Para la búsqueda solo uso los campos de unidad y fechaposicion. Anexo ejemplo de query para obtener histórico esto es para un rango de 1 de septiembre a hoy 5 de mayo, si la consulta tuviera rangos menores la consulta solo toma en cuenta las tablas que tendrán posibles valores para no comprobar tablas que no.

Esa búsqueda me da 5 millones de registros en 15 segundos, la de ayer de 14 millones es de varias tablas de histórico, pero sin búsqueda como en mis funciones.



**Query de ejemplo de histórico:**

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico20191028 where unidad='ST255' and fechaposicion between '2018-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico2019111 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico2019121 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico202011 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico202021 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico202033 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico202041 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

union all

select ipk,unidad,fechaposicion,latitud,longitud,localizacion,ignicion,CAST(CAST(velocidad AS FLOAT) AS bigint),placas,imei,evento,descrip\_evento,shipment,cliente,cliente\_nombre,tipo,odometro,fecha\_sistema,id\_prove,levelbat,volbat from historico202051 where unidad='ST255' and fechaposicion between '2019-09-01 00:00' and '2020-05-05 23:59'

order by fechaposicion desc

**/\*Índices generados\*/**

**Los índices que por ahora se generan son simples de dos campos correspondientes a unidad y fecha de posición.**

create index indexhistorico20191028 on historico20191028 (unidad,fechaposicion)

create index indexhistorico2019111 on historico2019111 (unidad,fechaposicion)

create index indexhistorico2019121 on historico2019121 (unidad,fechaposicion)

create index indexhistorico202011 on historico202011 (unidad,fechaposicion)

create index indexhistorico202021 on historico202021 (unidad,fechaposicion)

create index indexhistorico202033 on historico202033 (unidad,fechaposicion)

create index indexhistorico202041 on historico202041 (unidad,fechaposicion)

create index indexhistorico202051 on historico202051 (unidad,fechaposicion)

**Sentencia para eliminar indice**

drop index name\_index on name\_tabla

/\*Separacion de cadenas en sql server\*/

ALTER FUNCTION [dbo].[VerticesSeparados]( @Split VARCHAR(MAX) )

RETURNS

@retorno TABLE ([Coordenadas] [nvarchar] (500))

AS

BEGIN

DECLARE @coordenada NVARCHAR(255)

DECLARE @posicion INT

WHILE CHARINDEX('&', @Split) > 0

BEGIN

SELECT @posicion = CHARINDEX('&', @Split)

SELECT @coordenada = SUBSTRING(@Split, 1, @posicion-1)

INSERT INTO @retorno

SELECT @coordenada

SELECT @Split = SUBSTRING(@Split, @posicion+1, LEN(@Split)-@posicion)

END

INSERT INTO @retorno

SELECT @Split

RETURN

END

**Ejecucion de funcion**

DECLARE @Varvertice NVARCHAR(Max)

SET @Varvertice = (SELECT REPLACE(vertices, ',',' ') as [Vertices] FROM tbl\_puntos\_interes WHERE Id = 100)

PRINT @Varvertice

SELECT \* FROM dbo.VerticesSeparados(@Varvertice)

/\*el replace es opcional para la parte geométrica de polígonos en coordenadas euclideano (plano)\*/

**/\*Funciones geométricas en sql\*/**

/\*el replace es opcional para la parte geométrica de polígonos en coordenadas euclideano (plano)\*/

**Referencia de manual de sql**

DECLARE @g geometry;

DECLARE @h geometry;

SET @g = geometry::STGeomFromText('POLYGON((0 0, 2 0, 2 2, 0 2, 0 0))', 0);

SET @h = geometry::STGeomFromText('POINT(1 1)', 0);

SELECT @g.STContains(@h);

**Aplicado para comprobación coordenadas del servidor\_1**

DECLARE @g geometry;

DECLARE @h geometry;

SET @g = geometry::STGeomFromText('POLYGON((20.564372 -100.286640

20.564411 -100.286750,20.564599 -100.286767,

20.564678 -100.286639,20.565059 -100.286698,

20.565005 -100.286849,20.565389 -100.286908,

20.565487 -100.287025,20.565530 -100.287154,

20.565297 -100.289242,20.563890 -100.289035,

20.564197 -100.286621,20.564372 -100.286640))', 0);

SET @g = @g.MakeValid();

SET @h = geometry::STGeomFromText('POINT(20.564725 -100.287959)', 0);

SELECT @g.STContains(@h);

/\*Revisar que tome los valores la tabla de puntos de interés columna de vértices, por lo que tal vez solo se debe hacer un REPLACE ya que se valida los datos sean getricos con @g.MakeValid y ver el uso múltiples polígonos\*/