Stored Procedure

**What is a Stored Procedure?**

A **Stored Procedure** is a **precompiled block of code** stored in the database that performs a specific task.

**Key Points:**

* Written using **PL/SQL (Oracle)** or **T-SQL (SQL Server)**.
* Can accept **parameters (IN, OUT, IN OUT)**.
* Improves performance (compiled once, reused).
* Helps in **code reuse**, **modularization**, and **security**.
* Stored and executed **on the database server**.

**📊 2. Table Structures**

**🔹 Table: Accounts**

* AccountID: NUMBER (Primary Key)
* AccountHolderName: VARCHAR2(100)
* AccountType: VARCHAR2(20)
* Balance: NUMBER(15,2)

**🔹 Table: Employees**

* EmployeeID: NUMBER (Primary Key)
* EmployeeName: VARCHAR2(100)
* DepartmentID: NUMBER
* Salary: NUMBER(10,2)

**3. Table Creation Script**

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

AccountHolderName VARCHAR2(100),

AccountType VARCHAR2(20),

Balance NUMBER(15, 2)

);

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

EmployeeName VARCHAR2(100),

DepartmentID NUMBER,

Salary NUMBER(10, 2)

);

**📥 4. Insert Sample Data**

-- Accounts

INSERT INTO Accounts VALUES (201, 'Meena Gupta', 'Savings', 12000);

INSERT INTO Accounts VALUES (202, 'Rohit Das', 'Savings', 18000);

INSERT INTO Accounts VALUES (203, 'Kiran Rao', 'Savings', 23000);

INSERT INTO Accounts VALUES (204, 'Farah Khan', 'Savings', 28000);

INSERT INTO Accounts VALUES (205, 'Amar Singh', 'Current', 45000);

INSERT INTO Accounts VALUES (206, 'Divya Patel', 'Current', 70000);

INSERT INTO Accounts VALUES (207, 'Amitabh Roy', 'Current', 85000);

-- Employees

INSERT INTO Employees VALUES (301, 'Vinay Kumar', 101, 48000);

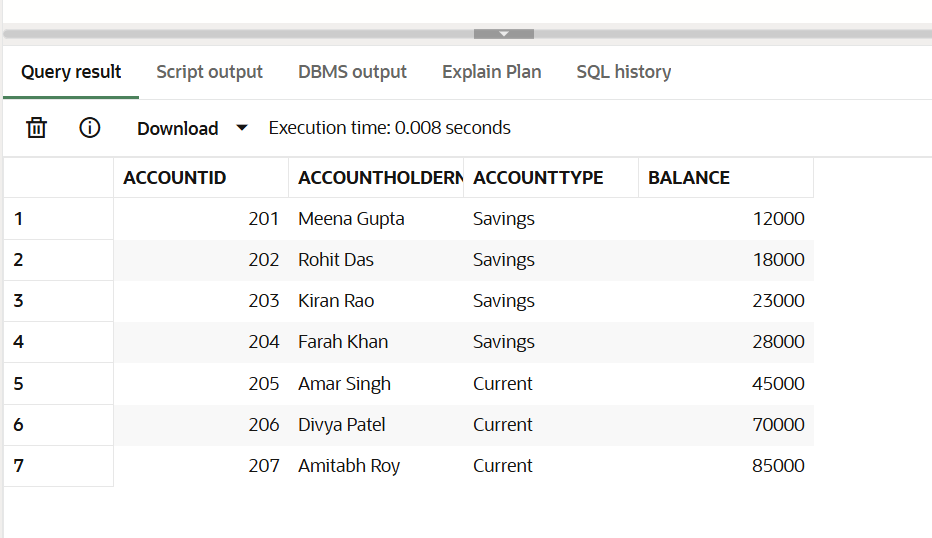
INSERT INTO Employees VALUES (302, 'Rani Paul', 101, 58000);

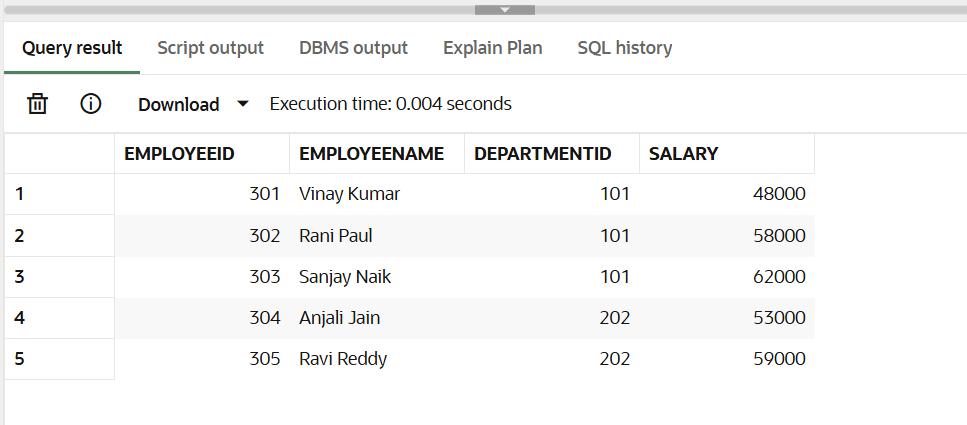
INSERT INTO Employees VALUES (303, 'Sanjay Naik', 101, 62000);

INSERT INTO Employees VALUES (304, 'Anjali Jain', 202, 53000);

INSERT INTO Employees VALUES (305, 'Ravi Reddy', 202, 59000);

COMMIT;





**Scenario 1: Process Monthly Interest (for Savings Accounts)**

**Description:**

Apply 1.2% interest to all savings accounts.

**Procedure Code:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.012)

WHERE AccountType = 'Savings';

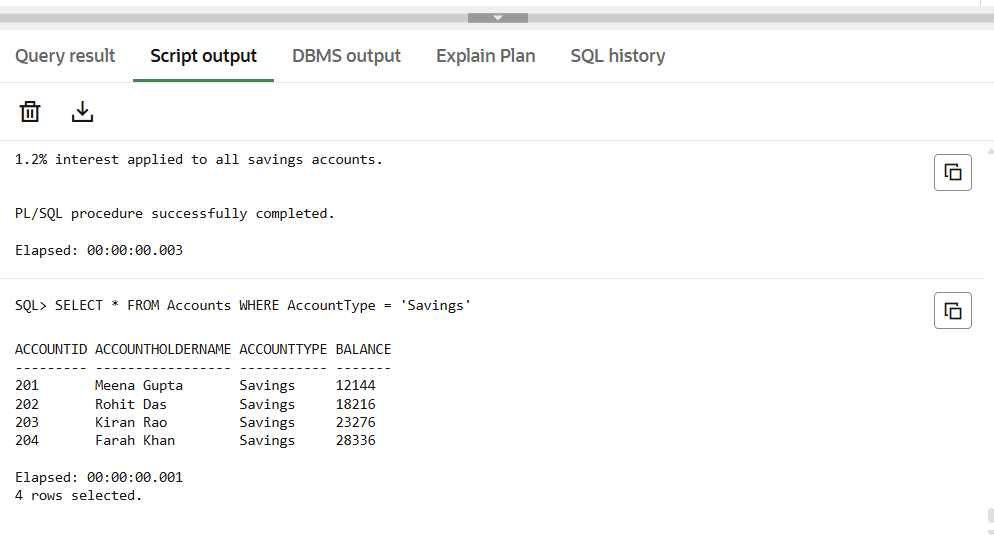
DBMS\_OUTPUT.PUT\_LINE('1.2% interest applied to all savings accounts.');

END;

/

**📖 Explanation:**

This procedure identifies all rows where AccountType = 'Savings' and updates their balances with 1.2% interest.



**Scenario 2: Update Employee Bonus (Department-wise)**

**Description:**

Increase salary for all employees in a specific department by a given bonus percentage.

**Procedure Code:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_id IN NUMBER,

bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE Employees

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

WHERE DepartmentID = dept\_id;

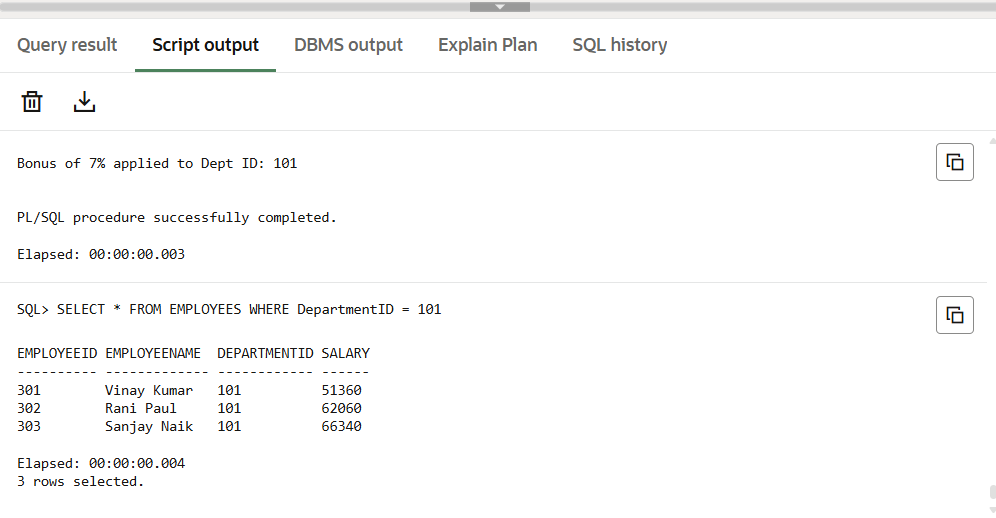
DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || bonus\_percent || '% applied to Dept ID: ' || dept\_id);

END;

/

**Explanation:**

This stored procedure increases salaries of employees in a selected department by the provided percentage. It uses parameters and basic arithmetic logic.



**Scenario 3: Transfer Funds Between Accounts**

**Description:**

Transfer funds from one account to another with validation and error handling.

**Procedure Code:**

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_account IN NUMBER,

to\_account IN NUMBER,

amount IN NUMBER

) IS

source\_balance NUMBER;

BEGIN

SELECT Balance INTO source\_balance

FROM Accounts

WHERE AccountID = from\_account

FOR UPDATE;

IF source\_balance < amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Transfer failed: insufficient funds.');

END IF;

UPDATE Accounts

SET Balance = Balance - amount

WHERE AccountID = from\_account;

UPDATE Accounts

SET Balance = Balance + amount

WHERE AccountID = to\_account;

DBMS\_OUTPUT.PUT\_LINE('₹' || amount || ' transferred from Account ' || from\_account || ' to Account ' || to\_account);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Invalid account ID(s).');

WHEN OTHERS THEN

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

END;

/

**Explanation:**

* Retrieves balance of sender account using SELECT ... FOR UPDATE
* Validates if sender has enough balance
* Deducts from sender and adds to receiver
* Handles exceptions like NO\_DATA\_FOUND and generic errors using WHEN OTHERS

