Assignment 3 (Banking System) By pradum

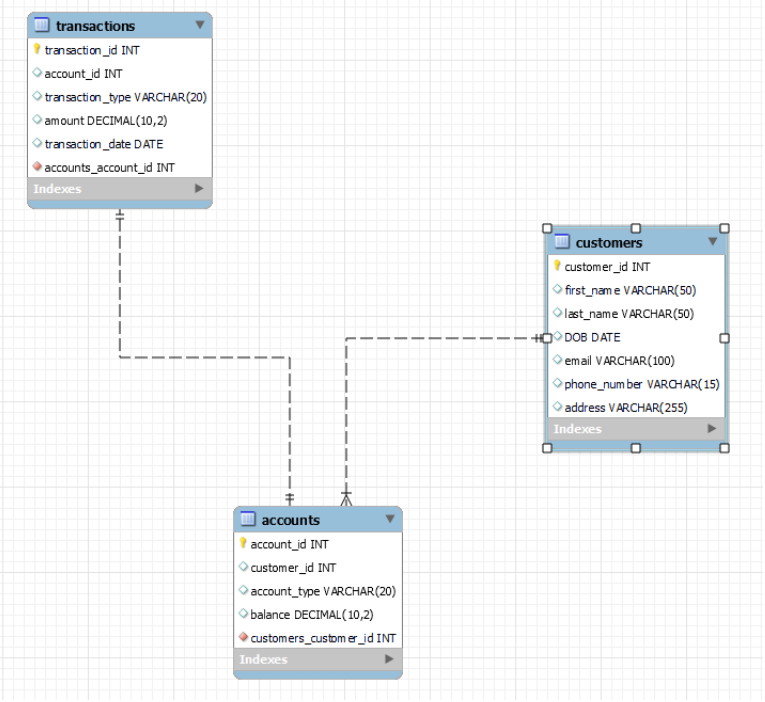
- ***Task 1 : DATABASE DESIGN***

1. *Create the database named "HMBank"*

   a.
```
create database HMbank;
use  HMbank;
```

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

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



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

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

```
CREATE TABLE Customers (
customer_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
DOB DATE,
email VARCHAR(100),
phone_number VARCHAR(15),
address VARCHAR(255)
);
```

- *Accounts*

```
CREATE TABLE Accounts (
account_id INT PRIMARY KEY,
customer_id INT REFERENCES Customers(customer_id),
account_type VARCHAR(20),
balance DECIMAL(10, 2),
CHECK (balance >= 0) -- Ensure balance is non-negative
);
```

- *Transactions*

```
CREATE TABLE Transactions (
transaction_id INT PRIMARY KEY,
account_id INT REFERENCES Accounts(account_id),
transaction_type VARCHAR(20),
amount DECIMAL(10, 2),
transaction_date DATE
);
```

- ***Task 2 : SELECT, WHERE BETWEEN AND LIKE:***

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

   a. *Customers*

i.
```
INSERT INTO Customers (customer_id, first_name, last_name, DOB, email,
phone_number, address)
VALUES
(1, 'Clark', 'Kent', '1978-05-03', 'clark.kent@email.com', '1234567890',
'344 Daily Planet'),
(2, 'Diana', 'Prince', '1985-01-30', 'diana.prince@email.com',
'9876543210', 'Themyscira'),
(3, 'Bruce', 'Wayne', '1972-11-15', 'bruce.wayne@email.com',
'5555555555', '1007 Wayne Manor'),
(4, 'Selina', 'Kyle', '1982-08-17', 'selina.kyle@email.com',
'1111111111', '22 Cat Street'),
(5, 'Tony', 'Stark', '1970-06-12', 'tony.stark@email.com', '9999999999',
'10880 Malibu Point'),
(6, 'Natasha', 'Romanoff', '1984-04-20', 'natasha.romanoff@email.com',
'4444444444', '64 Red Room Lane'),
(7, 'Wade', 'Wilson', '1976-02-22', 'wade.wilson@email.com',
'7777777777', '42 Mercenary Lane'),
(8, 'Peter', 'Parker', '1992-08-10', 'peter.parker@email.com',
'3333333333', '20 Ingram Street'),
(9, 'Barry', 'Allen', '1989-03-12', 'barry.allen@email.com',
'6666666666', '123 Speedster Lane'),
(10, 'Carol', 'Danvers', '1980-07-05', 'carol.danvers@email.com',
'8888888888', '44 Air Force Base');
```

customerID	firstname	lastname	email	phone	address
0	bruce	wayne	gothamsirens@gmail.com	111222333	23 - gotham street
3	clark	kent	clark@dailyplanet.com	67847343	32 florida farm
4	Bob	Miller	bob.miller@email.com	444-555-6666	101 Elm St
5	Eva	Brown	eva.brown@email.com	777-888-9999	555 Cedar St
6	David	Williams	david.williams@email.com	333-444-5555	222 Maple St
7	Grace	Jones	grace.jones@email.com	666-777-8888	789 Oak St
8	Sam	Anderson	sam.anderson@email.com	555-666-7777	456 Birch St

tomers 1

b. *Accounts*

i.
```
INSERT INTO Accounts (account_id, customer_id, account_type, balance)
VALUES
(1, 1, 'savings', 10000.50),
(2, 2, 'current', 7500.25),
(3, 3, 'savings', 5000.75),
(4, 4, 'current', 12000.30),
(5, 5, 'savings', 1500.20),
(6, 6, 'current', 9000.15),
(7, 7, 'savings', 3000.40),
(8, 8, 'current', 6000.60),
(9, 9, 'savings', 3000.75),
(10, 10, 'current', 6000.25);
```

account_id	customer_id	account_type	balance
1	1	savings	14000.50
2	2	current	7500.25
3	3	savings	5000.75
4	4	current	12000.30
5	5	savings	1500.20
6	6	current	9000.15
7	7	savings	3000.40

c. *Transactions*

i.
```
INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
VALUES
(1, 1, 'deposit', 500.75, '2024-01-22'),
(2, 2, 'withdrawal', 250.50, '2024-01-23'),
(3, 3, 'deposit', 1000.00, '2024-01-24'),
(4, 4, 'withdrawal', 500.25, '2024-01-25'),
(5, 5, 'deposit', 200.50, '2024-01-26'),
(6, 6, 'withdrawal', 150.20, '2024-01-27'),
(7, 7, 'deposit', 300.30, '2024-01-28'),
(8, 8, 'withdrawal', 450.60, '2024-01-29'),
(9, 9, 'deposit', 1000.50, '2024-02-01'),
(10, 10, 'withdrawal', 750.30, '2024-02-02');
```

transaction_id	account_id	transaction_type	amount	transaction_date
1	1	deposit	500.75	2024-01-22
2	2	withdrawal	250.50	2024-01-23
3	3	deposit	1000.00	2024-01-24
4	4	withdrawal	500.25	2024-01-25
5	5	deposit	200.50	2024-01-26
6	6	withdrawal	150.20	2024-01-27
7	7	deposit	300.30	2024-01-28

2. ***Write SQL queries for the following tasks:***

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

1.
```
SELECT first_name, last_name, account_type, email
FROM Customers
JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
```

first_name	last_name	account_type	email
Clark	Kent	savings	clark.kent@email.com
Diana	Prince	current	diana.prince@email.com
Bruce	Wayne	savings	bruce.wayne@email.com
Selina	Kyle	current	selina.kyle@email.com
Tony	Stark	savings	tony.stark@email.com
Natasha	Romanoff	current	natasha.romanoff@email.com

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

1.
```
SELECT c.first_name, c.last_name, t.*
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id
JOIN Transactions t ON a.account_id = t.account_id;
```

first_name	last_name	transaction_id	account_id	transaction_type	amount	transaction_date
Clark	Kent	1	1	deposit	500.75	2024-01-22
Diana	Prince	2	2	withdrawal	250.50	2024-01-23
Bruce	Wayne	3	3	deposit	1000.00	2024-01-24
Selina	Kyle	4	4	withdrawal	500.25	2024-01-25
Tony	Stark	5	5	deposit	200.50	2024-01-26
Natasha	Romanoff	6	6	withdrawal	150.20	2024-01-27
Wade	Wilson	7	7	deposit	300.30	2024-01-28

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

1.
```
UPDATE Accounts
SET balance = balance + 1000
WHERE account_id = 1;
```

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

1.
```
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM Customers;
```

full_name
Bruce Wayne
Selina Kyle
Tony Stark
Natasha Romanoff
Wade Wilson

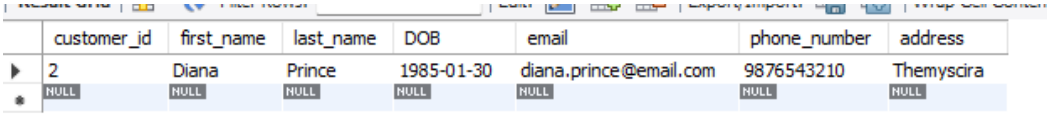
5.Write a SQL query to remove accounts with a balance of zero where the account
type is savings.

1.
```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM Accounts
WHERE balance = 0 AND account_type = 'savings';
SET SQL_SAFE_UPDATES = 1;
```

account_id	customer_id	account_type	balance
1	1	savings	14000.50
2	2	current	7500.25
3	3	savings	5000.75
4	4	current	12000.30
5	5	savings	1500.20

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

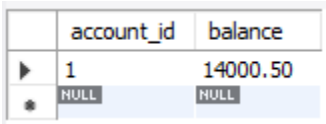
```
SELECT *
FROM Customers
WHERE address LIKE 'Themyscira'; -- Replace SpecificCity with the actual city
name
```



customer_id	first_name	last_name	DOB	email	phone_number	address
2	Diana	Prince	1985-01-30	diana.prince@email.com	9876543210	Themyscira
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

1.
```
SELECT account_id, balance
FROM Accounts
WHERE account_id = 1; -- Replace 1 with the specific account_id
```



account_id	balance
1	14000.50
NULL	NULL

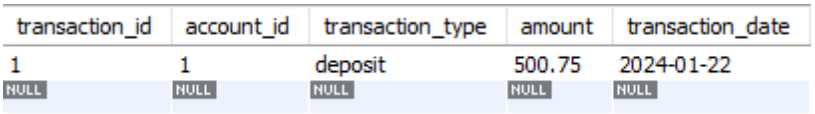
8.Write a SQL query to List all current accounts with a balance greater than $1,000.

1.
```
SELECT *
FROM Accounts
WHERE account_type = 'current' AND balance > 1000;
```

account_id	customer_id	account_type	balance
2	2	current	7500.25
4	4	current	12000.30
6	6	current	9000.15
8	8	current	6000.60
10	10	current	6000.25

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

1.
```
SELECT *
FROM Transactions
WHERE account_id = 1; -- Replace 1 with the specific account_id
```



transaction_id	account_id	transaction_type	amount	transaction_date
1	1	deposit	500.75	2024-01-22
NULL	NULL	NULL	NULL	NULL

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

1. ```
   SELECT account_id, balance * 0.05 AS interest_accrued
   FROM Accounts
   WHERE account_type = 'savings';
   ```

account_id	interest_accrued
1	700.0250
3	250.0375
5	75.0100
7	150.0200
9	150.0375

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

1. ```
   SELECT *
   FROM Accounts
   WHERE balance < 100000; -- Replace -500 with the specified overdraft limit
   ```

account_id	customer_id	account_type	balance
1	1	savings	14000.50
2	2	current	7500.25
3	3	savings	5000.75
4	4	current	12000.30
5	5	savings	1500.20

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

- ```
  SELECT *
  FROM Customers
  WHERE address NOT LIKE '%Themyscira%'; -- Replace SpecificCity with the actual city name
  ```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Clark	Kent	1978-05-03	clark.kent@email.com	1234567890	344 Daily Planet
3	Bruce	Wayne	1972-11-15	bruce.wayne@email.com	5555555555	1007 Wayne Manor
4	Selina	Kyle	1982-08-17	selina.kyle@email.com	1111111111	22 Cat Street
5	Tony	Stark	1970-06-12	tony.stark@email.com	9999999999	10880 Malibu Point
6	Natasha	Romanoff	1984-04-20	natasha.romanoff@email.com	4444444444	64 Red Room Lane

- ***Tasks 3: Aggregate functions, Having, Order By, GroupBy and Join***

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

   a. ```
      SELECT AVG(balance) AS average_balance
      FROM Accounts;
      ```

average_balance
6700.415000

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

   a.
```
SELECT customer_id, account_type, balance
FROM Accounts
ORDER BY balance DESC
LIMIT 10;
```

customer_id	account_type	balance
1	savings	14000.50
4	current	12000.30
6	current	9000.15
2	current	7500.25
8	current	6000.60

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

   a.
```
SELECT SUM(amount) AS total_deposits
FROM Transactions
WHERE transaction_type = 'deposit' AND transaction_date = '2024-01-22';
```

total_deposits
500.75

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

   a. *SELECT MIN(DOB) AS oldest_customer, MAX(DOB) AS newest_customer*
   FROM Customers;

oldest_customer	newest_customer
1970-06-12	1992-08-10

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

   a.
```
SELECT t.*, a.account_type
FROM Transactions t
JOIN Accounts a ON t.account_id = a.account_id;
```

transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1	1	deposit	500.75	2024-01-22	savings
2	2	withdrawal	250.50	2024-01-23	current
3	3	deposit	1000.00	2024-01-24	savings
4	4	withdrawal	500.25	2024-01-25	current
5	5	deposit	200.50	2024-01-26	savings

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

   a.
```
SELECT c., a.
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id;
```

customer_id	first_name	last_name	DOB	email	phone_number	address	accour
1	Clark	Kent	1978-05-03	clark.kent@email.com	1234567890	344 Daily Planet	1
2	Diana	Prince	1985-01-30	diana.prince@email.com	9876543210	Themyscira	2
3	Bruce	Wayne	1972-11-15	bruce.wayne@email.com	5555555555	1007 Wayne Manor	3
4	Selina	Kyle	1982-08-17	selina.kyle@email.com	1111111111	22 Cat Street	4

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

   a.
```
SELECT c.*, t*.
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 = 1; -- Replace 1 with the specific account_id
```

customer_id	first_name	last_name	DOB	email	phone_number	address	transaction_id	acco
1	Clark	Kent	1978-05-03	clark.kent@email.com	1234567890	344 Daily Planet	1	1

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

   a.
```
SELECT customer_id, COUNT(account_id) AS num_accounts
FROM Accounts
GROUP BY customer_id
HAVING COUNT(account_id) > 1;
```

customer_id	num_accounts

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

a.
```
SELECT account_id,
SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) -
SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS
net_transaction_amount
FROM Transactions
GROUP BY account_id;
```

account_id	net_transaction_amount
1	500.75
2	-250.50
3	1000.00
4	-500.25
5	200.50

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

a.
```
SELECT account_id, AVG(balance) AS average_daily_balance
FROM Accounts
WHERE DATEDIFF(NOW(), balance_date) <= 7
GROUP BY account_id;
```

account_id	net_transaction_amount
1	500.75
2	-250.50
3	1000.00
4	-500.25
5	200.50

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

a.
```
SELECT account_type, SUM(balance) AS total_balance
FROM Accounts
GROUP BY account_type;
```

account_id	net_transaction_amount
1	500.75
2	-250.50
3	1000.00
4	-500.25
5	200.50

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

a.
```
SELECT account_id, COUNT(transaction_id) AS num_transactions
FROM Transactions
GROUP BY account_id
ORDER BY num_transactions DESC;
```

account_id	num_transactions
1	1
2	1
3	1
4	1
5	1

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

    a.
```
SELECT c.customer_id, c.first_name, c.last_name, a.account_type,
SUM(a.balance) AS aggregate_balance
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id
GROUP BY c.customer_id, a.account_type
HAVING aggregate_balance > 5000; -- Set your threshold
```

customer_id	first_name	last_name	account_type	aggregate_balance
1	Clark	Kent	savings	14000.50
2	Diana	Prince	current	7500.25
3	Bruce	Wayne	savings	5000.75
4	Selina	Kyle	current	12000.30
6	Natasha	Romanoff	current	9000.15

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

    a.
```
SELECT amount, transaction_date, account_id, COUNT() AS num_duplicates
FROM Transactions
GROUP BY amount, transaction_date, account_id
HAVING COUNT() > 1;
```

amount	transaction_date	account_id	num_duplicates

- ***Tasks 4: Subquery and its type:***

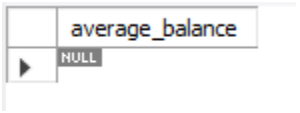
1. *Retrieve the customer(s) with the highest account balance.*

```
SELECT *
FROM Customers
WHERE customer_id = (SELECT customer_id FROM Accounts ORDER BY balance DESC LIMIT
1);
```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Clark	Kent	1978-05-03	clark.kent@email.com	1234567890	344 Daily Planet
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

   a.
```
SELECT AVG(balance) AS average_balance
FROM Accounts
WHERE customer_id IN (SELECT customer_id FROM Accounts GROUP BY customer_id
HAVING COUNT(account_id) > 1);
```



average_balance
NULL

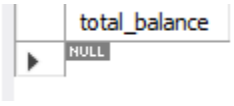
3. *Retrieve accounts with transactions whose amounts exceed the average transaction amount.*

   a.
```
SELECT *
FROM Accounts
WHERE account_id IN (SELECT account_id FROM Transactions WHERE amount >
(SELECT AVG(amount) FROM Transactions));
```

account_id	customer_id	account_type	balance
3	3	savings	5000.75
9	9	savings	3000.75
10	10	current	6000.25
NULL	NULL	NULL	NULL

4. *Identify customers who have no recorded transactions.*

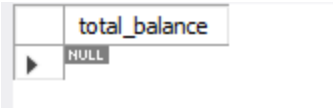
   a.
```
SELECT SUM(balance) AS total_balance
FROM Accounts
WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);
```



total_balance
NULL

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

   a.
```
SELECT SUM(balance) AS total_balance
FROM Accounts
WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);
```



total_balance
NULL

6. *Retrieve transactions for accounts with the lowest balance.*

a.
```
SELECT *
FROM Transactions
WHERE account_id IN (SELECT account_id FROM Accounts ORDER BY balance ASC
LIMIT 1);
```

customer_id

7. *Identify customers who have accounts of multiple types.*

a.
```
SELECT customer_id
FROM Accounts
GROUP BY customer_id
HAVING COUNT(DISTINCT account_type) > 1;
```

customer_id

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

a.
```
SELECT account_type,
COUNT() * 100.0 / (SELECT COUNT() FROM Accounts) AS percentage
FROM Accounts
GROUP BY account_type;
```

account_type	percentage
savings	50.00000
current	50.00000

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

a.
```
SELECT *
FROM Transactions
WHERE account_id IN (SELECT account_id FROM Accounts WHERE customer_id = 1); --
- Replace 1 with the specific customer_id
```

transaction_id	account_id	transaction_type	amount	transaction_date
1	1	deposit	500.75	2024-01-22
NULL	NULL	NULL	NULL	NULL

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

a.
```
SELECT account_type,
(SELECT SUM(balance) FROM Accounts WHERE account_type = a.account_type) AS
total_balance
FROM Accounts a
GROUP BY account_type;
```

account_type	total_balance
savings	26502.60
current	40501.55