create database bank;

use bank;

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

DOB DATE,

email VARCHAR(100),

phone\_number VARCHAR(15),

address VARCHAR(255)

);

CREATE TABLE Accounts (

account\_id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

account\_type VARCHAR(20),

balance DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

CREATE TABLE Transactions (

transaction\_id INT PRIMARY KEY AUTO\_INCREMENT,

account\_id INT,

transaction\_type VARCHAR(20),

amount DECIMAL(10, 2),

transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (account\_id) REFERENCES Accounts(account\_id)

);

INSERT INTO Customers (first\_name, last\_name, DOB, email, phone\_number, address) VALUES

('John', 'Doe', '1990-05-15', 'john.doe@example.com', '1234567890', '123 Main St, Anytown, USA'),

('Alice', 'Smith', '1985-08-21', 'alice.smith@example.com', '9876543210', '456 Elm St, Anycity, USA'),

('Bob', 'Johnson', '1978-12-03', 'bob.johnson@example.com', '5551234567', '789 Oak St, Anyville, USA'),

('Emily', 'Brown', '1993-04-29', 'emily.brown@example.com', '5559876543', '101 Pine St, Anysuburb, USA'),

('Michael', 'Davis', '1982-11-11', 'michael.davis@example.com', '3334445555', '222 Maple St, Anycountryside, USA'),

('Sara', 'Miller', '1991-09-18', 'sara.miller@example.com', '2223334444', '333 Oak Ave, Anyborough, USA'),

('David', 'Wilson', '1987-07-07', 'david.wilson@example.com', '7778889999', '444 Elm St, Anyvillage, USA'),

('Emma', 'Garcia', '1980-03-25', 'emma.garcia@example.com', '6667778888', '555 Pine St, Anycity, USA'),

('James', 'Martinez', '1995-01-12', 'james.martinez@example.com', '4445556666', '666 Maple St, Anytown, USA'),

('Jennifer', 'Lopez', '1989-06-30', 'jennifer.lopez@example.com', '9990001111', '777 Oak St, Anytown, USA');

INSERT INTO Accounts (customer\_id, account\_type, balance) VALUES

(1, 'Savings', 5000.00),

(2, 'Checking', 2500.00),

(3, 'Savings', 7500.00),

(4, 'Checking', 3000.00),

(5, 'Savings', 10000.00),

(6, 'Checking', 4000.00),

(7, 'Savings', 6000.00),

(8, 'Checking', 2000.00),

(9, 'Savings', 8000.00),

(10, 'Checking', 3500.00);

INSERT INTO Transactions (account\_id, transaction\_type, amount) VALUES

(1, 'Deposit', 2000.00),

(2, 'Withdrawal', 500.00),

(3, 'Deposit', 1000.00),

(4, 'Withdrawal', 300.00),

(5, 'Deposit', 1500.00),

(6, 'Deposit', 3000.00),

(7, 'Withdrawal', 1000.00),

(8, 'Deposit', 500.00),

(9, 'Withdrawal', 2000.00),

(10, 'Deposit', 800.00);

-- Basic commands

SELECT c.first\_name, c.last\_name, a.account\_type, c.email

FROM Customers c

JOIN Accounts a ON c.customer\_id = a.customer\_id;

SELECT c.first\_name, c.last\_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;

UPDATE Accounts

SET balance = balance + 5000

WHERE account\_id = 4;

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name

FROM Customers;

SELECT \*

FROM Customers

WHERE address LIKE '%Elm%';

SELECT balance

FROM Accounts

WHERE account\_id = 6;

SELECT \*

FROM Accounts

WHERE account\_type = 'current' AND balance > 1000.00;

SELECT \*

FROM Transactions

WHERE account\_id = 2;

SELECT \*

FROM Accounts

WHERE balance < 1000;

-- Aggregate Functions

SELECT AVG(balance) AS average\_balance

FROM Accounts;

SELECT \*

FROM Accounts

ORDER BY balance DESC

LIMIT 10;

SELECT SUM(amount) AS total\_deposits

FROM Transactions

WHERE transaction\_type = 'deposit'

AND DATE(transaction\_date) = '2024-03-08';

SELECT first\_name, last\_name, DOB

FROM Customers

ORDER BY DOB ASC LIMIT 1;

SELECT first\_name, last\_name, DOB

FROM Customers

ORDER BY DOB DESC LIMIT 1;

SELECT t.\*, a.account\_type

FROM Transactions t

JOIN Accounts a ON t.account\_id = a.account\_id;

SELECT c.\*, a.\*

FROM Customers c

JOIN Accounts a ON c.customer\_id = a.customer\_id;

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 = 3;

SELECT c.\*

FROM Customers c

JOIN Accounts a ON c.customer\_id = a.customer\_id

GROUP BY c.customer\_id

HAVING COUNT(\*) > 1;

SELECT account\_id, SUM(CASE WHEN transaction\_type = 'deposit' THEN amount ELSE -amount END) AS net\_amount

FROM Transactions

GROUP BY account\_id;

SELECT account\_id, AVG(balance) AS average\_daily\_balance

FROM Accounts

GROUP BY account\_id;

SELECT account\_type, SUM(balance) AS total\_balance

FROM Accounts

GROUP BY account\_type;

SELECT account\_id, COUNT(\*) AS transaction\_count

FROM Transactions

GROUP BY account\_id

ORDER BY transaction\_count DESC;

SELECT \*

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(\*) > 1

);

-- SubQueries

SELECT c.\*

FROM Customers c

JOIN Accounts a ON c.customer\_id = a.customer\_id

WHERE a.balance = (

SELECT MAX(balance) FROM Accounts

);

SELECT AVG(balance) AS average\_balance

FROM Accounts

WHERE customer\_id IN (

SELECT customer\_id

FROM Accounts

GROUP BY customer\_id

HAVING COUNT(\*) > 1

);

SELECT account\_id, transaction\_type, amount

FROM Transactions

WHERE amount > (

SELECT AVG(amount)

FROM Transactions

);

SELECT \*

FROM Customers

WHERE customer\_id NOT IN (

SELECT DISTINCT customer\_id

FROM Transactions

);

SELECT SUM(balance) AS total\_balance

FROM Accounts

WHERE account\_id NOT IN (

SELECT DISTINCT account\_id

FROM Transactions

);

SELECT \*

FROM Transactions

WHERE account\_id IN (

SELECT account\_id

FROM Accounts

WHERE balance = (

SELECT MIN(balance)

FROM Accounts

)

);

SELECT customer\_id

FROM Accounts

GROUP BY customer\_id

HAVING COUNT(DISTINCT account\_type) > 1;

SELECT account\_type,

COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM Accounts) AS percentage

FROM Accounts

GROUP BY account\_type;

SELECT \*

FROM Transactions

WHERE account\_id IN (

SELECT account\_id

FROM Accounts

WHERE customer\_id = 4

);

SELECT account\_type,

(SELECT SUM(balance) FROM Accounts WHERE account\_type = a.account\_type) AS total\_balance

FROM Accounts a

GROUP BY account\_type;