**Scenario 1: Apply 1% discount to loan interest rates for customers above 60 years old**

BEGIN

FOR c IN (SELECT l.LoanID, l.InterestRate, c.DOB

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID)

LOOP

IF MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12 > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

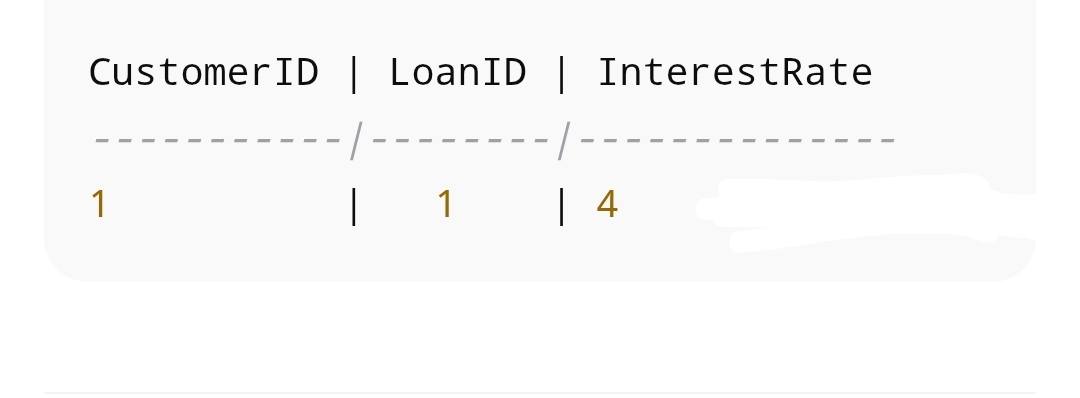
WHERE LoanID = c.LoanID;

END IF;

END LOOP;

COMMIT;

END;



**Scenario 2: Promote customers with balance > $10,000 to VIP**

BEGIN

FOR c IN (SELECT CustomerID, Balance FROM Customers)

LOOP

IF c.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

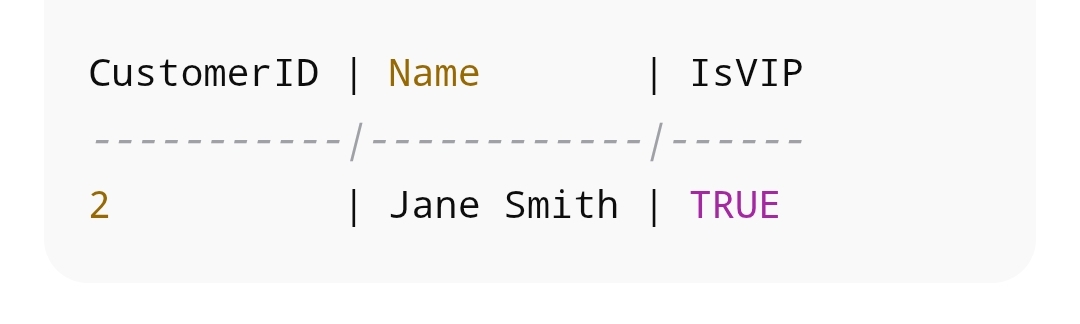
WHERE CustomerID = c.CustomerID;

END IF;

END LOOP;

COMMIT;

END;



#### **Scenario 3: Send reminders for loans due in next 30 days**

#### **BEGIN**

FOR l IN (SELECT l.LoanID, l.EndDate, c.Name

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

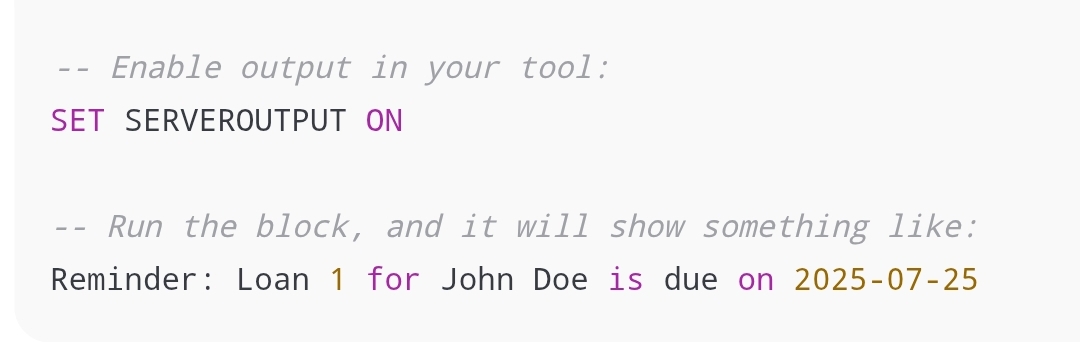
WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30)

LOOP

DBMS\_OUTPUT.PUT\_LINE(‘Reminder: Loan ‘ || l.LoanID || ‘ for ‘ || l.Name || ‘ is due on ‘ || TO\_CHAR(l.EndDate, ‘YYYY-MM-DD’));

END LOOP;

END;



#### **Scenario 1: Process monthly interest for savings accounts**

BEGIN

FOR l IN (SELECT l.LoanID, l.EndDate, c.Name

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

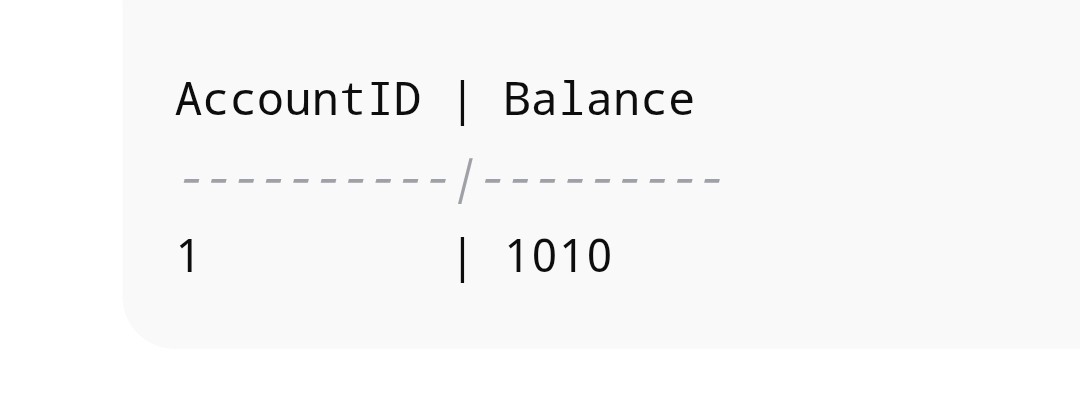
WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ' || l.LoanID || ' for ' || l.Name || ' is due on ' || TO\_CHAR(l.EndDate, 'YYYY-MM-DD'));

END LOOP;

END;

**Scenario 2: Bonus scheme for employees**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_dept IN VARCHAR2, p\_bonus\_pct IN NUMBER) IS

BEGIN

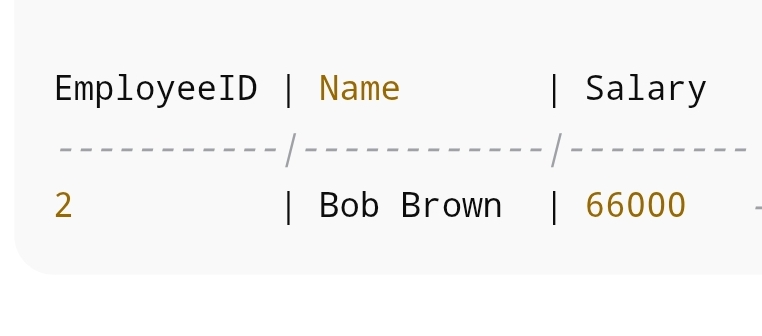
UPDATE Employees

SET Salary = Salary + (Salary \* p\_bonus\_pct / 100)

WHERE Department = p\_dept;

COMMIT;

END;



**Scenario 3: Transfer funds between accounts**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_source\_account IN NUMBER,

p\_dest\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_source\_account FOR UPDATE;

IF v\_balance >= p\_amount THEN

UPDATE Accounts SET Balance = Balance - p\_amount WHERE AccountID = p\_source\_account;

UPDATE Accounts SET Balance = Balance + p\_amount WHERE AccountID = p\_dest\_account;

COMMIT;

ELSE

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account');

END IF;

END;

