

CFGS

DESARROLLO DE APLICACIONES WEB

M02B: Gestión de base de datos

Unidad Formativa 3: Lenguajes SQL: DCL y extensión procedimental.

UF3_PAC09_ENUNCIADO



UF3. [PAC 09]. Lenguajes SQL: DCL y extensión procedimental.

INFORMACIÓN

Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y consultar de internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del libro digital para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Se valorará la presentación de vuestro trabajo hasta con un punto de la nota final.

EJERCICIOS

Para la realización de estos ejercicios utilizaremos el material didáctico en formato PDF, 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, hay que entregar un archivo comprimido con el contenido de los archivos .sql y el PDF con las pertinentes capturas de pantalla en cada ejercicio (De la resolución del ejercicio).

EJERCICIO 1

Para poder realizar la primera PAC hay que tener creadas dos tablas; una llamada alumnos y otra llamada profesores. No importan tanto los campos específicos de cada una de estas tablas porque en esta PAC no se insertan o modifican datos en dichas tablas.

Únicamente se dan permisos sobre las tablas a los usuarios que se crean. Aun así, a modo de ejemplo, podéis considerar crear las siguientes dos tablas.

1. Crea un usuario nuevo con `create user` en local del nombre “Miguel” que tenga permisos de solo conexión.
2. Crea un usuario local que se llame “Marta” usando la sintaxis `grant` con permisos de conexión y pueda editar y eliminar tablas.
3. Concede permisos de `select` a “Miguel” en la tabla alumnos.
4. Edita y elimina distintos datos con el usuario “Marta” en la tabla de alumnos.

EJERCICIO 2

Para el desarrollo de esta PAC necesitarás la base de datos Empresa cuyo script encontraréis junto a este documento en el campus virtual.

Debéis programar en PL/SQL:

1. Un procedimiento para mostrar el año actual.
2. Un procedimiento que sume uno a una variable numérica cada vez que se ejecute.
3. Un procedimiento al que se le pasen dos cadenas como parámetros y las muestre concatenadas y en mayúsculas.
4. Bloque anónimo que pida un código de empleado y muestre su salario actual, después lo disminuya en un tercio y muestre el nuevo salario.
5. Una función para mostrar el día de la semana según un valor de entrada numérico, para domingo, lunes, etc. utilizando la estructura "IF/IF ELSE".
6. La misma función que muestre el día de la semana según un valor de entrada numérico, para domingo, lunes, etc. Pero esta vez utilizando la estructura condicional "CASE".
7. Una función que devuelva el mayor de tres números pasados como parámetros, ...
8. Un procedimiento que muestre la suma de los primeros n números enteros, siendo n un parámetro de entrada.
9. Una función que determine si un número es primo devolviendo 0 ó 1.
10. Usando la función anterior crear otra que calcule la suma de los primeros m números primos empezando en el 1.

EJERCICIO 3

Para el desarrollo de este ejercicio necesitarás la base de datos Empresa cuyo script encontraréis junto a este documento en el campus virtual.

1. Programa anónimo que visualice el nombre y la localidad de todos los departamentos.
2. Programa que muestre los apellidos de los empleados que pertenecen al departamento de VENTAS. Hay que numerar cada línea secuencialmente.
3. Crea un procedimiento que visualice por pantalla los apellidos de los empleados del departamento que se recibe como argumento (se recibe el código del departamento). Haz dos versiones, una usando un " for " y otra sin usarlo. Ambas versiones tienen que informar en el caso de que no se muestre ningún empleado (bien porque el departamento pedido no exista o que no tenga empleados).
4. Crear una función que sirva para calcular cuántos empleados del departamento que pasamos por parámetro cobran comisión (significa comisión mayor que 0). Pasamos por parámetro el código del departamento. Hazlo sin utilizar la función de agregación COUNT (). Si el departamento no existe la función debe devolver el valor NULL. Si el departamento no tiene ningún empleado que cobre comisión la función debe devolver un 0.

EJERCICIO 4

1. Se necesita llevar un histórico de los cambios realizados en las condiciones contractuales de los empleados que están registrados en la tabla EMP. Para ello primero tienes que crear una tabla de nombre "auditaemple" con tres campos, uno de nombre "id_cambio" number (5) que será la clave primaria, otro de nombre "descripcion_cambio" de tipo VARCHAR2 (100) y otro de nombre "fecha_cambio" de tipo DATE. Crea un trigger que se llame "auditasueldo" y sirva para que cada vez que se cambie el salario de un empleado se inserten en la tabla "auditaemple" un registro con el mensaje "El salario del empleado codiempleado antes era de sueldoantiguo y ahora será sueldonuevo" (donde codiempleado es el código del empleado, sueldoantiguo es el sueldo antes del cambio y sueldonuevo es el sueldo después del cambio), la fecha actual y de id_cambio que será el valor más alto que hubiese en la tabla incrementado en 1.
2. Haz otro trigger de nombre "auditaemple" que sea una modificación del anterior para añadir que si se da de alta un nuevo empleado se registre en la tabla "auditaemple" el mensaje "Nuevo empleado con código emp_no" (donde emp_no es el valor del campo emp_no para ese registro) con fecha de hoy.
3. Haz un nuevo trigger de nombre "auditaemple2" para que cuando se intente modificar el sueldo de un empleado, sólo se registre este cambio en la tabla "auditaemple" si la modificación consiste en subir el salario al empleado más de un 10%. El mensaje registrado podría ser "El salario del empleado emp_no antes era de sueldoantiguo y ahora será sueldonuevo".
4. En la tabla DETALLE del archivo EMPRESA, el campo CANTIDAD hace referencia al número de unidades que se venden, mientras que PRECIO_VENDA hace referencia al precio de venta de cada una de esas unidades y el campo IMPORTE es el producto PRECIO_VENDA*CANTIDAD.
Pues bien, crear un trigger de nombre "verifica_unidades" que sirva para:
 - a) Que en ninguna fila de la tabla DETALLE se pueda registrar un valor del campo CANTIDAD mayor que 999 unidades. La forma de que un trigger pueda evitar que se lleve a cabo la operación es levantar una excepción y no tratarla. ¿Hace falta algún procedimiento concreto de SQL para esto último? En caso afirmativo, ¿cuál es ese procedimiento predeterminado de SQL para lanzar este tipo de errores y que no se lleguen a ejecutar las operaciones de insertar o actualizar?
 - b) En caso de que se pidan hasta 999 unidades y se admita el cambio, daros cuenta de que el valor del campo IMPORTE debe cambiar también porque es el producto de PRECIO_VENDA (es decir, precio de venta de cada unidad) por CANTIDAD (es decir, el número de UNIDADES). Así que, en este caso, el trigger también tiene que actualizar el campo IMPORTE.

ENTREGA:

En el campus, tenéis que colgar un archivo comprimido, el cual contendrá una única carpeta. En el interior de ésta un archivo SQL con todos los ejercicios de la PAC (poner comentarios como enunciado de las actividades) y un documento de texto (PDF), con las argumentaciones pertinentes y, en caso oportuno, los “printscreen” necesarios (Imprimir pantalla) o archivos complementarios.

El nombre del archivo y de la carpeta tiene que seguir el siguiente formato:

DAX_M02B_UF03_PAC09_ApellidoNombre

dónde ApellidoNombre se refiere a vuestro primer apellido y vuestro nombre. Todo aquel ejercicio que no siga dichas pautas será penalizado.

Siempre tenéis que argumentar las respuestas más teóricas.

No realicéis un copiar y pegar de información extraída de Internet, tenéis que explicarlo todo con vuestras palabras.

¡Buen trabajo!

