WEEK 2 - PL/SQL Programming

Exercise 1: Control Structures

TABLE CREATION:

CUTOMERS

```
CREATE TABLE customers (
    customer_id     NUMBER PRIMARY KEY,
    name            VARCHAR2(100),
    age             NUMBER,
    balance         NUMBER(10,2),
    is_vip          VARCHAR2(5)
);
```

LOANS

```
CREATE TABLE loans (
    loan_id         NUMBER PRIMARY KEY,
    customer_id     NUMBER,
    interest_rate   NUMBER(5,2),
    due_date        DATE,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

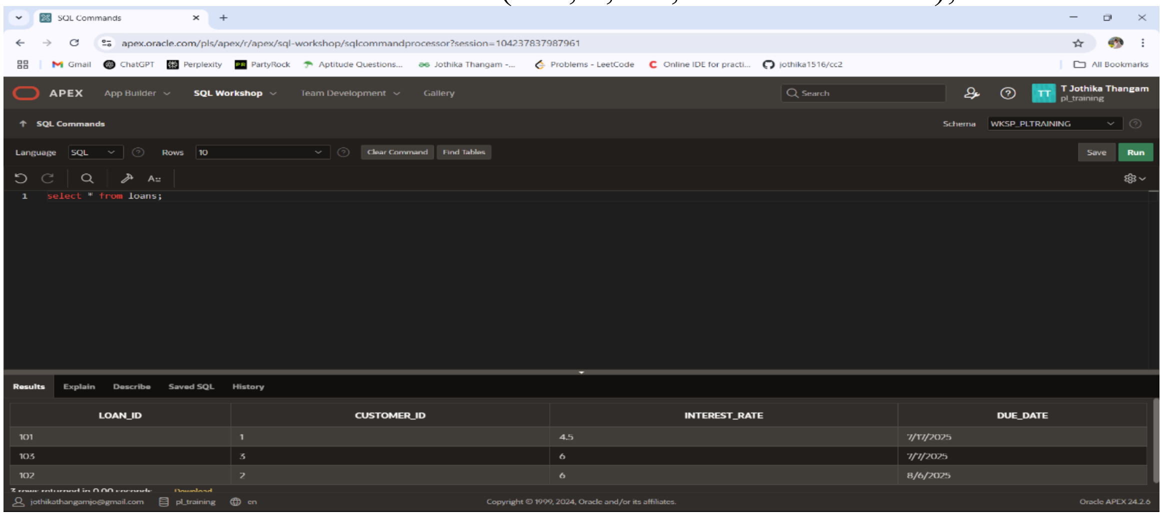
DATA INSERTION:

CUTOMERS

```
INSERT INTO Customers VALUES (1, 'Alice', 65, 12000, 'FALSE');
INSERT INTO Customers VALUES (2, 'Bob', 58, 8000, 'FALSE');
INSERT INTO Customers VALUES (3, 'Charlie', 70, 20000, 'FALSE');
```

LOANS

```
INSERT INTO Loans VALUES (101, 1, 5.5, SYSDATE + 20);
INSERT INTO Loans VALUES (102, 2, 6.0, SYSDATE + 40);
INSERT INTO Loans VALUES (103, 3, 7.0, SYSDATE + 10);
```



Schema: WKSP_PLTRAINING
Language: SQL
Rows: 10

```
select * from loans;
```

LOAN_ID	CUSTOMER_ID	INTEREST_RATE	DUE_DATE
101	1	4.5	7/17/2025
103	3	6	7/7/2025
102	2	6	8/6/2025

3 rows returned in 0.00 seconds

jothikathangamjo@gmail.com pl_training en

Copyright © 1999, 2024, Oracle and/or its affiliates.

Oracle APEX 24.2.6

SCENARIO 1:

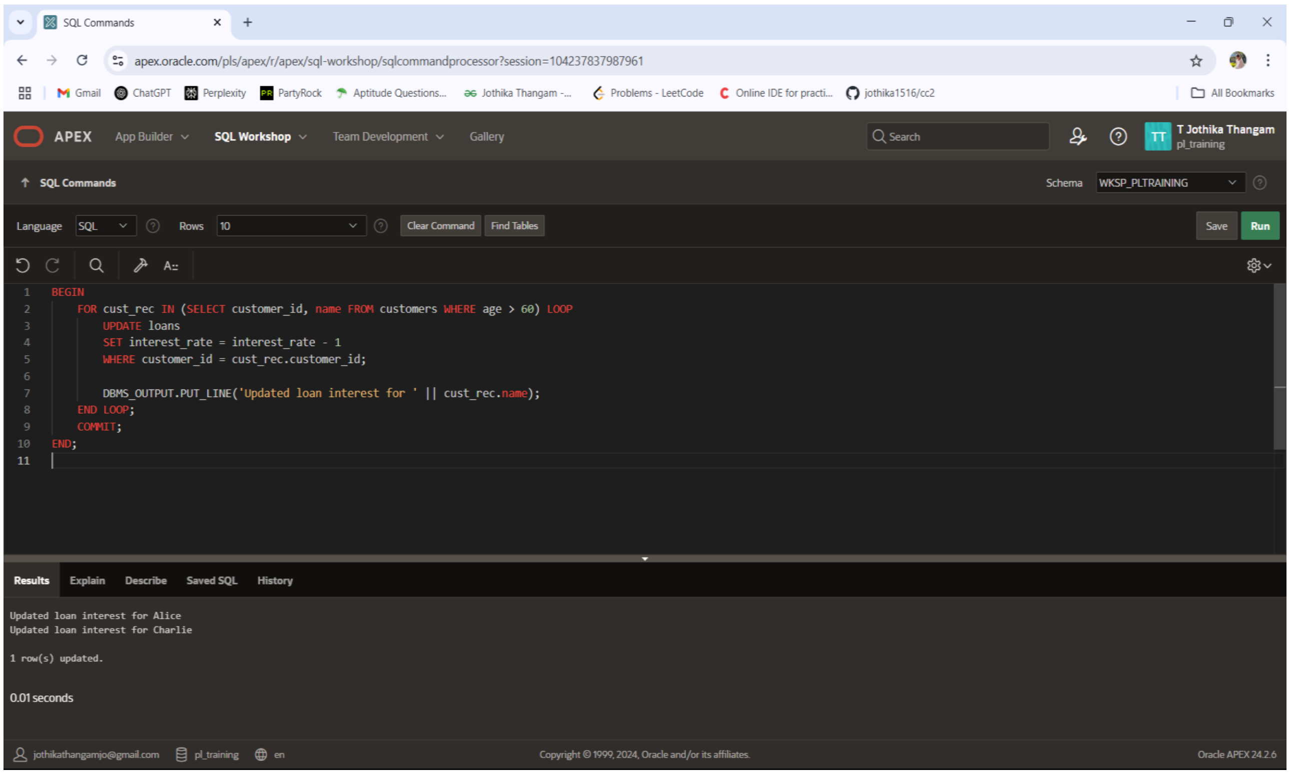
The bank wants to apply a discount to loan interest rates for customers above 60 years old.

- 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.

CODE:

```
BEGIN

FOR cust_rec IN (SELECT customer_id, name FROM customers WHERE age
> 60) LOOP

        UPDATE loans

        SET interest_rate = interest_rate - 1

        WHERE customer_id = cust_rec.customer_id;

        DBMS_OUTPUT.PUT_LINE('Updated loan interest for ' ||cust_rec.name);

    END LOOP;

    COMMIT;

END;
```

OUTPUT:



```
BEGIN
    FOR cust_rec IN (SELECT customer_id, name FROM customers WHERE age > 60) LOOP
        UPDATE loans
        SET interest_rate = interest_rate - 1
        WHERE customer_id = cust_rec.customer_id;

        DBMS_OUTPUT.PUT_LINE('Updated loan interest for ' || cust_rec.name);
    END LOOP;
    COMMIT;
END;
```

```
Updated loan interest for Alice
Updated loan interest for Charlie

1 row(s) updated.

0.01 seconds
```

SCENARIO 2:

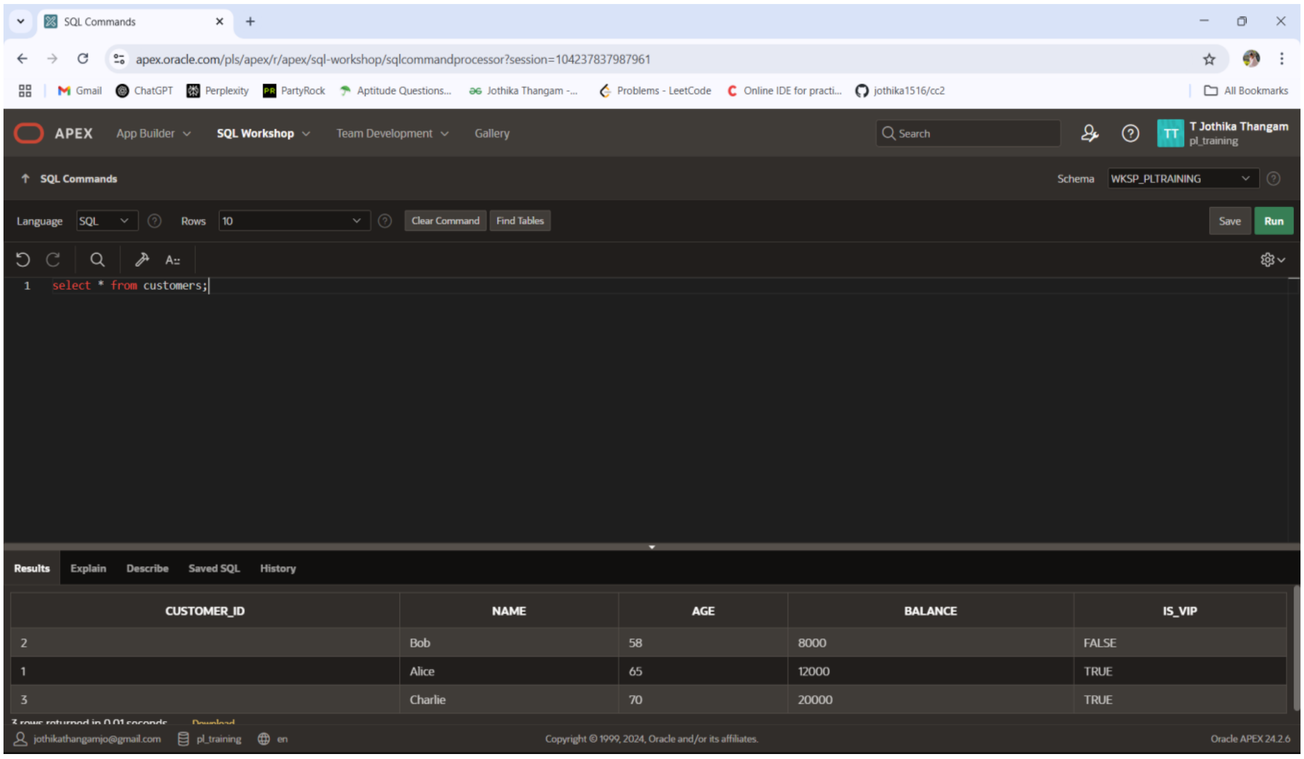
A customer can be promoted to VIP status based on their balance.

- **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.

CODE:

```
BEGIN
  FOR rec IN (SELECT CustomerID FROM Customers WHERE Balance >
10000)
  LOOP
    UPDATE Customers
    SET IsVIP = 'TRUE' -- if it's a VARCHAR2 column; if BOOLEAN, use
TRUE
    WHERE CustomerID = rec.CustomerID;
  END LOOP;

  COMMIT;
END;
```

OUTPUT:



```
select * from customers;
```

CUSTOMER_ID	NAME	AGE	BALANCE	IS_VIP
2	Bob	58	8000	FALSE
1	Alice	65	12000	TRUE
3	Charlie	70	20000	TRUE

SCENARIO 3:

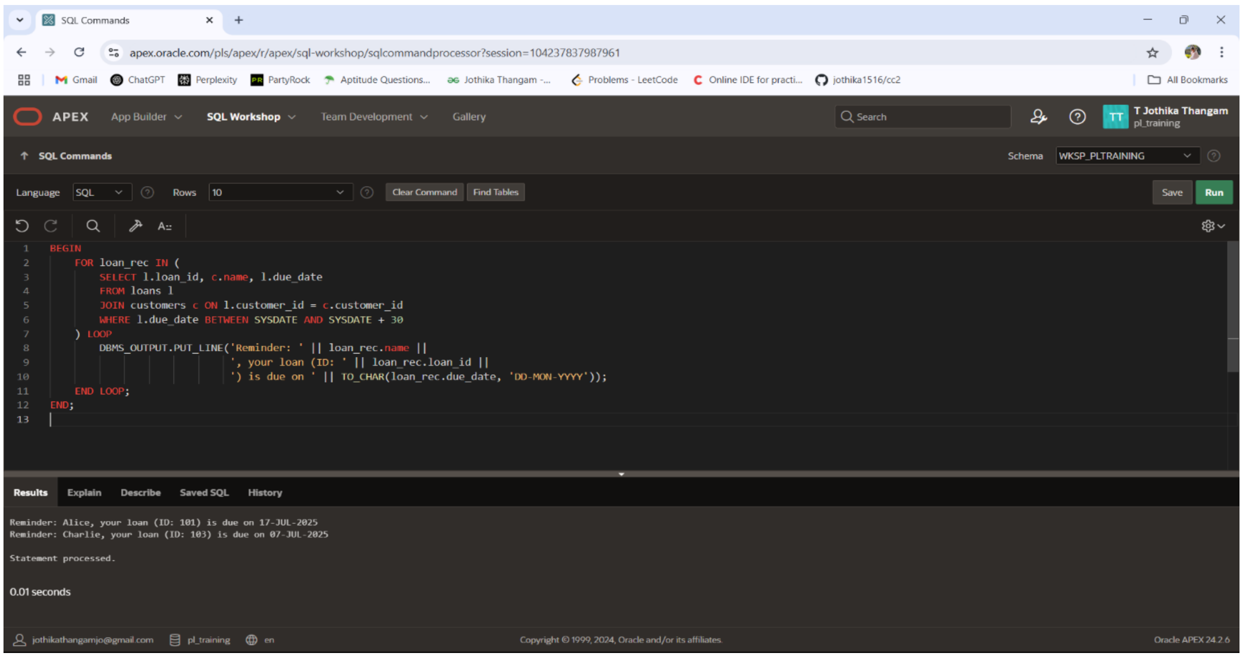
The bank wants to send reminders to customers whose loans are due within the next 30 days.

- Question: Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

CODE:

```
BEGIN
   FOR loan_rec IN (
      SELECT l.loan_id, c.name, l.due_date
      FROM loans l
      JOIN customers c ON l.customer_id = c.customer_id
      WHERE l.due_date BETWEEN SYSDATE AND SYSDATE + 30
   ) LOOP
      DBMS_OUTPUT.PUT_LINE('Reminder: ' || loan_rec.name ||
                           ', your loan (ID: ' || loan_rec.loan_id ||
                           ') is due on ' || TO_CHAR(loan_rec.due_date, 'DD-MON-
YYYY'));
   END LOOP;
END;
```

OUTPUT:



```
BEGIN
    FOR loan_rec IN (
        SELECT l.loan_id, c.name, l.due_date
        FROM loans l
        JOIN customers c ON l.customer_id = c.customer_id
        WHERE l.due_date BETWEEN SYSDATE AND SYSDATE + 30
    ) LOOP
        DBMS_OUTPUT.PUT_LINE('Reminder: ' || loan_rec.name ||
                             ', your loan (ID: ' || loan_rec.loan_id ||
                             ') is due on ' || TO_CHAR(loan_rec.due_date, 'DD-MON-YYYY'));
    END LOOP;
END;
```

```
Reminder: Alice, your loan (ID: 101) is due on 17-JUL-2025
Reminder: Charlie, your loan (ID: 103) is due on 07-JUL-2025

Statement processed.

0.01 seconds
```