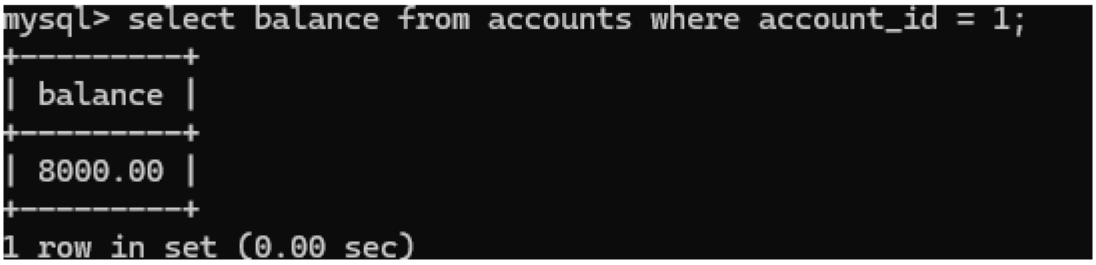
select * from customers where address like '%madurai%';



```
mysql> select * from customers where address like '%madurai%';
+-------------+------------+-----------+------------+--------------------+----------------+----------------------+
| customer_id | first_name | last_name | DOB        | email              | phone_number   | address              |
+-------------+------------+-----------+------------+--------------------+----------------+----------------------+
|           1 | Ibrahim    | Sheriff   | 2000-01-01 | sheriff@gmail.com  | +91-1234567890 | 123 main madurai     |
|           2 | Hari       | Sudhan    | 2002-06-07 | haris@gmail.com    | +91-7865458965 | 12 west madurai      |
|           4 | Umar       | Sheriff   | 1979-04-23 | umar@gmail.com     | +91-8627496874 | 74 east madurai      |
|           7 | Mohan      | Prasath   | 2001-09-12 | prasath@gmail.com  | +91-9876543210 | 107 simakkal madurai |
+-------------+------------+-----------+------------+--------------------+----------------+----------------------+
4 rows in set (0.00 sec)
```

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

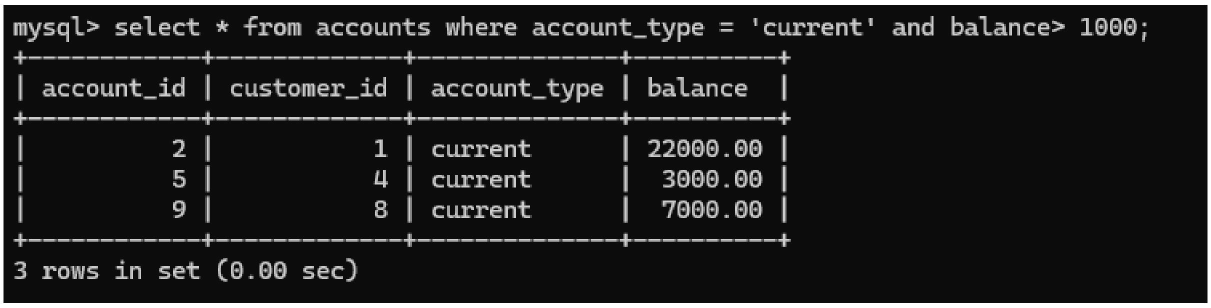
select balance from accounts where account_id = 1;



```
mysql> select balance from accounts where account_id = 1;
+---------+
| balance |
+---------+
| 8000.00 |
+---------+
1 row in set (0.00 sec)
```

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

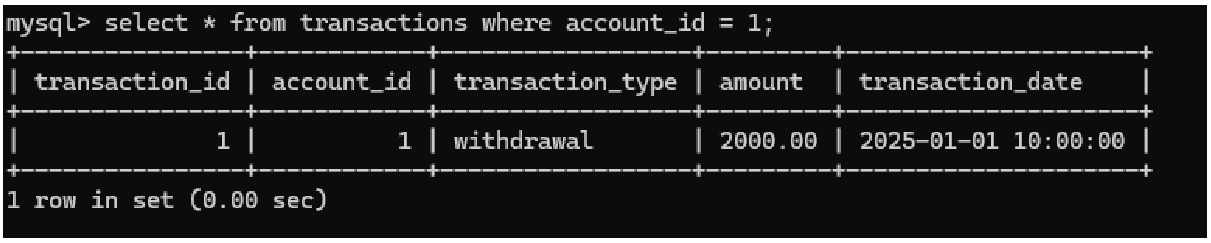
select * from accounts where account_type = 'current' and balance> 1000;



```
mysql> select * from accounts where account_type = 'current' and balance> 1000;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          2 |           1 | current      | 22000.00 |
|          5 |           4 | current      |  3000.00 |
|          9 |           8 | current      |  7000.00 |
+------------+-------------+--------------+----------+
3 rows in set (0.00 sec)
```

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

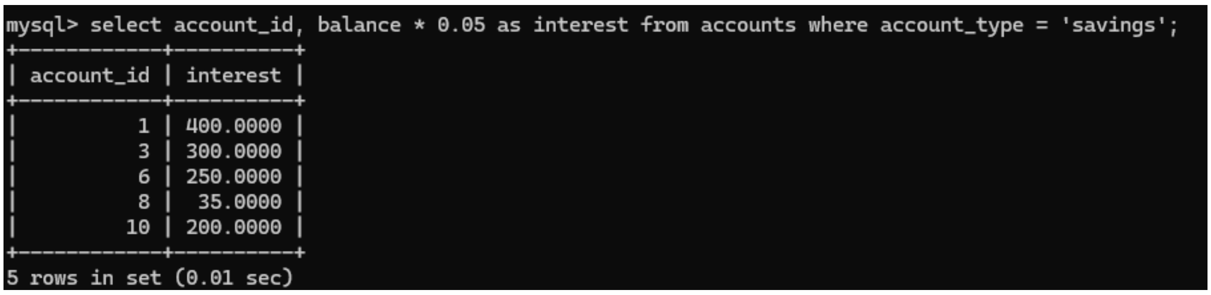
select * from transactions where account_id = 1;



```
mysql> select * from transactions where account_id = 1;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|              1 |          1 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 |
+----------------+------------+------------------+---------+---------------------+
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.

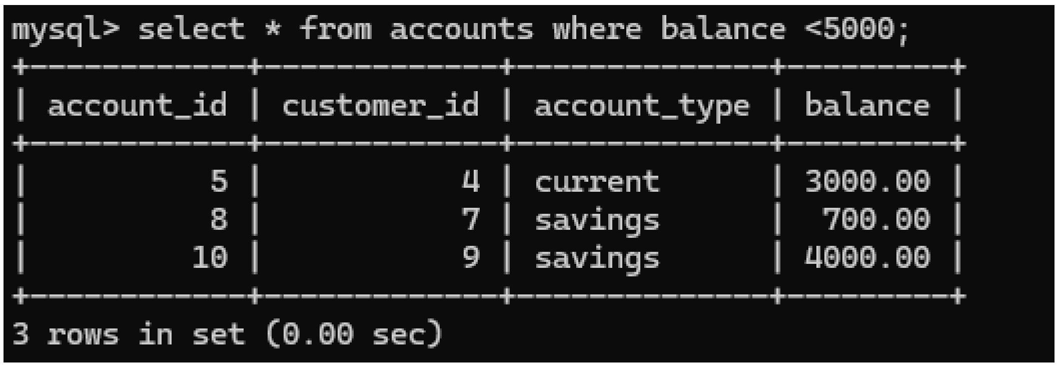
select account_id, balance * 0.05 as interest from accounts where account_type = 'savings';



```
mysql> select account_id, balance * 0.05 as interest from accounts where account_type = 'savings';
+------------+----------+
| account_id | interest |
+------------+----------+
|          1 | 400.0000 |
|          3 | 300.0000 |
|          6 | 250.0000 |
|          8 |  35.0000 |
|         10 | 200.0000 |
+------------+----------+
5 rows in set (0.01 sec)
```

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

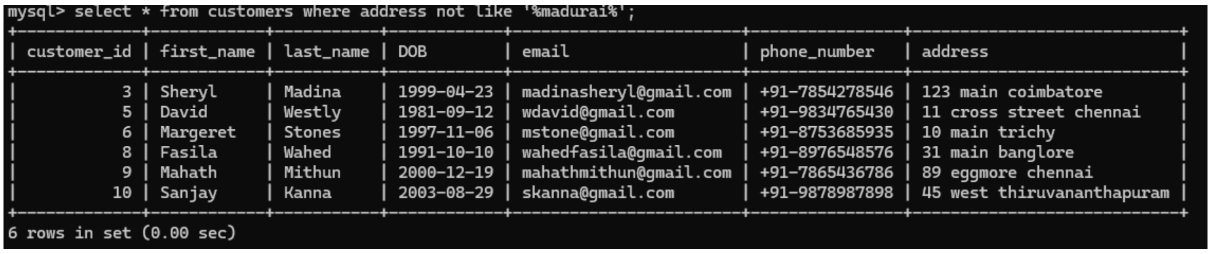
select * from accounts where balance <5000;



```
mysql> select * from accounts where balance <5000;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|          5 |           4 | current      | 3000.00 |
|          8 |           7 | savings      |  700.00 |
|         10 |           9 | savings      | 4000.00 |
+------------+-------------+--------------+---------+
3 rows in 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 '%madurai%';



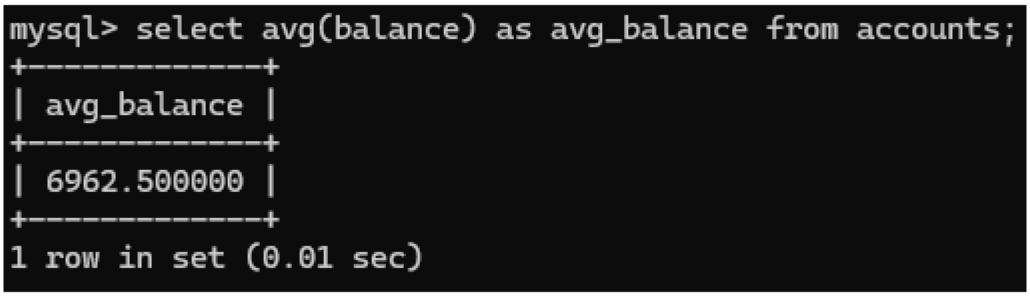
```
mysql> select * from customers where address not like '%madurai%';
+-------------+------------+-----------+------------+--------------------------+----------------+----------------------------+
| customer_id | first_name | last_name | DOB        | email                    | phone_number   | address                    |
+-------------+------------+-----------+------------+--------------------------+----------------+----------------------------+
|           3 | Sheryl     | Madina    | 1999-04-23 | madinasheryl@gmail.com   | +91-7854278546 | 123 main coimbatore        |
|           5 | David      | Westly    | 1981-09-12 | wdavid@gmail.com         | +91-9834765430 | 11 cross street chennai    |
|           6 | Margeret   | Stones    | 1997-11-06 | mstone@gmail.com         | +91-8753685935 | 10 main trichy             |
|           8 | Fasila     | Wahed     | 1991-10-10 | wahedfasila@gmail.com    | +91-8976548576 | 31 main banglore           |
|           9 | Mahath     | Mithun    | 2000-12-19 | mahathmithun@gmail.com   | +91-7865436786 | 89 eggmore chennai         |
|          10 | Sanjay     | Kanna     | 2003-08-29 | skanna@gmail.com         | +91-9878987898 | 45 west thiruvananthapuram |
+-------------+------------+-----------+------------+--------------------------+----------------+----------------------------+
6 rows in set (0.00 sec)
```

Mohammed Ibrahim Sheriff U
Batch_4

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

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

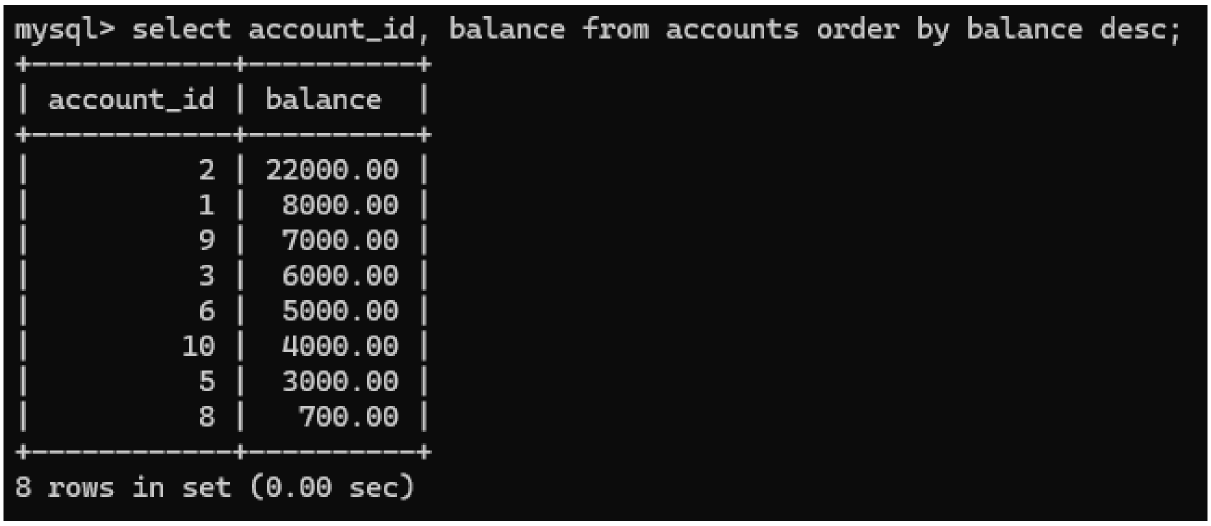
select avg(balance) as avg_balance from accounts;



```
mysql> select avg(balance) as avg_balance from accounts;
+-------------+
| avg_balance |
+-------------+
| 6962.500000 |
+-------------+
1 row in set (0.01 sec)
```

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

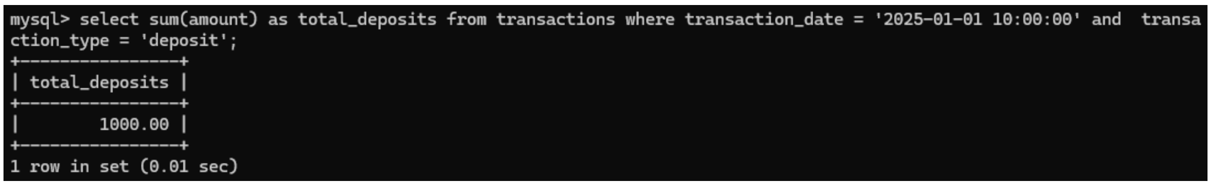
select account_id, balance from accounts order by balance desc;



```
mysql> select account_id, balance from accounts order by balance desc;
+------------+----------+
| account_id | balance  |
+------------+----------+
|          2 | 22000.00 |
|          1 |  8000.00 |
|          9 |  7000.00 |
|          3 |  6000.00 |
|          6 |  5000.00 |
|         10 |  4000.00 |
|          5 |  3000.00 |
|          8 |   700.00 |
+------------+----------+
8 rows in set (0.00 sec)
```

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

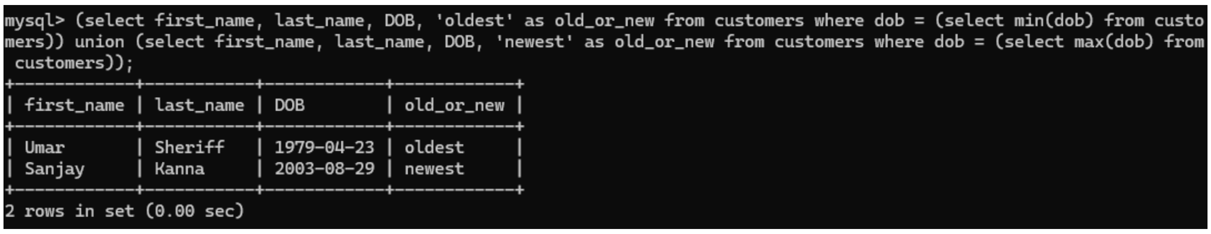
select sum(amount) as total_deposits from transactions where transaction_date = '2025-01-01 10:00:00' and transaction_type = 'deposit';



```
mysql> select sum(amount) as total_deposits from transactions where transaction_date = '2025-01-01 10:00:00' and  transa
ction_type = 'deposit';
+----------------+
| total_deposits |
+----------------+
|       1000.00 |
+----------------+
1 row in set (0.01 sec)
```

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

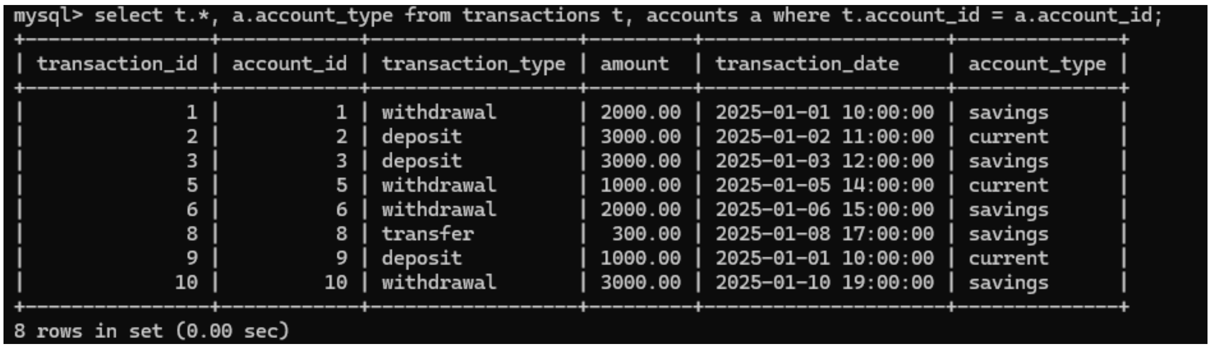
(select first_name, last_name, DOB, 'oldest' as old_or_new from customers where dob = (select min(dob) from customers)) union (select first_name, last_name, DOB, 'newest' as old_or_new from customers where dob = (select max(dob) from customers));



```
mysql> (select first_name, last_name, DOB, 'oldest' as old_or_new from customers where dob = (select min(dob) from custo
mers)) union (select first_name, last_name, DOB, 'newest' as old_or_new from customers where dob = (select max(dob) from
 customers));
+------------+-----------+------------+------------+
| first_name | last_name | DOB        | old_or_new |
+------------+-----------+------------+------------+
| Umar       | Sheriff   | 1979-04-23 | oldest     |
| Sanjay     | Kanna     | 2003-08-29 | newest     |
+------------+-----------+------------+------------+
2 rows in set (0.00 sec)
```

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

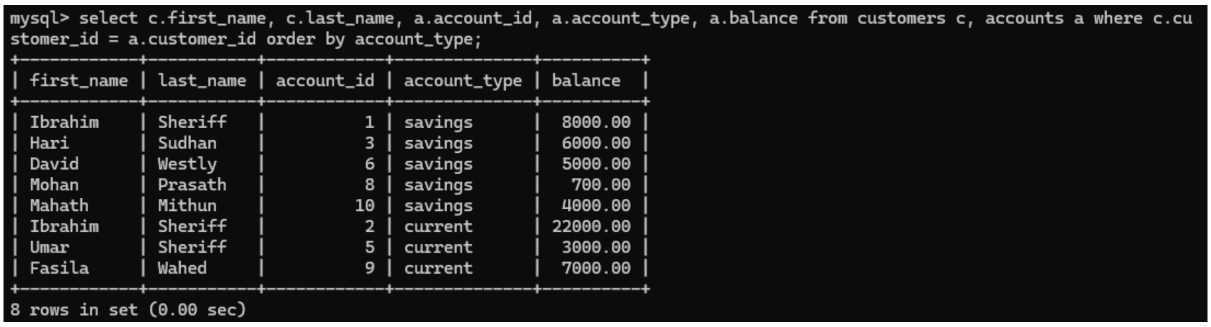
select t.*, a.account_type from transactions t, accounts a where t.account_id = a.account_id;



```
mysql> select t.*, a.account_type from transactions t, accounts a where t.account_id = a.account_id;
+----------------+------------+------------------+---------+---------------------+--------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    | account_type |
+----------------+------------+------------------+---------+---------------------+--------------+
|              1 |          1 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 | savings      |
|              2 |          2 | deposit          | 3000.00 | 2025-01-02 11:00:00 | current      |
|              3 |          3 | deposit          | 3000.00 | 2025-01-03 12:00:00 | savings      |
|              5 |          5 | withdrawal       | 1000.00 | 2025-01-05 14:00:00 | current      |
|              6 |          6 | withdrawal       | 2000.00 | 2025-01-06 15:00:00 | savings      |
|              8 |          8 | transfer         |  300.00 | 2025-01-08 17:00:00 | savings      |
|              9 |          9 | deposit          | 1000.00 | 2025-01-01 10:00:00 | current      |
|             10 |         10 | withdrawal       | 3000.00 | 2025-01-10 19:00:00 | savings      |
+----------------+------------+------------------+---------+---------------------+--------------+
8 rows in set (0.00 sec)
```

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

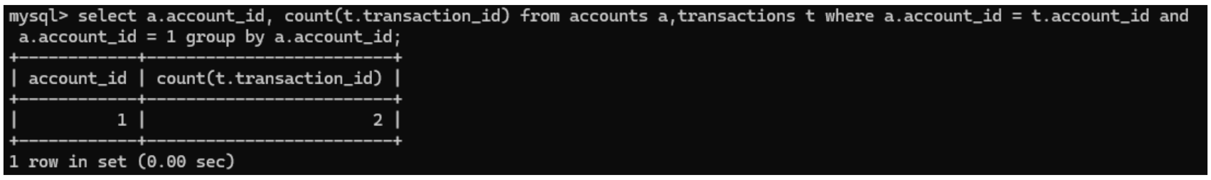
select c.first_name, c.last_name, a.account_id, a.account_type, a.balance from customers c, accounts a where c.customer_id = a.customer_id order by account_type;



```
mysql> select c.first_name, c.last_name, a.account_id, a.account_type, a.balance from customers c, accounts a where c.cu
stomer_id = a.customer_id order by account_type;
+------------+-----------+------------+--------------+----------+
| first_name | last_name | account_id | account_type | balance  |
+------------+-----------+------------+--------------+----------+
| Ibrahim    | Sheriff   |          1 | savings      |  8000.00 |
| Hari       | Sudhan    |          3 | savings      |  6000.00 |
| David      | Westly    |          6 | savings      |  5000.00 |
| Mohan      | Prasath   |          8 | savings      |   700.00 |
| Mahath     | Mithun    |         10 | savings      |  4000.00 |
| Ibrahim    | Sheriff   |          2 | current      | 22000.00 |
| Umar       | Sheriff   |          5 | current      |  3000.00 |
| Fasila     | Wahed     |          9 | current      |  7000.00 |
+------------+-----------+------------+--------------+----------+
8 rows in set (0.00 sec)
```

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

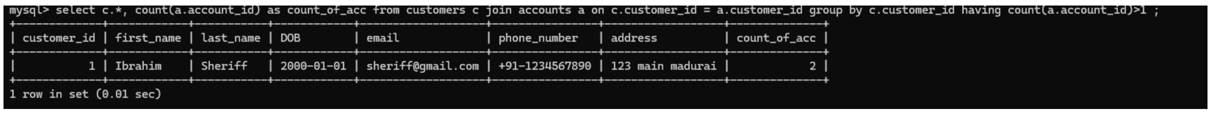
select a.account_id, count(t.transaction_id) from accounts a,transactions t where a.account_id = t.account_id and a.account_id = 1 group by a.account_id;



```
mysql> select a.account_id, count(t.transaction_id) from accounts a,transactions t where a.account_id = t.account_id and
 a.account_id = 1 group by a.account_id;
+------------+-------------------------+
| account_id | count(t.transaction_id) |
+------------+-------------------------+
|          1 |                       2 |
+------------+-------------------------+
1 row in set (0.00 sec)
```

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

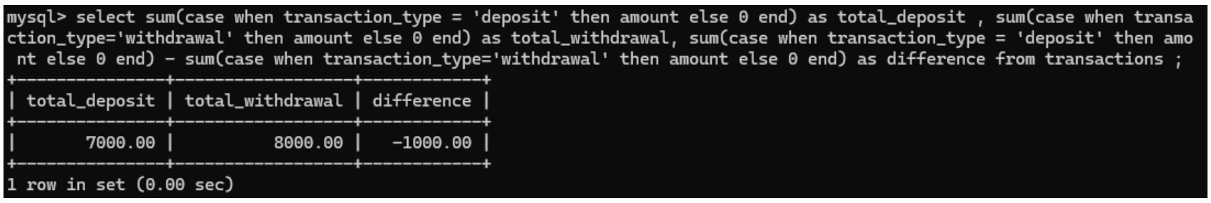
select c.*, count(a.account_id) as count_of_acc from customers c join accounts a on c.customer_id = a.customer_id group by c.customer_id having count(a.account_id)>1 ;



```
mysql> select c.*, count(a.account_id) as count_of_acc from customers c join accounts a on c.customer_id = a.customer_id group by c.customer_id having count(a.account_id)>1 ;
+-------------+------------+-----------+------------+-------------------+-----------------+------------------+--------------+
| customer_id | first_name | last_name | DOB        | email             | phone_number    | address          | count_of_acc |
+-------------+------------+-----------+------------+-------------------+-----------------+------------------+--------------+
|           1 | Ibrahim    | Sheriff   | 2000-01-01 | sheriff@gmail.com | +91-1234567890  | 123 main madurai |            2 |
+-------------+------------+-----------+------------+-------------------+-----------------+------------------+--------------+
1 row in set (0.01 sec)
```

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

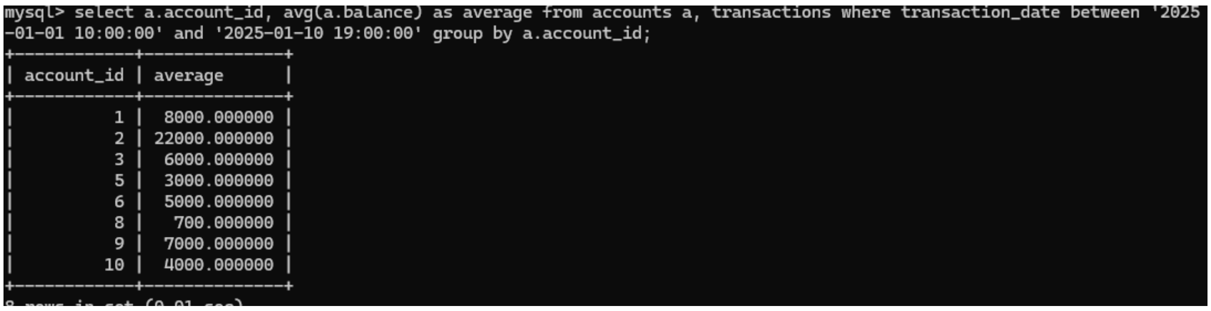
select sum(case when transaction_type = 'deposit' then amount else 0 end) as total_deposit , sum(case when transaction_type='withdrawal' then amount else 0 end) as total_withdrawal, sum(case when transaction_type = 'deposit' then amo nt else 0 end) - sum(case when transaction_type='withdrawal' then amount else 0 end) as difference from transactions ;



```
mysql> select sum(case when transaction_type = 'deposit' then amount else 0 end) as total_deposit , sum(case when transa
ction_type='withdrawal' then amount else 0 end) as total_withdrawal, sum(case when transaction_type = 'deposit' then amo
 nt else 0 end) - sum(case when transaction_type='withdrawal' then amount else 0 end) as difference from transactions ;
+---------------+------------------+------------+
| total_deposit | total_withdrawal | difference |
+---------------+------------------+------------+
|       7000.00 |          8000.00 |   -1000.00 |
+---------------+------------------+------------+
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.

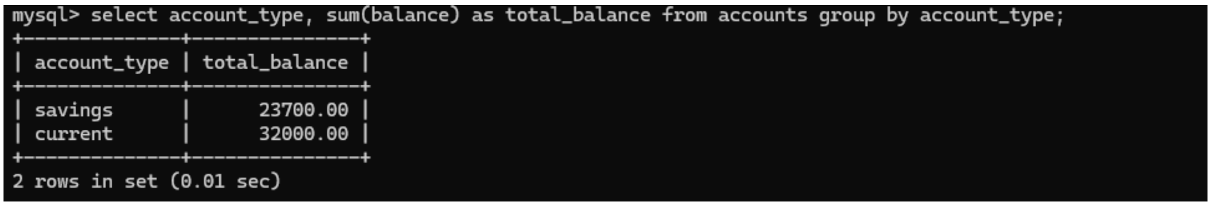
select a.account_id, avg(a.balance) as average from accounts a, transactions where transaction_date between '2025-01-01 10:00:00' and '2025-01-10 19:00:00' group by a.account_id;



```
mysql> select a.account_id, avg(a.balance) as average from accounts a, transactions where transaction_date between '2025
-01-01 10:00:00' and '2025-01-10 19:00:00' group by a.account_id;
+------------+--------------+
| account_id | average      |
+------------+--------------+
|          1 |  8000.000000 |
|          2 | 22000.000000 |
|          3 |  6000.000000 |
|          5 |  3000.000000 |
|          6 |  5000.000000 |
|          8 |   700.000000 |
|          9 |  7000.000000 |
|         10 |  4000.000000 |
+------------+--------------+
```

11. Calculate the total balance for each account type.

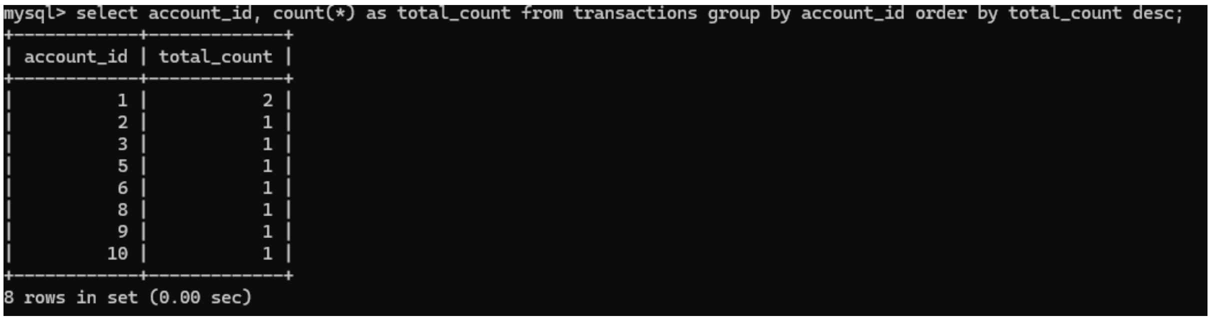
select account_type, sum(balance) as total_balance from accounts group by account_type;



```
mysql> select account_type, sum(balance) as total_balance from accounts group by account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      23700.00 |
| current      |      32000.00 |
+--------------+---------------+
2 rows in set (0.01 sec)
```

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

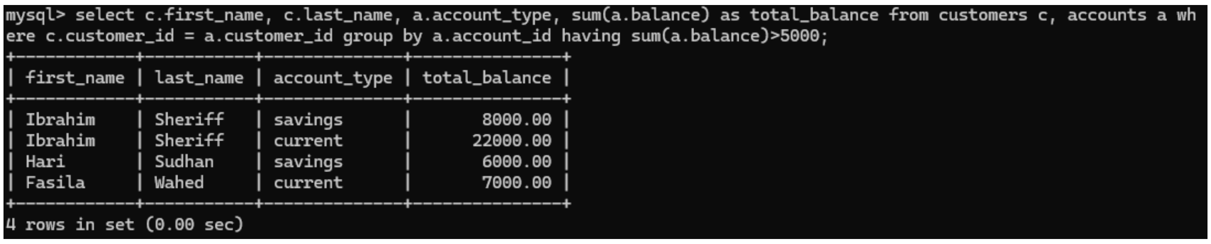
select account_id, count(*) as total_count from transactions group by account_id order by total_count desc;



```
mysql> select account_id, count(*) as total_count from transactions group by account_id order by total_count desc;
+------------+-------------+
| account_id | total_count |
+------------+-------------+
|          1 |           2 |
|          2 |           1 |
|          3 |           1 |
|          5 |           1 |
|          6 |           1 |
|          8 |           1 |
|          9 |           1 |
|         10 |           1 |
+------------+-------------+
8 rows in set (0.00 sec)
```

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

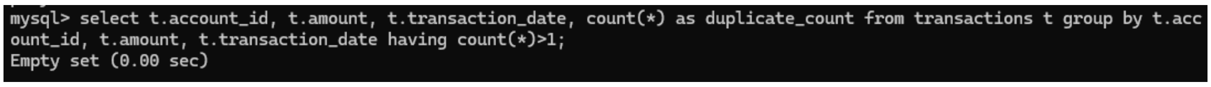
select c.first_name, c.last_name, a.account_type, sum(a.balance) as total_balance from customers c, accounts a where c.customer_id = a.customer_id group by a.account_id having sum(a.balance)>5000;



```
mysql> select c.first_name, c.last_name, a.account_type, sum(a.balance) as total_balance from customers c, accounts a wh
ere c.customer_id = a.customer_id group by a.account_id having sum(a.balance)>5000;
+------------+-----------+--------------+---------------+
| first_name | last_name | account_type | total_balance |
+------------+-----------+--------------+---------------+
| Ibrahim    | Sheriff   | savings      |       8000.00 |
| Ibrahim    | Sheriff   | current      |      22000.00 |
| Hari       | Sudhan    | savings      |       6000.00 |
| Fasila     | Wahed     | current      |       7000.00 |
+------------+-----------+--------------+---------------+
4 rows in set (0.00 sec)
```

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

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



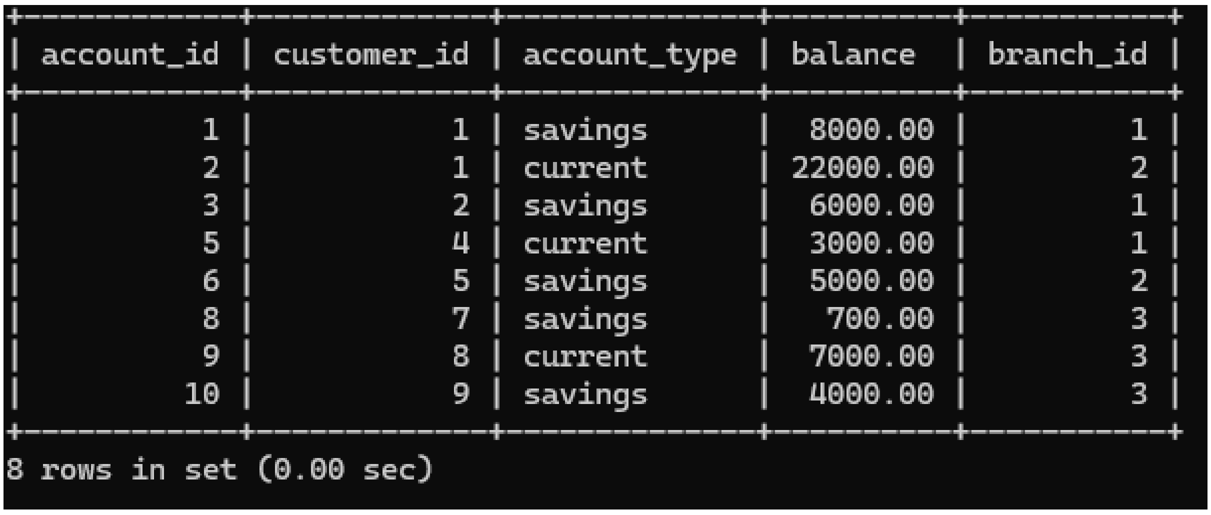
```
mysql> select t.account_id, t.amount, t.transaction_date, count(*) as duplicate_count from transactions t group by t.acc
ount_id, t.amount, t.transaction_date having count(*)>1;
Empty set (0.00 sec)
```

Mohammed Ibrahim Sheriff U
Batch_4

Tasks 4: Subquery and its type:

Accounts table updated : (updated to use branch_id)

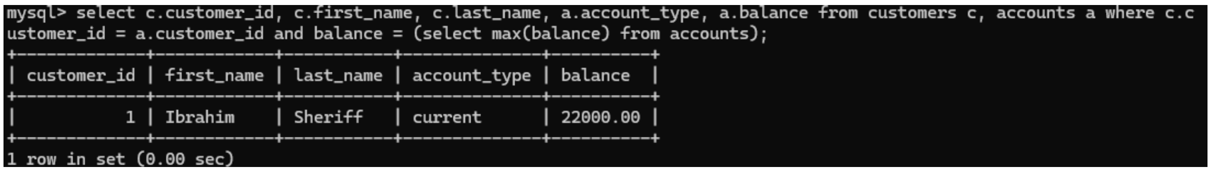
alter table accounts add branch_id int;



```
+------------+-------------+--------------+----------+-----------+
| account_id | customer_id | account_type | balance  | branch_id |
+------------+-------------+--------------+----------+-----------+
|          1 |           1 | savings      |  8000.00 |         1 |
|          2 |           1 | current      | 22000.00 |         2 |
|          3 |           2 | savings      |  6000.00 |         1 |
|          5 |           4 | current      |  3000.00 |         1 |
|          6 |           5 | savings      |  5000.00 |         2 |
|          8 |           7 | savings      |   700.00 |         3 |
|          9 |           8 | current      |  7000.00 |         3 |
|         10 |           9 | savings      |  4000.00 |         3 |
+------------+-------------+--------------+----------+-----------+
8 rows in set (0.00 sec)
```

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

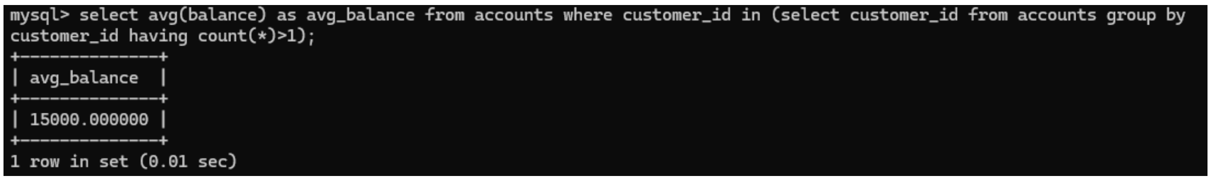
select c.customer_id, c.first_name, c.last_name, a.account_type, a.balance from customers c, accounts a where c.customer_id = a.customer_id and balance = (select max(balance) from accounts);



```
mysql> select c.customer_id, c.first_name, c.last_name, a.account_type, a.balance from customers c, accounts a where c.c
ustomer_id = a.customer_id and balance = (select max(balance) from accounts);
+-------------+------------+-----------+--------------+----------+
| customer_id | first_name | last_name | account_type | balance  |
+-------------+------------+-----------+--------------+----------+
|           1 | Ibrahim    | Sheriff   | current      | 22000.00 |
+-------------+------------+-----------+--------------+----------+
1 row in set (0.00 sec)
```

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

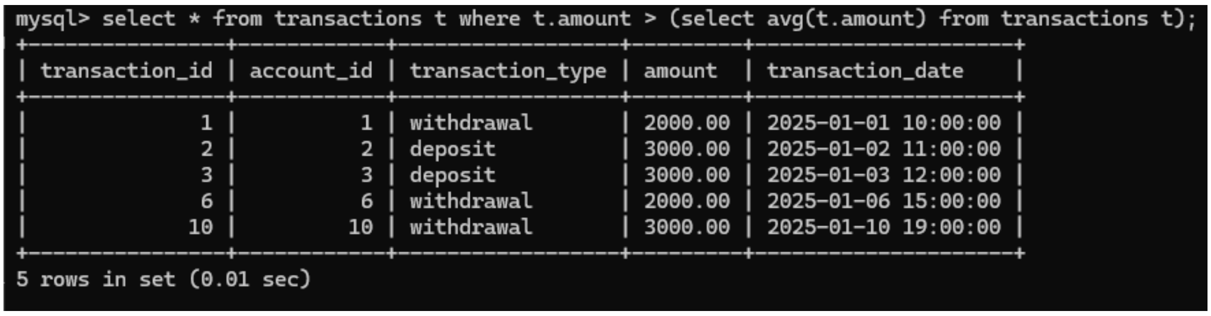
select avg(balance) as avg_balance from accounts where customer_id in (select customer_id from accounts group by customer_id having count(*)>1);



```
mysql> select avg(balance) as avg_balance from accounts where customer_id in (select customer_id from accounts group by
customer_id having count(*)>1);
+--------------+
| avg_balance  |
+--------------+
| 15000.000000 |
+--------------+
1 row in set (0.01 sec)
```

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

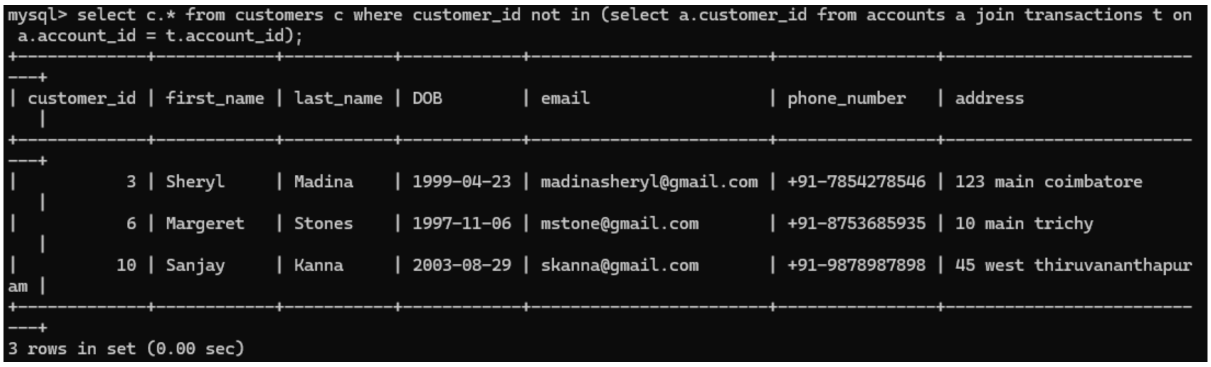
select * from transactions t where t.amount > (select avg(t.amount) from transactions t);



```
mysql> select * from transactions t where t.amount > (select avg(t.amount) from transactions t);
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|              1 |          1 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 |
|              2 |          2 | deposit          | 3000.00 | 2025-01-02 11:00:00 |
|              3 |          3 | deposit          | 3000.00 | 2025-01-03 12:00:00 |
|              6 |          6 | withdrawal       | 2000.00 | 2025-01-06 15:00:00 |
|             10 |         10 | withdrawal       | 3000.00 | 2025-01-10 19:00:00 |
+----------------+------------+------------------+---------+---------------------+
5 rows in set (0.01 sec)
```

4. Identify customers who have no recorded transactions.

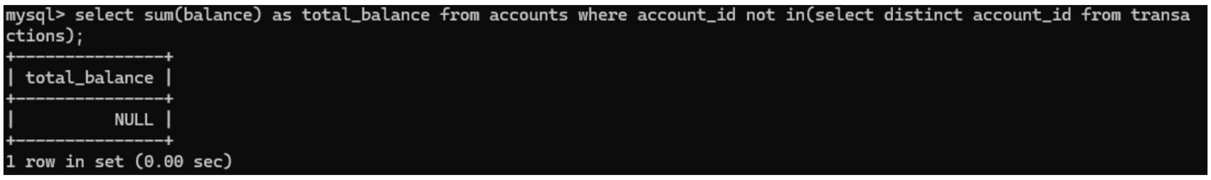
select c.* from customers c where customer_id not in (select a.customer_id from accounts a join transactions t on a.account_id = t.account_id);



```
mysql> select c.* from customers c where customer_id not in (select a.customer_id from accounts a join transactions t on
 a.account_id = t.account_id);
+-------------+------------+-----------+------------+------------------------+---------------+------------------------
---+
| customer_id | first_name | last_name | DOB        | email                  | phone_number  | address
   |
+-------------+------------+-----------+------------+------------------------+---------------+------------------------
---+
|           3 | Sheryl     | Madina    | 1999-04-23 | madinasheryl@gmail.com | +91-7854278546 | 123 main coimbatore
   |
|           6 | Margeret   | Stones    | 1997-11-06 | mstone@gmail.com       | +91-8753685935 | 10 main trichy
   |
|          10 | Sanjay     | Kanna     | 2003-08-29 | skanna@gmail.com       | +91-9878987898 | 45 west thiruvananthapur
am |
+-------------+------------+-----------+------------+------------------------+---------------+------------------------
---+
3 rows in set (0.00 sec)
```

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

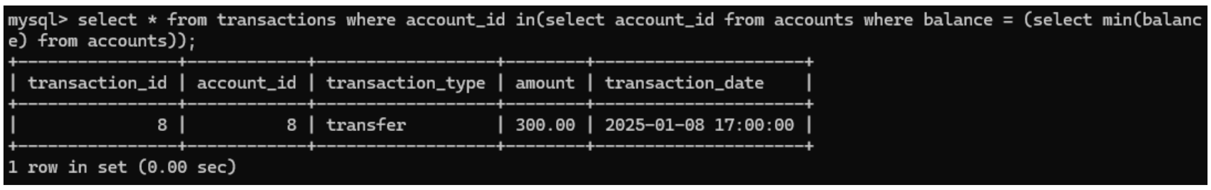
select sum(balance) as total_balance from accounts where account_id not in(select distinct account_id from transactions);



```
mysql> select sum(balance) as total_balance from accounts where account_id not in(select distinct account_id from transa
ctions);
+---------------+
| total_balance |
+---------------+
|          NULL |
+---------------+
1 row in set (0.00 sec)
```

6. Retrieve transactions for accounts with the lowest balance.

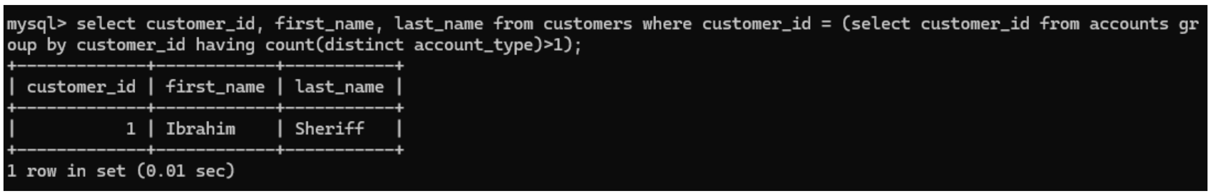
select * from transactions where account_id in(select account_id from accounts where balance = (select min(balance) from accounts));



```
mysql> select * from transactions where account_id in(select account_id from accounts where balance = (select min(balanc
e) from accounts));
+----------------+------------+------------------+--------+---------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date    |
+----------------+------------+------------------+--------+---------------------+
|              8 |          8 | transfer         | 300.00 | 2025-01-08 17:00:00 |
+----------------+------------+------------------+--------+---------------------+
1 row in set (0.00 sec)
```

7. Identify customers who have accounts of multiple types.

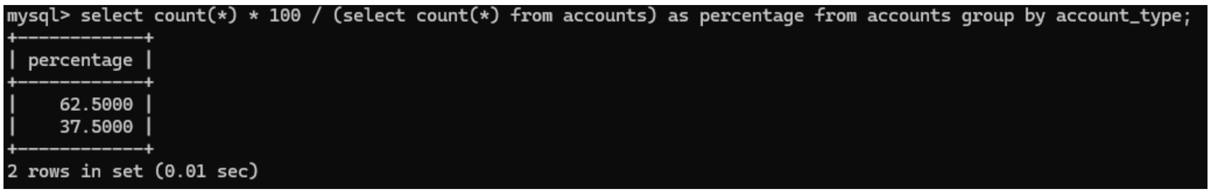
select customer_id, first_name, last_name from customers where customer_id = (select customer_id from accounts group by customer_id having count(distinct account_type)>1);



```
mysql> select customer_id, first_name, last_name from customers where customer_id = (select customer_id from accounts gr
oup by customer_id having count(distinct account_type)>1);
+-------------+------------+-----------+
| customer_id | first_name | last_name |
+-------------+------------+-----------+
|           1 | Ibrahim    | Sheriff   |
+-------------+------------+-----------+
1 row in set (0.01 sec)
```

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

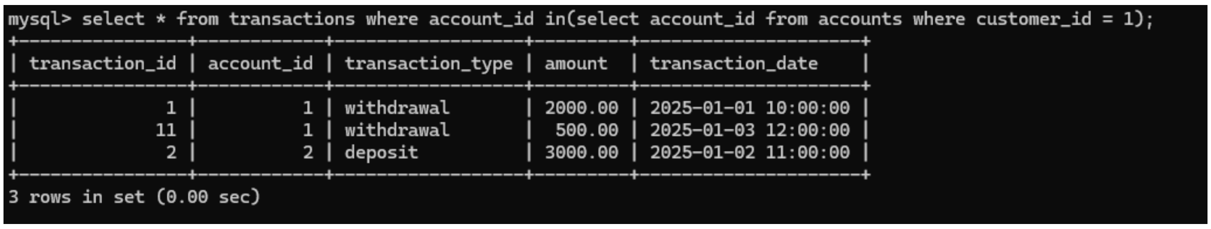
select count(*) * 100 / (select count(*) from accounts) as percentage from accounts group by account_type;



```
mysql> select count(*) * 100 / (select count(*) from accounts) as percentage from accounts group by account_type;
+------------+
| percentage |
+------------+
|    62.5000 |
|    37.5000 |
+------------+
2 rows in set (0.01 sec)
```

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

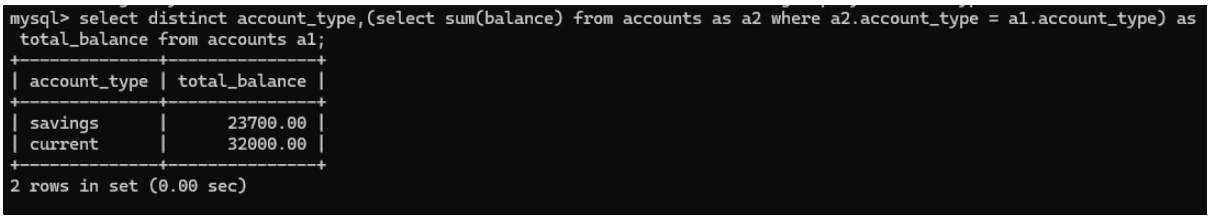
select * from transactions where account_id in(select account_id from accounts where customer_id = 1);



```
mysql> 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 | withdrawal       | 2000.00 | 2025-01-01 10:00:00 |
|             11 |          1 | withdrawal       |  500.00 | 2025-01-03 12:00:00 |
|              2 |          2 | deposit          | 3000.00 | 2025-01-02 11:00:00 |
+----------------+------------+------------------+---------+---------------------+
3 rows in set (0.00 sec)
```

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

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



```
mysql> select distinct account_type,(select sum(balance) from accounts as a2 where a2.account_type = a1.account_type) as
 total_balance from accounts a1;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      23700.00 |
| current      |      32000.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```