

-- Schema Creation

```
CREATE TABLE Customers (  
    CustomerID NUMBER PRIMARY KEY,  
    Name VARCHAR2(100),  
    DOB DATE,  
    Balance NUMBER,  
    IsVIP BOOLEAN DEFAULT FALSE,  
    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  
);
```

-- Sample Data Insertion

```
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)  
VALUES (1, 'John Doe', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);
```

```
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)  
VALUES (2, 'Jane Smith', TO_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);
```

```
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)  
VALUES (1, 1, 'Savings', 1000, SYSDATE);
```

```
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)  
VALUES (2, 2, 'Checking', 1500, SYSDATE);
```

```
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (1, 1, SYSDATE, 200, 'Deposit');
```

```
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (2, 2, SYSDATE, 300, 'Withdrawal');
```

```
INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)
VALUES (1, 1, 5000, 5, SYSDATE, ADD_MONTHS(SYSDATE, 60));
```

```
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO_DATE('2015-06-15', 'YYYY-MM-DD'));
```

```
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO_DATE('2017-03-20', 'YYYY-MM-DD'));
```

```
-- PL/SQL Blocks
```

```
-- Scenario 1: Apply Discount to Loan Interest Rates for Customers Above 60
```

```
DECLARE
```

```
    v_age NUMBER;
```

```
    v_interest_rate NUMBER;
```

```
    v_new_interest_rate NUMBER;
```

```
BEGIN
```

```
    FOR rec IN (SELECT CustomerID, (FLOOR(MONTHS_BETWEEN(SYSDATE, DOB) / 12)) AS age FROM
Customers)
```

```
    LOOP
```

```
        IF rec.age > 60 THEN
```

```
            SELECT InterestRate
```

```
            INTO v_interest_rate
```

```
            FROM Loans
```

```

WHERE CustomerID = rec.CustomerID;

v_new_interest_rate := v_interest_rate - 1;

UPDATE Loans
SET InterestRate = v_new_interest_rate
WHERE CustomerID = rec.CustomerID;

DBMS_OUTPUT.PUT_LINE('Updated interest rate for CustomerID ' || rec.CustomerID || ' to '
|| v_new_interest_rate || '%');
END IF;
END LOOP;
END;
/

```

-- Scenario 2: Promote Customer to VIP Status Based on Balance

```

BEGIN
FOR rec IN (SELECT CustomerID, Balance FROM Customers)
LOOP
IF rec.Balance > 10000 THEN
UPDATE Customers
SET IsVIP = TRUE
WHERE CustomerID = rec.CustomerID;

DBMS_OUTPUT.PUT_LINE('CustomerID ' || rec.CustomerID || ' is now a VIP.');
```

```

END IF;
END LOOP;
END;
/

```

-- Scenario 3: Send Reminders for Loans Due in the Next 30 Days

DECLARE

 v_customer_name VARCHAR2(100);

 v_due_date DATE;

BEGIN

 FOR rec IN (SELECT LoanID, CustomerID, EndDate FROM Loans WHERE EndDate BETWEEN
SYSDATE AND SYSDATE + 30)

 LOOP

 SELECT Name, EndDate

 INTO v_customer_name, v_due_date

 FROM Customers

 WHERE CustomerID = rec.CustomerID;

 DBMS_OUTPUT.PUT_LINE('Reminder: Loan ' || rec.LoanID || ' for customer ' ||
v_customer_name || ' is due on ' || TO_CHAR(v_due_date, 'YYYY-MM-DD'));

 END LOOP;

END;

/