**PL / SQL :**

Table creation :

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

);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified) VALUES (101, 'Arjun Mehta', TO\_DATE('1988-03-12', 'YYYY-MM-DD'), 20000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified) VALUES (102, 'Sneha Reddy', TO\_DATE('1992-09-25', 'YYYY-MM-DD'), 35000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified) VALUES (103, 'Rahul Sharma', TO\_DATE('1985-07-05', 'YYYY-MM-DD'), 15000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (1001, 101, 'Savings', 20000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (1002, 102, 'Checking', 15000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (1003, 103, 'Savings', 15000, SYSDATE);

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (5001, 1001, SYSDATE - 10, 5000, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (5002, 1001, SYSDATE - 5, 2000, 'Withdrawal');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (5003, 1002, SYSDATE - 3, 10000, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType) VALUES (5004, 1003, SYSDATE - 1, 3000, 'Withdrawal');

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (2001, 101, 500000, 6.5, TO\_DATE('2022-01-01', 'YYYY-MM-DD'), TO\_DATE('2027-01-01', 'YYYY-MM-DD'));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (2002, 102, 250000, 7.2, TO\_DATE('2023-06-15', 'YYYY-MM-DD'), TO\_DATE('2028-06-15', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (301, 'Meena Iyer', 'Branch Manager', 85000, 'Operations', TO\_DATE('2016-04-10', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (302, 'Karthik Nair', 'Loan Officer', 60000, 'Loans', TO\_DATE('2018-08-22', 'YYYY-MM-DD'));

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (303, 'Priya Verma', 'Customer Service', 40000, 'Support', TO\_DATE('2020-02-18', 'YYYY-MM-DD'));

**Exercise 1: Control Structures**

BEGIN

FOR rec IN (

SELECT l.LoanID, l.InterestRate, c.DOB

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

) LOOP

IF MONTHS\_BETWEEN(SYSDATE, rec.DOB) / 12 > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

END IF;

END LOOP;

END;

/

SELECT l.LoanID, c.Name, c.DOB, l.InterestRate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID;

ALTER TABLE Customers ADD IsVIP CHAR(1);

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'Y'

WHERE CustomerID = rec.CustomerID;

ELSE

UPDATE Customers

SET IsVIP = 'N'

WHERE CustomerID = rec.CustomerID;

END IF;

END LOOP;

END;

/

SELECT CustomerID, Name, Balance, IsVIP FROM Customers;

UPDATE Loans

SET EndDate = SYSDATE + 10

WHERE LoanID = 2002;

COMMIT;

BEGIN

FOR rec IN (

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

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'📢 Reminder: Loan ' || rec.LoanID || ' for ' || rec.Name ||

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

);

END LOOP;

END;

/

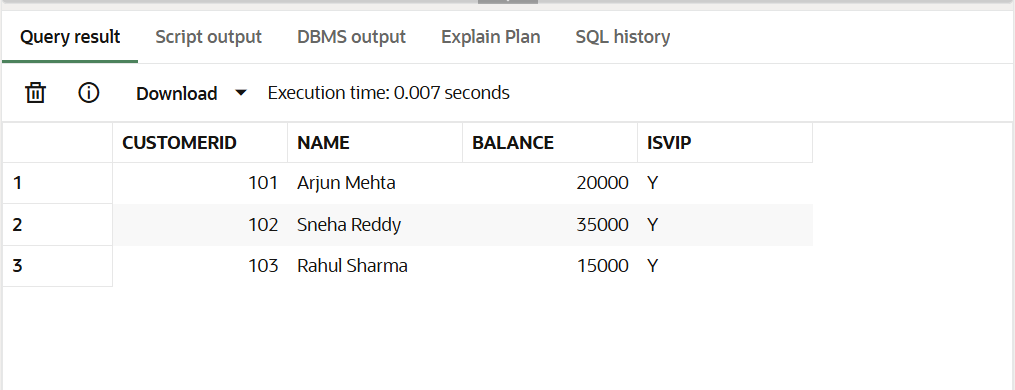
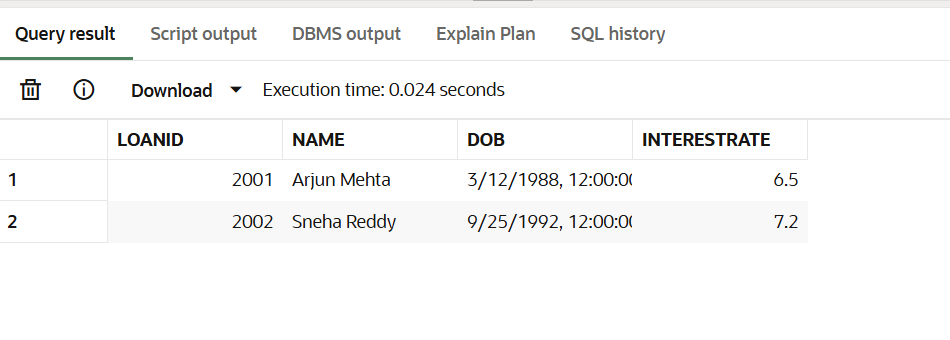
SELECT l.LoanID, c.Name, l.EndDate

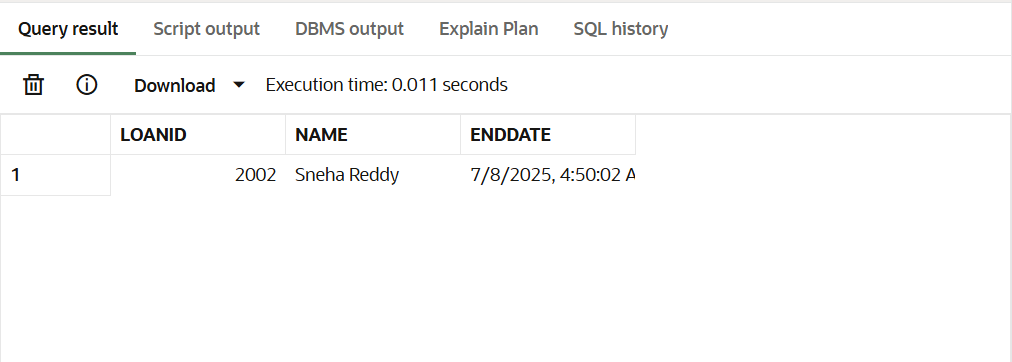
FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate <= SYSDATE + 30;

**Output :**





**Exercise 3: Stored Procedures :**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings'

) LOOP

UPDATE Accounts

SET Balance = acc.Balance \* 1.01,

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

END LOOP;

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

SELECT AccountID, AccountType, Balance, LastModified

FROM Accounts

WHERE AccountType = 'Savings';

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (

SELECT EmployeeID, Salary

FROM Employees

WHERE Department = dept\_name

) LOOP

UPDATE Employees

SET Salary = emp.Salary + (emp.Salary \* bonus\_percent / 100)

WHERE EmployeeID = emp.EmployeeID;

END LOOP;

END;

/

BEGIN

UpdateEmployeeBonus('Support', 10);

END;

/

SELECT EmployeeID, Name, Department, Salary

FROM Employees

WHERE Department = 'Support';

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_acct IN NUMBER,

to\_acct IN NUMBER,

amt IN NUMBER

) IS

from\_balance NUMBER;

BEGIN

-- Get current balance of source account

SELECT Balance INTO from\_balance FROM Accounts WHERE AccountID = from\_acct;

-- Check if sufficient funds

IF from\_balance < amt THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance in source account.');

END IF;

-- Deduct from source

UPDATE Accounts

SET Balance = Balance - amt,

LastModified = SYSDATE

WHERE AccountID = from\_acct;

-- Add to target

UPDATE Accounts

SET Balance = Balance + amt,

LastModified = SYSDATE

WHERE AccountID = to\_acct;

END;

/

BEGIN

TransferFunds(1001, 1002, 5000);

END;

/

SELECT AccountID, CustomerID, AccountType, Balance, LastModified

FROM Accounts

WHERE AccountID IN (1001, 1002);

**Output :**

