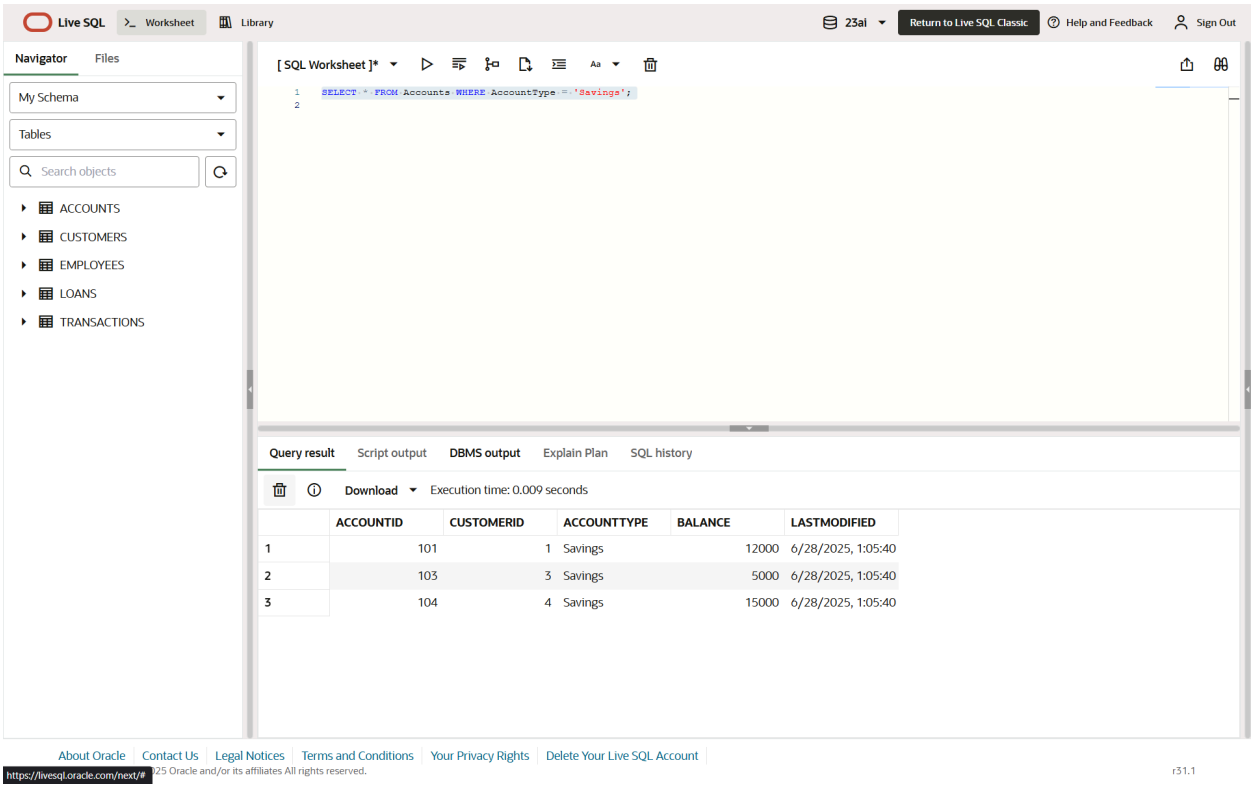
EXERCISE 3 Stored Procedures

→ Scenario 1: The bank needs to process monthly interest for all savings accounts

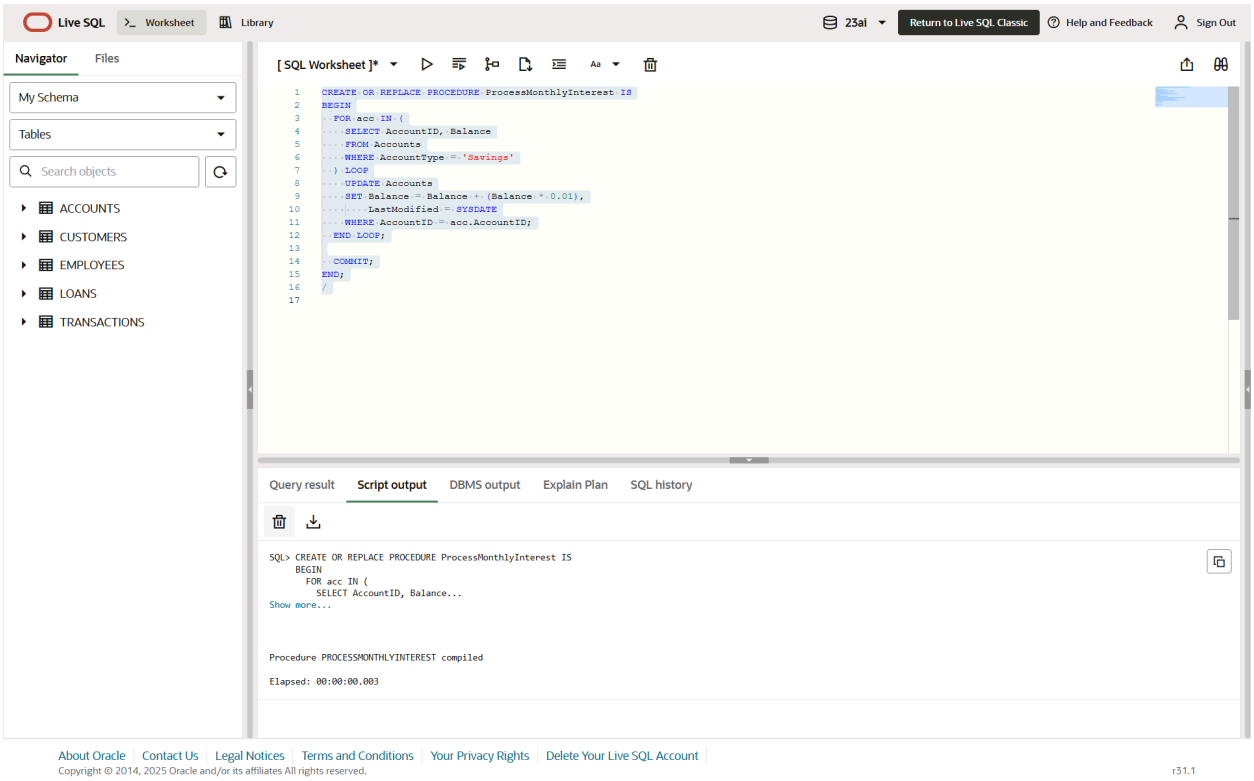
- before



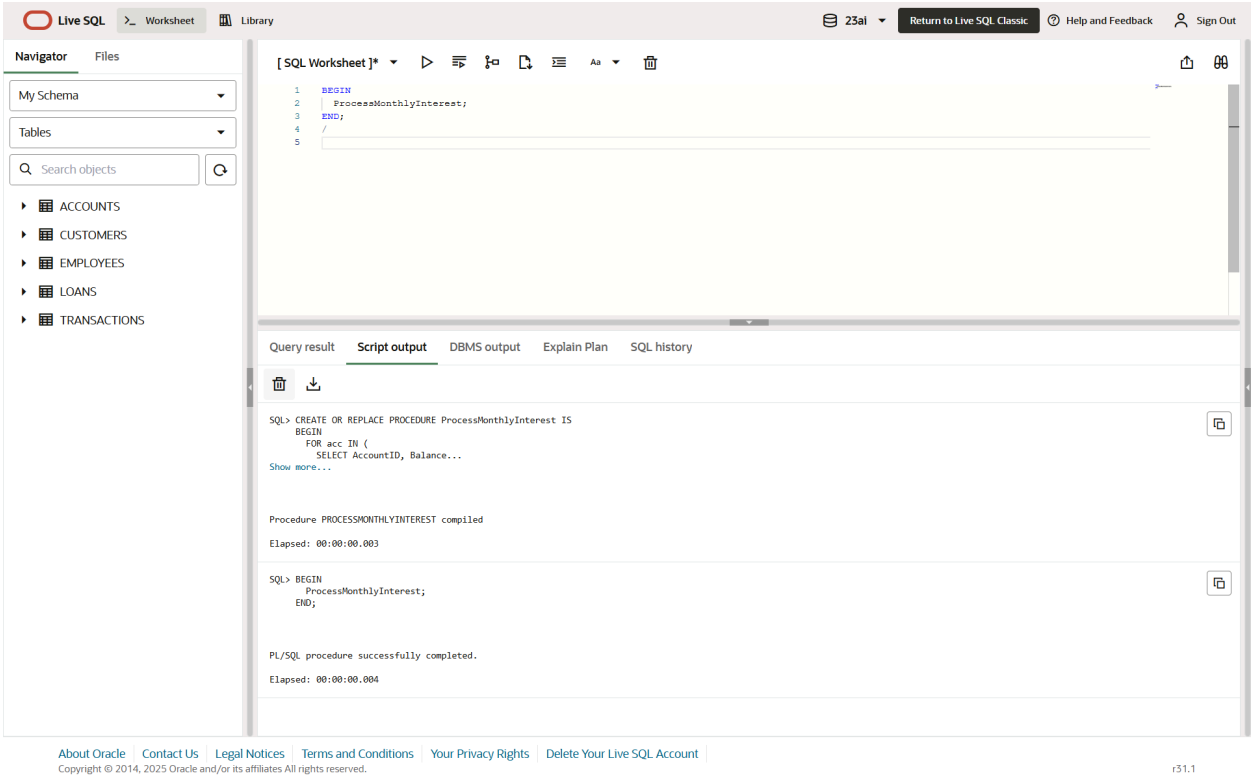
- Stored Procedure: `ProcessMonthlyInterest`

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
BEGIN
  FOR acc IN (
    SELECT AccountID, Balance
    FROM Accounts
    WHERE AccountType = 'Savings'
  ) LOOP
    UPDATE Accounts
```

```
    SET Balance = Balance + (Balance * 0.01),
        LastModified = SYSDATE
    WHERE AccountID = acc.AccountID;
  END LOOP;

  COMMIT;
END;
```

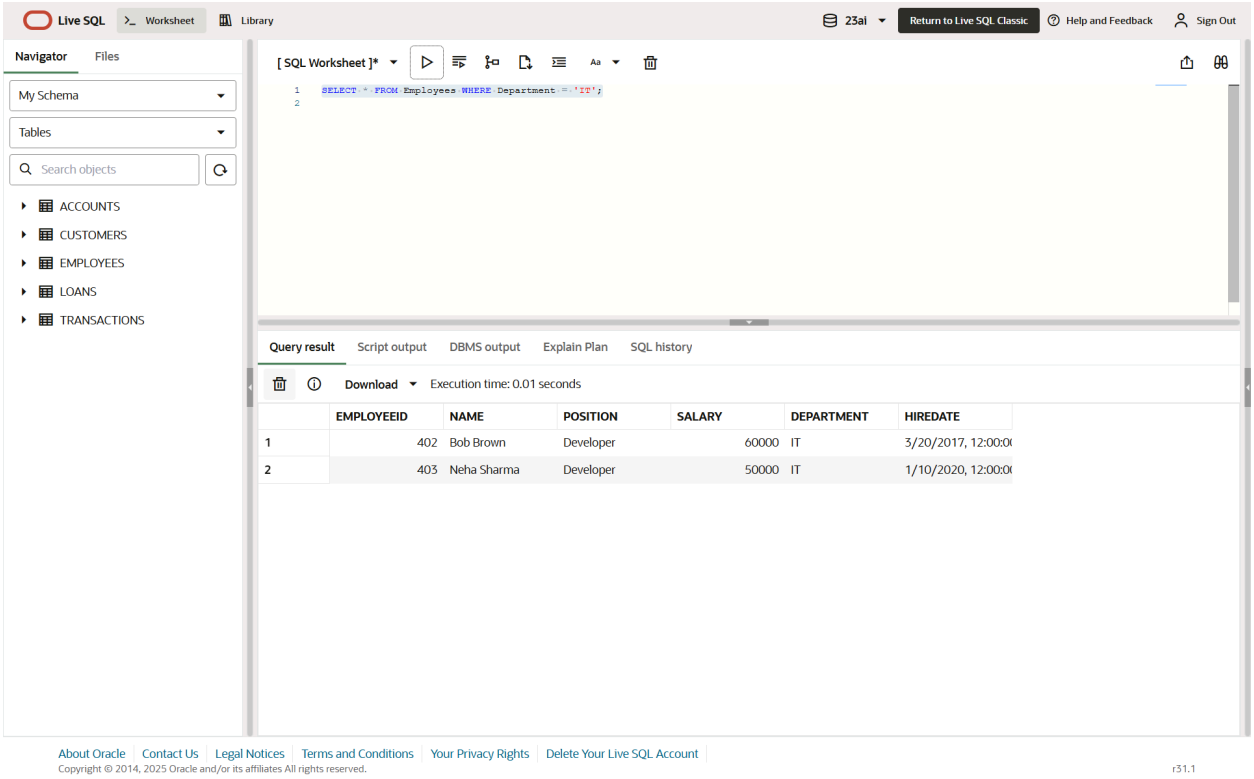


- Output



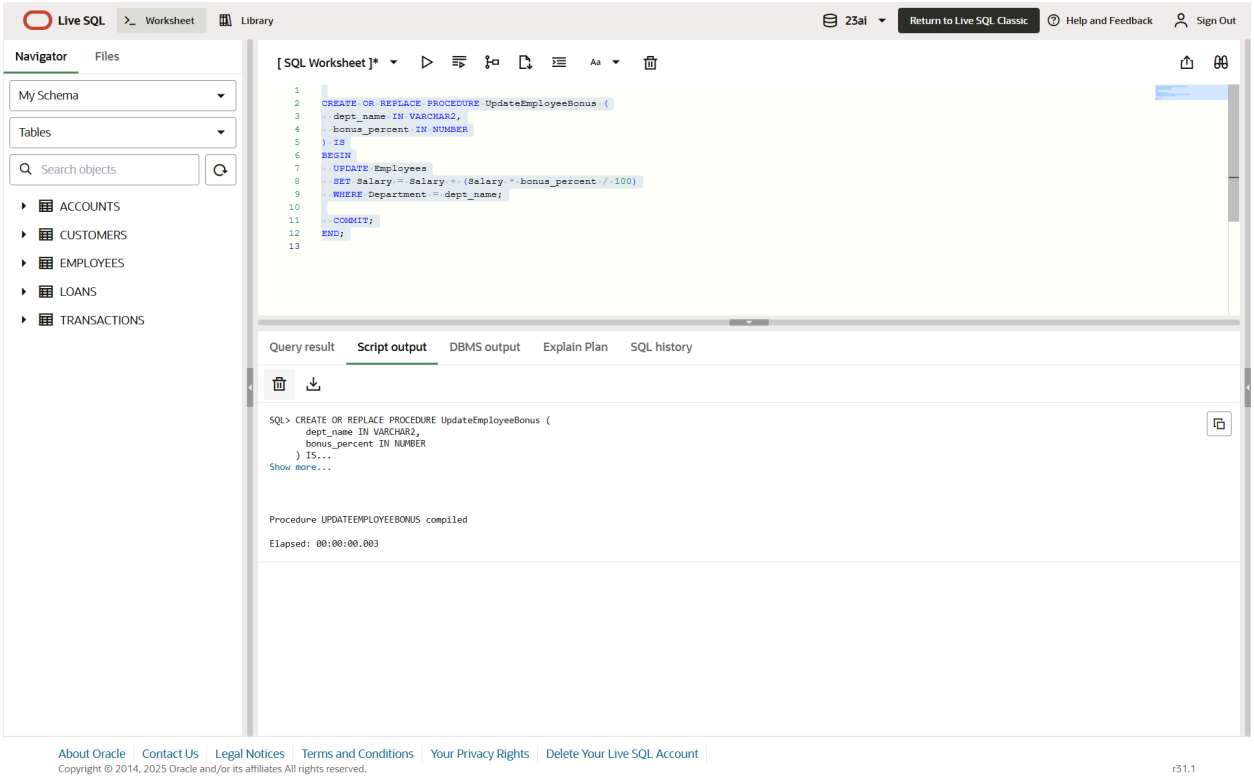
→ Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

- Before

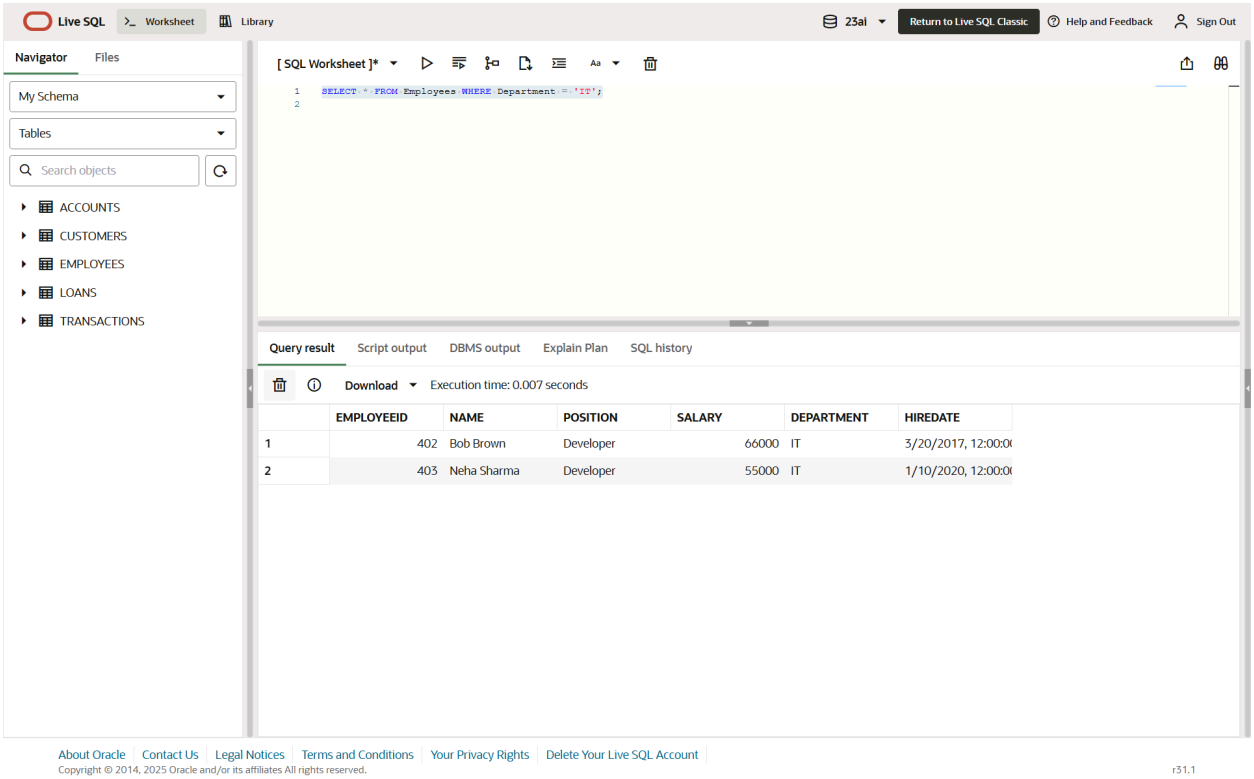


- **Stored Procedure:** `UpdateEmployeeBonus`

```
sql
CopyEdit
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;
```

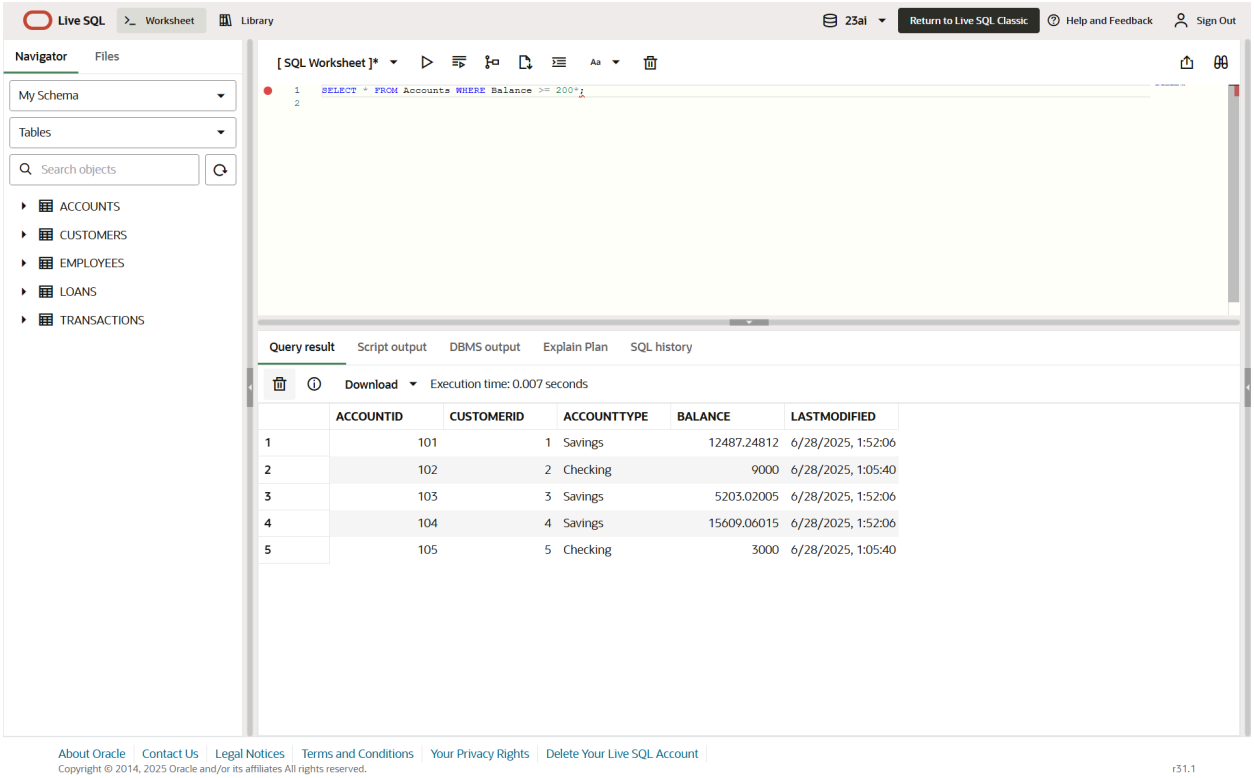

Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Help and Feedback
Sign Out
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
dept_name IN VARCHAR2,
bonus_percent IN NUMBER
) IS
BEGIN
UPDATE Employees
SET Salary = Salary + (Salary * bonus_percent / 100)
WHERE Department = dept_name;
COMMIT;
END;
Query result
Script output
DBMS output
Explain Plan
SQL history
SQL> CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
dept_name IN VARCHAR2,
bonus_percent IN NUMBER
) IS...
Show more...
Procedure UPDATEEMPLOYEEBONUS compiled
Elapsed: 00:00:00.003
About Oracle
Contact Us
Legal Notices
Terms and Conditions
Your Privacy Rights
Delete Your Live SQL Account
Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.
r31.1

- output


Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Help and Feedback
Sign Out
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
SELECT * FROM Employees WHERE Department = 'IT';
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.007 seconds
EMPLOYEEID
NAME
POSITION
SALARY
DEPARTMENT
HIREDATE
1
402
Bob Brown
Developer
66000
IT
3/20/2017, 12:00:0
2
403
Neha Sharma
Developer
55000
IT
1/10/2020, 12:00:0
About Oracle
Contact Us
Legal Notices
Terms and Conditions
Your Privacy Rights
Delete Your Live SQL Account
Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.
r31.1

→ Scenario 3: Customers should be able to transfer funds between their accounts.

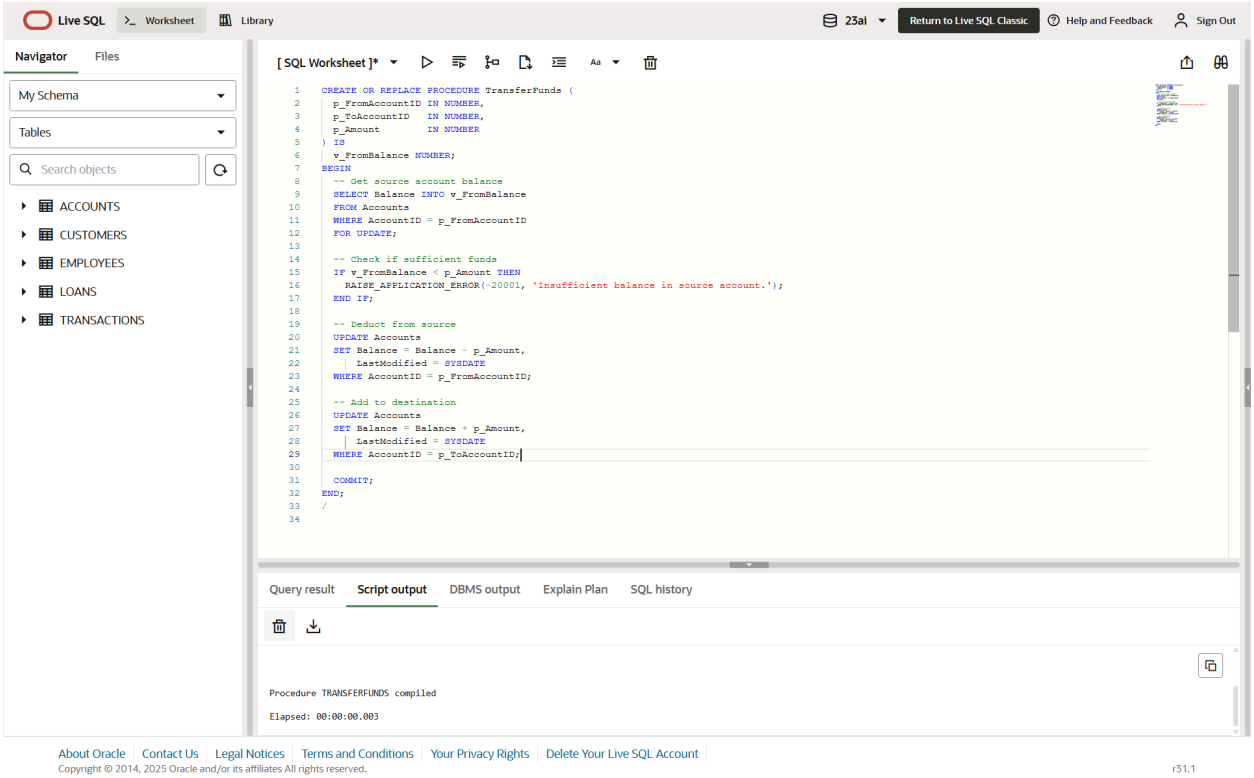
- Before



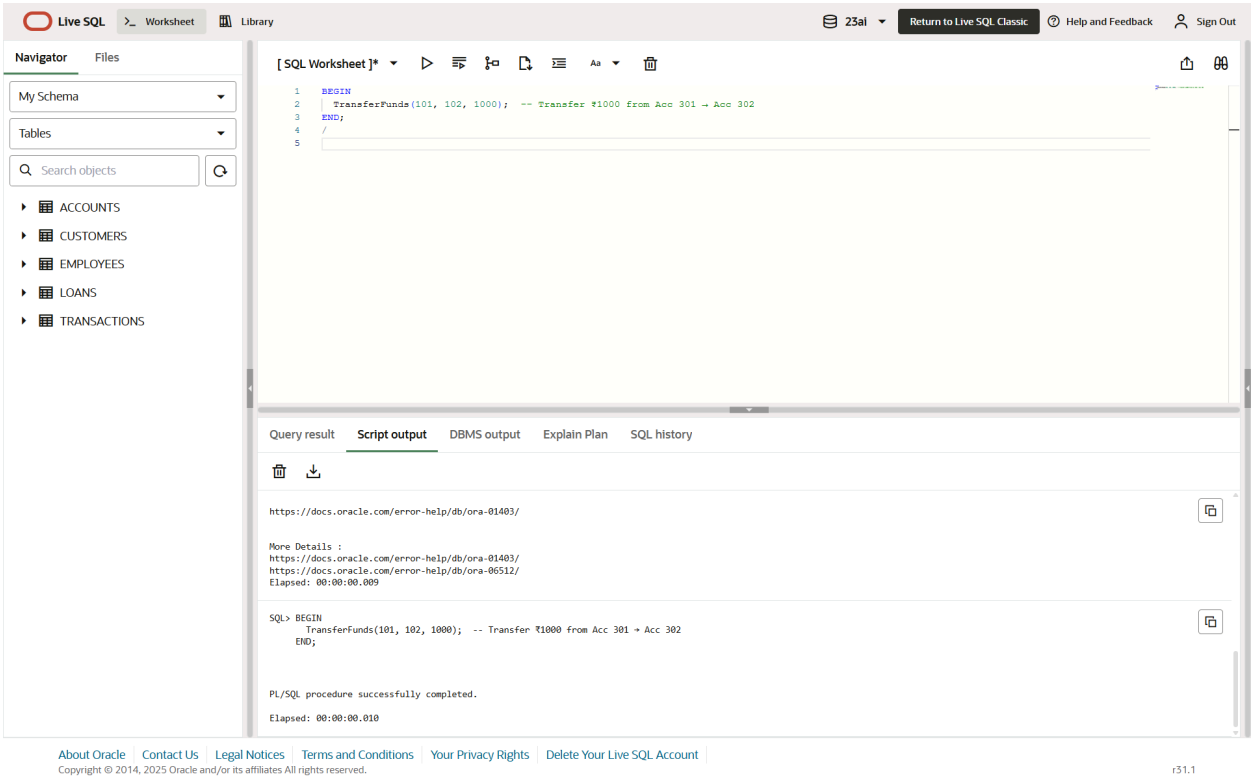
- **Stored Procedure** `TransferFunds`

```
CREATE OR REPLACE PROCEDURE TransferFunds (
  p_FromAccountID IN NUMBER,
  p_ToAccountID   IN NUMBER,
  p_Amount        IN NUMBER
) IS
  v_FromBalance NUMBER;
BEGIN
  -- Get source account balance
  SELECT Balance INTO v_FromBalance
  FROM Accounts
  WHERE AccountID = p_FromAccountID
```

```
  FOR UPDATE;

  -- Check if sufficient funds
  IF v_FromBalance < p_Amount THEN
    RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account.')
  END IF;

  -- Deduct from source
  UPDATE Accounts
  SET Balance = Balance - p_Amount,
      LastModified = SYSDATE
  WHERE AccountID = p_FromAccountID;

  -- Add to destination
  UPDATE Accounts
  SET Balance = Balance + p_Amount,
      LastModified = SYSDATE
  WHERE AccountID = p_ToAccountID;

  COMMIT;
END;
/
```


Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Help and Feedback
Sign Out
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
CREATE OR REPLACE PROCEDURE TransferFunds (
p_FromAccountID IN NUMBER,
p_ToAccountID IN NUMBER,
p_Amount IN NUMBER
) IS
v_FromBalance NUMBER;
BEGIN
-- Get source account balance
SELECT Balance INTO v_FromBalance
FROM Accounts
WHERE AccountID = p_FromAccountID
FOR UPDATE;
-- Check if sufficient funds
IF v_FromBalance < p_Amount THEN
RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account.');
END IF;
-- Deduct from source
UPDATE Accounts
SET Balance = Balance - p_Amount,
LastModified = SYSDATE
WHERE AccountID = p_FromAccountID;
-- Add to destination
UPDATE Accounts
SET Balance = Balance + p_Amount,
LastModified = SYSDATE
WHERE AccountID = p_ToAccountID;
COMMIT;
END;
/
Query result
Script output
DBMS output
Explain Plan
SQL history
Procedure TRANSFERFUNDS compiled
Elapsed: 00:00:00.003
About Oracle
Contact Us
Legal Notices
Terms and Conditions
Your Privacy Rights
Delete Your Live SQL Account
Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.
r31.1

- Output


Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Help and Feedback
Sign Out
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
BEGIN
TransferFunds(101, 102, 1000); -- Transfer ₹1000 from Acc 301 → Acc 302
END;
/
Query result
Script output
DBMS output
Explain Plan
SQL history
https://docs.oracle.com/error-help/db/ora-01403/
More Details :
https://docs.oracle.com/error-help/db/ora-01403/
https://docs.oracle.com/error-help/db/ora-06512/
Elapsed: 00:00:00.009
SQL> BEGIN
TransferFunds(101, 102, 1000); -- Transfer ₹1000 from Acc 301 → Acc 302
END;
PL/SQL procedure successfully completed.
Elapsed: 00:00:00.010
About Oracle
Contact Us
Legal Notices
Terms and Conditions
Your Privacy Rights
Delete Your Live SQL Account
Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.
r31.1