

Nama : Afina Putri Dayanti
NIM : 825200049
Jurusan : Sistem Informasi
Mata Kuliah : Database Design and Management (Praktikum)

Vocabulary

Identify the vocabulary word for each definition below:

WHILE Loops	Repeats a sequence of statements until the controlling condition is no longer TRUE.
FOR Loops	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.

Answer :

```
declare
  v_name    wf_countries.country_name%type;
  v_country_id wf_countries.country_id%type := 51;
begin
  select country_name, country_id into v_name, v_country_id
  from wf_countries
  where country_id = v_country_id;
  while v_country_id <= 55 loop
    dbms_output.put_line('country name : ' || v_name || ', country id : ' || v_country_id);
    v_country_id := v_country_id + 1;
  end loop;
end;
```

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.

Answer :

```
declare
  v_name    wf_countries.country_name%type;
  v_country_id wf_countries.country_id%type := 51;
begin
  for i in reverse v_country_id..55 loop
    select country_name, country_id into v_name, v_country_id
    from wf_countries
    where country_id = i;
    dbms_output.put_line('country name : ' || v_name || ', country id : ' || v_country_id);
  end loop;
end;
```

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;
```

Answer :

```
declare
```

```
    v_empno          new_emps.employee_id%type :=&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;
```

```
    dbms_output.put('salary id ' || v_empno || ' : ');
```

```
    for i in 1..v_sal_in_thousands loop
```

```
        dbms_output.put('*');
```

```
    end loop;
```

```
    dbms_output.new_line;
```

```
    update new_emps
```

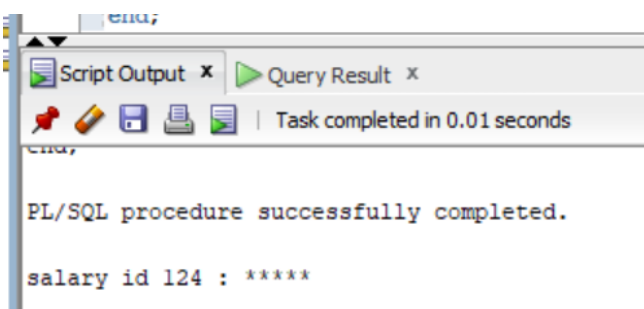
```
    set stars = v_asterisk
```

```
    where employee_id = v_empno;
```

```
end;
```

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

Answer :



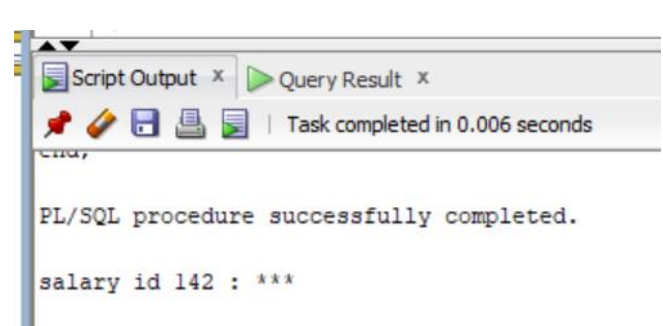
```
end;
```

Script Output x Query Result x

Task completed in 0.01 seconds

PL/SQL procedure successfully completed.

salary id 124 : *****



```
end;
```

Script Output x Query Result x

Task completed in 0.006 seconds

PL/SQL procedure successfully completed.

salary id 142 : ***