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';
	x	

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_deptno departments.department_id%type;
BEGIN
    SELECT MAX(department_id) INTO v_max_deptno FROM departments;
    DBMS_OUTPUT.PUT_LINE('Maximum department_id ' || v_max_deptno);
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 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;
```

A. Now modify the block to use ‘IT_PROG’ instead of ‘AD_PRES’ and re-run it. Why does it fail this time?

Porque la consulta devuelve mas de una fila, por eso no se puede almacenar en las variables.

B. Now modify the block to use ‘IT_PRAG’ instead of ‘IT_PROG’ and re-run it. Why does it still fail?

No encontró ningún empleado con un job_id llamado ‘IT_PRAG’

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.

El código tiene ambigüedad, el identificador de la variable y la columna tienen el mismo nombre.

Al ejecutarlo cambiara los last_name de todas las filas por Fay

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 emp_dup;

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

   El Código no funciona por la ambigüedad de la variable y columna.

D. Select the first_name and last_name columns from emp_dup again to confirm your theory.

   SELECT first_name, last_name FROM emp_dup;

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.

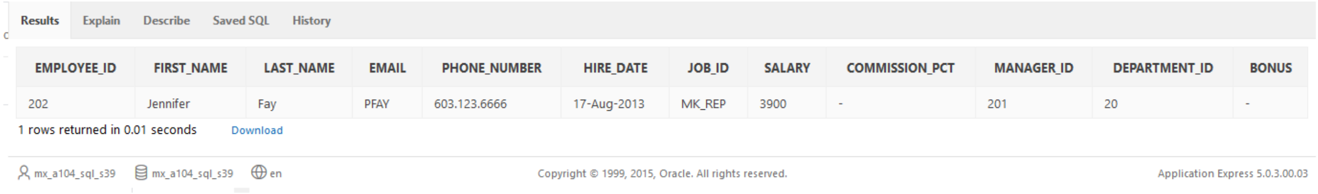
DROP TABLE emp_dup;

CREATE TABLE emp_dup AS SELECT * FROM EMPLOYEES;

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.

```
 DECLARE
   v_last_name VARCHAR2(25) := 'Fay';
BEGIN
   UPDATE emp_dup SET first_name = 'Jennifer'
     WHERE last_name = v_last_name;
END;
```

G. Confirm that your update statement worked correctly.


Results
Explain
Describe
Saved SQL
History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	BONUS
202	Jennifer	Fay	PFAY	603.123.6666	17-Aug-2013	MK_REP	3900	-	201	20	-

1 rows returned in 0.01 seconds   Download

mx_a104_sql_s39
mx_a104_sql_s39
en
Copyright © 1999, 2015, Oracle. All rights reserved.
Application Express 5.0.3.00.03

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.

Si, es posible tener una columna con el mismo nombre que la tabla.

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.

Si, es posible, pero no es recomendado porque al momento de que otra persona lea tu código, le será más difícil comprenderlo.

```
CREATE TABLE d_employee_id(
d_employee_id VARCHAR2(25)
);

INSERT INTO d_employee_id(d_employee_id) VALUES('23')

DECLARE

d_employee_id VARCHAR2(25);
BEGIN
 select d_employee_id INTO d_employee_id FROM d_employee_id;
END;
```