

Assignment-3:

Tasks 1:

Database Design:

1. Create the database named "HMBank";

```
MySQL 8.0 Command Line Client

mysql> create database HMBank;
Query OK, 1 row affected (0.02 sec)

mysql> use HMBank;
Database changed
mysql> -
```

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

```
MySQL 8.0 Command Line Client

mysql> create table Customers(customer_id int primary key,
-> first_name varchar(255),
-> last_name varchar(255),
-> DOB date,
-> email varchar(255),
-> phone_number varchar(20),
-> address text);
Query OK, 0 rows affected (0.12 sec)

mysql> desc Customers;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_id | int | NO | PRI | NULL | 
| first_name | varchar(255) | YES | | NULL | 
| last_name | varchar(255) | YES | | NULL | 
| DOB | date | YES | | NULL | 
| email | varchar(255) | YES | | NULL | 
| phone_number | varchar(20) | YES | | NULL | 
| address | text | YES | | NULL | 
+-----+-----+-----+-----+-----+
7 rows in set (0.04 sec)

mysql> CREATE TABLE Accounts (
->     account_id INT PRIMARY KEY,
->     customer_id INT,
->     account_type VARCHAR(50),
->     balance DECIMAL(15, 2),
->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
-> );
Query OK, 0 rows affected (0.08 sec)

mysql> desc Accounts;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_id | int | NO | PRI | NULL | 
| customer_id | int | YES | MUL | NULL | 
| account_type | varchar(50) | YES | | NULL | 
| balance | decimal(15,2) | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```

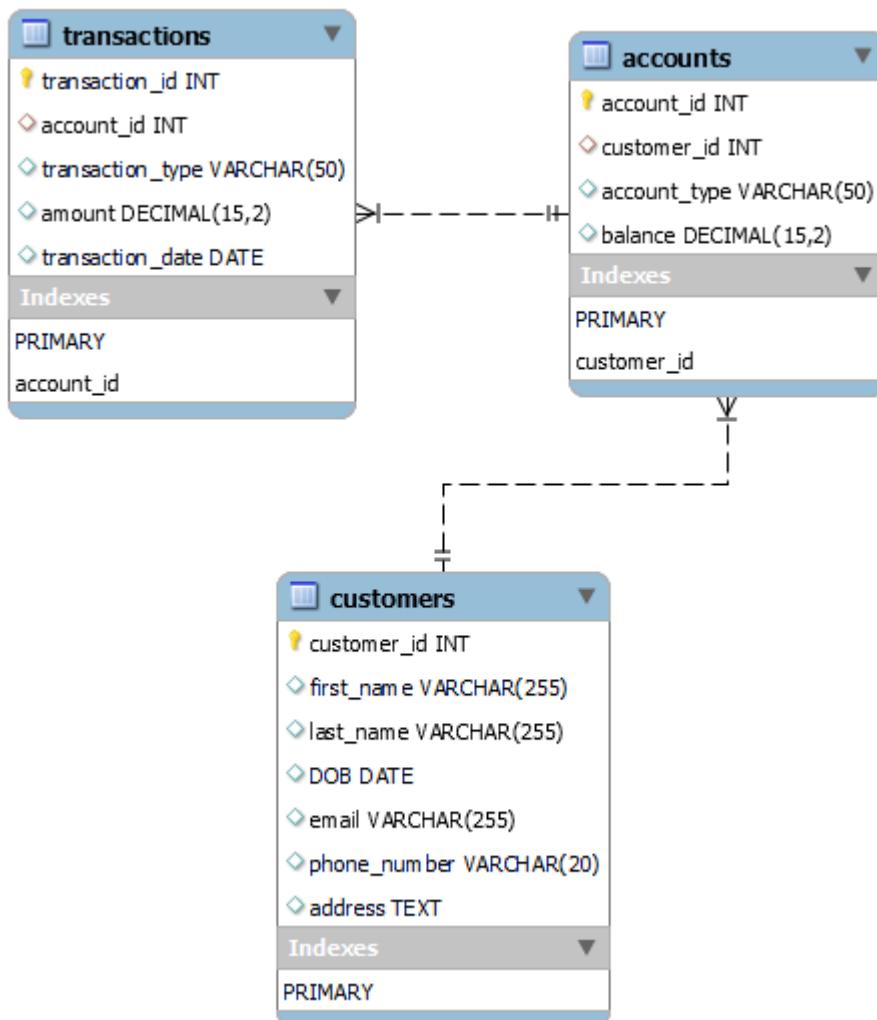
mysql> CREATE TABLE Transactions (
->     transaction_id INT PRIMARY KEY,
->     account_id INT,
->     transaction_type VARCHAR(50),
->     amount DECIMAL(15, 2),
->     transaction_date DATE,
->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
-> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc Transactions;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| transaction_id | int | NO | PRI | NULL |       |
| account_id | int | YES | MUL | NULL |       |
| transaction_type | varchar(50) | YES |       | NULL |       |
| amount | decimal(15,2) | YES |       | NULL |       |
| transaction_date | date | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> -

```

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



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

```
MySQL 8.0 Command Line Client
5 rows in set (0.00 sec)

mysql> desc Customers;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_id | int | NO | PRI | NULL | 
| first_name | varchar(255) | YES | NULL | 
| last_name | varchar(255) | YES | NULL | 
| DOB | date | YES | NULL | 
| email | varchar(255) | YES | NULL | 
| phone_number | varchar(20) | YES | NULL | 
| address | text | YES | NULL | 
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> desc Accounts;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_id | int | NO | PRI | NULL | 
| customer_id | int | YES | MUL | NULL | 
| account_type | varchar(50) | YES | NULL | 
| balance | decimal(15,2) | YES | NULL | 
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc Transactions;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| transaction_id | int | NO | PRI | NULL | 
| account_id | int | YES | MUL | NULL | 
| transaction_type | varchar(50) | YES | NULL | 
| amount | decimal(15,2) | YES | NULL | 
| transaction_date | date | YES | NULL | 
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

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

- Customers
- Accounts
- Transactions

```
[MySQL] MySQL 8.0 Command Line Client
```

```
5 rows in set (0.00 sec)
```

```
mysql> desc Customers;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
DOB	date	YES		NULL	
email	varchar(255)	YES		NULL	
phone_number	varchar(20)	YES		NULL	
address	text	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> desc Accounts;
```

Field	Type	Null	Key	Default	Extra
account_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
account_type	varchar(50)	YES		NULL	
balance	decimal(15,2)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> desc Transactions;
```

Field	Type	Null	Key	Default	Extra
transaction_id	int	NO	PRI	NULL	
account_id	int	YES	MUL	NULL	
transaction_type	varchar(50)	YES		NULL	
amount	decimal(15,2)	YES		NULL	
transaction_date	date	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql>
```

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

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

- Customers
- Accounts
- Transaction

MySQL 8.0 Command Line Client

```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number, address)
-> VALUES
-> (1, 'Astha', 'Raj', '1990-06-20', 'astha.raj@gmail.com', '7036987450', 'Motihari Bihar'),
-> (2, 'Asmita', 'Si', '1990-05-20', 'asmita.si@gmail.com', '9002500456', '748 salt lake'),
-> (3, 'Prateeti', 'Maji', '1990-05-25', 'prateeti.maji@gmail.com', '3960256025', '254 Durgapur'),
-> (4, 'Monu', 'Sharma', '1996-06-22', 'monu.sharma@gmail.com', '9098885502', '32 Rohini Delhi'),
-> (5, 'Alice', 'Doe', '2002-01-14', 'alice.doe@gmail.com', '9934567890', '123 lake St'),
-> (6, 'Bob', 'Smith', '1990-05-20', 'bob.smith@gmail.com', '9988552233', '456 Main St'),
-> (7, 'John', 'Smith', '1999-09-01', 'john.smith@gmail.com', '9988555533', '123 Main St'),
-> (8, 'Harry', 'Smith', '1999-09-19', 'harry.smith@gmail.com', '9900552233', '897 Main St'),
-> (9, 'julie', 'Anthony', '1998-12-11', 'julie.anthony@gmail.com', '9088552233', '55 Berlin'),
-> (10, 'Cooper', 'Ana', '2001-02-15', 'cooper.ana@gmail.com', '9988550233', '44 London');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from customers;
```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Astha	Raj	1990-06-20	astha.raj@gmail.com	7036987450	Motihari Bihar
2	Asmita	Si	1990-05-20	asmita.si@gmail.com	9002500456	748 salt lake
3	Prateeti	Maji	1990-05-25	prateeti.maji@gmail.com	3960256025	254 Durgapur
4	Monu	Sharma	1996-06-22	monu.sharma@gmail.com	9098885502	32 Rohini Delhi
5	Alice	Doe	2002-01-14	alice.doe@gmail.com	9934567890	123 lake St
6	Bob	Smith	1990-05-20	bob.smith@gmail.com	9988552233	456 Main St
7	John	Smith	1999-09-01	john.smith@gmail.com	9988555533	123 Main St
8	Harry	Smith	1999-09-19	harry.smith@gmail.com	9900552233	897 Main St
9	julie	Anthony	1998-12-11	julie.anthony@gmail.com	9088552233	55 Berlin
10	Cooper	Ana	2001-02-15	cooper.ana@gmail.com	9988550233	44 London

10 rows in set (0.00 sec)

```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
```

```
-> VALUES
-> (101, 1, 'savings', 5000.00),
-> (102, 2, 'current', 1500.00),
-> (103, 3, 'savings', 16000.00),
-> (104, 4, 'savings', 18000.00),
-> (105, 5, 'zero_balance', 0.00),
-> (106, 6, 'savings', 8000.00),
-> (107, 7, 'savings', 875000.00),
-> (108, 6, 'current', 15900.00),
-> (109, 8, 'savings', 80000.00),
-> (110, 9, 'current', 55500.00),
-> (111, 10, 'zero_balance', 15.00),
-> (112, 10, 'savings', 8990.00);
```

Query OK, 12 rows affected (0.01 sec)

Records: 12 Duplicates: 0 Warnings: 0

```
mysql> select * from Accounts;
```

account_id	customer_id	account_type	balance
101	1	savings	5000.00
102	2	current	1500.00
103	3	savings	16000.00
104	4	savings	18000.00
105	5	zero_balance	0.00
106	6	savings	8000.00
107	7	savings	875000.00
108	6	current	15900.00
109	8	savings	80000.00
110	9	current	55500.00
111	10	zero_balance	15.00
112	10	savings	8990.00

12 rows in set (0.00 sec)

```

mysql> INSERT INTO Transactions(transaction_id, account_id, transaction_type, amount, transaction_date)
-> VALUES
-> (1001, 101, 'deposit', 1000.00, '2022-01-10'),
-> (1002, 102, 'withdrawal', 500.00, '2022-01-15'),
-> (1003, 103, 'deposit', 2200.00, '2022-02-01'),
-> (1004, 104, 'deposit', 1100.00, '2022-03-05'),
-> (1005, 105, 'deposit', 8500.00, '2022-03-10'),
-> (1006, 106, 'deposit', 2000.00, '2022-06-01'),
-> (1007, 107, 'withdrawal', 1000.00, '2022-04-15'),
-> (1008, 106, 'transfer', 3000.00, '2022-05-01'),
-> (1009, 108, 'deposit', 20000.00, '2022-04-01'),
-> (1010, 109, 'withdrawal', 1000.00, '2023-04-18'),
-> (1011, 110, 'deposit', 30000.00, '2023-11-01'),
-> (1012, 111, 'withdrawal', 1000.00, '2024-01-15');
Query OK, 12 rows affected (0.00 sec)
Records: 12 Duplicates: 0 Warnings: 0

mysql> select * from Transactions;
+-----+-----+-----+-----+-----+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+-----+-----+-----+-----+-----+
| 1001 | 101 | deposit | 1000.00 | 2022-01-10 |
| 1002 | 102 | withdrawal | 500.00 | 2022-01-15 |
| 1003 | 103 | deposit | 2200.00 | 2022-02-01 |
| 1004 | 104 | deposit | 1100.00 | 2022-03-05 |
| 1005 | 105 | deposit | 8500.00 | 2022-03-10 |
| 1006 | 106 | deposit | 2000.00 | 2022-06-01 |
| 1007 | 107 | withdrawal | 1000.00 | 2022-04-15 |
| 1008 | 106 | transfer | 3000.00 | 2022-05-01 |
| 1009 | 108 | deposit | 20000.00 | 2022-04-01 |
| 1010 | 109 | withdrawal | 1000.00 | 2023-04-18 |
| 1011 | 110 | deposit | 30000.00 | 2023-11-01 |
| 1012 | 111 | withdrawal | 1000.00 | 2024-01-15 |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> -

```

(2). Write SQL queries for the following tasks:

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

```

MySQL 8.0 Command Line Client

mysql> SELECT
->     Customers.first_name AS FirstName,
->     Customers.last_name AS LastName,
->     Accounts.account_type AS AccountType,
->     Customers.email AS Email
-> FROM
->     Customers
-> JOIN
->     Accounts ON Customers.customer_id = Accounts.customer_id;
+-----+-----+-----+-----+
| FirstName | LastName | AccountType | Email           |
+-----+-----+-----+-----+
| Astha    | Raj      | savings    | astha.raj@gmail.com
| Asmita   | Si       | current    | asmita.si@gmail.com
| Prateeti | Maji     | savings    | prateeti.maji@gmail.com
| Monu    | Sharma   | savings    | monu.sharma@gmail.com
| Alice    | Doe      | zero_balance | alice.doe@gmail.com
| Bob     | Smith    | savings    | bob.smith@gmail.com
| Bob     | Smith    | current    | bob.smith@gmail.com
| John    | Smith    | savings    | john.smith@gmail.com
| Harry   | Smith    | savings    | harry.smith@gmail.com
| julie   | Anthony  | current    | julie.anthony@gmail.com
| Cooper  | Ana      | zero_balance | cooper.ana@gmail.com
| Cooper  | Ana      | savings    | cooper.ana@gmail.com
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> -

```

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

```

MySQL 8.0 Command Line Client
mysql> SELECT
->     Transactions.transaction_id AS TransactionID,
->     Customers.first_name AS FirstName,
->     Customers.last_name AS LastName,
->     Transactions.transaction_type AS TransactionType,
->     Transactions.amount AS Amount,
->     Transactions.transaction_date AS TransactionDate
-> FROM
->     Transactions
-> JOIN
->     Accounts ON Transactions.account_id = Accounts.account_id
-> JOIN
->     Customers ON Accounts.customer_id = Customers.customer_id;
+-----+-----+-----+-----+-----+-----+
| TransactionID | FirstName | LastName | TransactionType | Amount | TransactionDate |
+-----+-----+-----+-----+-----+-----+
| 1001 | Astha    | Raj      | deposit       | 1000.00 | 2022-01-10 |
| 1002 | Asmita   | Si       | withdrawal   | 500.00  | 2022-01-15 |
| 1003 | Prateeti  | Maji     | deposit       | 2200.00 | 2022-02-01 |
| 1004 | Monu     | Sharma   | deposit       | 1100.00 | 2022-03-05 |
| 1005 | Alice     | Doe      | deposit       | 8500.00 | 2022-03-10 |
| 1006 | Bob       | Smith    | deposit       | 2000.00 | 2022-06-01 |
| 1008 | Bob       | Smith    | transfer     | 3000.00 | 2022-05-01 |
| 1009 | Bob       | Smith    | deposit       | 20000.00 | 2022-04-01 |
| 1007 | John      | Smith    | withdrawal   | 1000.00 | 2022-04-15 |
| 1010 | Harry     | Smith    | withdrawal   | 1000.00 | 2023-04-18 |
| 1011 | julie    | Anthony  | deposit       | 30000.00 | 2023-11-01 |
| 1012 | Cooper   | Ana      | withdrawal   | 1000.00 | 2024-01-15 |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> -

```

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

```

mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE IncreaseBalance(IN AccountIDParam INT, IN AmountParam DECIMAL(15,2))
-> BEGIN
->
-> UPDATE Accounts
-> SET balance = balance + AmountParam
-> WHERE account_id = AccountIDParam;
-> END ##
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER ;
mysql> call IncreaseBalance(101, 1000.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Accounts;
+-----+-----+-----+-----+
| account_id | customer_id | account_type | balance |
+-----+-----+-----+-----+
| 101 | 1 | savings | 6000.00 |
| 102 | 2 | current | 1500.00 |
| 103 | 3 | savings | 16000.00 |
| 104 | 4 | savings | 18000.00 |
| 105 | 5 | zero_balance | 0.00 |
| 106 | 6 | savings | 8000.00 |
| 107 | 7 | savings | 875000.00 |
| 108 | 6 | current | 15900.00 |
| 109 | 8 | savings | 80000.00 |
| 110 | 9 | current | 55500.00 |
| 111 | 10 | zero_balance | 15.00 |
| 112 | 10 | savings | 8990.00 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>

```

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

```
mysql> select concat(first_name , ' ' , last_name)as full_name from Customers;
+-----+
| full_name |
+-----+
| Astha Raj
| Asmita Si
| Prateeti Maji
| Monu Sharma
| Alice Doe
| Bob Smith
| John Smith
| Harry Smith
| julie Anthony
| Cooper Ana
+-----+
10 rows in set (0.00 sec)

mysql>
```

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

```
mysql> DELETE FROM Accounts
      -> WHERE account_type = 'savings' AND balance = 0;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

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

```
mysql> DELETE FROM Accounts
      -> WHERE account_type = 'savings' AND balance = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
      ->     customer_id,
      ->     first_name,
      ->     last_name,
      ->     email,
      ->     phone_number,
      ->     address
      ->   FROM
      ->     Customers
      ->   WHERE
      ->     address LIKE '%Motihari%';
+-----+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | email           | phone_number | address        |
+-----+-----+-----+-----+-----+-----+
|         1 | Astha      | Raj       | astha.raj@gmail.com | 7036987450  | Motihari Bihar |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> ■
```

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

```
mysql> DELIMITER ;
mysql>
mysql> call GetBalance(101);
ERROR 1054 (42S22): Unknown column 'AccountIDParam' in 'where clause'
mysql> DELIMITER ##
mysql> CREATE PROCEDURE GetFullBalance(In AccountIDParam INT)
-> BEGIN
->   SELECT
->     account_id,account_type,balance
->   FROM
->     Accounts
->   WHERE
->     account_id = AccountIDParam;
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call GetBalance(101);
ERROR 1054 (42S22): Unknown column 'AccountIDParam' in 'where clause'
mysql> call GetFullBalance(101);
+-----+-----+-----+
| account_id | account_type | balance |
+-----+-----+-----+
|      101 | savings      | 6000.00 |
+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

mysql>
```

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

```
mysql> select account_id, account_type, balance
->   from Accounts
->   where account_type = 'current' and balance > 1000.00;
+-----+-----+-----+
| account_id | account_type | balance |
+-----+-----+-----+
|      102 | current      | 1500.00 |
|      108 | current      | 15900.00 |
|      110 | current      | 55500.00 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

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

```

MySQL 8.0 Command Line Client
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE GetTransaction(In AccountIDParam INT)
-> BEGIN
->
-> SELECT
->     transaction_id,
->     transaction_type,
->     amount,
->     transaction_date
-> FROM
->     Transactions
-> WHERE
->     account_id = AccountIDParam;
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call GetTransaction(101);
+-----+-----+-----+-----+
| transaction_id | transaction_type | amount | transaction_date |
+-----+-----+-----+-----+
|      1001 | deposit          | 1000.00 | 2022-01-10       |
+-----+-----+-----+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>

```

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

```

mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE CalculateInterest(In InterestParam DECIMAL(10,2))
-> BEGIN
->
-> SELECT
->     t.account_id,
->     a.account_type,
->     a.balance,
->     a.balance *InterestParam * DATEDIFF(NOW(), t.transaction_date) / 365 AS accrued_interest
-> FROM
->     Transactions t
-> JOIN
->     Accounts a ON t.account_id = a.account_id
-> WHERE
->     a.account_type = 'savings';
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call CalculateInterest(11.00);
+-----+-----+-----+-----+
| account_id | account_type | balance | accrued_interest |
+-----+-----+-----+-----+
|      101 | savings          | 6000.00 | 133808.21917808 |
|      103 | savings          | 16000.00 | 346213.69863014 |
|      104 | savings          | 18000.00 | 372131.50684932 |
|      106 | savings          | 8000.00  | 144175.34246575 |
|      106 | savings          | 8000.00  | 151649.31506849 |
|      107 | savings          | 875000.00 | 17008561.64383562 |
|      109 | savings          | 80000.00 | 667835.61643836 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

```

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

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE IdentifyAccounts(In OverDraftLimitParam DECIMAL(15,2))
-> BEGIN
->
-> SELECT
->     account_id,
->     account_type,
->     balance
-> FROM
->     Accounts
-> WHERE
->     balance < OverDraftLimitParam;
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
mysql> call IdentifyAccounts(100000.00);
+-----+-----+-----+
| account_id | account_type | balance |
+-----+-----+-----+
|      101 | savings      | 6000.00 |
|      102 | current      | 1500.00 |
|      103 | savings      | 16000.00 |
|      104 | savings      | 18000.00 |
|      105 | zero_balance |    0.00 |
|      106 | savings      | 8000.00 |
|      108 | current      | 15900.00 |
|      109 | savings      | 80000.00 |
|      110 | current      | 55500.00 |
|      111 | zero_balance |   15.00 |
|      112 | savings      | 8990.00 |
+-----+-----+-----+
11 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> _
```

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

```
mysql> SELECT
->     customer_id,
->     first_name,
->     last_name,
->     email,
->     phone_number,
->     address
-> FROM
->     Customers
-> WHERE
->     NOT address LIKE '%Main St%';
+-----+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | email           | phone_number | address        |
+-----+-----+-----+-----+-----+-----+
|      1 | Astha     | Raj       | astha.raj@gmail.com | 7036987450  | Motihari Bihar |
|      2 | Asmita    | Si        | asmita.si@gmail.com | 9002500456  | 748 salt lake  |
|      3 | Prateeti   | Maji      | prateeti.maji@gmail.com | 3960256025 | 254 Durgapur   |
|      4 | Monu       | Sharma    | monu.sharma@gmail.com | 9098855002  | 32 Rohini Delhi |
|      5 | Alice       | Doe       | alice.doe@gmail.com  | 9934567890  | 123 lake St    |
|      9 | julie      | Anthony   | julie.anthony@gmail.com | 9088552233 | 55 Berlin      |
|     10 | Cooper     | Ana       | cooper.ana@gmail.com | 9988550233  | 44 London      |
+-----+-----+-----+-----+-----+-----+
7 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.

```
MySQL 8.0 Command Line Client
mysql> select customer_id, AVG(balance) as Average_Balance
-> from Accounts
-> group by customer_id;
+-----+-----+
| customer_id | Average_Balance |
+-----+-----+
|      1 |       6000.000000 |
|      2 |       1500.000000 |
|      3 |       16000.000000 |
|      4 |       18000.000000 |
|      5 |          0.000000 |
|      6 |       11950.000000 |
|      7 |       875000.000000 |
|      8 |       80000.000000 |
|      9 |       55500.000000 |
|     10 |       4502.500000 |
+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

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

```
mysql> SELECT
->     account_id,
->     account_type,
->     balance
-> FROM
->     Accounts
-> ORDER BY
->     balance DESC
-> LIMIT 10;
+-----+-----+-----+
| account_id | account_type | balance |
+-----+-----+-----+
|      107 |   savings | 875000.00 |
|      109 |   savings | 80000.00  |
|      110 |   current | 55500.00  |
|      104 |   savings | 18000.00  |
|      103 |   savings | 16000.00  |
|      108 |   current | 15900.00  |
|      112 |   savings | 8990.00   |
|      106 |   savings | 8000.00   |
|      101 |   savings | 6000.00   |
|      102 |   current | 1500.00   |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> -
```

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

MySQL 8.0 Command Line Client

```
mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE CalTotalDeposit(In DateParam DATE)
-> BEGIN
->
-> SELECT
->     c.customer_id,c.first_name, c.last_name,
->     SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE 0 END) AS total_deposits
-> FROM
->     Customers c
-> JOIN
->     Accounts a ON c.customer_id = a.customer_id
-> JOIN
->     Transactions t ON a.account_id = t.account_id
-> WHERE
->     t.transaction_date = DateParam
-> GROUP BY
->     c.customer_id, c.first_name, c.last_name;
-> END ##
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
```

```
mysql> call CalTotalDeposit('2022-02-01');
+-----+-----+-----+-----+
| customer_id | first_name | last_name | total_deposits |
+-----+-----+-----+-----+
|         3 | Prateeti   | Maji      |      2200.00 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> -
```

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

```

MySQL 8.0 Command Line Client
mysql> -- Oldest Customer
mysql> SELECT
->     customer_id,first_name,last_name,DOB
->   FROM
->     Customers
-> ORDER BY
->     DOB ASC
-> LIMIT 1;
+-----+-----+-----+
| customer_id | first_name | last_name | DOB      |
+-----+-----+-----+
|          2   | Asmita    | Si        | 1990-05-20 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
mysql> -- Newest Customer
mysql> SELECT
->     customer_id,first_name,last_name,DOB
->   FROM
->     Customers
-> ORDER BY
->     DOB DESC
-> LIMIT 1;
+-----+-----+-----+
| customer_id | first_name | last_name | DOB      |
+-----+-----+-----+
|          5   | Alice     | Doe       | 2002-01-14 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> -

```

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

```

MySQL 8.0 Command Line Client
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT
->     t.transaction_id,t.account_id,a.account_type,t.transaction_type,
->     t.amount,t.transaction_date
->   FROM
->     Transactions t
-> JOIN
->     Accounts a ON t.account_id = a.account_id;
+-----+-----+-----+-----+-----+-----+
| transaction_id | account_id | account_type | transaction_type | amount   | transaction_date |
+-----+-----+-----+-----+-----+-----+
|      1001 |      101 | savings      | deposit        | 1000.00 | 2022-01-10 |
|      1002 |      102 | current      | withdrawal    | 500.00  | 2022-01-15 |
|      1003 |      103 | savings      | deposit        | 2200.00 | 2022-02-01 |
|      1004 |      104 | savings      | deposit        | 1100.00 | 2022-03-05 |
|      1005 |      105 | zero_balance | deposit        | 8500.00 | 2022-03-10 |
|      1006 |      106 | savings      | deposit        | 2000.00 | 2022-06-01 |
|      1007 |      107 | savings      | withdrawal    | 1000.00 | 2022-04-15 |
|      1008 |      106 | savings      | transfer      | 3000.00 | 2022-05-01 |
|      1009 |      108 | current      | deposit        | 20000.00 | 2022-04-01 |
|      1010 |      109 | savings      | withdrawal    | 1000.00 | 2023-04-18 |
|      1011 |      110 | current      | deposit        | 30000.00 | 2023-11-01 |
|      1012 |      111 | zero_balance | withdrawal    | 1000.00 | 2024-01-15 |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

```

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

```

MySQL 8.0 Command Line Client
+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | DOB | email | phone_number | address | account_id | account_type | balance |
+-----+-----+-----+-----+-----+
| 1 | Astha | Raj | 1990-06-20 | astha.raj@gmail.com | 7036987450 | Motihari Bihar | 101 | savings | 6000.00 |
| 2 | Asmita | Si | 1990-05-20 | asmita.si@gmail.com | 9002500456 | 748 salt lake | 102 | current | 1500.00 |
| 3 | Prateeti | Maji | 1990-05-25 | prateeti.maji@gmail.com | 3960256025 | 254 Durgapur | 103 | savings | 16000.00 |
| 4 | Monu | Sharma | 1996-06-22 | monu.sharma@gmail.com | 9098885502 | 32 Rohini Delhi | 104 | savings | 18000.00 |
| 5 | Alice | Doe | 2002-01-14 | alice.doe@gmail.com | 9934567890 | 123 lake St | 105 | zero_balance | 0.00 |
| 6 | Bob | Smith | 1990-05-20 | bob.smith@gmail.com | 9988552233 | 456 Main St | 106 | savings | 8000.00 |
| 6 | Bob | Smith | 1990-05-20 | bob.smith@gmail.com | 9988552233 | 456 Main St | 108 | current | 15900.00 |
| 7 | John | Smith | 1999-09-01 | john.smith@gmail.com | 9988555533 | 123 Main St | 107 | savings | 875000.00 |
| 8 | Harry | Smith | 1999-09-19 | harry.smith@gmail.com | 9980552233 | 897 Main St | 109 | savings | 80000.00 |
| 9 | Julie | Anthony | 1998-12-11 | julie.anthony@gmail.com | 9088552233 | 55 Berlin | 110 | current | 55500.00 |
| 10 | Cooper | Ana | 2001-02-15 | cooper.ana@gmail.com | 9988550233 | 44 London | 111 | zero_balance | 15.00 |
| 10 | Cooper | Ana | 2001-02-15 | cooper.ana@gmail.com | 9988550233 | 44 London | 112 | savings | 8990.00 |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> -

```

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

```

MySQL 8.0 Command Line Client
mysql> DELIMITER ##
mysql> CREATE PROCEDURE TransactionDetails(In AccountIDParam INT)
-> BEGIN
-> 
-> SELECT
->     c.customer_id,c.first_name,c.last_name,c.DOB, c.email,c.phone_number,c.address,
->     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
-> WHERE
->     a.account_id = AccountIDParam;
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call TransactionDetails(104);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | DOB | email | phone_number | address | transaction_id | transaction_type | amount | transaction_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | Monu | Sharma | 1996-06-22 | monu.sharma@gmail.com | 9098885502 | 32 Rohini Delhi | 1004 | deposit | 1100.00 | 2022-03-05 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> -

```

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

```

mysql> SELECT
->     c.customer_id,c.first_name,c.last_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;
+-----+-----+-----+
| customer_id | first_name | last_name | account_count |
+-----+-----+-----+
| 6 | Bob | Smith | 2 |
| 10 | Cooper | Ana | 2 |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT
->     account_id,
->     SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) AS total_deposits,
->     SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS total_withdrawals,
->     SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END) AS net_difference
-> FROM
->     Transactions
-> GROUP BY
->     account_id;
+-----+-----+-----+-----+
| account_id | total_deposits | total_withdrawals | net_difference |
+-----+-----+-----+-----+
|    101 |      1000.00 |          0.00 |      1000.00 |
|    102 |        0.00 |       500.00 |      -500.00 |
|    103 |      2200.00 |          0.00 |      2200.00 |
|    104 |      1100.00 |          0.00 |      1100.00 |
|    105 |      8500.00 |          0.00 |      8500.00 |
|    106 |      2000.00 |          0.00 |      -1000.00 |
|    107 |        0.00 |      1000.00 |      -1000.00 |
|    108 |     20000.00 |          0.00 |     20000.00 |
|    109 |        0.00 |      1000.00 |      -1000.00 |
|    110 |     30000.00 |          0.00 |     30000.00 |
|    111 |        0.00 |      1000.00 |      -1000.00 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

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

```
mysql> DELIMITER ##
mysql>
mysql> CREATE PROCEDURE CalAvgDailyBalance(IN StartDateParam DATE, IN EndDateParam DATE)
-> BEGIN
->
->     SELECT
->         t.account_id, AVG(a.balance) AS average_daily_balance
->     FROM
->         Transactions t
->     JOIN
->         Accounts a ON t.account_id = a.account_id
->     WHERE
->         t.transaction_date BETWEEN StartDateParam AND EndDateParam
->     GROUP BY
->         t.account_id, DATE(t.transaction_date);
-> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call CalAvgDailyBalance('2022-05-01', '2023-12-12');
+-----+-----+
| account_id | average_daily_balance |
+-----+-----+
|      106 |      8000.000000 |
|      106 |      8000.000000 |
|      109 |      80000.000000 |
|     110 |      55500.000000 |
+-----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

11. Calculate the total balance for each account type.

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT
->     account_type,
->     SUM(balance) AS total_balance
-> FROM
->     Accounts
-> GROUP BY
->     account_type;
+-----+-----+
| account_type | total_balance |
+-----+-----+
| savings      | 1011990.00 |
| current      | 72900.00  |
| zero_balance | 15.00    |
+-----+-----+
3 rows in set (0.00 sec)
```

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

```
MySQL 8.0 Command Line Client
mysql>
mysql> SELECT
->     t.account_id,
->     COUNT(t.transaction_id) AS transaction_count
-> FROM
->     Transactions t
-> GROUP BY
->     t.account_id
-> ORDER BY
->     transaction_count DESC;
+-----+-----+
| account_id | transaction_count |
+-----+-----+
| 106        | 2              |
| 101        | 1              |
| 102        | 1              |
| 103        | 1              |
| 104        | 1              |
| 105        | 1              |
| 107        | 1              |
| 108        | 1              |
| 109        | 1              |
| 110        | 1              |
| 111        | 1              |
+-----+-----+
11 rows in set (0.00 sec)

mysql> ■
```

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

```
MySQL 8.0 Command Line Client
mysql> SELECT
->     c.customer_id, 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, c.first_name, c.last_name, a.account_type
-> HAVING
->     SUM(a.balance) > 55000;
+-----+-----+-----+-----+
| customer_id | first_name | last_name | account_type | total_balance |
+-----+-----+-----+-----+
| 7           | John       | Smith     | savings      | 875000.00   |
| 8           | Harry      | Smith     | savings      | 80000.00    |
| 9           | julie      | Anthony   | current      | 55500.00   |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT
->     t1.transaction_id, t1.account_id, t1.amount, t1.transaction_date
->   FROM
->     Transactions t1
->   JOIN
->     Transactions t2 ON t1.account_id = t2.account_id
->                   AND t1.amount = t2.amount
->                   AND t1.transaction_date = t2.transaction_date
->                   AND t1.transaction_id <> t2.transaction_id;
Empty set (0.00 sec)

mysql>
```

Tasks 4: Subquery and its type:

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

```
MySQL 8.0 Command Line Client

mysql> SELECT
->     customer_id,first_name,last_name,email,phone_number
->   FROM
->     Customers
-> WHERE
->     customer_id = (
->       SELECT customer_id
->         FROM Accounts
->           ORDER BY balance DESC LIMIT 1);
+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | email          | phone_number |
+-----+-----+-----+-----+-----+
|      7       | John       | Smith     | john.smith@gmail.com | 9988555533   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT AVG(avg_balance) AS average_account_balance
->   FROM ( SELECT c.customer_id, AVG(a.balance) AS avg_balance FROM Customers c
->   JOIN Accounts a ON c.customer_id = a.customer_id
->   GROUP BY c.customer_id
->   HAVING COUNT(a.account_id) > 1) AS customer_with_multiple_accounts;
+-----+
| average_account_balance |
+-----+
|      8226.25000000000  |
+-----+
1 row in set (0.00 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT a.account_id, a.account_type, t.transaction_id, t.amount,t.transaction_date
   -> FROM Accounts a
   -> JOIN Transactions t ON a.account_id = t.account_id
   -> WHERE t.amount > (SELECT AVG(amount) FROM Transactions);
+-----+-----+-----+-----+
| account_id | account_type | transaction_id | amount      | transaction_date |
+-----+-----+-----+-----+
|      105 | zero_balance |          1005 | 8500.00    | 2022-03-10   |
|      108 | current      |          1009 | 20000.00   | 2022-04-01   |
|     110 | current      |          1011 | 30000.00   | 2023-11-01   |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> -
```

4. Identify customers who have no recorded transactions.

```
MySQL 8.0 Command Line Client

3 rows in set (0.00 sec)

mysql> SELECT customer_id, first_name, last_name, email, phone_number
   -> FROM Customers
   -> WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Transactions);
Empty set (0.00 sec)

mysql> -
```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT SUM(balance) AS total_balance
   -> FROM Accounts
   -> WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);
+-----+
| total_balance |
+-----+
|      8990.00 |
+-----+
1 row in set (0.00 sec)

mysql> -
```

6. Retrieve transactions for accounts with the lowest balance.

```
MySQL 8.0 Command Line Client

1 row in set (0.00 sec)

mysql> SELECT t.transaction_id, t.account_id, t.transaction_type, t.amount, t.transaction_date
   -> FROM Transactions t
   -> JOIN Accounts a ON t.account_id = a.account_id
   -> WHERE a.balance = ( SELECT MIN(balance) FROM Accounts);
+-----+-----+-----+-----+
| transaction_id | account_id | transaction_type | amount      | transaction_date |
+-----+-----+-----+-----+
|         1005 |       105 | deposit        | 8500.00    | 2022-03-10   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

7. Identify customers who have accounts of multiple types.

```
MySQL 8.0 Command Line Client

mysql> SELECT c.customer_id,c.first_name,c.last_name,c.email,c.phone_number
-> FROM Customers c
-> WHERE c.customer_id IN (SELECT customer_id FROM Accounts GROUP BY customer_id
-> HAVING COUNT(DISTINCT account_type) > 1);
+-----+-----+-----+-----+
| customer_id | first_name | last_name | email | phone_number |
+-----+-----+-----+-----+
|       6 | Bob        | Smith      | bob.smith@gmail.com | 9988552233 |
|      10 | Cooper     | Ana        | cooper.ana@gmail.com | 9988550233 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client

mysql> SELECT
->     account_type,
->     COUNT(*) AS account_count,
->     (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS percentage
->   FROM Accounts GROUP BY account_type;
+-----+-----+-----+
| account_type | account_count | percentage |
+-----+-----+-----+
| savings      |         7 | 58.33333 |
| current      |         3 | 25.00000 |
| zero_balance |         2 | 16.66667 |
+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client

mysql> DELIMITER ;
mysql> call GetAllTransaction(101);
ERROR 1054 (42S22): Unknown column 'CustomerIDParam' in 'where clause'
mysql> call GetAllTransactions(101);
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call GetAllTransactions(104);
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call GetAllTransactions(4);
+-----+-----+-----+-----+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+-----+-----+-----+-----+
|       1004 |       104 | deposit          | 1100.00 | 2022-03-05 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

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

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT account_type, (SELECT SUM(balance) FROM Accounts a2 WHERE a2.account_type = a1.account_type) AS total_balance
-> FROM (SELECT DISTINCT account_type FROM Accounts) a1;
+-----+-----+
| account_type | total_balance |
+-----+-----+
| savings      |    1011990.00 |
| current      |     72900.00 |
| zero_balance |       15.00 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> -
```