**Control Structures :**

The bank wants to apply a discount to loan interest rates for customers above60years

**Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

CODE :

DROP TABLE IF EXISTS customers;

CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name TEXT,

age INTEGER

);

INSERT INTO customers (customer\_id, name, age) VALUES (1, 'ram', 45);

INSERT INTO customers (customer\_id, name, age) VALUES (2, 'raju', 62);

INSERT INTO customers (customer\_id, name, age) VALUES (3, 'sravani', 70);

INSERT INTO customers (customer\_id, name, age) VALUES (4, 'harshi', 59);

INSERT INTO customers (customer\_id, name, age) VALUES (5, 'bhanu', 40);

ALTER TABLE customers ADD loan\_interest\_rate REAL;

---Assumed interest rate

UPDATE customers SET loan\_interest\_rate = 10.5;

--applying 1% discount

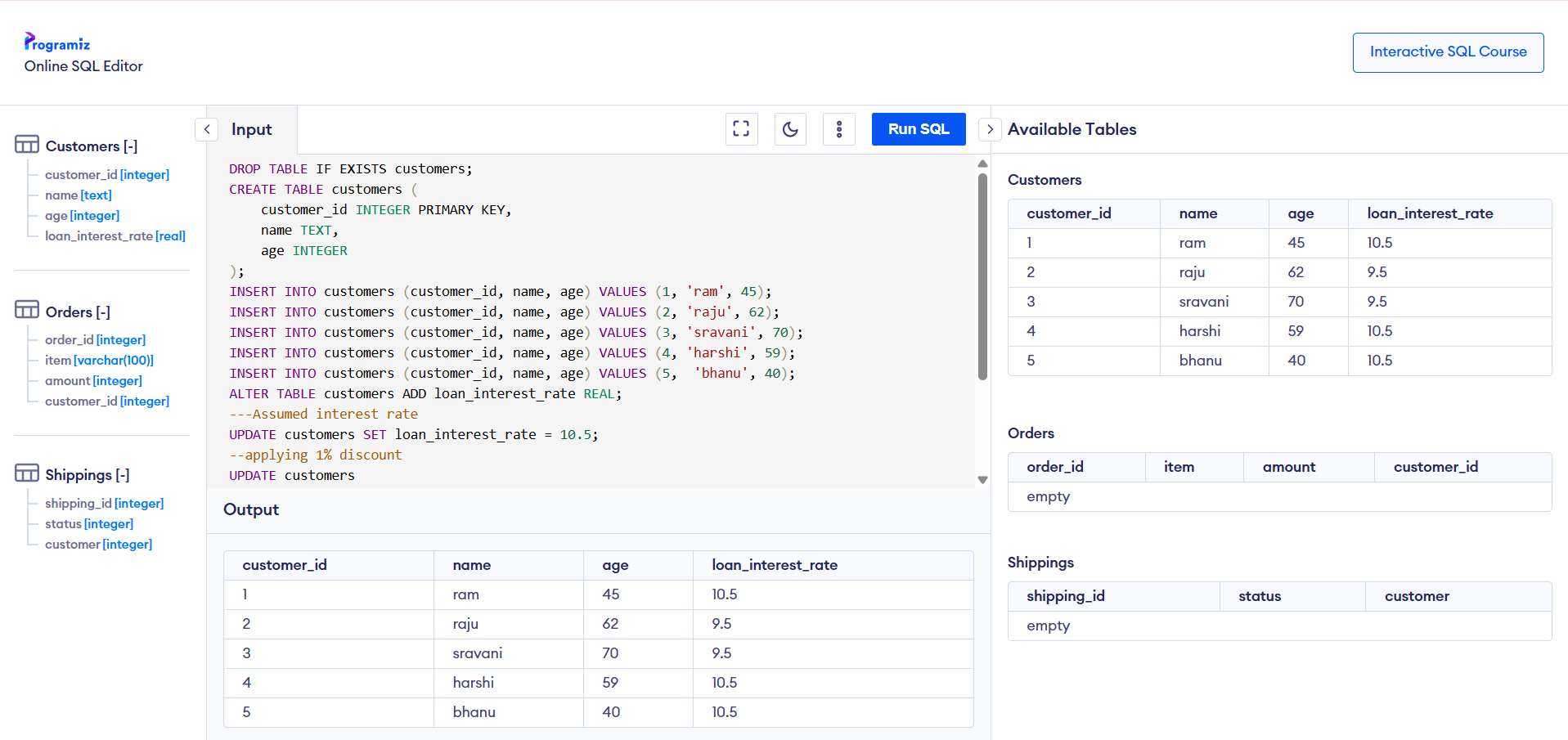
UPDATE customers

SET loan\_interest\_rate = loan\_interest\_rate - 1

WHERE age > 60;

SELECT \* FROM customers;

OUTPUT :



A customer can be promoted to VIP status based on their balance.

* **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUEfor those with a balance over $10,000.

CODE :

DROP TABLE IF EXISTS customers;

CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name TEXT,

balance INTEGER,

isVIP TEXT

);

INSERT INTO customers VALUES (1, 'ram', 12000, NULL);

INSERT INTO customers VALUES (2, 'raju', 9500, NULL);

INSERT INTO customers VALUES (3, 'sravani', 15000, NULL);

INSERT INTO customers VALUES (4, 'harshi', 6000, NULL);

INSERT INTO customers VALUES (5, 'bhanu', 25000, NULL);

--vip is applied

UPDATE customers

SET isVIP = 'TRUE'

WHERE balance > 10000;

---if not vip

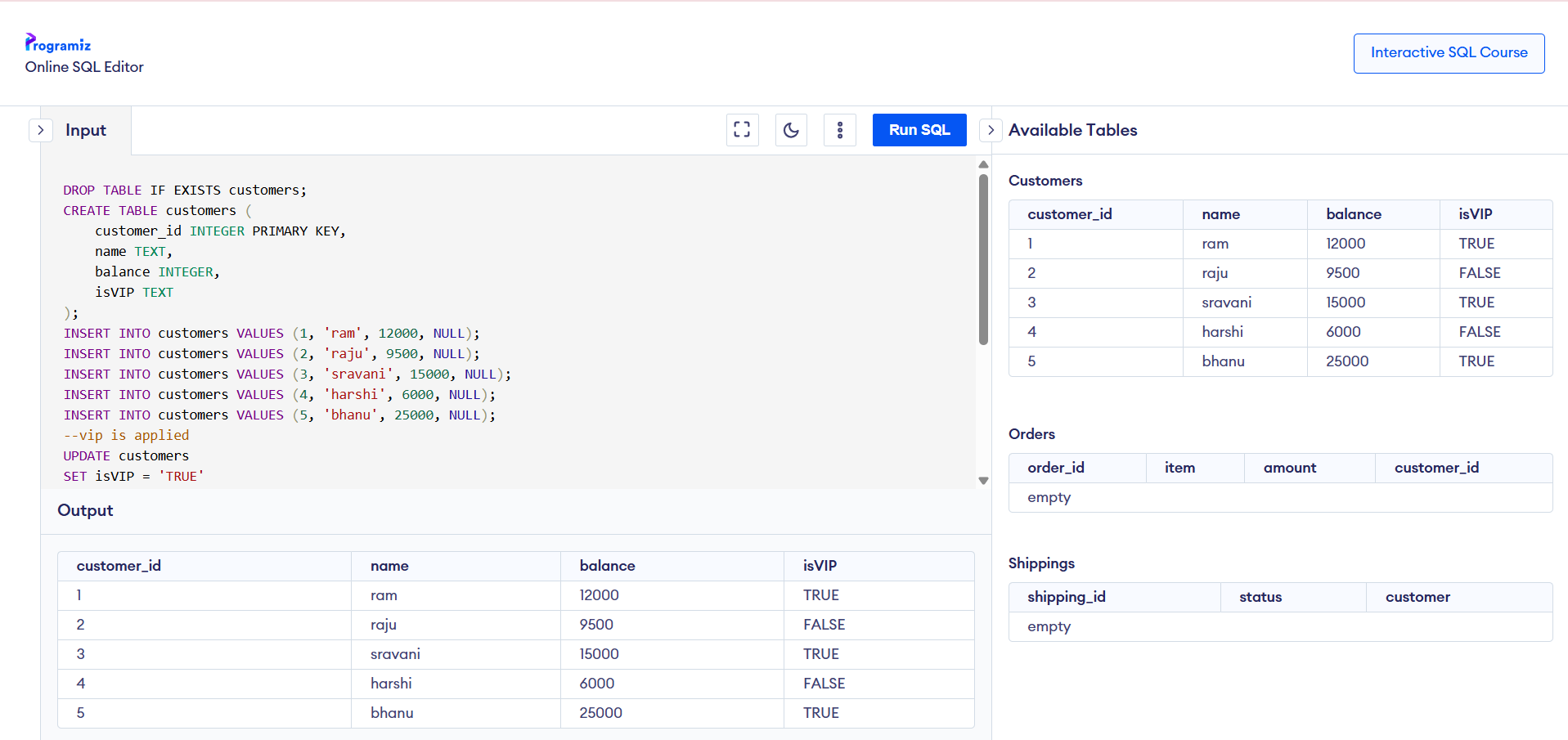
UPDATE customers

SET isVIP = 'FALSE'

WHERE balance <= 10000 OR balance IS NULL;

SELECT \* FROM customers;

OUTPUT :



The bank wants to send reminders to customers whose loans are due within the next 30 days.

**Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

CODE :

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(50)

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

due\_date DATE,

amount NUMBER,

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

);

--sample names

INSERT INTO customers VALUES (1, 'Ram');

INSERT INTO customers VALUES (2, 'Ravi');

INSERT INTO customers VALUES (3, 'Bhanu');

INSERT INTO loans VALUES (201, 1, SYSDATE + 5, 10000);

INSERT INTO loans VALUES (202, 2, SYSDATE + 15, 15000);

INSERT INTO loans VALUES (203, 3, SYSDATE + 40, 20000);

COMMIT;

SET SERVEROUTPUT ON;

DECLARE

CURSOR loan\_cursor IS

SELECT l.loan\_id, l.due\_date, l.amount, c.name

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30;

v\_loan\_id loans.loan\_id%TYPE;

v\_due\_date loans.due\_date%TYPE;

v\_amount loans.amount%TYPE;

v\_name customers.name%TYPE;

BEGIN

OPEN loan\_cursor;

LOOP

FETCH loan\_cursor INTO v\_loan\_id, v\_due\_date, v\_amount, v\_name;

EXIT WHEN loan\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Hello ' || v\_name || ', your Loan ID ' || v\_loan\_id ||

' of ₹' || v\_amount || ' is due on ' || TO\_CHAR(v\_due\_date, 'DD-Mon-YYYY')

);

END LOOP;

CLOSE loan\_cursor;

END;

/

OUTPUT :

