QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES
INFORMATION TECHNOLOGY DEPARTMENT

Week 7
Control Structure - Looping

IM101 - Advanced Database Systems

LEARNING OUTCOMES:

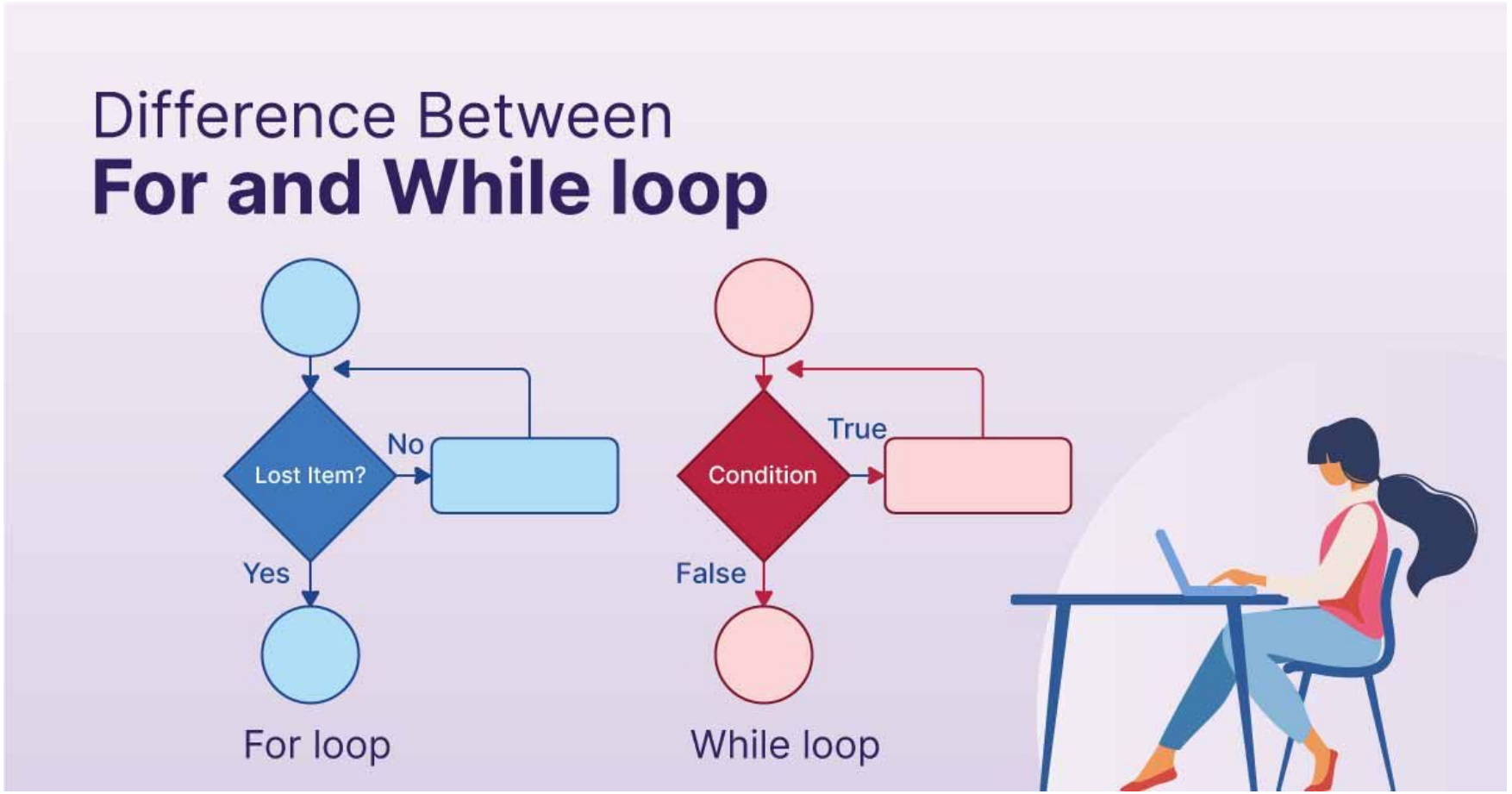
At the end of the session, the students should be able to:

1. Recognize different types of PL/SQL loops.
2. Create PL/SQL blocks containing basic loop, for loop, and while loop, nested loop.

Introduction to Loops in PL/SQL

- Loops are used to execute a sequence of statements multiple times.
- Three main types of loops in PL/SQL:
  - **Basic LOOP**
  - **FOR LOOP**
  - **WHILE LOOP**
- Loops improve efficiency by reducing code repetition.

Looping


Difference Between
For and While loop
Lost Item?
No
Yes
For loop
Condition
True
False
While loop

Basic LOOP

- Executes a sequence of statements repeatedly until explicitly terminated using EXIT.

Syntax

```
BEGIN
    LOOP
        -- Statements
        EXIT WHEN
condition;
    END LOOP;
END;
```

Sample

```
DECLARE
    counter NUMBER := 1;
BEGIN
    LOOP
        DBMS_OUTPUT.PUT_LINE('Iteration: ' ||
counter);
        counter := counter + 1;
        EXIT WHEN counter > 5;
    END LOOP;
END;
```

FOR LOOP

- Iterates over a range of values with a known number of iterations.

Syntax

```
BEGIN
    FOR counter IN start_value .. end_value LOOP
        -- Statements
    END LOOP;
END;
```

FOR LOOP

Sample

```
BEGIN
     FOR i IN 1..5 LOOP
          DBMS_OUTPUT.PUT_LINE('Iteration: ' || i);
     END LOOP;
END;
```

WHILE LOOP

- Executes statements as long as the condition remains TRUE.

Syntax

```
BEGIN
    WHILE condition LOOP
        -- Statements
    END LOOP;
END;
```

WHILE LOOP

Sample

```
DECLARE
    counter NUMBER := 1;
BEGIN
    WHILE counter <= 5 LOOP
        DBMS_OUTPUT.PUT_LINE('Iteration: ' || counter);
        counter := counter + 1;
    END LOOP;
END;
```

Nested Loops

- A loop inside another loop.
- Useful for processing multi-dimensional data structures.

```
BEGIN
    FOR i IN 1..3 LOOP
        FOR j IN 1..2 LOOP
            DBMS_OUTPUT.PUT_LINE('Outer: ' || i || ', Inner: ' || j);
        END LOOP;
    END LOOP;
END;
```

Best Practices for Using Loops

- Avoid infinite loops by ensuring termination conditions are met.
- Use **EXIT WHEN** to control loop execution.
- Optimize loops by minimizing redundant computations inside the loop.
- Consider using bulk operations for large data processing.

Summary

- PL/SQL supports three main types of loops: **Basic LOOP, FOR LOOP, WHILE LOOP**.
- Nested loops allow handling of complex operations.
- Proper loop termination and efficiency considerations are essential.

END OF PRESENTATION.
THANK YOU!