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

Vocabulary

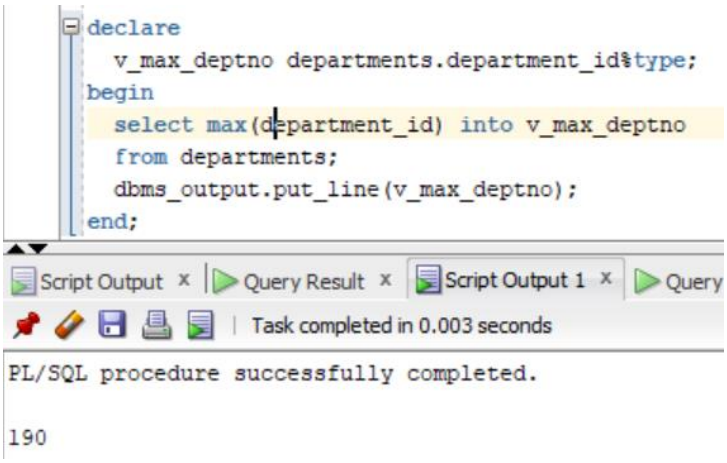
No new vocabulary for this lesson:

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';	✓	
CREATE TABLE test (a NUMBER);	✓	
DROP TABLE test;	✓	
SELECT emp_id INTO v_id FROM employees;		✓
GRANT SELECT ON employees TO PUBLIC;	✓	
INSERT INTO grocery_items (product_id, brand, description) VALUES (199, 'Coke', 'Soda');	✓	
REVOKE UPDATE ON employees FROM PUBLIC;		✓
ALTER TABLE employees RENAME COLUMN employee_id TO emp_id;	✓	
DELETE FROM grocery_items WHERE description = 'Soap';	✓	

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.



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    wf_countries.country_name%TYPE := 'Federative Republic of Brazil';
  v_lowest_elevation  wf_countries.lowest_elevation%TYPE;
  v_highest_elevation wf_countries.highest_elevation%TYPE;
BEGIN
  SELECT lowest_elevation, highest_elevation
  INTO v_lowest_elevation, v_highest_elevation
  FROM wf_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;
```

Script Output | Query Result

Task completed in 0.002 seconds

```
PL/SQL procedure successfully completed.

The lowest elevation in Federative Republic of Brazil is 0 and the highest elevation is 3014.
```

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?
      Answer : because "exact fetch returns more than requested number of rows"

   B. Now modify the block to use ‘IT_PRAG’ instead of ‘IT_PROG’ and re-run it. Why does it still fail?
      Answer : because "no data found"

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;
```

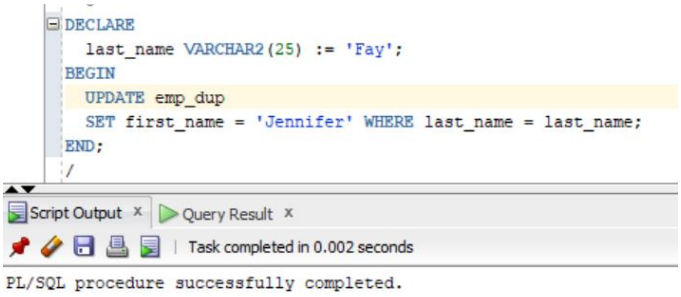
What do you think would happen if you ran the above code? Write your answer here and then follow the steps below to test your theory.

Answer : error, because table emp_dup doesn't exist

A. Create a table called emp_dup that is a duplicate of employees.
   Answer : create table emp_dup as (select * from employees);

B. Select the first_name and last_name values for all rows in emp_dup.
   Answer : select first_name, last_name from emp_dup;

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



D. Select the first_name and last_name columns from emp_dup again to confirm your theory.
   Answer : success displays the data, because in step 1 table emp_dup has been created

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.
   Answer : 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.
   Answer :

```
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.
Answer :

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

Script Output | Task completed in 0.006 seconds

```
PL/SQL procedure successfully completed.
```

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.
Answer : yes it's possible

```
create table test_no6(
  test_no6 number(3,0) primary key,
  brand varchar2(20),
  description varchar2(50)
  );
insert into test_no6(test_no6, brand, description) values (110,'Colgate','Toothpaste');
select * from test_no6;
```

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0.006 seconds

	TEST_NO6	BRAND	DESCRIPTION
1	110	Colgate	Toothpaste

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.
Answer : yes it's possible, but it is almost certainly a bad idea to do so.

```
create table test_no7(
  test_no7 number(3,0) primary key,
  brand varchar2(20),
  description varchar2(50)
  );
insert into test_no7(test_no7, brand, description) values (110,'Colgate','Toothpaste');


declare
    test_no7 number(3,0) := 200;
begin
    select test_no7 into test_no7 from test_no7 where test_no7 = test_no7;
    dbms_output.put_line(test_no7);
END;
```

Script Output | Query Result | Task completed in 0.002 seconds

```
PL/SQL procedure successfully completed.

110
```