****

**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\_les09.ppt:

9.1 Construya un código PL/SQL que busque el apellido y ID del empleado cuyo apellido sea 'King'. No use cursores. Maneje las excepciones que se puedan presentar. Muestre por pantalla la siguiente información:

"Apellido del empleado: <<Apellido>>"

"ID del empleado: <<ID>>"

9.2 Construya un código PL/SQL que inserte un registro en la tabla de empleados. Los valores a insertar son:

"ID empleado [NULL], Apellido [Reyes]"

Maneje una excepción no-predefinida. Muestre por pantalla el siguiente mensaje:

"Insersión fallida..."

"<<Mensaje de error capturado por SQL>>

9.3 Realice las siguientes actividades:

1. Cree la tabla CT\_ERRORS con la siguiente estructura:

E\_USER VARCHAR2(10)

E\_DATE DATE

E\_CODE NUMBER(6)

E\_MSG VARCHAR2(255)

2. Construya un código PL/SQL que inserte un registro en la tabla de departamentos. Los valores a insertar son:

"ID Departamento [NULL], Nombre [Publishing]"

Maneje la excepción usando WHEN OTHERS. Muestre por pantalla el siguiente mensaje:

"Insersión fallida, mensaje de error grabado en la tabla CT\_ERRORS"

En la tabla ct\_errors inserte:

- Usuario de la sesión

- Fecha del sistema

- Código y mensaje de error

3. Muestre el contenido de la tabla CT\_ERRORS.

9.4 Construya un código PL/SQL que actualice el salario de un empleado dado en un % dado. Si el registro fue actualizado mostrar por pantalla el siguiente mensaje:

"Registro actualizado exitosamente..."

Si la actualización dio error mostrar por pantalla el siguiente mensaje de error:

"El empleado no existe. Revise el ID!"

Use un tipo de excepción definida por el usuario.

9.5 Construya un código PL/SQL que busque las localidades que se encuentren en las ciudades de Toronto, Munich y Mexico City. Esto puede hacerlo usando un cursor. Use el cursor en el bloque interno. Ordene los datos por ID de localidad. En un bloque interno aumente en un 25% el salario de los empleados cuyo departamento se encuentre en las localidades de las ciudades dadas. Para actualizar los datos puede usar un sub-query. Muestre la cantidad de registros actualizados en cada localidad. Maneje cualquier error que surja al momento de la actualización usando excepciones manejadas por el usuario. Defina la excepción en el bloque externo. Muestre el siguiente mensaje de error:

'Departamento no existe. Favor revise la localidad: <<ID Localidad>>'

Maneje también cualquier error producido durante el manejo del cursor en el bloque interno. Muestre el siguiente mensaje de error: 'Acceso a cursor inválido!'

9.6 Construya un código PL/SQL que elimine un empleado a partir de un apellido dado. Maneje cualquier error presentado durante el borrado a través del uso de un RAISE\_APPLICATION\_ERROR en la sección de ejecución. Muestre el siguiente mensaje de error:

ORA-20089: Apellido introducido no es válido

9.7 Construya un código PL/SQL que elimine un empleado a partir de un apellido dado. Maneje cualquier error presentado durante el borrado a través del uso de un RAISE\_APPLICATION\_ERROR en la sección de excepción. Muestre el siguiente mensaje de error:

ORA-20089: Apellido introducido no es válido

I. Realice los siguientes ejercicios luego de estudiar el documento D64254GC11\_les10.ppt:

10.1 Construir un código PL/SQL que cree un procedimiento que actualice el salario en un 5% de un empleado suministrado por pantalla. Muestre el apellido y el salario del empleado, antes y después de actualizar el registro. Ejecute el procedimiento desde un bloque de programa no nombrado.

10.2 Construir un código PL/SQL anónimo que invoque un procedimiento pasándole como parámetro un ID del empleado que fue introducido por pantalla y que imprima por pantalla el apellido y el salario de dicho empleado. Desplegar los datos como:

"Detalle del empleado: <<Apellido>>, <<Salario>>"

Construir un código PL/SQL que cree un procedimiento que busque el apellido y el salario de un empleado recibido del código anterior. Definir dos parámetros de salida para devolver el apellido y el salario de dicho empleado.

10.3 A) Construir un código PL/SQL que construya un procedimiento llamado raise\_comm, el cual reciba dos parámetros (ID del empleado y Tasa de aumento). Este procedimiento tiene como labor actualizar el salario del empleado dado a la tasa de aumento recibida.

B) Construir un código PL/SQL que construya un procedimiento llamado employee\_proc, el cual construya un cursor que busque el ID del empleado, el apellido y el salario de todos los empleados que trabajan como IT\_PROG.

Si el empleado gana entre 4000 y 5000, deberá tener una tasa de aumento de 10%.

Si el empleado gana entre 5001 y 6000, deberá tener una tasa de aumento de 8%.

Si el empleado gana entre 6001 y 7000, deberá tener una tasa de aumento de 6%.

Si el empleado gana cualquier otro valor, deberá tener una tasa de aumento de 4%.

Ejecute el procedimiento raise\_comm pasando como parámetros el ID del empleado y la tasa de aumento correspondiente.

Desplegar la siguiente información:

--- Datos del empleado ---

\*\*\* Apellido: <<apellido>>

\*\*\* Aumento [%]: <<aumento>>% --> 8%

\*\*\* Salario [V]: <<salario viejo>> --> $99,999.00

\*\*\* Salario [N]: <<salario nuevo>> --> $99,999.00

Ejecute el procedimiento employee\_proc usando EXECUTE.

10.4 Construya un código PL/SQL que cree la función tax, la cual reciba como parámetro el salario de un empleado dado y que retorne el salario aumentado en un 10%.

Realice una consulta usando una instrucción SQL. En la misma seleccione el ID del empleado y el valor devuelto por la función (llamar función), para todos aquellos empleados cuyo valor devuelto por la función (llamar función) sea mayor (query principal) al valor máximo devuelto por la función (llamar función) de los empleados que laboren en el departamento 30 (subquery). Ordene los valores por el valor devuelto por la función (llamar función).