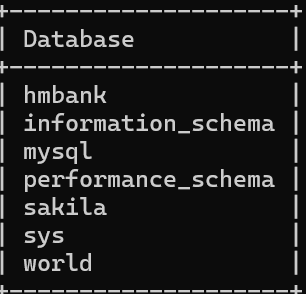
Task : 01

CREATE DATABASE HMbank;



CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

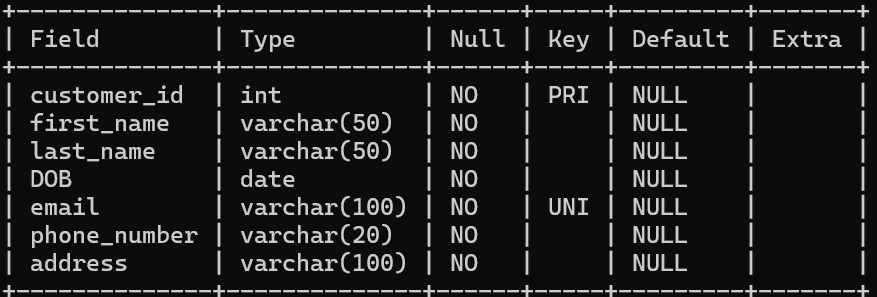
last\_name VARCHAR(50) NOT NULL,

DOB DATE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone\_number VARCHAR(20) NOT NULL

);



CREATE TABLE Accounts (

account\_id INT PRIMARY KEY,

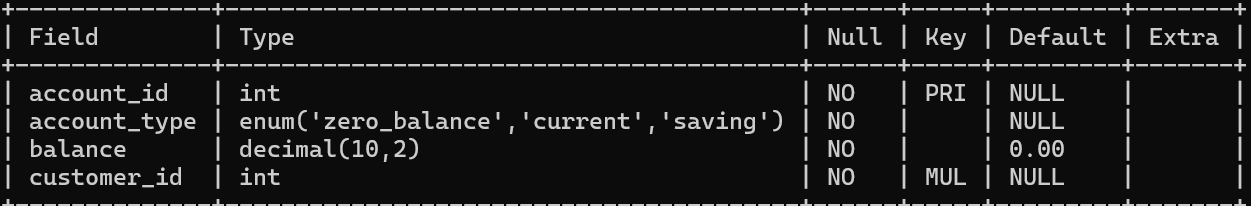
customer\_id INT NOT NULL,

account\_type ENUM('savings', 'current', 'zero\_balance') NOT NULL,

balance DECIMAL(10, 2) NOT NULL DEFAULT 0.00,

CONSTRAINT fk\_customer\_id FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);



CREATE TABLE Transactions (

transaction\_id INT PRIMARY KEY AUTO\_INCREMENT,

account\_id INT NOT NULL,

transaction\_type ENUM('deposit', 'withdrawal', 'transfer') NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

transaction\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

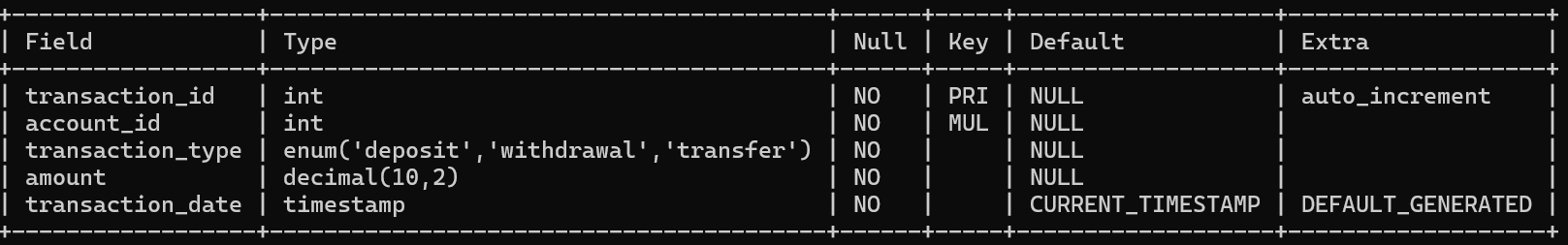
CONSTRAINT fk\_account\_id FOREIGN KEY (account\_id)

REFERENCES Accounts(account\_id)

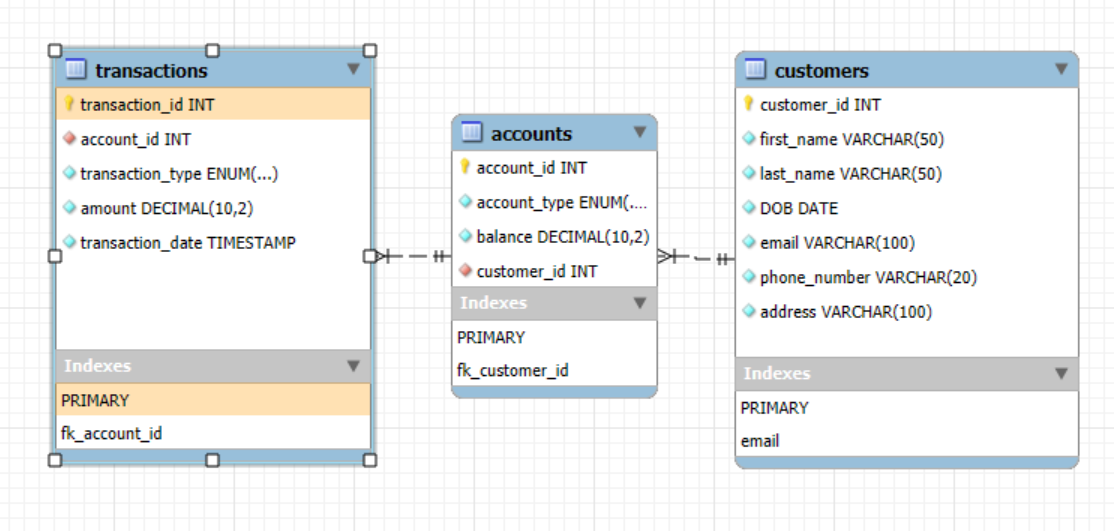
ON DELETE CASCADE

ON UPDATE CASCADE

);



ER Diagram



Task : 02

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

VALUES

(1, 1, 'deposit', 1000.00, '2022-01-01 10:00:00'),

(2, 2, 'withdrawal', 500.00, '2022-01-05 12:00:00'),

(3, 3, 'deposit', 2000.00, '2022-01-10 14:00:00'),

(4, 4, 'transfer', 1500.00, '2022-01-15 16:00:00'),

(5, 5, 'deposit', 3000.00, '2022-01-20 10:00:00'),

(6, 6, 'withdrawal', 2000.00, '2022-01-25 12:00:00'),

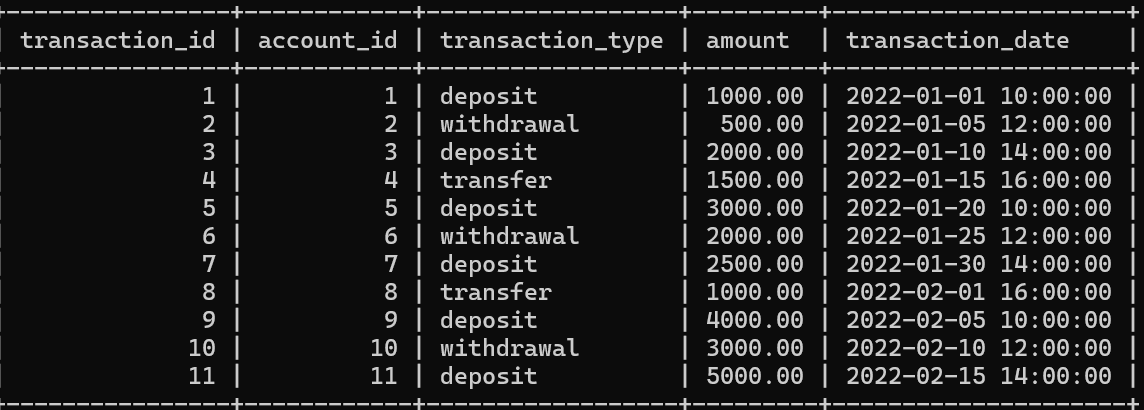
(7, 7, 'deposit', 2500.00, '2022-01-30 14:00:00'),

(8, 8, 'transfer', 1000.00, '2022-02-01 16:00:00'),

(9, 9, 'deposit', 4000.00, '2022-02-05 10:00:00'),

(10, 10, 'withdrawal', 3000.00, '2022-02-10 12:00:00'),

(11, 11, 'deposit', 5000.00, '2022-02-15 14:00:00');



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

VALUES

(1, 'savings', 1000.00, 2),

(2, 'current', 5000.00, 3),

(3, 'zero\_balance', 0.00, 4),

(4, 'savings', 2000.00, 5),

(5, 'current', 3000.00, 9012001),

(6, 'savings', 1500.00, 9012002),

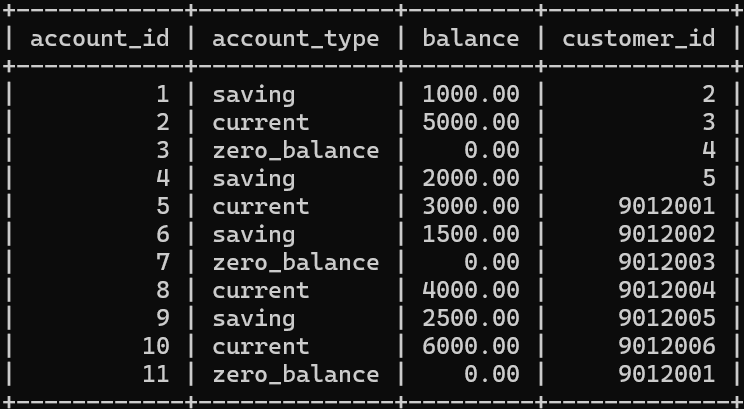
(7, 'zero\_balance', 0.00, 9012003),

(8, 'current', 4000.00, 9012004),

(9, 'savings', 2500.00, 9012005),

(10, 'current', 6000.00, 9012006),

(11, 'zero\_balance', 0.00, 9012001);



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

VALUES

(‘9012001’ , ‘Sameer’ , ’ Shukla’ ,’2001-01-01’ ,‘sameershukla@gmail.com’ , ‘6260575863’ , ‘ Candla Gujarat’),

(‘9012002’ , ‘Adesh’ , ‘Tomar’ , ‘2001-03-14’ , ‘tomaradesh@gmail.com’ , ‘ 6268679445’ , ‘Andheri Mumbai’),

( ‘9012003’ , ‘Roohi’ , ‘Sharma’ , ‘2002-02-28’ , ‘sharmaroohi@gmail.com’ , ‘6266578912’ , ‘jaipur’),

(‘9012004’ , ‘Arohi’ , ‘Tomar’ , ‘2002-02-14’ , ‘TomarArohi@gmail.com’ , ‘6261565419’ , ‘Gwalior’),

(‘9012005’ , ‘zoya’ , ‘Trivedi’ , ‘2001-09-14’ , ‘ZoyaTrivedi@gmail.com’ , ‘7566347812’ , ‘DELHI’),

(’9012006’ , ‘Arvi’ , ‘Sharma’ , ‘2001-09-03’ , ‘ArviSharma@gmail.com’ , ‘7564677812’ , ‘DELHI’),

(2, 'Emily', 'Chen', '1995-08-12', 'emily.chen@example.com', '+1-987-654-3210', '123 Main St, New York, NY 10001'),

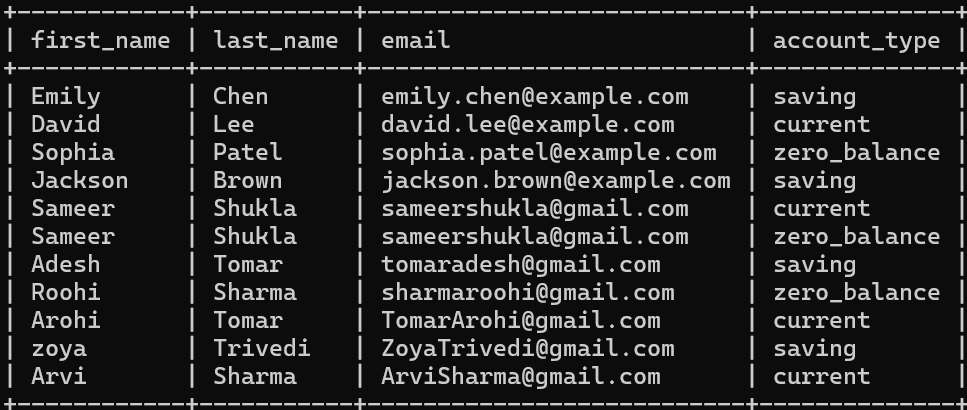
(3, 'David', 'Lee', '1980-02-28', 'david.lee@example.com', '+1-555-123-4567', '456 Elm St, Los Angeles, CA 90001'),

(4, 'Sophia', 'Patel', '1992-01-15', 'sophia.patel@example.com', '+1-111-222-3333', '789 Oak St, Chicago, IL 60601'),

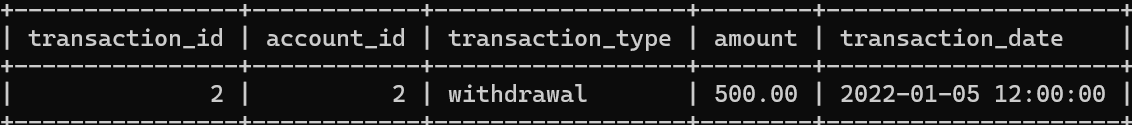
(5, 'Jackson', 'Brown', '1985-10-20', 'jackson.brown@example.com', '+1-444-555-6666', '321 Maple St, Houston, TX 77001');



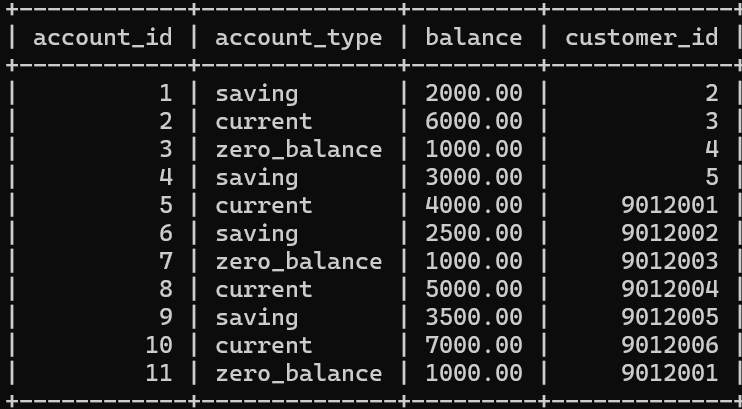
SELECTfirst\_name,last\_name,email,account\_type FROM customers,accounts WHERE customers.customer\_id = accounts.customer\_id;



SELECT\* FROM transactions WHERE account\_id = '2';



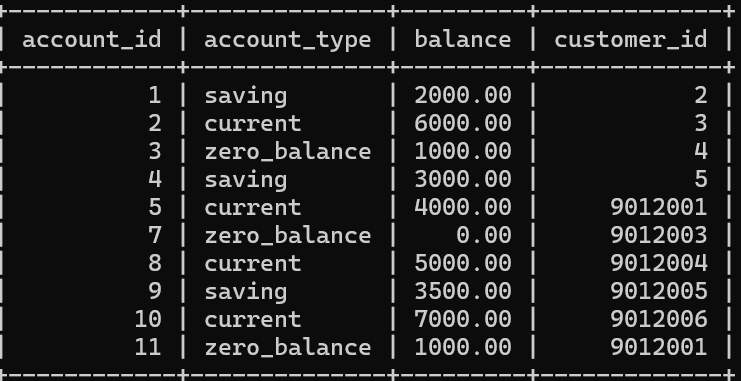
UPDATE accounts SET balance = balance + '1000' ;



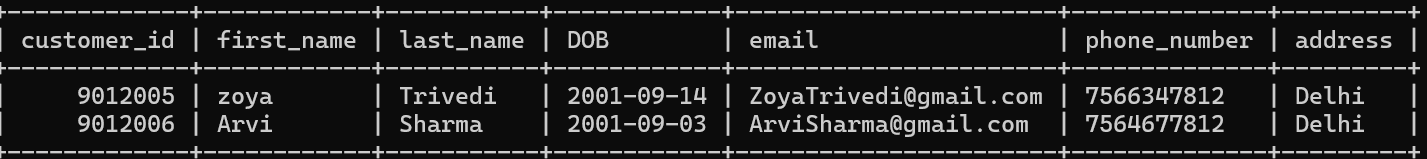
SELECT CONCAT(first\_name,' ',last\_name) AS full\_name FROM customers;



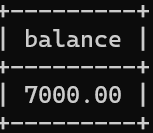
DELETE FROM accounts WHERE balance = '0' and account\_type = 'saving';



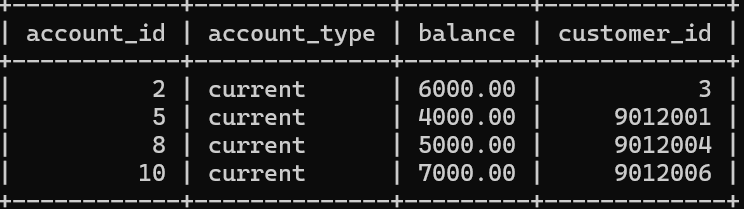
SELECT\* FROM customers WHERE address ='DELHI';



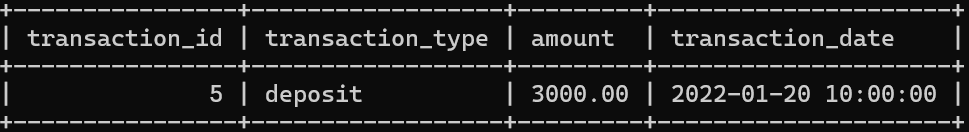
SELECT balance FROM accounts WHERE account\_id = '10';



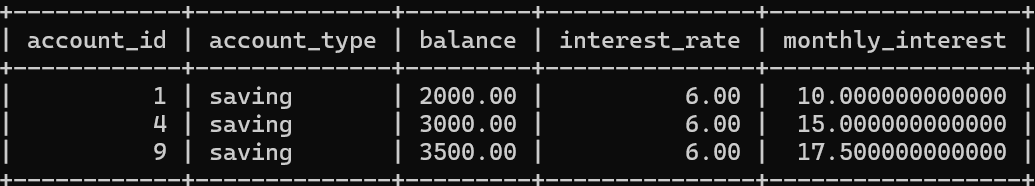
SELECT \* FROM accounts WHERE account\_type = 'current' and balance > '1000';



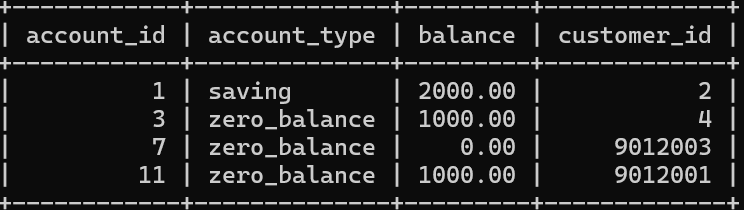
SELECT transaction\_id,transaction\_type,amount,transaction\_date FROM transactions WHERE account\_id ='5';



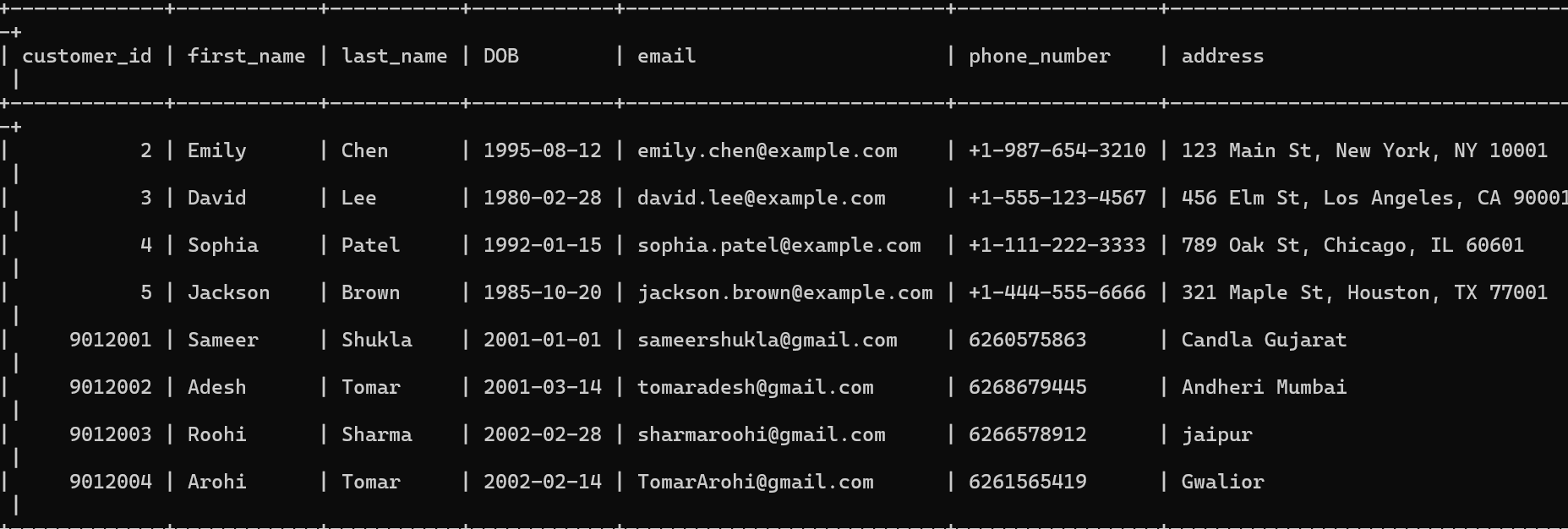
SELECT account\_id,account\_type,balance,interest\_rate,(balance \* interest\_rate / 100 / 12) AS monthly\_interest FROM accounts WHERE account\_type = 'saving';



SELECT account\_id, account\_type,balance,customer\_id FROM accounts WHERE balance < ‘2300’;

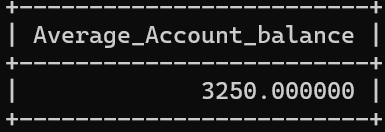


SELECT \* FROM customers WHERE address != 'DELHI';

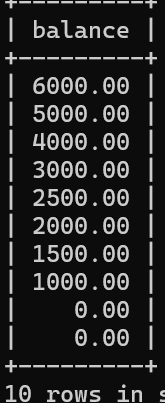


Task : 03

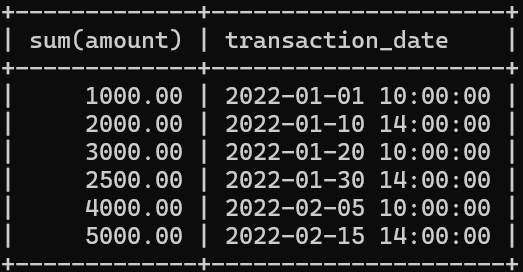
SELECT AVG(balance) Average\_Account\_balance FROM accounts;



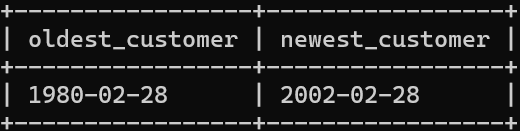
SELECT balance FROM accounts ORDER BY accounts DESC limit 10;



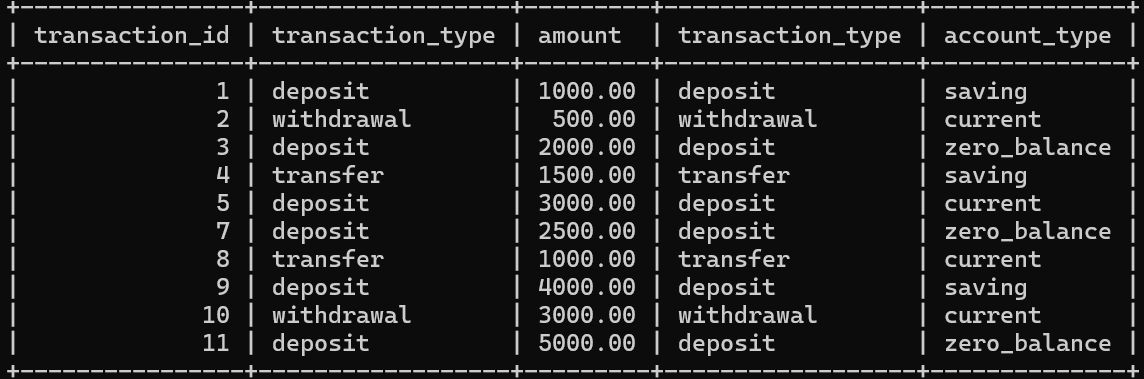
SELECT SUM(amount),transaction\_date FROM transactions WHERE transaction\_type = 'deposit' GROUP BY transaction\_date;



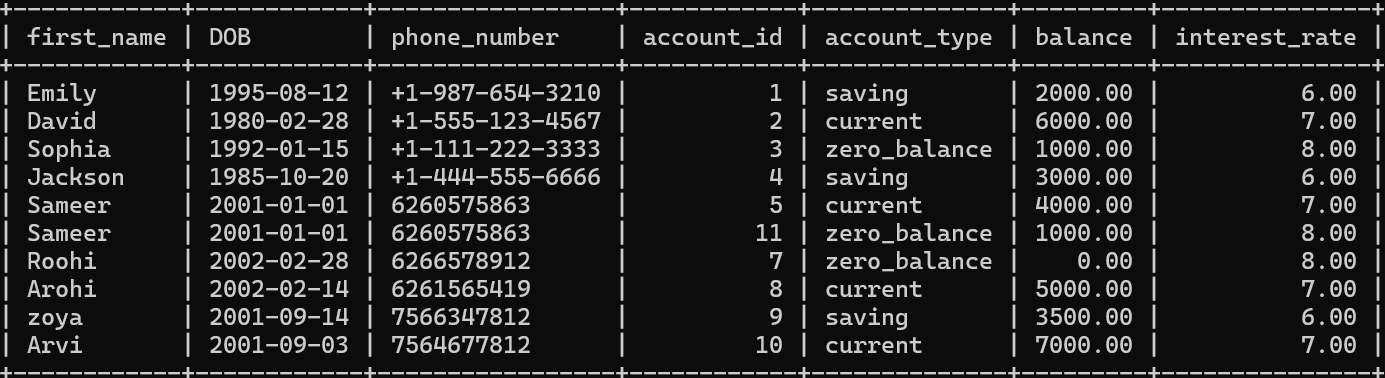
SELECT MIN(DOB) AS oldest\_customer,MAX(DOB) AS newest\_customer FROM customers;



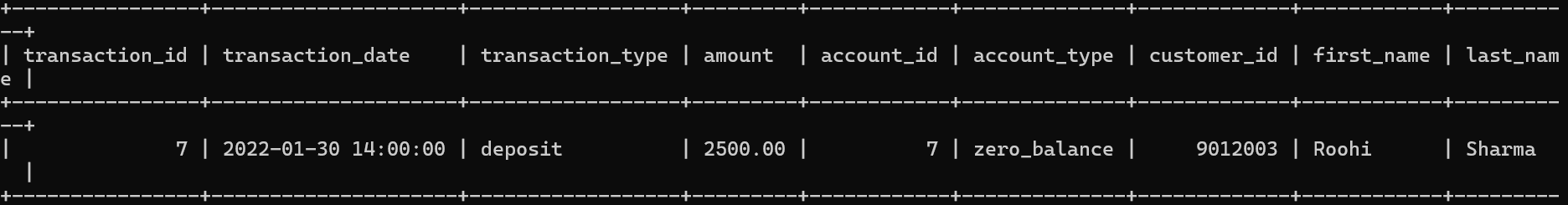
SELECT transaction\_id,transaction\_type,amount,transaction\_type,account\_type FROM transactions t, accounts a WHERE t.account\_id = a.account\_id;



SELECTfirst\_name ,DOB,phone\_number,account\_id,account\_type,balance,interest\_rate FROM customers c, accounts a WHERE c.customer\_id = a.customer\_id;



SELECTt.transaction\_id,t.transaction\_date,t.transaction\_type,t.amount,a.account\_id,a.account\_type,c.customer\_id,c.first\_name,c.last\_name FROM transactions t INNER JOIN accounts a ON t.account\_id = a.account\_id INNER JOIN customers c ON a.customer\_id = c.customer\_id WHERE a.account\_id = '7';



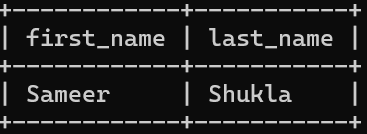
SELECTc.first\_name, c.last\_name

FROM customers c

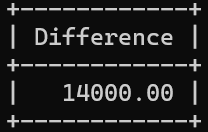
WHERE c.customer\_id IN (

SELECTa.customer\_id FROM accounts a GROUP BY a.customer\_id HAVING COUNT(a.account\_id) >1

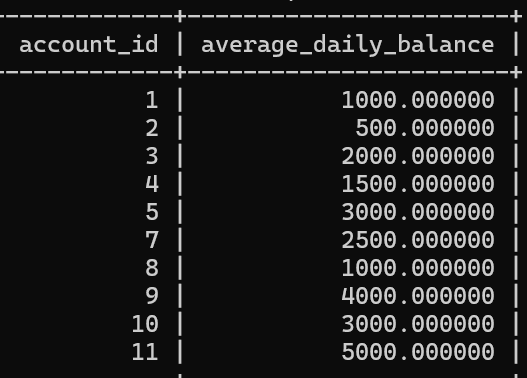
);



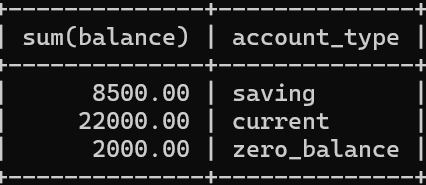
SELECT((SELECTSUM(amount) FROM transactions WHERE transaction\_type ='deposit') - (SELECTSUM(amount) FROM transactions WHERE transaction\_type='withdrawal')) as Difference;



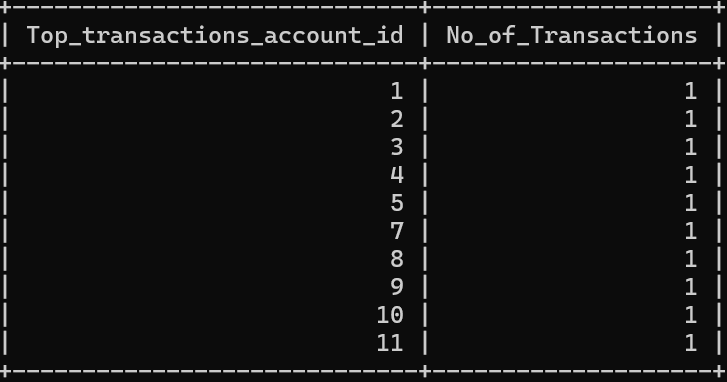
SELECTa.account\_id,AVG(d.daily\_balance) AS average\_daily\_balance FROM accounts a JOIN (SELECTt.account\_id,t.transaction\_date,SUM(t.amount) OVER (PARTITION BY t.account\_id ORDER BY t.transaction\_date) AS daily\_balance FROM transactions t WHERE t.transaction\_date BETWEEN '2022-01-01' AND '2022-12-31') d ON a.account\_id = d.account\_id GROUP BY a.account\_id;



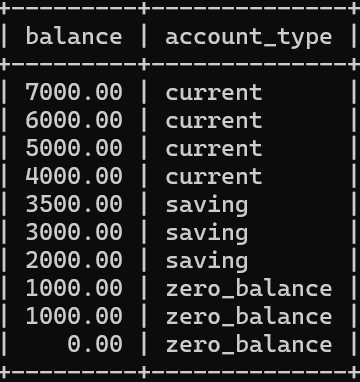
SELECTSUM(balance) , account\_type FROM accounts GROUP BY account\_type ;



SELECTaccount\_id as Top\_transactions\_account\_id ,count(transaction\_id) as No\_of\_Transactions FROM transactions GROUP BY account\_id ORDER BY No\_of\_Transactions DESC;



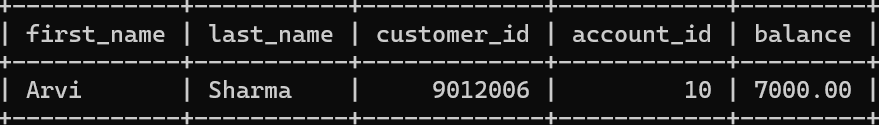
SELECTbalance, account\_type FROM accounts ORDER BY balance DESC;



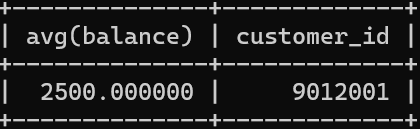
**SELECT**account\_id,transaction\_date,amount , **count**(\*) **as** Duplicate\_transactions **FROM** transactions **GROUP BY** account\_id,transaction\_date,amount **HAVING** Duplicate\_transactions >1;

Task : 04

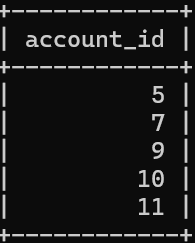
SELECTc.first\_name,c.last\_name,c.customer\_id,a.account\_id,a.balance FROM accounts a ,customers c WHERE a.customer\_id = c.customer\_id AND balance = (SELECTMAX(balance) FROM accounts);



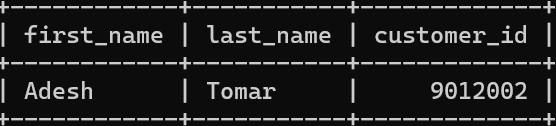
SELECTAVG(balance) , customer\_id FROM accounts GROUP BY (customer\_id) HAVING COUNT(account\_id) > 1;



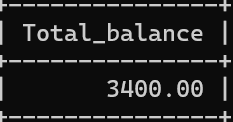
SELECTaccount\_id FROM transactions WHERE amount > (SELECTAVG(amount) FROM transactions);



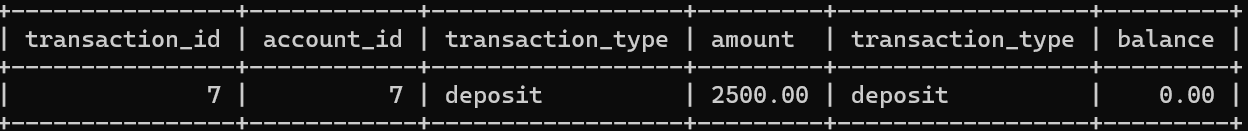
SELECTc.first\_name ,c.last\_name,c.customer\_id FROM customers c WHERE c.customer\_id NOT IN (SELECTa.customer\_id FROM accounts a , transactions t WHERE a.account\_id = t.account\_id);



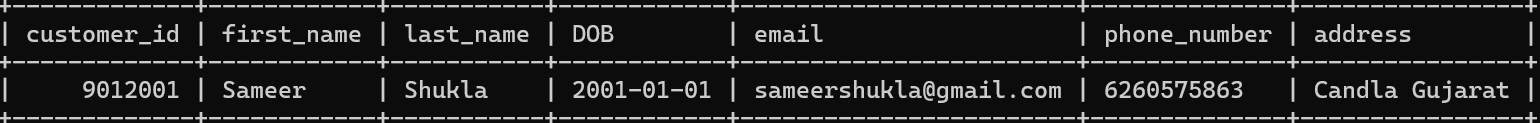
SELECTSUM(balance) AS Total\_balance FROM accounts WHERE account\_id NOT IN (SELECTaccount\_id FROM transactions);



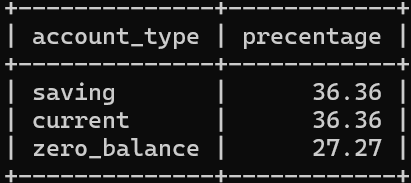
SELECTt.transaction\_id,t.account\_id ,t.transaction\_type,t.amount,t.transaction\_type , a.balance FROM transactions t, accounts a WHERE t.account\_id = a.account\_id AND balance = (SELECTMIN(balance) FROM accounts);



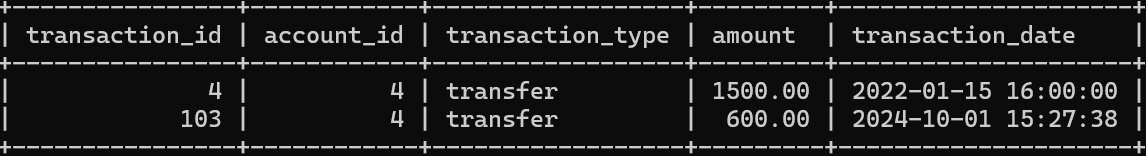
SELECT\* FROM customers WHERE customer\_id IN (SELECTcustomer\_id FROM accounts GROUP BY customer\_id HAVING COUNT(customer\_id)>1);



SELECTaccount\_type , ROUND((COUNT(\*)/(SELECTCOUNT(\*) FROM accounts))\*100,2) as precentage FROM accounts GROUP BY account\_type;



SELECT\* FROM transactions WHERE account\_id = (SELECTaccount\_id FROM accounts WHERE customer\_id = '5');



SELECTaccount\_type ,(SELECTSUM(balance) FROM accounts a WHERE a.account\_type = accounts.account\_type) AS total\_balance FROM accounts GROUP BY account\_type;

OR

SELECTaccount\_type, SUM(balance) FROM accounts GROUP BY account\_type;

