**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.

BEGIN

FOR cust IN (

SELECT c.CustomerID

FROM Customers c

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

) LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = cust.CustomerID

AND InterestRate > 1; -- Prevent negative integers

UPDATE Customers

SET LastModified = SYSDATE

WHERE CustomerID = cust.CustomerID;

END LOOP;

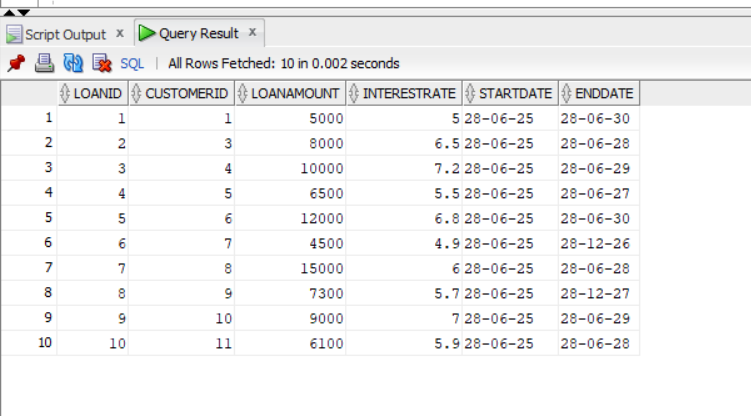
COMMIT;

END;

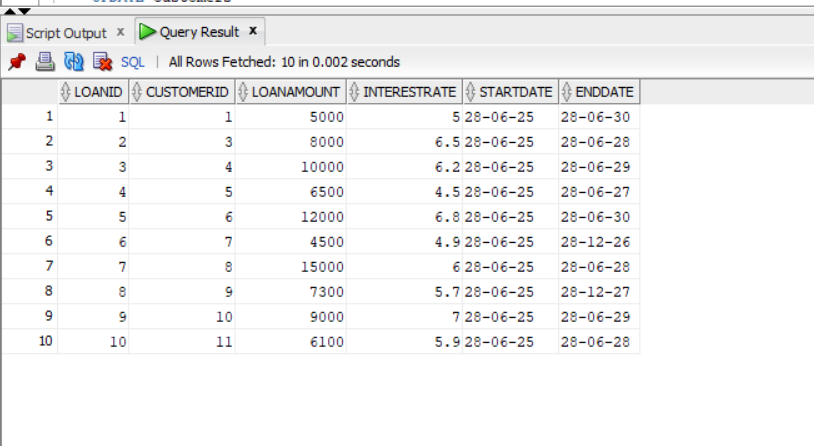
/

Select \* from Loans; //to check the difference between before and after committing

BEFORE:



AFTER:



**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.

ALTER TABLE Customers

ADD isVIP VARCHAR2(3) DEFAULT 'N’ // adding new column to check their status

BEGIN

FOR cust IN (

SELECT CustomerID, Balance

FROM Customers

) LOOP

IF cust.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'Y'

WHERE CustomerID = cust.CustomerID;

ELSE

UPDATE Customers

SET IsVIP = 'N'

WHERE CustomerID = cust.CustomerID;

END IF;

END LOOP;

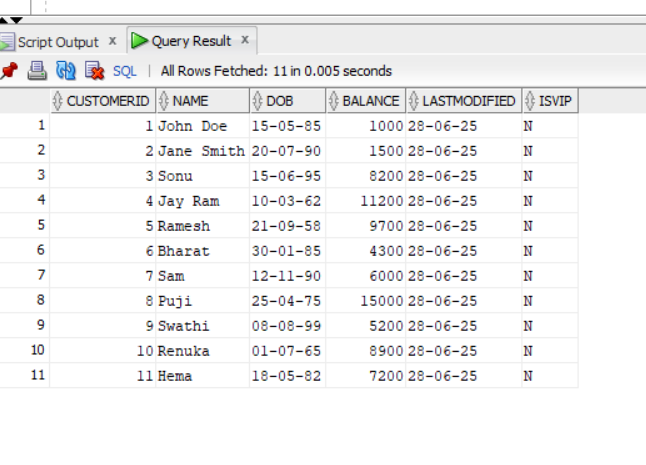
COMMIT;

END;

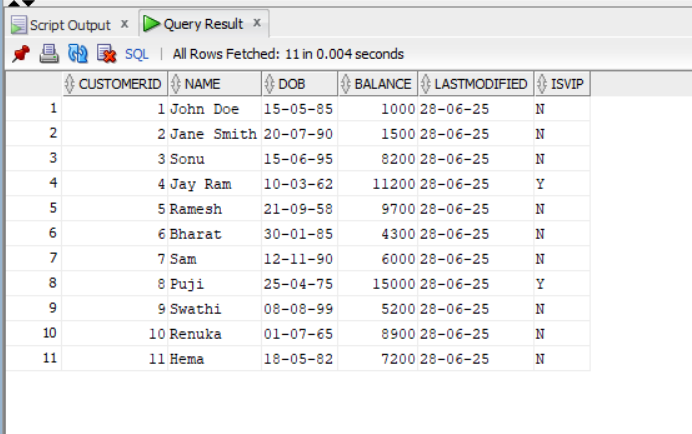
/

Select \* from Customers;//to check the changes before and after committing

BEFORE:



AFTER:



**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.

SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (

SELECT c.Name, c.CustomerID, l.LoanID, l.EndDate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

' Reminder: Loan ID ' || rec.LoanID ||

' for customer "' || rec.Name || '" (CustomerID: ' || rec.CustomerID ||

') is due on ' || TO\_CHAR(rec.EndDate, 'YYYY-MM-DD')

);

END LOOP;

END;