Assignment - 3

An advanced banking system that includes various types of accounts, such as savings and current accounts. The system should support account creation, deposits, withdrawals, and interest calculations.

TASK-1 DATABASE DESIGN

1) Create a database named “HMBank”.

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

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

Customers

```
mysql> CREATE TABLE Customers(
    -> customer_id INT PRIMARY KEY,
    -> first_name VARCHAR(25),
    ->  last_name VARCHAR(25),
    -> DOB DATE,
    -> email VARCHAR(50),
    ->  phone_number BIGINT,
    ->  address TEXT
    -> );
Query OK, 0 rows affected (0.09 sec)
```

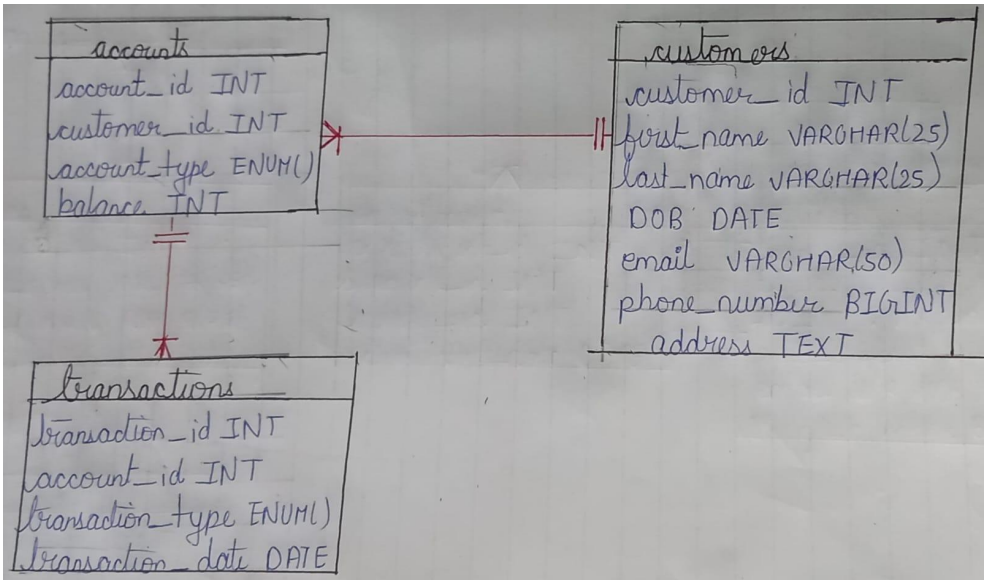
Accounts

```
mysql> CREATE TABLE Accounts(
    -> account_id INT PRIMARY KEY,
    -> customer_id INT,
    -> account_type  ENUM('savings','current','zero_balance'),
    -> balance INT,
    -> FOREIGN KEY(customer_id) REFERENCES customers(customer_id)
    -> );
Query OK, 0 rows affected (0.09 sec)
```

Transactions

```
ysql>  CREATE TABLE Transactions(
    -> transaction_id INT PRIMARY KEY,
    -> account_id INT,
    -> transcation_type ENUM('deposit','withdrawal','transfer'),
    -> amount INT,
    -> transaction_date DATE,
    -> FOREIGN KEY(account_id) REFERENCES Accounts(account_id)
    -> );
uery OK, 0 rows affected (0.09 sec)
```

4) Create an E-R(entity relationship diagram) for the database .



5) Create appropriate Primary key and Foreign key constraint for referential integrity.

Primary key

```
+--------------+------------------+
| TABLE_NAME   | CONSTRAINT_NAME  |
+--------------+------------------+
| customers    | PRIMARY          |
| accounts     | PRIMARY          |
| transactions | PRIMARY          |
+--------------+------------------+
```

Foreign key

```
+--------------+----------------------+
| TABLE_NAME   | CONSTRAINT_NAME      |
+--------------+----------------------+
| accounts     | accounts_ibfk_1      |
| transactions | transactions_ibfk_1  |
+--------------+----------------------+
```

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

Accounts

Field	Type	Null	Key	Default
account_id	int	NO	PRI	NULL
customer_id	int	YES	MUL	NULL
account_type	enum('savings','current','zero_balance')	YES		NULL
balance	int	YES		NULL

Customers

Field	Type	Null	Key	Default
customer_id	int	NO	PRI	NULL
first_name	varchar(25)	YES		NULL
last_name	varchar(25)	YES		NULL
DOB	date	YES		NULL
email	varchar(50)	YES		NULL
phone_number	bigint	YES		NULL
address	text	YES		NULL

Transactions

Field	Type	Null	Key	Default
transaction_id	int	NO	PRI	NULL
account_id	int	YES	MUL	NULL
transaction_type	enum('deposit','withdrawal','transfer')	YES		NULL
amount	int	YES		NULL
transaction_date	date	YES		NULL

TASK-2 SELECT , WHERE , BETWEEN , AND , LIKE

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

Customers

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Alice	Smith	1995-08-20	alice@example.com	9234567890	1234 Elm Street
2	Bob	Johnson	1988-04-10	bob@example.com	9876543210	5678 Oak Avenue
3	Emily	Brown	1992-12-05	emily@example.com	5554443333	910 Cedar Road
4	Daniel	Miller	1985-06-25	daniel@example.com	1112223333	6789 Pine Lane
5	Sophia	Davis	1998-09-14	sophia@example.com	9998887777	3210 Maple Street
6	Michael	Wilson	1990-02-28	michael@example.com	7776665555	4567 Birch Avenue
7	Olivia	Martinez	1994-07-03	olivia@example.com	2223334444	7890 Ash Street
8	William	Taylor	1987-11-19	william@example.com	4445556666	2345 Pinecrest Drive
9	Ava	Anderson	1996-03-22	ava@example.com	8889991111	8765 Cedar Lane
10	James	Garcia	1984-10-08	james@example.com	6667778888	5432 Elm Avenue

Accounts

account_id	customer_id	account_type	balance
101	1	savings	5000
102	2	current	10000
103	3	savings	7500
104	4	zero_balance	0
105	5	current	5500
106	6	savings	12000
107	7	current	9300
108	8	zero_balance	0
109	9	current	2000
110	10	savings	9000

Transactions

transaction_id	account_id	transaction_type	amount	transaction_date
501	101	deposit	1000	2023-01-15
502	101	withdrawal	500	2023-02-03
503	102	deposit	2000	2023-02-10
504	103	transfer	1500	2023-03-20
505	104	withdrawal	800	2023-04-05
506	102	transfer	3000	2023-04-18
507	103	deposit	1200	2023-05-01
508	104	withdrawal	1000	2023-06-10
509	105	deposit	500	2023-07-15
600	105	withdrawal	200	2023-07-25

2) Write SQL query for the following tasks

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

```
mysql> update accounts set balance  = balance+50 where customer_id  = 5;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

2) Write a SQL query to combine the first and last name of customers as full_name.

full_name
Alice Smith
Bob Johnson
Emily Brown
Daniel Miller
Sophia Davis
Michael Wilson
Olivia Martinez
William Taylor
Ava Anderson
James Garcia

3) Write a SQl query to remove accounts with a balance of 0 where account type is saving

```
mysql> delete from accounts where account_type  = 'savings' and balance  = 0;
Query OK, 0 rows affected (0.00 sec)
```

4) Write a SQL query to find customers living in a specific city.

first_name	last_name	address
Daniel	Miller	6789 Pine Lane

5) Write a SQL query to get the account balance for a specific account.

balance
9000

6) Write a SQl query to List all current accounts with a balance greater than 1000.

account_id	customer_id	account_type	balance
101	1	savings	5000
102	2	current	10000
103	3	savings	7500
105	5	current	5550
106	6	savings	12000
107	7	current	9300
109	9	current	2000
110	10	savings	9000

7) Write a SQL query to retrieve all transactions for a specific account.

account_id	customer_id	account_type	balance
101	1	savings	5000
102	2	current	10000
103	3	savings	7500
105	5	current	5550
106	6	savings	12000
107	7	current	9300
109	9	current	2000
110	10	savings	9000

8) Write a SQL query to list all transaction corresponding customer.

concat(c.first_name,' ',c.last_name)	transaction_type	amount	transaction_date
Alice Smith	deposit	1000	2023-01-15
Alice Smith	withdrawal	500	2023-02-03
Bob Johnson	deposit	2000	2023-02-10
Emily Brown	transfer	1500	2023-03-20
Daniel Miller	withdrawal	800	2023-04-05
Bob Johnson	transfer	3000	2023-04-18
Emily Brown	deposit	1200	2023-05-01
Daniel Miller	withdrawal	1000	2023-06-10
Sophia Davis	deposit	500	2023-07-15
Sophia Davis	withdrawal	200	2023-07-25

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

first_name	last_name	account_type	email
Alice	Smith	savings	alice@example.com
Bob	Johnson	current	bob@example.com
Emily	Brown	savings	emily@example.com
Daniel	Miller	zero_balance	daniel@example.com
Sophia	Davis	current	sophia@example.com
Michael	Wilson	savings	michael@example.com
Olivia	Martinez	current	olivia@example.com
William	Taylor	zero_balance	william@example.com
Ava	Anderson	current	ava@example.com
James	Garcia	savings	james@example.com

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

Interest rate = 3%

account_id	initial_balance	accrued_interest
101	5000	150.00
103	7500	225.00
106	12000	360.00
110	9000	270.00

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

`1000` is considered as the assumed overdraft limit

account_id	initial_balance	accrued_interest
101	5000	150.00
103	7500	225.00
106	12000	360.00
110	9000	270.00

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

City = "New York"

customer_id	first_name	last_name	address
1	Alice	Smith	1234 Elm Street
2	Bob	Johnson	5678 Oak Avenue
3	Emily	Brown	910 Cedar Road
4	Daniel	Miller	6789 Pine Lane
5	Sophia	Davis	3210 Maple Street
6	Michael	Wilson	4567 Birch Avenue
7	Olivia	Martinez	7890 Ash Street
8	William	Taylor	2345 Pinecrest Drive
9	Ava	Anderson	8765 Cedar Lane
10	James	Garcia	5432 Elm Avenue

Task 3. AGGREGATE FUNCTIONS, HAVING, ORDER BY, GROUP BY and JOINS:

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

average_balance
6035.0000

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

transaction_id	account_id	transaction_type	amount	transaction_date
506	102	transfer	3000	2023-04-18
503	102	deposit	2000	2023-02-10
504	103	transfer	1500	2023-03-20

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

Date = 2023-04-18

total_deposits
2000

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

first_name	last_name	date_of_birth	customer_type
James	Garcia	1984-10-08	Oldest
Sophia	Davis	1998-09-14	Newest

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

transaction_id	account_type	transaction_type	amount	transaction_date
501	savings	deposit	1000	2023-01-15
502	savings	withdrawal	500	2023-02-03
503	current	deposit	2000	2023-02-10
504	savings	transfer	1500	2023-03-20
505	zero_balance	withdrawal	800	2023-04-05
506	current	transfer	3000	2023-04-18
507	savings	deposit	1200	2023-05-01
508	zero_balance	withdrawal	1000	2023-06-10
509	current	deposit	500	2023-07-15
600	current	withdrawal	200	2023-07-25

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

customer_id	first_name	last_name	email	phone_number	address	account_id	account_type	balance
1	Alice	Smith	alice@example.com	9234567890	1234 Elm Street	101	savings	500
2	Bob	Johnson	bob@example.com	9876543210	5678 Oak Avenue	102	current	1000
3	Emily	Brown	emily@example.com	5554443333	910 Cedar Road	103	savings	700
4	Daniel	Miller	daniel@example.com	1112223333	6789 Pine Lane	104	zero_balance	0
5	Sophia	Davis	sophia@example.com	9998887777	3210 Maple Street	105	current	550
6	Michael	Wilson	michael@example.com	7776665555	4567 Birch Avenue	106	savings	1200
7	Olivia	Martinez	olivia@example.com	2223334444	7890 Ash Street	107	current	900
8	William	Taylor	william@example.com	4445556666	2345 Pinecrest Drive	108	zero_balance	0
9	Ava	Anderson	ava@example.com	8889991111	8765 Cedar Lane	109	current	200
10	James	Garcia	james@example.com	6667778888	5432 Elm Avenue	110	savings	900

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

name	email	phone	id	type	amount	date
Sophia Davis	sophia@example.com	9998887777	509	deposit	500	2023-07-15
Sophia Davis	sophia@example.com	9998887777	600	withdrawal	200	2023-07-25

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

```
ysql> select customer_id,COUNT(account_id) AS accounts from  Accounts group by customer_id having COUNT(account_id) > 1;
Empty set (0.00 sec)
```

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

difference
2200

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

account_id	average_daily_balance
101	5000.0000
101	5000.0000
102	10000.0000
103	7500.0000

11) Calculate the total balance for each account type.

account_type	total_balance
savings	33500
current	26850
zero_balance	0

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

account_id	num_transactions
101	2
102	2
103	2
104	2
105	2

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

customer_id	first_name	last_name	account_type	total_balance
6	Michael	Wilson	savings	12000
2	Bob	Johnson	current	10000
7	Olivia	Martinez	current	9300
10	James	Garcia	savings	9000
3	Emily	Brown	savings	7500
5	Sophia	Davis	current	5550
1	Alice	Smith	savings	5000
9	Ava	Anderson	current	2000
4	Daniel	Miller	zero_balance	0
8	William	Taylor	zero_balance	0

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

```
mysql>  SELECT account_id, amount,transaction_date,COUNT(*) AS duplicate_count FROM Transactions GROUP BY account_id, amount, transact
ion_date HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

TASK -4 SUBQUERY AND IT'S TYPES

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

customer_id	first_name	last_name	max_balance
6	Michael	Wilson	12000

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

```
mysql> SELECT c.customer_id,AVG(a.balance) AS average_balance FROM Customers c  JOIN Accounts a ON c.customer_id = a.customer_id WHERE
 c.customer_id IN ( SELECT customer_id FROM Accounts GROUP BY customer_id HAVING COUNT(account_id) > 1) GROUP BY c.customer_id;
Empty set (0.00 sec)
```

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

account_id	transaction_id	amount	avg_transaction_amount
102	503	2000	1170.0000
103	504	1500	1170.0000
102	506	3000	1170.0000
103	507	1200	1170.0000

4) Identify customers who have no recorded transactions.

customer_id	first_name	last_name
6	Michael	Wilson
7	Olivia	Martinez
8	William	Taylor
9	Ava	Anderson
10	James	Garcia

5) Calculate the total balance of accounts with no recorded transaction.

total_balance_no_transactions
32300

6) Retrieve transactions for accounts with the lowest balance.

transaction_id	account_id	transaction_type	amount	transaction_date
505	104	withdrawal	800	2023-04-05
508	104	withdrawal	1000	2023-06-10

7) Identify customers who have accounts of multiple types.

```
mysql> SELECT c.customer_id,c.first_name,c.last_name FROM Customers c JOIN (SELECT customer_id FROM Accounts GROUP BY customer_id HAVI
NG COUNT(DISTINCT account_type) > 1) multi_type_accounts ON c.customer_id = multi_type_accounts.customer_id;
Empty set (0.00 sec)
```

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

account_type	num_accounts	percentage
savings	4	40.00
current	4	40.00
zero_balance	2	20.00

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

transaction_type	transaction_date	account_id	account_type	amount
withdrawal	2023-04-05	104	zero_balance	800
withdrawal	2023-06-10	104	zero_balance	1000

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

account_type	total_balance
savings	33500
current	26850
zero_balance	0