WEEK\_2

PL/SQL

SCHEMAS :-

*CREATE TABLE Customers (*

*CustomerID NUMBER PRIMARY KEY,*

*Name VARCHAR2(100),*

*DOB DATE,*

*Balance NUMBER,*

*LastModified DATE*

*);*

*CREATE TABLE Accounts (*

*AccountID NUMBER PRIMARY KEY,*

*CustomerID NUMBER,*

*AccountType VARCHAR2(20),*

*Balance NUMBER,*

*LastModified DATE,*

*FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)*

*);*

*CREATE TABLE Transactions (*

*TransactionID NUMBER PRIMARY KEY,*

*AccountID NUMBER,*

*TransactionDate DATE,*

*Amount NUMBER,*

*TransactionType VARCHAR2(10),*

*FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)*

*);*

*CREATE TABLE Loans (*

*LoanID NUMBER PRIMARY KEY,*

*CustomerID NUMBER,*

*LoanAmount NUMBER,*

*InterestRate NUMBER,*

*StartDate DATE,*

*EndDate DATE,*

*FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)*

*);*

*CREATE TABLE Employees (*

*EmployeeID NUMBER PRIMARY KEY,*

*Name VARCHAR2(100),*

*Position VARCHAR2(50),*

*Salary NUMBER,*

*Department VARCHAR2(50),*

*HireDate DATE*

*);*

INPUTS:-

*-- Insert into Customers*

*INSERT INTO Customers VALUES (1, 'Alice Johnson', TO\_DATE('1955-03-15', 'YYYY-MM-DD'), 12000, SYSDATE);*

*INSERT INTO Customers VALUES (2, 'Bob Smith', TO\_DATE('1980-07-22', 'YYYY-MM-DD'), 8000, SYSDATE);*

*INSERT INTO Customers VALUES (3, 'Charlie Davis', TO\_DATE('1990-11-10', 'YYYY-MM-DD'), 9500, SYSDATE);*

*INSERT INTO Customers VALUES (4, 'Diana Prince', TO\_DATE('1948-01-05', 'YYYY-MM-DD'), 15500, SYSDATE);*

*INSERT INTO Customers VALUES (5, 'Evan Stone', TO\_DATE('1975-06-30', 'YYYY-MM-DD'), 7000, SYSDATE);*

*-- Insert into Accounts*

*INSERT INTO Accounts VALUES (101, 1, 'Savings', 5000, SYSDATE);*

*INSERT INTO Accounts VALUES (102, 2, 'Checking', 3000, SYSDATE);*

*INSERT INTO Accounts VALUES (103, 3, 'Savings', 2500, SYSDATE);*

*INSERT INTO Accounts VALUES (104, 4, 'Checking', 9000, SYSDATE);*

*INSERT INTO Accounts VALUES (105, 5, 'Savings', 1500, SYSDATE);*

*-- Insert into Transactions*

*INSERT INTO Transactions VALUES (1001, 101, SYSDATE - 10, 1000, 'DEPOSIT');*

*INSERT INTO Transactions VALUES (1002, 102, SYSDATE - 7, 500, 'WITHDRAW');*

*INSERT INTO Transactions VALUES (1003, 103, SYSDATE - 5, 200, 'DEPOSIT');*

*INSERT INTO Transactions VALUES (1004, 104, SYSDATE - 3, 1200, 'WITHDRAW');*

*INSERT INTO Transactions VALUES (1005, 105, SYSDATE - 1, 800, 'DEPOSIT');*

*-- Insert into Loans*

*INSERT INTO Loans VALUES (201, 1, 100000, 0.08, TO\_DATE('2021-01-01', 'YYYY-MM-DD'), TO\_DATE('2025-07-01', 'YYYY-MM-DD'));*

*INSERT INTO Loans VALUES (202, 2, 75000, 0.07, TO\_DATE('2022-06-15', 'YYYY-MM-DD'), TO\_DATE('2025-07-15', 'YYYY-MM-DD'));*

*INSERT INTO Loans VALUES (203, 3, 50000, 0.09, TO\_DATE('2023-03-10', 'YYYY-MM-DD'), TO\_DATE('2026-04-01', 'YYYY-MM-DD'));*

*INSERT INTO Loans VALUES (204, 4, 125000, 0.065, TO\_DATE('2020-08-20', 'YYYY-MM-DD'), TO\_DATE('2025-07-05', 'YYYY-MM-DD'));*

*INSERT INTO Loans VALUES (205, 5, 30000, 0.1, TO\_DATE('2023-11-01', 'YYYY-MM-DD'), TO\_DATE('2026-11-01', 'YYYY-MM-DD'));*

*-- Insert into Employees*

*INSERT INTO Employees VALUES (301, 'Sophia Lee', 'Manager', 90000, 'Operations', TO\_DATE('2015-04-01', 'YYYY-MM-DD'));*

*INSERT INTO Employees VALUES (302, 'James Kim', 'Teller', 45000, 'Retail', TO\_DATE('2018-10-12', 'YYYY-MM-DD'));*

*INSERT INTO Employees VALUES (303, 'Maria Gomez', 'Analyst', 60000, 'Finance', TO\_DATE('2020-03-05', 'YYYY-MM-DD'));*

*INSERT INTO Employees VALUES (304, 'Robert Chen', 'IT Officer', 75000, 'Technology', TO\_DATE('2016-07-30', 'YYYY-MM-DD'));*

*INSERT INTO Employees VALUES (305, 'Aisha Khan', 'HR Executive', 50000, 'HR', TO\_DATE('2019-01-20', 'YYYY-MM-DD'));*

**Exercise 1: Control Structures**

**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

* + **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**SQL COMMANDS:-**

**BEGIN**

**FOR cust IN (**

**SELECT CustomerID**

**FROM Customers**

**WHERE MONTHS\_BETWEEN(SYSDATE, DOB) / 12 > 60**

**)**

**LOOP**

**UPDATE Loans**

**SET InterestRate = InterestRate - 0.01**

**WHERE CustomerID = cust.CustomerID;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('Interest rates updated for customers over 60.');**

**END;**

**/**

**OUTPUT:**

**Interest rates updated for customers over 60.**

**PL/SQL procedure successfully completed.**

**Elapsed: 00:00:00.091**

****

**Scenario 2:** A customer can be promoted to VIP status based on their balance.

* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

SQL-COMMANDS:-

ALTER TABLE Customers ADD IsVIP VARCHAR2(5);

BEGIN

FOR cust IN (

SELECT CustomerID

FROM Customers

WHERE Balance > 10000

)

LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('VIP status updated for eligible customers.');

END;

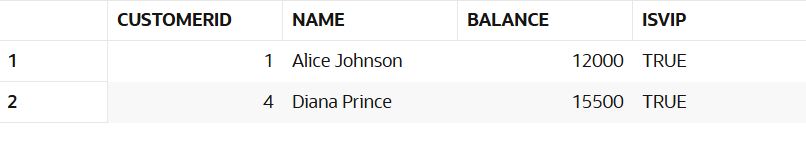
/

OUTPUT:

VIP status updated for eligible customers.

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.086



**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

SQL-COMMANDS:-

BEGIN

FOR loan\_rec IN (

SELECT c.Name, l.EndDate

FROM Loans l

JOIN Customers c ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: ' || loan\_rec.Name ||

', your loan is due on ' || TO\_CHAR(loan\_rec.EndDate, 'DD-Mon-YYYY')

);

END LOOP;

END;

/

OUTPUT:-

Reminder: Alice Johnson, your loan is due on 01-Jul-2025

Reminder: Bob Smith, your loan is due on 15-Jul-2025

Reminder: Diana Prince, your loan is due on 05-Jul-2025

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.020

**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.

SQL COMMAND:-

BEGIN

FOR cust IN (

SELECT CustomerID

FROM Customers

WHERE MONTHS\_BETWEEN(SYSDATE, DOB) / 12 > 60

)

LOOP

UPDATE Loans

SET InterestRate = InterestRate - 0.01

WHERE CustomerID = cust.CustomerID;

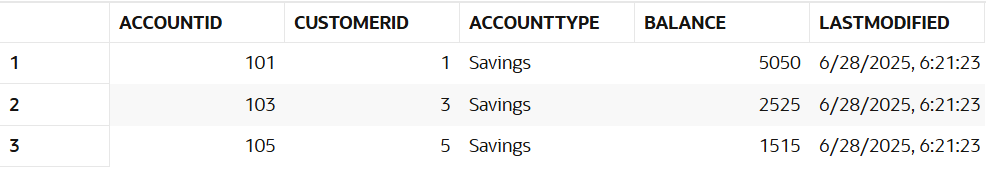
END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Interest rates updated for customers over 60.');

END;

/

OUTPUT:-



**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.

SQL COMMAND:-

ALTER TABLE Customers ADD IsVIP VARCHAR2(5);

BEGIN

FOR cust IN (

SELECT CustomerID

FROM Customers

WHERE Balance > 10000

)

LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust.CustomerID;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('VIP status updated for eligible customers.');

END;

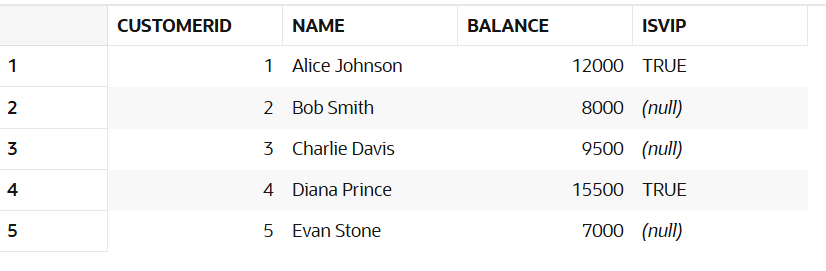
/

SELECT CustomerID, Name, Balance, IsVIP

FROM Customers

ORDER BY CustomerID;

OUTPUT:-



**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.

SQL COMMAND:-

BEGIN

FOR loan\_rec IN (

SELECT c.Name, l.EndDate

FROM Loans l

JOIN Customers c ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: ' || loan\_rec.Name ||

', your loan is due on ' || TO\_CHAR(loan\_rec.EndDate, 'DD-Mon-YYYY')

);

END LOOP;

END;

/

OUTPUT:-

Reminder: Alice Johnson, your loan is due on 01-Jul-2025  
Reminder: Bob Smith, your loan is due on 15-Jul-2025  
Reminder: Diana Prince, your loan is due on 05-Jul-2025  
  
  
PL/SQL procedure successfully completed.