**WEEK-2**

**Exercise 1: Control Structures**

**Code:**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DateOfBirth DATE,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

RateOfInterest NUMBER,

DueDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Customers VALUES (1, 'Puja', TO\_DATE('1950-01-15', 'YYYY-MM-DD'), 10000, 'FALSE');

INSERT INTO Customers VALUES (2, 'Taylor', TO\_DATE('1985-06-20', 'YYYY-MM-DD'), 8000, 'FALSE');

INSERT INTO Customers VALUES (3, 'Yaasmin', TO\_DATE('1948-11-05', 'YYYY-MM-DD'), 17000, 'FALSE');

INSERT INTO Loans VALUES (101, 1, 10.5, SYSDATE + 20);

INSERT INTO Loans VALUES (102, 2, 8.0, SYSDATE + 40);

INSERT INTO Loans VALUES (103, 3, 13.0, SYSDATE + 10);

Select\* from CUSTOMERS;

Select\* from Loans;

COMMIT;

/\*Scenario-1\*/

BEGIN

FOR cust IN (

SELECT c.CustomerID, l.LoanID

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE MONTHS\_BETWEEN(SYSDATE, c.DateOfBirth) / 12 > 60

)

LOOP

UPDATE Loans

SET RateOfInterest = RateOfInterest - 1

WHERE LoanID = cust.LoanID;

END LOOP;

COMMIT;

END;

/

Select\* from Loans;

/\*Scenario-2\*/

BEGIN

FOR cust IN (SELECT CustomerID FROM Customers WHERE Balance > 10000)

LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

END LOOP;

COMMIT;

END;

/

Select\* from CUSTOMERS;

/\*Scenario-3\*/

BEGIN

FOR rec IN (

SELECT c.Name, l.DueDate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: ' || rec.Name || ', your loan is due on ' || TO\_CHAR(rec.DueDate, 'DD-MON-YYYY'));

END LOOP;

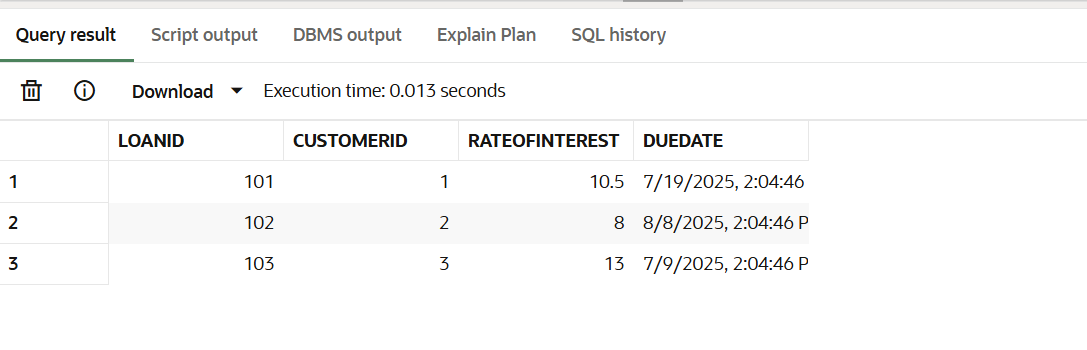
END;

/

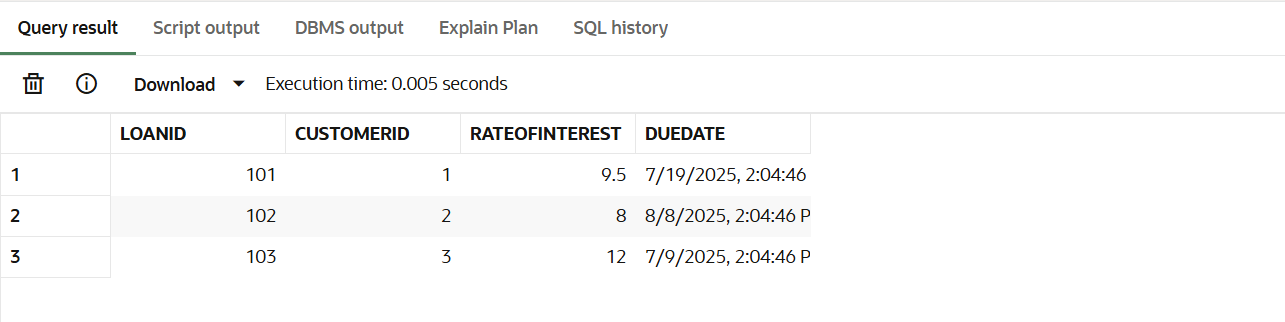
**Output:**

**Scenario-1**

**Before applying the procedure**

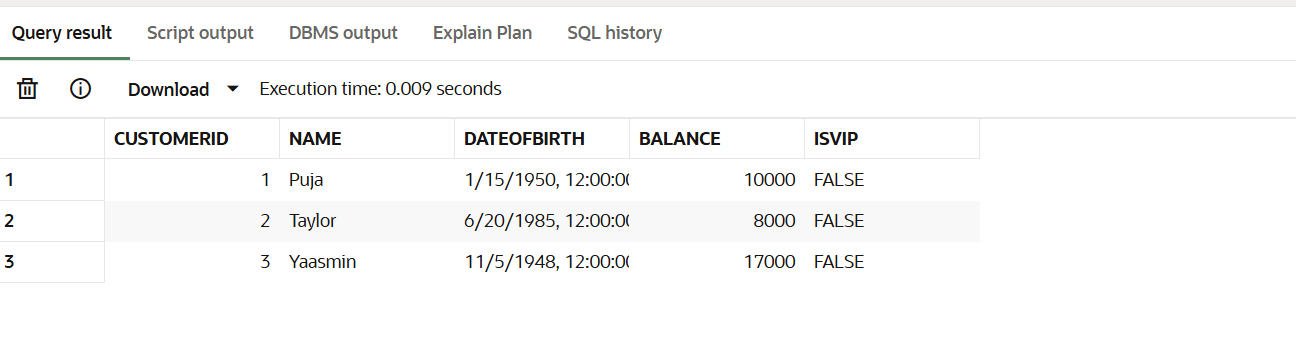
****

**After applying the procedure**

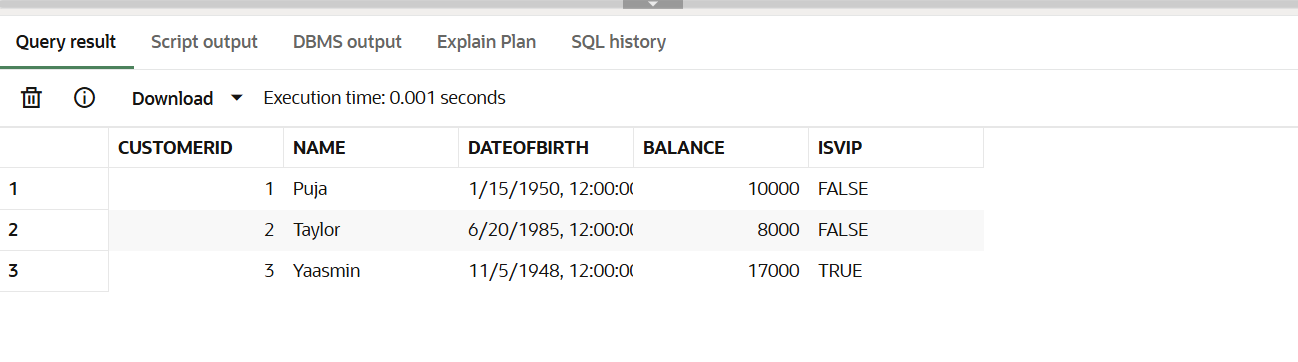
****

**Scenario-2**

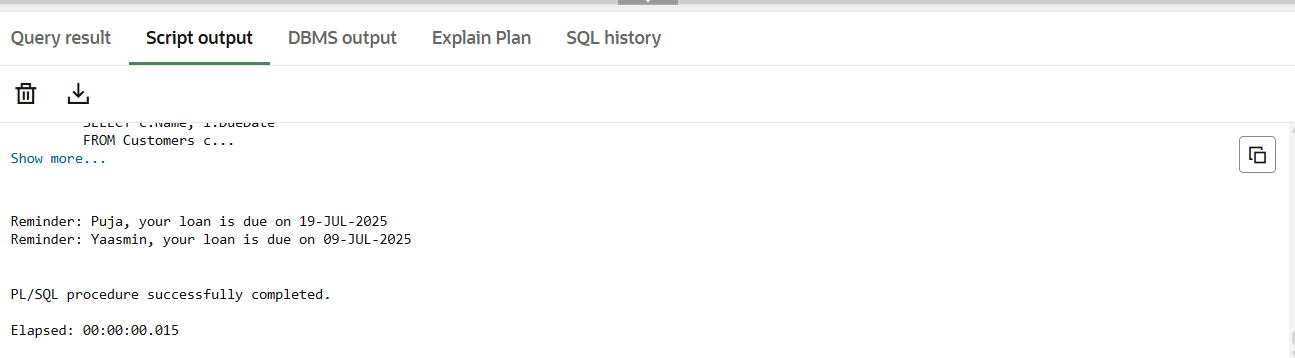
**Before applying the procedure**

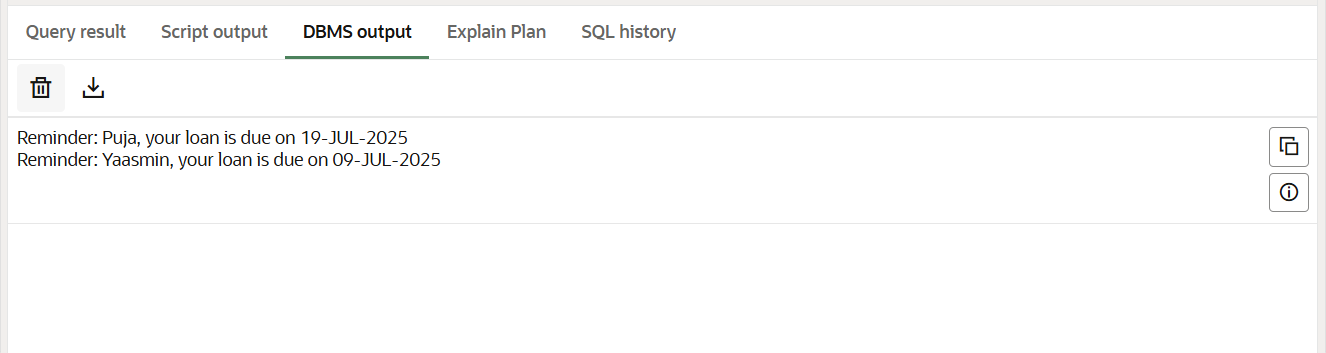
****

**After applying the procedure**

****

**Scenario-3**

****

****

**Exercise 3: Stored Procedures**

**Code:**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

Balance NUMBER,

AccountType VARCHAR2(20)

);

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER

);

INSERT INTO Accounts VALUES (101, 1, 10000, 'Savings');

INSERT INTO Accounts VALUES (102, 2, 8000, 'Checking');

INSERT INTO Accounts VALUES (103, 1, 15000, 'Savings');

Select\* from ACCOUNTS;

INSERT INTO Employees VALUES (1, 'Puja', 'Sales', 40000);

INSERT INTO Employees VALUES (2, 'Taylor', 'Sales', 55000);

INSERT INTO Employees VALUES (3, 'Yaasmin', 'Marketing', 65000);

Select\* from EMPLOYEES;

COMMIT;

--Scenario -1

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

COMMIT;

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

Select\* from ACCOUNTS;

--Scenario -2

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

deptName IN VARCHAR2,

bonusPercent IN NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* bonusPercent / 100)

WHERE Department = deptName;

COMMIT;

END;

/

BEGIN

UpdateEmployeeBonus('Sales', 10);

END;

/

Select\* from EMPLOYEES;

--Scenario -3

CREATE OR REPLACE PROCEDURE TransferFunds (

fromAccountID IN NUMBER,

toAccountID IN NUMBER,

amount IN NUMBER

) IS

insufficient\_funds EXCEPTION;

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = fromAccountID FOR UPDATE;

IF v\_balance < amount THEN

RAISE insufficient\_funds;

END IF;

UPDATE Accounts

SET Balance = Balance - amount

WHERE AccountID = fromAccountID;

UPDATE Accounts

SET Balance = Balance + amount

WHERE AccountID = toAccountID;

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient balance in source account.');

ROLLBACK;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: ' || SQLERRM);

ROLLBACK;

END;

/

BEGIN

TransferFunds(101, 102, 2000);

END;

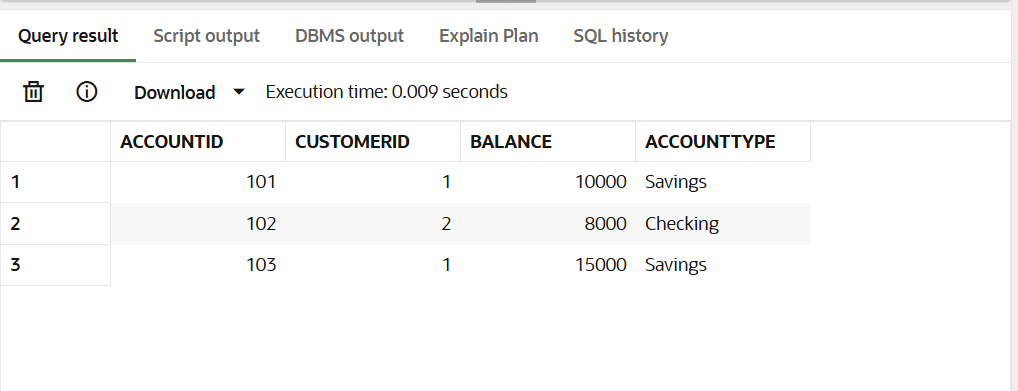
/

Select\* from Accounts;

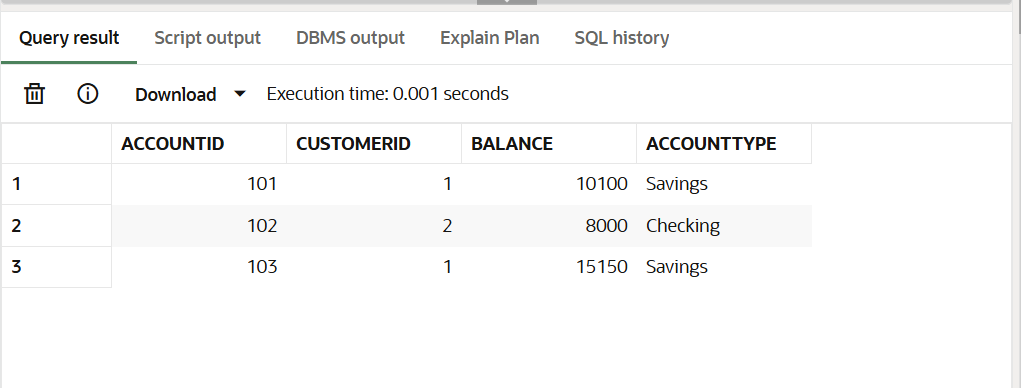
**Output:**

**Scenario-1**

**Before applying the procedure:**

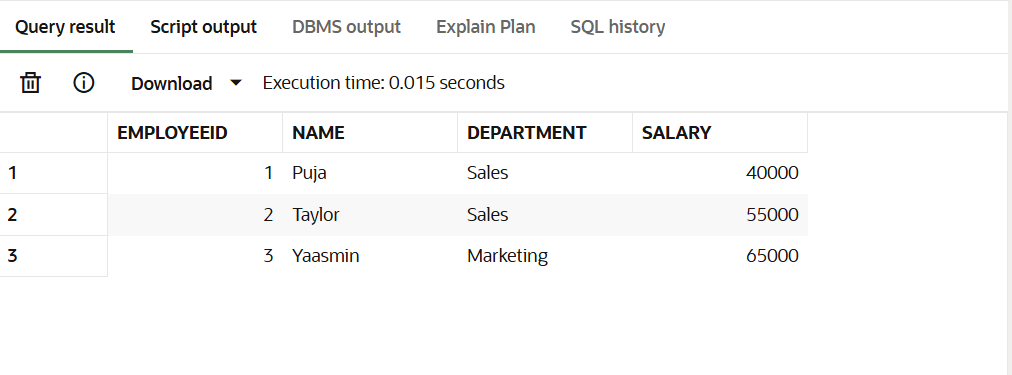
****

**After applying the procedure:**

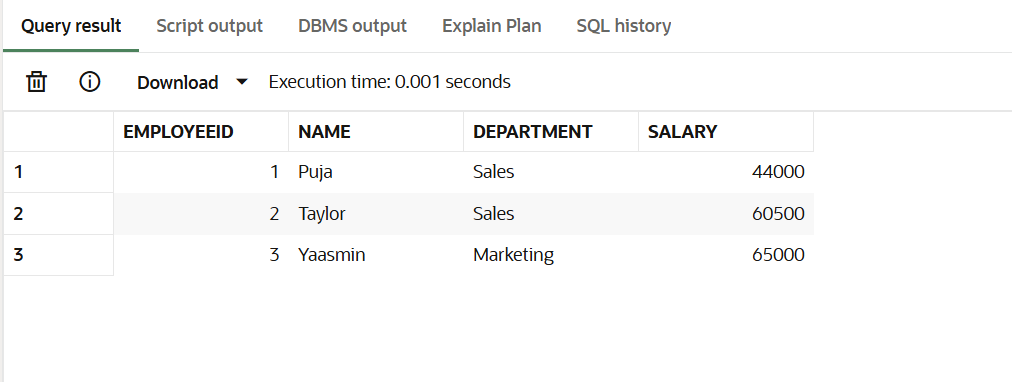
****

**Scenario-2**

**Before applying the procedure:**

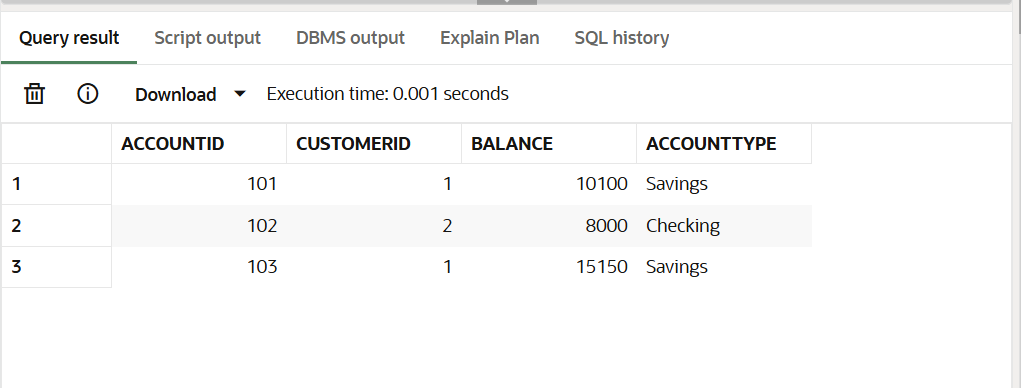
****

**After applying the procedure:**

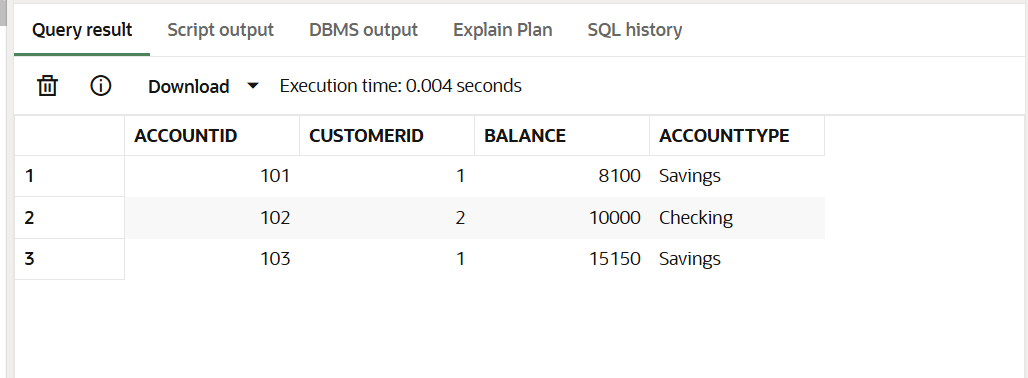
****

**Scenario-3**

**Before applying the procedure:**

****

**After applying the procedure:**

****