**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.**
* **Code for this problem:**

set serveroutput on;

create or replace PROCEDURE ProcessMonthlyInterest IS

begin

update ACCOUNTS

set balance= balance + (balance\*0.01)

WHERE ACCOUNTTYPE = 'Savings';

end;

**To execute this program:**

exec processmonthlyinterest;

select \* from ACCOUNTS;

**Output of the code:**

**A black and white text

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**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.**
* **Code for this program:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE EMPLOYEES

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

WHERE DEPARTMENT = p\_dept\_id;

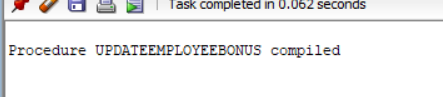
EXCEPTION

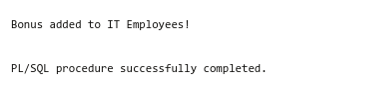
WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20100, 'Error updating employee bonuses: ' || SQLERRM);

END;

**Output of the code:**

****

****

**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.**
* **Code for this problem:**

create or replace PROCEDURE TransferFunds(

from\_account IN number,

to\_account IN number,

amount IN number

)IS

v\_balance number;

begin

select BALANCE into v\_balance

from ACCOUNTS

where ACCOUNTID = from\_account

for update;

if v\_balance<amount then

RAISE\_APPLICATION\_ERROR(-20001,'Insufficient Balance');

end if;

update ACCOUNTS

set BALANCE = BALANCE - amount , lastModified = sysdate

where ACCOUNTID = from\_account;

update ACCOUNTS

set BALANCE = BALANCE + amount , lastModified = sysdate

where ACCOUNTID = to\_account;

end;

**To execute this problem:**

begin

TransferFunds(1,2,500);

end;

**Output of the code:**

**A close-up of a word

AI-generated content may be incorrect.**

**A close-up of a computer screen

AI-generated content may be incorrect.**