Assignment – 3 Banking System (SQL)

Task 1:

1. Create the database named "HMBank"

```
create database HMBank;
use HMBank;
```

create database HMBank	1 row(s) affected	0.000 sec
use HMBank	0 row(s) affected	0.000 sec

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

Customers Table:

- customer_id (Primary Key, INT)
- first_name (TEXT)
- last_name (TEXT)
- DOB (DATE)
- email (TEXT)
- phone_number (TEXT)
- address (TEXT)

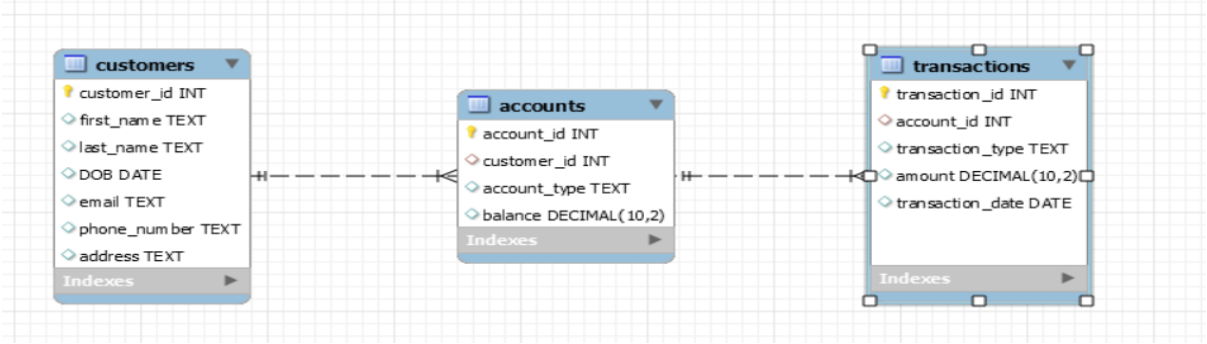
Accounts Table:

- account_id (Primary Key, INT)
- customer_id (Foreign Key, INT referencing Customers(customer_id))
- account_type (TEXT)
- balance (DECIMAL(10, 2))

Transactions Table:

- transaction_id (Primary Key, INT)
- account_id (Foreign Key, INT referencing Accounts(account_id))
- transaction_type (TEXT)
- amount (DECIMAL(10, 2))
- transaction_date (DATE)

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



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

Field	Type	Null	Key	Default
customer_id	int	NO	PRI	NULL
first_name	text	YES		NULL
last_name	text	YES		NULL
DOB	date	YES		NULL
email	text	YES		NULL
phone_number	text	YES		NULL
address	text	YES		NULL

Field	Type	Null	Key	Default
account_id	int	NO	PRI	NULL
customer_id	int	YES	MUL	NULL
account_type	text	YES		NULL
balance	decimal(10,2)	YES		NULL

Field	Type	Null	Key	Default
transaction_id	int	NO	PRI	NULL
accou	int	YES	MUL	NULL
transaction_type	text	YES		NULL
amount	decimal(10,2)	YES		NULL
transaction_date	date	YES		NULL

account_id

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 TEXT,
    last_name TEXT,
    DOB DATE,
    email TEXT,
    phone_number TEXT,
    address TEXT
);
```

Field	Type	Null	Key	Default
customer_id	int	NO	PRI	NULL
first_name	text	YES		NULL
last_name	text	YES		NULL
DOB	date	YES		NULL
email	text	YES		NULL
phone_number	text	YES		NULL
address	text	YES		NULL

• Accounts

```
CREATE TABLE Accounts (
    account_id INT PRIMARY KEY,
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
    account_type TEXT,
    balance DECIMAL(10, 2)
);
```

Field	Type	Null	Key	Default
account_id	int	NO	PRI	NULL
customer_id	int	YES	MUL	NULL
account_type	text	YES		NULL
balance	decimal(10,2)	YES		NULL

• Transactions

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

Field	Type	Null	Key	Default
transaction_id	int	NO	PRI	NULL
accou account_id	int	YES	MUL	NULL
transaction_type	text	YES		NULL
amount	decimal(10,2)	YES		NULL
transaction_date	date	YES		NULL

Task 2:

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

• Customers

```
INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number, address)
VALUES
    (1, 'Raj', 'Kumar', '1990-05-15', 'raj.kumar@email.com', '9876543210', 'Flat 101, Sunflower Apartments, MG Road, Bangalore'),
    (2, 'Anjali', 'Sharma', '1985-12-20', 'anjali.sharma@email.com', '8765432109', 'Plot 45, Sector 8, Rohini, Delhi'),
    (3, 'Vijay', 'Singh', '1988-08-07', 'vijay.singh@email.com', '7654321098', 'House 32, Gandhi Street, Chennai'),
    (4, 'Priya', 'Patel', '1995-02-25', 'priya.patel@email.com', '9843210987', 'Villa 5, Green Valley, Gandhinagar'),
    (5, 'Amit', 'Verma', '1980-11-10', 'amit.verma@email.com', '9432109876', 'A-402, Sky Towers, Powai, Mumbai'),
    (6, 'Neha', 'Gupta', '1992-09-18', 'neha.gupta@email.com', '9321098765', 'B-205, Rosewood Residency, Baner, Pune'),
    (7, 'Rahul', 'Yadav', '1983-04-30', 'rahul.yadav@email.com', '9210987654', 'Cottage 12, Lakeview, Nainital'),
    (8, 'Sneha', 'Rajput', '1997-07-05', 'sneha.rajput@email.com', '9109876543', 'Apartment 303, Emerald Enclave, Hyderabad'),
    (9, 'Sanjay', 'Malhotra', '1987-01-12', 'sanjay.malhotra@email.com', '9098765432', 'H-501, Silver Springs, Kolkata'),
    (10, 'Pooja', 'Bose', '1994-06-28', 'pooja.bose@email.com', '9876543210', 'D-102, Serene Heights, Whitefield, Bangalore');
```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Raj	Kumar	1990-05-15	raj.kumar@email.com	9876543210	Flat 101, Sunflower Apartments, MG Road, Ban...
2	Anjali	Sharma	1985-12-20	anjali.sharma@email.com	8765432 9876543210	5, Sector 8, Rohini, Delhi
3	Vijay	Singh	1988-08-07	vijay.singh@email.com	7654321098	House 32, Gandhi Street, Chennai
4	Priya	Patel	1995-02-25	priya.patel@email.com	9843210987	Villa 5, Green Valley, Gandhinagar
5	Amit	Verma	1980-11-10	amit.verma@email.com	9432109876	A-402, Sky Towers, Powai, Mumbai
6	Neha	Gupta	1992-09-18	neha.gupta@email.com	9321098765	B-205, Rosewood Residency, Baner, Pune
7	Rahul	Yadav	1983-04-30	rahul.yadav@email.com	9210987654	Cottage 12, Lakeview, Nainital
8	Sneha	Rajput	1997-07-05	sneha.rajput@email.com	9109876543	Apartment 303, Emerald Enclave, Hyderabad
9	Sanjay	Malhotra	1987-01-12	sanjay.malhotra@email.com	9098765432	H-501, Silver Springs, Kolkata
10	Pooja	Bose	1994-06-28	pooja.bose@email.com	9876543210	D-102, Serene Heights, Whitefield, Bangalore

• Accounts

```sql
select * from customers;
INSERT INTO Accounts (account_id, customer_id, account_type, balance)
VALUES
    (101, 1, 'savings', 50000.00),
    (102, 2, 'current', 75000.50),
    (103, 3, 'zero_balance', 0.00),
    (104, 4, 'savings', 120000.75),
    (105, 5, 'current', 25000.25),
    (106, 6, 'savings', 100000.50),
    (107, 7, 'zero_balance', 0.00),
    (108, 8, 'current', 30000.00),
    (109, 9, 'savings', 80000.25),
    (110, 10, 'current', 40000.50);
```

account_id	customer_id	account_type	balance
101	1	savings	50000.00
102	2	current	75000.50
103	3	zero_balance	0.00
104	4	savings	120000.75
105	5	current	25000.25
106	6	savings	100000.50
107	7	zero_balance	0.00
108	8	current	30000.00
109	9	savings	80000.25
110	10	current	40000.50

• Transactions

```sql
INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
VALUES
    (201, 101, 'deposit', 10000.00, '2023-01-05'),
    (202, 102, 'withdrawal', 5000.50, '2023-02-12'),
    (203, 103, 'deposit', 2000.75, '2023-03-20'),
    (204, 104, 'withdrawal', 15000.25, '2023-04-15'),
    (205, 105, 'deposit', 5000.50, '2023-05-28'),
    (206, 106, 'transfer', 10000.75, '2023-06-10'),
    (207, 107, 'deposit', 3000.00, '2023-07-18'),
    (208, 108, 'withdrawal', 8000.25, '2023-08-22'),
    (209, 109, 'transfer', 12000.50, '2023-09-30'),
    (210, 110, 'withdrawal', 6000.00, '2023-10-14');
```

transaction_id	account_id	transaction_type	amount	transaction_date
201	101	deposit	10000.00	2023-01-05
202	202	withdrawal	5000.50	2023-02-12
203	103	deposit	2000.75	2023-03-20
204	104	withdrawal	15000.25	2023-04-15
205	105	deposit	5000.50	2023-05-28
206	106	transfer	10000.75	2023-06-10
207	107	deposit	3000.00	2023-07-18
208	108	withdrawal	8000.25	2023-08-22
209	109	transfer	12000.50	2023-09-30
210	110	withdrawal	6000.00	2023-10-14

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 AS customer_name,account_type,email
FROM Customers JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
```

customer_name	account_type	email
Raj	savings	raj.kumar@email.com
Anjali	current	anjali.sharma@email.com
Vijay	zero_balance	vijay.singh@email.com
Priya	savings	priya.patel@email.com
Amit	current	amit.verma@email.com
Neha	savings	neha.gupta@email.com
Rahul	zero_balance	rahul.yadav@email.com

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

```
SELECT c.first_name AS customer_name,t.transaction_id,t.transaction_type,t.amount,t.transaction_date
FROM Customers c JOIN Accounts a ON c.customer_id = a.customer_id JOIN Transactions t ON a.account_id = t.account_id;
```

customer_name	transaction_id	transaction_type	amount	transaction_date
Raj	201	deposit	10000.00	2023-01-05
Anjali	202	withdrawal	5000.50	2023-02-12
Vijay	203	deposit	2000.75	2023-03-20
Priya	204	withdrawal	15000.25	2023-04-15
Amit	205	deposit	5000.50	2023-05-28
Neha	206	transfer	10000.75	2023-06-10
Rahul	207	deposit	3000.00	2023-07-18
Sneha	208	withdrawal	8000.25	2023-08-22

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

```
UPDATE Accounts SET balance = balance + 5000.00 WHERE account_id = 101;
```

account_id	customer_id	account_type	balance
101	1	savings	55000.00

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

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

full_name
Raj Kumar
Anjali Sharma
Vijay Singh
Priya Patel
Amit Verma
Neha Gupta
Rahul Yadav

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

type is savings.

```
DELETE FROM Accounts WHERE balance = 0.00 AND account_type = 'savings';
```

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

```
SELECT * FROM Customers WHERE address LIKE 'Plot 45, Sector 8, Rohini, Delhi';
```

customer_id	first_name	last_name	DOB	email	phone_number	address
2	Anjali	Sharma	1985-12-20	anjali.sharma@email.com	8765432109	Plot 45, Sector 8, Rohini, Delhi

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

```
SELECT account_id,balance FROM Accounts WHERE account_id = 101;
```

account_id	balance
101	55000.00

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

```
SELECT account_id,customer_id,balance FROM Accounts
WHERE account_type = 'current' AND balance > 1000.00;
```

account_id	customer_id	balance
102	2	75000.50
105	5	25000.25
108	8	30000.00
110	10	40000.50

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

```
SELECT t.transaction_id,t.transaction_type,t.amount,t.transaction_date
FROM Transactions t JOIN Accounts a ON t.account_id = a.account_id
WHERE a.account_id = 101;
```

transaction_id	transaction_type	amount	transaction_date
201	deposit	10000.00	2023-01-05

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

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

account_id	customer_id	balance	interest_accrued
101	1	55000.00	2750.0000
104	4	120000.75	6000.0375
106	6	100000.50	5000.0250
109	9	80000.25	4000.0125

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

```
SELECT account_id,customer_id,balance
FROM Accounts
WHERE balance < 40000.00;
```

account_id	customer_id	balance
103	3	0.00
105	5	25000.25
107	7	0.00
108	8	30000.00

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

```
SELECT * FROM Customers WHERE LOWER(address) NOT LIKE '%delhi%';
```

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Raj	Kumar	1990-05-15	raj.kumar@email.com	9876543210	Flat 101, Sunflower Apartments, MG Road, Ban...
3	Vijay	Singh	1988-08-07	vijay.singh@email.com	7654321098	House 32, Gandhi Street, Chennai
4	Priya	Patel	1995-02-25	priya.patel@email.com	9843210987	Villa 5, Green Valley, Gandhinagar
5	Amit	Verma	1980-11-10	amit.verma@email.com	9432109876	A-402, Sky Towers, Powai, Mumbai
6	Neha	Gupta	1992-09-18	neha.gupta@email.com	9321098765	B-205, Rosewood Residency, Baner, Pune
7	Rahul	Yadav	1983-04-30	rahul.yadav@email.com	9210987654	Cottage 12, Lakeview, Nainital
8	Sneha	Rajput	1997-07-05	sneha.rajput@email.com	9109876543	Apartment 303, Emerald Enclave, Hyderabad
9	Sanjay	Malhotra	1987-01-12	sanjay.malhotra@email.com	9098765432	H-501, Silver Springs, Kolkata

Task 3:

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

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

average_balance
52500.275000

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

```
SELECT account_id,customer_id,balance FROM Accounts ORDER BY balance DESC LIMIT 10;
```

account_id	customer_id	balance
104	4	120000.75
106	6	100000.50
109	9	80000.25
102	2	75000.50
101	1	55000.00
110	10	40000.50
108	8	30000.00
105	5	25000.25

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

```
SELECT c.customer_id,c.first_name,SUM(t.amount) 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_type = 'deposit' AND t.transaction_date = '2023-01-05'
GROUP BY c.customer_id, c.first_name;
```

customer_id	first_name	total_deposits
1	Raj	10000.00

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

```
SELECT   customer_id,first_name, DOB AS birthdate FROM Customers ORDER BY DOB ASC LIMIT 1;
SELECT customer_id,first_name,DOB AS birthdate FROM Customers ORDER BY DOB DESC LIMIT 1;
```

customer_id	first_name	birthdate
5	Amit	1980-11-10

customer_id	first_name	birthdate
8	Sneha	1997-07-05

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

```
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
201	101	savings	deposit	10000.00	2023-01-05
202	102	current	withdrawal	5000.50	2023-02-12
203	103	zero_balance	deposit	2000.75	2023-03-20
204	104	savings	withdrawal	15000.25	2023-04-15
205	105	current	deposit	5000.50	2023-05-28
206	106	savings	transfer	10000.75	2023-06-10
207	107	zero_balance	deposit	3000.00	2023-07-18
208	108	current	withdrawal	8000.25	2023-08-22

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

```
SELECT c.customer_id,c.first_name,a.account_id,a.account_type,a.balance
FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.customer_id;
```

customer_id	first_name	account_id	account_type	balance
1	Raj	101	savings	55000.00
2	Anjali	102	current	75000.50
3	Vijay	103	zero_balance	0.00
4	Priya	104	savings	120000.75
5	Amit	105	current	25000.25
6	Neha	106	savings	100000.50
7	Rahul	107	zero_balance	0.00
8	Sneha	108	current	30000.00

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

```
SELECT c.customer_id,c.first_name,c.DOB AS date_of_birth,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 = 101;
```

customer_id	first_name	date_of_birth	transaction_id	transaction_type	amount	transaction_date
1	Raj	1990-05-15	201	deposit	10000.00	2023-01-05

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

```
SELECT customer_id,first_name FROM Customers c
WHERE EXISTS (SELECT * FROM Accounts a WHERE c.customer_id = a.customer_id HAVING COUNT(a.account_id) > 1 );
```

customer_id	first_name
NULL	NULL

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

```
SELECT SUM(amount * (transaction_type = 'deposit') - amount * (transaction_type = 'withdrawal')) AS transaction_difference
FROM Transactions;
```

transaction_difference
-13999.75

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

```
SELECT a.account_id,
AVG((SELECT COALESCE(SUM(t2.amount * IFNULL(NULLIF(SIGN(t2.amount), -1), 1)), 0)
FROM Transactions t2
WHERE t2.account_id = a.account_id
AND t2.transaction_date <= t1.transaction_date)) AS average_daily_balance
FROM Accounts a JOIN Transactions t1 ON a.account_id = t1.account_id
WHERE t1.transaction_date BETWEEN '2012-01-01' AND '2012-12-20'
GROUP BY  a.account_id;
```

11. Calculate the total balance for each account type.

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

account_type	total_balance
savings	355001.50
current	170001.25
zero_balance	0.00

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

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

account_id	transaction_count
101	1
102	1
103	1
104	1
105	1
106	1
107	1
108	1

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

```
SELECT c.customer_id,c.first_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, c.first_name, a.account_type
ORDER BY aggregate_balance DESC;
```

customer_id	first_name	account_type	aggregate_balance
4	Priya	savings	120000.75
6	Neha	savings	100000.50
9	Sanjay	savings	80000.25
2	Anjali	current	75000.50
1	Raj	savings	55000.00
10	Pooja	current	40000.50
8	Sneha	current	30000.00
5	Amit	current	25000.25

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

```
SELECT transaction_id,account_id,transaction_type,amount,transaction_date
FROM Transactions WHERE (amount, transaction_date, account_id) IN (SELECT amount,transaction_date,account_id FROM Transactions GROUP BY amount, transaction_date, account_id
HAVING COUNT(transaction_id) > 1);
```

transaction_id	account_id	transaction_type	amount	transaction_date
NULL	NULL	NULL	NULL	NULL

Tasks 4:

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

```
SELECT c.customer_id,c.first_name,MAX(a.balance) AS highest_balance
FROM Customers c JOIN Accounts a ON c.customer_id = a.customer_id
GROUP BY c.customer_id
ORDER BY highest_balance DESC LIMIT 1;
```

customer_id	first_name	highest_balance
4	Priya	120000.75

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

```
SELECT c.customer_id,c.first_name,AVG(a.balance) AS average_balance
FROM Customers c JOIN Accounts a ON c.customer_id = a.customer_id
GROUP BY c.customer_id
HAVING COUNT(DISTINCT a.account_id) > 1;
```

customer_id	first_name	average_balance
1	Raj	52500.000000

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

```
SELECT * FROM Transactions t WHERE t.amount > (SELECT AVG(amount) FROM Transactions t2 WHERE t2.account_id = t.account_id);
```

transaction_id	account_id	transaction_type	amount	transaction_date
NULL	NULL	NULL	NULL	NULL

4. Identify customers who have no recorded transactions.

```
SELECT c.customer_id,c.first_name
FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.customer_id LEFT JOIN Transactions t ON a.account_id = t.account_id
WHERE t.transaction_id IS NULL;
```

customer_id	first_name
1	Raj

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

```
SELECT SUM(balance) AS total_balance FROM Accounts a LEFT JOIN Transactions t ON a.account_id = t.account_id  WHERE t.transaction_id IS NULL;
```

total_balance
50000.00

6. Retrieve transactions for accounts with the lowest balance.

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

transaction_id	account_id	transaction_type	amount	transaction_date
203	103	deposit	2000.75	2023-03-20

7. Identify customers who have accounts of multiple types.

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

transaction_id	account_id	transaction_type	amount	transaction_date
203	103	deposit	2000.75	2023-03-20

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

```
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	4	36.36364
current	5	45.45455
zero_balance	2	18.18182

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

```
SELECT * FROM Transactions t JOIN Accounts a ON t.account_id = a.account_id WHERE a.customer_id = '5';
```

transaction_id	account_id	transaction_type	amount	transaction_date	account_id	customer_id	account_type	balance
205	105	deposit	5000.50	2023-05-28	105	5	current	25000.25

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

```
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	355001.50
current	220001.25
zero_balance	0.00