

EJERCICIOS PROPUESTOS

Usando la base de datos **BDREPARACIONES**, realice lo siguiente:

1. Cree una función de tabla que devuelva el monto máximo, el monto mínimo y el monto promedio de la tabla Orden de Servicio usando el siguiente formato:

Tipo Valor	monto
=====	
Monto máximo	mostrar el valor calculado
Monto mínimo	mostrar el valor calculado
Monto promedio	mostrar el valor calculado

2. Implemente una función de tabla que retorne el código y descripción de los tipos de trabajadores, así como, la cantidad de equipos registrados por tipo de trabajador, pero únicamente, muestre los tipos que tengan menos de X órdenes emitidas y la descripción del tipo empiece con cierta letra. Envíe como parámetros el valor de X y la letra inicial de la descripción.
3. Implemente un cursor en un procedimiento almacenado donde liste lo siguiente :

Trabajador	Cantidad de Ordenes	Mensaje
Luis, Alva Z.	0	No emitió órdenes de servicio
Ana, Salazar M.	8	Emitió entre 1 y 10 órdenes de servicio
Juan, Pérez S.	15	Emitió más de 10 órdenes de servicio

4. Utilice case – when para el mensaje y respete el formato que muestra los datos del trabajador. Tendrá como parámetro un tipo de trabajador y un rango de años de emisión de la orden de servicio. Al finalizar muestre el total de trabajadores procesados.
5. Implemente una función escalar donde retorne la cantidad de trabajadores que pertenezcan a cierto tipo de trabajador (parámetro) y hayan ingresado a laborar entre dos años (parámetros).
6. Defina una función de tabla que liste los trabajadores que no emitieron órdenes de servicio en un rango de años y que sean de cierto tipo de trabajador, ambos enviados como parámetro. No se olvide de los trabajadores que no han emitieron ninguna orden.
7. Implemente un procedimiento almacenado donde liste las órdenes de servicio que Han registrado cierto TIPO de equipo cuyo código de tipo es enviado como parámetro. Valide con una función la existencia del tipo de equipo.

8. Implemente un cursor dentro de un *stored procedure* liste a los equipos, así como, la descripción de los tipos de equipos, que no están registrados en las órdenes de servicio del año enviado como parámetro. No se olvide de aquellos equipos que no se han registrado en ninguna orden. Considere únicamente a los equipos que cuya fecha de fabricación se encuentre entre los años enviados como parámetros.

Usando la base de datos BDPESCA, realice lo siguiente:

9. Implemente una función escalar donde retorne la cantidad de faenas registradas entre dos fechas de inicio y el código de un barco.
10. Defina una función de tabla que liste los barcos que no realizaron faenas en un rango de años y en cierta zona, ambos enviados como parámetro. No se olvide de los barcos que no han realizado ninguna faena.
11. Implemente un procedimiento almacenado donde liste las especies que son ingredientes de cierto TIPO producto cuyo código es enviado como parámetro. Valide con una función la existencia del tipo de producto.
12. Implemente un cursor en un procedimiento almacenado donde liste el nombre de la especie y la cantidad total de especies recolectadas por cada barco en el que se han realizado faenas en un determinado año. Al finalizar muestre la cantidad total de especies recolectadas.
13. Implemente una función de tabla que retorne el nombre del barco, el tiempo de servicio en años del barco, la cantidad de faenas realizadas por un determinado barco (parámetro) en un determinado año (parámetro) y un mensaje (use case).

Barco	tiempo	cantidad	mensaje
			Entre 1 y 5 años
			Entre 5 y 10
			Más de 10 años

14. Implemente un cursor en un procedimiento almacenado donde liste el nombre de la especie y la cantidad total de especies recolectadas por cada barco en el que se han realizado en faenas en un determinado año (quiebre por barco) Totalice por barco y al finalizar muestre la cantidad total.
15. Implemente un procedimiento almacenado que retorne el nombre del barco, el tiempo de servicio en años y la cantidad de faenas realizadas por un determinado empleado en un determinado año.

Parámetro de entrada: id del empleado y año.

Parametro de salida : nombre del barco,el tiempo de servicio en años y la cantidad de faenas).

Usando la base de datos BDANFITRIONAS, realice lo siguiente:

16. Crea un procedimiento almacenado que realice el mantenimiento de la tabla TIPOANFITRIONA. Las acciones válidas son (N)uevo, (E)limina, (M)odifica, (L)istado, recibiendo siempre como parámetros todos los datos de la tabla y la Accion a realizar, debe usar el control de transacciones para deshacer lo realizado si es que ocurriera un error.
17. Crea un procedimiento almacenado que inserte un registro en la tabla Anfitrionas y muestre todos los registros de dicha tabla. Antes de insertar el registro deberá validar que el mes de la fecha de nacimiento se encuentre entre 1 y 12, en caso no se cumpla con la condición deberá mostrar el mensaje de error e impedir que se inserte el registro, use TRY... CATCH y transacciones. Finalmente, deberá utilizar un cursor para mostrar todos los registros en donde se debe visualizar el último registro insertado.
18. Crea un trigger que al momento de ejecutar cualquier instrucción DML en la tabla ANFITRIONAS, registre en una tabla LOG el tipo de operación realizada, la hora de acción y el usuario que ejecuto el comando. Debe crear la tabla LOG.
Cuando se ejecute un insert o update, validar que el tipo anfitriona exista en la tabla TIPOANFITRIONA, de no existir registrarlo y en la descripción asignarle el valor "POR REGULARIZAR".
Cuando se ejecute un delete, eliminar el tipo anfitriona en la tabla TIPOANFITRIONA siempre y cuando sea el único asignado a ese tipo.

Usando la base de datos BDNegocios

19. Implemente una función que muestre la condición de venta de pedidos realizados por los empleados en un determinado año (use parámetro para el valor del año):
Si el empleado no realizó pedidos: NO realizó pedidos
Si el empleado realizó de 1 a 5 pedidos: no cumplió la meta
Si el empleado realizó más de 5 pedidos: cumplió la meta
Visualice en dicho proceso: Nombre y Apellidos del empleado, la cantidad de pedidos y la condición de venta del empleado
20. Implemente un procedimiento almacenado, que permita realizar la consulta de pedidos por producto en un determinado año (definir un parámetro en el proceso), almacenando la consulta en un cursor, el cual deberá imprimir el total, por cada producto, la cantidad solicitada; y al finalizar imprimir el total general de la cantidad de productos. La consulta deberá contener los campos siguientes: IdPedido, fechaPedido, NombreProducto (quiebre), precioUnidad y cantidad (ambos de tb_pedidosCabe)
21. Implemente una función de tabla que permita mostrar, por cada Cliente, la cantidad de pedidos registrados entre un rango de dos fechas (parámetros de entradas) y de aquellos Clientes que no registraron pedido en dicho rango de fechas.
22. Implemente un procedimiento almacenado que permita realizar el proceso de inserción o actualización de un registro a la tabla tb_productos, donde en dicho proceso deberá evaluarse, si es inserción, que no repita dos veces el idproducto y el nombreProducto; y si es actualización que el campo precio o

Unidades en Existencia sean mayores a cero . Para la operación utilice el parámetro @accion (0 : inserción, 1: actualización)

23. Implemente un cursor que permite realizar actualizar el precio de los productos, considerando la siguiente condición :
Si el stock es mayor o igual a 1000, el precio se le aplicará un descuento del 50%; sino se le aplicará un descuento del 20% del precio. Imprima el nombre del producto y el nuevo precio.
24. Cree un sp del usuario usando transacciones y try-catch que incremente en 10% los productos cuyo de la categoría se envía como parámetro y que tengan como unidades en existencia sea mayor a cero. Asimismo, tendrá como salida la cantidad de registros procesados (@@rowcount).
25. Cree un TRIGGER que permita insertar los datos de un Producto siempre y cuando el código y la descripción del producto sean únicos.
26. Cree un TRIGGER el cual permita eliminar Clientes los cuales no han registrado algún pedido. De eliminarse algún Cliente que no cumpla con dicha condición la operación no deberá ejecutarse.
27. Cree un TRIGGER que valide el precio unitario y su Stock de un producto, donde dichos datos sean mayores a cero.

Lo que se siembra se cosecha. A estudiar!!!

Dios los bendiga,



prof. Lidia Sánchez