

Práctica Unidad 3 : Interacción con Oracle

Primera Parte: Sentencias de manipulación de datos

1. Crear un bloque PL/Sql que solicite el número de empleado usando una variable de sustitución y dependiendo del monto de su sueldo incrementar su comisión según el siguiente criterio:
 - Si el sueldo es menor a 1300 el incremento es de 10%
 - Si el sueldo está entre 1300 y 1500 el incremento es de 15%
 - Si el sueldo es mayor a 1500 el incremento es de 20%
 - Tener en cuenta que puede haber comisiones en NULL
 - Si el empleado no existe mandar un mensaje de error.
2. Modificar el ejercicio anterior para actualizar la comisión de todos los empleados de acuerdo a su sueldo usando los mismos criterios. Desplegar mensajes indicando cuantos registros fueron actualizados según cada criterio.
3. Crear un bloque PL/Sql que permita dar de baja cargos que ya no se usan (usar la tabla JOB):
 - Eliminar de la tabla JOB aquella fila cuyo Job_Id es ingresado con una variable de sustitución del SqlDeveloper.
 - Capturar e informar mediante excepciones o atributos del cursor , las siguientes eventualidades: no existe el código de cargo ingresado (Sql%Notfound o Sql%Rowcount) no puede eliminar un cargo que está asignado a empleados (Asociar una excepción con el error correspondiente) .
4. Escribir un bloque PL/Sql para insertar un nuevo cargo en la tabla JOB:
 - El Job_Id debe generarse sumando 1 al máximo Job_Id existente. Para esto primero encontrar el Max(Job_Id) y guardarlo en una variable.
 - El nombre del cargo (Function) debe ser informado desde una variable de sustitución del SqlDeveloper (usar Becario o Estudiante). En la tabla JOB los nombres de función deben estar en mayúsculas.
 - Asentar en la base de datos este insert (Commit).
 - Una vez que se ejecutó el bloque PL/Sql consultar desde SqlDeveloper todo el contenido de la tabla JOB.
5. Escribir un bloque PL/SQL que actualice el precio de lista de un producto de acuerdo al número de veces que el producto se vendió:
 - Use una variable de sustitución para ingresar el producto.
 - Calcule las veces que el producto se vendió (Tabla ITEM). Si el producto se vendió 2 veces o menos decremente su precio en un 10%. Si se vendió más de 2 veces decremente su precio en un 20% y no se vendió nunca en un 50%.
 - Tenga en cuenta que el precio de lista vigente de un producto es aquel que tiene la columna END_DATE en null.
6. Modificar el ejercicio 5 de la práctica 2 para mostrar un mensaje en caso de no existir la orden.
7. Modificar el ejercicio 8 de la práctica 2 para ingresar el apellido del empleado y mostrar su id, nombre, salario y asteriscos de acuerdo al salario.
 - Mostrar mensajes si no existe empleado con dicho apellido, o si hay más de un empleado con ese apellido.

Segunda Parte: Cursores explícitos

8. Usando un cursor recorrer las tablas Sales_order e Ítem para generar un listado sobre todas las órdenes y los productos que se ordenaron en ellas. Mostrar los siguientes datos: Order_id, order_date, product_id.
9. Escribir un bloque que reciba un código de cliente e informe el nro. de orden, la fecha de toda orden generada por él y la descripción de los productos ordenados. (Usar las tablas Sales_order, Ítem y Product). Si no hay registros desplegar un mensaje de error.
10. Necesitamos tener una lista de los empleados que son candidatos a un aumento de salario en los distintos departamentos:
Indicar el id de departamento a través de una variable de sustitución
Recuperar apellido, nombre, y salario de los empleados que trabajan en el departamento dado y cuyo cargo sea 'CLERK'.
Si el salario es menor que 1000 desplegar el mensaje:
Last_Name, First_name candidato a un aumento
Si el salario supera los 1000 (o es igual) desplegar :
Last_Name, First_name no es candidato a un aumento
El listado debe estar ordenado por apellido.
Si en el departamento no existe la función 'CLERK' desplegar el mensaje:
El departamento *Department_Id* no tiene candidatos a aumento de salario.
Probar el bloque con distintos departamentos.
11. Escribir un bloque PL/Sql que muestre los 5 productos más caros.
12. Usando dos cursores, recorrer las tablas Department y Employee para generar un listado mostrando los datos de todos los departamentos y por cada uno el nombre completo y fecha de ingreso de sus empleados. Ordenar los datos por id de departamento y nombre de empleado.
El listado deberá mostrarse así:

<u>10 - ACCOUNTING - NEW YORK</u>		
CLARK,CAROL		09-Jun-1985
KING,FRANCIS		17-Nov-1985

<u>12 - RESEARCH - NEW YORK</u>		
ALBERTS,CHRIS		06-Apr-1985
13. Hacer un bloque que dado un departamento, despliega el nombre completo, cargo y salario de todos sus empleados ordenados por salario en forma ascendente y luego en forma descendente.
Para esto cargar en una tabla en memoria los empleados ordenados por salario.
Si el departamento no existe o no tiene empleados cancelar
Cargar la tabla con Bulk Collect.

Opcional

14. Escribir un bloque PL/Sql donde se guarden en una tabla PL/Sql indexada todos los datos de la tabla Department. Los datos de cada departamento deberán guardarse en la posición correspondiente a su Id. A continuación mostrar la cantidad de departamento que hay y luego todos los datos cargados.