SQL

Structure Query Language.

Lenguaje estructurado de consultas.

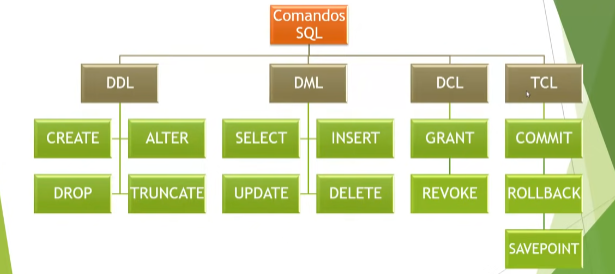
4 grandes grupos de comandos:

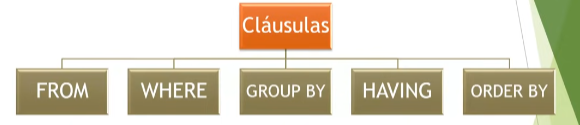
DDL. Data definition language. Crea, modifica y elimina tablas.

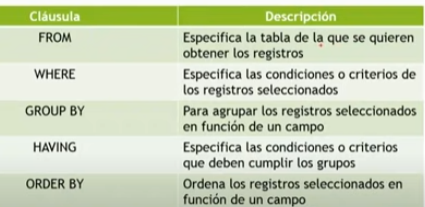
DML. Data manipulation language. Sirve para seleccionar, eliminar registros.

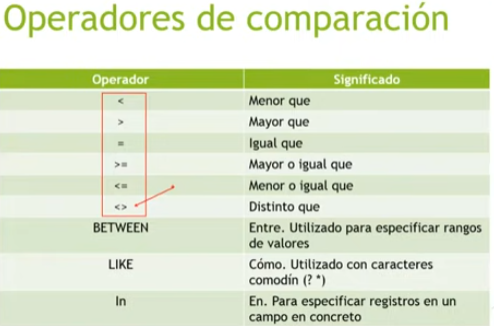
DCL. Data control language. Da seguridad a la info.

TCL. Transaction control language. Gestionan los cambios en los datos.

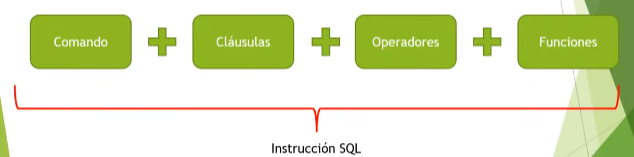












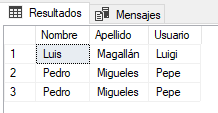
No están necesariamente las 4 presentes en una instrucción SQL.

Usa Access y MySQL

Mayúscula o minúscula es lo mismo

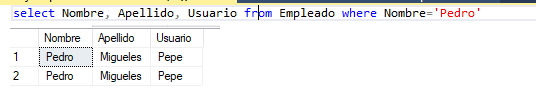
select Nombre, Apellido, Usuario from Empleado

Empleado es la tabla. La consulta se ejecuta tocando sobre la BBDD.

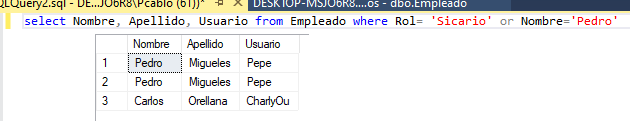


Desaconseja incluir acentos

Datos de tipo texto van entrecomillados (a lo string).

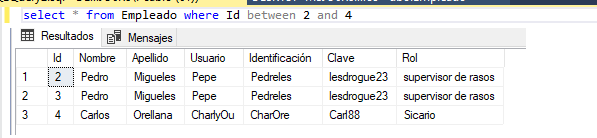


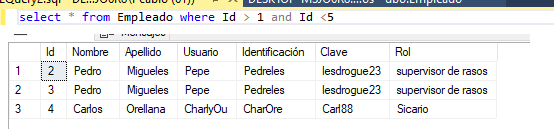
Ahí use el where. Con comillas simples.



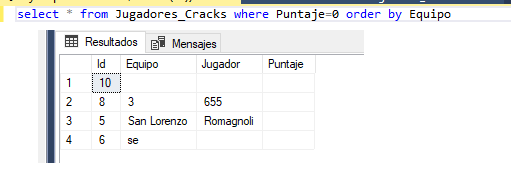
Ahí fue el Or.

Si se le pone a un campo VarChar después no se podrían hacer operaciones matemáticas en caso de ingresarse números.

Between y and.

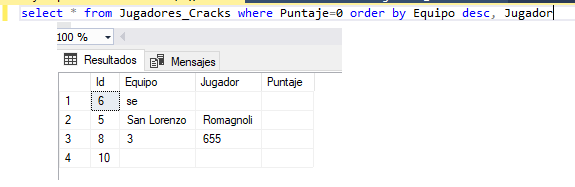
Ahí va con los operadores de comparación. Hay que repetir el campo Id.

Ahora va order by.



Ahí aparece la tablita ordenada según equipo. Primero hay un vacío, después un número, y después sa, luego se.

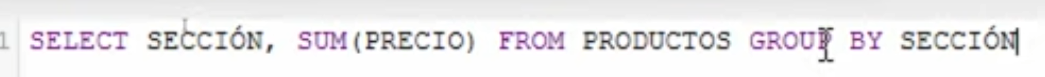
Para que se ordene de Z-A o de mayor a menor se hace agregando desc al final



Ahí se ve el desc.

Agregué Jugador después de la coma. Eso hace que se agregue un segundo criterio de ordenación, que agarre lo ya ordenado y ordene eso, en función del nuevo criterio.





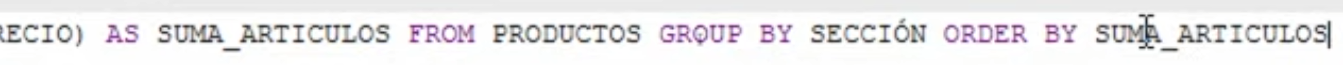
Con eso lo que logra es sumar todos los precios de las categorías que están en sección, de la tabla productos.

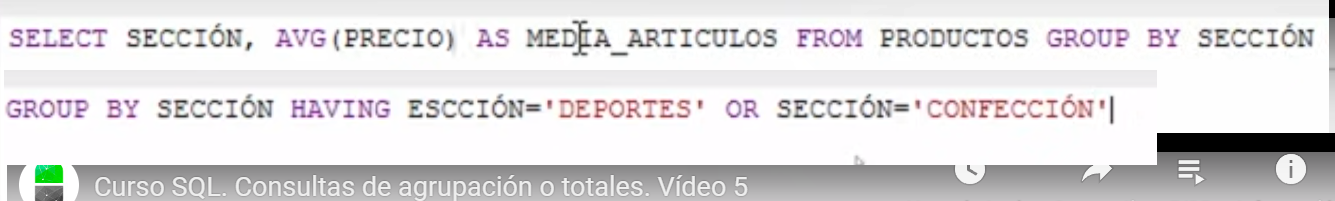


Alias. Permite cambiarle el nombre a un campo en una consulta. En la tabla, obviamente no cambia.

De otra forma, el order by Precio no funcionaría, ya que eso cambió.

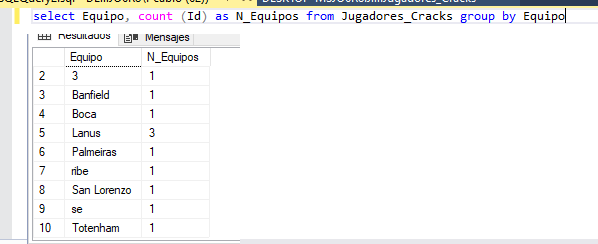
Un alias se define con un as, en la consulta original, y luego se lo utiliza con el order by.



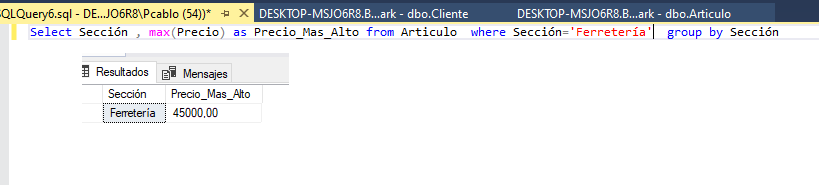


Ahí está sacando el promedio que se da en la sección deportes y en confección. Usa AVG, para eso, y en vez de where utiliza having.

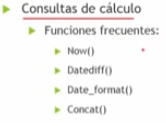
El Count no sirve si se utiliza sobre un campo que está vacío, ya que no lo contará. Por lo que suele utilizarse el campo Id. Que es autoincrementable, por lo que se autocompleta.



Con esa query cuenta cuantos elementos de cada equipo hay. Hay 3 de Lanús.



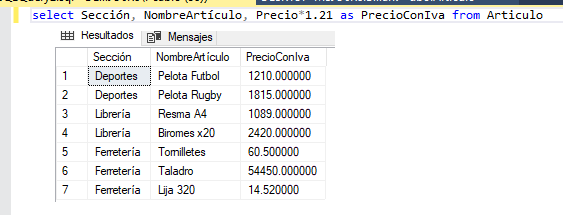
Con ese comando te devuelve, dentro del campo sección, el elemento con el valor precio mas alto dentro de precio.



Agrega otras 2. Round y truncate.

Las funciones dependen del gestor de base de datos. Es decir, podrían haber diferencias entre MySQL y SqlServer. Estas 4 creo que son comunes a varios gestores.

Estas consultas se hacen sobre cálculos individuales, y no sobre grupos.



Ahí se hace una consulta aplicándole el Iva a los valores, y usando un alias para que la columna 3 tenga ese nombre.



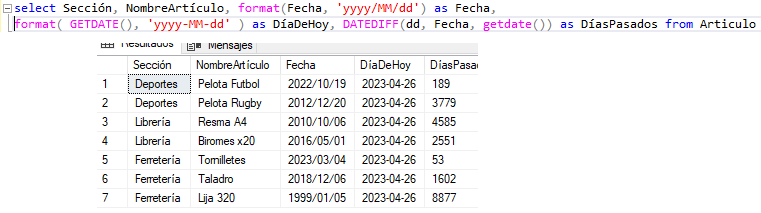
El round (valor, cantidad de decimales) se usa para redondear. A mi no me funcionó. En internet figura que esto puede pasar, puede tener que ver con el tipo de dato, que sería float.

Cálculos con flechas:

Pildoras da la función now(), a mi me funca con getdate().



Ahí armé esta tabla, con 2 campos nuevos (cuyo nombre es el alias que uno les puso). Uno se construye con getdate y el otro con datediff([dia, mes o año], fechasantiguas, fechaactual).



Con format (campodefechas, ‘yyyy-MM-dd’) establezco el formato. Usé alguna variación yo. Se pueden utilizar otras.

Consultas multitabla



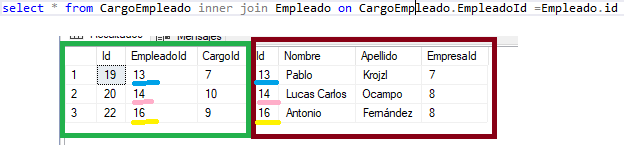
Union requiere que haya mismo número de campos y datos de campo compatibles. Por ejemplo, varchar e int no son compatibles. El nombre de las columnas no tiene por qué ser igual.



Se hacen 2 select, unidos por el unión.

Union all a diferencia de unión muestra, si hay registros repetidos, todas sus repeticiones. Union no. 2 registros son iguales, si todos sus campos son iguales.

Inner Join



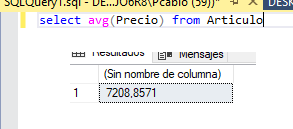
CargoEmpleado inner join Empleado son las 2 tablas que se vinculan. Después del on se pone el nombredetabla.nombredelcampoquetienenigual.

Subconsultas

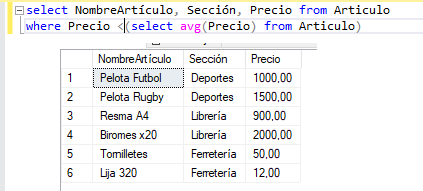
Hay una consulta hija dentro de una consulta padre. La primera hace un filtrado de resultados.

Subconsulta escalonada

Devuelve un campo y un registro. Eso es lo que se compara con selecct padre. Sirve como criterio o filtro. En este caso es el promedio de todos los precios de los productos.

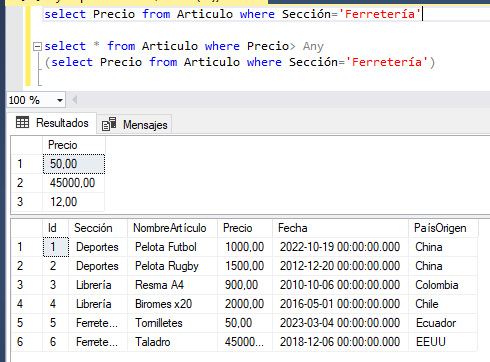


Luego, eso primero se pone dentro de paréntesis y se lo utiliza al final de este nuevo comando.



Subconsultas de lista

Devuelve una lista de registros. También es una consulta dentro de otra.



En la consulta “completa” te devuelve los artículos que sean mayores que alguno (any) de los artículos de la sección ferretería. Como se ve, todos son mayores que eso que cuesta 12.

Predicados In y Not In



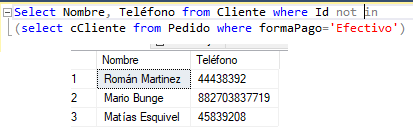
Id refiere a la tabla cliente. cCliente refiere a la tabla pedido.

Se obtienen los datos de la primera línea de los clientes que figuran en la tabla Pedido que hayan abonado en Efectivo.

También podría hacerse con un if, también con Inner Join (sería mas dificultoso)



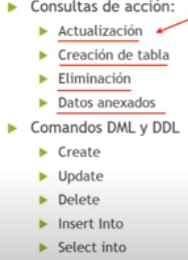
Lo bueno de esto es que se relaciona tablas que podrían no estar relacionadas.



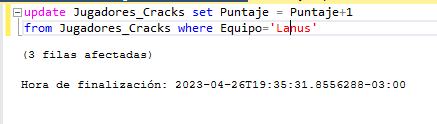
Ahí con el not in, agrega a la tabla a los que no pagaron en efectivo.

Consultas de acción

Modifican la información que hay en las tablas, o crean tablas nuevas a partir de otras existentes. Pueden llevar info de una tabla a otra.



Update:



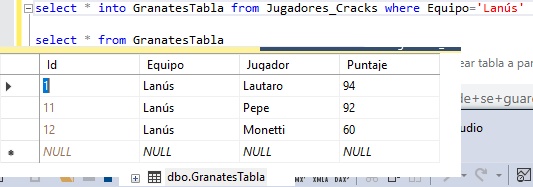
Le suma 1 al puntaje de los que están en equipo=Lanús.

update Jugadores\_Cracks set Equipo='Lanús'

from Jugadores\_Cracks where Equipo='Lanus'

Con eso cambia el nombre de Lanus a Lanús.

Crear tabla a partir de otra



Select columnas into nombreNuevaTabla from TablaOriginal where condicion.

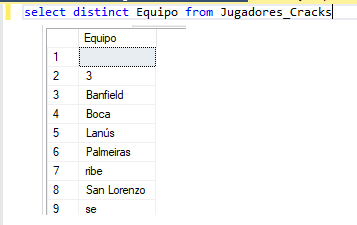
Eliminación de registros

delete from Jugadores\_Cracks where Equipo='Totenham'

delete from Jugadores\_Cracks where Jugador=''

Eso para borrar campos nulls

Distinct facilita que no se Repita un registro.



Hay 3 jugadores de Lanús, pero acá aparece una sola vez.

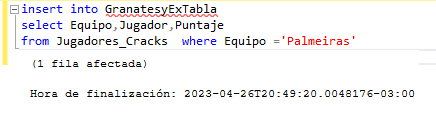
DistinctRow mirá no solo un campo sino todos los campos de una fila. Si se usa un Id autoincrementable nunca habrán 2 filas iguales.

Insertar registro desde 0

insert into Jugadores\_Cracks (Equipo,Jugador,Puntaje)

select 'Boca','Benedetto', 68

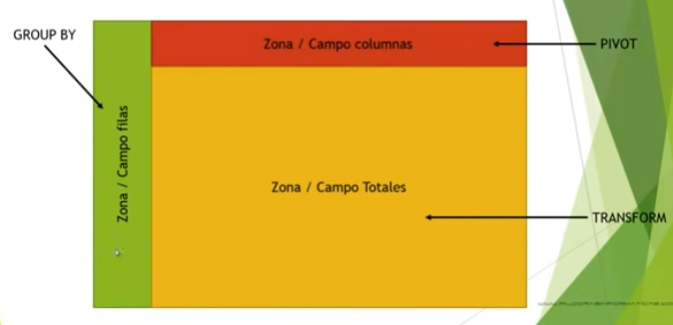
Insertar registro desde otra tabla

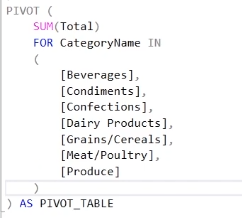


Ahí va joya. Me marca error en el nombre de la tabla, por ortiva. El Id se genera automáticamente.

Consultas de referencias cruzadas

Es lo mismo que una tabla dinámica de Excel.





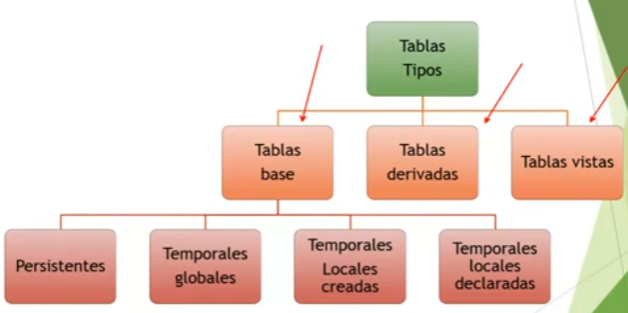
Así lo hace un muchachín.



Crea una fila por cada uno de los registros del campo CategoryName.

Existe también unpivot.

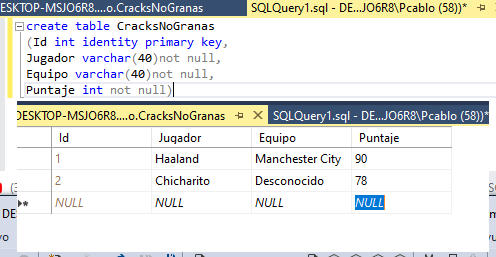
Data Definition Language







Crear Tabla



Eliminar tabla

drop table GranatesyExTabla

Cambiar tipo de dato

alter table Jugadores\_Cracks alter column Puntaje varchar(50)

alter table +nombreTabla+ alter columna + nombreColumna +tipoDeDatoAInstaurar

Agregar columna a tabla

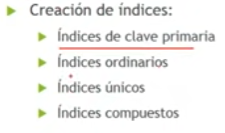
alter table CracksNoGranas Add Debut date

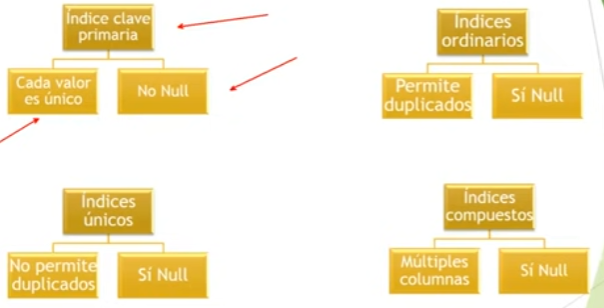


Eliminar columna de tabla

alter table CracksNoGranas drop column Debut

Crear y modificar índices

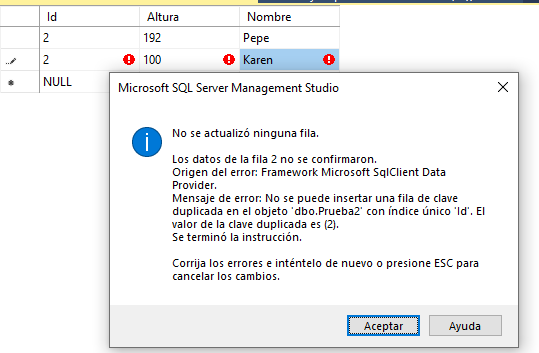




La Primary Key no tiene por qué se numérica ni autoincrementable. Pero no puede repetirse.

Esto funcó:

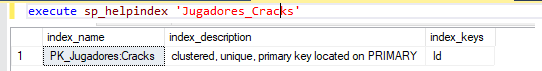
Create unique index Id on Prueba2 (Id)



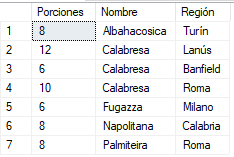
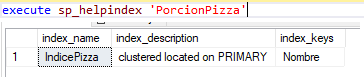
Indices pueden ser agrupados (cluster) o no agrupados (no cluster).

Creo que si está agrupado de alguna forma se ordena.

Obtener info de index





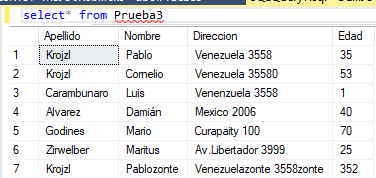


Ahí se ve que se puede repetir, pero se ordena agrupando por el índice, según orden alfabético.

Las no agrupadas se usan para mejorar la eficiencia cuando hay muchos registros.

nonclustered

create nonclustered index IDXApellido on Prueba3 (Apellido)



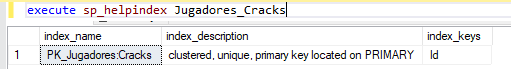
Ahí se ve, que no está ordenado. Pobre Pablozonte.

Eliminar indices

drop index IDXApellido on Prueba3

No hay que poner en qué campo está

La instrucción DROP INDEX no elimina los índices creados con las restricciones [**PRIMARY KEY**](https://estradawebgroup.com/Post/-Que-es-PRIMARY-KEY-en-SQL-Server-y-como-funciona-/20122) o [**UNIQUE**](https://estradawebgroup.com/Post/Como-restringir-que-los-valores-de-una-columna-sean-unicos-en-SQL-Server-con-UNIQUE/20391). Para eliminar índices asociados con estas restricciones, utiliza la instrucción [**ALTER TABLE**](https://estradawebgroup.com/Post/Como-modificar-una-tabla-SQL-Server-con-Alter-Table/20392) DROP CONSTRAINT.



Funca:

alter table Jugadores\_Cracks drop constraint [PK\_Jugadores:Cracks]

Lo que está dentro del segundo corchete es el nombre del índice.

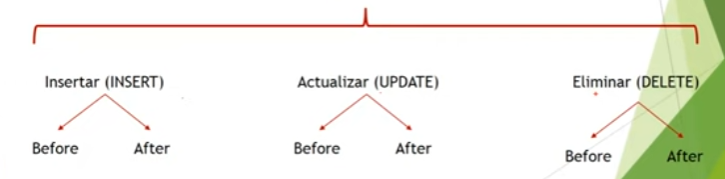
Triggers

Disparadores.

Es un objeto que se crea. Siempre va a estar asociado a una tabla, no existen en libertad.

Desencadenan una acción cuando ocurre algo en la tabla. Cuando alguien inserte, elimine o actualice algo.

Por ejemplo, podrían usarse para crear un backup o copia de seguridad de alguna tabla que va a ser modificada.



Se usen o no triggers siempre se crean, al insertarse registros, una tabla inserted temporalmente. Al borrarse algo, se crea una tabla temporal deleted. El comando update utiliza esa tabla temporal.

Un trigger es como un procedimiento almacenado.

Eso funcaría:

1\_ Trigger a crear.

2\_Nombre de tabla original.

4\_La tabla a la que se le insertan los datos. Se declara una variable y luego se la iguala a la variable de nombre deseado que está en la tabla temporal inserted. Al final se llama a la funcón getdate().

create trigger Articulo\_AI

on Jugadores\_Cracks for insert

As

begin

declare @Nombre varchar (30)

select @Nombre= Jugador from inserted

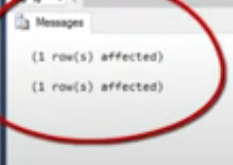
declare @Puntaje int

select @Puntaje=Puntaje from inserted

insert into HistorialJugadores (Nombre, Puntaje, Fecha) values (@Nombre , @Puntaje, getdate())

end

go



Cuando se inserta un registro pasa eso. El muchacho dice que está mal. Para que eso no se vea, hace: set nocount on tras el AS



Es un trigger de update. Ahí lo que hace es que se reste la cantidad de stock de determinado producto, haciendo un inner join entre la tabla productos y la tabla temporal por intermedio del campo cod\_prod.

Habla de que los triggers son after o instead of. Este último se dispara antes de que ocurra un cambio en la tabla.

Instead of insert

create trigger TablaPrimeraAI

on TablaPrimera for update

As

Begin

declare @Código varchar (50)

declare @Producto varchar (50)

declare @Cantidad int

declare @Precio money

select @Código = Código from inserted

select @Producto = Producto from inserted

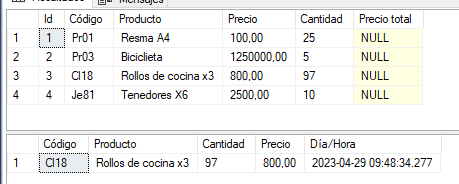
select @Cantidad = Cantidad from inserted

select @Precio = Precio from inserted

insert into TablaSegunda (Código, Producto, Cantidad, Precio, [Día/Hora]) values (@Código, @Producto, @Cantidad, @Precio, getdate())

end

go



Se realizó un update sobre la primera tabla, seteando la cantidad donde el código cl18 en cantidad=cantidad-3.

Esto de acá no está bien, aunque contiene algo de bondad en sus líneas:

as

begin

declare @Cantidad int

declare @Precio money

select @Cantidad= Cantidad from inserted

select @Precio= Precio from inserted

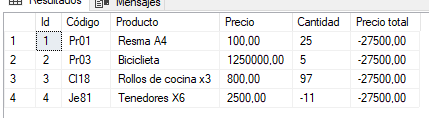
declare @PrecioTotal money

select @PrecioTotal = @Cantidad\*@Precio

update TablaPrimera set [Precio total]= @PrecioTotal

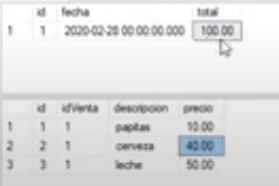
end

go



Hdeleon

https://www.youtube.com/watch?v=uTx7xd4ojkk



Hacen una venta. Vendiendo los 3 productos que figuran abajo, y automáticamente se hace un total y se agrega la fecha.

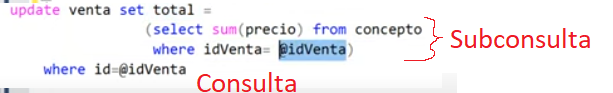
Venta es la tabla de arriba, concepto la de abajo.

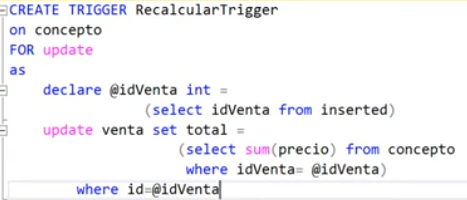
Con el declare @ se crea una variable.



Lo hace así, mas cortito.

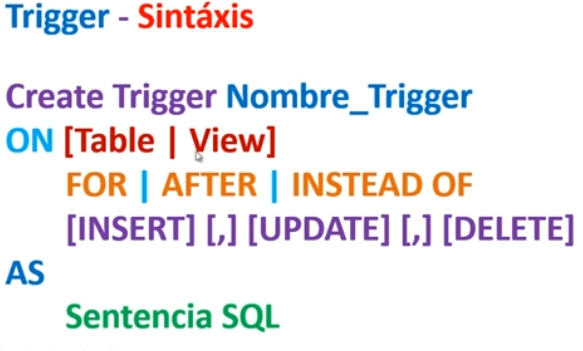
Where idVenta= @idVenta significa que solo se van a sumar los precios que en la tabla concepto se haya agregado a la tabla inserted , es decir los que se acaban de agregar.

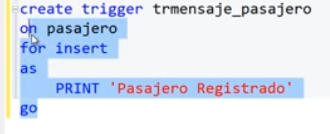




Ante cada modificación que le haga a los valores se le va a modificar el valor de la tabla venta.

[Juan Carlos Arcila Díaz](https://www.youtube.com/@jcarlosad7)

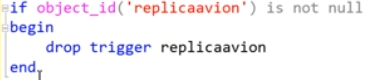




Usa print el loco este.

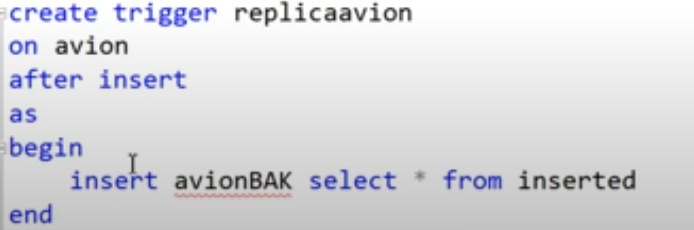
Luego usa for insert, update, los 2 juntos

Usa If para eliminar trigger si existe

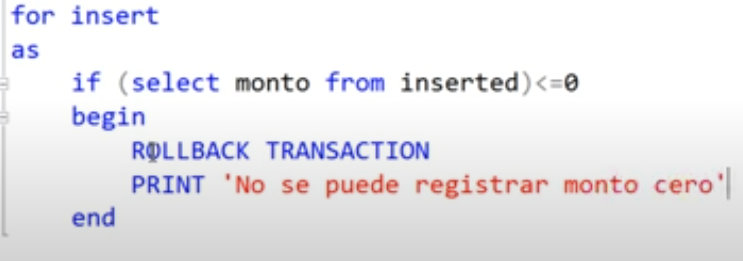


Replica exacta de una tabla a otra. Usa after

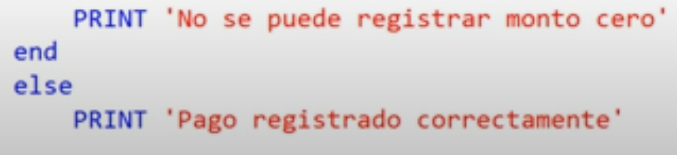
Tiene los mismos campos la tabla 2 que la 1. Hasta se llaman igual



Cancela operación. Controla que el pago sea mayor a 0

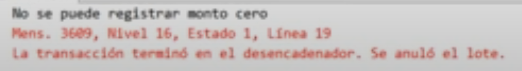


El rollback niega la transacción.



Ahí va otro print.

Resultado fallido.

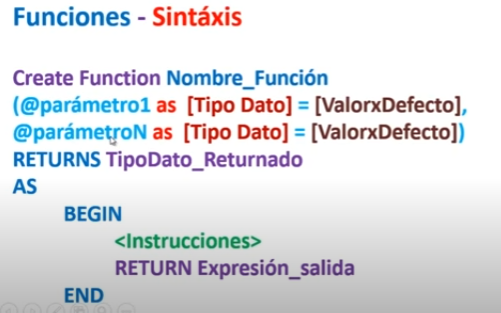


Funciones

<https://www.incanatoit.com/2015/10/funciones-sql-server-2014.html>

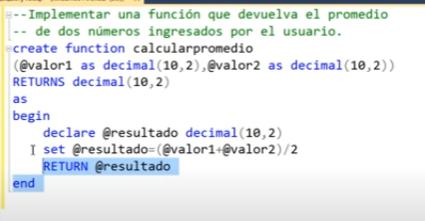
Rutina almacenada que recibe parámetros escalares.

Hay funciones del sistema, así como la posibilidad de crear las propias.

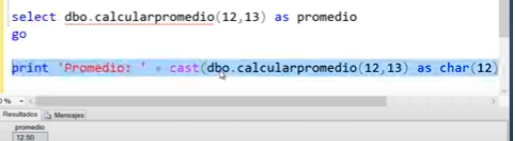


10 es el largo, 2 es la cantidad de decimales máxima.

Tiene valor1 y valor2 como entradas y devuelve un solo valor.

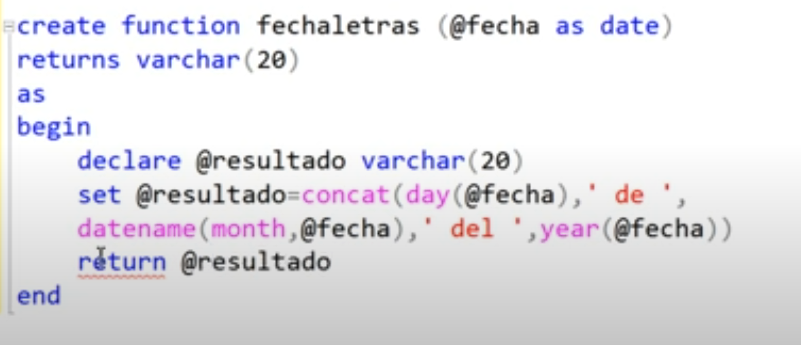


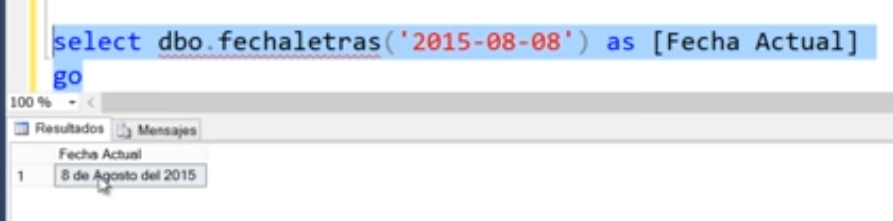
Así la pasó los parámetros de entrada (valor 1 y 2).



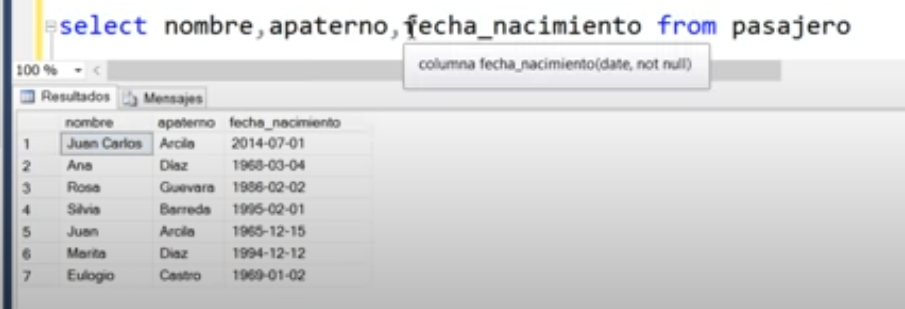
Así printea, dice que hace falta hacer el cast.

Concat. Concatena el día de la variable day, luego deja un espacio en blanco y sigue. Datename te da el nombre, del mes en este caso.

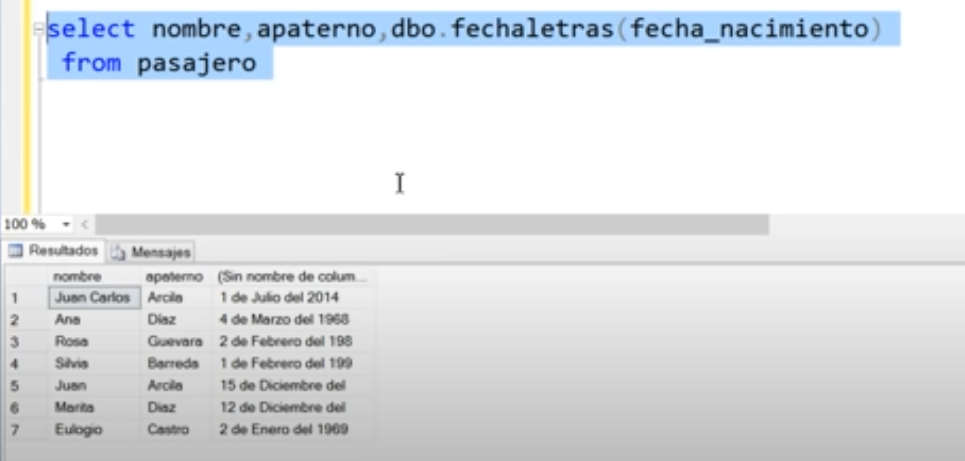




Eso es lo que obtiene. Prolijo, separado y con el mes en palabras.



Hace ese select respecto de una tabla vieja y así se ven las fechas.



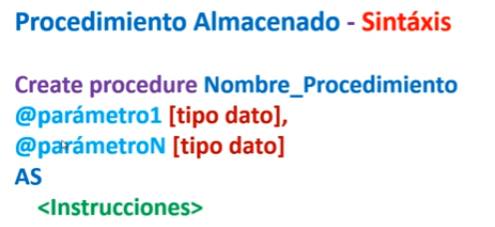
De esa forma llama a la función creada y se modifica el modo en que aparecen las fechas.

Algunas fechas salen cortadas porque les puso el varchar en 20.

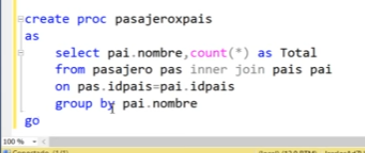
Procedimientos almacenados

<https://www.incanatoit.com/2015/10/procedimientos-almacenados-sql-server-2014.html>

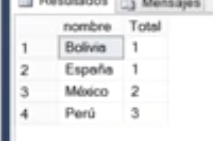
Inician en sp



Select pai( es la tabla).nombre (es el campo),count(\*) (va a contar la cantidad de veces que aparezca un pasajero de ese país) as total (establece alias). Despues usa un inner join.

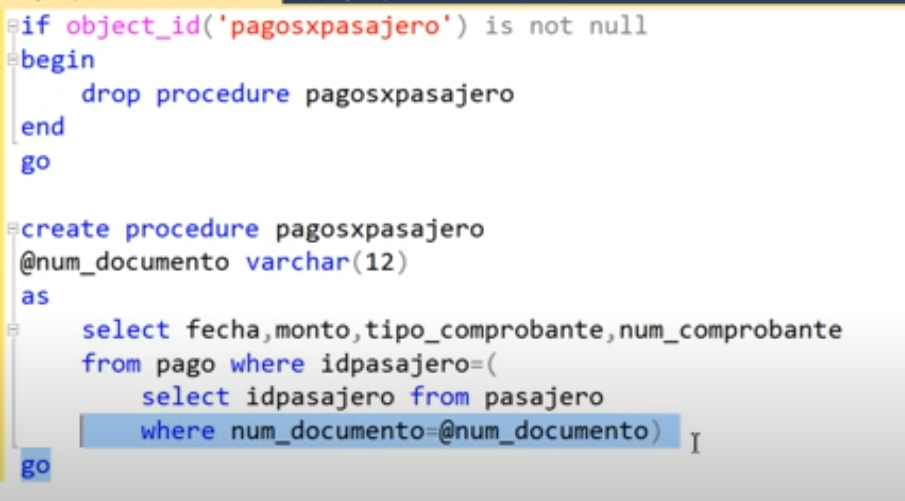


Hace eso. No le pasa parámetros.

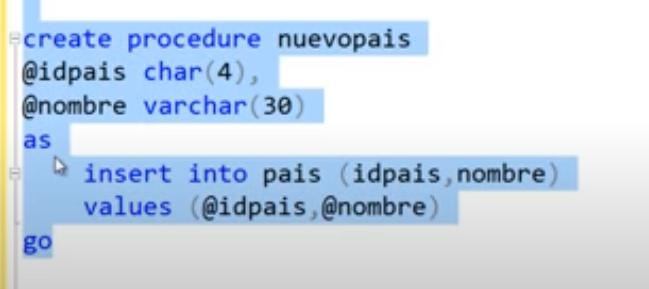


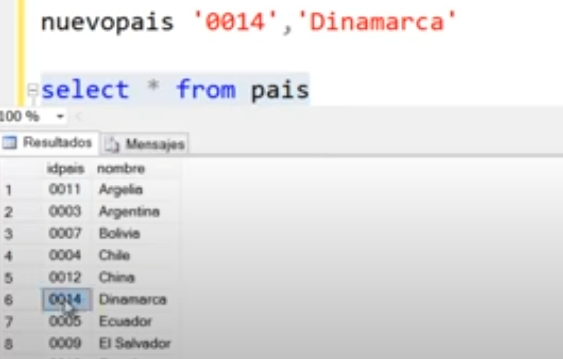
Todo lo que está dentro del paréntesis del Idpasajero es una subconsulta. Sirve para obtener los datos solamente del pasajero cuyo documento se le pasa por parámetro.

El @num\_documento del ppio es lo que se le pasa por parámetro.

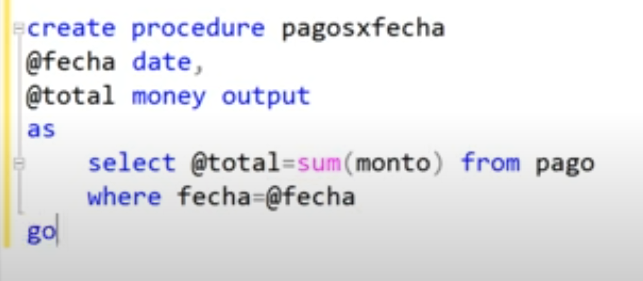


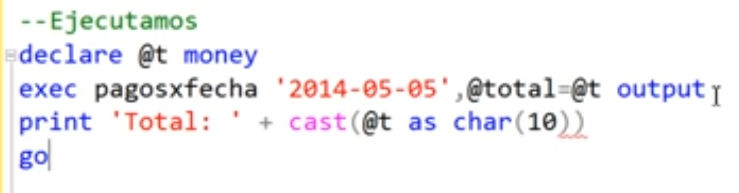
Este otro método vendría a reemplazar a un insert into, no se bien con qué ganancia. Se pasan 2 parámetros los cuales son agregados a una tabla.





Acá money no es un parámetro de entrada sino un output.





Es medio raro esto. Lo que devuelva money es capturado por la variable declarada @t de tipo money, luego eso lo castea a char.