**WEEK\_2\_SOLUTIONS**

**PL\_SQL\_programming**

**Exercise 3: Stored Procedures**

Scenario 1: The bank needs to process monthly interest for all savings accounts.

* + Question: Write a stored procedure ProcessMonthlyInterest that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

* + Question: Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

Scenario 3: Customers should be able to transfer funds between their accounts.

* + Question: Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Step 1: Table Creation**

CREATE TABLE CUSTOMERS (

CUSTOMERID NUMBER PRIMARY KEY,

NAME VARCHAR2(100),

DOB DATE,

BALANCE NUMBER,

LASTMODIFIED DATE

);

CREATE TABLE ACCOUNTS (

ACCOUNTID NUMBER PRIMARY KEY,

CUSTOMERID NUMBER,

ACCOUNTTYPE VARCHAR2(20),

BALANCE NUMBER,

LASTMODIFIED DATE,

FOREIGN KEY ( CUSTOMERID )

REFERENCES CUSTOMERS ( CUSTOMERID )

);

CREATE TABLE TRANSACTIONS (

TRANSACTIONID NUMBER PRIMARY KEY,

ACCOUNTID NUMBER,

TRANSACTIONDATE DATE,

AMOUNT NUMBER,

TRANSACTIONTYPE VARCHAR2(10),

FOREIGN KEY ( ACCOUNTID )

REFERENCES ACCOUNTS ( ACCOUNTID )

);

CREATE TABLE LOANS (

LOANID NUMBER PRIMARY KEY,

CUSTOMERID NUMBER,

LOANAMOUNT NUMBER,

INTERESTRATE NUMBER,

STARTDATE DATE,

ENDDATE DATE,

FOREIGN KEY ( CUSTOMERID )

REFERENCES CUSTOMERS ( CUSTOMERID )

);

CREATE TABLE EMPLOYEES (

EMPLOYEEID NUMBER PRIMARY KEY,

NAME VARCHAR2(100),

POSITION VARCHAR2(50),

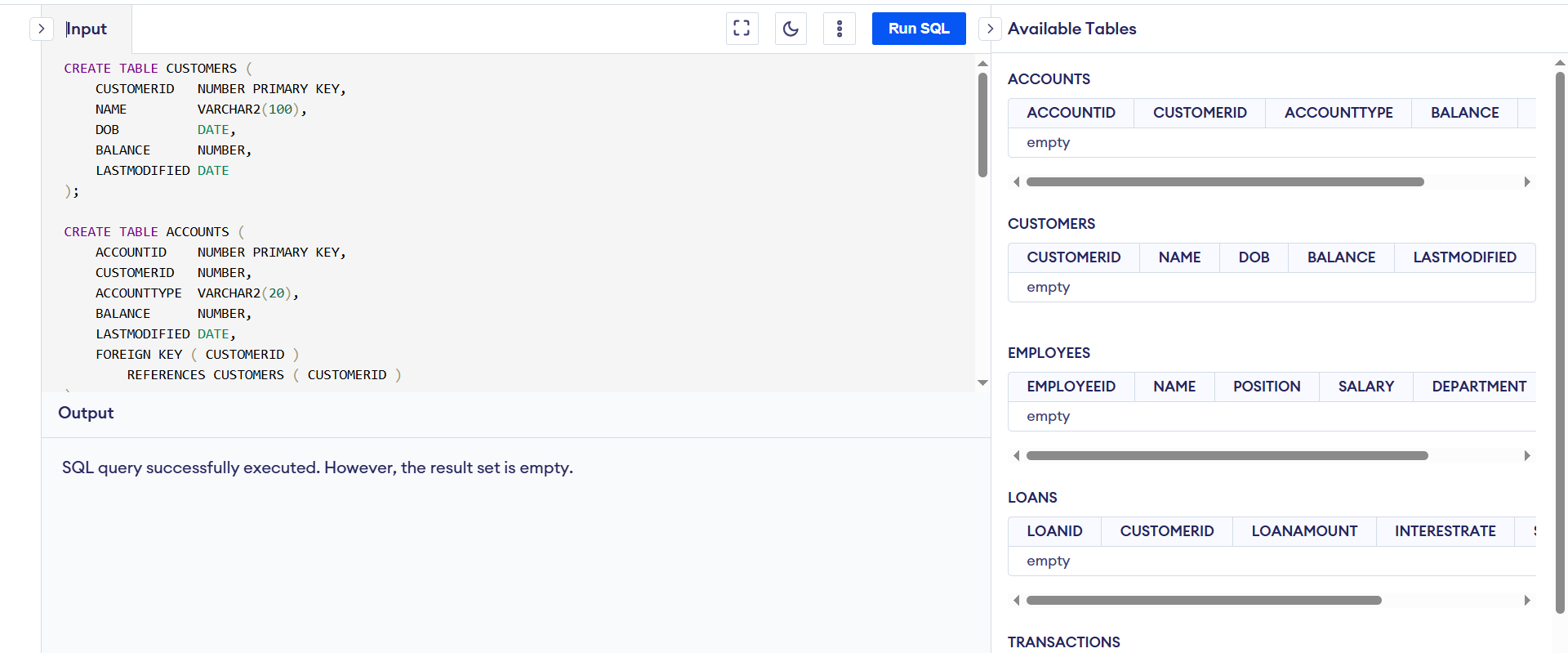
SALARY NUMBER,

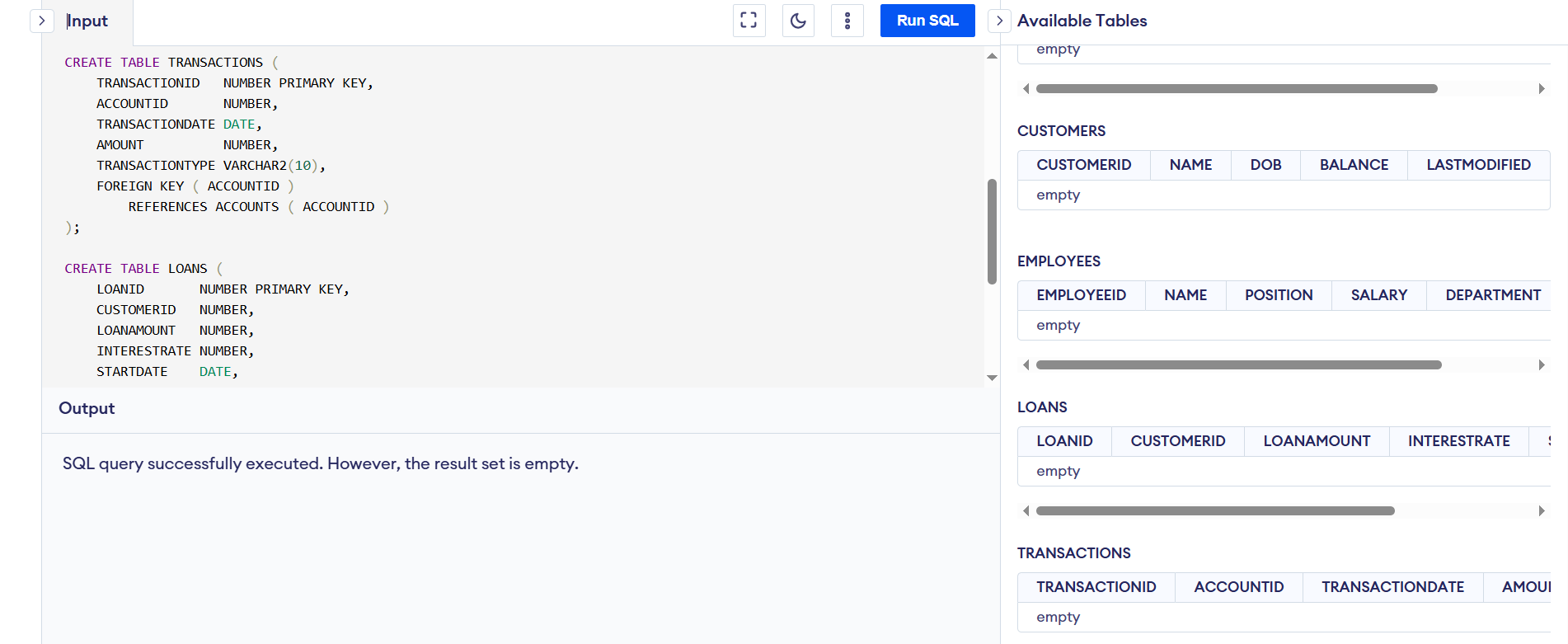
DEPARTMENT VARCHAR2(50),

HIREDATE DATE

**);**

**OUTPUT:**





**Step 2: Sample Data**

-- Insert into CUSTOMERS

INSERT INTO CUSTOMERS (CUSTOMERID, NAME, DOB, BALANCE, LASTMODIFIED)

VALUES (1, 'John Doe', '1985-05-15', 1000, DATE('now'));

INSERT INTO CUSTOMERS (CUSTOMERID, NAME, DOB, BALANCE, LASTMODIFIED)

VALUES (2, 'Jane Smith', '1990-07-20', 1500, DATE('now'));

-- Insert into ACCOUNTS

INSERT INTO ACCOUNTS (ACCOUNTID, CUSTOMERID, ACCOUNTTYPE, BALANCE, LASTMODIFIED)

VALUES (1, 1, 'Savings', 1000, DATE('now'));

INSERT INTO ACCOUNTS (ACCOUNTID, CUSTOMERID, ACCOUNTTYPE, BALANCE, LASTMODIFIED)

VALUES (2, 2, 'Checking', 1500, DATE('now'));

-- Insert into TRANSACTIONS

INSERT INTO TRANSACTIONS (TRANSACTIONID, ACCOUNTID, TRANSACTIONDATE, AMOUNT, TRANSACTIONTYPE)

VALUES (1, 1, DATE('now'), 200, 'Deposit');

INSERT INTO TRANSACTIONS (TRANSACTIONID, ACCOUNTID, TRANSACTIONDATE, AMOUNT, TRANSACTIONTYPE)

VALUES (2, 2, DATE('now'), 300, 'Withdrawal');

-- Insert into LOANS

INSERT INTO LOANS (LOANID, CUSTOMERID, LOANAMOUNT, INTERESTRATE, STARTDATE, ENDDATE)

VALUES (1, 1, 5000, 5, DATE('now'), DATE('now', '+60 months'));

-- Insert into EMPLOYEES

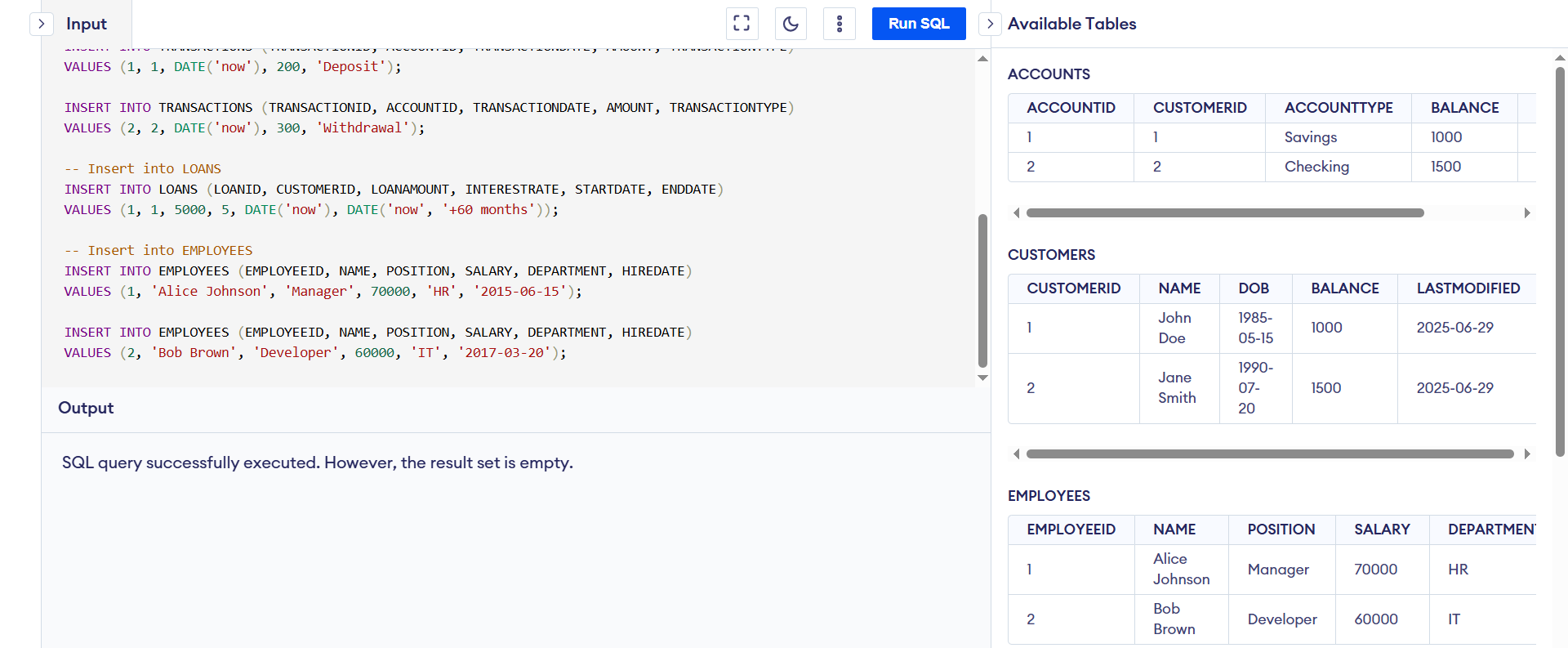
INSERT INTO EMPLOYEES (EMPLOYEEID, NAME, POSITION, SALARY, DEPARTMENT, HIREDATE)

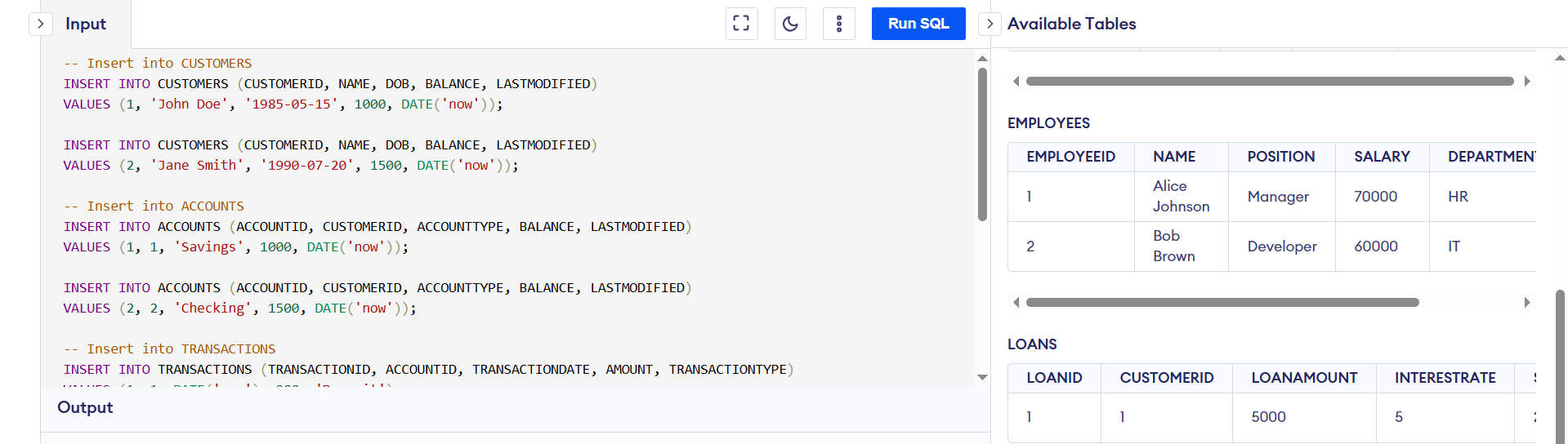
VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', '2015-06-15');

INSERT INTO EMPLOYEES (EMPLOYEEID, NAME, POSITION, SALARY, DEPARTMENT, HIREDATE)

VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', '2017-03-20');

**OUTPUT:**





**-- SCENARIO 1**

Write a stored procedure ProcessMonthlyInterest that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

UPDATE ACCOUNTS

SET BALANCE = BALANCE \* 1.01,

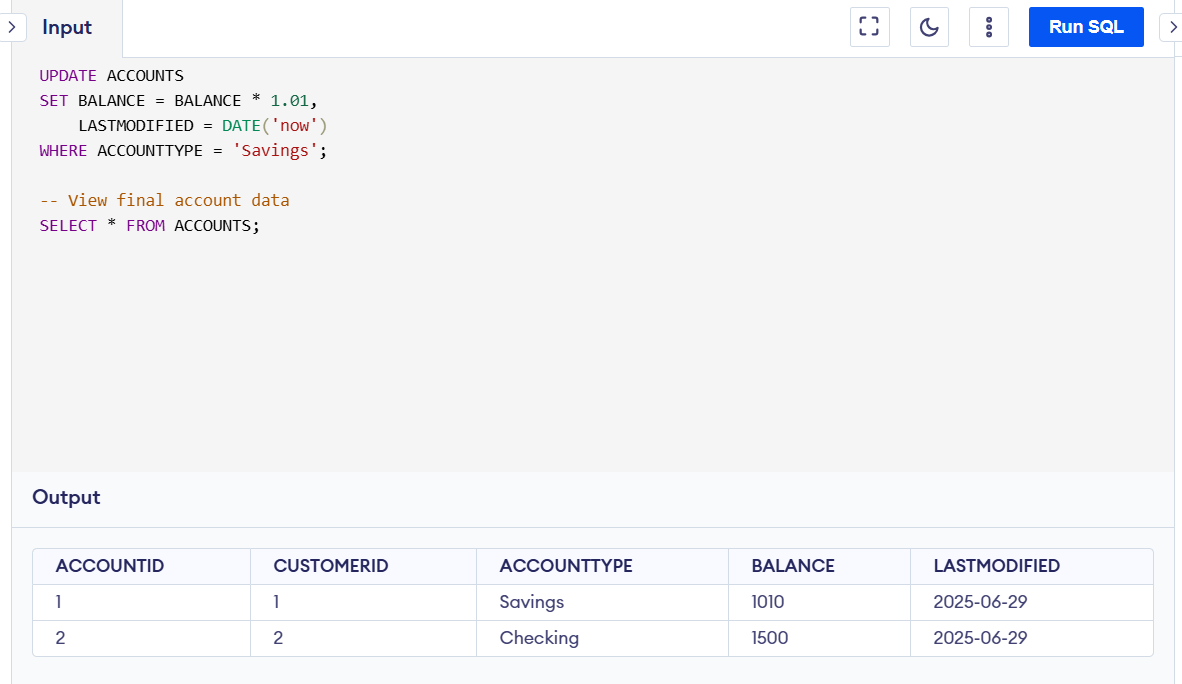
LASTMODIFIED = DATE('now')

WHERE ACCOUNTTYPE = 'Savings';

-- View final account data

SELECT \* FROM ACCOUNTS;

**OUTPUT:**



**-- SCENARIO 2**

Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**CODE:**

UPDATE EMPLOYEES

SET SALARY = SALARY \* 1.03,

HIREDATE = DATE('now')

WHERE DEPARTMENT = 'HR';

UPDATE EMPLOYEES

SET SALARY = SALARY \* 1.05,

HIREDATE = DATE('now')

WHERE DEPARTMENT = 'IT';

Select \* from EMPLOYEES;

**OUTPUT:**



**-- SCENARIO 3**

Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CODE:

-- Step 1: Check from account has enough balance

SELECT BALANCE FROM ACCOUNTS WHERE ACCOUNTID = 1;

-- (Assume BALANCE >= 100)

-- Step 2: Deduct from source account

UPDATE ACCOUNTS

SET BALANCE = BALANCE - 100,

LASTMODIFIED = DATE('now')

WHERE ACCOUNTID = 1;

-- Step 3: Add to destination account

UPDATE ACCOUNTS

SET BALANCE = BALANCE + 100,

LASTMODIFIED = DATE('now')

WHERE ACCOUNTID = 2;

-- Step 4: Check updated data

SELECT \* FROM ACCOUNTS;

**OUTPUT:**

