

EJERCICIOS DE RUTINAS ALMACENADAS Y CURSORES (I)

1. Escribe las sentencias SQL necesarias para crear una base de datos llamada centro, una tabla llamada comunidad y 6 sentencias de inserción para inicializar la tabla. La tabla comunidad está formada por las siguientes columnas:

- id (entero sin signo y clave primaria)
- nombre (cadena de caracteres)
- apellido1 (cadena de caracteres)
- apellido2 (cadena de caracteres)
- fecha_nacimiento (fecha)
- tipo (profesor, alumno, otros)

Una vez creada la tabla se decide añadir una nueva columna a la tabla llamada edad que será un valor calculado a partir de la columna fecha_nacimiento. Escribe la sentencia SQL necesaria para modificarla tabla y añadir la nueva columna. Escribe una función llamada calcular_edad que reciba una fecha y devuelva el número de años que han pasado desde la fecha actual hasta la fecha pasada como parámetro:

- Función: calcular_edad
- Entrada: Fecha
- Salida: Número de años (entero)

Ahora escriba un procedimiento que permita calcular la edad de todas las personas de la comunidad que ya existen en la tabla. Para esto será necesario crear un procedimiento llamado actualizar_columna_edad que calcule la edad de cada uno y actualice la tabla. Este procedimiento hará uso de la función calcular_edad que hemos creado en el paso anterior.

2. Modifica la tabla del ejercicio anterior para añadir una nueva columna email. Una vez que hemos modificado la tabla necesitamos asignarle una dirección de correo electrónico de forma automática. Escribe un procedimiento llamado crear_email que dados los parámetros de entrada: nombre, apellido1, apellido2 y dominio, cree una dirección de email y la devuelva como salida.

- Procedimiento: crear_email
- Entrada:
 - nombre (cadena de caracteres)
 - apellido1 (cadena de caracteres)
 - apellido2 (cadena de caracteres)
 - dominio (cadena de caracteres)
- Salida:
 - email (cadena de caracteres) devuelva una dirección de correo electrónico con el siguiente formato:
 - El primer carácter del parámetro nombre.
 - Los tres primeros caracteres del parámetro apellido1.
 - Los tres primeros caracteres del parámetro apellido2.
 - El carácter @.
 - El dominio pasado como parámetro.

Ahora escriba un procedimiento que permita crear un email para una persona que ya existe en la tabla. Para esto será necesario crear un procedimiento llamado actualizar_columna_email que actualice la columna email de la tabla personas. Este procedimiento hará uso del procedimiento crear_email que hemos creado en el paso anterior.

3. Escribe un procedimiento llamado crear_lista_emails_tipo que devuelva la lista de emails de la tabla según el tipo indicado y separados por un punto y coma. Ejemplo: jperfer@iescelia.org;operher@iescelia.org;pgonlan@iescelia.org;afermar@iescelia.org.

EJERCICIOS DE RUTINAS ALMACENADAS Y CURSORES (II)

1. Crea una función para insertar registros de movimientos en una cuenta de un cliente comprobando previamente que la fecha es menor que la actual y que la operación no deja la cuenta en negativo. La función devolverá un 0 en caso de error de entrada y 1 en cualquier otro caso. (Usar la base ebanca)

FUNCTION insertar_mov(cantidad int, codcuenta int, fecham datetime)

2. Crea un procedimiento que genere n registros aleatorios en la tabla movimientos de la base ebanca. Cada registro deberá contener datos de clientes y cuentas existentes. La cantidad deberá estar entre 1 y 100.000 y la fecha será la actual. PROCEDURE gen_mov (n int)

3. A partir de la base de datos Nmotor, realiza en cada caso un procedimiento usando cursores para que muestre:

- El autor que ha publicado más noticias y el que ha publicado menos.
- Los autores que han publicado entre M y N noticias
- El autor que menos noticias ha publicado en un día concreto.

4. Crea un procedimiento en la base de datos World que muestre para cada continente, los nombres y la población de los 5 países más poblados. Utiliza un cursor que recorra los distintos continentes que aparecen en la tabla country.

5. Crear un procedimiento que actualice el costo de reemplazo de las películas de una determinada categoría en la base de datos Sakila. Para calcular el nuevo valor del costo de reemplazo de cada película, se escogerá el mayor entre estos dos valores: el valor del costo incrementado en un porcentaje o bien un importe fijado de entrada.

Por lo tanto, se indicarán tres parámetros de entrada: el nombre de la categoría, un importe y un porcentaje.

6. Con la base de datos Ebanca, realiza en cada caso un procedimiento para que:

- Muestre los datos del cliente, la cuenta y el saldo de los clientes con saldo negativo en alguna de sus cuentas (usando cursores)
- Muestre la suma de todos los ingresos y cargos (por separado) en todas las cuentas de ebanca.

