

Database Programming with PL/SQL

4-3: Iterative Control: Basic Loops

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

Basic loop	Encloses a sequence of statements between the keywords LOOP and END LOOP and must execute at least once.
WHEN EXIT	Statement to terminate a loop.

Try It / Solve It

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

Repetir cierto numero de veces una o varias sentencias

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

LOOP

WHILE

FOR

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

EXIT WHEN

4. Write a PL/SQL block to display the country_id and country_name values from the COUNTRIES table for country_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_country_name COUNTRIES.COUNTRY_name%TYPE;

v_counter NUMBER:=1;

BEGIN

loop

if v_counter <= 3 then

select country_name INTO v_country_name FROM countries WHERE country_id = v_counter;

DBMS_OUTPUT.PUT_LINE(v_country_NAME);

v_counter:= v_counter+1;

END IF;

IF v_counter > 3 then EXIT;

END IF;

end loop;

END;

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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

DECLARE

v_country_name COUNTRIES.COUNTRY_name%TYPE;

v_counter NUMBER:=1;

BEGIN

loop

select country_name INTO v_country_name FROM countries WHERE country_id = v_counter;

```
DBMS_OUTPUT.PUT_LINE(v_country_NAME);  
v_counter:= v_counter+1;  
  
EXIT WHEN v_counter > 3;  
END IF;  
end loop;  
END;
```

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

A. To create the messages table.




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

B. Write a PL/SQL block to insert numbers into the MESSAGES table. Insert the numbers 1 through 10, excluding 6 and 8.

```
DECLARE  
v_counter NUMBER:=1;  
BEGIN  
LOOP  
    INSERT INTO MESSAGES VALUES(v_counter);  
    v_counter:=v_counter+1;  
    IF v_counter = 6 or v_counter = 8 then  
        v_counter:=v_counter+1;  
    END IF;  
    EXIT WHEN v_counter = 11;  
END LOOP;  
END;
```

C. Execute a SELECT statement to verify that your PL/SQL block worked.

SELECT * FROM MESSAGES;

Results	Explain	Describe	Saved SQL	History
RESULTS				
1				
2				
3				
4				
5				
7				
9				
10				
8 rows returned in 0.00 seconds Download				
<div> mx_a104_sql_s39  mx_a104_sql_s39  en</div> <div>Copyright © 1999, 2015, Oracle. All rights reserved. Application Express 5.0.3.00.03</div>				