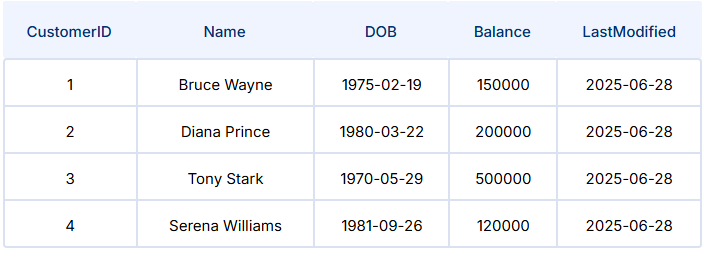
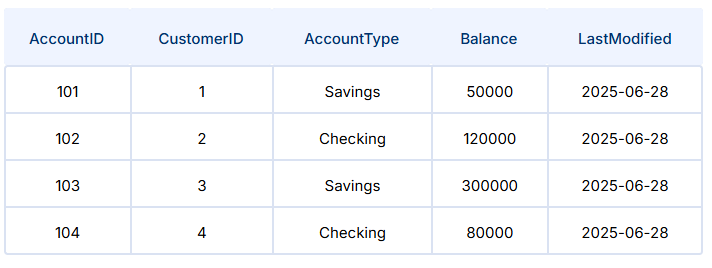
**Exercise 3: Stored Procedures**

Sample tables in database after rows insertion:

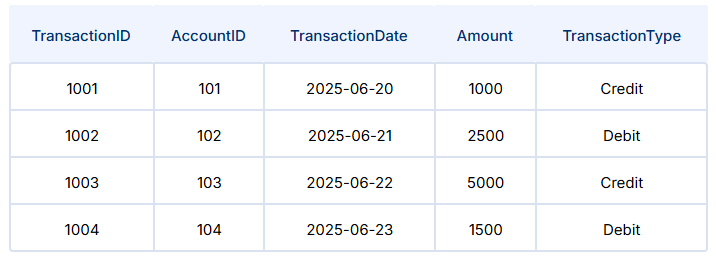
CUSTOMERS



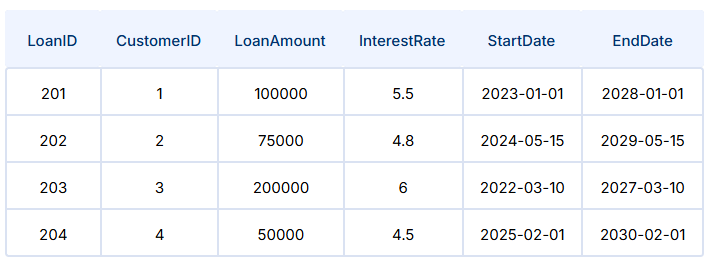
ACCOUNTS



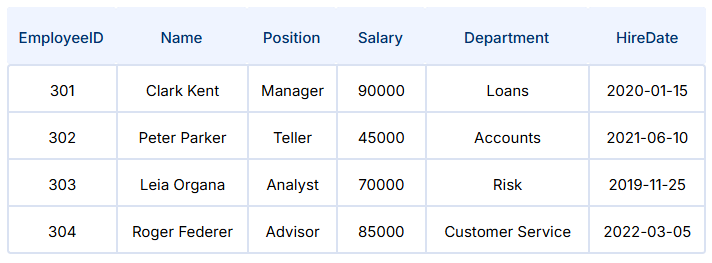
TRANSACTIONS



LOANS



EMPLOYEES



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.

Code:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

    -- Update all savings accounts by applying 1% interest

    UPDATE Accounts

    SET Balance = Balance \* 1.01

    WHERE AccountType = 'Savings';

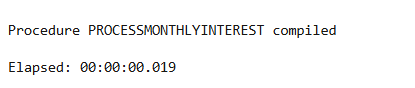
    COMMIT;

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

END;

/

Output:



SET SERVEROUTPUT ON;

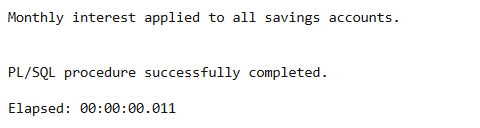
BEGIN

ProcessMonthlyInterest;

END;

/

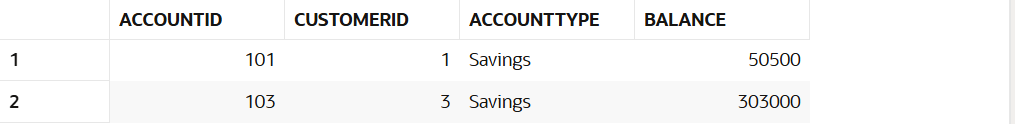
Output:



SELECT AccountID, CustomerID, AccountType, Balance

FROM Accounts

WHERE AccountType = 'Savings';



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.

Code:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    p\_department     IN VARCHAR2,

    p\_bonus\_percent  IN NUMBER

) AS

BEGIN

    -- Update salary by bonus percentage for given department

    UPDATE Employees

    SET Salary = Salary + (Salary \* p\_bonus\_percent / 100)

    WHERE Department = p\_department;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE(

        'Bonuses applied to department: ' || p\_department ||

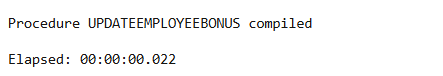
        ' with ' || p\_bonus\_percent || '% bonus.'

    );

END;

/

Output:



SET SERVEROUTPUT ON;

BEGIN

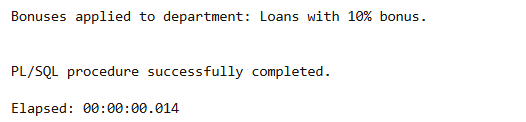
    -- Example: Add 10% bonus to all employees in 'Loans' department

    UpdateEmployeeBonus('Loans', 10);

END;

/

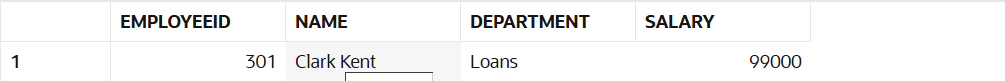
Output:



SELECT EmployeeID, Name, Department, Salary

FROM Employees

WHERE Department = 'Loans';



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.

Code:

CREATE OR REPLACE PROCEDURE TransferFunds (

    p\_from\_account\_id IN NUMBER,

    p\_to\_account\_id   IN NUMBER,

    p\_amount          IN NUMBER

) AS

    v\_from\_balance Accounts.Balance%TYPE;

    v\_to\_exists     NUMBER;

BEGIN

    -- Validate positive transfer amount

    IF p\_amount <= 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Transfer amount must be greater than zero.');

    END IF;

    -- Check if source account exists and get its balance

    SELECT Balance

    INTO v\_from\_balance

    FROM Accounts

    WHERE AccountID = p\_from\_account\_id;

    -- Check if destination account exists

    SELECT COUNT(\*) INTO v\_to\_exists

    FROM Accounts

    WHERE AccountID = p\_to\_account\_id;

    IF v\_to\_exists = 0 THEN

        RAISE\_APPLICATION\_ERROR(-20002, 'Destination account does not exist.');

    END IF;

    -- Check for sufficient balance

    IF v\_from\_balance < p\_amount THEN

        RAISE\_APPLICATION\_ERROR(-20003, 'Insufficient funds in source account.');

    END IF;

    -- Perform transfer: deduct and add balance

    UPDATE Accounts

    SET Balance = Balance - p\_amount

    WHERE AccountID = p\_from\_account\_id;

    UPDATE Accounts

    SET Balance = Balance + p\_amount

    WHERE AccountID = p\_to\_account\_id;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Transfer of $' || p\_amount || ' from Account ' ||

                         p\_from\_account\_id || ' to Account ' || p\_to\_account\_id || ' completed successfully.');

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RAISE\_APPLICATION\_ERROR(-20004, 'Source account does not exist.');

    WHEN OTHERS THEN

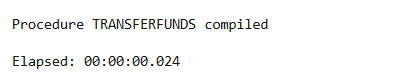
        ROLLBACK;

        RAISE;

END;

/

Output:



SET SERVEROUTPUT ON;

BEGIN

-- Example: Transfer $500 from Account 101 to Account 102

TransferFunds(101, 102, 500);

END;

/

