

Database Programming with PL/SQL

4-4: Iterative Control: WHILE and FOR Loops

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

	Repeats a sequence of statements until the controlling condition is no longer TRUE.
	Repeats a sequence of statements until a set number of iterations have been completed.

Try It / Solve It

1. 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 51 through 55. Use a WHILE loop. Increment a variable from 51 through 55. Test your variable to see when it reaches 55. EXIT the loop after you have displayed the 5 countries.
2. 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 51 through 55 *in the reverse order*. Use a FOR loop.
3. Execute the following statements to build a new_emps table.

```
DROP TABLE new_emps;
```

```
CREATE TABLE new_emps AS SELECT * FROM employees;
```

```
ALTER TABLE new_emps ADD stars VARCHAR2(50);
```

- A. Create a PL/SQL block that inserts an asterisk in the stars column for every whole \$1,000 of an employee's salary. For example, if an employee has salary of \$7,800, the string "*****" would be inserted, and, if an employee has salary of \$3,100, the string "***" would be inserted. Use the following code as a starting point.

```

DECLARE
    v_empno                new_emps.employee_id%TYPE := <employee_id>;
    v_asterisk              new_emps.stars%TYPE := NULL;
    v_sal_in_thousands     new_emps.salary%TYPE;
BEGIN
    SELECT NVL(TRUNC(salary/1000), 0) INTO v_sal_in_thousands
    FROM new_emps WHERE employee_id = v_empno;
    FOR ...
    ...
    UPDATE new_emps
        SET stars = v_asterisk
        WHERE employee_id = v_empno;
END;
```

- B. Test your code using employee_ids 124 and 142, then confirm the results.