

WEEK 2 - PL/SQL PROGRAMMING

EXERCISE 1:CONTROL STRUCTURES

TABLE CREATION:

CUSTOMERS

```
CREATE TABLE Customers (  
    CustomerID NUMBER PRIMARY KEY,  
    Name VARCHAR2(100),  
    Age NUMBER,  
    Balance NUMBER,  
    IsVIP VARCHAR2(5)  
);
```

LOANS

```
CREATE TABLE Loans (  
    LoanID NUMBER PRIMARY KEY,  
    CustomerID NUMBER,  
    InterestRate NUMBER,  
    DueDate DATE,  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);
```

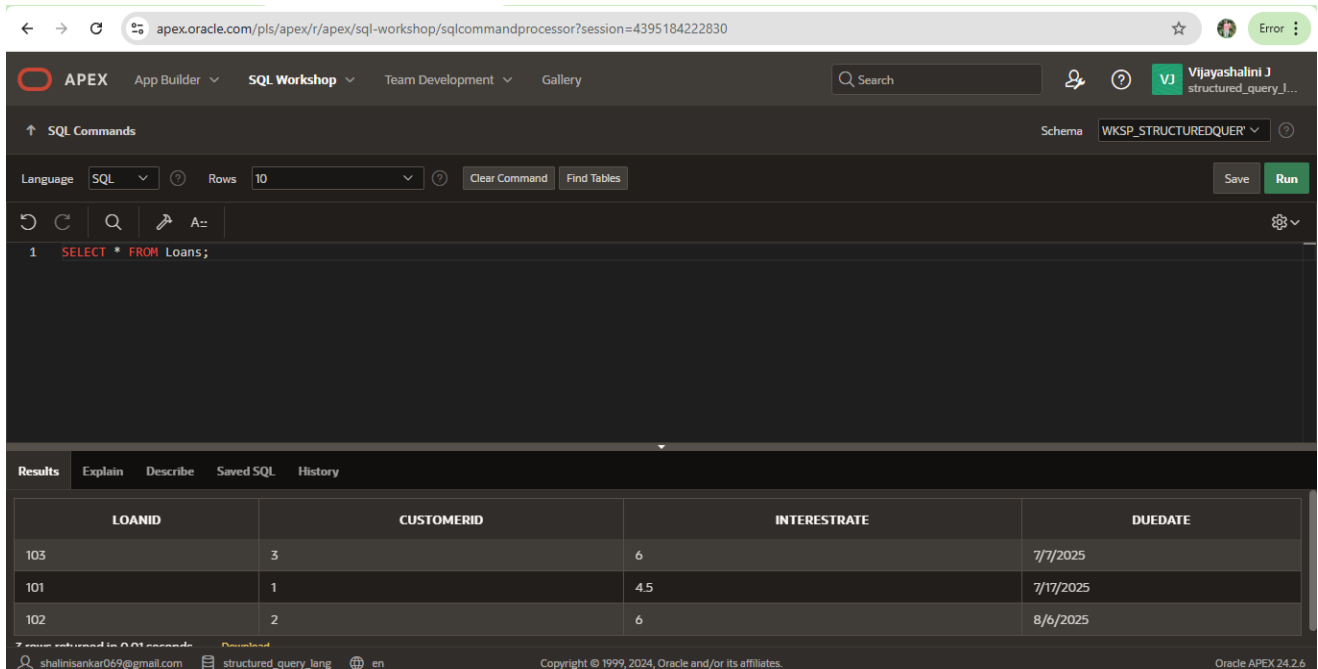
DATA INSERTION

CUSTOMER:

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



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The main area displays the 'SQL Commands' tab with a query editor containing the command: `1 SELECT * FROM Loans;`. Below the editor, the 'Results' tab is active, showing a table with the following data:

LOANID	CUSTOMERID	INTERESTRATE	DUE DATE
103	3	6	7/7/2025
101	1	4.5	7/17/2025
102	2	6	8/6/2025

The bottom of the interface shows the user 'shalinisankar069@gmail.com' and the version '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

CustomerID

```
WHERE c.Age > 60 BEGIN
```

```
FOR rec IN (
```

```
SELECT c.CustomerID, l.LoanID, l.InterestRate
```

```

FROM Customers c

JOIN Loans l ON c.CustomerID =

)

LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

END LOOP;

COMMIT;

END;

```

OUTPUT:

The screenshot displays the Oracle APEX SQL Workshop interface. The browser address bar shows the URL: `apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=4395184222830`. The interface includes a top navigation bar with tabs for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile (Vijayashalini J) are also present.

The main area is titled "SQL Commands" and shows the following SQL script:

```

11 WHERE LoanID = rec.LoanID;
12
13 -- Output message with correct variable
14 DBMS_OUTPUT.PUT_LINE('Updated the loan interest for : ' || rec.Name);
15 END LOOP;
16
17 COMMIT;
18 END;
19 /
20

```

Below the script editor, the "Results" tab is active, displaying the output of the execution:

```

Updated the loan interest for : Shalini
Updated the loan interest for : Alice

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.

```
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:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The main area displays the 'SQL Commands' tab with a single command: `select * from customers;`. The 'Results' tab is active, showing a table with the following data:

CUSTOMERID	NAME	AGE	BALANCE	ISVIP
3	Shallini	70	20000	TRUE
2	Bob	58	8000	FALSE
1	Alice	65	12000	TRUE

The bottom of the interface shows the footer with 'Copyright © 1999, 2024, Oracle and/or its affiliates.' and 'Oracle APEX 24.2.6'.

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.

```
BEGIN
```

```
FOR rec IN (
```

```
    SELECT l.LoanID, l.DueDate, c.Name
```

```
    FROM Loans l
```

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

```
    WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30
```

```
)
```

```
LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Reminder : Loan ' || rec.LoanID || ' for ' || rec.Name ||
```

```
        ' is due on ' || TO_CHAR(rec.DueDate, 'DD-MON-YYYY'));
```

```
END LOOP;
```

```
END;
```

OUTPUT

← → ↺ apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=4395184222830 ☆ Error

APEX

App Builder

SQL Workshop

Team Development

Gallery

Search

Vijayashalini J

structured_query_1...

SQL Commands

Schema WKSP_STRUCTUREDQUER

Language SQL

Rows 10

Clear Command

Find Tables

Save

Run

↺ ↻ 🔍 📄 A⇄ ⚙️

1 BEGIN

2 FOR rec IN (

3 SELECT l.LoanID, l.DueDate, c.Name

4 FROM Loans l

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

6 WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

7)

8 LOOP

9 DBMS_OUTPUT.PUT_LINE('Reminder: Loan ' || rec.LoanID || ' for ' || rec.Name ||

10 ' is due on ' || TO_CHAR(rec.DueDate, 'DD-MON-YYYY'));

11 END LOOP;

Results

Explain

Describe

Saved SQL

History

Reminder: Loan 103 for Shalini is due on 07-JUL-2025

Reminder: Loan 101 for Alice is due on 17-JUL-2025

Statement processed.

0.02 seconds

shalinisankar069@gmail.com structured query lang en

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

Oracle APEX 24.2.6