



## **Práctica Bases de Datos Avanzadas**

### **Consultas Avanzadas**

Breidy Núñez | 2017-5633

Felipe Ramírez | 2018-6104

Luis Franco | 2018-6342

**Prof. Leandro Fondeur**

4/7/2019

I. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les03.ppt:

3.1 Construya un bloque PL/SQL donde declare dos variables numéricas, una caracter, una lógica y una tipo fecha, y asígnele valores durante la declaración. Restrinja la variable caracter para que no permita valores nulos y una de las variables numéricas definela como constante. Muestre por pantalla los valores asignados.

```
SET SERVEROUTPUT ON
--Declare
v_num1 CONSTANT NUMBER := 4;
v_num2 NUMBER:= 12;
v_text1 VARCHAR2(25)NOT NULL := 'HOLA MUNDO';
v_log BOOLEAN := true;
v_fecha Date := '15,ENE,2020';
BEGIN
DBMS_OUTPUT.PUT_LINE('Valor de la fecha: '|| v_fecha||CHR(10)||
                        'Valor Variable Numérica: '|| v_num2||CHR(10)||
                        'Valor Constante: '|| v_num1||CHR(10)||
                        'Valor Variable Caracter: '|| v_text1||CHR(10)||
                        'Valor Variable logica: '|| v_log);
END;
/
```

Salida de Script x

Tarea terminada en 0.081 segundos

Informe de error -

ORA-06550: line 8, column 22:

PLS-00306: wrong number or types of arguments in call to '||'

ORA-06550: line 8, column 1:

PL/SQL: Statement ignored

06550. 00000 - "line %s, column %s:\n%s"

\*Cause: Usually a PL/SQL compilation error.

\*Action:

*Como se puede ver, este bloque provoca un error. Esto es debido a que DBMS\_OUTPUT.PUT\_LINE no está sobrecargada para admitir valores booleanos. Una forma en la que se podría solucionar es convertir el valor BOOLEAN a una cadena. Esto es posible hacerlo, por ejemplo, con un CASE, así:*

```
Hoja de Trabajo  Generador de Consultas
--3.1
SET SERVEROUTPUT ON
DECLARE
    v_num1  CONSTANT NUMBER(8,2) := 4;
    v_num2  NUMBER(7,2) := 12;
    v_char  VARCHAR2(20) NOT NULL := 'HOLA MUNDO';
    v_logic BOOLEAN := true;
    v_date  DATE:= '15-ENE-2020';
BEGIN
    DBMS_OUTPUT.PUT_LINE('v_num1: '||v_num1||CHR(10)||
                          'v_num2: '||v_num2||CHR(10)||
                          'v_char: '||v_char||CHR(10)||
                          'v_logic: '||CASE WHEN v_logic = true THEN 'true' ELSE 'false' END||CHR(10)||
                          'v_date: '||v_date);
END;
```

Salida de Script x

Tarea terminada en 0.172 segundos

v\_num1: 4  
v\_num2: 12  
v\_char: HOLA MUNDO  
v\_logic: true  
v\_date: 15-ENE-20

Procedimiento PL/SQL terminado correctamente.

3.2 Modifique el código del ejercicio 1 y asígnele un valor diferente a la variable numérica que fue declarada como constante en el cuerpo del bloque.

```
Hoja de Trabajo  Generador de Consultas
--3.2
SET SERVEROUTPUT ON
DECLARE
    v_num1  CONSTANT NUMBER(8,2) := 3000;
    v_num2  NUMBER(7,2) := 5000;
    v_char  VARCHAR2(20) NOT NULL := 'Federico';
    v_logic BOOLEAN := true;
    v_date  DATE:= '21-JUN-2019';
BEGIN
    v_num1 := 2000;
    DBMS_OUTPUT.PUT_LINE('v_num1: '||v_num1||CHR(10)||
                          'v_num2: '||v_num2||CHR(10)||
                          'v_char: '||v_char||CHR(10)||
                          'v_logic: '||CASE WHEN v_logic = true THEN 'true' ELSE 'false' END||CHR(10)||
                          'v_date: '||v_date);
END;
```

Salida de Script x


Tarea terminada en 0.078 segundos

Informe de error -  
ORA-06550: line 8, column 9:  
PLS-00363: expression 'V\_NUM1' cannot be used as an assignment target  
ORA-06550: line 8, column 9:  
PL/SQL: Statement ignored  
06550. 00000 - "line %s, column %s:\n%s"  
\*Cause: Usually a PL/SQL compilation error.  
\*Action:

*Esto es un error porque no se puede modificar el valor de una variable ya definida como constante.*

3.3 Construya un bloque PL/SQL donde se busque el nombre y apellido del empleado 100. Declare la variable del apellido usando %TYPE a partir de la columna de la tabla y la variable del nombre usando %TYPE a partir de la variable del apellido. Muestre en pantalla el contenido de las variables.

```
SET SERVEROUTPUT ON
DECLARE
v_apellido employees.last_name%TYPE;
v_nombre v_apellido%TYPE;
BEGIN
SELECT last_name, first_name into v_apellido, v_nombre
from employees where employee_id = 100;
DBMS_OUTPUT.PUT_LINE('Nombre: '||v_nombre||CHR(10)||'Apellido: '||v_apellido);
END;
/
```



Salida de Script x

Tarea terminada en 0.086 segundos

Nombre: Steven  
Apellido: King

Procedimiento PL/SQL terminado correctamente.

3.4 Construya un bloque PL/SQL busque el nombre y apellido del empleado 100 y guarde los valores en variables de enlace. Muestre el contenido de las variables usando el comando PRINT.

```
VARIABLE b_apellido VARCHAR2
VARIABLE b_nombre VARCHAR2
SET AUTOPRINT ON
BEGIN
SELECT last_name, first_name into :b_apellido, :b_nombre
from employees where employee_id = 100;
END;
/
PRINT b_nombre
PRINT b_apellido;
```

Salida de Script x

Tarea terminada en 0.098 segundos

Procedimiento PL/SQL terminado correctamente.

B\_APELLIDO

----

King

B\_NOMBRE

-----

Steven

3.5 Construya un bloque PL/SQL donde se pida el ID del empleado 204 usando una variable de sustitución en la sección declarativa. Guarde los valores del nombre, apellido y salario en variables de enlace. Permita que se impriman automáticamente sin usar el comando PRINT. Fuera del bloque de código construya una instrucción SELECT para listar todos los empleados que tengan el mismo salario del empleado 204; use la variable de enlace.

```
variable b_emp_fname varchar2
variable b_emp_lname varchar2
variable b_emp_salary number
SET AUTOPRINT ON
declare
    v_empid number (6) := &empid;
begin
    select first_name , last_name , salary  into :b_emp_fname, :b_emp_lname, :b_emp_salary
    from employees
    where employee_id = v_empid;
end;
/
select first_name, last_name, salary
from employees
where salary =:b_emp_salary;
```

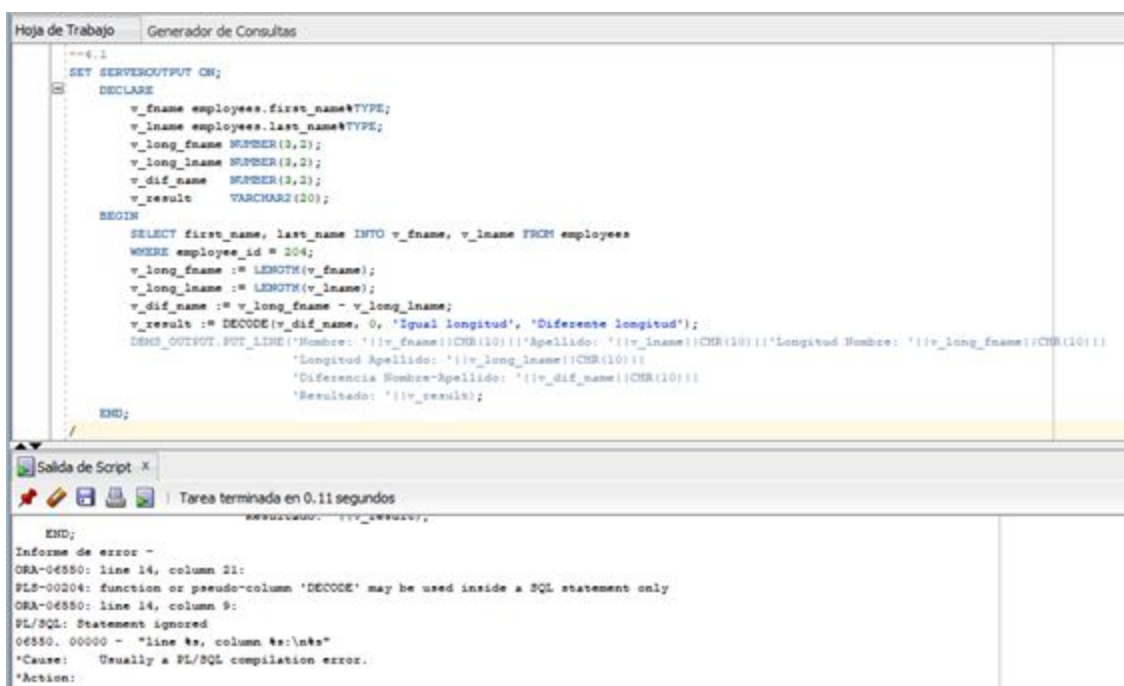
Salida de Script x | Tarea terminada en 4.321 segundos

B\_EMP\_LNAME  
-----  
Baer  
B\_EMP\_SALARY  
-----  
10000

FIRST_NAME	LAST_NAME	SALARY
Peter	Tucker	10000
Janette	King	10000
Harrison	Bloom	10000

II. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les04.ppt:

4.1 Escriba un bloque de código PL/SQL que lea el nombre y apellido del empleado 204. Los valores deben ser guardados en dos variables declaradas con %TYPE basadas en las columnas de la tabla. Halle la longitud del nombre y del apellido y también la diferencia de ambas longitudes. Guarde los valores en variables separadas. Use la función DECODE como una instrucción de PL/SQL para evaluar el resultado de la diferencia de las longitudes del nombre y del apellido. Si el resultado es 0, guardar en una variable el valor "Igual longitud", de lo contrario guarde el valor "Diferente longitud". Muestre en pantalla la longitud del nombre, apellido, la diferencia de las longitudes y el resultado devuelto por el DECODE.



```
--4.1
SET SERVEROUTPUT ON;

DECLARE
    v_fname employees.first_name%TYPE;
    v_lname employees.last_name%TYPE;
    v_long_fname NUMBER(3,2);
    v_long_lname NUMBER(3,2);
    v_dif_name NUMBER(3,2);
    v_result VARCHAR2(20);
BEGIN
    SELECT first_name, last_name INTO v_fname, v_lname FROM employees
    WHERE employee_id = 204;
    v_long_fname := LENGTH(v_fname);
    v_long_lname := LENGTH(v_lname);
    v_dif_name := v_long_fname - v_long_lname;
    v_result := DECODE(v_dif_name, 0, 'Igual longitud', 'Diferente longitud');
    DBMS_OUTPUT.PUT_LINE('Nombre: '||v_fname||CHR(10)||'Apellido: '||v_lname||CHR(10)||'Longitud Nombre: '||v_long_fname||CHR(10)||
    'Longitud Apellido: '||v_long_lname||CHR(10)||
    'Diferencia Nombre-Apellido: '||v_dif_name||CHR(10)||
    'Resultado: '||v_result);
END;
```

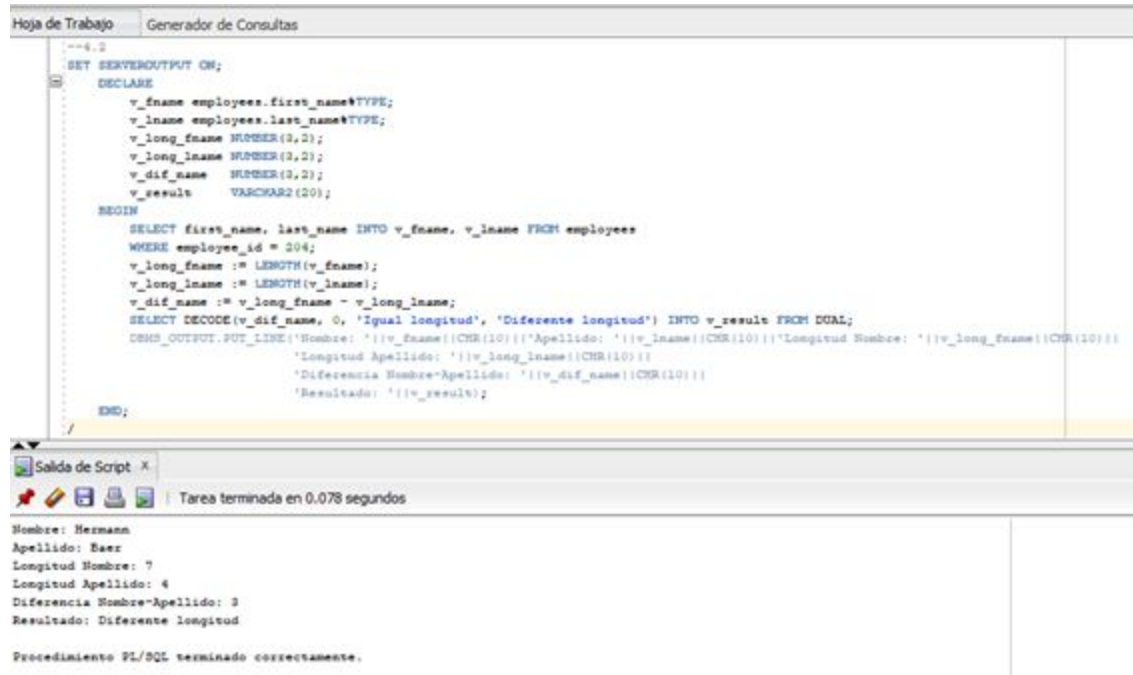
Salida de Script X

Tarea terminada en 0.11 segundos

```
END;
Informe de error -
ORA-06550: line 14, column 21:
PLS-00204: function or pseudo-column 'DECODE' may be used inside a SQL statement only
ORA-06550: line 14, column 9:
PL/SQL: Statement ignored
06550. 00000 - "line %s, column %s:\n%s"
*Cause: Usually a PL/SQL compilation error.
*Action:
```



4.2 Reescriba el código del ejercicio 4.1 para que pueda producir algún resultado.



The screenshot shows the SQL Developer interface. The top pane displays a PL/SQL script with the following code:

```
--4.2
SET SERVEROUTPUT ON;
DECLARE
  v_fname employees.first_name%TYPE;
  v_lname employees.last_name%TYPE;
  v_long_fname NUMBER(3,2);
  v_long_lname NUMBER(3,2);
  v_dif_name NUMBER(3,2);
  v_result VARCHAR2(20);
BEGIN
  SELECT first_name, last_name INTO v_fname, v_lname FROM employees
  WHERE employee_id = 204;
  v_long_fname := LENGTH(v_fname);
  v_long_lname := LENGTH(v_lname);
  v_dif_name := v_long_fname - v_long_lname;
  SELECT DECODE(v_dif_name, 0, 'Igual longitud', 'Diferente longitud') INTO v_result FROM DUAL;
  DBMS_OUTPUT.PUT_LINE('Nombre: '||v_fname||CHR(10)||'Apellido: '||v_lname||CHR(10)||'Longitud Nombre: '||v_long_fname||CHR(10)||
    'Longitud Apellido: '||v_long_lname||CHR(10)||
    'Diferencia Nombre-Apellido: '||v_dif_name||CHR(10)||
    'Resultado: '||v_result);
END;
```

The bottom pane shows the output of the script:

```
Nombre: Hermann
Apellido: Baer
Longitud Nombre: 7
Longitud Apellido: 4
Diferencia Nombre-Apellido: 3
Resultado: Diferente longitud

Procedimiento PL/SQL terminado correctamente.
```

4.3 Construya un código PL/SQL que busque la fecha mínima y máxima de los empleados. Busque los meses que hay entre ambas fechas. Use la función MONTHS\_BETWEEN y almacene su resultado en una variable. El resultado debe expresarse en valores enteros. Muestre por pantalla el resultado.



The screenshot shows the SQL Developer interface. The top pane displays a PL/SQL script with the following code:

```
set serveroutput on
declare
  v_months number(5);
begin
  select round( months_between (max(hire_date),min(hire_date))) into v_months
  from employees;
  DBMS_OUTPUT.put_line('Tienen '||v_months||' meses de diferencia');
end;
```

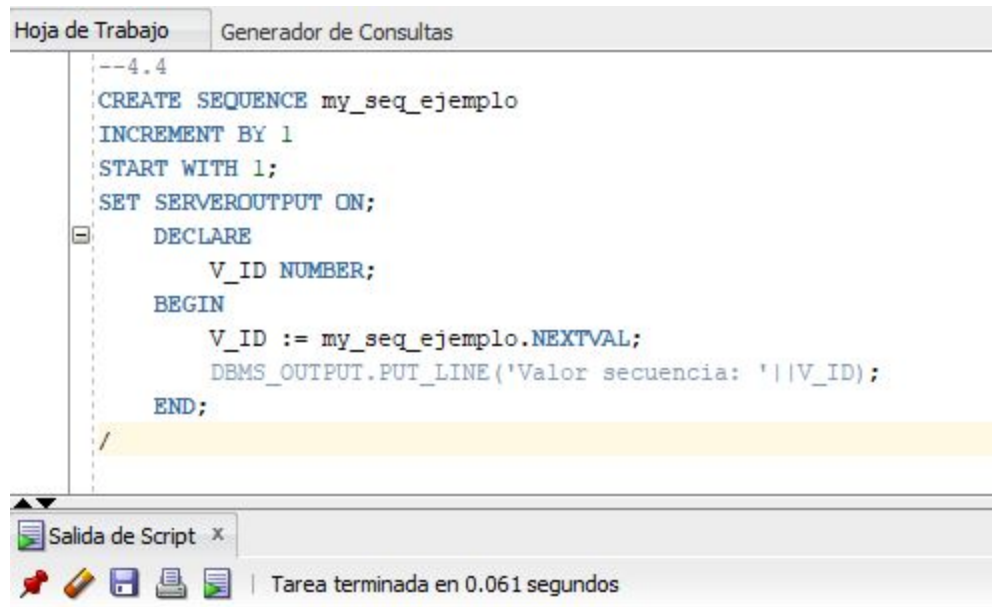
The bottom pane shows the output of the script:

```
Tienen 87 meses de diferencia

Procedimiento PL/SQL terminado correctamente.
```



4.4 Construya un código de PL/SQL donde use una secuencia y asigne el valor directamente a una variable, tal como lo permite Oracle 11g. Muestre el valor de la variable por pantalla.



The screenshot shows the Oracle SQL Developer interface. At the top, there are two tabs: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Generator). The 'Hoja de Trabajo' tab is active, displaying a PL/SQL script. The script starts with a comment '--4.4' and then creates a sequence named 'my\_seq\_ejemplo' with an increment of 1, starting at 1. It then sets 'SERVEROUTPUT ON'. The script declares a variable 'V\_ID' of type 'NUMBER', begins a block, assigns 'my\_seq\_ejemplo.NEXTVAL' to 'V\_ID', and uses 'DBMS\_OUTPUT.PUT\_LINE' to print the value of 'V\_ID'. The script ends with 'END;' and a forward slash '/' to execute it. Below the script editor, there is a 'Salida de Script' (Script Output) window. It shows the output of the script execution: 'Sequence MY\_SEQ\_EJEMPLO creado.' (Sequence MY\_SEQ\_EJEMPLO created.), 'Valor secuencia: 1' (Sequence value: 1), and 'Procedimiento PL/SQL terminado correctamente.' (PL/SQL procedure terminated successfully.). The window also indicates that the task was completed in 0.061 seconds.

```
--4.4
CREATE SEQUENCE my_seq_ejemplo
INCREMENT BY 1
START WITH 1;
SET SERVEROUTPUT ON;
DECLARE
    V_ID NUMBER;
BEGIN
    V_ID := my_seq_ejemplo.NEXTVAL;
    DBMS_OUTPUT.PUT_LINE('Valor secuencia: '||V_ID);
END;
/
```

Salida de Script x

Tarea terminada en 0.061 segundos

Sequence MY\_SEQ\_EJEMPLO creado.

Valor secuencia: 1

Procedimiento PL/SQL terminado correctamente.

4.5 Construya un bloque PL/SQL que contenga bloques anidados y así pueda practicar el alcance y visibilidad de las variables. En el bloque principal busque el apellido y la fecha de ingreso del empleado 204 y en el bloque interno busque el apellido y la fecha de ingreso de su supervisor. En el bloque interno y externo muestre el apellido y la fecha de ingreso de ambos empleados. Explique el resultado que produce el código. Si da algún error indique cómo pudiera evitarlo.

```

--4.5
SET SERVEROUTPUT ON;
DECLARE
    v_lname_ex employees.last_name%TYPE;
    v_date_ex employees.hire_date%TYPE;
BEGIN
    SELECT last_name, hire_date INTO v_lname_ex, v_date_ex
    FROM employees WHERE employee_id = 204;
    DECLARE
        v_lname_in employees.last_name%TYPE;
        v_date_in employees.hire_date%TYPE;
    BEGIN
        SELECT last_name, hire_date INTO v_lname_in, v_date_in FROM employees
        WHERE employee_id = (SELECT manager_id FROM employees WHERE employee_id = 204);
        DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname_in||CHR(10)||'Fecha: '||v_date_in);
        DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname_ex||CHR(10)||'Fecha: '||v_date_ex);
    END;
    DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname_in||CHR(10)||'Fecha: '||v_date_in);
    DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname_ex||CHR(10)||'Fecha: '||v_date_ex);
END;
/

```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0.046 segundos

Informe de error -

```

ORA-06550: line 16, column 50:
PLS-00201: identifier 'V_LNAME_IN' must be declared
ORA-06550: line 16, column 13:
PL/SQL: Statement ignored
06550. 00000 - "line %s, column %s:\n%s"

```

En el bloque principal se hace una consulta que va a guardar en las variables con sufijo *\_ex* los datos solicitados del empleado 204. En el bloque interno va a guardar en las variables declaradas internamente gracias a la acción de una consulta que devuelve los datos solicitados del supervisor del empleado 204 (cuya identificación fue obtenida con una subconsulta). El caso es que el bloque no puede ser ejecutado en ambos bloques, pues las variables del bloque interno fueron declaradas dentro del bloque, y no tienen visibilidad fuera de él. Una solución para esto es declarar todas las variables en el bloque principal.

Hoja de Trabajo

Generador de Consultas

--4.5

SET SERVEROUTPUT ON;

DECLARE

v\_lname\_ex employees.last\_name%TYPE;

v\_date\_ex employees.hire\_date%TYPE;

v\_lname\_in employees.last\_name%TYPE;

v\_date\_in employees.hire\_date%TYPE;

BEGIN

SELECT last\_name, hire\_date INTO v\_lname\_ex, v\_date\_ex

FROM employees WHERE employee\_id = 204;

BEGIN

SELECT last\_name, hire\_date INTO v\_lname\_in, v\_date\_in FROM employees

WHERE employee\_id = (SELECT manager\_id FROM employees WHERE employee\_id = 204);

DBMS\_OUTPUT.PUT\_LINE('Apellido: '||v\_lname\_in||CHR(10)||'Fecha: '||v\_date\_in);

DBMS\_OUTPUT.PUT\_LINE('Apellido: '||v\_lname\_ex||CHR(10)||'Fecha: '||v\_date\_ex||CHR(10));

END;

DBMS\_OUTPUT.PUT\_LINE('Apellido: '||v\_lname\_in||CHR(10)||'Fecha: '||v\_date\_in);





DBMS\_OUTPUT.PUT\_LINE('Apellido: '||v\_lname\_ex||CHR(10)||'Fecha: '||v\_date\_ex||CHR(10));

END;

/

Salida de Script x

Resultado de la Consulta x



Tarea terminada en 0.054 segundos

Apellido: Kochhar  
Fecha: 21-SEP-05  
Apellido: Baer  
Fecha: 07-JUN-02  
  
Apellido: Kochhar  
Fecha: 21-SEP-05  
Apellido: Baer  
Fecha: 07-JUN-02  
  
Procedimiento PL/SQL terminado correctamente.

4.6 Reconstruya el código del ejercicio 4.5 y guarde los valores de las consultas externas e internas en variables con el mismo nombre. Use las etiquetas para imprimir los valores correspondientes a cada bloque.

The screenshot shows the SQL Developer interface with a script editor and a results pane. The script is as follows:

```
--4.6
SET SERVEROUTPUT ON;
<<outer>>
  DECLARE
    v_lname employees.last_name%TYPE;
    v_date employees.hire_date%TYPE;
  BEGIN
    SELECT last_name, hire_date INTO v_lname, v_date
    FROM employees WHERE employee_id = 204;
    <<inner>>
      DECLARE
        v_lname employees.last_name%TYPE;
        v_date employees.hire_date%TYPE;
      BEGIN
        SELECT last_name, hire_date INTO v_lname, v_date FROM employees
        WHERE employee_id = (SELECT manager_id FROM employees WHERE employee_id = 204);
        DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname||CHR(10)||'Fecha: '||v_date||CHR(10));
        DBMS_OUTPUT.PUT_LINE('Apellido: '||outer.v_lname||CHR(10)||'Fecha: '||outer.v_date||CHR(10));
      END inner;
    DBMS_OUTPUT.PUT_LINE('Apellido: '||v_lname||CHR(10)||'Fecha: '||v_date||CHR(10));
  END outer;
```

The results pane shows the output of the script:

```
Apellido: Kochhar
Fecha: 21-SEP-05

Apellido: Baer
Fecha: 07-JUN-02

Apellido: Baer
Fecha: 07-JUN-02
```

4.7 Analice el siguiente código y determine los valores de cada variable de acuerdo a las reglas de alcance.

1. Valor de v\_message en la posición 1.  
*CLERK not eligible for commission*
2. Valor de v\_total\_comp en la posición 2.  
*El valor no está definido en el bloque principal (el valor solo tiene alcance local)*
3. Valor de v\_comm en la posición 1.  
*0*
4. Valor de outer.v\_comm en la posición 1.  
*9000*
5. Valor de v\_comm en la posición 2.  
*9000*
6. Valor de v\_message en la posición 2.  
*SALESMANCLERK not eligible for commission*

```
SET SERVEROUTPUT ON
BEGIN <<outer>>
DECLARE
```

```
v_sal    NUMBER(7,2) := 30000;
v_comm   NUMBER(7,2) := v_sal * 0.15;
v_message VARCHAR2(255) := 'eligible for commission';
BEGIN
  DECLARE
    v_sal        NUMBER(7,2) := 45000;
    v_comm       NUMBER(7,2) := 0;
    v_total_comp NUMBER(7,2) := v_sal + v_comm;
  BEGIN
    v_message := 'CLERK not'||v_message; -- Posición 1
    outer.v_comm := v_sal * 0.20;
  END;
  v_message := 'SALESMAN'||v_message; -- Posición 2
END;
END outer;
/
```