

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
ASPECTOS IMPORTANTES.....	3
INTRODUCCIÓN	4
ENUNCIADO GENERAL.....	4
FORMA DE ENTREGA	4
CONFIGURACIÓN INICIAL	5
EJERCICIOS.....	6
EJERCICIO 1) REPASO SQL (TABLAS Y VISTAS)	6
EJERCICIO 2) PROCEDIMIENTO.....	7
EJERCICIO 3) FUNCION	8
EJERCICIO 4) TRIGGER	9
EJERCICIO 5) BLOQUES ANÓNIMOS	10

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.

ASPECTOS IMPORTANTES

- Para la realización de estos ejercicios utilizaremos el material didáctico y los ejercicios realizados en las videotutorías donde se han realizado varios ejemplos de código que sirven de gran ayuda para realizar la PAC.
- Los ejercicios hay que realizarlos con el gestor de base de datos de Oracle utilizado en las videotutorías o en su defecto en algún editor de Oracle como el ORACLE LIVE.
- Las PAC disponen de una calificación numérica que oscila del 0 al 10.
- Si no se entrega la PAC queda con un No Presentado.
- Se va a evaluar si los ejercicios realizan la función requerida y por otra parte también se va a evaluar la estructura y presentación del código.

INTRODUCCIÓN

ENUNCIADO GENERAL

Un alumno del ciclo de DAM se ha propuesto poder gestionar las notas de su expediente académico en una base de datos a modo de calculadora, con la idea de que, introduciendo los datos de las diferentes unidades formativas de las asignaturas, pueda saber la nota final de cada asignatura y su nota media de final del ciclo

En esta PAC se van a realizar diferentes ejercicios para que la calculadora de notas sea útil y eficiente.

FORMA DE ENTREGA

La entrega será en un único archivo con el código de los ejercicios propuestos, para ello se tendrá una plantilla de código que servirá para completarla con cada uno de los ejercicios:

- Archivo “.sql” con el código de los ejercicios propuestos
- El nombre del archivo seguir la siguiente nomenclatura:
 - DAX_M02B_UF3_PAC_*nombre_apellidos*.sql
- Todo aquel ejercicio que no siga dichas pautas será penalizado.
- El código con la solución ha de poder ejecutarse de forma completa sin errores
- Es importante tener el símbolo “/” entre cada bloque, procedimiento, función o triggers
- Realizar los ejercicios en los bloques delimitados en la plantilla de solución
- **¡Realizar comentarios en el código escrito que aporten valor!**
 - No se trata de comentar línea por línea como por ejemplo la declaración de variables, pero si ha de tener comentarios de las partes más importantes del código y por qué se realiza.

CONFIGURACIÓN INICIAL

Pasos previos a la realización de los ejercicios:

- Para la PAC puedes usar cualquier usuario que tenga todos los privilegios
- En SQL Developer crea una nueva conexión llamada “PAC_DES”
- Ejecutar el script **PAC_Desarrollo_2s2223_UF3_Configuracion_Inicial.sql**
 - Ejecuta el código que contiene la creación de 2 tablas
 - Revisa que las dos tablas están creadas correctamente
 - Revisa el contenido que se ha insertado en las tablas
 - La definición de las tablas es la siguiente:
 - **ASIGNATURAS**
 - COD_ASIG – Código único de cada asignatura (PK)
 - ABV_ASIG – Abreviatura del nombre de la asignatura
 - DES_ASIG – Descripción de la asignatura
 - SEMES – Número de semestre en el que se debiera cursar la asignatura
 - HORAS – Número de horas que tiene la asignatura
 - PRECIO – Precio de la asignatura
 - NUM_UFS – Número de Unidades Formativas de la asignatura
 - **UFS**
 - COD_ASIG – Código único de cada asignatura (PK)
 - COD_UF – Código de cada UF (PK)
 - DES_UF – Descripción de la UF
 - NUM_HORAS – Número de horas de la UF
 - PONDERA_UF – Ponderación que tiene la UF en la asignatura según sus horas
 - TOT_PACS_UF – Total de PACs que tiene la UF
 - MIN_PACS_ENT – Mínimo de PACs a entregar para hacer Examen de Ordinaria
 - NUM_PACS_ENT – Número de PACs que ha entregado el alumno de esa UF
 - NOTA_MEDIA_PACS – Nota media de las PACs entregadas de la UF
 - NOTA_EXAM – Nota del examen de la UF
 - CONV_EXAM – Convocatoria (ORDINARIA O EXTRAORDINARIA O PROYECTO)

EJERCICIOS

EJERCICIO 1) REPASO SQL (TABLAS Y VISTAS)

1.1. Añadir campos a las tablas ASIGNATURAS y UFS

- **Tabla ASIGNATURAS**
 - Campo NOM_PROFE tipo VARCHAR(50)
 - Campo APRO_UFS tipo INT
 - Campo NOTA_MEDIA_ASIG tipo NUMBER(4,2)
- **Tabla UFS**
 - Campo NOTA_MEDIA_UF tipo NUMBER(4,2)
 - Campo NOTA_FINAL_UF tipo INT
 - Campo STAT_UF tipo VARCHAR(10)

1.2. Crear una VISTA llamada "EXPEDIENTE".

Ha de ser INNER JOIN de las tablas ASIGNATURAS y UFS. Ha de contener los siguientes campos de cada tabla:

- **Tabla ASIGNATURAS**
 - ABV_ASIG
 - DES_ASIG
- **Tabla UFS**
 - PORC_PACS_ENTRE
 - Campo calculado que será % de PACs Entregadas según el total de PACs de la UF
 - $\text{ROUND}(\text{num_pacs_ent} / \text{tot_pacs_uf} * 100, 2)$ AS PORC_PACS_ENTRE
 - NOTA_MEDIA_PACS
 - NOTA_EXAM
 - CONV_EXAM
 - NOTA_MEDIA_UF
 - NOTA_FINAL_UF
 - STAT_UF

1.3. Actualizar registros de la tabla ASIGNATURAS.

- **Tabla ASIGNATURAS**
 - Actualizar el campo NOM_PROFE de los registros de las asignaturas de bases de datos
 - COD_ASIG = ICB0102A y COD_ASIG = ICB0102B:
 - Nombre del profesor = Emilio Saurina

EJERCICIO 2) PROCEDIMIENTO

Crear un procedimiento llamado "P_CALCULAR_NOTA_MEDIA_ASIG"

- Crear un procedimiento que a partir de un código asignatura pasado por parámetro nos devuelva el número de UFS aprobadas y la nota media de la asignatura si estan todas aprobadas
 - Variables de entrada
 - VIN_COD_ASIG
 - Variables de salida
 - VOUT_APRO_UFS
 - VOUT_NOTA_MEDIA_ASIG
- Para los cálculos se ha de tener en cuenta lo siguiente:
 - Si una UF está aprobada ($\text{NOTA_FINAL_UF} \geq 5$)
 - Se calcula la parte de nota de uf $\text{NOTA_FINAL_UF} * \text{PONDERA_UF}$
 - Se van sumando las notas ponderadas de cada UF
 - Durante el proceso se han de contar las UFS aprobadas y las NO aprobadas
 - Se va a devolver el número de UFS aprobadas
 - Si hay una o mas UFs NO aprobadas no se puede establecer nota media
 - El resultado final de $\text{NOTA_MEDIA_ASIG} = \text{NULL}$
 - Si todas las UFS estan aprobadas
 - El resultado final es el resultado calculado

EJERCICIO 3) FUNCION

Crea una función llamada “F_NOTA_MEDIA_UF”

- Crear una función devuelva la nota media de una UF a partir de los datos siguientes:
 - Variables de entrada
 - VIN_CONV_EXAM
 - VIN_NUM_PACS_ENT
 - VIN_MIN_PACS_ENT
 - VIN_NOTA_MEDIA_PACS
 - VIN_NOTA_EXAM
- Para los cálculos se ha de tener en cuenta lo siguiente:
 - Si la convocatoria es igual a “EXTRAORDINARIA”
 - La nota media es igual a la nota de examen
 - Si la convocatoria es “ORDINARIA” tenemos los siguientes casos:
 - La nota media es igual a la nota media de pacs por 0,4 en estos casos
 - Si el número de pacs entregadas es menor que mínimo de pacs a entregar
 - Si la nota de examen es menor a 4,75
 - Si la nota media de PACs es menor a 7 y la nota examen entre 4,75 y 4,89 a
 - Cuando no se dan ninguno de los demás casos la nota se calcula como
 - Nota media es igual a nota media de pacs por 0,4 mas nota examen por 0,6
 - Si la convocatoria es “PROYECTO” es un caso peculiar ya que el proyecto no tiene PACS
 - La nota media es igual a la nota de examen que es donde se pondrá la nota que tiene el proyecto

EJERCICIO 4) TRIGGER

Crea una trigger llamado “T_ACTUALIZA_NOTA_FINAL”

- Crear un Trigger que se dispare cuando se actualice la tabla de UFS.
- Usa la función creada para calcular la nota media actualizada de la UF modificada
- Con la nueva NOTA_MEDIA_UF se puede calcular la NOTA_FINAL_UF según la siguiente premisa
 - Si la convocatoria nueva es **EXTRAORDINARIA**
 - Si la NOTA_MEDIA_UF está entre 4,5 y 4,74
 - Coger la parte entera de la NOTA_MEDIA_UF
 - Por ejemplo 4,74 → 4
 - En todos los demás casos se aplica un redondeo normal
 - Redondeo normal de la NOTA_MEDIA_UF al alza en 0,5 o mas
 - Por ejemplo 4,75 → 5 o 5,49 → 5 o 3,65 → 4
 - Si la convocatoria nueva es **ORDINARIA**
 - Si la NOTA_MEDIA_UF está entre 4,5 y 4,74
 - Coger la parte entera de la NOTA_MEDIA_UF
 - Por ejemplo 4,74 → 4
 - Si la NOTA_MEDIA_UF está entre 4,75 y 4,89 y además la NOTA_MEDIA_PACS < 7
 - Coger la parte entera de la NOTA_MEDIA_UF
 - Por ejemplo 4,84 con NOTA_MEDIA_PACS = 6 → 4
 - Si la NOTA_MEDIA_UF está entre 4,75 y 4,89 y además la NOTA_MEDIA_PACS > 7
 - Redondeo normal de la NOTA_MEDIA_UF al alza en 0,5 o mas
 - Por ejemplo 4,84 con NOTA_MEDIA_PACS = 8 → 5
 - Si la NOTA_MEDIA_UF es mayor a 4,9
 - Redondeo normal de la NOTA_MEDIA_UF al alza en 0,5 o mas
 - Por ejemplo 4,94 → 5
 - Si la convocatoria nueva es **PROYECTO**
 - Si la NOTA_MEDIA_UF entre 4,5 y 4,99 Coger la parte entera
 - En los demás casos se hace Redondeo Normal
- En el trigger también actualizamos el nuevo valor del campo APRO_SUSP
 - Si la NOTA_FINAL_UF < 5 → SUSPENSO
 - Si la NOTA_FINAL_UF >= 5 → APROBADO
 - Si la NOTA_FINAL_IF IS NUL → PENDIENTE

EJERCICIO 5) BLOQUES ANÓNIMOS

5.1. Crear un bloque anónimo que actualice la NOTA_MEDIA_UF de todas las UFS

- Usa un cursor con la sentencia FOR UPDATE OF NOTA_MEDIA_UF para recorrer todos los registros de la tabla UFS y con la ayuda de la función creada F_NOTA_MEDIA_UF se ha de actualizar cada uno de los registros.
- Si el Trigger está correcto, al modificar la NOTA_MEDIA_UF se han de actualizar los campos NOTA_FINAL_UF y STAT_UF de forma automática

5.2. Crear un bloque anónimo que actualice NOTA_MEDIA_ASIG y APRO_UFS de todas las ASIGNATURAS

- Usa un cursor FOR UPDATE OF APRO_UFS, NOTA_MEDIA_ASIG para recorrer todos los registros de la tabla ASIGNATURAS y utiliza el procedimiento creado P_NOTA_MEDIA_ASIG para poder actualizar los valores de cada asignatura.

5.3. Crear un bloque anónimo que calcule y muestre la nota media final del ciclo

- Usa un cursor para recorrer todos los registros de la tabla ASIGNATURAS
 - Recuerda que se ha de ir multiplicando la NOTA_MEDIA_ASIG por PONDERA_ASIG que es la ponderación que tiene cada Asignatura en este ciclo
 - En caso de que una asignatura no tenga NOTA_MEDIA_ASIG no lo contamos para la media y la contabilizamos en una variable contador de V_NUM_ASIG_FALTA
 - Todas las Notas calculadas se irán sumando en una variable V_NOTA_MEDIA_CICLO
- Al terminar tendremos que sacar un mensaje por pantalla:
 - Si no hay asignaturas restantes el mensaje será:
 - El ciclo se ha terminado con una nota media de: "V_NOTA_MEDIA_CICLO"
 - Si tiene asignaturas restantes el mensaje será:
 - A falta de "V_NUM_ASIG_FALTA" asignaturas por aprobar. La nota media del ciclo es de: "V_NOTA_MEDIA_CICLO"