

Ejercicios de Procedimientos y funciones almacenadas, triggers y eventos.

Se envía el procedimiento y el resultado.

Parte I.

Ejercicios sobre la base de datos SoftUni

1. Crea un procedimiento almacenado que devuelva los nombres y apellidos (FirstName y LastName) de los empleados con salario mayor o igual a 35000.
2. Crea un procedimiento almacenado que devuelva los nombres y apellidos (FirstName y LastName) de los empleados con salario mayor o igual a una cantidad X.
3. Escribe un procedimiento almacenado que acepte una cadena como parámetro y devuelva todas las ciudades (town) cuyo nombre empiece por las letras pasadas como parámetro. El resultado debe estar ordenado alfabéticamente.
4. Desarrolla un procedimiento almacenado que reciba el nombre de una ciudad (town) y devuelva el nombre (first and last name) de todos los empleados que viven en esa ciudad. Los resultados deben estar ordenados por FirstName.
5. Haz una función almacenada que recibe el salario de un empleado y devuelva el nivel de ese salario, de acuerdo a lo siguiente:
 - a) Si el salario <30000 devolverá "Low".
 - b) Si el salario está entre 30000 y 50000 (ambos incluidos) devolverá "Average".
 - c) Y si el salario es >50000 devolverá "High".
6. Haz un procedimiento almacenado que reciba como parámetro un nivel de salario (Low, Average, High) y devuelva todos los empleados (FirstName y LastName) de los empleados que están en ese nivel de salario. El resultado debe estar ordenado en orden descendente por FirstName y LastName.
7. Escribe una función que reciba como parámetros un conjunto de letras y una palabra, y devuelva verdadero o falso dependiendo de si todas las letras de la palabra aparecen en el conjunto de letras.
 - a) Ejemplo 1: letras: oistmiahf, palabra: Sofia, resultado: True (1)
 - b) Ejemplo 2: letras: oistmiahf, palabra: halves, resultado: False (0)

- c) Ejemplo 3: letras: bobr, palabra: Rob, resultado: True (1)
 - d) Ejemplo 4: letras: pppp, palabra: Guy, resultado: False (0)
8. Crea una tabla que sirva para almacenar el salario medio, el mínimo y el máximo de los empleados. Crea un trigger que actualice los datos de esa tabla.
 9. Modifica el trigger del ejercicio anterior para que la tabla sea un histórico de los salarios medios, mínimos y máximos, y que no almacene solamente los valores actuales.

Parte II

Ejercicios sobre la base da datos bank_db

10. Crea un procedimiento almacenado que selecciona el nombre completo de los que tienen una cuenta bancaria (account_holders). Los resultados deben estar ordenados por ese nombre completo.
11. Escribe un procedimiento almacenado que recibe una cantidad y devuelva nombre y apellidos (ordenados alfabéticamente) de toda la gente que tiene, sumando el total de sus cuentas, una cantidad mayor que la suma pasada como parámetro.
12. La siguiente tarea es crear una función que acepte como parámetros: suma, tasa de interés anual y número de años. Debe calcular y devolver el valor futuro de la suma inicial. Usando la siguiente fórmula:

$$FV = I((1 + R))^T \quad (1)$$

I es la cantidad incial, R la tasa de interés anual y T el número de años.

13. Crea un procedimiento almacenado, que use la función anterior, que calcule el dinero que un usuario tendrá en una cuenta en 5 años. El procedimiento recibirá un id de cuenta, y una tasa de interés. Devolverá el número de cuenta, nombre, apellido, interés, balance actual y balance a 5 años.
14. Elabora un trigger que informe de la cantidad de dinero cada vez que se actualice el dinero en la cuenta.
15. Crea una tabla en la que se almacene un límite (general para todos) mínimo de dinero a tener en cuenta. Crea un trigger que impida retirar dinero si ese límite se alcanza.
16. Crea un procedimiento almacenado que establezca un límite mínimo en cuenta para cada cuenta (rellenando la nueva tabla). Por ejemplo un 1 % de lo que hay en cuenta.
17. Realiza un procedimiento almacenado para retirar dinero de una cuenta (id, cantidad). Hay que comprobar que hay dinero suficiente y que la cantidad a retirar es positiva. Saca por pantalla el número de cuenta y el balance antes y después de retirar el dinero.
18. Realiza un procedimiento almacenado para depositar dinero en una cuenta (id,cantidad). Saca por pantalla el número de cuenta y el balance antes y después de ingresar el dinero.

19. Crea un procedimiento almacenado que reciba una cuenta de origen, una cantidad, una cuenta de destino y transfiera esa cuenta desde el origen al destino. Usa transacciones para que no se pierda el dinero en medio de la operación.
20. Haz una serie de triggers para conseguir lo que se hace en el ejercicio anterior. ¿Ha sido más fácil o más difícil?
21. Crea un trigger en la tabla accounts que almacene en la tabla logs (créala si no existe) (históricos) cada vez que hay un cambio en la cantidad de dinero de una cuenta. Se debe almacenar un log_id, account_id, old_sum, new_sum.
22. Realiza un trigger sobre la tabla logs que cree un .email cada vez que se modifica el balance de una cuenta. Debe emitir un mensaje que sea:
 - receptor: account_id
 - asunto: Cambio en el balance de la cuenta: account_id
 - mensaje: A fecha {date} el balance de su cuenta {account_id} ha cambiado de {old} a {new}.