WEEK – 2

Nipuna Amanapu

Superset id: 6432842

**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.

Solution:

1)bank\_setup.sql :

CREATE TABLE savings\_accounts (

    account\_id NUMBER PRIMARY KEY,

    customer\_name VARCHAR2(100),

    balance NUMBER

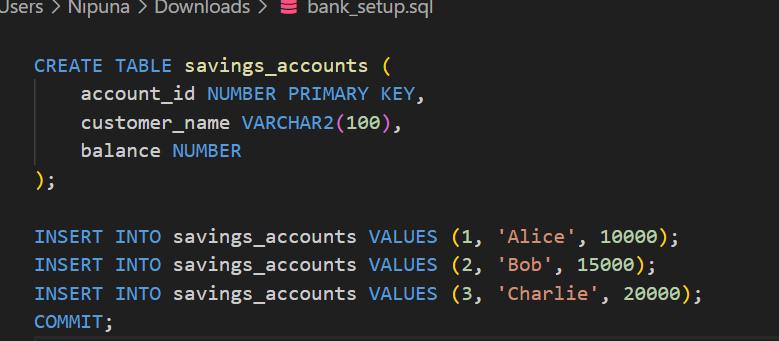
);

INSERT INTO savings\_accounts VALUES (1, 'Alice', 10000);

INSERT INTO savings\_accounts VALUES (2, 'Bob', 15000);

INSERT INTO savings\_accounts VALUES (3, 'Charlie', 20000);

COMMIT;



2)process\_interest.sql :

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

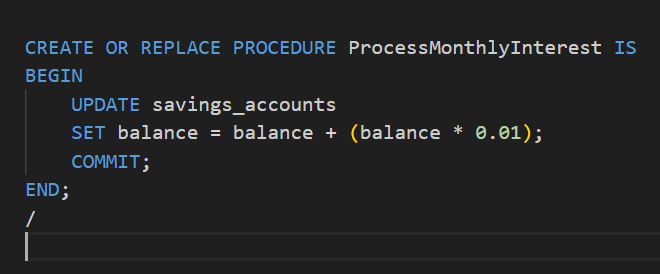
    UPDATE savings\_accounts

    SET balance = balance + (balance \* 0.01);

    COMMIT;

END;

/



3)run\_interest.sql :

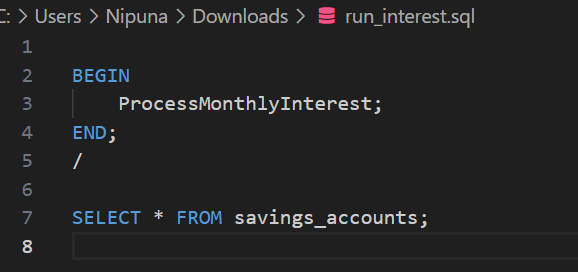
BEGIN

    ProcessMonthlyInterest;

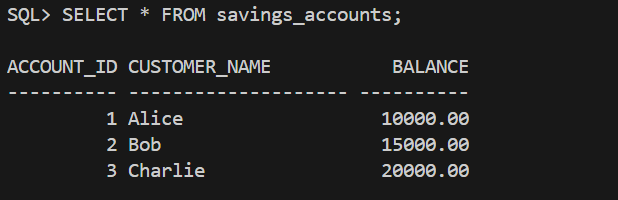
END;

/

SELECT \* FROM savings\_accounts;



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.

Solution :

1)employee\_setup.sql :

CREATE TABLE employees (

    employee\_id NUMBER PRIMARY KEY,

    name VARCHAR2(100),

    department VARCHAR2(50),

    salary NUMBER

);

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

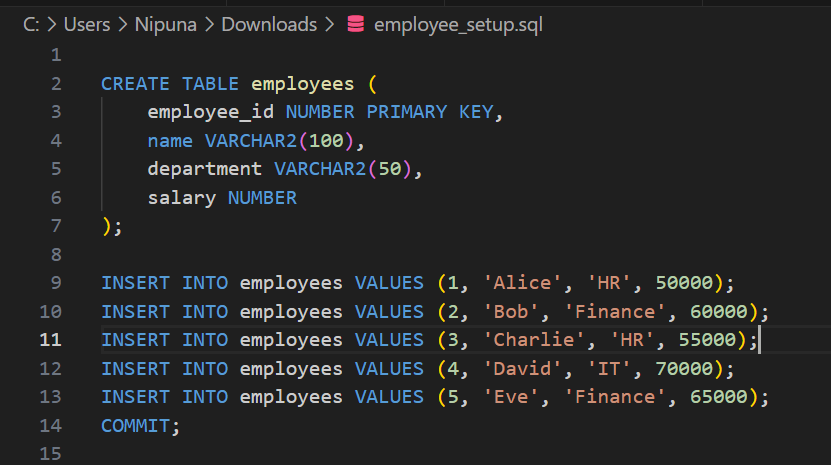
INSERT INTO employees VALUES (2, 'Bob', 'Finance', 60000);

INSERT INTO employees VALUES (3, 'Charlie', 'HR', 55000);

INSERT INTO employees VALUES (4, 'David', 'IT', 70000);

INSERT INTO employees VALUES (5, 'Eve', 'Finance', 65000);

COMMIT;



2)update\_bonus\_proc :

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept\_name IN VARCHAR2,

    bonus\_percent IN NUMBER

) IS

BEGIN

    UPDATE employees

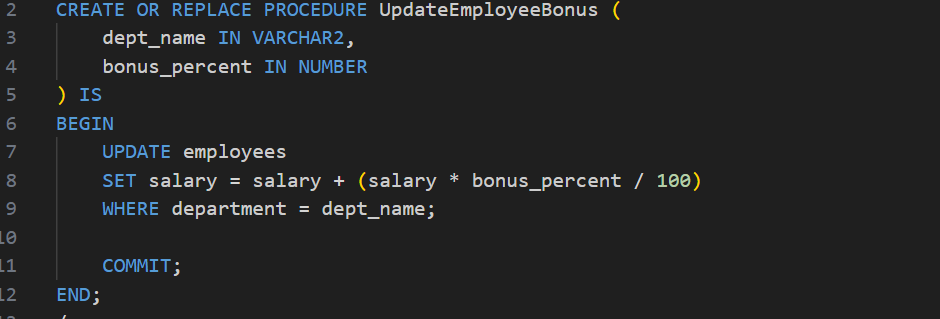
    SET salary = salary + (salary \* bonus\_percent / 100)

    WHERE department = dept\_name;

    COMMIT;

END;

/



3)run\_bonus\_update :

SET LINESIZE 100

SET PAGESIZE 50

COLUMN employee\_id FORMAT 999

COLUMN name FORMAT A15

COLUMN department FORMAT A12

COLUMN salary FORMAT 999999.99

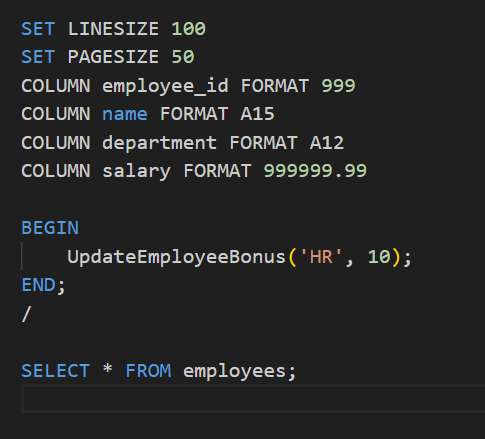
BEGIN

    UpdateEmployeeBonus('HR', 10);

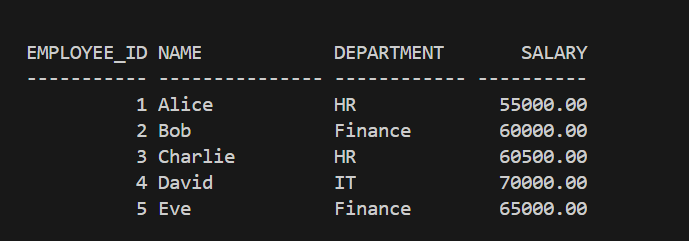
END;

/

SELECT \* FROM employees;



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.

Solution :

1)account\_setup.sql :

CREATE TABLE bank\_accounts (

    account\_id NUMBER PRIMARY KEY,

    customer\_name VARCHAR2(100),

    balance NUMBER

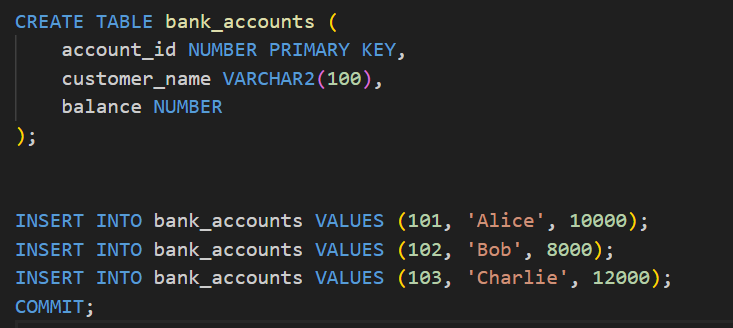
);

INSERT INTO bank\_accounts VALUES (101, 'Alice', 10000);

INSERT INTO bank\_accounts VALUES (102, 'Bob', 8000);

INSERT INTO bank\_accounts VALUES (103, 'Charlie', 12000);

COMMIT;



2)transfer\_funds\_proc.sql :

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_account\_id IN NUMBER,

    to\_account\_id IN NUMBER,

    amount IN NUMBER

) IS

    from\_balance NUMBER;

BEGIN

    SELECT balance INTO from\_balance

    FROM bank\_accounts

    WHERE account\_id = from\_account\_id;

    IF from\_balance < amount THEN

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

    END IF;

    UPDATE bank\_accounts

    SET balance = balance - amount

    WHERE account\_id = from\_account\_id;

    UPDATE bank\_accounts

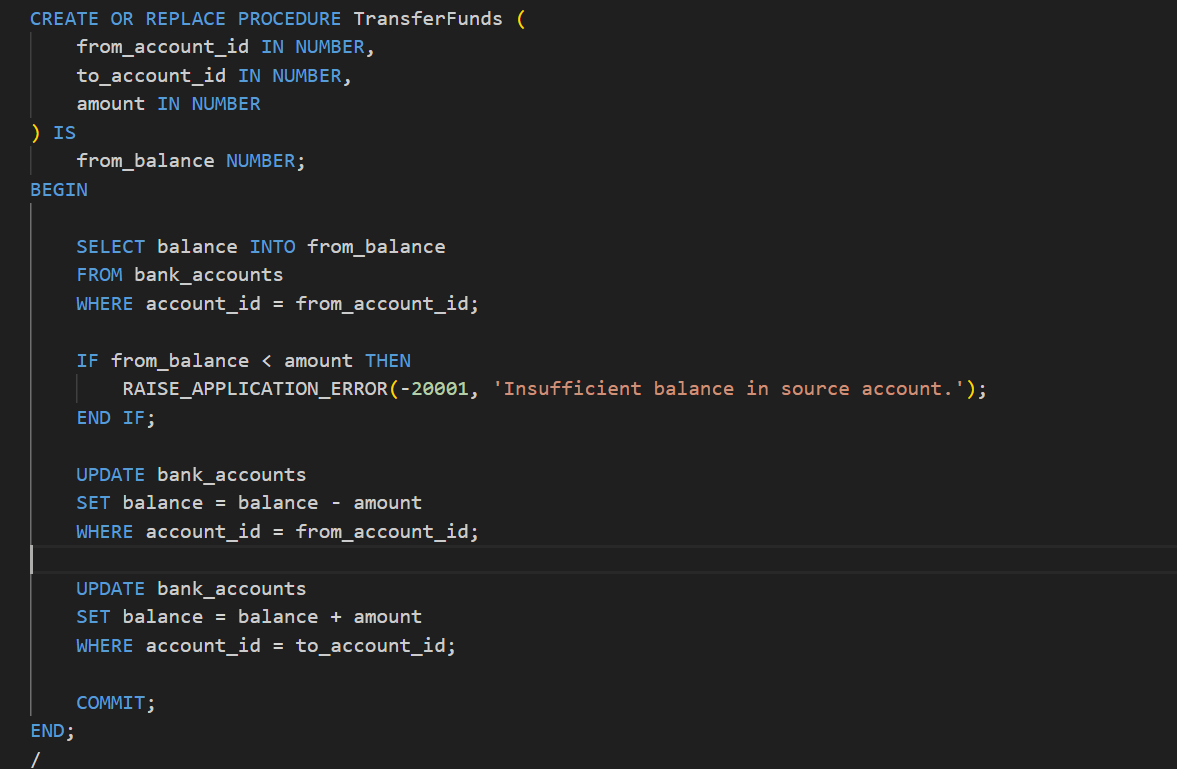
    SET balance = balance + amount

    WHERE account\_id = to\_account\_id;

    COMMIT;

END;

/



3)run\_transfer.sql :

SET LINESIZE 100

SET PAGESIZE 50

COLUMN account\_id FORMAT 999

COLUMN customer\_name FORMAT A15

COLUMN balance FORMAT 999999.99

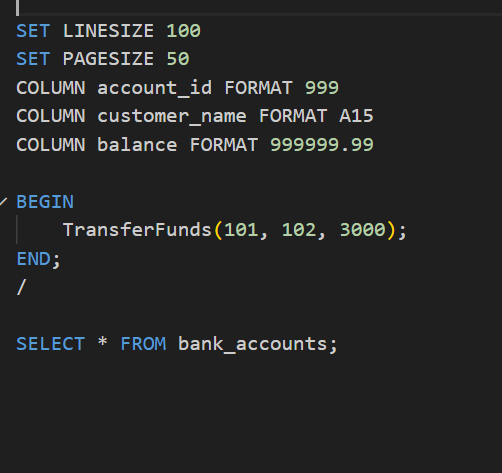
BEGIN

    TransferFunds(101, 102, 3000);

END;

/

SELECT \* FROM bank\_accounts;



Output :

