**PRODUCTO FINAL. AVANCE 2**

Use factura

1. Implemente un bloque de instrucciones que permita eliminar un registro de Cliente por su código. Si hay un error en el proceso, visualice un mensaje de error. Implemente funciones especiales de error.
2. Implemente un proceso que elimine los productos cuyo valor de sus unidades en existencia, sea menor a 20, en caso de no ejecutar exitosamente el error, implemente el CATCH para controlar el error.
3. Trabajando con un cursor dinámico, defina un cursor dinámico que permita visualizar: el primer registro, el registro en la posición 6 y el último registro.
4. Trabajando con un cursor, imprimir un listado de los pedidos realizados por cada año. En dicho proceso, listar por cada año los pedidos registrados y totalizando dichos pedidos por dicho año.
5. Para actualizar los datos de un cursor, debemos especificar la cláusula FOR UPDATE después de la sentencia SELECT en la declaración del cursor, y WHERE CURRENT OF <nombre\_cursor> en la sentencia UPDATE.

Actualice el precio de los productos: si su stock es mayor o igual a 1000,

se descuenta el precio al 50%, sino se descuenta el precio al 20%.

1. Crear una función de tabla en línea que permita mostrar los pedidos realizados por un empleado (idempleado).
2. Crear un Trigger que guarde los datos en la tabla ControlTrigger cuando se relice una modificación en un empleado. Guardar la hora de la actualización en un campo aparte en la tabla ControlTrigger. (Añadir un campo)
3. Queremos que se guarde en una tabla *EmpleadosInsertados*  el historial de inserciones de registros realizadas en la tabla empleados, además de los datos del empleado se deberá guardar en la tabla el usuario que realizó la inserción del empleado y la fecha/hora de la operación.
4. Crear un Trigger que borre sobre la tabla Empleados. Mostrar el registro borrado al ejecutar el Trigger, incluyendo la fecha de la operación y el usuario.

-----------------------------------------------------------------------

FUNCIONES ARITMETICAS.

ABS(n): Devuelve el valor absoluto de “n”.

ROUND(m, n): Redondea el número “m” con el número de decimales indicado en “n”,

si no se indica “n” asume cero decimales.

SQRT(n): Devuelve la raíz cuadrada del parámetro que se le pase.

POWER(m, n): Devuelve la potencia de “m” elevada el exponente “n”.

FUNCIONES DE CADENAS

LOWER(c): Devuelve la cadena “c” con todas las letras convertidas a minúsculas.

UPPER(c): Devuelve la cadena “c” con todas las letras convertidas a mayúsculas.

LTRIM(c): Elimina los espacios por la izquierda de la cadena “c”.

RTRIM(c): Elimina los espacios por la derecha de la cadena “c”.

REPLACE(c, b, s): Sustituye en la cadena “c” el valor buscado “b” por el valor indicado en “s”.

REPLICATE(c, n): Devuelve el valor de la cadena “c” el número de veces “n” indicado.

LEFT(c, n): Devuelve “n” caracteres por la izquierda de la cadena “c”.

RIGHT(c, n): Devuelve “n” caracteres por la derecha de la cadena “c”.

SUBSTRING(c, m, n): Devuelve una sub-cadena obtenida de la cadena “c”, a partir de la posición “m” y

tomando “n” caracteres.

SOUNDEX(c): Devuelve una cadena con la representación fonética para indexación en

inglés (algoritmo Soundex) de la cadena “c”.

FUNCIONES DE FECHAS

YEAR(d): Devuelve el año correspondiente de la fecha “d”.

MONTH(d): Devuelve el mes de la fecha “d”.

DAY(d): Devuelve el día del mes de la fecha “d”.

DATEADD(f, n, d): Devuelve una fecha “n” periodos (días, meses años, según lo indicado) superior a la fecha “d”. Si se le pasa un número “n” negativo, devuelve una fecha “n” periodos inferior. De gran utilidad en consultas.

FUNCIONES DE CONVERSIÓN

En SQL Server tenemos funciones como CAST, CONVERT o PARSE, que en función de lo que especifiquemos en sus parámetros convertirán los datos en el tipo que le indiquemos. Ver funciones de conversión de SQL Server.

En Oracle tenemos funciones como TO\_CHAR, TO\_DATE, TO\_NUMBER. Ver funciones de conversión de Oracle.

En MySQL solucionamos la mayor parte de las conversiones con la función CAST. Ver conversiones en MySQL.