**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.

Reference table creation and insertion of values

CREATE TABLE SavingsAccounts (

  account\_id    NUMBER PRIMARY KEY,

  customer\_id   NUMBER,

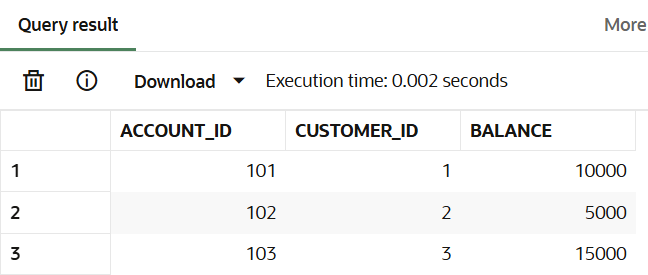
  balance       NUMBER

);

INSERT INTO SavingsAccounts VALUES (101, 1, 10000);

INSERT INTO SavingsAccounts VALUES (102, 2, 5000);

INSERT INTO SavingsAccounts VALUES (103, 3, 15000);



**Code:**

create or replace procedure ProcessMonthlyInterest is

begin

    for i in (select account\_id, balance from SavingsAccounts) loop

        update SavingsAccounts

        set balance = i.balance \* 1.01

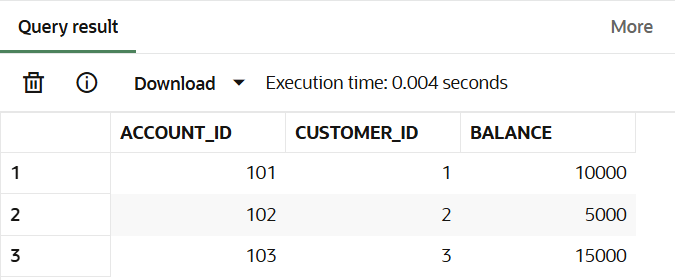
        where account\_id = i.account\_id;

    end loop;

    commit;

end;

**Output:**

****

**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.

Reference table creation and insertion of values

CREATE TABLE Employees (

  emp\_id        NUMBER PRIMARY KEY,

  name          VARCHAR2(50),

  department    VARCHAR2(30),

  salary        NUMBER

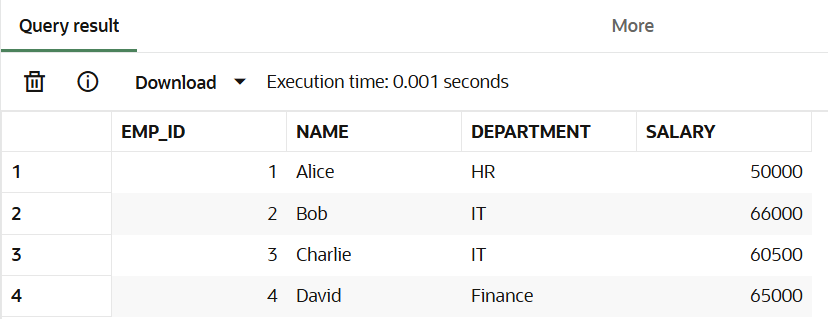
);

INSERT INTO Employees VALUES (1, 'Alice', 'HR', 50000);

INSERT INTO Employees VALUES (2, 'Bob', 'IT', 60000);

INSERT INTO Employees VALUES (3, 'Charlie', 'IT', 55000);

INSERT INTO Employees VALUES (4, 'David', 'Finance', 65000);



**Code:**

create or replace procedure UpdateEmployeeBonus(

    dept\_name in varchar2, bonus\_percent in number

) is

begin

    update employees

    set salary = salary + (salary \* bonus\_percent / 100)

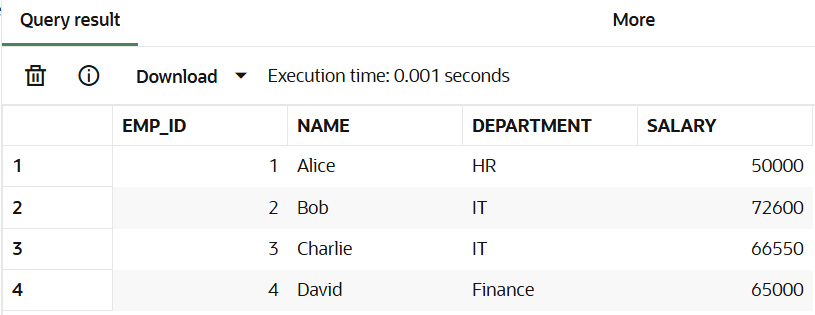
    where department = dept\_name;

    commit;

end;

exec UpdateEmployeeBonus('IT', 10);

**Output:**

****

**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.

Reference table creation and insertion of values

CREATE TABLE BankAccounts (

  account\_id    NUMBER PRIMARY KEY,

  customer\_id   NUMBER,

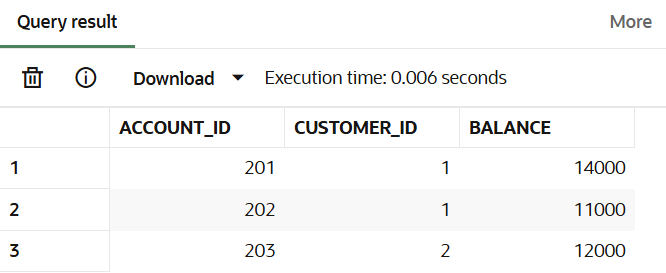
  balance       NUMBER

);

INSERT INTO BankAccounts VALUES (201, 1, 20000);

INSERT INTO BankAccounts VALUES (202, 1, 5000);

INSERT INTO BankAccounts VALUES (203, 2, 12000);



**Code:**

create or replace procedure TransferFunds(

    from\_account in number,

    to\_account in number,

    amount in number

) is

from\_balance number;

insufficient\_balance exception;

begin

    select balance into from\_balance

    from bankaccounts

    where account\_id = from\_account;

    if from\_balance < amount then

        raise insufficient\_balance;

    end if;

    update bankaccounts

    set balance = balance - amount

    where account\_id = from\_account;

    update bankaccounts

    set balance = balance + amount

    where account\_id = to\_account;

    dbms\_output.put\_line('Ammount transferd successfully');

    commit;

exception

    when insufficient\_balance then dbms\_output.put\_line('Insufficient balance to transfer');

end;

exec TransferFunds(201, 202, 3000);

**Output:**

