**Exercise 1: Control Structures**

**Scenario 1: The bank wants to apply a discount to loan interest rates for customers above 60 years old.**

**Code:**

CREATE TABLE Customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

isVIP CHAR(1) DEFAULT 'N'

);

CREATE TABLE Loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

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

);

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (1, 'John Doe', 65, 15000, 'Y');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (2, 'Jane Smith', 55, 12000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (3, 'Robert Brown', 70, 20000, 'Y');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (4, 'Emily White', 45, 9000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (5, 'Michael Green', 62, 18000, 'Y');

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (101, 1, 5, TO\_DATE('2024-12-31', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (102, 2, 6, TO\_DATE('2024-11-30', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (103, 3, 7, TO\_DATE('2024-10-15', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (104, 4, 8, TO\_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (105, 5, 5, TO\_DATE('2024-08-25', 'YYYY-MM-DD'));

BEGIN

FOR rec IN (

SELECT loan\_id

FROM Loans

WHERE customer\_id IN (

SELECT customer\_id FROM Customers WHERE age > 60

)

) LOOP

UPDATE Loans

SET interest\_rate = interest\_rate - 1

WHERE loan\_id = rec.loan\_id;

END LOOP;

END;

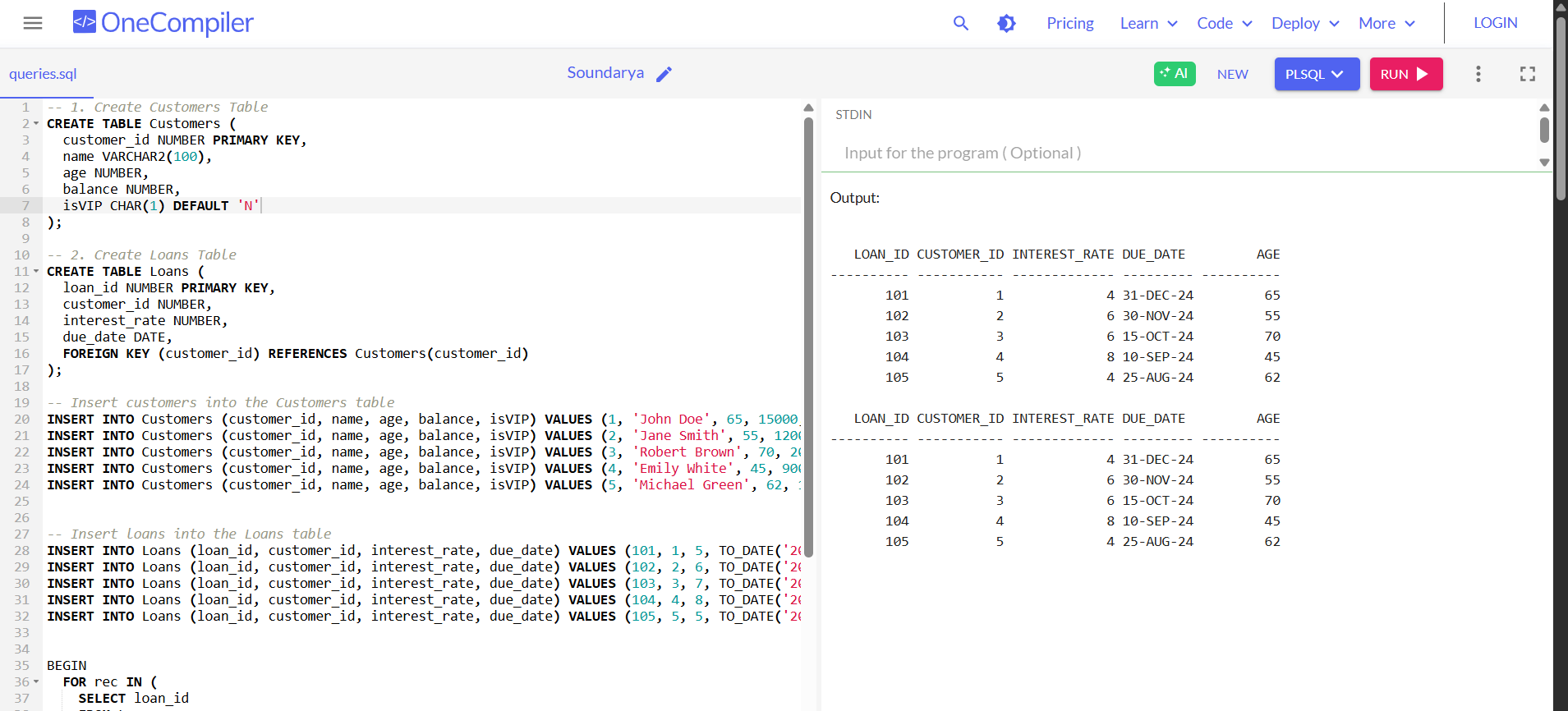
/

SELECT l.loan\_id, l.customer\_id, l.interest\_rate, l.due\_date, c.age

FROM Loans l

JOIN Customers c ON l.customer\_id = c.customer\_id;

**Output:**



**Scenario 2: A customer can be promoted to VIP status based on their balance.**

**Code:**

CREATE TABLE Customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

isVIP CHAR(1) DEFAULT 'N'

);

CREATE TABLE Loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

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

);

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (1, 'Alice Williams', 72, 25000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (2, 'Bob Johnson', 64, 18000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (3, 'Charlie Davis', 68, 220, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (4, 'Diana Martin', 42, 1050, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (5, 'Ethan Clark', 38, 13000, 'N');

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (101, 1, 6, TO\_DATE('2025-01-15', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (102, 2, 7, TO\_DATE('2025-02-28', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (103, 3, 8, TO\_DATE('2025-03-10', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (104, 4, 5, TO\_DATE('2025-04-05', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (105, 5, 4, TO\_DATE('2025-05-20', 'YYYY-MM-DD'));

BEGIN

FOR rec IN (

SELECT customer\_id, balance

FROM Customers

) LOOP

IF rec.balance > 10000 THEN

UPDATE Customers

SET isVIP = 'Y'

WHERE customer\_id = rec.customer\_id;

END IF;

END LOOP;

COMMIT;

END;

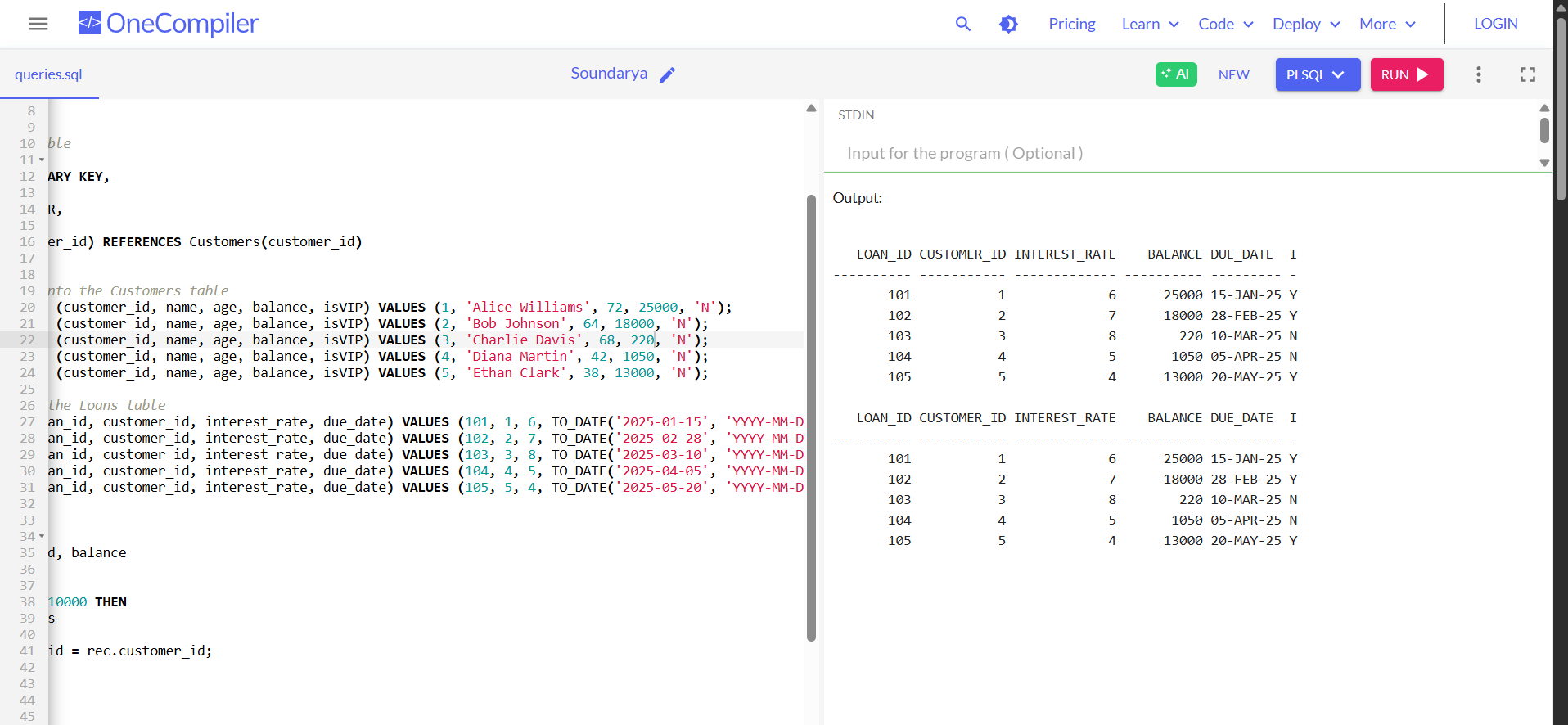
/

SELECT l.loan\_id, l.customer\_id, l.interest\_rate, c.balance,l.due\_date, c.isVIP

FROM Loans l

JOIN Customers c ON l.customer\_id = c.customer\_id;

**Output:**



**Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.**

**Code:**

CREATE TABLE Customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

isVIP CHAR(1) DEFAULT 'N'

);

CREATE TABLE Loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

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

);

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (1, 'Alice Williams', 72, 25000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (2, 'Bob Johnson', 64, 18000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (3, 'Charlie Davis', 68, 22000, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (4, 'Diana Martin', 42, 10500, 'N');

INSERT INTO Customers (customer\_id, name, age, balance, isVIP) VALUES (5, 'Ethan Clark', 38, 13000, 'N');

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (101, 1, 6, TO\_DATE('2025-01-15', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (102, 2, 7, TO\_DATE('2025-02-28', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (103, 3, 8, TO\_DATE('2025-03-10', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (104, 4, 5, TO\_DATE('2025-04-05', 'YYYY-MM-DD'));

INSERT INTO Loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES (105, 5, 4, TO\_DATE('2025-05-20', 'YYYY-MM-DD'));

DECLARE

v\_due\_date DATE := SYSDATE + 30;

BEGIN

FOR rec IN (

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

FROM Loans l

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

WHERE l.due\_date <= v\_due\_date

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || rec.name ||

', your loan (Loan ID: ' || rec.loan\_id ||

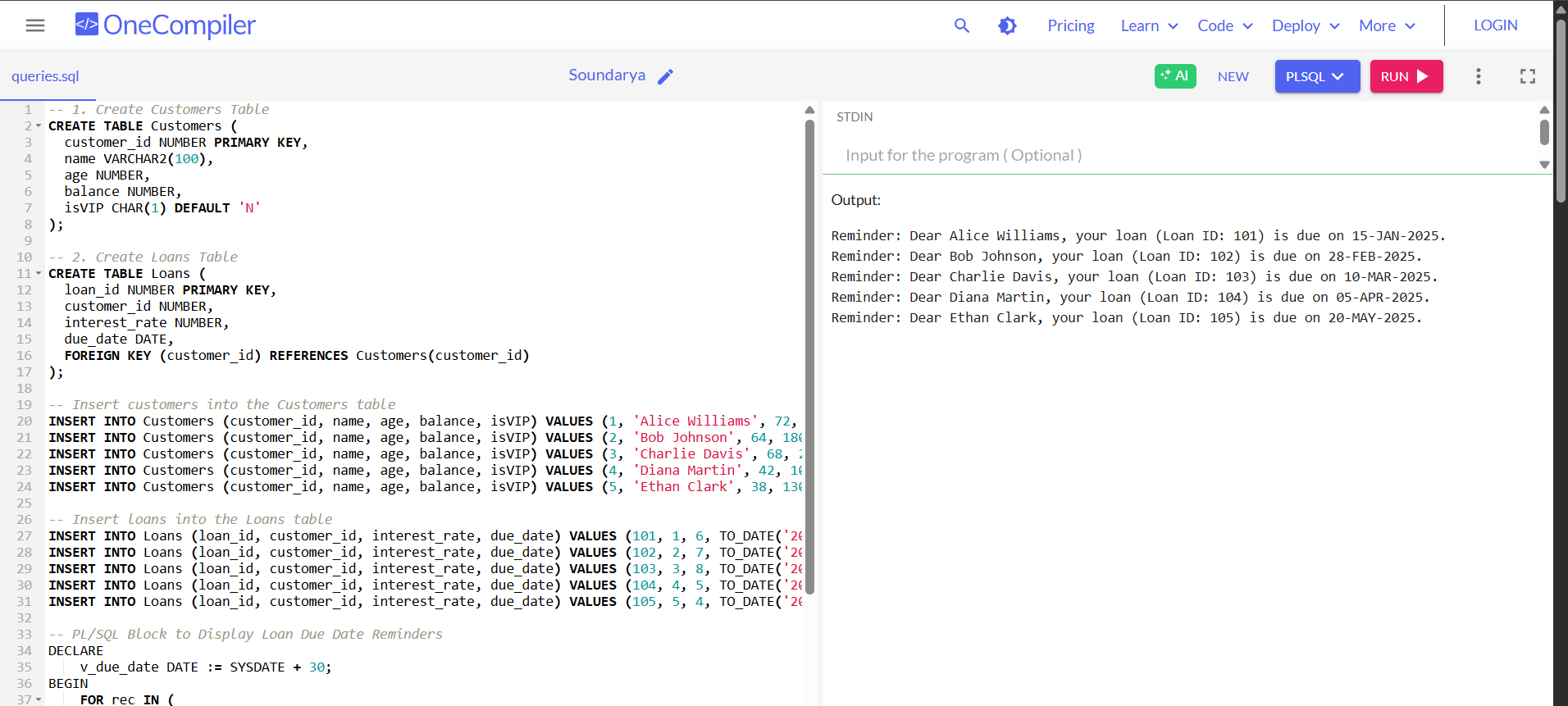
') is due on ' || TO\_CHAR(rec.due\_date, 'DD-MON-YYYY') || '.');

END LOOP;

END;

/

**Output:**



**Exercise 3: Stored Procedures**

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

**Code:**

CREATE TABLE accounts (

acc\_id NUMBER PRIMARY KEY,

acc\_name VARCHAR2(100),

bal NUMBER(10, 2),

type CHAR(1) -- 's' for savings, 'c' for checking

);

INSERT INTO accounts (acc\_id, acc\_name, bal, type) VALUES (1, 'Alice Williams', 10000.00, 's');

INSERT INTO accounts (acc\_id, acc\_name, bal, type) VALUES (2, 'Bob Johnson', 15000.00, 's');

INSERT INTO accounts (acc\_id, acc\_name, bal, type) VALUES (3, 'Charlie Davis', 20000.00, 's');

INSERT INTO accounts (acc\_id, acc\_name, bal, type) VALUES (4, 'Diana Martin', 5000.00, 'c');

INSERT INTO accounts (acc\_id, acc\_name, bal, type) VALUES (5, 'Ethan Clark', 8000.00, 's');

CREATE OR REPLACE PROCEDURE PMI AS

BEGIN

-- Update balance by adding 1% interest for savings accounts

UPDATE accounts

SET bal = bal + (bal \* 0.01)

WHERE type = 's';

COMMIT;

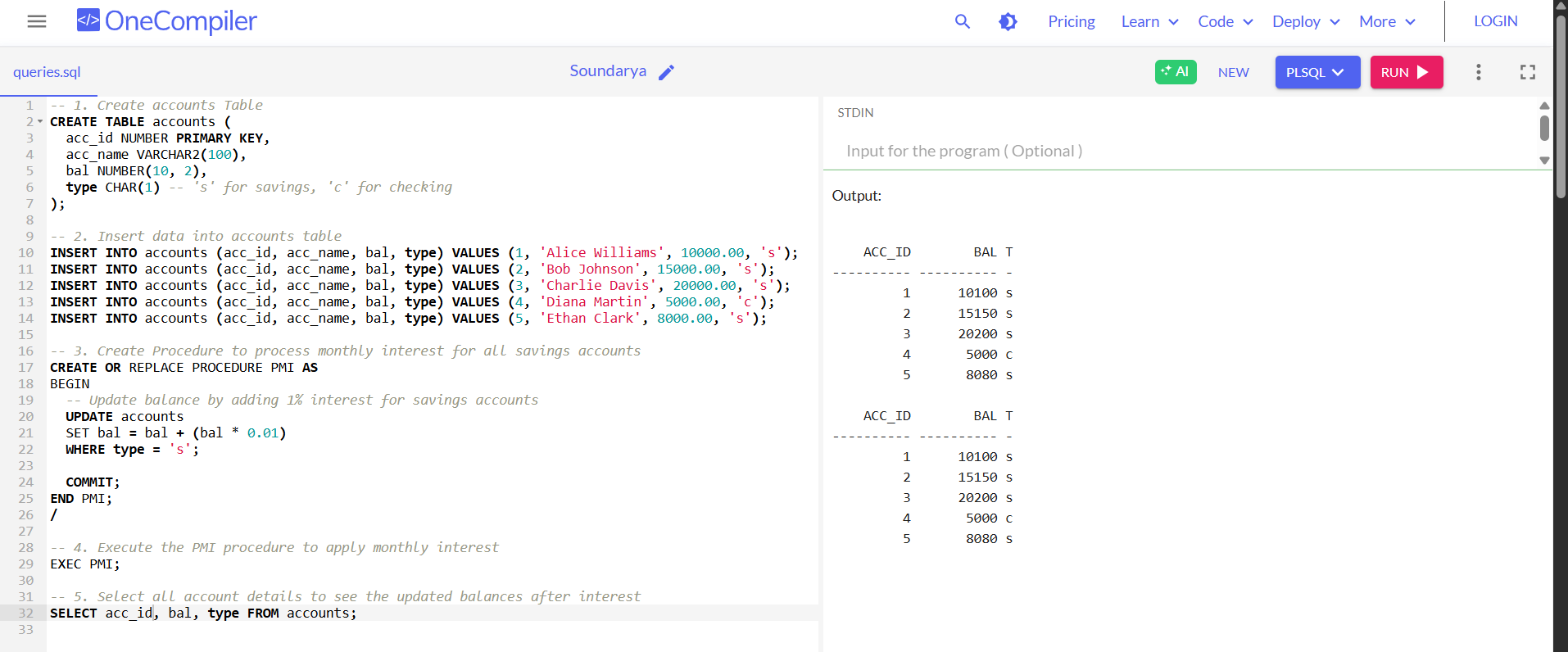
END PMI;

/

EXEC PMI;

SELECT acc\_id, bal, type FROM accounts;

**Output:**



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

**performance.**

**Code:**

CREATE TABLE emp (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(100),

sal NUMBER(10, 2),

dept INT

);

INSERT INTO emp (emp\_id, emp\_name, sal, dept) VALUES (1, 'Doe', 5000.00, 101);

INSERT INTO emp (emp\_id, emp\_name, sal, dept) VALUES (2, 'Smith', 90000.00, 102);

INSERT INTO emp (emp\_id, emp\_name, sal, dept) VALUES (3, 'Robert', 555000.00, 101);

INSERT INTO emp (emp\_id, emp\_name, sal, dept) VALUES (4, 'Emily', 9800.00, 103);

INSERT INTO emp (emp\_id, emp\_name, sal, dept) VALUES (5, 'Michael', 670000.00, 102);

CREATE OR REPLACE PROCEDURE UEB(d IN INT, b IN NUMBER) AS

BEGIN

-- Update salary based on department and percentage increase

UPDATE emp

SET sal = sal + (sal \* b / 100)

WHERE dept = d;

COMMIT;

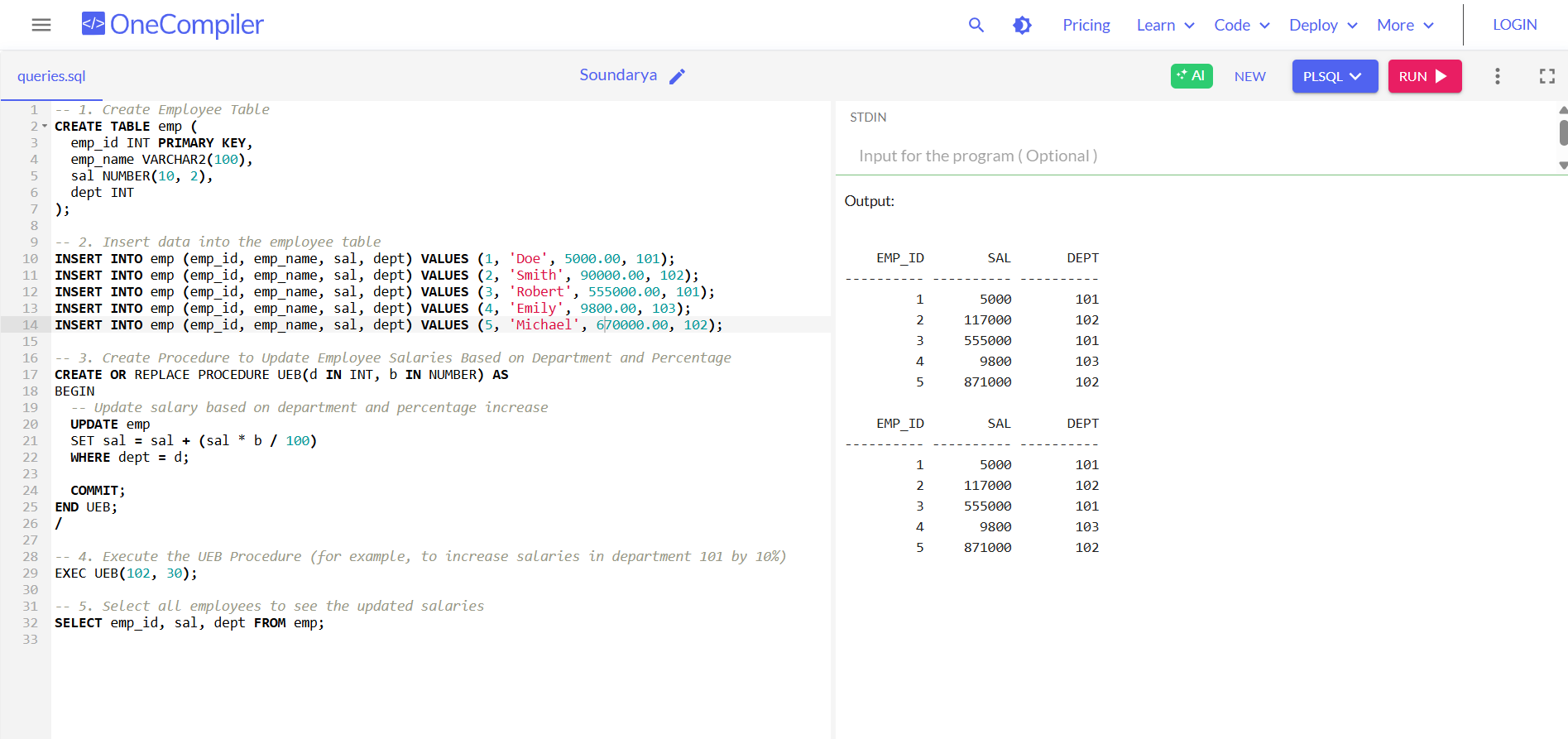
END UEB;

/

EXEC UEB(102, 30);

SELECT emp\_id, sal, dept FROM emp;

**Output:**

****

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

**Code:**

CREATE TABLE acc (

id INT PRIMARY KEY,

name VARCHAR2(100),

bal DECIMAL(10, 2)

);

INSERT INTO acc (id, name, bal) VALUES (1, 'Alice Williams', 10000.00);

INSERT INTO acc (id, name, bal) VALUES (2, 'Bob Johnson', 15000.00);

INSERT INTO acc (id, name, bal) VALUES (3, 'Charlie Davis', 20000.00);

INSERT INTO acc (id, name, bal) VALUES (4, 'Diana Martin', 5000.00);

INSERT INTO acc (id, name, bal) VALUES (5, 'Ethan Clark', 8000.00);

CREATE OR REPLACE PROCEDURE TF(f IN INT, t IN INT, a IN DECIMAL) AS

b DECIMAL(10, 2);

BEGIN

SELECT bal INTO b FROM acc WHERE id = f;

IF b >= a THEN

UPDATE acc SET bal = bal - a WHERE id = f;

UPDATE acc SET bal = bal + a WHERE id = t;

END IF;

COMMIT;

END TF;

/

EXEC TF(1, 2, 1000);

SELECT id, bal FROM acc;

**Output:**

