****

**Práctica Bases de Datos Avanzadas**

**Consultas Avanzadas**

Breidy Núñez | 2017-5633

Felipe Ramírez | 2018-6104

Luis Franco | 2018-6342

**Prof.**  Leandro Fondeur

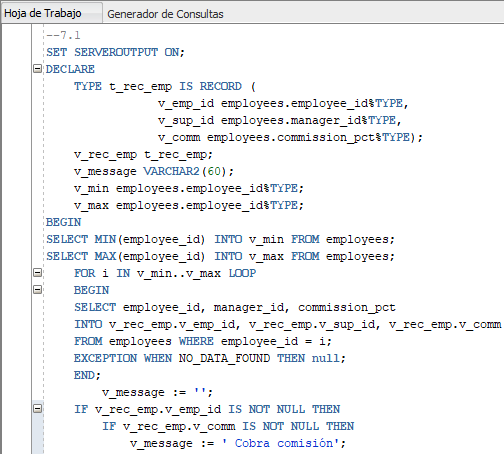
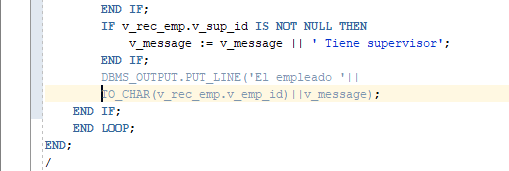
25/7/2019

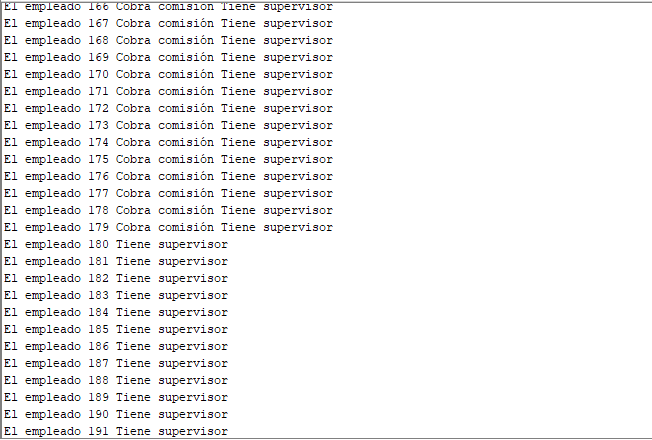
I. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les07.ppt:

7.1 Construya un código PL/SQL que lea el ID del empleado, ID del supervisor y comisión. Si el empleado cobra comisión que guarde en una variable "Cobra comisión". Además, si el empleado se reporta a un supervisor que guarde en la misma variable "Tiene supervisor". Al final, mostrar en pantalla el siguiente mensaje:

"El empleado <<ID del empleado>> <<Mensaje>>".

Guarde la información leída de la tabla de empleados en una variable tipo RECORD. Permitir que el código sea reusable.



7.2 Primero deben crear una tabla [EMP\_GRADES] que tenga la siguiente

estructura:

EMP\_ID NUMBER(6)

NAME VARCHAR2(60) -- Lastname + ', ' + Firstname

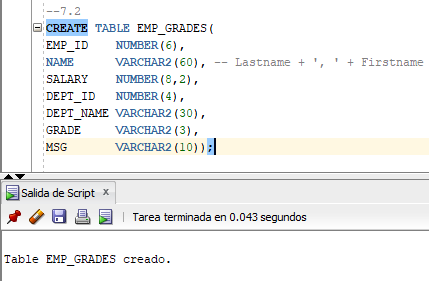
SALARY NUMBER(8,2)

DEPT\_ID NUMBER(4)

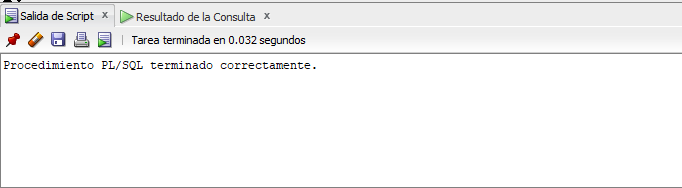
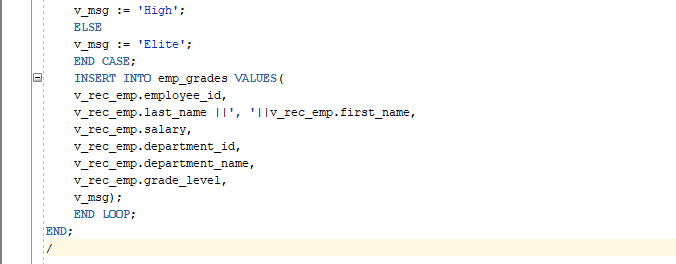
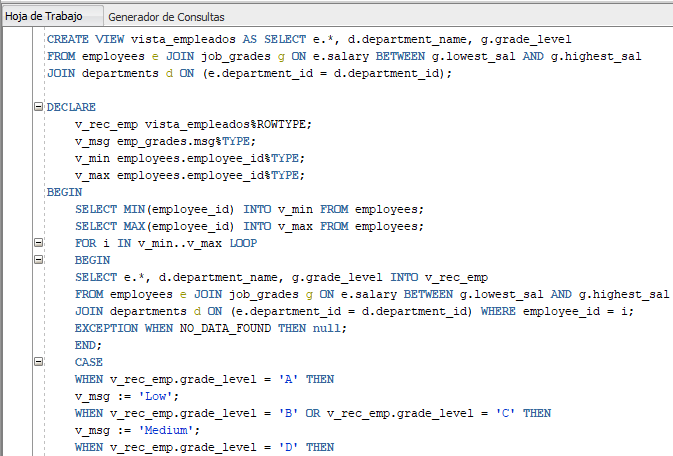
DEPT\_NAME VARCHAR2(30)

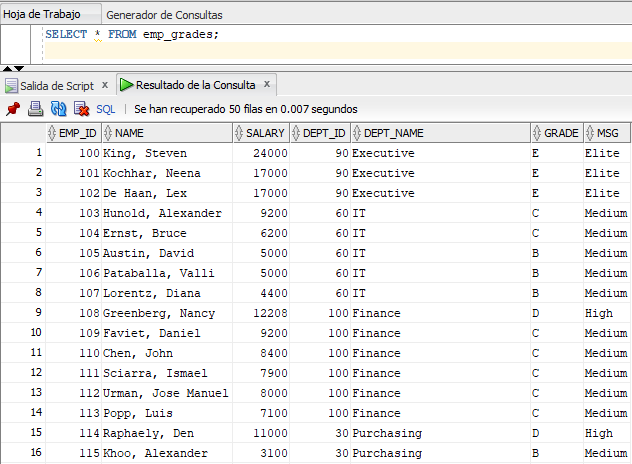
GRADE VARCHAR2(3)

MSG VARCHAR2(10)

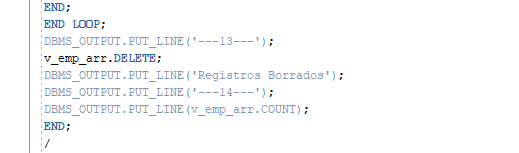
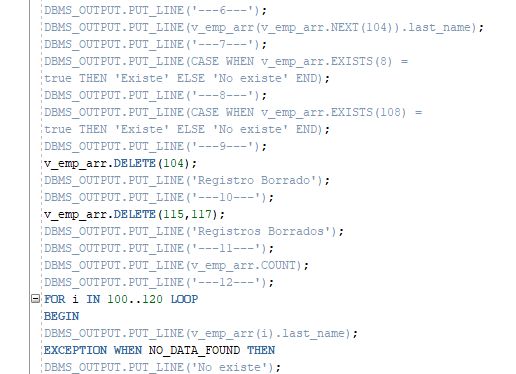
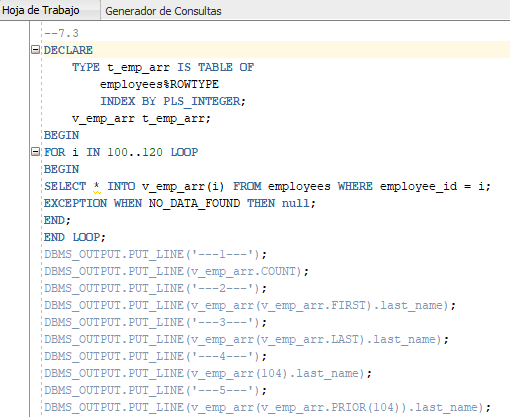


Cree una variable RECORD que contenga el registro completo de empleados, el nombre de departamento y el nivel salarial (para simplificar la construcción de la variable RECORD cree una vista). En el cuerpo del código, deberán leer la información correspondiente de las tablas de empleados, departamentos y niveles salariales y guardar los datos en la variable tipo RECORD. A continuación deberán construir una estructura CASE para evaluar el nivel salarial. Deberán almacenar en una variable un texto dependiendo del nivel salarial. Si es "A" almacenan "Low", si es "B" o "C" almacenan "Medium", si es "D" almacenan "High", de lo contrario almacenan "Elite". Finalmente deberán insertar en la tabla creada inicialmente los valores obtenidos. Fuera del código, revise el contenido de la tabla EMP\_GRADES.





7.3 Construir un código PL/SQL usando una estructura de arreglo asociativa (INDEX-BY table). En la misma deberás guardar los registros de la tabla de empleados para los ID de empleados del 100 al 120. Las siguientes tareas deberán realizarse a partir del arreglo asociativo construido y usando los métodos de tablas INDEX BY:



1. Mostrar en pantalla la cantidad de registros.



2. Mostrar el apellido del primer registro.



3. Mostrar el apellido del último registro.



4. Mostrar el apellido del registro 104.



5. Mostrar el apellido del registro anterior al 104.



6. Mostrar el apellido del registro posterior al 104.



7. Mostrar si el registro 8 existe o no.



8. Mostrar si el registro 108 existen o no.



9. Eliminar el registro 104.



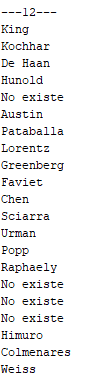
10. Eliminar el rango de registros del 115 al 117.



11. Mostrar la cantidad de registros.



12. Mostrar el apellido de todos los registros del 100 al 120, en caso de que un índice no exista en dicho rango, mostrar "No existe".



13. Borrar todos los registros.



14. Mostrar la cantidad de registros.



7.4 Este ejercicio es opcional --> Crear una tabla que tenga una columna NESTED TABLE. Las siguientes son las tareas a realizar en ese mismo orden:

1. Crear el tipo de datos my\_tab\_t como TABLE de VARCHAR2(30).

2. Describir el tipo de datos.

3. Seleccionar el diccionario de datos USER\_SOURCE para MY\_TAB\_T.

4. Crear la tabla NESTED\_TABLE con la siguiente estructura:

id NUMBER

col1 my\_tab\_t

Usar la columna col1 como NESTED TABLE y su STORE AS sea col1\_tab.

5. Revisar la estructura de la tabla NESTED\_TABLE.

6. Revisar la estructura de la tabla col1\_tab.

7. Seleccionar el diccionario de datos USER\_TABLES.

8. Seleccionar el diccionario de datos USER\_TAB\_COLS.

9. Seleccionar el diccionario de datos USER\_NESTED\_TABLES

10. Insertar los siguientes registros en la tabla anidada y asentar los cambios a la BD

1, ['A']

2, ['B','C']

3, ['D','E','F']

11. Mostrar el contenido de la tabla NESTED\_TABLE.

12. Construya un programa PL/SQL que actualice la tabla anidada para que el ID 1 quede como sigue:

1, ['A','Z']

13. Mostrar el contenido de la tabla NESTED\_TABLE.

14. Cree una instrucción SELECT que permita desanidar (Collection Unnesting) la sub-tabla

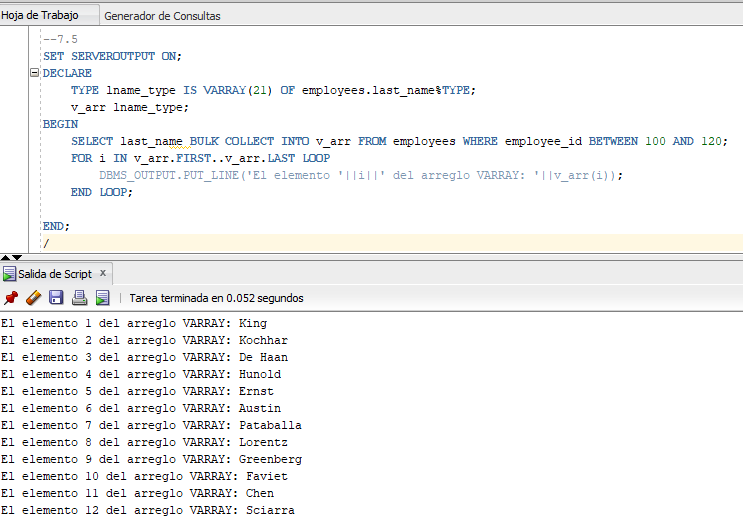
15. Borrar la sub-tabla

16. Eliminar la columna COL1 de la tabla NESTED\_TABLE.

17. Revisar la estructura de la tabla NESTED\_TABLE.

7.5 Construir un código PL/SQL que cree una variable tipo VARRAY de tamaño 21 para almacenar los apellidos de los ID de empleados 100 al 120. Para llenar el arreglo, usen BULK COLLECT INTO. Mostrar por pantalla los datos obtenidos con el siguiente formato:

Elemento <<# de elemento>> del arreglo VARRAY: <<Apellido>>

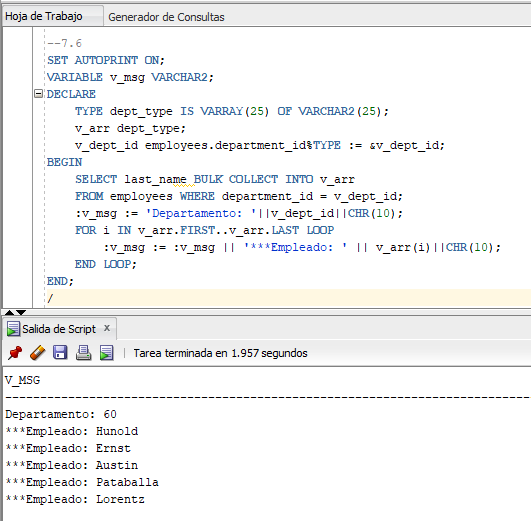


7.6 Crear un tipo de dato VARRAY (dept\_type) que sea VARCHAR2(25). El tamaño del arreglo debe ser de 25. Luego deberán crear una variable que use el tipo VARRAY creado anteriormente. En dicha variable almacenen todos los apellidos de los empleados que trabajan en un departamento dado por pantalla. Mostrar la información de la siguiente forma:

Departamento: <<ID del departamento>>

\*\*\* Empleado: <<Apellido>>

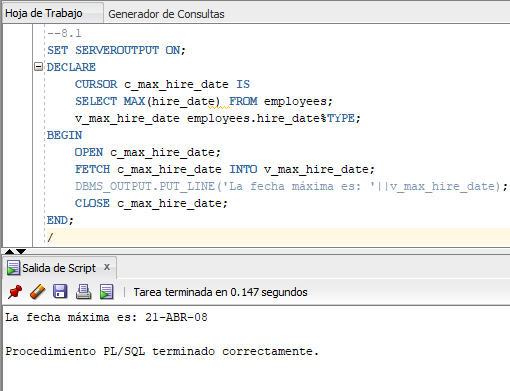
Usen los comandos para definir una variable de ambiente y para no visualizar el texto de un comando.



II. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les08.ppt:

8.1 Construya un código PL/SQL que usando un CURSOR explícito busque la fecha máxima de contratación en la tabla de empleados. Use FETCH para leer cada fila del cursor. Analice cuántos registros retornará el cursor. Basado en su análisis elija la mejor forma de construir el código. Muestre el siguiente mensaje:

La máxima fecha es <<Fecha>>

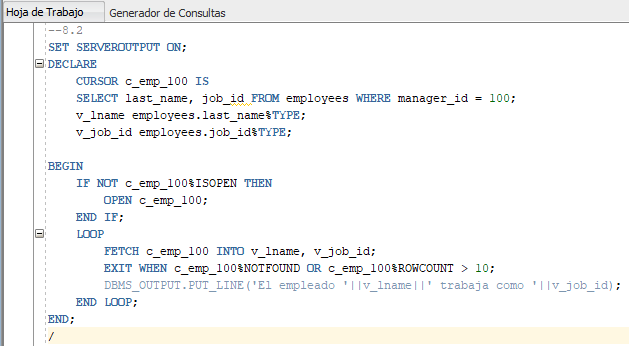


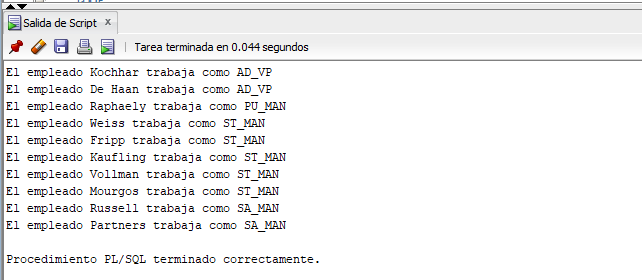
*Como las funciones de múltiples filas devuelven un solo valor por sí mismas, el Cursor sólo tendrá un valor, por tanto, con un solo FETCH sin iterar es suficiente para mostrar el resultado.*

8.2 Construya un código PL/SQL usando un CURSOR explícito que busque el apellido y la posición de todos los empleados que se reportan al supervisor 100. Mostrar el siguiente mensaje sólo para los primeros 10 empleados:

El empleado <<Apellido>> trabaja como <<Posición>>

Utilice FETCH, %NOTFOUND y %ROWCOUNT. Ante de abrir el CURSOR valide que el mismo no se encuentre abierto usando %ISOPEN.



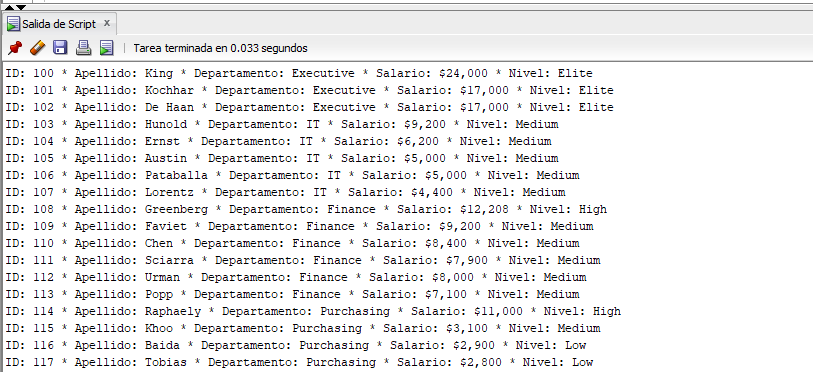
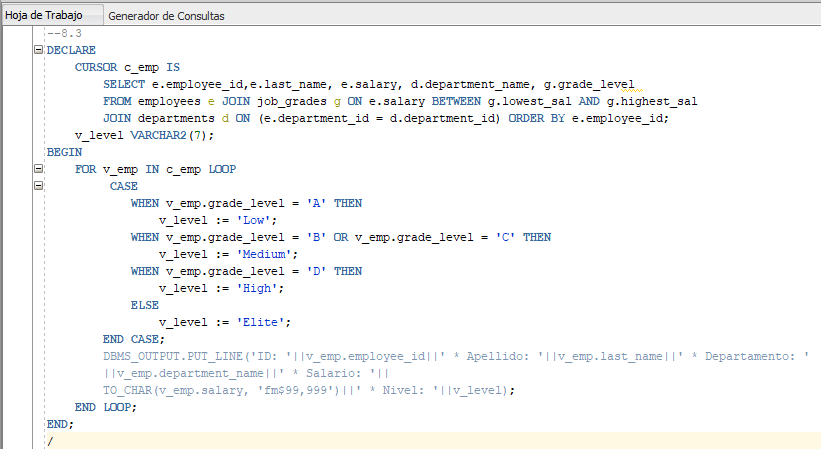


8.3 Construya un código PL/SQL usando cursor FOR LOOP que lea los ID del empleado, apellido, salario, nombre de departamento y nivel salarial. Relacione sus datos con la tabla JOB\_GRADES para conocer su nivel salarial. Si el nivel del empleado es 'A' se desea desplegar "Low", si es 'B' o 'C' que despliegue 'Medium', si es 'D' que despliegue 'High' y si no es ninguno de estos niveles, que despliegue 'Elite'. Ordene los datos por el nivel salarial. Mueste la información por pantalla como sigue:

Id: <<Id del empleado>> \* Apellido: <<Apellido>> \*

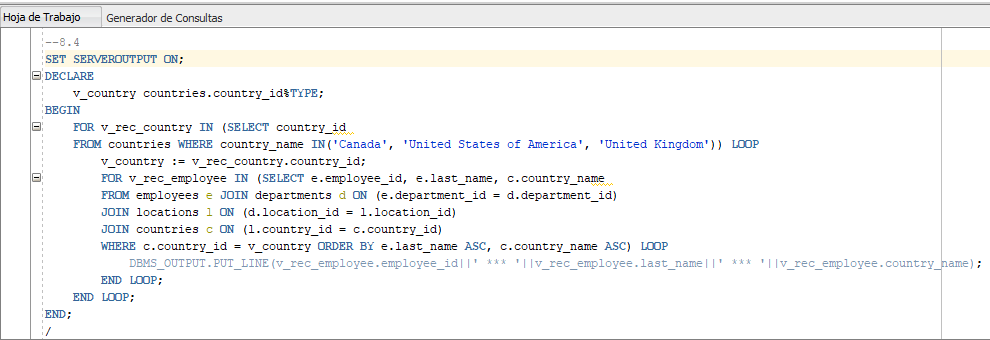
Departamento: <<Nombre del Departamento>> \* Salary: <<Salario>> \* Nivel: <<Nivel>>

El salario debe salir con el siguiente formato: 'fm$99,999'



8.4 Construya un código PL/SQL usando un cursor FOR LOOP con subquery. Muestre el ID, apellido y país de todos los empleados que trabajen en Canadá y United Kingdom. Ordene la data por país y apellido de manera ascendente. El formato de la salida debe ser como sigue:

ID \*\*\* Apellido \*\*\* País

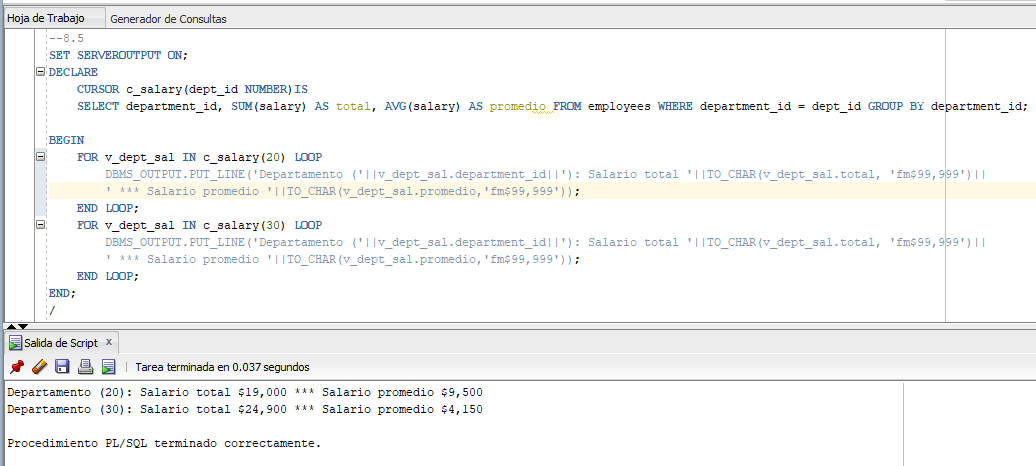




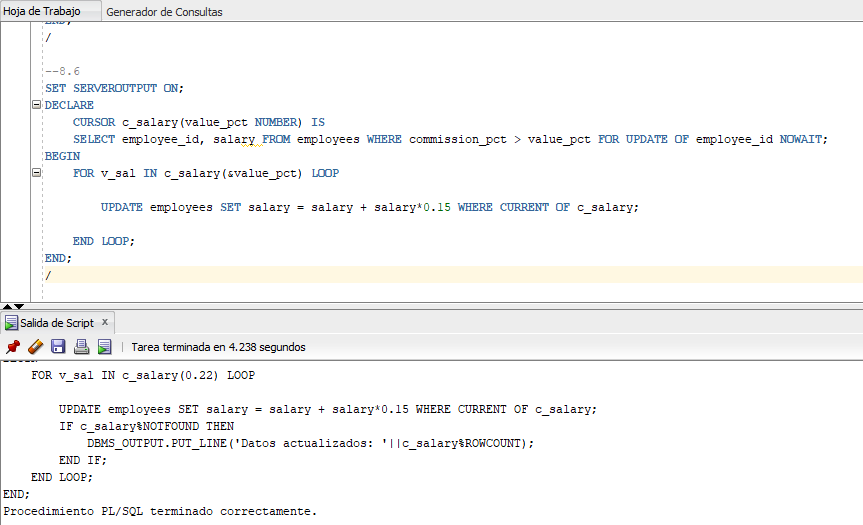
8.5 Construya un código PL/SQL con un cursor que busque el salario total y el salario promedio que trabajan en un departamento dado. Al momento de abrir el cursor pase los valores 20 y 30 como parámetros. Muestre la información como sigue:

Departamento(<<ID>>): Salario total <<total>> \*\*\* Salario promedio <<promedio>>

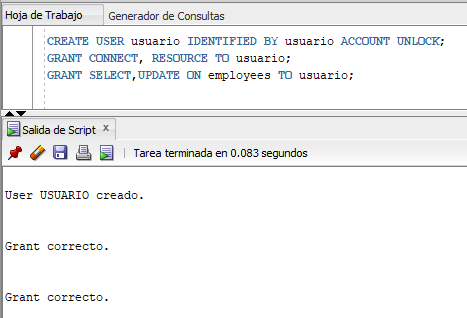
Los valores del salario deben salir con el siguiente formato: 'fm$99,999'

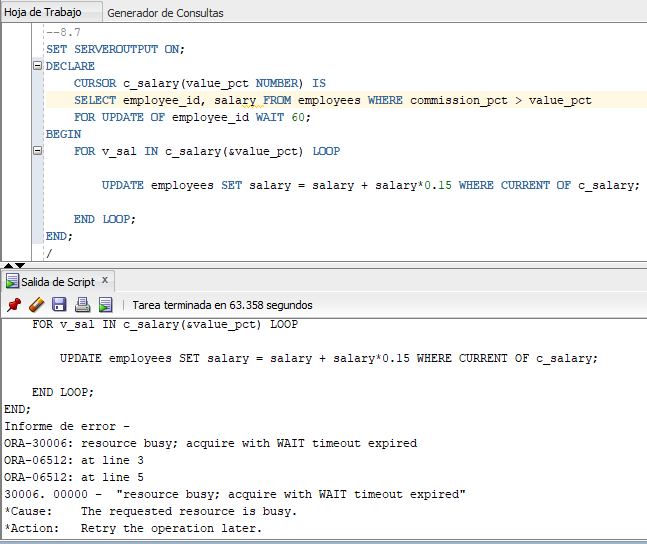


8.6 Construya un código PL/SQL que busque el ID y salario de los empleados que ganan una comisión mayor a un valor dado y bloquee el salario para que no sea actualizado por otra sesión. En la sección de ejecución actualice el salario de los empleados que vengan en el cursor en un 15%. Para realizar esta práctica use un cursor FOR LOOP con parámetros, FOR UDPATE OF, NOWAIT y CURRENT OF.



8.7 Realice la práctica 8.6 usando WAIT por 60 segundos. Cree un usuario al cual se le dé privilegios de actualización sobre la tabla EMPLOYEES de HR. Actualice el salario en un 15% al empleado 146 (no asiente los cambios). Corra el programa y comente qué pasa. Luego en la otra sesión, deshaga el cambio y comente el comportamiento del programa.





*El programa estuvo esperando por 60 segundos en la espera de que el recurso se liberara para efectuar el UPDATE. No se pudo efectuar porque no se habían asentado los cambios realizados por usuario.*

*Tras hacer el rollback desde usuario, los recursos fueron liberados y el programa pudo ejecutarse correctamente.*

