****

**Práctica Bases de Datos Avanzadas**

**Consultas Avanzadas**

Breidy Núñez | 2017-5633

Felipe Ramírez | 2018-6104

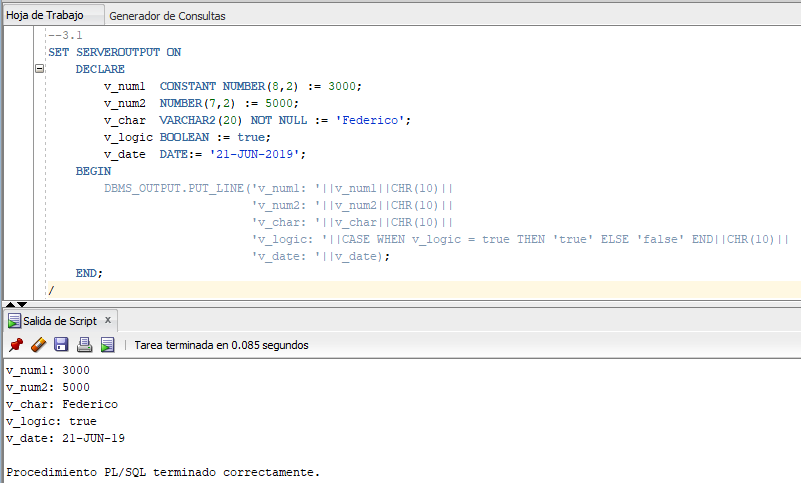
Luis Franco | 2018-6342

**Prof.**  Leandro Fondeur

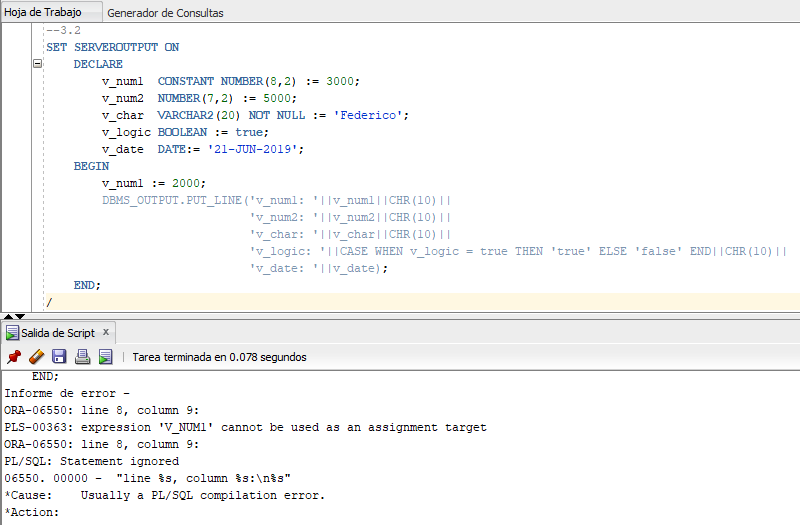
4/7/2019

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

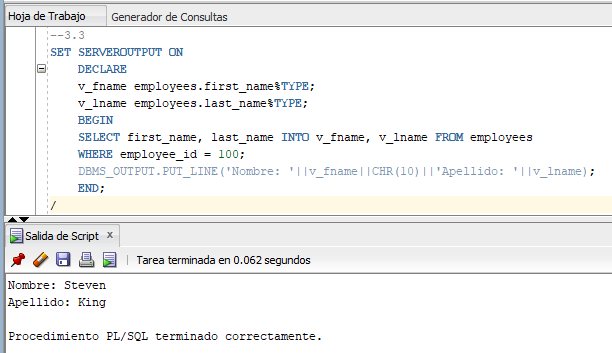
3.1 Construya un bloque PL/SQL donde declare dos variables numéricas, una caracter, una lógica y una tipo fecha, y asígnele valores durante la declaración. Restrinja la variable caracter para que no permita valores nulos y una de las variables numéricas defínela como constante. Muestre por pantalla los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



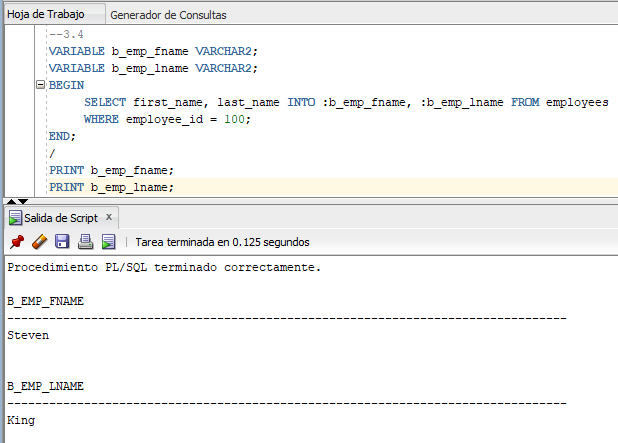
3.2 Modifique el código del ejercicio 1 y asígnele un valor diferente a la variable numérica que fue declarada como constante en el cuerpo del bloque.



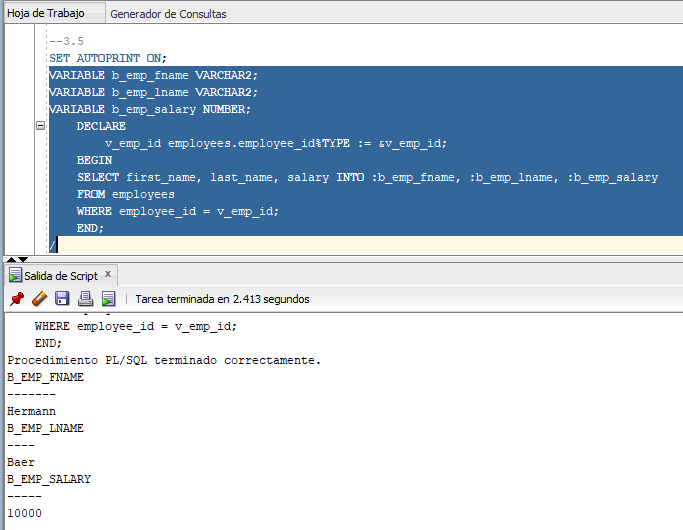
3.3 Construya un bloque PL/SQL donde se busque el nombre y apellido del empleado 100. Declare la variable del apellido usando %TYPE a partir de la columna de la tabla y la variable del nombre usando %TYPE a partir de la variable del apellido. Muestre en pantalla el contenido de las variables.



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.

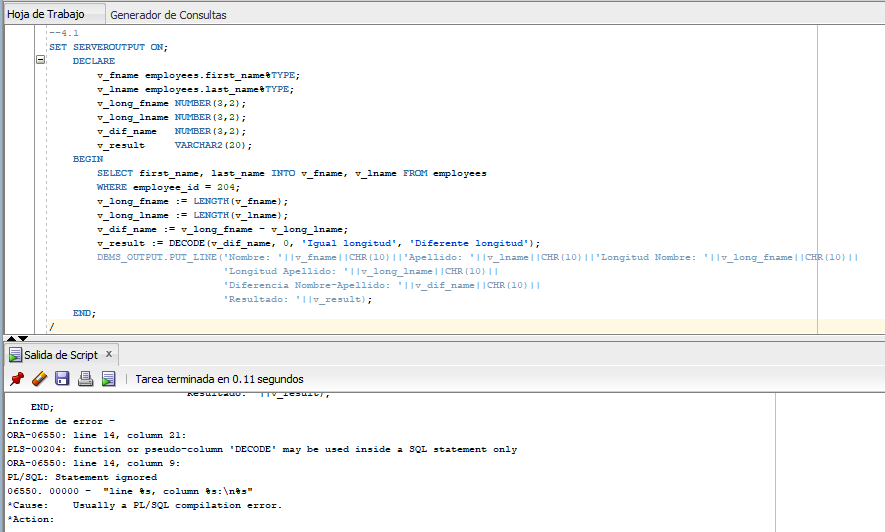


3.5 Construya un bloque PL/SQL donde se pida el ID del empleado 204 usando una variable de substitución en la sección declarativa. Guarde los valores del nombre, apellido y salario en variables de enlace. Permita que se impriman automáticamente sin usar el comando PRINT. Fuera del bloque de código construya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

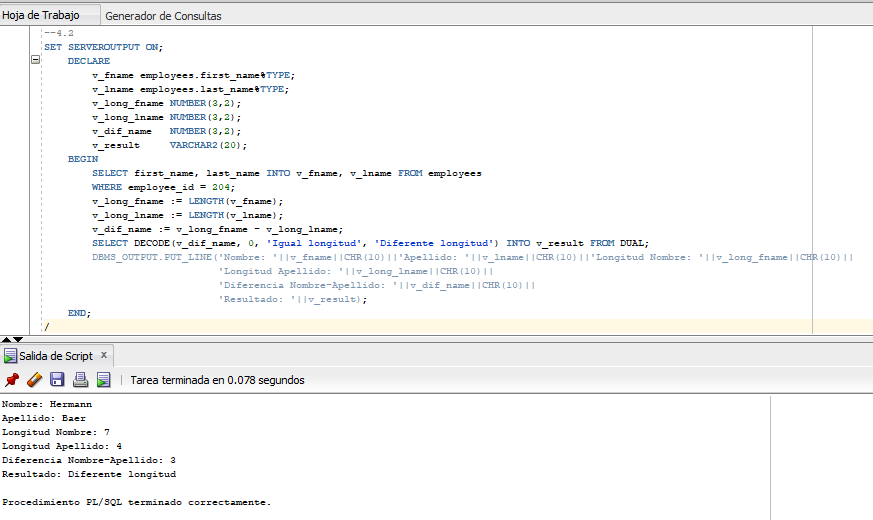


II. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les04.ppt:

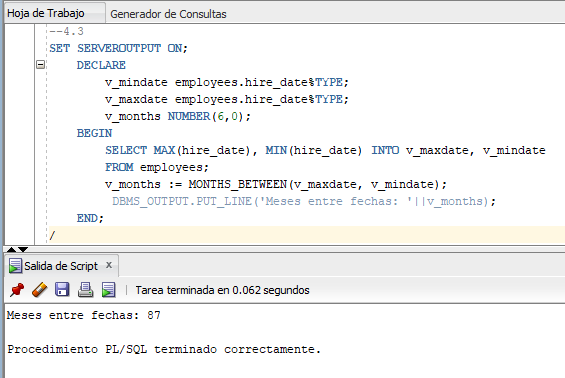
4.1 Escriba un bloque de código PL/SQL que lea el nombre y apellido del empleado 204. Los valores deben ser guardados en dos variables declaradas con %TYPE basadas en las columnas de la tabla. Halle la longitud del nombre y del apellido y también la diferencia de ambas longitudes. Guarde los valores 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.



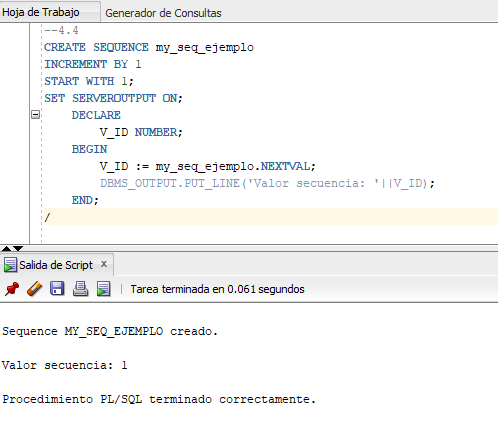
4.2 Reescriba el código del ejercicio 4.1 para que pueda producir algún resultado.



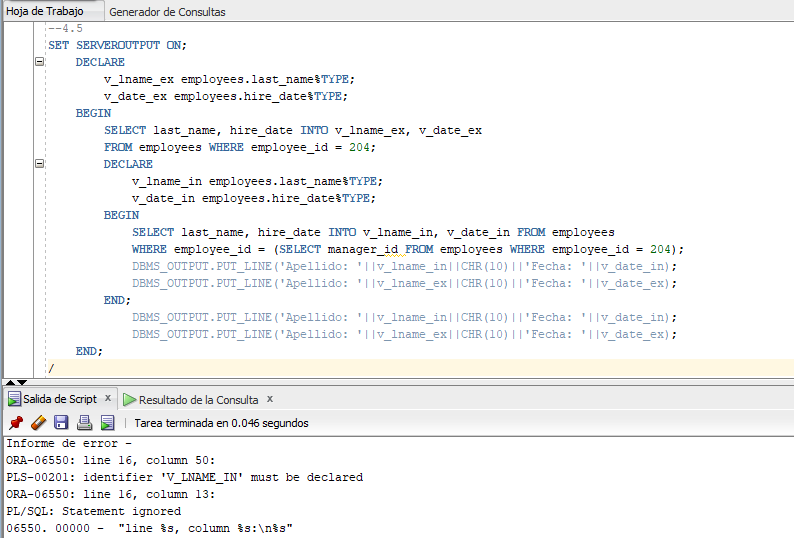
4.3 Construya un código PL/SQL que busque la fecha mínima y máxima de los empleados. Busque los meses que hay entre ambas fechas. Use la función MONTHS\_BETWEEN y almacene su resultado en una variable. El resultado debe expresarse en valores enteros. Muestre por pantalla el resultado.

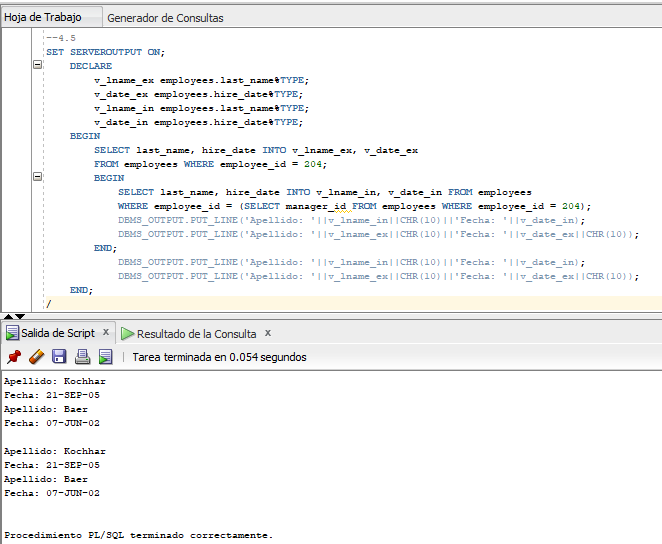


4.4 Construya un código de PL/SQL donde use una secuencia y asigne el valor directamente a una variable, tal como lo permite Oracle 11g. Muestre el valor de la variable por pantalla.

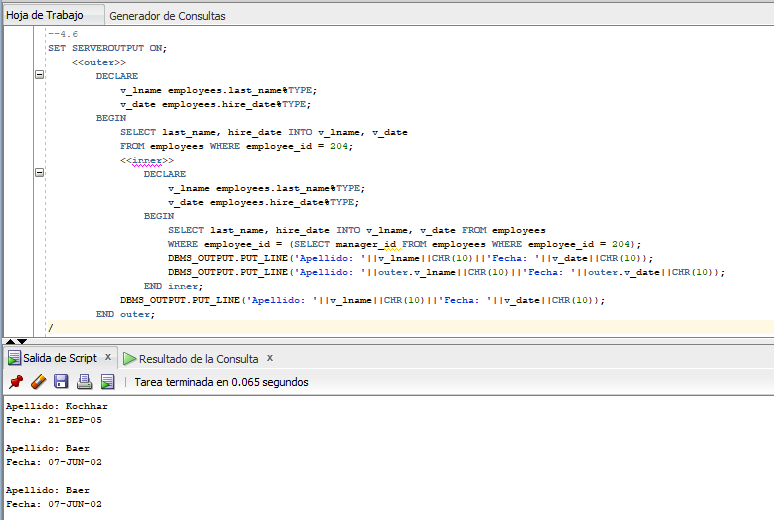


4.5 Construya un bloque PL/SQL que contenga bloques anidados y así pueda practicar el alcance y visibilidad de las variables. En el bloque principal busque el apellido y la fecha de ingreso del empleado 204 y en el bloque interno busque el apellido y la fecha de ingreso de su supervisor. En el bloque 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.

*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.*



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.



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*

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;

/