

Práctica de T-SQL

Según el modelo dado resuelva:

1. Hacer una función que dado un artículo y un deposito devuelva un string que indique el estado del depósito según el artículo. Si la cantidad almacenada es menor al límite retornar "OCUPACION DEL DEPOSITO XX %" siendo XX el % de ocupación. Si la cantidad almacenada es mayor o igual al límite retornar "DEPOSITO COMPLETO".
2. Realizar una función que dado un artículo y una fecha, retorne el stock que existía a esa fecha

Haces top 1 de los que tengan empl_jefe is null y ordenas por saldo mayor

3. Cree el/los objetos de base de datos necesarios para corregir la tabla empleado en caso que sea necesario. Se sabe que debería existir un único gerente general (debería ser el único empleado sin jefe). Si detecta que hay más de un empleado sin jefe deberá elegir entre ellos el gerente general, el cual será seleccionado por mayor salario. Si hay más de uno se seleccionara el de mayor antigüedad en la empresa. Al finalizar la ejecución del objeto la tabla deberá cumplir con la regla de un único empleado sin jefe (el gerente general) y deberá retornar la cantidad de empleados que había sin jefe antes de la ejecución.

Para retornar el codigo podes poner un print

4. Cree el/los objetos de base de datos necesarios para actualizar la columna de empleado empl_comision con la sumatoria del total de lo vendido por ese empleado a lo largo del último año. Se deberá retornar el código del vendedor que más vendió (en monto) a lo largo del último año.

No hace falta cursor

Hacer set empl_comision y ahi dentro una subquery

5. Realizar un procedimiento que complete con los datos existentes en el modelo provisto la tabla de hechos denominada Fact_table tiene la siguiente definición:

```
Create table Fact_table
( anio char(4),
  mes char(2),
  familia char(3),
  rubro char(4),
  zona char(3),
  cliente char(6),
  producto char(8),
  cantidad decimal(12,2),
  monto decimal(12,2)
)
```

```
Alter table Fact_table
```

```
Add constraint primary key(anio,mes,familia,rubro,zona,cliente,producto)
```

6. Realizar un procedimiento que si en alguna factura se facturaron componentes que conforman un combo determinado (o sea que juntos componen otro producto de mayor nivel), en cuyo caso deberá reemplazar las filas

correspondientes a dichos productos por una sola fila con el producto que componen con la cantidad de dicho producto que corresponda.

7. Hacer un procedimiento que dadas dos fechas complete la tabla Ventas. Debe insertar una línea por cada artículo con los movimientos de stock generados por las ventas entre esas fechas. La tabla se encuentra creada y vacía.

Otro procedure

VENTAS

Código	Detalle	Cant. Mov.	Precio de Venta	Renglón	Ganancia
Código del articulo	Detalle del articulo	Cantidad de movimientos de ventas (Item factura)	Precio promedio de venta	Nro. de línea de la tabla	Precio de Venta – Cantidad * Costo Actual

8. Realizar un procedimiento que complete la tabla Diferencias de precios, para los productos facturados que tengan composición y en los cuales el precio de facturación sea diferente al precio del cálculo de los precios unitarios por cantidad de sus componentes, se aclara que un producto que compone a otro, también puede estar compuesto por otros y así sucesivamente, la tabla se debe crear y está formada por las siguientes columnas:

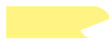
Procedure

DIFERENCIAS

Código	Detalle	Cantidad	Precio_generado	Precio_facturado
Código del articulo	Detalle del articulo	Cantidad de productos que conforman el combo	Precio que se compone a través de sus componentes	Precio del producto

9. Crear el/los objetos de base de datos que ante alguna modificación de un ítem de factura de un artículo con composición realice el movimiento de sus correspondientes componentes. Hay que actualizar el stock. acer el join con inserted y delted
10. Crear el/los objetos de base de datos que ante el intento de borrar un artículo verifique que no exista stock y si es así lo borre en caso contrario que emita un mensaje de error.
11. Cree el/los objetos de base de datos necesarios para que dado un código de empleado se retorne la cantidad de empleados que este tiene a su cargo (directa o indirectamente). Solo contar aquellos empleados (directos o indirectos) que tengan un código mayor que su jefe directo.

12. Cree el/los objetos de base de datos necesarios para que nunca un producto pueda ser compuesto por sí mismo. Se sabe que en la actualidad dicha regla se cumple y que la base de datos es accedida por n aplicaciones de diferentes tipos y tecnologías. No se conoce la cantidad de niveles de composición existentes.
13. Cree el/los objetos de base de datos necesarios para implantar la siguiente regla “Ningún jefe puede tener un salario mayor al 20% de la suma de los salarios de sus empleados totales (directos + indirectos)”. Se sabe que en la actualidad dicha regla se cumple y que la base de datos es accedida por n aplicaciones de diferentes tipos y tecnologías



Tengo que hacer para mañana

14. Agregar el/los objetos necesarios para que si un cliente compra un producto compuesto a un precio menor que la suma de los precios de sus componentes que imprima la fecha, que cliente, que productos y a qué precio se realizó la compra. No se deberá permitir que dicho precio sea menor a la mitad de la suma de los componentes.
15. Cree el/los objetos de base de datos necesarios para que el objeto principal reciba un producto como parametro y retorne el precio del mismo.
Se debe prever que el precio de los productos compuestos sera la sumatoria de los componentes del mismo multiplicado por sus respectivas cantidades. No se conocen los niveles de anidamiento posibles de los productos. Se asegura que nunca un producto esta compuesto por si mismo a ningun nivel. El objeto principal debe poder ser utilizado como filtro en el where de una sentencia select.
16. Desarrolle el/los elementos de base de datos necesarios para que ante una venta automaticamente se descuenten del stock los articulos vendidos. Se descontaran del deposito que mas producto posea y se supone que el stock se almacena tanto de productos simples como compuestos (si se acaba el stock de los compuestos no se arman combos)
En caso que no alcance el stock de un deposito se descontara del siguiente y asi hasta agotar los depositos posibles. En ultima instancia se dejara stock negativo en el ultimo deposito que se desconta.
17. Sabiendo que el punto de reposicion del stock es la menor cantidad de ese objeto que se debe almacenar en el deposito y que el stock maximo es la maxima cantidad de ese producto en ese deposito, cree el/los objetos de base de datos necesarios para que dicha regla de negocio se cumpla automaticamente. No se conoce la forma de acceso a los datos ni el procedimiento por el cual se incrementa o descuenta stock

18. Sabiendo que el limite de credito de un cliente es el monto maximo que se le puede facturar mensualmente, cree el/los objetos de base de datos necesarios para que dicha regla de negocio se cumpla automaticamente. No se conoce la forma de acceso a los datos ni el procedimiento por el cual se emiten las facturas

Necesitasc cursor

19. Cree el/los objetos de base de datos necesarios para que se cumpla la siguiente regla de negocio automáticamente “Ningún jefe puede tener menos de 5 años de antigüedad y tampoco puede tener más del 50% del personal a su cargo (contando directos e indirectos) a excepción del gerente general”. Se sabe que en la actualidad la regla se cumple y existe un único gerente general.

Hacer dos funciones apartes para la parte de año y cant de empelados

20. Crear el/los objeto/s necesarios para mantener actualizadas las comisiones del vendedor.

El cálculo de la comisión está dado por el 5% de la venta total efectuada por ese vendedor en ese mes, más un 3% adicional en caso de que ese vendedor haya vendido por lo menos 50 productos distintos en el mes.

CON CURSOR

21. Desarrolle el/los elementos de base de datos necesarios para que se cumpla automaticamente la regla de que en una factura no puede contener productos de diferentes familias. En caso de que esto ocurra no debe grabarse esa factura y debe emitirse un error en pantalla.

CON CURSOR

Hice una funcion a partir para la parte de las familias

22. Se requiere recategorizar los rubros de productos, de forma tal que nigrubro tenga más de 20 productos asignados, si un rubro tiene más de 20 productos asignados se deberan distribuir en otros rubros que no tengan mas de 20 productos y si no entran se debra crear un nuevo rubro en la misma familia con la descirpción “RUBRO REASIGNADO”, cree el/los objetos de base de datos necesarios para que dicha regla de negocio quede implementada.

23. Desarrolle el/los elementos de base de datos necesarios para que ante una venta automaticamente se controle que en una misma factura no puedan venderse más de dos productos con composición. Si esto ocurre debera rechazarse la factura.

24. Se requiere recategorizar los encargados asignados a los depositos. Para ello cree el o los objetos de bases de datos necesarios que lo resueva, teniendo en cuenta que un deposito no puede tener como encargado un empleado que pertenezca a un departamento que no sea de la misma zona que el deposito, si esto ocurre a dicho deposito debera asignársele el empleado con menos depositos asignados que pertenezca a un departamento de esa zona.

25. Desarrolle el/los elementos de base de datos necesarios para que no se permita que la composición de los productos sea recursiva, o sea, que si el producto A

compone al producto B, dicho producto B no pueda ser compuesto por el producto A, hoy la regla se cumple.

26. Desarrolle el/los elementos de base de datos necesarios para que se cumpla automaticamente la regla de que una factura no puede contener productos que sean componentes de otros productos. En caso de que esto ocurra no debe grabarse esa factura y debe emitirse un error en pantalla.

27. Se requiere reasignar los encargados de stock de los diferentes depósitos. Para ello se solicita que realice el o los objetos de base de datos necesarios para asignar a cada uno de los depósitos el encargado que le corresponda, entendiendo que el encargado que le corresponde es cualquier empleado que no es jefe y que no es vendedor, o sea, que no está asignado a ningún cliente, se deberán ir asignando tratando de que un empleado solo tenga un deposito asignado, en caso de no poder se irán aumentando la cantidad de depósitos progresivamente para cada empleado.

28. Se requiere reasignar los vendedores a los clientes. Para ello se solicita que realice el o los objetos de base de datos necesarios para asignar a cada uno de los clientes el vendedor que le corresponda, entendiendo que el vendedor que le corresponde es aquel que le vendió más facturas a ese cliente, si en particular un cliente no tiene facturas compradas se le deberá asignar el vendedor con más venta de la empresa, o sea, el que en monto haya vendido más.

29. Desarrolle el/los elementos de base de datos necesarios para que se cumpla automaticamente la regla de que una factura no puede contener productos que sean componentes de diferentes productos. En caso de que esto ocurra no debe grabarse esa factura y debe emitirse un error en pantalla.

[Esta repetida esta](#)

30. Agregar el/los objetos necesarios para crear una regla por la cual un cliente no pueda comprar más de 100 unidades en el mes de ningún producto, si esto ocurre no se deberá ingresar la operación y se deberá emitir un mensaje “Se ha superado el límite máximo de compra de un producto”. Se sabe que esta regla se cumple y que las facturas no pueden ser modificadas. [Use el if exists\(select 1\)](#)

[Hecha con cursor y en la query ademas del inserte, agregue y joinee item_factura con factura](#)

31. Desarrolle el o los objetos de base de datos necesarios, para que un jefe no pueda tener más de 20 empleados a cargo, directa o indirectamente, si esto ocurre debera asignarsele un jefe que cumpla esa condición, si no existe un jefe para asignarle se le deberá colocar como jefe al gerente general que es aquel que no tiene jefe.