**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.

**Procedure:**

SQL> CREATE OR REPLACE PROCEDURE AddNewCustomer (

2 p\_CustomerID IN NUMBER,

3 p\_Name IN VARCHAR2,

4 p\_DOB IN DATE,

5 p\_Balance IN NUMBER

6 ) AS

7 v\_Count NUMBER;

8 BEGIN

9 -- Check if customer ID already exists

10 SELECT COUNT(\*)

11 INTO v\_Count

12 FROM Customers

13 WHERE CustomerID = p\_CustomerID;

14

15 IF v\_Count > 0 THEN

16 RAISE\_APPLICATION\_ERROR(-20003, 'Customer ID already exists.');

17 END IF;

18

19 -- Insert the new customer

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

21 VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

22

23 -- Commit the transaction

24 COMMIT;

25 EXCEPTION

26 WHEN OTHERS THEN

27 -- Rollback the transaction in case of error

28 ROLLBACK;

29 -- Log the error message

30 DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);

31 END AddNewCustomer;

32 /

**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.

**Procedure:**

SQL> CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_department IN VARCHAR2, p\_bonus\_percentage IN NUMBER) AS

2 BEGIN

3 UPDATE Employees

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

5 WHERE Department = p\_department;

6

7 COMMIT;

8 END;

9 /

**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.

**Procedure:**

SQL> CREATE OR REPLACE PROCEDURE TransferFunds(

2 p\_source\_account\_id IN NUMBER,

3 p\_target\_account\_id IN NUMBER,

4 p\_amount IN NUMBER

5 ) AS

6 v\_source\_balance NUMBER;

7 BEGIN

8 -- Check if source account has sufficient balance

9 SELECT Balance INTO v\_source\_balance

10 FROM Accounts

11 WHERE AccountID = p\_source\_account\_id;

12

13 IF v\_source\_balance < p\_amount THEN

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

15 END IF;

16

17 -- Perform the transfer

18 UPDATE Accounts

19 SET Balance = Balance - p\_amount

20 WHERE AccountID = p\_source\_account\_id;

21

22 UPDATE Accounts

23 SET Balance = Balance + p\_amount

24 WHERE AccountID = p\_target\_account\_id;

25

26 -- Record the transaction for both accounts

27 INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

28 VALUES (Transactions\_seq.NEXTVAL, p\_source\_account\_id, SYSDATE, p\_amount, 'Transfer Out');

29

30 INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

31 VALUES (Transactions\_seq.NEXTVAL, p\_target\_account\_id, SYSDATE, p\_amount, 'Transfer In');

32

33 COMMIT;

34 END;

35 /