**Exercise 3: Stored Procedures**

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

* + **Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

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

* + **Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

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

* + **Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Solution**

First we create three tables.

CREATE TABLE SAVINGS\_ACCOUNT (

ACCOUNT\_ID NUMBER PRIMARY KEY,

CUSTOMER\_NAME VARCHAR2(50),

BALANCE NUMBER(12,2)

);

INSERT INTO SAVINGS\_ACCOUNT VALUES (1001, 'Alice', 5000.00);

INSERT INTO SAVINGS\_ACCOUNT VALUES (1002, 'Bob', 1200.50);

INSERT INTO SAVINGS\_ACCOUNT VALUES (1003, 'Carol', 15000.75);

INSERT INTO SAVINGS\_ACCOUNT VALUES (1004, 'David', 800.00);

CREATE TABLE EMPLOYEE (

EMP\_ID NUMBER PRIMARY KEY,

EMP\_NAME VARCHAR2(50),

DEPARTMENT\_ID VARCHAR2(10),

SALARY NUMBER(12,2)

);

INSERT INTO EMPLOYEE VALUES (2001, 'Eve', 'SALES', 60000);

INSERT INTO EMPLOYEE VALUES (2002, 'Frank', 'SALES', 55000);

INSERT INTO EMPLOYEE VALUES (2003, 'Grace', 'IT', 70000);

INSERT INTO EMPLOYEE VALUES (2004, 'Heidi', 'IT', 72000);

CREATE TABLE ACCOUNT (

ACCT\_ID NUMBER PRIMARY KEY,

OWNER\_NAME VARCHAR2(50),

BALANCE NUMBER(12,2)

);

INSERT INTO ACCOUNT VALUES (3001, 'Ivan', 3000.00);

INSERT INTO ACCOUNT VALUES (3002, 'Judy', 1500.00);

COMMIT;

**SCENARIO 1: ProcessMonthlyInterest (1% interest on savings)**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR rec IN (SELECT ACCOUNT\_ID, BALANCE FROM SAVINGS\_ACCOUNT) LOOP

UPDATE SAVINGS\_ACCOUNT

SET BALANCE = rec.BALANCE \* 1.01

WHERE ACCOUNT\_ID = rec.ACCOUNT\_ID;

DBMS\_OUTPUT.PUT\_LINE(

'Acct '||rec.ACCOUNT\_ID

||' new balance: $'

||TO\_CHAR(rec.BALANCE \* 1.01, 'FM9999990.00')

);

END LOOP;

COMMIT;

END ProcessMonthlyInterest;

/

-- Run it

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- Processing monthly interest ---');

ProcessMonthlyInterest;

END;

/

-- Verify

SELECT \* FROM SAVINGS\_ACCOUNT ORDER BY ACCOUNT\_ID;

**OUTPUT**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SCENARIO 2: UpdateEmployeeBonus (add bonus % to salary per dept)**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept\_id IN VARCHAR2,

p\_bonus\_pct IN NUMBER -- e.g. pass 10 for 10%

) IS

BEGIN

FOR rec IN (

SELECT EMP\_ID, SALARY

FROM EMPLOYEE

WHERE DEPARTMENT\_ID = p\_dept\_id

) LOOP

UPDATE EMPLOYEE

SET SALARY = rec.SALARY \* (1 + p\_bonus\_pct/100)

WHERE EMP\_ID = rec.EMP\_ID;

DBMS\_OUTPUT.PUT\_LINE(

'Emp '||rec.EMP\_ID

||' updated salary: $'

||TO\_CHAR(rec.SALARY \* (1 + p\_bonus\_pct/100), 'FM9999990.00')

);

END LOOP;

COMMIT;

END UpdateEmployeeBonus;

/

-- Run it for SALES dept with 10% bonus

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- Applying 10% bonus to SALES ---');

UpdateEmployeeBonus('SALES', 10);

END;

/

-- Verify

SELECT \* FROM EMPLOYEE ORDER BY EMP\_ID;

**OUTPUT**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SCENARIO 3:TransferFunds (with balance check)**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_acct IN NUMBER,

p\_to\_acct IN NUMBER,

p\_amount IN NUMBER

) IS

v\_from\_bal NUMBER;

BEGIN

-- 1) Check source balance

SELECT BALANCE INTO v\_from\_bal

FROM ACCOUNT

WHERE ACCT\_ID = p\_from\_acct;

IF v\_from\_bal < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001,

'Insufficient funds in acct '||p\_from\_acct

||'. Available: '||v\_from\_bal||', requested: '||p\_amount

);

END IF;

-- 2) Debit source

UPDATE ACCOUNT

SET BALANCE = BALANCE - p\_amount

WHERE ACCT\_ID = p\_from\_acct;

-- 3) Credit destination

UPDATE ACCOUNT

SET BALANCE = BALANCE + p\_amount

WHERE ACCT\_ID = p\_to\_acct;

DBMS\_OUTPUT.PUT\_LINE(

'Transferred $'||TO\_CHAR(p\_amount,'FM999990.00')

||' from '||p\_from\_acct

||' to '||p\_to\_acct

);

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('ERROR: Account not found.');

WHEN OTHERS THEN

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

END TransferFunds;

/

-- Run a successful transfer

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- Transfer $500 from 3001→3002 ---');

TransferFunds(3001, 3002, 500);

END;

/

-- Run a failing transfer (insufficient funds)

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- Transfer $5000 from 3002→3001 ---');

TransferFunds(3002, 3001, 5000);

END;

/

-- Verify final balances

SELECT \* FROM ACCOUNT ORDER BY ACCT\_ID;

**OUTPUT**

A screenshot of a computer

AI-generated content may be incorrect.