PL/SQL 4-3: Iterative Control: Basic Loops

Agustín Alejandro Mota Hinojosa

November 10, 2023

Contents

1 Vocabulary

Encloses a sequence of statements between the keywords LOOP and END LOOP and must execute at least once.

Basic Loop

Statement to terminate a loop.

EXIT

2 Try it / Solve it

1. What purpose does a loop serve in PL/SQL?

   The purpose of loops in PL/SQL is to repeat the same or similar code a specified number of times or until a certain condition is met.

2. List the types of loops in PL/SQL.

   - Basic Loop
   - FOR Loop
   - WHILE Loop

3. What statement is used to explicitly end a loop?

   The EXIT statement is used to explicitly end a loop. It can be used alone or with a condition (EXIT WHEN condition).

4. Write a PL/SQL block to display the country$_{\text{id}}$ and country$_{\text{name}}$ values from the COUNTRIES table for country$_{\text{id}}$ whose values range from 1 through 3. Use a basic loop. Increment a variable from 1 through 3. Use an IF statement to test your variable and EXIT the loop after you have displayed the first 3 countries.

```
DECLARE
 v_counter NUMBER(1) := 1;
 v_country_name wf_countries.country_name%TYPE;
BEGIN
 LOOP
  SELECT country_name INTO v_country_name
  FROM wf_countries
  WHERE country_id = v_counter;
  DBMS_OUTPUT.PUT_LINE(v_country_name);
  v_counter := v_counter + 1;
  IF v_counter > 3 THEN EXIT;
  END IF;
 END LOOP;
END;
```

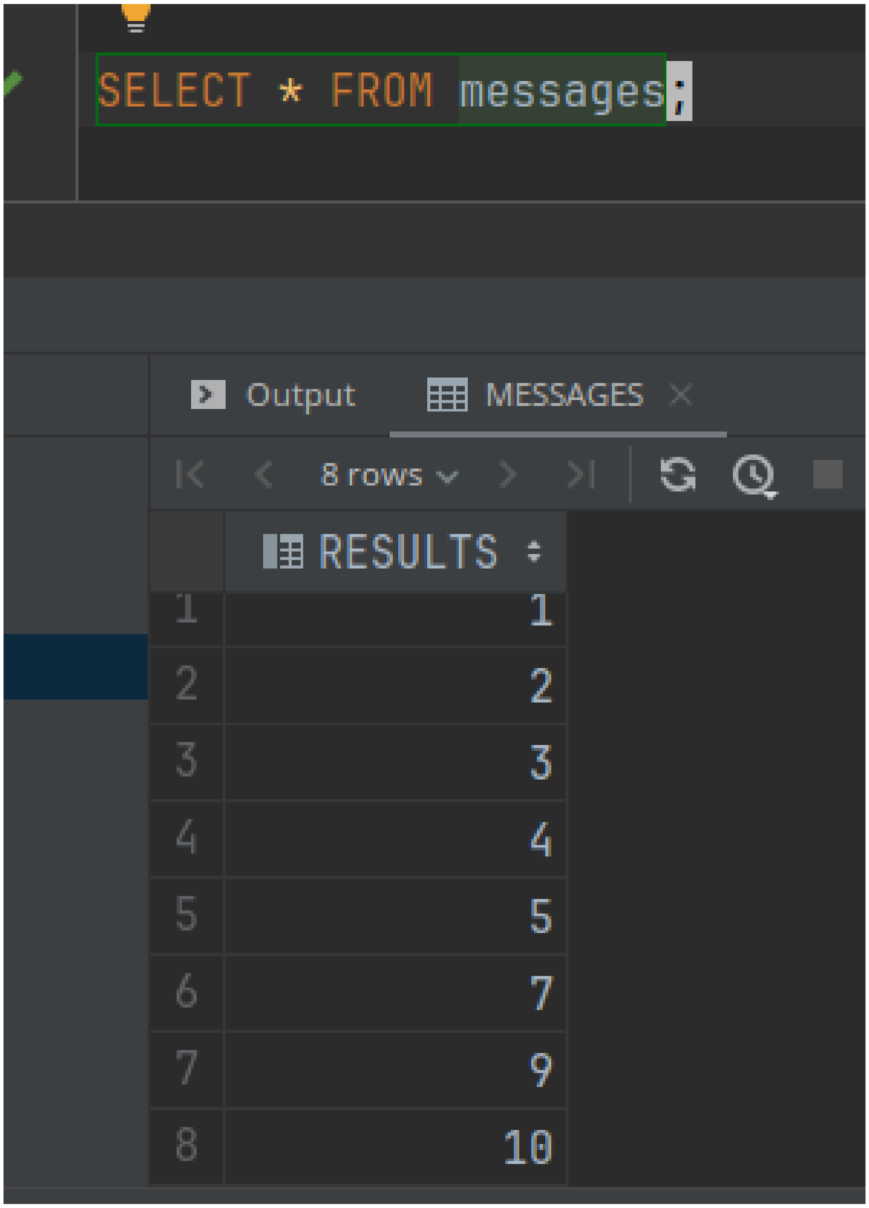
5. Modify your solution to question 4 above, replacing the IF statement with an EXIT. . . .WHEN statement.

```
DECLARE
 v_counter NUMBER(1) := 1;
 v_country_name wf_countries.country_name%TYPE;
BEGIN
 LOOP
  SELECT country_name INTO v_country_name
  FROM wf_countries
  WHERE country_id = v_counter;
  DBMS_OUTPUT.PUT_LINE(v_country_name);
  v_counter := v_counter + 1;
  EXIT WHEN v_counter > 3;
 END LOOP;
END;
```

6. Create a MESSAGES table and insert several rows into it.

```
DROP TABLE messages;
CREATE TABLE messages (results NUMBER(2));

DECLARE
 v_counter NUMBER(2) := 1;
BEGIN
 LOOP
  IF v_counter <> 6 AND v_counter <> 8 THEN
  INSERT INTO messages
  VALUES (v_counter);
  END IF;
  v_counter := v_counter + 1;
  EXIT WHEN v_counter > 10;
 END LOOP;
END;
```



```
SELECT * FROM messages;
```

	RESULTS
1	1
2	2
3	3
4	4
5	5
6	7
7	9
8	10