**Quiz**

There are three types of loops: basic, FOR, and WHILE.

1. True  
2. False

**Answer: True**

**Practice 5: Writing Control Structures**

**In this practice, you create PL/SQL blocks that incorporate loops and conditional control structures. This practice tests your understanding of various IS statements and LOOP constructs.**

1)  Execute the command in the lab\_05\_01.sql file to create the messages table. Write a PL/SQL block to insert numbers into the messages table.

**CREATE TABLE messages (**

**results VARCHAR2(50)**

**);**

**Изображение выглядит как текст

Автоматически созданное описание**

a)  Insert the numbers 1 through 10, excluding 6 and 8.

**BEGIN**

**FOR i IN 1..10 LOOP**

**IF i = 6 OR i = 8 THEN**

**CONTINUE;**

**ELSE**

**INSERT INTO messages(results)**

**VALUES(i);**

**END IF;**

**END LOOP;**

**END;**

**/**

b)  Commit before the end of the block.

**BEGIN**

**FOR i IN 1..10 LOOP**

**IF i = 6 OR i = 8 THEN**

**CONTINUE;**

**ELSE**

**INSERT INTO messages(results)**

**VALUES(i);**

**END IF;**

**END LOOP;**

**COMMIT;**

**END;**

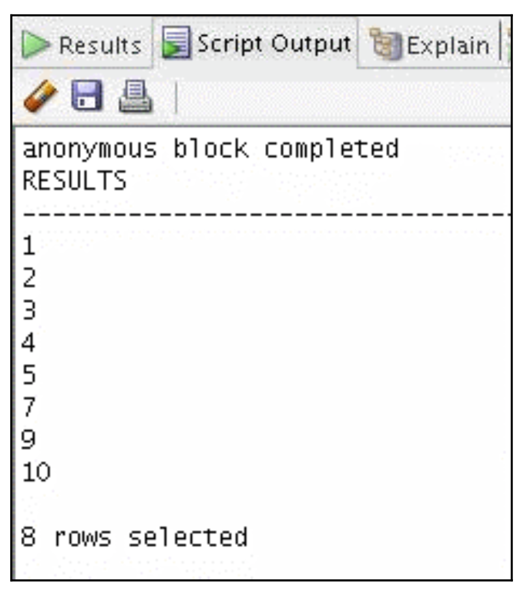
**/**

**Изображение выглядит как текст

Автоматически созданное описание**

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

Result: You should see the following output:



Изображение выглядит как текст

Автоматически созданное описание

2)  Execute the lab\_05\_02.sql script. This script creates an emp table that is a replica of the employees table. It alters the emp table to add a new column, stars, of VARCHAR2 data type and size 50. Create a PL/SQL block that inserts an asterisk in the starts column for every $1000 of an employee’s salary. Save your script as lab\_05\_02\_soln.sql.

**CREATE TABLE emp AS SELECT\*FROM employees;**

**ALTER TABLE emp ADD stars VARCHAR2(50);**

Изображение выглядит как текст

Автоматически созданное описание

a)  In the declarative section of the block, declare a variable v\_empno of type emp.employee\_id and initialize it to 176. Declare a variable v\_asterisk of type emp.stars and initialize it to NULL. Create a variable v\_sal of type emp.salary.

**DECLARE**

**v\_empno emp.employee\_id%TYPE := 176;**

**v\_asterisk emp.stars%TYPE := NULL;**

**v\_sal emp.salary%TYPE;**

**Изображение выглядит как текст

Автоматически созданное описание**

b)  In the executable section, write logic to append an asterisk (\*) to the string for every $1,000 of the salary. For example, if the employee earns $8,000, the string of asterisks should contain eight asterisks. If the employee earns $12,500, the string of asterisks should contain 13 asterisks.

**BEGIN**

**SELECT salary**

**INTO v\_sal**

**FROM emp WHERE employee\_id = v\_empno;**

**FOR i IN 1..v\_sal/1000 LOOP**

**v\_asterisk := v\_asterisk || '\*';**

**END LOOP;**

Изображение выглядит как текст

Автоматически созданное описание

c) Update the stars column for the employee with the string of asterisks. Commit before the end of the block.

**UPDATE emp SET stars = v\_asterisk**

**WHERE employee\_id = v\_empno;**

**COMMIT;**

**END;**

Изображение выглядит как текст

Автоматически созданное описание

d)  Display the row from the emp table to verify whether your PL/SQL block has executed successfully.

**SELECT employee\_id, salary, stars**

**FROM emp**

**WHERE employee\_id = 176;**

e)  Execute and save your script as lab\_05\_02\_soln.sql. The output is as follows:

Изображение выглядит как стол

Автоматически созданное описание

