**Exercise 3: Stored Procedures**

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

**o 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.

**o 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.

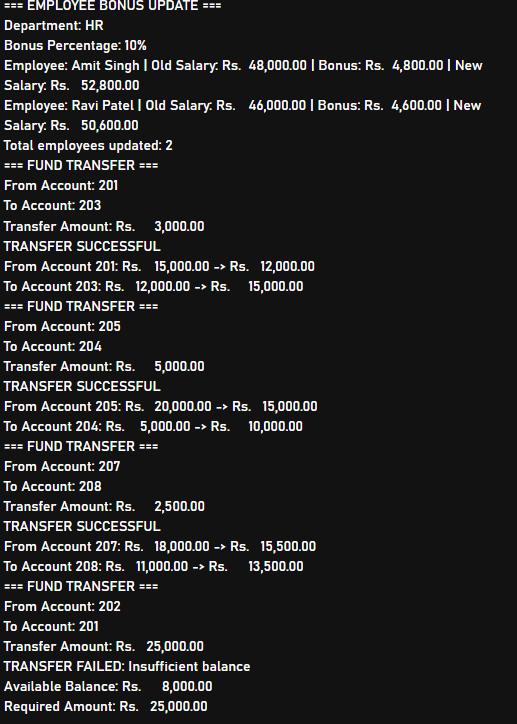
**Scenario 3:** Customers should be able to transfer funds between their accounts.

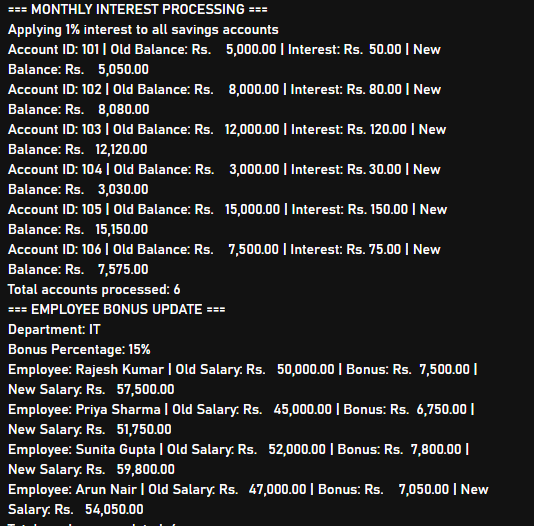
**o Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Source Code :**

| CREATE TABLE savings\_accounts (  account\_id NUMBER PRIMARY KEY,  customer\_id NUMBER,  balance NUMBER(15,2),  account\_type VARCHAR2(20) DEFAULT 'SAVINGS' );  CREATE TABLE employees (  employee\_id NUMBER PRIMARY KEY,  name VARCHAR2(100),  department VARCHAR2(50),  salary NUMBER(10,2) );  CREATE TABLE accounts (  account\_id NUMBER PRIMARY KEY,  customer\_id NUMBER,  balance NUMBER(15,2),  account\_type VARCHAR2(20) );  INSERT INTO savings\_accounts VALUES (101, 1, 5000.00, 'SAVINGS'); INSERT INTO savings\_accounts VALUES (102, 2, 8000.00, 'SAVINGS'); INSERT INTO savings\_accounts VALUES (103, 3, 12000.00, 'SAVINGS'); INSERT INTO savings\_accounts VALUES (104, 4, 3000.00, 'SAVINGS'); INSERT INTO savings\_accounts VALUES (105, 5, 15000.00, 'SAVINGS'); INSERT INTO savings\_accounts VALUES (106, 6, 7500.00, 'SAVINGS');  INSERT INTO employees VALUES (1, 'Rajesh Kumar', 'IT', 50000.00); INSERT INTO employees VALUES (2, 'Priya Sharma', 'IT', 45000.00); INSERT INTO employees VALUES (3, 'Amit Singh', 'HR', 48000.00); INSERT INTO employees VALUES (4, 'Sunita Gupta', 'IT', 52000.00); INSERT INTO employees VALUES (5, 'Ravi Patel', 'HR', 46000.00); INSERT INTO employees VALUES (6, 'Kavya Reddy', 'Finance', 55000.00); INSERT INTO employees VALUES (7, 'Arun Nair', 'IT', 47000.00); INSERT INTO employees VALUES (8, 'Meera Joshi', 'Finance', 53000.00);  INSERT INTO accounts VALUES (201, 1, 15000.00, 'CHECKING'); INSERT INTO accounts VALUES (202, 1, 8000.00, 'SAVINGS'); INSERT INTO accounts VALUES (203, 2, 12000.00, 'CHECKING'); INSERT INTO accounts VALUES (204, 2, 5000.00, 'SAVINGS'); INSERT INTO accounts VALUES (205, 3, 20000.00, 'CHECKING'); INSERT INTO accounts VALUES (206, 4, 9000.00, 'SAVINGS'); INSERT INTO accounts VALUES (207, 5, 18000.00, 'CHECKING'); INSERT INTO accounts VALUES (208, 6, 11000.00, 'SAVINGS');  COMMIT;  -- SCENARIO 1: Process Monthly Interest for Savings Accounts  CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS  v\_account\_id savings\_accounts.account\_id%TYPE;  v\_old\_balance savings\_accounts.balance%TYPE;  v\_new\_balance savings\_accounts.balance%TYPE;  v\_interest\_amount NUMBER;  v\_count NUMBER := 0;    CURSOR c\_savings IS  SELECT account\_id, balance  FROM savings\_accounts;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== MONTHLY INTEREST PROCESSING ===');  DBMS\_OUTPUT.PUT\_LINE('Applying 1% interest to all savings accounts');  DBMS\_OUTPUT.PUT\_LINE('');    FOR rec IN c\_savings LOOP  v\_account\_id := rec.account\_id;  v\_old\_balance := rec.balance;  v\_interest\_amount := v\_old\_balance \* 0.01;  v\_new\_balance := v\_old\_balance + v\_interest\_amount;    UPDATE savings\_accounts   SET balance = v\_new\_balance   WHERE account\_id = v\_account\_id;    DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_account\_id ||   ' | Old Balance: Rs.' || TO\_CHAR(v\_old\_balance, '9,99,999.99') ||  ' | Interest: Rs.' || TO\_CHAR(v\_interest\_amount, '999.99') ||  ' | New Balance: Rs.' || TO\_CHAR(v\_new\_balance, '9,99,999.99'));  v\_count := v\_count + 1;  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('');  DBMS\_OUTPUT.PUT\_LINE('Total accounts processed: ' || v\_count);  DBMS\_OUTPUT.PUT\_LINE('');  COMMIT;   EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);  ROLLBACK; END ProcessMonthlyInterest; /  -- SCENARIO 2: Update Employee Bonus  CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(  p\_department IN VARCHAR2,  p\_bonus\_percentage IN NUMBER ) AS  v\_employee\_id employees.employee\_id%TYPE;  v\_name employees.name%TYPE;  v\_old\_salary employees.salary%TYPE;  v\_bonus\_amount NUMBER;  v\_new\_salary NUMBER;  v\_count NUMBER := 0;    CURSOR c\_employees IS  SELECT employee\_id, name, salary  FROM employees  WHERE department = p\_department;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== EMPLOYEE BONUS UPDATE ===');  DBMS\_OUTPUT.PUT\_LINE('Department: ' || p\_department);  DBMS\_OUTPUT.PUT\_LINE('Bonus Percentage: ' || p\_bonus\_percentage || '%');  DBMS\_OUTPUT.PUT\_LINE('');    FOR rec IN c\_employees LOOP  v\_employee\_id := rec.employee\_id;  v\_name := rec.name;  v\_old\_salary := rec.salary;  v\_bonus\_amount := v\_old\_salary \* (p\_bonus\_percentage / 100);  v\_new\_salary := v\_old\_salary + v\_bonus\_amount;    UPDATE employees   SET salary = v\_new\_salary   WHERE employee\_id = v\_employee\_id;    DBMS\_OUTPUT.PUT\_LINE('Employee: ' || v\_name ||   ' | Old Salary: Rs.' || TO\_CHAR(v\_old\_salary, '9,99,999.99') ||  ' | Bonus: Rs.' || TO\_CHAR(v\_bonus\_amount, '99,999.99') ||  ' | New Salary: Rs.' || TO\_CHAR(v\_new\_salary, '9,99,999.99'));  v\_count := v\_count + 1;  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('');  DBMS\_OUTPUT.PUT\_LINE('Total employees updated: ' || v\_count);  DBMS\_OUTPUT.PUT\_LINE('');  COMMIT;   EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);  ROLLBACK; END UpdateEmployeeBonus; /  -- SCENARIO 3: Transfer Funds Between Accounts  CREATE OR REPLACE PROCEDURE TransferFunds(  p\_from\_account IN NUMBER,  p\_to\_account IN NUMBER,  p\_amount IN NUMBER ) AS  v\_from\_balance accounts.balance%TYPE;  v\_to\_balance accounts.balance%TYPE;  v\_new\_from\_balance NUMBER;  v\_new\_to\_balance NUMBER;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== FUND TRANSFER ===');  DBMS\_OUTPUT.PUT\_LINE('From Account: ' || p\_from\_account);  DBMS\_OUTPUT.PUT\_LINE('To Account: ' || p\_to\_account);  DBMS\_OUTPUT.PUT\_LINE('Transfer Amount: Rs.' || TO\_CHAR(p\_amount, '9,99,999.99'));  DBMS\_OUTPUT.PUT\_LINE('');      SELECT balance INTO v\_from\_balance   FROM accounts   WHERE account\_id = p\_from\_account;     IF v\_from\_balance < p\_amount THEN  DBMS\_OUTPUT.PUT\_LINE('TRANSFER FAILED: Insufficient balance');  DBMS\_OUTPUT.PUT\_LINE('Available Balance: Rs.' || TO\_CHAR(v\_from\_balance, '9,99,999.99'));  DBMS\_OUTPUT.PUT\_LINE('Required Amount: Rs.' || TO\_CHAR(p\_amount, '9,99,999.99'));  DBMS\_OUTPUT.PUT\_LINE('');  RETURN;  END IF;    SELECT balance INTO v\_to\_balance   FROM accounts   WHERE account\_id = p\_to\_account;    v\_new\_from\_balance := v\_from\_balance - p\_amount;  v\_new\_to\_balance := v\_to\_balance + p\_amount;    UPDATE accounts   SET balance = v\_new\_from\_balance   WHERE account\_id = p\_from\_account;    UPDATE accounts   SET balance = v\_new\_to\_balance   WHERE account\_id = p\_to\_account;    DBMS\_OUTPUT.PUT\_LINE('TRANSFER SUCCESSFUL');  DBMS\_OUTPUT.PUT\_LINE('From Account ' || p\_from\_account || ': Rs.' ||   TO\_CHAR(v\_from\_balance, '9,99,999.99') || ' -> Rs.' ||   TO\_CHAR(v\_new\_from\_balance, '9,99,999.99'));  DBMS\_OUTPUT.PUT\_LINE('To Account ' || p\_to\_account || ': Rs.' ||   TO\_CHAR(v\_to\_balance, '9,99,999.99') || ' -> Rs.' ||   TO\_CHAR(v\_new\_to\_balance, '9,99,999.99'));  DBMS\_OUTPUT.PUT\_LINE('');  COMMIT;   EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Error: Account not found');  DBMS\_OUTPUT.PUT\_LINE('');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);  DBMS\_OUTPUT.PUT\_LINE('');  ROLLBACK; END TransferFunds; /  -- EXECUTE THE STORED PROCEDURES  -- Test Scenario 1: Process Monthly Interest EXECUTE ProcessMonthlyInterest;  -- Test Scenario 2: Update Employee Bonus (15% bonus to IT department) EXECUTE UpdateEmployeeBonus('IT', 15);  -- Test Scenario 2: Update Employee Bonus (10% bonus to HR department) EXECUTE UpdateEmployeeBonus('HR', 10);  -- Test Scenario 3: Transfer Funds EXECUTE TransferFunds(201, 203, 3000); EXECUTE TransferFunds(205, 204, 5000); EXECUTE TransferFunds(207, 208, 2500); EXECUTE TransferFunds(202, 201, 25000); |
| --- |

**Output :**

****

****