Exercise 1: Control Structures

Scenario 1: Apply 1% Discount for Customers Over 60

BEGIN

FOR cust IN (SELECT CustID, Name, Age, LoanInterestRate FROM CUSTOMERS) LOOP

IF cust.Age > 60 THEN

UPDATE CUSTOMERS

SET LoanInterestRate = LoanInterestRate - 1

WHERE CustID = cust.CustID;

DBMS\_OUTPUT.PUT\_LINE('1% discount applied to ' || cust.Name || '. New Rate Pending Update.');

END IF;

END LOOP;

END;

A close up of a text

AI-generated content may be incorrect.

Scenario 2: Mark Customers as VIP if Balance > 10,000

BEGIN

FOR cust IN (SELECT CustID, Name, Balance FROM CUSTOMERS) LOOP

IF cust.Balance > 10000 THEN

UPDATE CUSTOMERS

SET IsVIP = 'TRUE'

WHERE CustID = cust.CustID;

DBMS\_OUTPUT.PUT\_LINE('Customer ' || cust.Name || ' is now a VIP.');

END IF;

END LOOP;

END;

A close up of words

AI-generated content may be incorrect.

Scenario 3: Loan Reminders for Due Loans (within 30 Days)

BEGIN

FOR loan IN (

SELECT l.LoanID, l.CustID, l.DueDate, c.Name

FROM LOANS l

JOIN CUSTOMERS c ON l.CustID = c.CustID

WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

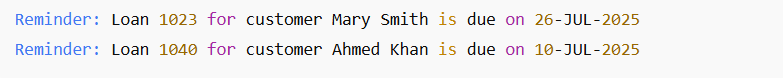
) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ' || loan.LoanID || ' for customer ' || loan.Name ||

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

END LOOP;

END;



Exercise 3: Stored Procedures

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT AccountID, Balance FROM SAVINGS\_ACCOUNTS) LOOP

UPDATE SAVINGS\_ACCOUNTS

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountID = acc.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Interest applied to Account ID ' || acc.AccountID ||

'. New balance will be updated.');

END LOOP;

END;

A number on a white background

AI-generated content may be incorrect.

Scenario 2: UpdateEmployeeBonus

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept\_id IN EMPLOYEES.DepartmentID%TYPE,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (SELECT EmployeeID, Salary FROM EMPLOYEES WHERE DepartmentID = p\_dept\_id) LOOP

UPDATE EMPLOYEES

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

WHERE EmployeeID = emp.EmployeeID;

DBMS\_OUTPUT.PUT\_LINE('Bonus added to Employee ID ' || emp.EmployeeID ||

'. New salary will be updated.');

END LOOP;

END;

A number of numbers on a white background

AI-generated content may be incorrect.

Scenario 3: TransferFunds

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN ACCOUNTS.AccountID%TYPE,

p\_to\_account IN ACCOUNTS.AccountID%TYPE,

p\_amount IN NUMBER

) IS

v\_balance ACCOUNTS.Balance%TYPE;

BEGIN

SELECT Balance INTO v\_balance FROM ACCOUNTS WHERE AccountID = p\_from\_account;

IF v\_balance < p\_amount THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: insufficient balance.');

ELSE

UPDATE ACCOUNTS

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account;

UPDATE ACCOUNTS

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amount || ' from Account ' ||

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

END IF;

END;

A black and yellow text

AI-generated content may be incorrect.

Usage Example for Each:

Sql

1. Process interest

BEGIN

ProcessMonthlyInterest;

END;

/

2. Add 10% bonus to Dept 3

BEGIN

UpdateEmployeeBonus(3, 10);

END;

/

3. Transfer 500 from Account 1010 to 1020

BEGIN

TransferFunds(1010, 1020, 500);

END;

/