**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.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

AS

BEGIN

UPDATE accounts

SET balance=balance\*1.01

WHERE accounttype='Savings';

COMMIT;

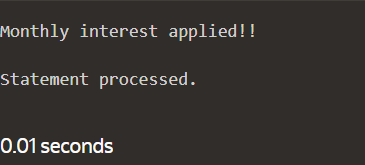
DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied!!');

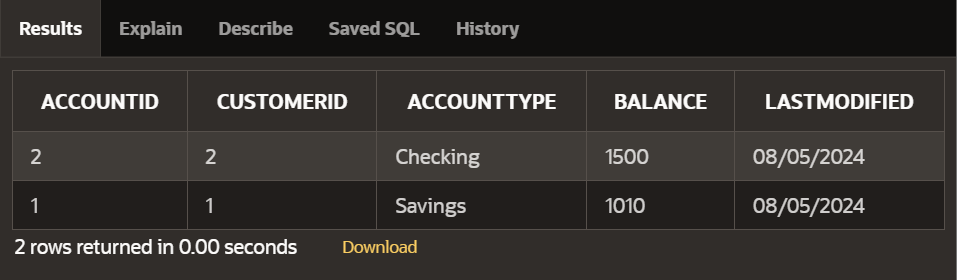
END;

BEGIN

ProcessMonthlyInterest;

END

****

****

**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.

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

dept in VARCHAR2,

bonus\_percentage in int

)

AS

BEGIN

UPDATE employees

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

WHERE department=dept;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to employees in '|| dept ||' Department');

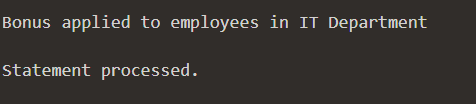
END;

BEGIN

UpdateEmployeeBonus('IT',10);

END





**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.

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_accid IN NUMBER,

to\_accid IN NUMBER,

transfer\_amount IN NUMBER

)

AS

from\_balance NUMBER;

to\_balance NUMBER;

insufficient\_balance EXCEPTION;

BEGIN

SELECT balance INTO from\_balance

FROM accounts

WHERE accountid=from\_accid;

IF from\_balance<transfer\_amount THEN

RAISE insufficient\_balance;

END IF;

UPDATE accounts

SET balance=balance-transfer\_amount

WHERE accountid = from\_accid;

UPDATE accounts

SET balance=balance+transfer\_amount

WHERE accountid=to\_accid;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer Completed Successfully!');

EXCEPTION

WHEN insufficient\_balance THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance in the source account.');

ROLLBACK;

WHEN OTHERS THEN

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

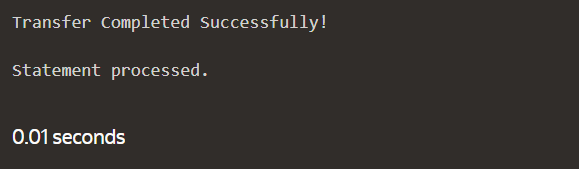
ROLLBACK;

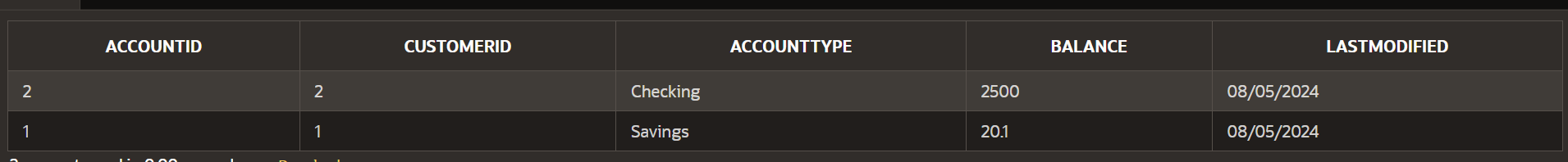
END;

BEGIN

TransferFunds(1,2,1000);

END



****