Exercise 1 : Control Structures

Scenario 1 -

BEGIN

  UPDATE customers

  SET interest\_rate = interest\_rate - 1

  WHERE age > 60;

COMMIT;

END;

Scenario 2 -

BEGIN

  UPDATE customers

  SET isVIP = 'TRUE'

  WHERE balance > 10000;

  COMMIT;

END;

Scenario 3 –

BEGIN

  FOR loan IN (

    SELECT customer\_id, loan\_due\_date

    FROM loans

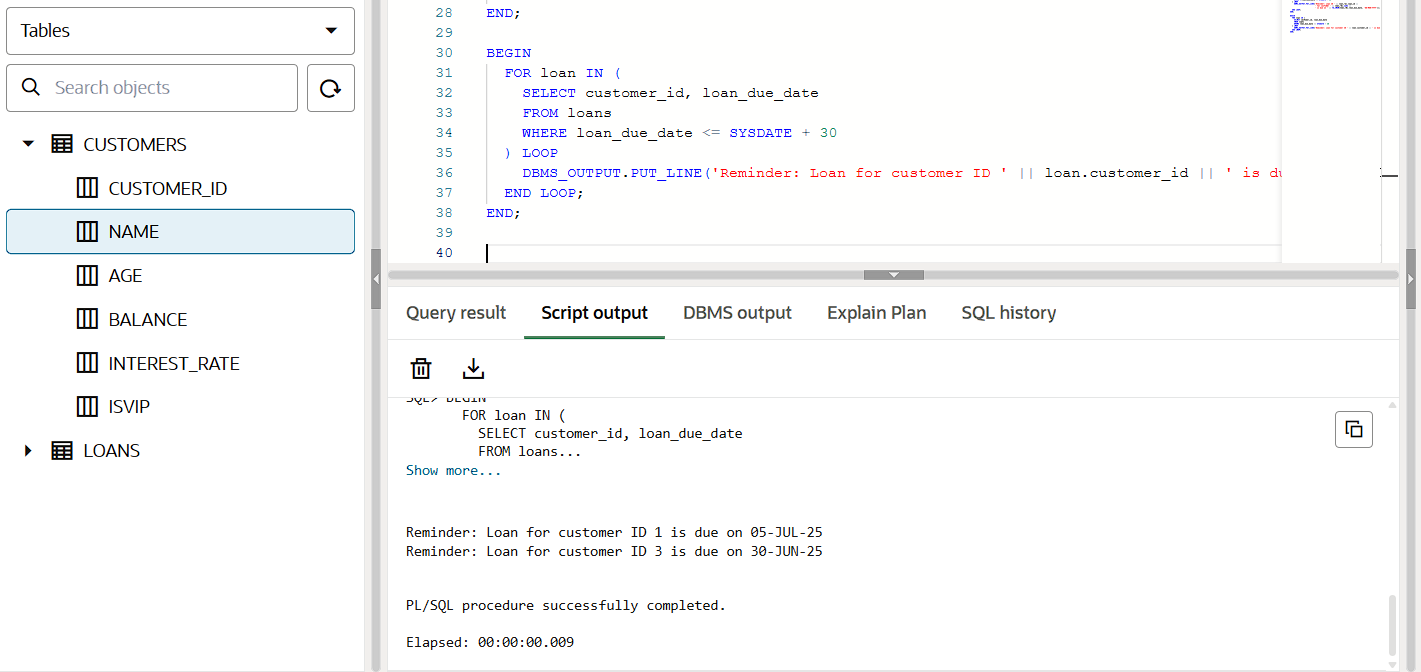
    WHERE loan\_due\_date <= SYSDATE + 30

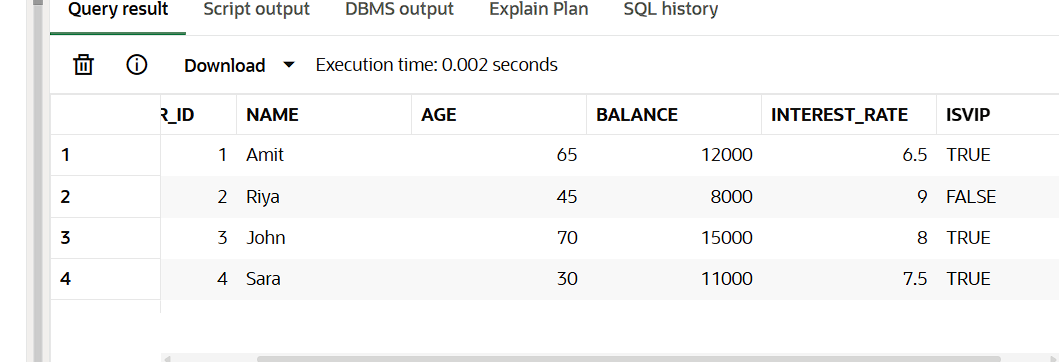
  ) LOOP

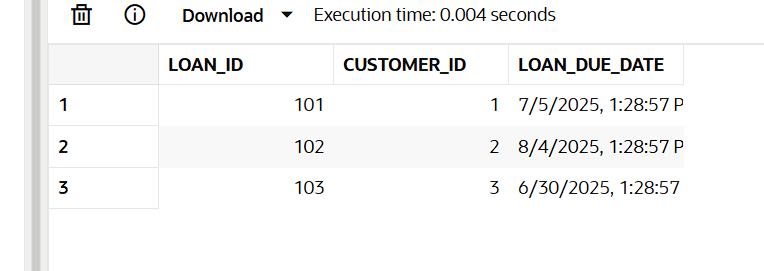
    DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for customer ID ' || loan.customer\_id || ' is due on ' || loan.loan\_due\_date);

  END LOOP;

END;







Exercise – 3 : Stored Procedures

Scenario – 1

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

  UPDATE accounts

  SET balance = balance + (balance \* 0.01)

  WHERE account\_type = 'SAVINGS';

  COMMIT;

END;

Scenario – 2

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

  dept\_id IN NUMBER,

  bonus\_percent IN NUMBER

) IS

BEGIN

  UPDATE employees

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

  WHERE department\_id = dept\_id;

  COMMIT;

END;

//By adding 10% bonus as parameter..

BEGIN

  UpdateEmployeeBonus(10, 10);

END;

Scenario – 3

CREATE OR REPLACE PROCEDURE TransferFunds (

  from\_acc IN NUMBER,

  to\_acc IN NUMBER,

  amount IN NUMBER

) IS

  from\_balance NUMBER;

BEGIN

  SELECT balance INTO from\_balance FROM accounts WHERE account\_id = from\_acc;

IF from\_balance >= amount THEN

    UPDATE accounts

    SET balance = balance - amount

    WHERE account\_id = from\_acc;

    UPDATE accounts

    SET balance = balance + amount

    WHERE account\_id = to\_acc;

COMMIT;

  ELSE

    DBMS\_OUTPUT.PUT\_LINE('Insufficient balance!!');

  END IF;

END;

BEGIN

  TransferFunds(103, 104, 2000);

END;

