Database Programming with PL/SQL

3-2: Retrieving Data in PL/SQL

Practice Activities

Embedded queries tre sa returneze EXACT un rand. Select...Into...From...

Vocabulary

No new vocabulary for this lesson

Try It / Solve It

1. State whether each of the following SQL statements can be included directly in a PL/SQL block.

|  |  |  |
| --- | --- | --- |
| **Statement** | Valid in PL/SQL | Invalid in PL/SQL |
| ALTER USER SET password = 'oracle'; | nein(DCL statement e invalid In PL/SQL)\* | ja |
| CREATE TABLE test (a NUMBER); | nein\* | ja |
| DROP TABLE test; | nein\* | ja |
| SELECT emp\_id INTO v\_id FROM employees; | ja | nein |
| GRANT SELECT ON employees TO PUBLIC; | nein\* | ja |
| INSERT INTO grocery\_items (product\_id, brand,  description) VALUES (199, 'Coke', 'Soda'); | ja | nein |
| REVOKE UPDATE ON employees FROM PUBLIC; | nein\* | ja |
| ALTER TABLE employees  RENAME COLUMN employee\_id TO emp\_id; | nein\* | ja |
| DELETE FROM grocery\_items  WHERE description = 'Soap'; | ja | nein |

2. Create a PL/SQL block that selects the maximum department\_id in the departments table and

stores it in the v\_max\_deptno variable. Display the maximum department\_id. Declare

v\_max\_deptno to be the same datatype as the department\_id column. Include a SELECT

statement to retrieve the highest department\_id from the departments table. Display the variable

v\_max\_deptno.

DECLARE

v\_max\_depno departments.department\_id%TYPE;

BEGIN

SELECT MAX(department\_id)

INTO v\_max\_depno

FROM departments;

DBMS\_OUTPUT.PUT\_LINE(v\_max\_depno);

END;

3. The following code is supposed to display the lowest and highest elevations for a country name

entered by the user. However, the code does not work. Fix the code by following the guidelines for

retrieving data that you learned in this lesson.

DECLARE

v\_country\_name

countries.country\_name%TYPE := Federative Republic of Brazil;

v\_lowest\_elevation countries.lowest\_elevation%TYPE;

v\_highest\_elevation countries.highest\_elevation%TYPE;

BEGIN

SELECT lowest\_elevation, highest\_elevation

FROM countries;

DBMS\_OUTPUT.PUT\_LINE('The lowest elevation in '

|| v\_country\_name || ' is ' || v\_lowest\_elevation

|| ' and the highest elevation is ' || v\_highest\_elevation || '.');

END;

DECLARE

v\_country\_name countries.country\_name%TYPE := 'Federative Republic of Brazil';

v\_lowest\_elevation countries.lowest\_elevation%TYPE;

v\_highest\_elevation countries.highest\_elevation%TYPE;

BEGIN

SELECT lowest\_elevation, highest\_elevation

INTO v\_lowest\_elevation, v\_highest\_elevation

FROM countries

WHERE country\_name = v\_country\_name;

DBMS\_OUTPUT.PUT\_LINE('The lowest elevation in '

|| v\_country\_name || ' is ' || v\_lowest\_elevation

|| ' and the highest elevation is ' || v\_highest\_elevation || '.');

END;

4. Run the following anonymous block. It should execute successfully.

**DECLARE**

**v\_emp\_lname employees.last\_name%TYPE;**

**v\_emp\_salary employees.salary%TYPE;**

**BEGIN**

**SELECT last\_name, salary INTO v\_emp\_lname, v\_emp\_salary**

**FROM employees**

**WHERE job\_id = 'AD\_PRES';**

**DBMS\_OUTPUT.PUT\_LINE(v\_emp\_lname || ' ' || v\_emp\_salary);**

**END;  
King 24000**

A. Now modify the block to use ‘IT\_PROG’ instead of ‘AD\_PRES’ and re-run it. Why does it fail

this time?

Select-ul returneaza mai mult de o linie

B. Now modify the block to use ‘IT\_PRAG’ instead of ‘IT\_PROG’ and re-run it. Why does it still

fail?

Nu gaseste date

5. Use (but don't execute) the following code to answer this question:

**DECLARE**

**last\_name VARCHAR2(25) := 'Fay';**

**BEGIN**

**UPDATE emp\_dup**

**SET first\_name = 'Jennifer'**

**WHERE last\_name = last\_name;**

**END;**

What do you think would happen if you ran the above code? Write your answer here and then

follow the steps below to test your theory.

Ar trebui sa updateze first name la Jennifer daca gaseste pe cineva cu last name Fay (asa zicea in curs)

A. Create a table called emp\_dup that is a duplicate of employees.

create table emp\_dup as

(select \* from employees)

B. Select the first\_name and last\_name values for all rows in emp\_dup.

select first\_name, last\_name from employees

C. Run the anonymous PLSQL block shown at the beginning of this question.

gata

D. Select the first\_name and last\_name columns from emp\_dup again to confirm your theory.

select first\_name, last\_name from employees

E. Now we are going to correct the code so that it changes only the first name for the employee

whose last name is “Fay”. Drop emp\_dup and re-create it.

Drop table emp\_dup

F. Modify the code shown at the beginning of this question so that for the employee whose

last\_name = ”Fay”, the first\_name is updated to Jennifer. Run your modified block.

G. Confirm that your update statement worked correctly.

DECLARE

last\_name VARCHAR2(25) := 'Fay';

BEGIN

UPDATE emp\_dup

SET first\_name = 'Jennifer'

WHERE last\_name = ‘Fay’;

END;

6. Is it possible to name a column in a table the same name as the table? Create a table to test this

question. Don't forget to populate the table with data.

Create table dummyt

(coloana NUMBER)

Insert into dummyt values (30)

alter table dummyt

change column coloana to dummyt

R: nu se poate. Invalid alter table option

7. Is it possible to have a column, table, and variable, all with the same name? Using the table you

created in the question above, write a PL/SQL block to test your theory.

Aflam la lab!