**PL/SQL programming**

**Exercise 3: Stored Procedures**

**Accounts Table**

**Create:**

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

account\_type VARCHAR2(20),

balance NUMBER  
);

**Insert:**

INSERT INTO accounts VALUES (101, 'savings', 1000);

INSERT INTO accounts VALUES (102, 'current', 2000);

INSERT INTO accounts VALUES (103, 'savings', 1500);

**Employees Table**

**Create:**

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

department\_id NUMBER,

salary NUMBER  
);

**Insert:**

INSERT INTO employees VALUES (1, 'Maha', 10, 50000);

INSERT INTO employees VALUES (2, 'Subha', 10, 60000);

INSERT INTO employees VALUES (3, 'Thenmozhi', 20, 55000);

1. **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.**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_type = 'savings';

COMMIT;

END;

BEGIN

ProcessMonthlyInterest;

END;

SELECT \* FROM accounts;

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

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\_id = p\_dept\_id;

COMMIT;

END;

BEGIN

UpdateEmployeeBonus(10, 10);

END;

SELECT \* FROM employees;

1. **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.**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_acc\_id IN NUMBER,

p\_to\_acc\_id IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

insufficient\_balance EXCEPTION;

BEGIN

-- Get balance of source account

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_acc\_id

FOR UPDATE;

IF v\_balance < p\_amount THEN

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

END IF;

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_acc\_id;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_acc\_id;

COMMIT;

EXCEPTION  
WHEN insufficient\_balance THEN  
DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance');  
ROLLBACK;  
WHEN OTHERS THEN  
DBMS\_OUTPUT.PUT\_LINE('Transfer failed: ' || SQLERRM);  
ROLLBACK;  
END;

END;

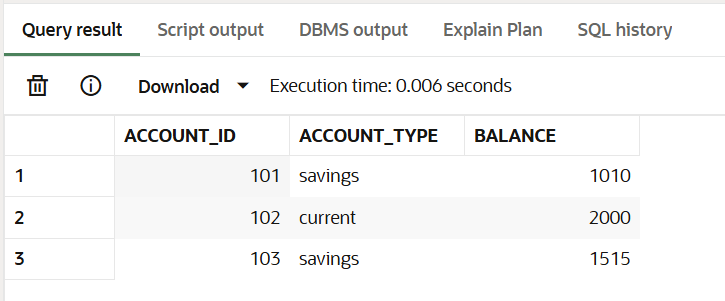
BEGIN

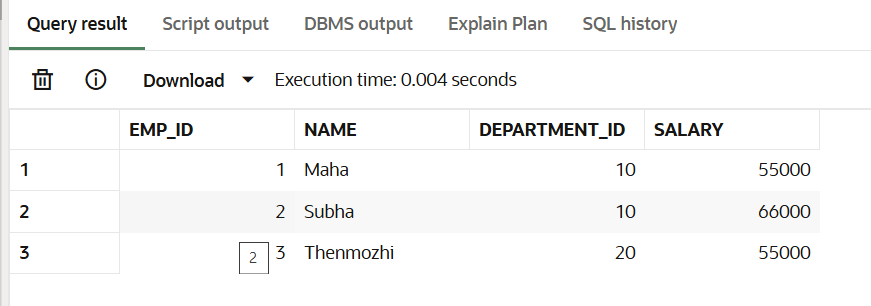
TransferFunds(201, 202, 1000);

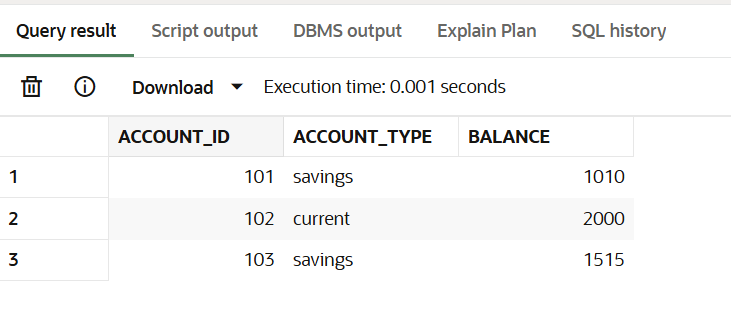
END;

SELECT \* FROM accounts;

**OUTPUT:**

****

****

****