 



Database Programming with PL/SQL 3-2: Retrieving Data in PL/SQL Practice Activities

# 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** | **Not Valid in PL/SQL** |
| ALTER USER SET password = 'oracle'; |  | X |
| CREATE TABLE test (a NUMBER); |  | X |
| DROP TABLE test; |  | X |
| SELECT emp\_id INTO v\_id FROM employees; | X |  |
| GRANT SELECT ON employees TO PUBLIC; |  | X |
| INSERT INTO grocery\_items (product\_id, brand, description) VALUES (199, 'Coke', 'Soda'); | X |  |
| REVOKE UPDATE ON employees FROM PUBLIC; |  | X |
| ALTER TABLE employees  RENAME COLUMN employee\_id TO emp\_id; |  | X |
| DELETE FROM grocery\_items WHERE description = 'Soap'; |  |  |

1. 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\_deptno departments.department\_id%TYPE;

BEGIN

SELECT MAX(department\_id)

INTO v\_max\_deptno

FROM departments;

DBMS\_OUTPUT.PUT\_LINE('Id max:' || v\_max\_deptno);

END;

1. 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;

1. 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;

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

Pentru ca interogarea genereaza mai mult de o linie.

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

Dacă modificăm și scriem IT\_PRAG atunci ni se returnează ca nu au fost gasite date, no data found.

1. 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.

Se va genera o eroare pt ca incercam sa executam update-ul, iar tabelul respectiv nu exista.

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

CREATE table emp\_dup

AS (SELECT\* FROM employees);

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

SELECT first\_name, last\_name

FROM emp\_dup;

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

DECLARE

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

BEGIN

UPDATE emp\_dup

SET first\_name = 'Jennifer' WHERE last\_name = last\_name;

END;

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

SELECT first\_name, last\_name

FROM emp\_dup;

* 1. 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;

CREATE TABLE emp\_dup

AS (SELECT\* FROM employees);

* 1. 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.

BEGIN

UPDATE emp\_dup

SET first\_name = 'Jennifer'

WHERE last\_name = 'Fay';

END;

* 1. Confirm that your update statement worked correctly.

Yes, it worked correctly.

1. 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 money

(prog\_id NUMBER(6, 0) NOT NULL,

money NUMBER(8, 2) DEFAULT 0);

INSERT INTO money(prog\_id, money)

VALUES(1, 100);

INSERT INTO money(prog\_id)

VALUES(2);

INSERT INTO money(prog\_id, money)

VALUES(3, 7000);

1. 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.

DECLARE

money NUMBER(8,2);

BEGIN

UPDATE money

SET money = 1000

WHERE money = 7000;

END;