**Week2\_PL/SQL programming**

**Exercise 1: Control Structures**

-- Step 1: Create Tables

CREATE TABLE CUSTOMERS (

Cust\_ID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Age NUMBER,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE LOANS (

Loan\_ID NUMBER PRIMARY KEY,

Cust\_ID NUMBER REFERENCES CUSTOMERS(Cust\_ID),

InterestRate NUMBER(5,2),

Due\_Date DATE

);

-- Step 2: Insert Data

INSERT INTO CUSTOMERS VALUES (1, 'John', 65, 12000, 'FALSE');

INSERT INTO CUSTOMERS VALUES (2, 'Sara', 45, 8000, 'FALSE');

INSERT INTO CUSTOMERS VALUES (3, 'Mike', 70, 15000, 'FALSE');

INSERT INTO LOANS VALUES (101, 1, 10.50, SYSDATE + 15);

INSERT INTO LOANS VALUES (102, 2, 12.00, SYSDATE + 40);

INSERT INTO LOANS VALUES (103, 3, 11.00, SYSDATE + 20);

COMMIT;

-- Scenario 1: Apply 1% interest discount for age > 60

BEGIN

FOR loan\_rec IN (

SELECT l.Loan\_ID

FROM LOANS l

JOIN CUSTOMERS c ON l.Cust\_ID = c.Cust\_ID

WHERE c.Age > 60

)

LOOP

UPDATE LOANS

SET InterestRate = InterestRate - 1

WHERE Loan\_ID = loan\_rec.Loan\_ID;

END LOOP;

COMMIT;

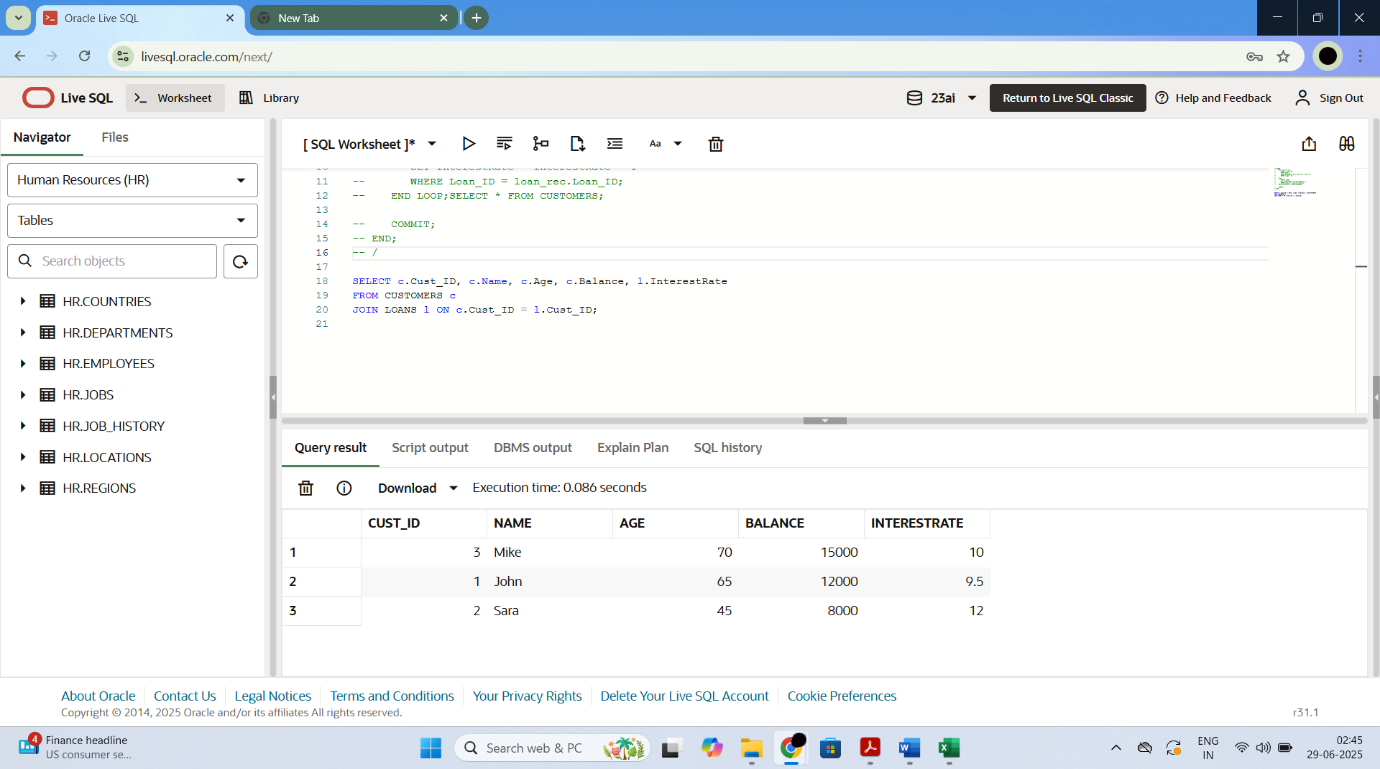
END;

/

SELECT c.Cust\_ID, c.Name, c.Age, c.Balance, l.InterestRate

FROM CUSTOMERS c

JOIN LOANS l ON c.Cust\_ID = l.Cust\_ID;



-- Scenario 2: Promote to VIP if balance > 10000

BEGIN

FOR cust\_rec IN (

SELECT Cust\_ID

FROM CUSTOMERS

WHERE Balance > 10000 AND (IsVIP IS NULL OR IsVIP = 'FALSE')

)

LOOP

UPDATE CUSTOMERS

SET IsVIP = 'TRUE'

WHERE Cust\_ID = cust\_rec.Cust\_ID;

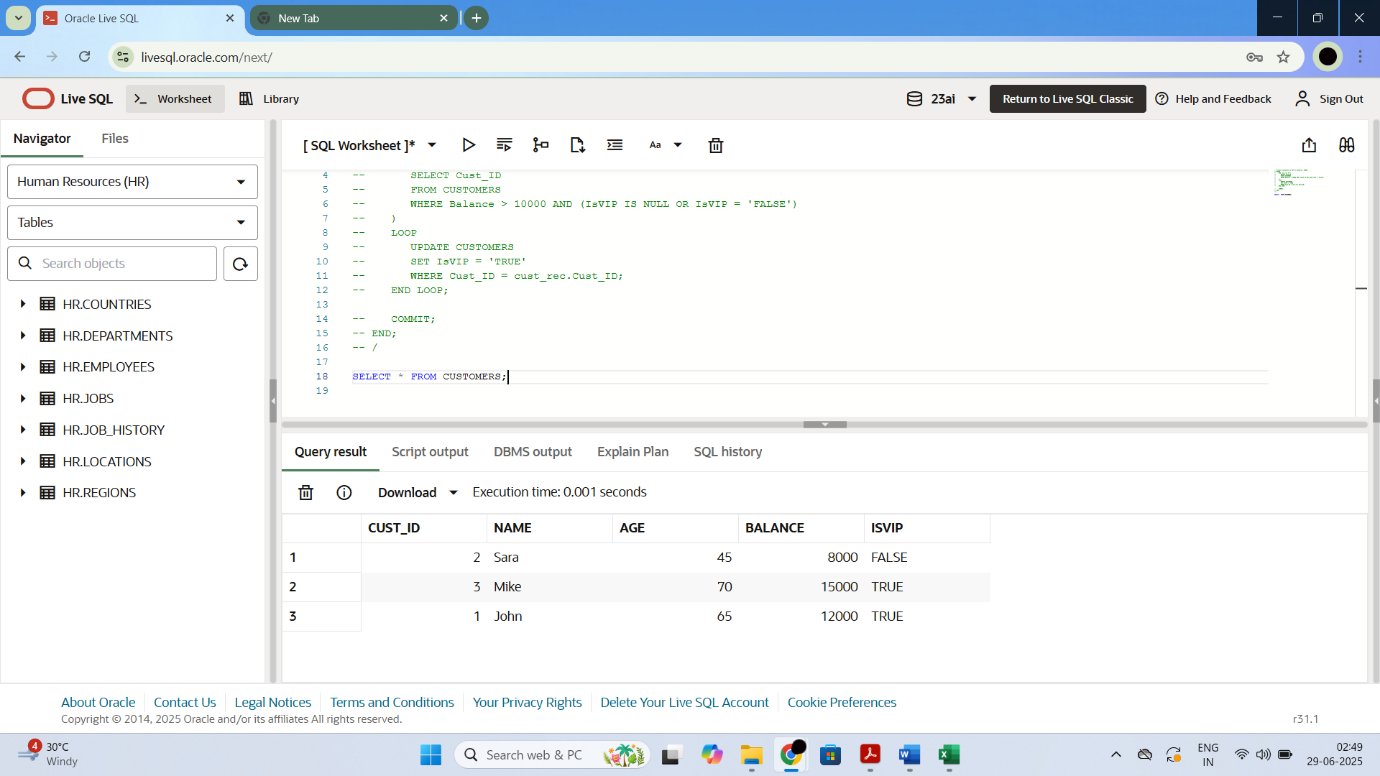
END LOOP;

COMMIT;

END;

/

SELECT \* FROM CUSTOMERS;



-- Scenario 3: Print reminders for loans due in 30 days

BEGIN

FOR loan\_rec IN (

SELECT c.Name, l.Due\_Date

FROM LOANS l

JOIN CUSTOMERS c ON l.Cust\_ID = c.Cust\_ID

WHERE l.Due\_Date <= SYSDATE + 30

)

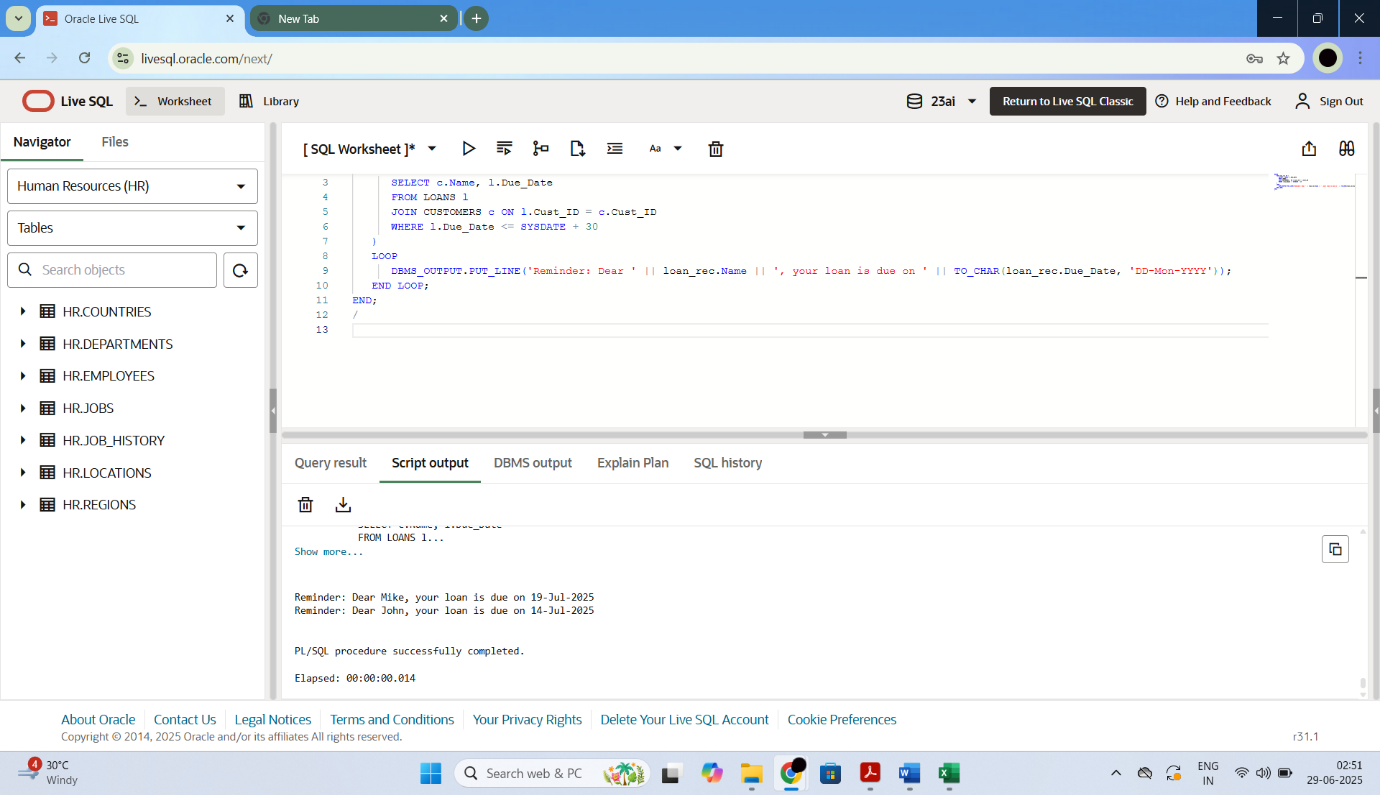
LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || loan\_rec.Name || ', your loan is due on ' || TO\_CHAR(loan\_rec.Due\_Date, 'DD-Mon-YYYY'));

END LOOP;

END;

/



**Exercise 3: Stored Procedures**

-- Creating tables

CREATE TABLE ACCOUNTS (

Account\_ID NUMBER PRIMARY KEY,

Cust\_ID NUMBER,

Balance NUMBER,

Account\_Type VARCHAR2(20)

);

CREATE TABLE EMPLOYEES (

Emp\_ID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER

);

-- Inserting sample data

INSERT INTO ACCOUNTS VALUES (1, 101, 10000, 'Savings');

INSERT INTO ACCOUNTS VALUES (2, 102, 8000, 'Savings');

INSERT INTO ACCOUNTS VALUES (3, 103, 12000, 'Current');

INSERT INTO ACCOUNTS VALUES (4, 104, 5000, 'Savings');

INSERT INTO EMPLOYEES VALUES (1, 'Anil', 'HR', 30000);

INSERT INTO EMPLOYEES VALUES (2, 'Ravi', 'IT', 40000);

INSERT INTO EMPLOYEES VALUES (3, 'Sneha', 'HR', 32000);

INSERT INTO EMPLOYEES VALUES (4, 'Divya', 'Sales', 28000');

COMMIT;

-- Scenario 1:Procedure to apply 1% monthly interest to all savings accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE ACCOUNTS

SET Balance = Balance + (Balance \* 0.01)

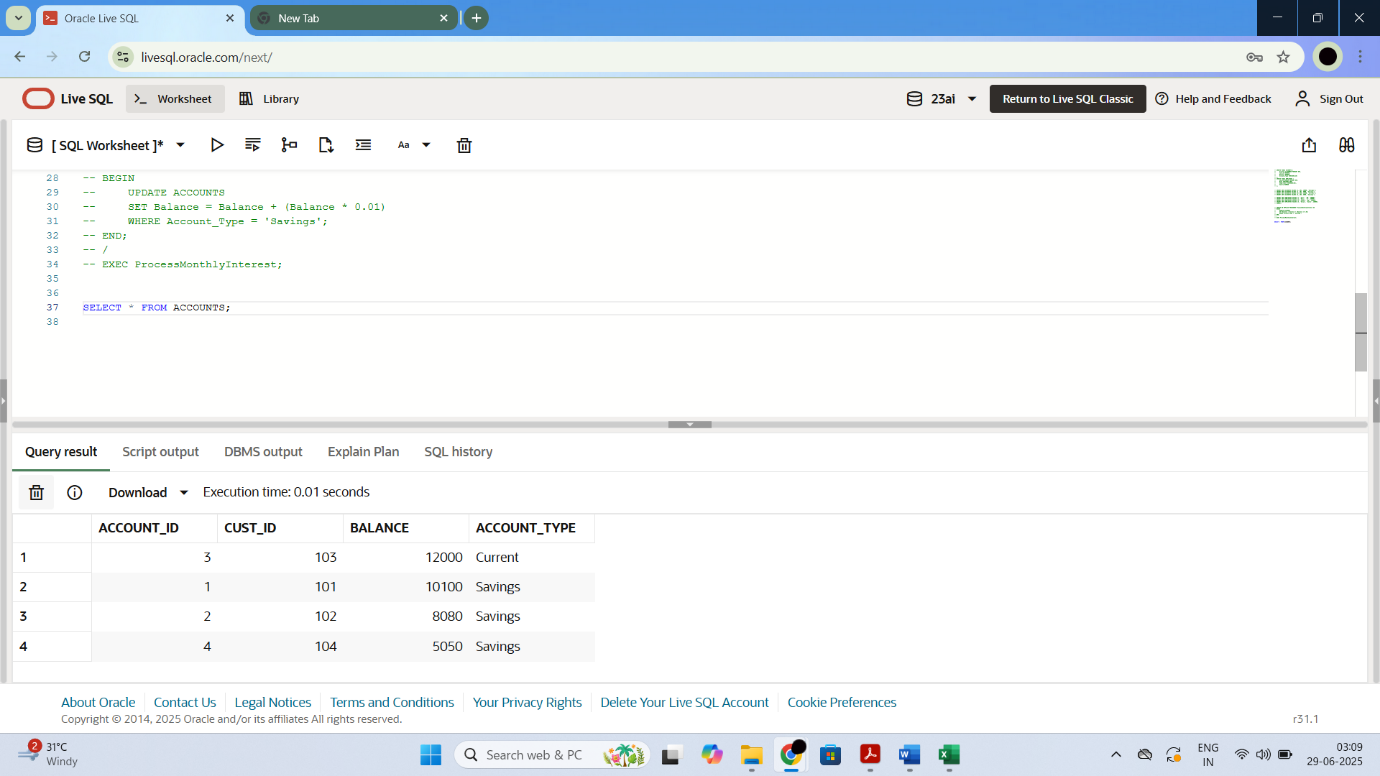
WHERE Account\_Type = 'Savings';

END;

/

EXEC ProcessMonthlyInterest;

SELECT \* FROM ACCOUNTS;



-- Scenario 2:Procedure to apply bonus to employees of a specific department

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept IN VARCHAR2,

bonus\_percent IN NUMBER

) AS

BEGIN

UPDATE EMPLOYEES

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

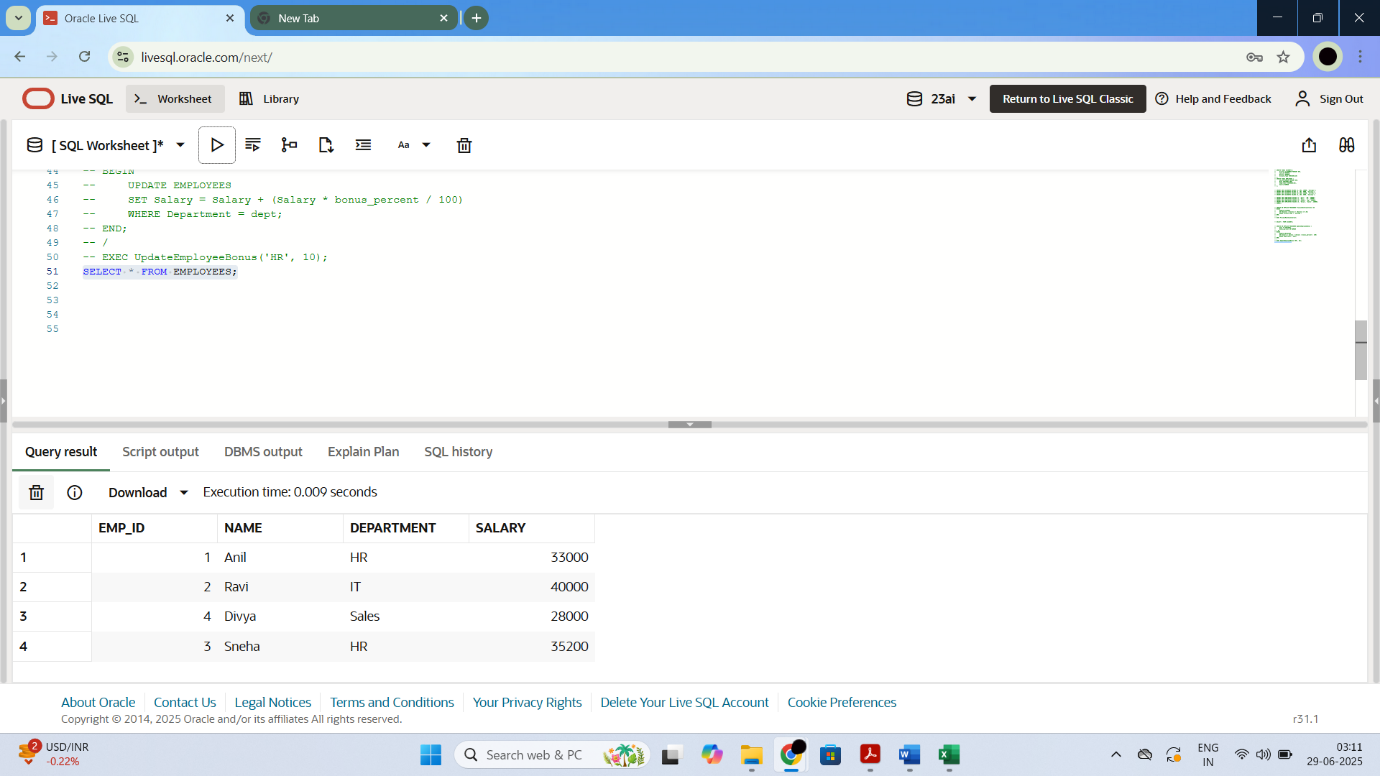
WHERE Department = dept;

END;

/

EXEC UpdateEmployeeBonus('HR', 10);

SELECT \* FROM EMPLOYEES;



-- Scenario 3:Procedure to transfer funds from one account to another

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_acct IN NUMBER,

to\_acct IN NUMBER,

amount IN NUMBER

) AS

insufficient\_balance EXCEPTION;

from\_balance ACCOUNTS.Balance%TYPE;

BEGIN

SELECT Balance INTO from\_balance

FROM ACCOUNTS

WHERE Account\_ID = from\_acct;

IF from\_balance < amount THEN

RAISE insufficient\_balance;

ELSE

UPDATE ACCOUNTS

SET Balance = Balance - amount

WHERE Account\_ID = from\_acct;

UPDATE ACCOUNTS

SET Balance = Balance + amount

WHERE Account\_ID = to\_acct;

END IF;

EXCEPTION

WHEN insufficient\_balance THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance.');

END;

/

EXEC TransferFunds(1, 3, 2000);

SELECT \* FROM ACCOUNTS;

