**Exercise 2: Error Handling**

**Scenario 1:** Handle exceptions during fund transfers between accounts.

* + **Question:** Write a stored procedure **SafeTransferFunds** that transfers funds between two accounts. Ensure that if any error occurs (e.g., insufficient funds), an appropriate error message is logged and the transaction is rolled back.

**Code :**

CREATE OR REPLACE PROCEDURE SafeTransferFunds(p\_FromAcct NUMBER, p\_ToAcct NUMBER, p\_Amount NUMBER) IS

BEGIN

  UPDATE Accounts SET Balance = Balance - p\_Amount WHERE AccountID = p\_FromAcct;

  IF SQL%ROWCOUNT = 0 THEN RAISE\_APPLICATION\_ERROR(-20001, 'Source account not found'); END IF;

  UPDATE Accounts SET Balance = Balance + p\_Amount WHERE AccountID = p\_ToAcct;

  IF SQL%ROWCOUNT = 0 THEN RAISE\_APPLICATION\_ERROR(-20002, 'Destination account not found'); END IF;

  COMMIT;

EXCEPTION

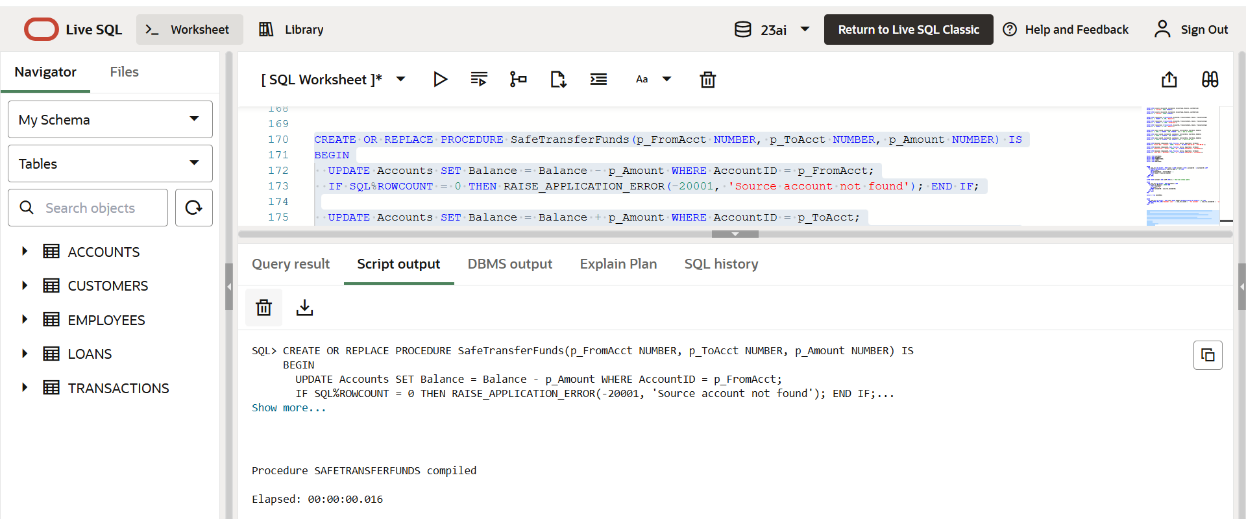
  WHEN OTHERS THEN

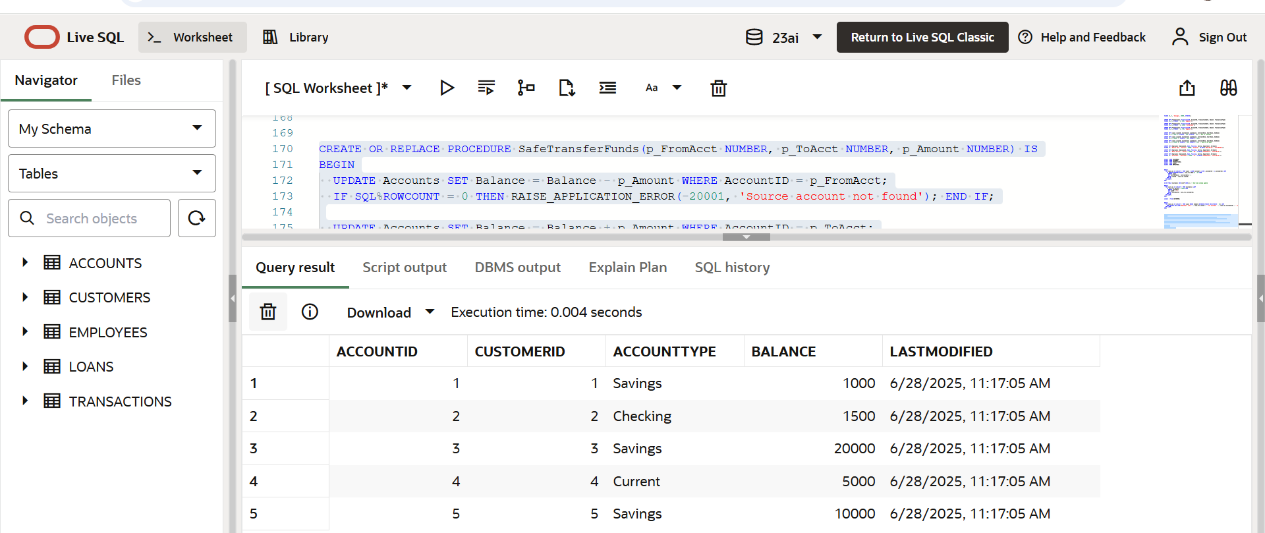
    ROLLBACK;

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

END;

select \* from ACCOUNTS;





**Scenario 2:** Manage errors when updating employee salaries.

* + **Question:** Write a stored procedure **UpdateSalary** that increases the salary of an employee by a given percentage. If the employee ID does not exist, handle the exception and log an error message.

**Code :**

CREATE OR REPLACE PROCEDURE UpdateSalary(p\_EmpID NUMBER, p\_Percent NUMBER) IS

BEGIN

  UPDATE Employees

  SET Salary = Salary + (Salary \* p\_Percent / 100)

  WHERE EmployeeID = p\_EmpID;

  IF SQL%ROWCOUNT = 0 THEN

    DBMS\_OUTPUT.PUT\_LINE('Error: Employee not found.');

  END IF;

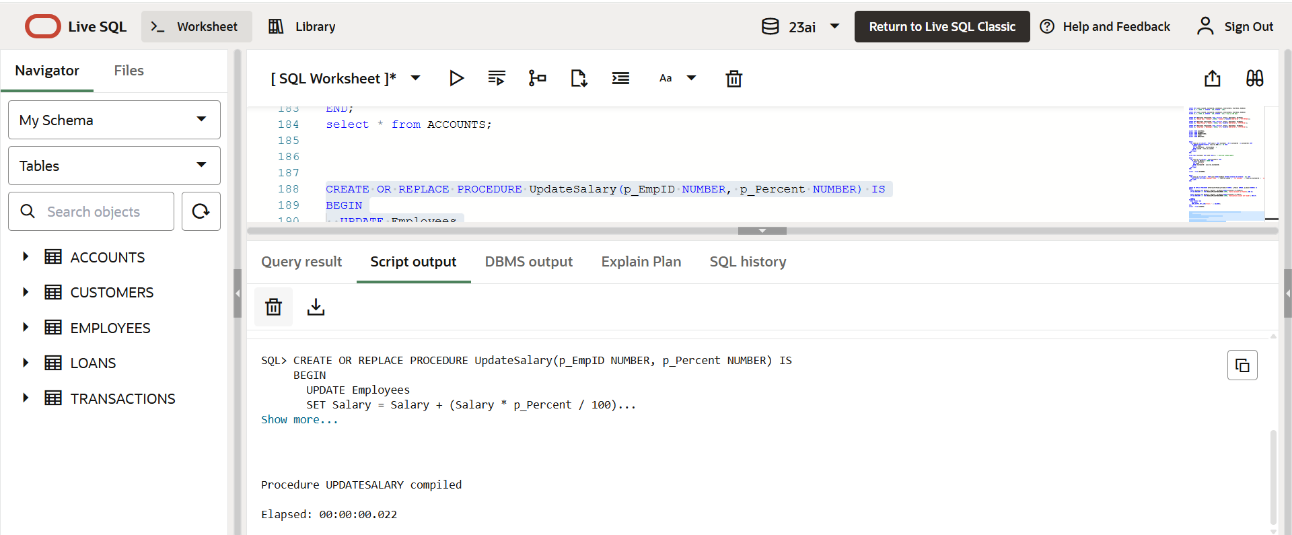
EXCEPTION

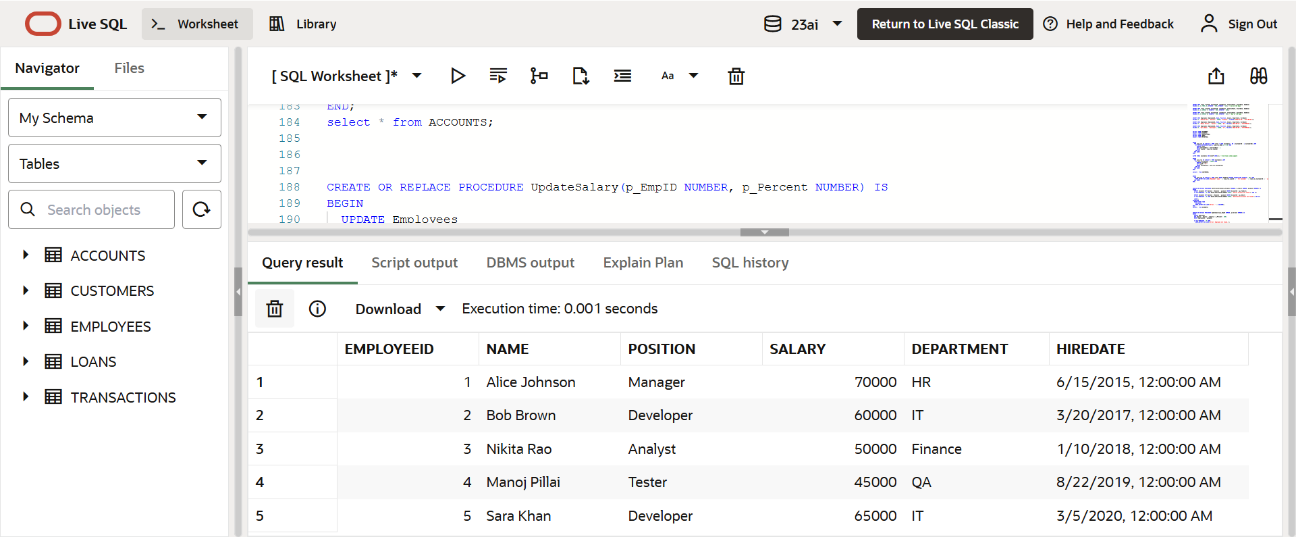
  WHEN OTHERS THEN

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

END;

select \* from EMPLOYEES;





**Scenario 3:** Ensure data integrity when adding a new customer.

* + **Question:** Write a stored procedure **AddNewCustomer** that inserts a new customer into the Customers table. If a customer with the same ID already exists, handle the exception by logging an error and preventing the insertion.

**Code :**

CREATE OR REPLACE PROCEDURE AddNewCustomer(p\_ID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS

BEGIN

  INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

  VALUES (p\_ID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

EXCEPTION

  WHEN DUP\_VAL\_ON\_INDEX THEN

    DBMS\_OUTPUT.PUT\_LINE('Error: Customer with this ID already exists.');

  WHEN OTHERS THEN

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

END;

select \* from CUSTOMERS;

