

## PAC DESARROLLO

CFGS Desarrollo de Aplicaciones Multiplataforma

CFGS Desarrollo de Aplicaciones Web

# Módulo 2B: Bases de Datos

UF3. Lenguajes SQL: DCL y extensión procedimental



## Índice de la PAC

INFORMACIÓN IMPORTANTE	3
REQUISITOS DEL TRABAJO	3
CRITERIOS IMPORTANTES	3
EJERCICIOS	4
INTRODUCCIÓN	4
CONFIGURACIÓN INICIAL	5
EJERCICIOS	7
1) GESTIÓN DE USUARIOS Y TABLAS	7
2) BLOQUES ANONIMOS	7
3) PROCEDIMIENTOS Y FUNCIONES SIMPLES	7
4) PROCEDIMIENTOS Y FUNCIONES COMPLEJAS	8
5) GESTIÓN DE TRIGGERS	8

# INFORMACIÓN IMPORTANTE

Para la correcta realización de la PAC el alumno deberá consultar los contenidos del material didáctico

## REQUISITOS DEL TRABAJO

- Las PACs de desarrollo se enviarán solo a través de la plataforma en los plazos de entrega establecidos en la guía didáctica. En caso de no cumplir el plazo, NO se podrá enviar de forma posterior.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado esta información.
- No se aceptarán copias literales de Internet. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas debe ser de elaboración propia.
- Las respuestas deben estar debidamente argumentadas. No se admiten respuestas escuetas.
- Es responsabilidad del alumno comprobar que el archivo subido en la plataforma es el correcto, ya que en ningún caso el profesor revisará el documento antes del periodo de corrección.
- El día y hora máximo para entregar una PAC de desarrollo es el día especificado en la guía didáctica.
- Si no se entrega una PAC de desarrollo, la calificación equivaldrá a un 0.
- Si se detecta que dos alumnos presentan dos PAC iguales la nota será 0 PARA LOS DOS ALUMNOS.

## CRITERIOS IMPORTANTES

- Las PAC disponen de una calificación numérica que oscila del 0 al 10. Respecto a la calificación de cada PAC de desarrollo, el profesor podrá disminuir hasta 1 punto la nota obtenida en caso de que la PAC contenga errores ortográficos y/o su presentación no se adecúe a lo establecido por el profesor.
- Debéis redactar las respuestas en color azul oscuro o negro.
- Podéis utilizar la opción de negrita y subrayado para resaltar palabras clave, enunciados, etc., NUNCA para responder la totalidad de la actividad. No se podrá utilizar la función de resaltado.
- La actividad debe ser redactada en minúsculas siguiendo las normas ortográficas básicas.

# EJERCICIOS

## INTRODUCCIÓN

Para la realización de estos ejercicios utilizaremos el material didáctico, así como cualquier otro material que se aporte en la plataforma, por supuesto, si el alumno lo considera necesario podrá acudir a fuentes externas de información SIEMPRE añadiendo las fuentes consultadas en la bibliografía/webgrafía.

Estos ejercicios hay que realizarlos con el gestor de base de datos de Oracle utilizado en las videoclases.

La entrega será en un archivo comprimido que ha de contener 2 archivos:

- Un archivo “.sql” con el código de los ejercicios propuestos
- Un archivo “.pdf” que contendrá pantallazos y explicación de la resolución de ejercicios Y LA DOCUMENTACIÓN DE SU EJECUCIÓN:
  - **Por ejemplo:** Como se ha realizado el código, cosas a comentar y , paso a paso, la ejecución del código para comprobar su correcto funcionamiento.
  - **Los ejercicios que no se comenten o no se documente su ejecución NO SE VAN A CORREGIR NINGUNA DE LAS DOS PARTES DE ESE APARTADO, TANTO LA DE CÓDIGO COMO LA DE DOCUMENTACIÓN, AUNQUE UNA DE LAS DOS ESTÉ HECHA!**
- El nombre del archivo tanto ZIP/ PDF / SQL ha de seguir la siguiente nomenclatura:
  - DAX\_M02B\_UF3\_PAC\_Desarrollo\_nombre\_apellidos.zip
    - DAX\_M02B\_UF3\_PAC\_Desarrollo\_nombre\_apellidos.pdf
    - DAX\_M02B\_UF3\_PAC\_Desarrollo\_nombre\_apellidos.sql
- Todo aquel ejercicio que no siga dichas pautas será penalizado.

Aspectos ¡**IMPORTANTES**! sobre los ejercicios:

- El Script ha de poder ejecutarse de forma completa sin errores
- Para ello entre cada bloque, procedimiento o función se ha de poner el símbolo “/”
- Realizar los ejercicios en los bloques delimitados y comentar el código escrito.
  - No es necesario comentar línea por línea, vale con que se hace en general.

## CONFIGURACIÓN INICIAL

### Pasos previos a la realización de las actividades:

- Crea una nueva conexión llamada “PAC\_UF3” usando el usuario administrador
- Crea un nuevo usuario “ILERNA\_PAC” y contraseña “1234” y asignar todos los privilegios.

```
alter session set "_ORACLE_SCRIPT"=true;

CREATE USER ILERNA_PAC IDENTIFIED BY 1234;

GRANT ALL PRIVILEGES TO ILERNA_PAC;
```

- Conéctate con el usuario “ILERNA\_PAC” a la conexión de base de datos “PAC\_UF3”



- Todos los ejercicios se realizarán con el usuario “ILERNA\_PAC” en la conexión “PAC\_UF3”
- Ejecutar el script **PAC\_Desarrollo\_UF3\_Configuracion Inicial.sql** para tener las tablas y registros
  - Tablas que se van a crear con el script:
    - **alumnos\_pac, asignaturas\_pac, jugadores\_pac, ranking\_pac**

## EJERCICIOS

### 1) GESTIÓN DE USUARIOS Y TABLAS

*\*Recuerda que para gestionar tablas creadas por un usuario distinto al que se usa hay que poner **nombre\_usuario.nombre\_tabla** ya que sino no la va a encontrar*

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:
    - En la tabla **alumnos** añade un nuevo campo “CIUDAD” - VARCHAR(30)
    - En la tabla **asignaturas** modifica el campo “NOMBRE\_PROFESOR” a VARCHAR(50)
    - En la tabla **asignaturas** elimina el campo “CREDITOS”
    - 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:
  - **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*
  - **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’
  - **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.
  - Por ejemplo, si el valor de la variable “puntos\_actuales” es 1000. La salida por pantalla será:
    - Puntos actuales: 1000
    - Ranking: Bronze
    - Incremento de 300: 1300
  
    - Puntos actuales: 1300
    - Ranking: Plata
    - Incremento de 300: 1600
  
    - Puntos actuales: 1600
    - Ranking: Oro
    - Incremento de 300: 1900
  
    - Puntos actuales: 1900
    - Ranking: Platino
    - Incremento de 300: 2200

### 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”**.

### 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.

### 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’.

## 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”.

### 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”

### 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.”

### 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**)”