

# **TRABAJO PRACTICO N° 4**

## **PL/SQL**

**CARRERA: ANALISTA DE SISTEMAS**

**CURSO: 3er. CUATRIMESTRE**

**AÑO LECTIVO: 2016**

**PROFESOR: Ing. GLATZEL, Adrian**

**FECHA DE ENTREGA: 18/11/2016**

## Objetivo

- Incorporar técnicas de programación mediante la creación de procedimientos, funciones y triggers haciendo uso de PL/SQL.

## Consideraciones

- El motor de base de datos utilizado será Oracle Database Express Edition 11g Release 2  
<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>
- Para acceder a la base de datos se utilizará SQL Developer 4.1.3 (versión recomendada Windows 64-bit with JDK 8 included)  
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>
- **Se deberá trabajar sobre el esquema HR** (USER: HR / PASS: HR).  
Ver Anexo I para desbloquear el usuario HR.
- La entrega se realizará vía mail a [adrian.glatzel@davinci.edu.ar](mailto:adrian.glatzel@davinci.edu.ar) (con copia a todos los integrantes del grupo) y se evaluará en forma presencial en el aula 125.
- El asunto del mail debe ser **Grupo N – TP #** donde **N** es el número de grupo que se les asignará y **#** es el número de trabajo práctico. Por ejemplo: **Grupo 1 – TP 1**
- Se debe entregar un archivo con extensión **.sql** que contenga todas las sentencias solicitadas. El nombre del archivo debe ser **GrupoN\_TP#** donde **N** es el número de grupo que se les asignará y **#** es el numero de trabajo práctico. Por ejemplo: **Grupo1\_TP1.sql**
- La estructura del archivo debe ser:  

```
--EJERCICIO 1  
SENTENCIA 1;  
  
--EJERCICIO 2  
SENTENCIA 2;  
...  
--EJERCICIO N  
SENTENCIA N;
```

## Enunciado

1. Crear una función llamada **YEAR** que reciba un parámetro de tipo fecha y retorne el año correspondiente. Elaborar una sentencia SQL que retorne los empleados que ingresaron en el año 2007 utilizando la función YEAR.
2. Crear una función llamada **MONTH** que reciba un parámetro de tipo fecha y retorne el nombre del mes correspondiente (Enero, Febrero, Marzo, etc.).
3. Crear una función llamada **DAY** que reciba un parámetro de tipo fecha y retorne el nombre del día correspondiente (Lunes, Martes, etc.).
4. Crear una función llamada **FN\_REGION** que reciba un identificador de LOCATION (LOCATION\_ID) y retorne el nombre de la región a la cual pertenece.
5. Crear una función llamada **FN\_CALCULAR\_ANTIG** que reciba un id de empleado y retorne la cantidad de años trabajados desde su contratación hasta la actualidad.
6. Crear un procedimiento llamado **PR\_AUMENTO\_SUELDO\_DEPTO** que reciba como parámetros un identificador de departamento y un valor numérico de porcentaje de aumento. Dicho procedimiento debe efectuar un aumento del salario de cada empleado del departamento en base al valor (porcentaje) recibido como parámetro. El procedimiento debe devolver mediante un tercer parámetro de salida de tipo texto el valor 'OK' en caso de aplicar el aumento para todos los empleados o bien retornar 'ERROR' en caso que ocurra algún error durante la ejecución. Se debe verificar previamente a aplicar los cambios que ambos parámetros de entrada no sean nulos (si alguno es nulo retornar 'ERROR – VERIFICAR PARAMETROS NULOS' en el parámetro de salida).
7. Crear un procedimiento llamado **PR\_DETALLE\_EMPLEADOS\_DEPTO** que reciba como parámetro un identificador de departamento e imprima en pantalla como primera línea el nombre del departamento y luego detalle el nombre y apellido de cada empleado del departamento junto con su función y salario. Si el empleado cumple 10 años de antigüedad debe indicarse también. El formato que debe adoptar la salida es el siguiente:

```
-----  
DEPARTAMENTO: IT  
-----  
EMPLEADO: Alexander Hunold  
FUNCION: Programmer  
SALARIO: 12472,74  
** CUMPLE 10 AÑOS **  
-----  
EMPLEADO: Bruce Ernst  
FUNCION: Programmer  
SALARIO: 8315,15  
-----
```

8. Crear un procedimiento llamado **PR\_AUMENTOS** que lea los registros de la tabla temporal CFG\_AUMENTO\_DEPARTAMENTO mediante un cursor y ejecute el procedure del punto 6 (PR\_AUMENTO\_SUELDO\_DEPTO) para cada registro configurado.

Los campos de la tabla son:

- DEPARTMENT\_ID - NUMBER
- PORCENTAJE\_AUMENTO - NUMBER(5,2)

El proceso debe ir mostrando en pantalla el nombre del departamento procesado y el resultado de ejecutar el procedure PR\_AUMENTO\_SUELDO\_DEPTO. Por ejemplo:

DEPARTAMENTO: Administration  
RESULTADO: OK

DEPARTAMENTO: Marketing  
RESULTADO: OK

9. Elaborar una sentencia que permita modificar la tabla EMPLOYEES agregándole dos nuevos campos:

- USUARIO - VARCHAR2(30)
- FECHA\_MODIFICACION - DATE

Crear un TRIGGER en la tabla EMPLOYEES que grabe, ante un insert o update de algún registro, en los campos USUARIO y FECHA\_MODIFICACION los valores correspondientes al usuario que realiza la acción y la fecha de ocurrencia.

10. Elaborar una sentencia SQL que permita crear la tabla EMPLOYEES\_AUDIT con los siguientes campos:

- EMPLOYEE\_ID – NUMBER(6)
- SALARY\_OLD – NUMBER(8,2)
- SALARY\_NEW – NUMBER(8,2)
- FECHA\_MODIFICACION – DATE

Crear un TRIGGER en la tabla EMPLOYEES que grabe, ante una modificación de salario de algún empleado, los valores de salario actual y nuevo en la tabla EMPLOYEES\_AUDIT (campos SALARY\_OLD y SALARY\_NEW) junto con la fecha de modificación correspondiente.

## **ANEXO I – Habilitar Esquema HR**

Para habilitar el esquema HR incluido en la instalación, abrir *SQL Developer* e ingresar a la base de datos con el usuario SYS (rol SYSDBA y contraseña establecida durante la instalación) y ejecutar las siguientes sentencias:

```
ALTER USER HR IDENTIFIED BY HR ACCOUNT UNLOCK;  
GRANT CONNECT, RESOURCE TO HR;
```

Luego ingresar con el usuario HR (rol "valor por defecto" y contraseña HR)

Nombre de Conexión XE\_HR

Usuario HR

Contraseña ••

☒ Guardar Contraseña ☒ Color de Conexión

**Oracle**

Tipo de Conexión Básico Rol valor por defecto

Nombre del Host LOCALHOST

Puerto 1521

☒ SID XE

☐ Nombre del Servicio

☐ Autenticación del Sistema Operativo ☐ Autenticación Kerberos Avanzadas...

ANEXO II – Modelo de Tablas de Esquema HR

