

### Exercise 3: Stored Procedures

```
CREATE TABLE Customers_01 (  
  CustomerID NUMBER PRIMARY KEY,  
  Name VARCHAR2(100),  
  DOB DATE,  
  Balance NUMBER,  
  LastModified DATE  
);
```

e

```
CREATE TABLE Accounts_01 (  
  AccountID NUMBER PRIMARY KEY,  
  CustomerID NUMBER,  
  AccountType VARCHAR2(20),  
  Balance NUMBER,  
  LastModified DATE,  
  FOREIGN KEY (CustomerID) REFERENCES Customers_01(CustomerID)  
);
```

```
CREATE TABLE Transactions_01 (  
  TransactionID NUMBER PRIMARY KEY,  
  AccountID NUMBER,  
  TransactionDate DATE,  
  Amount NUMBER,  
  TransactionType VARCHAR2(10),  
  FOREIGN KEY (AccountID) REFERENCES Accounts_01(AccountID)  
);
```

```
CREATE TABLE Loans_01 (  
  LoanID NUMBER PRIMARY KEY,  
  CustomerID NUMBER,  
  LoanAmount NUMBER,  
  InterestRate NUMBER,  
  StartDate DATE,  
  EndDate DATE,  
  FOREIGN KEY (CustomerID) REFERENCES Customers_01(CustomerID)  
);
```

```
CREATE TABLE Employees_01 (  
EmployeeID NUMBER PRIMARY KEY,  
Name VARCHAR2(100),  
Position VARCHAR2(50),  
Salary NUMBER,  
Department VARCHAR2(50),  
HireDate DATE  
);  
DROP TABLE Transactions CASCADE CONSTRAINTS;  
DROP TABLE Accounts CASCADE CONSTRAINTS;  
DROP TABLE Loans CASCADE CONSTRAINTS;  
DROP TABLE Employees CASCADE CONSTRAINTS;  
DROP TABLE Customers CASCADE CONSTRAINTS;  
  
-- Customers  
INSERT INTO Customers_01 VALUES (1, 'John Doe', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);  
INSERT INTO Customers_01 VALUES (2, 'Jane Smith', TO_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);  
  
-- Accounts  
INSERT INTO Accounts_01 VALUES (1, 1, 'Savings', 1000, SYSDATE);  
INSERT INTO Accounts_01 VALUES (2, 2, 'Checking', 1500, SYSDATE);  
  
-- Transactions  
INSERT INTO Transactions_01 VALUES (1, 1, SYSDATE, 200, 'Deposit');  
INSERT INTO Transactions_01 VALUES (2, 2, SYSDATE, 300, 'Withdrawal');  
  
-- Loans  
INSERT INTO Loans_01 VALUES (1, 1, 5000, 5, SYSDATE, ADD_MONTHS(SYSDATE, 60));  
  
-- Employees  
INSERT INTO Employees_01 VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR',  
TO_DATE('2015-06-15', 'YYYY-MM-DD'));  
INSERT INTO Employees_01 VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT',  
TO_DATE('2017-03-20', 'YYYY-MM-DD'));  
  
COMMIT;
```

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest_01 IS
BEGIN
UPDATE Accounts_01
SET Balance = Balance * 1.01,
LastModified = SYSDATE
WHERE LOWER(AccountType) = 'savings';
END;
/
```

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus_01 (
dept_name IN VARCHAR2,
bonus_percent IN NUMBER
) IS
BEGIN
UPDATE Employees_01
SET Salary = Salary + (Salary * bonus_percent / 100)
WHERE Department = dept_name;
END;
/
```

```
CREATE OR REPLACE PROCEDURE TransferFunds_01 (
from_account IN NUMBER,
to_account IN NUMBER,
amount IN NUMBER
) IS
from_balance NUMBER;
new_txn_id NUMBER;
BEGIN
-- Check and lock source account
SELECT Balance INTO from_balance
FROM Accounts_01
WHERE AccountID = from_account
FOR UPDATE;

IF from_balance < amount THEN
RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account');
END IF;

-- Deduct from source
UPDATE Accounts_01
```

```
SET Balance = Balance - amount,  
LastModified = SYSDATE  
WHERE AccountID = from_account;
```

```
-- Add to destination
```

```
UPDATE Accounts_01  
SET Balance = Balance + amount,  
LastModified = SYSDATE  
WHERE AccountID = to_account;
```

```
-- Get next transaction ID
```

```
SELECT NVL(MAX(TransactionID), 0) + 1 INTO new_txn_id FROM Transactions_01;
```

```
-- Log both sides
```

```
INSERT INTO Transactions_01 VALUES (new_txn_id, from_account, SYSDATE, amount,  
'Transfer-Out');
```

```
INSERT INTO Transactions_01 VALUES (new_txn_id + 1, to_account, SYSDATE, amount,  
'Transfer-In');
```

```
END;
```

```
/
```

```
EXEC ProcessMonthlyInterest_01;
```

```
SELECT * FROM Accounts_01;
```

```
EXEC UpdateEmployeeBonus_01('IT', 10);
```

```
SELECT * FROM Employees_01;
```

```
EXEC TransferFunds_01(1, 2, 200);
```

```
SELECT * FROM Accounts_01;
```

```
SELECT * FROM Transactions_01;
```

Oracle Live SQL

livesql.oracle.com/next/

School

Live SQL

Worksheet

Library

23ai

Return to Live SQL Classic

Help and Feedback

Sign Out

Navigator

Files

My Schema

Tables

Search objects

ACCOUNTS\_01

CUSTOMERS\_01

EMPLOYEES\_01

LOANS\_01

TRANSACTIONS\_01

[ SQL Worksheet ]\*

END;

/

EXEC ProcessMonthlyInterest\_01;

SELECT \* FROM Accounts\_01;

EXEC UpdateEmployeeBonus\_01('IT', 10);

SELECT \* FROM Employees\_01;

Query result

Script output

DBMS output

Explain Plan

SQL history

Download

Execution time: 0.008 seconds

	EMPLOYEEID	NAME	POSITION	SALARY	DEPARTMENT	HIREDATE
1	1	Alice Johnson	Manager	70000	HR	6/15/2015, 12:00:00
2	2	Bob Brown	Developer	66000	IT	3/20/2017, 12:00:00

Oracle Live SQL

livesql.oracle.com/next/

To exit full screen, press and hold **esc**

Live SQL Worksheet Library 23ai Return to Live SQL Classic Help and Feedback Sign Out

Navigator Files

My Schema

Tables

Search objects

ACCOUNTS\_01  
CUSTOMERS\_01  
EMPLOYEES\_01  
LOANS\_01  
TRANSACTIONS\_01

[SQL Worksheet]\*

```
134 END;  
135 /  
136 EXEC ProcessMonthlyInterest_01;  
137 SELECT * FROM Accounts_01;  
138
```

Query result Script output DBMS output Explain Plan SQL history

Download Execution time: 0.007 seconds

	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	1010	6/27/2025, 12:44:4t
2	2	2	Checking	1500	6/27/2025, 12:41:2t

About Oracle Contact Us Legal Notices Terms and Conditions Your Privacy Rights Delete Your Live SQL Account Cookie Preferences

Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.

r31.1

Oracle Live SQL

livesql.oracle.com/next/

To exit full screen, press and hold **esc**

Live SQL Worksheet Library 23ai Return to Live SQL Classic Help and Feedback Sign Out

Navigator Files

My Schema

Tables

Search objects

ACCOUNTS\_01  
CUSTOMERS\_01  
EMPLOYEES\_01  
LOANS\_01  
TRANSACTIONS\_01

[SQL Worksheet]\*

```
136 EXEC ProcessMonthlyInterest_01;  
137 SELECT * FROM Accounts_01;  
138  
139  
140 EXEC UpdateEmployeeBonus_01('IT', 10);  
141 SELECT * FROM Employees_01;  
142  
143  
144 EXEC TransferFunds_01(1, 2, 200);  
145 SELECT * FROM Accounts_01;  
146 SELECT * FROM Transactions_01;  
147
```

Query result Script output DBMS output Explain Plan SQL history

Download Execution time: 0.001 seconds

	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	1010	6/27/2025, 12:44:4t
2	2	2	Checking	1500	6/27/2025, 12:41:2t

About Oracle Contact Us Legal Notices Terms and Conditions Your Privacy Rights Delete Your Live SQL Account Cookie Preferences

Copyright © 2014, 2025 Oracle and/or its affiliates All rights reserved.

r31.1