SUPERSET ID:6383982  
  
PL/SQL:1  
  
SELECT CustomerID,

       Name,

       DOB,

       TRUNC(MONTHS\_BETWEEN(SYSDATE, DOB) / 12) AS Age

FROM Customers;

A screenshot of a computer

AI-generated content may be incorrect.

SET SERVEROUTPUT ON;

DECLARE

CURSOR cust\_cursor IS

SELECT c.CustomerID, c.Name, l.LoanID, l.InterestRate,

TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12) AS Age

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12) > 60;

v\_new\_rate Loans.InterestRate%TYPE;

BEGIN

FOR cust IN cust\_cursor LOOP

v\_new\_rate := cust.InterestRate - (cust.InterestRate \* 0.01);

UPDATE Loans

SET InterestRate = v\_new\_rate

WHERE LoanID = cust.LoanID;

DBMS\_OUTPUT.PUT\_LINE(' Discount applied to ' || cust.Name ||

' (CustomerID: ' || cust.CustomerID || ') - ' ||

'Old Rate: ' || cust.InterestRate || ', New Rate: ' || ROUND(v\_new\_rate, 2));

END LOOP;

COMMIT;

END;

/  
  
A screenshot of a computer program

AI-generated content may be incorrect.  
  
  
  
SET SERVEROUTPUT ON;

DECLARE

CURSOR vip\_cursor IS

SELECT CustomerID, Name, Balance

FROM Customers;

BEGIN

FOR cust IN vip\_cursor LOOP

IF cust.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('VIP ' || cust.Name || ' (CustomerID: ' || cust.CustomerID ||

') marked as VIP. Balance: ' || cust.Balance);

END IF;

END LOOP;

COMMIT;

END;

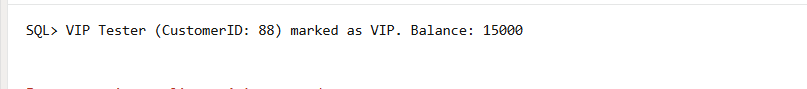
/  
  
  
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (88, 'VIP Tester', TO\_DATE('1990-01-01', 'YYYY-MM-DD'), 15000, SYSDATE);

COMMIT;

SELECT CustomerID, Name, Balance, IsVIP FROM Customers WHERE IsVIP = 'TRUE';  
A screenshot of a computer

AI-generated content may be incorrect.



SET SERVEROUTPUT ON;

DECLARE

    CURSOR due\_soon\_cursor IS

        SELECT l.LoanID, l.CustomerID, l.EndDate, c.Name

        FROM Loans l

        JOIN Customers c ON l.CustomerID = c.CustomerID

        WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

    FOR loan IN due\_soon\_cursor LOOP

        DBMS\_OUTPUT.PUT\_LINE(' Reminder: Dear ' || loan.Name ||

                             ', your loan (ID: ' || loan.LoanID ||

                             ') is due on ' || TO\_CHAR(loan.EndDate, 'DD-Mon-YYYY'));

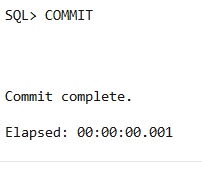
    END LOOP;

END;

/

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (77, 1, 8000, 6.5, SYSDATE, SYSDATE + 15);

COMMIT;  
  
  


Reminder: Dear John Doe, your loan (ID: 77) is due on 14-Jul-2025

PL/SQL:3

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    UPDATE Accounts

    SET Balance = Balance + (Balance \* 0.01)

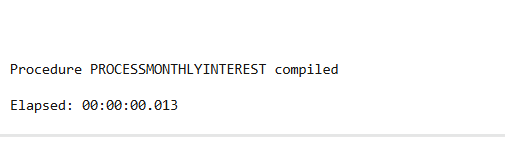
    WHERE AccountType = 'Savings';

    DBMS\_OUTPUT.PUT\_LINE(' Monthly interest applied to all savings accounts.');

    COMMIT;

END;

/

  
BEGIN

    ProcessMonthlyInterest;

END;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE Employees

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

WHERE Department = p\_department;

DBMS\_OUTPUT.PUT\_LINE('A black text on a white background

AI-generated content may be incorrect.Bonus of ' || p\_bonus\_percent || '% applied to ' || p\_department || ' department.');

COMMIT;

END;

/A screenshot of a computer

AI-generated content may be incorrect.BEGIN/

A screenshot of a computer

AI-generated content may be incorrect.

BEGIN

UpdateEmployeeBonus('IT', 10);

END;

/

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

-- Get balance of source account

SELECT Balance INTO v\_balance

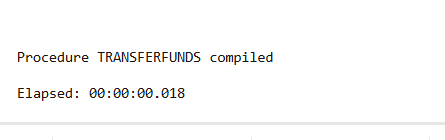
FROM Accounts

WHERE AccountID = p\_from\_account

FOR UPDATE;

-- Check sufficient funds

IF v\_balance < p\_amount THEN

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

END IF;

-- Deduct from source

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account;

-- Add to destination

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account;

DBMS\_OUTPUT.PUT\_LINE(' Transfer of ' || p\_amount || ' completed from Account ' || p\_from\_account ||

' to Account ' || p\_to\_account);

COMMIT;

END;

A screenshot of a computer

AI-generated content may be incorrect.

BEGIN

TransferFunds(1, 2, 300);

END;

/

