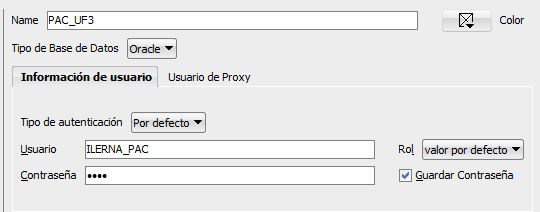


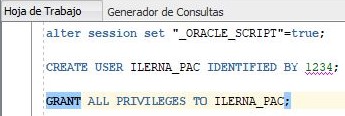
**David Nuñez Merino**

**2S2022 – DAW**

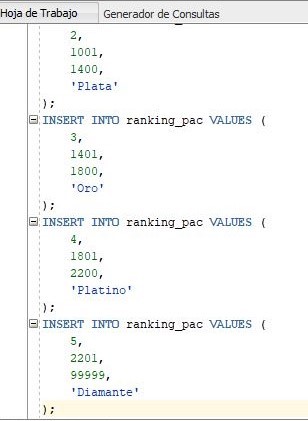
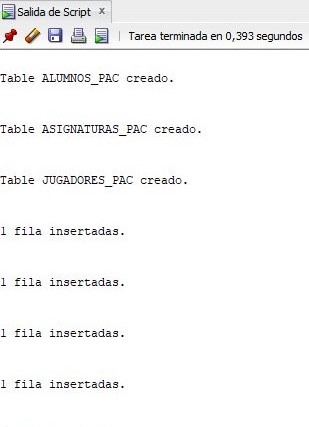
**CONFIGURACIÓN INICIAL**

Antes de continuar con las actividades creamos una nueva conexión llamada “PAC\_UF3” usando el usuario administrador y después creo un nuevo usuario “ILERNA\_PAC” y contraseña “1234” y le asigno todos los privilegios:



****

A continuación, ejecutamos el script “PAC\_Desarrollo\_UF3\_Configuracion Inicial.sql “ para tener las tablas y registros que necesitamos para las practicas.

**EJERCICIOS**

**1) GESTIÓN DE USUARIOS Y TABLAS**

**1.** Crear un usuario llamado “GESTOR” y contraseña “1234”:

● Asígnale los privilegios necesarios para que aparte de conectare. le permitan modificar, añadir o eliminar campos (No los registros) de las tablas Alumnos y Asignaturas.

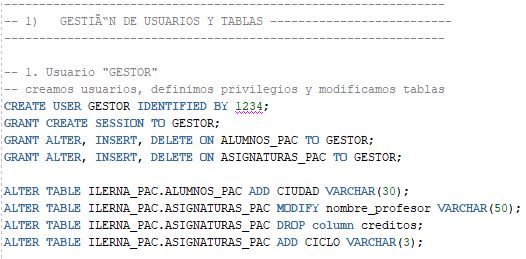
● Conéctate con el usuario “GESTOR” y realiza lo siguiente con este usuario:

o En la tabla alumnos añade un nuevo campo “CIUDAD” - VARCHAR(30)

o En la tabla asignaturas modifica el campo “NOMBRE\_PROFESOR” a VARCHAR(50)

o En la tabla asignaturas elimina el campo “CREDITOS”

o En la tabla asignaturas añade un nuevo campo llamado “CICLO” - VARCHAR (3)



2. Crea un rol llamado “ROL\_DIRECTOR” y un nuevo usuario llamado “DIRECTOR” y contraseña “1234”

● Al nuevo rol añádele los privilegios necesarios para que aparte de conectare, permitan seleccionar, insertar y modificar registros de las tablas alumnos y asignaturas.

● Asigna el nuevo rol “ROL\_DIRECTOR” al usuario “DIRECTOR”

● Conéctate con el usuario “DIRECTOR” y realiza lo siguiente:

o Insertar un registro en la tabla Alumnos con tus datos

▪ Id\_alumno = (Las 2 primeras letras de tu nombre y tus apellidos junta)

● Ejemplo: Juan Soria Morales JUSOMO

o Insertar un registro en la tabla Asignaturas con los datos de esta asignatura

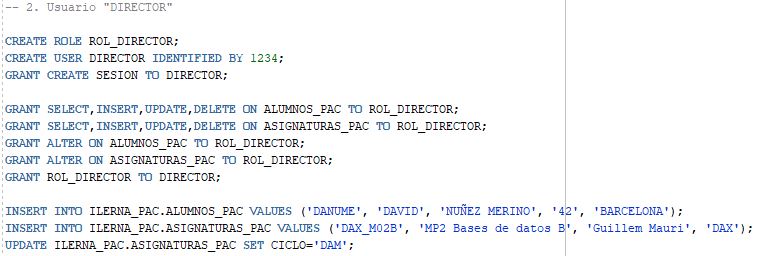
▪ Id\_asignatura = ‘DAX\_M02B’ ▪ Nombre\_asignatura = ‘MP2. Bases de datos B’

▪ Nombre\_profesor = (Nombre y apellidos del profesor actual)

▪ Ciclo = ‘DAX’

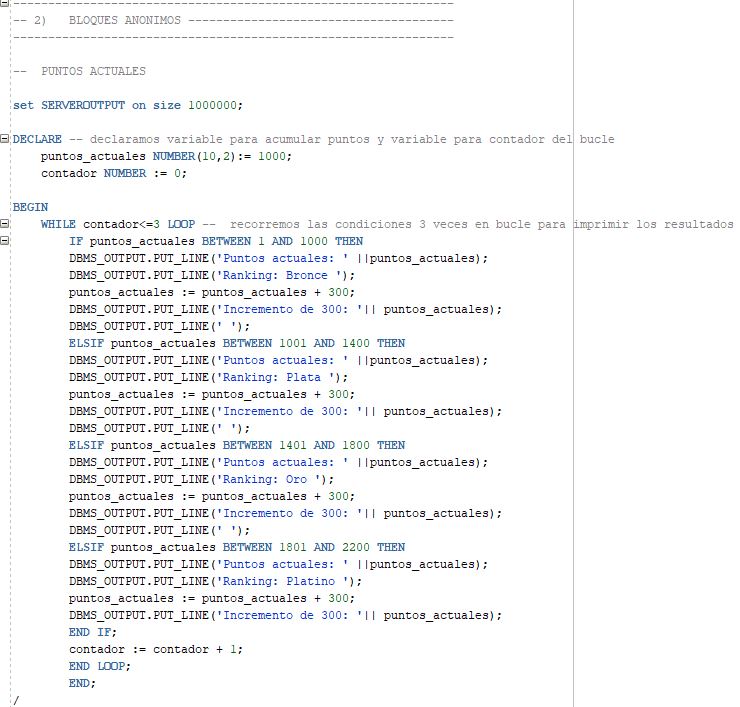
o Modificar el Ciclo del registro de la tabla Asignaturas insertado anteriormente

▪ Ciclo = (Poner DAM o DAW según te corresponde)



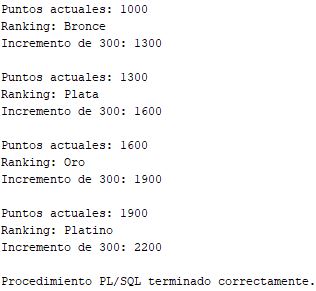
**2) BLOQUES ANONIMOS**

1. Crea un bloque anónimo donde se declare una variable constante llamada “puntos\_actuales” del tipo NUMBER(10, 2) inicializa la variable con un valor de puntos como el de los jugadores creados en la configuración inicial. Muestra en qué ranking estaría ese jugador y una vez hecho incremente esos puntos en 300, realiza esta acción 3 veces más.



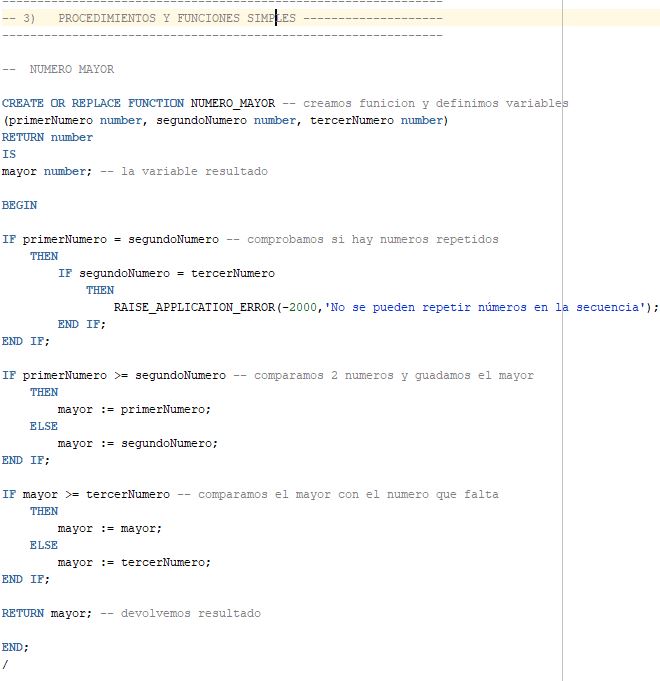
Para resolver el ejercicio, declaramos dos variables, un contador para contar las veces que se ejecuta el bucle y otra puntos actuales que iremos modificando en cada iteración.

El resultado es el siguiente:



**3) PROCEDIMIENTOS Y FUNCIONES SIMPLES**

1. Crea una función llamada “NUMERO\_MAYOR” que devuelva el mayor de 3 números pasados como parámetros, en casos de que se repita algún número, se ha de gestionar una excepción de error, diciendo “No se pueden repetir números en la secuencia”.

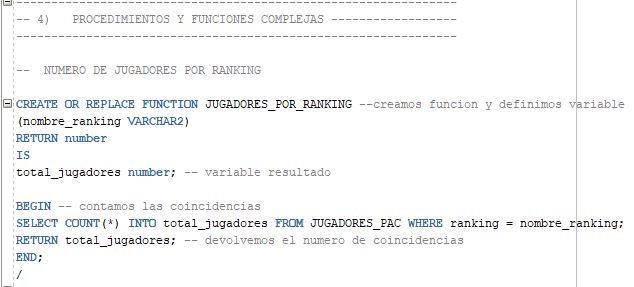


Para realizar el ejercicio definimos la función NUMERO\_MAYOR con tres variables que son los tres números que queremos comparar y una variable llamada “mayor” que era la que guardará el número mayor.

En primer lugar, realizamos unas comparaciones para comprobar que no tenemos ningún número repetido y después comparamos los dos primeros números y guardamos el mayor, a continuación, comparamos el almacenado con el último número y determinamos así cual es el mayor de los tres.

**4) PROCEDIMIENTOS Y FUNCIONES COMPLEJAS**

1. Crea una función llamada “JUGADORES\_POR\_RANKING” qué, dado un nombre de ranking pasado por parámetro, devuelva el total de jugadores que se encuentran en ese mismo ranking.

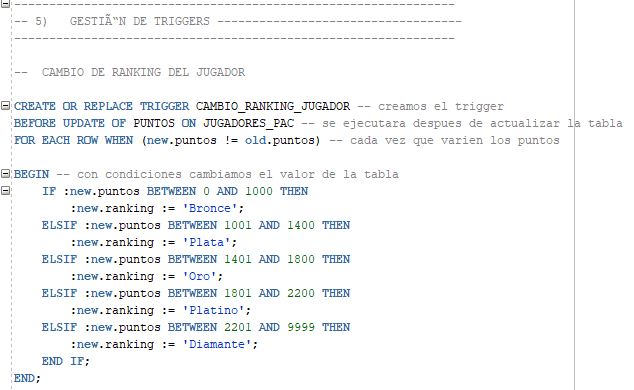


En este ejercicio, creamos una función llamada JUGADORES\_POR\_RANKING con la variable nombre\_ranking y total\_jugadores que guardara el resultado.

Para ello buscamos las coincidencias del ranking que definimos en la variable nombre\_ranking en la tabla JUGADORES\_PAC y las guardamos en la variable total\_jugadores.

**5) GESTIÓN DE TRIGGERS**

1. Crea un trigger llamado “CAMBIO\_RANKING\_JUGADOR” que, al modificar la puntuación de un jugador, si la modificación implica un cambio en el ranking, se cambie automáticamente el ranking del jugador, ejemplo, pasar de tener 1390 puntos a 1410 puntos, el ranking del jugador cambiará automáticamente del que tenía en ese momento, ‘Plata’, al siguiente disponible, ‘Oro’.

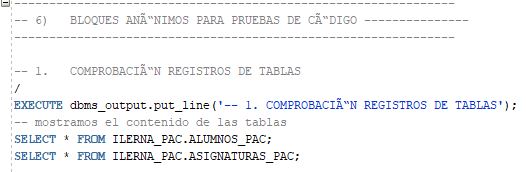


Crearemos el trigger CAMBIO\_RANKING\_JUGADOR que se ejecutara cuando se modifique el valor PUNTOS de cualquier fila de la tabla JUGADORES\_PAC. Esto ocurrirá cuando el valor nuevo de puntos sea diferente del valor antiguo (antes de la modificación).

Cuando esto ocurra, comprobaremos el rango en el que se encuentra el nuevo valor de puntos y según cuál de las condiciones se cumplan le asignaremos un nuevo ranking.

**6) BLOQUES ANÓNIMOS PARA PRUEBAS**

**1. COMPROBACIÓN GESTIÓN USUARIOS Y TABLAS** Crea un bloque anónimo que muestre el registro de la tabla “alumnos\_pac” y el de la tabla “asignaturas\_pac”.



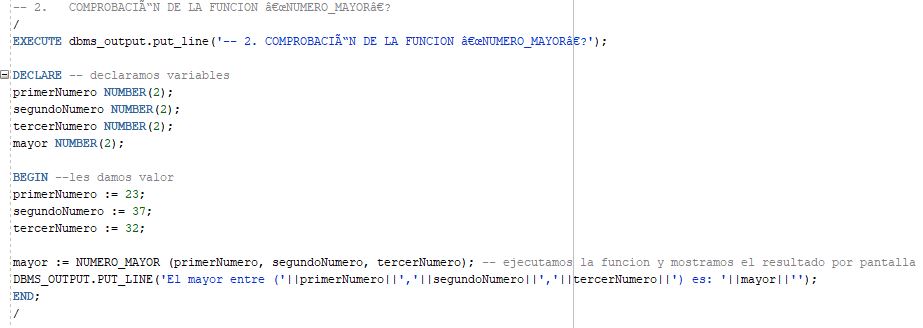
Para mostrar el registro ejecutaremos un SELECT de todos los registros de la tabla ALUMNOS\_PAC Y ASIGNATURAS\_PAC.

**2. COMPROBACIÓN DE LA FUNCION “NUMERO\_MAYOR”**

Crea un bloque anónimo que use la función “NUMERO\_MAYOR”.

● Números de la prueba: 23, 37, 32

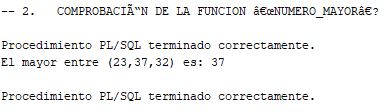
● Salida por pantalla: “El mayor entre (23, 37, 32) es: 37”



Para resolver este ejercicio declaramos las tres variables de los números que usa la función NUMERO\_MAYOR y un más para almacenar el resultado.

Se definen las variables con los valores 23, 37 y 32. La variable Mayor será el resultado de la ejecución de la función NUMERO\_MAYOR con el valor de las tres variables.

En la salida de texto mostraremos los tres valores y el resultado alojado en la variable mayor que es 37.

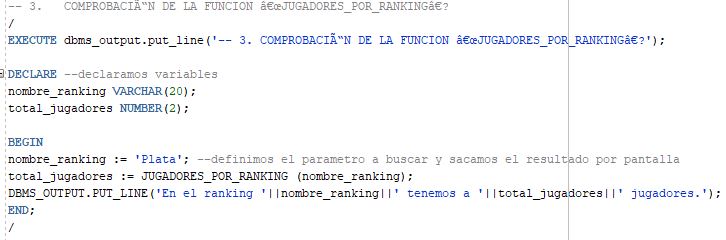


**3. COMPROBACIÓN DE LA FUNCION “JUGADORES\_POR\_RANKING”**

Crea un bloque anónimo que use la función “JUGADORES\_POR\_RANKING”.

● Nombre del ranking: Plata

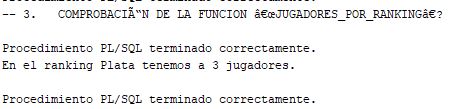
● Salida por pantalla: “En el ranking Plata, tenemos a 2 jugadores.”



En este ejercicio declaramos dos variables, nombre\_ranking donde definimos el ranking a buscar y total\_jugadores donde guardaremos el número de coincidencias.

Definimos nombre\_ranking con el valor “Plata” y total\_jugadores como el resultado de la ejecución de la función JUGADORES\_POR\_RANKING y la variable nombre\_ranking.

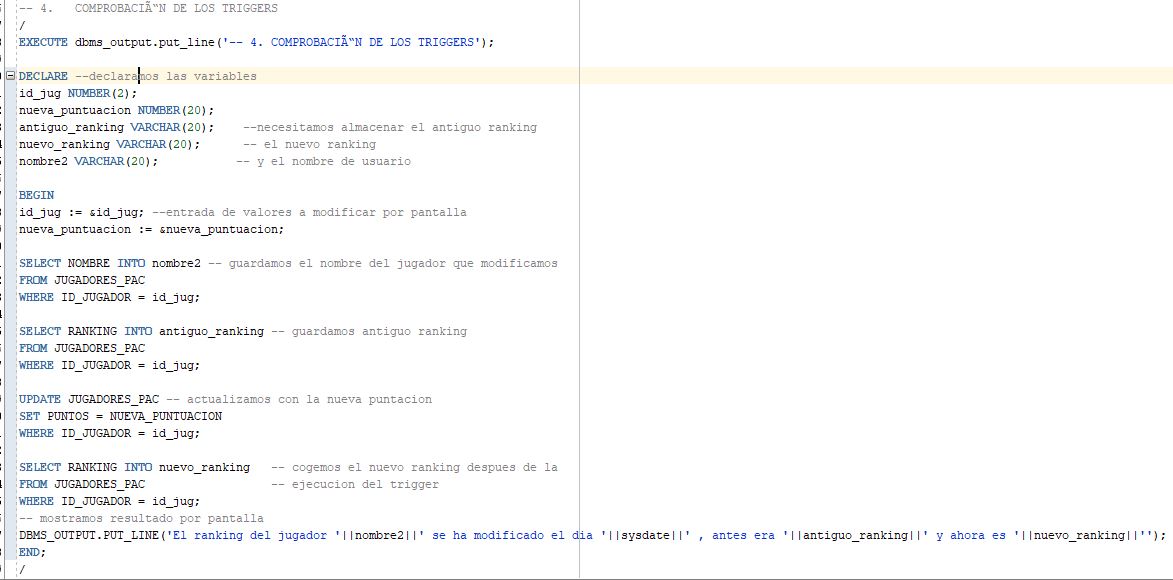
Mostraremos por pantalla el texto y el nombre del ranking y el número de coincidencias que se han encontrado.



**4. COMPROBACIÓN DE LOS TRIGGERS “CAMBIO\_RANKING\_JUGADOR”**

Crea un bloque anónimo que pida por pantalla un id de un jugador y una nueva puntuación. Ha de actualizar el ranking del jugador si existe y comprobaremos que el trigger ha funcionado.

● Salida por pantalla: “El ranking del jugador (Nombre jugador) se ha modificado el día (Última Fecha y hora de modificación), antes era (Ranking Anterior) y ahora es (Nuevo Ranking)



Para comprobar los triggers primero tenemos que definir cinco variables:

* Id\_jug donde guardaremos la id introducida por pantalla.
* nueva\_puntuacion donde guardaremos la puntuación introducida por pantalla.
* Antiguo\_ranking donde almacenaremos el ranking antes de modificarlo con el trigger.
* Nuevo\_ranking donde almacenaremos el ranking después de modificarlo con el trigger.
* Nombre2 donde guardaremos el nombre correspondiente al id introducido.

Primero pediremos el id del jugador y la nueva puntuación por pantalla.

A continuación, buscaremos el nombre que corresponde a esa id en la tabla JUGADORES\_PAC y lo almacenaremos en la variable nombre2.

Haremos lo mismo con el ranking que corresponde a esa id en la variable antiguo\_ranking.

A continuación, realizaremos un UPDATE de la tabla JUGADORES\_PAC y cambiaremos el valor de PUNTOS por la variable NUEVA\_PUNTUACION del id que hemos introducido por pantalla.

Después guardaremos el nuevo ranking modificado por el trigger que hemos definido en la variable nuevo\_ranking.

Por pantalla mostraremos el nombre que hemos guardado en la variable nombre2 la fecha y hora de modificación, el ranking guardado en antiguo\_ranking y el ranking almacenado en nuevo\_ranking.

Para que la fecha se muestre bien hemos introducido en la cabecera del script el siguiente codigo que nos dará la salida en día-mes-año y la hora con minutos y segundos que por defecto no nos muestra el sistema:

alter SESSION set NLS\_DATE\_FORMAT = 'DD-MM-YYYY HH24:MI:SS';

También en la cabecera del script introducimos el modificador set verify off para que no se nos muestren por la salida de script el código del bloque anónimo.

El resultado es este:

