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

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.
- 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.

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 en el código o el formato no sea claro.

INTRODUCCIÓN

ASPECTOS IMPORTANTES

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
    - **Por ejemplo**: Como se ha realizado el código o los pasos en la ejecución del código para comprobar su correcto funcionamiento.
    - **Los ejercicios que no se comenten recibirán menor puntuación**
- El nombre del archivo tanto ZIP/ PDF / SQL ha de seguir la siguiente nomenclatura:
    - DAX_M02B_UF3_PAC_Desarrollo_*nombre_apellidos*.zip o .rar
        - 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:

- Crear una nueva conexión llamada "**PACDES_UF3**" usando el usuario administrador
- Ejecutar el script **PAC_Desarrollo_UF3_Configuracion_Inicial.sql**
    - Ejecuta el código del script para tener el usuario, tablas y registros de la PAC
    - Se crea un nuevo usuario "**ILERNA_PAC**" y contraseña "**i1234**"
        - Se le asignan todos los privilegios a este usuario
        - Los ejercicios se realizarán con el usuario "**ILERNA_PAC**" en la conexión "**PACDES_UF3**"
    - Se conecta con el usuario "**ILERNA_PAC**" (CONN Ilerna_pac / i1234;)
    - Tablas que se van a crear con el script:
        - **ASIGNATURAS_PAC**
            - Para realizar la parte de gestión de usuarios
            - Un nuevo usuario para poder modificar la estructura y los registros de la tabla
        - **RANKING_PAC**
            - Tabla de ranking con id, nombre, puntos mínimos y puntos máximos.
            - Se crean 5 nombres diferentes de niveles de Ranking
        - **JUGADORES_PAC**
            - Tabla de jugadores con id, nombre, apellidos y puntos
            - Se insertan 10 Jugadores nuevos con sus puntos

EJERCICIOS

GESTIÓN DE USUARIOS Y ROLES

- ✓ **Crea un rol llamado "ROL_GESTOR" y un nuevo usuario llamado "GESTOR" y** contraseña **"g1234"**
  - **"ROL_GESTOR"**, con privilegios de conexión, modificar la estructura de la tabla, seleccionar, insertar y modificar registros de la tabla ASIGNATURAS_PAC y asignarlo al usuario **"GESTOR**"
  - Conéctate con el usuario "**GESTOR**" (CONN gestor / g1234;) y realiza lo siguiente:
    - Modificar la Estructura de la tabla
      - Eliminar el campo "**CREDITOS**" y Añadir campo llamado "**CICLO**" de tipo VARCHAR (3)
    - 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 del profesor actual),* **CICLO** = 'DAX'
    - Modificar el dato de "Ciclo" del registro de la tabla Asignaturas insertado anteriormente
      - Ciclo = *(Poner DAM o DAW según te corresponde)*

Para los siguientes ejercicios nos conectamos de nuevo al usuario ILERNA_PAC (CONN Ilerna_pac / i1234;)

PROCEDIMIENTO

- ✓ Crea un **procedimiento** llamado "**RANKING_JUGADOR**" qué, dado un id de jugador y un número de puntos que puede ser positivo o negativo, nos devuelva el nombre y apellidos del jugador junto con el nuevo total de puntos que tendrá el jugador sumando o restando el número de puntos al que tiene actualmente y también devuelva el nombre del nivel de ranking que le correspondería con la nueva puntuación.
  a. No actualizar la tabla, solo devolver nombre jugador, nuevo total de puntos y nombre ranking

FUNCION

- ✓ 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
  a. **Ejemplo**: Si pasamos por parámetro "**BRONCE"** ha de devolver un **2**

TRIGGER

- ✓ Crea un **TRIGGER** llamado "**ACTUALIZA_RANKING_JUGADOR**", que al INSERTAR o ACTUALIZAR un jugador en la tabla JUGADORES_PAC, se ha de notificar la fecha actual, el nombre, apellidos, nombre de ranking, variación de puntos y nueva puntuación total del jugador. Puedes usar el procedimiento de "RANKING_ JUGADOR" para determinar el nuevo ranking. La salida ha de ser como lo siguiente ejemplos:
  a. "A fecha de <**HOY**>. El jugador <**NOMBRE**> <**APELLIDOS**> está en el nivel <**NOMBRE_RANKING**> con un total de <**PUNTOS**> puntos"

BLOQUES ANÓNIMOS PARA PROBAR EJERCICIOS

COMPROBACIÓN GESTIÓN USUARIOS Y ROLES

Crea un bloque anónimo que muestre por pantalla los datos de la asignatura de la tabla "**ASIGNATURAS_PAC**" a partir del ID_ASIGNATURA = 'DAX_M02B'. Recoger excepción si no existe el valor

- **Salida por pantalla: "**El profesor de <**NOMBRE_ASIGNATURA**> se llama <**NOMBRE_PROFESOR**>"

COMPROBACIÓN DE TRIGGER "ACTUALIZA_RANKING_JUGADOR"

Crea un bloque anónimo donde primero se inserte un nuevo registro a la tabla JUGADORES con estos datos:

- id_jugador = 11
- nombre = *(Nombre alumno)*
- apellidos = *(Apellidos Alumno)*
- puntos = 0

Seguido de la inserción, se han de actualizar los puntos del jugador 11 a un número de puntos introducidos por comando. Antes de hacer el UPDATE, comprobar errores de si el jugador existe, o si la nueva puntuación es negativa o mayor a 9999, en estos casos gestionar las excepciones con mensajes de error.

- **Salidas por pantalla:** El texto generado por el Trigger tanto al INSERTAR como al ACTUALIZAR
- **Salida por pantalla si error:** Se ha de ver un texto relativo al error generado

COMPROBACIÓN DEL PROCEDIMIENTO "RANKING_JUGADOR"

Crea un bloque anónimo que use el procedimiento "**RANKING_JUGADOR**". Introduciendo los valores de ID_JUGADOR y PUNTOS EXTRA por comando, si no existe el jugador se ha de dar una excepción, en caso de existir, ha de generar el siguiente texto de salida por pantalla.

- **Salida por pantalla: "**El jugador <**NOMBRE**> <**APELLIDOS**>, tendrá <**PUNTOS**> puntos y pasa al nivel de ranking <**NOMBRE RANKING**>"

COMPROBACIÓN DE LA FUNCION "JUGADORES_POR_RANKING"

Crea un bloque anónimo que use la función "**JUGADORES_POR_RANKING**". Introduciremos el valor de un nombre de ranking por comando. Si no existe el nombre de ranking hemos de gestionar el error.

- **Salida por pantalla:** "En el ranking <**NOMBRE RANKING**>", tenemos a <**TOTAL**> jugadores."