

C.5 Laboratorio Semana 5- Angel Gabriel Ortega Corzo

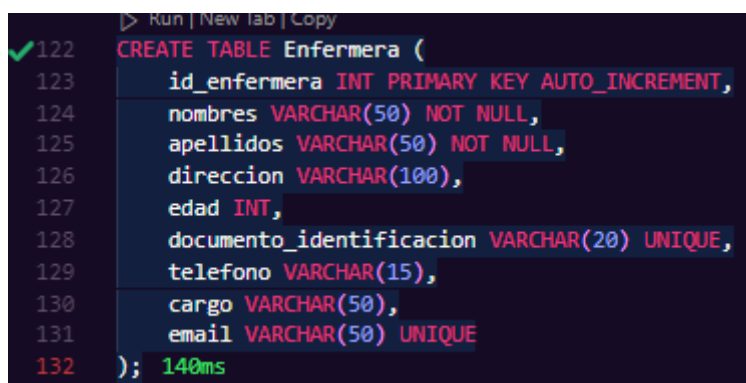
La indexación de bases de datos es uno de los componentes más críticos para mejorar el rendimiento y la velocidad de las consultas. Si no se indexan adecuadamente las tablas de la base de datos, las consultas pueden volverse lentas e ineficientes. Explicaremos qué es la indexación de bases de datos, por qué es importante y cómo se puede aplicar de manera efectiva para optimizar el rendimiento de las consultas. :

Indexación en Base de datos relacionales

Cree la tabla Enfermera y realizar las siguientes tareas usando comandos SQL.

De esta manera considero que podria quedar la tabla enfermera, no veo necesario por el momento añadirla a alguna otra tabla, por lo mismo tendria la siguiente estructura

```
CREATE TABLE Enfermera (  
    id_enfermera INT PRIMARY KEY AUTO_INCREMENT,  
    nombres VARCHAR(50) NOT NULL,  
    apellidos VARCHAR(50) NOT NULL,  
    direccion VARCHAR(100),  
    edad INT,  
    documento_identificacion VARCHAR(20) UNIQUE,  
    telefono VARCHAR(15),  
    cargo VARCHAR(50),  
    email VARCHAR(50) UNIQUE  
);
```



```
> Run | New tab | Copy  
✓ 122 CREATE TABLE Enfermera (  
123     id_enfermera INT PRIMARY KEY AUTO_INCREMENT,  
124     nombres VARCHAR(50) NOT NULL,  
125     apellidos VARCHAR(50) NOT NULL,  
126     direccion VARCHAR(100),  
127     edad INT,  
128     documento_identificacion VARCHAR(20) UNIQUE,  
129     telefono VARCHAR(15),  
130     cargo VARCHAR(50),  
131     email VARCHAR(50) UNIQUE  
132 ); 140ms
```

- Crear un procedimiento almacena para poder registrar mediante una iteracion 10000 datos.

Para poder crear esto podemos hacer uso de un while:

```

CREATE PROCEDURE InsertarEnfermeras()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= 10000 DO
        INSERT INTO Enfermera (nombres, apellidos, direccion, edad,
documento_identificacion, telefono, cargo, email)
        VALUES (
            CONCAT('Nombre', i),
            CONCAT('Apellido', i),
            CONCAT('Direccion ', i),
            FLOOR(18 + (RAND() * (60 - 18))),
            CONCAT('DOC-', i),
            CONCAT('999-999-', i),
            'Enfermera',
            CONCAT('email', i, '@hospitalx.com')
        );
        SET i = i + 1;
    END WHILE;
END

```

Luego solo tendríamos que llamarlo y se deberían insertar correctamente, puede tardar un tiempo desde luego.

```

21
22 CALL InsertarEnfermeras(); AffectedRows: 10000 23.4s
23

```

podemos ver que la cantidad de registros es correcta.

	id_enfermera int(11)	nombres varchar(50)	apellidos varchar(50)	direccion varchar(100)	edad int(11)	documento_identificac varchar(20)	telefono varchar(15)	cargo varchar(50)	email varchar(50)
1		Nombre1	Apellido1	Direccion 1	43	DOC-1	999-999-1	Enfermera	email1@hospitalx.com
2		Nombre2	Apellido2	Direccion 2	39	DOC-2	999-999-2	Enfermera	email2@hospitalx.com
3		Nombre3	Apellido3	Direccion 3	47	DOC-3	999-999-3	Enfermera	email3@hospitalx.com
4		Nombre4	Apellido4	Direccion 4	19	DOC-4	999-999-4	Enfermera	email4@hospitalx.com
5		Nombre5	Apellido5	Direccion 5	18	DOC-5	999-999-5	Enfermera	email5@hospitalx.com
6		Nombre6	Apellido6	Direccion 6	18	DOC-6	999-999-6	Enfermera	email6@hospitalx.com

- Utilizando sentencias sql agregar una restricción “Primary key” sobre el campo id_enfermera

Desde el inicio como añadi la sentencia **PRIMARY KEY** esta restricción ya estaría añadida, podemos confirmarlo consultando el index con un simple **DESCRIBE**

DESCRIBE Enfermera

	Field varchar	Type blob	Null varchar	Key varchar	Default blob	Extra varchar
>	id_enfermera	int(11)	NO	PRI	(NULL)	auto_increment
>	nombres	varchar(50)	NO		(NULL)	
>	apellidos	varchar(50)	NO		(NULL)	
>	direccion	varchar(100)	YES		(NULL)	
>	edad	int(11)	YES		(NULL)	
>	documento_identificacion	varchar(20)	YES	UNI	(NULL)	
>	telefono	varchar(15)	YES		(NULL)	
>	cargo	varchar(50)	YES		(NULL)	
>	email	varchar(50)	YES	UNI	(NULL)	

- Identificar y crear índices en los atributos que vea usted necesario que tenga un Índice.

los mas importantes para mi seria por id y por documento de identificacion.

```
CREATE INDEX idx_documento_identificacion ON
Enfermera(documento_identificacion);
CREATE INDEX idx_enfermera ON Enfermera(id_enfermera);
```

```
28 Run | New Tab
29 CREATE INDEX idx_documento_identificacion ON Enfermera(documento_identificacion);
Run | New Tab
✓ 30 CREATE INDEX idx_enfermera ON Enfermera(id_enfermera); 34ms ...
```

- Crear un índice compuesto sobre los campos “nombre” y “apellido” (Investigacion)

esto es bastante sencillo solo tenemos que hacer un índice cualquiera y pasamos los 2 campos que estamos interesados, en este caso nombre y apellido

```
CREATE INDEX idx_nombre_apellido ON Enfermera(nombres, apellidos);
```

```
31 Run | New Tab
✓ 32 CREATE INDEX idx_nombre_apellido ON Enfermera(nombres, apellidos); 84ms
```

- Crear un índice compuesto sobre tres campos en la tabla Enfermera (Investigacion).

podemos unificarlos en este punto de manera bastante sencilla.

```
CREATE INDEX idx_nombre_apellido_id ON Enfermera(nombres, apellidos,
id_enfermera);
```

```
33
Run | New Tab
✓ 34 CREATE INDEX idx_nombre_apellido_id ON Enfermera(nombres, apellidos, id_enfermera); 44ms ...
```

- Listar todos los índices de la tabla Enfermera.

con un show podemos listarlo de manera sencilla para poder visualizar todos los campos que estan asignados a un indice.

SHOW INDEX FROM enfermera

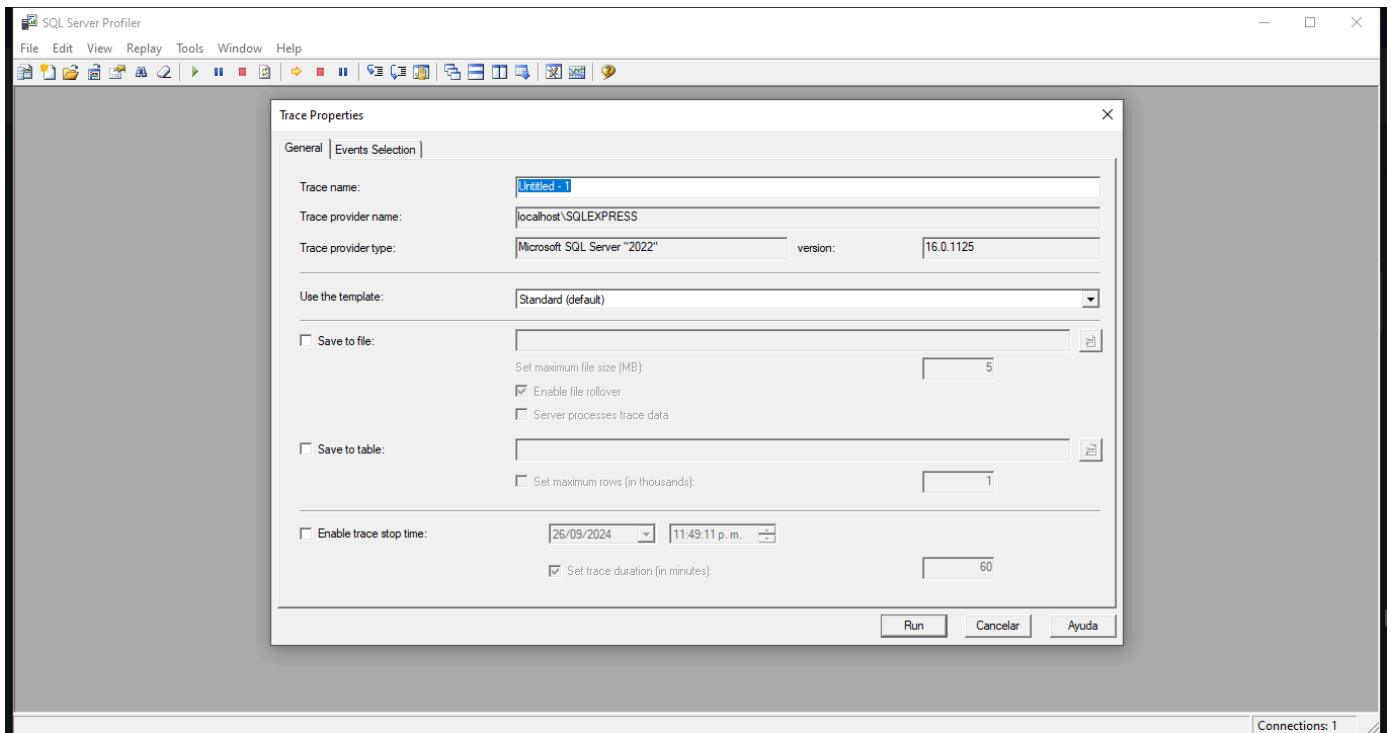
	Table varchar	Non unique bigint	Key_name varchar	Seq_in_index bigint	Column_name varchar	Collation varchar	Cardinality bigint	Sub_part bigint	Packed varchar	Null varchar
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
	> enfermera	0	PRIMARY	1	id_enfermera	A	9760	(NULL)	(NULL)	
	> enfermera	0	documento_identificacion	1	documento_identificacion	A	9760	(NULL)	(NULL)	YES
	> enfermera	0	email	1	email	A	9760	(NULL)	(NULL)	YES
	> enfermera	1	idx_documento_identificac	1	documento_identificacion	A	9760	(NULL)	(NULL)	YES
	> enfermera	1	idx_enfermera	1	id_enfermera	A	9760	(NULL)	(NULL)	
	> enfermera	1	idx_nombre_apellido	1	nombres	A	9760	(NULL)	(NULL)	
	> enfermera	1	idx_nombre_apellido	2	apellidos	A	9760	(NULL)	(NULL)	
	> enfermera	1	idx_nombre_apellido_id	1	nombres	A	9760	(NULL)	(NULL)	
	> enfermera	1	idx_nombre_apellido_id	2	apellidos	A	9760	(NULL)	(NULL)	
	> enfermera	1	idx_nombre_apellido_id	3	id_enfermera	A	9760	(NULL)	(NULL)	

SQL Server Profiler

- Cree un rastro (trace), utilizando al menos tres templates

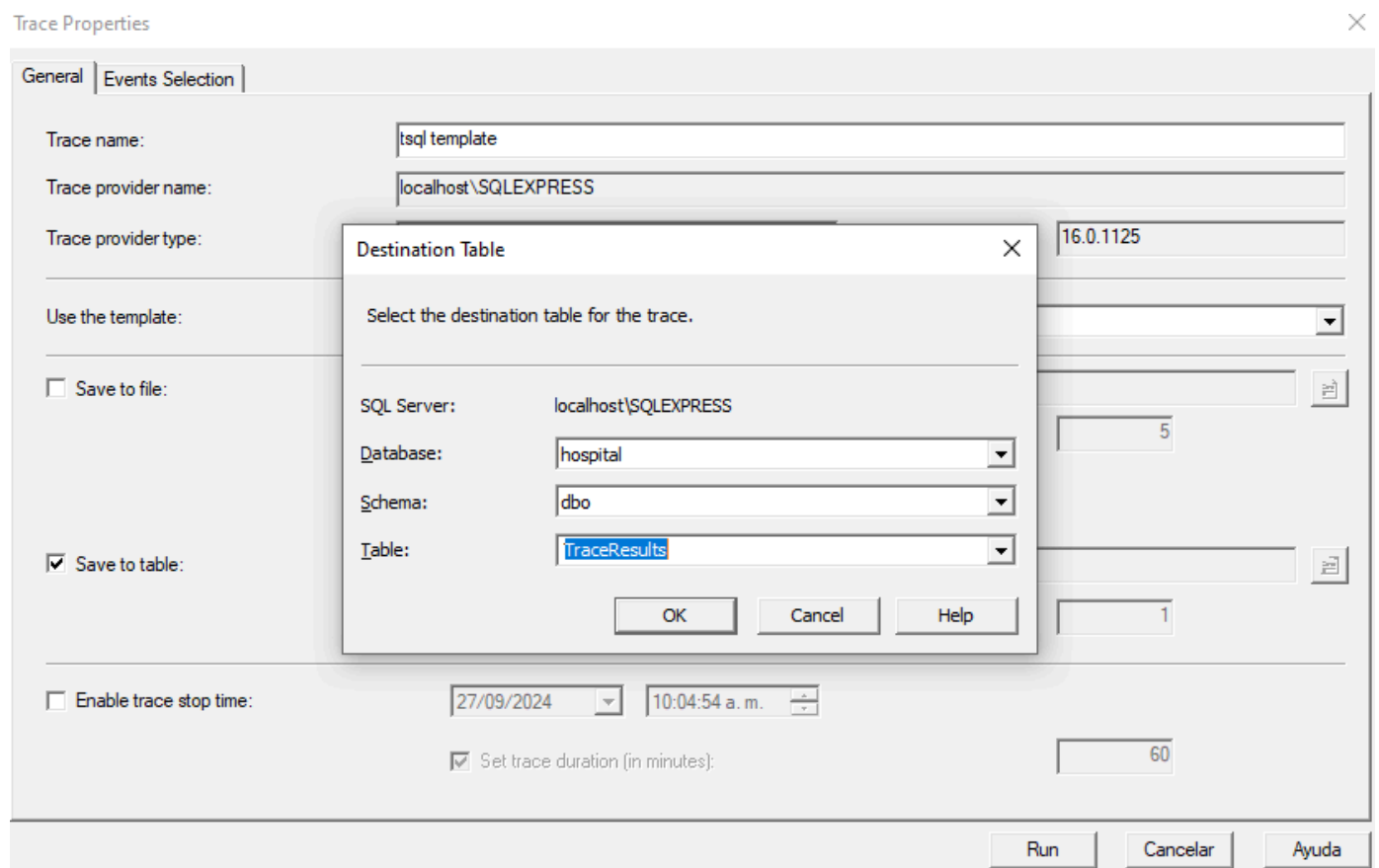
para tener la necesidad de realizar un trace tendríamos que tener una consulta muy mal optimizada y se necesite hacerle un seguimiento, en este caso haremos un trace a tres templates, lo primero que nos pedira es la instancia, creare una tabla para esto

```
CREATE TABLE TraceResults (
    TraceID INT PRIMARY KEY IDENTITY(1,1),
    ActionType VARCHAR(50),
    TableName VARCHAR(50),
    RecordID INT,
    ActionTimestamp DATETIME DEFAULT GETDATE(),
    OldValue TEXT,
    NewValue TEXT
);
```



luego crearemos el trace respectivo, en mi caso quiero crear, un tsql.

podemos ver que nos deja seleccionar una tabla destino y podemos dar solucion al punto "Guarde los resultados del rastro en una tabla de base de datos (Investigacion)"" de una vez, la table el la crea de una vez lo cual nos ahorramucho tiempo



ademas podemos asignarle las acciones que deseamos que raastree, en este caso dejaremos todas

Events	TextData	SPID	StartTime	BinaryData
Security Audit				
<input checked="" type="checkbox"/> Audit Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Audit Logout		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Sessions				
<input checked="" type="checkbox"/> ExistingConnection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Stored Procedures				
<input checked="" type="checkbox"/> RPC:Starting	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL				
<input checked="" type="checkbox"/> SQL:BatchStarting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

y procedemos a crear otros 2, yo cree uno tuning y otro standard

tuning

el tuning realiza una especie de seguimiento en cuanto a la carga de las consultas en el servidor

EventClass	TextData	Duration	SPID	DatabaseID	DatabaseName	Object Type	LoginName
SP:StmtCompleted	SELECT c1mns.name AS [Name], c1mns.c...	1	53	6	hospital	20816 - PQ	angelroot
RPC:Completed	exec sp_executesql N'SELECT c1mns.na...	1	53	6	hospital		angelroot
RPC:Completed	exec sp_reset_connection	0	53	6	hospital		angelroot
SQL:BatchCompleted	use [hospital]	0	53	6	hospital		angelroot
SP:StmtCompleted	SELECT c1mns.name AS [Name], c1mns.c...	0	53	6	hospital	20816 - PQ	angelroot
RPC:Completed	exec sp_executesql N'SELECT c1mns.na...	1	53	6	hospital		angelroot
RPC:Completed	exec sp_reset_connection	0	53	6	hospital		angelroot
SQL:BatchCompleted	use [hospital]	0	53	6	hospital		angelroot
SP:StmtCompleted	SELECT c1mns.name AS [Name], c1mns.c...	0	53	6	hospital	20816 - PQ	angelroot
RPC:Completed	exec sp_executesql N'SELECT c1mns.na...	1	53	6	hospital		angelroot
SQL:BatchCompleted	Use hospital SELECT * FROM TraceR...	387	52	6	hospital		angelroot

Use hospital
SELECT * FROM TraceResults

Trace is running. Ln 253, Col 1 Rows: 253

standard

el standard al registrar los audit de login y logout al parecer identifco solo el logout que realice cuando me sali del sql profiler

Displaying 1 Events

event_id	name	[TextData]	client_app_name	nt_username	cpu_time
21	logout	NULL	SQLServerCEIP	NT SERVICE\S...	0

< >

Event:logout (2024-09-27 11:12:57.9255881)

Details

Field	Value
attach_activity_id.g...	6AA1B78A-B5C9-4199-BC4B-AF9B205C0F62
attach_activity_id.s...	1
attach_activity_id_...	5D76749C-41B1-4A50-A8CB-3EDA8BD05347
attach_activity_id_...	0
client_app_name	SQLServerCEIP
client_pid	2828
cpu_time	0
duration	279720000
event_sequence	21
is_cached	False
is_dac	False
is_recovered	False
logical_reads	704
nt_username	NT SERVICE\SQLTELEMETRY\$SQLEXPRESS
page_server_reads	0

- Observe los resultados del rastro mientras se está ejecutando

cuando ejecutamos el trazo genera un codigo binario conforme avanza su ejecucion

tsql template (localhost\SQLEXPRESS)						
EventClass	TextData	SPID	StartTime	BinaryData		
Trace Skipped Records						
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:37:35...	0X00000...		
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:37:35...	0X00000...		
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:37:35...	0X00000...		
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:37:35...	0X00000...		
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:37:35...	0X00000...		
Trace Skipped Records						
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:38:35...	0X00000...		
Trace Skipped Records						
RPC:Starting	exec sp_cursor 180150003,4,0,N'[dbo]...	52	2024-09-27 09:38:55...	0X00000...		

si hacemos una consulta a la tabla podemos ver que se estan guardando los resultados en el caso del tsql que le asignamos una tabla destino

Use hospital

SELECT * FROM TraceResults

100 %

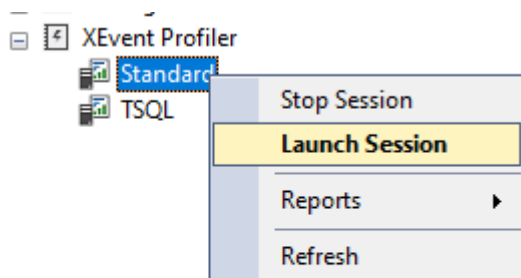
Results Messages

	RowNumber	EventClass	TextData	SPID	StartTime	BinaryData
1	0	65528	NULL	NULL	NULL	0xFFFE900209004D006900630072006F00730
2	1	65534	NULL	NULL	2024-09-27 09:37:34.657	NULL
3	2	17	-- network protocol: LPC set quoted_identifier on set ...	51	2024-09-27 09:34:30.377	NULL
4	3	17	-- network protocol: LPC set quoted_identifier on set ...	67	2024-09-27 09:04:48.660	NULL
5	4	17	-- network protocol: LPC set quoted_identifier on set ...	73	2024-09-27 08:45:14.037	NULL
6	5	11	exec sp_executesql N'exec sp_trace_getdata @P1, 0...	67	2024-09-27 09:37:34.667	0x00000000030000001A00730070005F00650
7	6	14	-- network protocol: LPC set quoted_identifier on set ...	52	2024-09-27 09:37:34.883	NULL
8	7	13	select * from [dbo].[TraceResults]	52	2024-09-27 09:37:34.883	NULL
9	8	13	DROP TABLE [dbo].[TraceResults]	52	2024-09-27 09:37:34.903	NULL
10	9	13	CREATE TABLE [dbo].[TraceResults] ([RowNumber] i...	52	2024-09-27 09:37:34.920	NULL
11	10	11	declare @p1 int set @p1=NULL declare @p3 int set...	52	2024-09-27 09:37:34.937	0x00000000050000001A00730070005F00630
12	11	11	exec sp_cursor 180150003,4,0,N'[dbo].[TraceResults]...	52	2024-09-27 09:37:34.960	0x00000000090000001200730070005F00630
13	12	11	exec sp_cursor 180150003,4,0,N'[dbo].[TraceResults]...	52	2024-09-27 09:37:35.000	0x00000000090000001200730070005F00630
14	13	11	exec sp_cursor 180150003,4,0,N'[dbo].[TraceResults]...	52	2024-09-27 09:37:35.003	0x00000000090000001200730070005F00630
15	14	11	exec sp_cursor 180150003,4,0,N'[dbo].[TraceResults]...	52	2024-09-27 09:37:35.003	0x00000000090000001200730070005F00630
16	15	11	exec sp_cursor 180150003,4,0,N'[dbo].[TraceResults]...	52	2024-09-27 09:37:35.007	0x00000000090000001200730070005F00630

Query executed successfully. | localhost\SQLEXPRESS (16.0 ... | angelroot (52) | hospital | 00:00:00 | 68 rows

- Iniciar, detener, pausar y modificar el resultado del rastro

esto podemos hacerlo de manera bastante sencilla desde el ssml solo dar click derecho y seleccionamos si queremos detenerlo, una vez detenido podemos iniciarlo de igual manera



tambien podemos hacerlo desde el sql profiler con el menu de herramientas superior

File Edit View Replay Tools Window Help

tuning (localhost\SQLEXPRESS)

EventClass	TextData	Duration	SPID	DatabaseID	DatabaseName	Object Type	LoginName
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	59	1	master		NT SER...
SQL:BatchCompleted	SELECT target_data FROM sy...	96	59	1	master		NT SER...
RPC:Completed	exec sp_reset_connection	0	59	1	master		NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	59	1	master		NT SER...
SQL:BatchCompleted	if not exists (select * from sys.dm_...	2	59	1	master		NT SER...
RPC:Completed	exec sp_reset_connection	0	59	1	master		NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	59	1	master		NT SER...
SQL:BatchCompleted	SELECT target_data FROM sy...	100	59	1	master		NT SER...
RPC:Completed	exec sp_reset_connection	0	59	1	master		NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	59	1	master		NT SER...
SQL:BatchCompleted	if not exists (select * from sys.dm_...	2	59	1	master		NT SER...

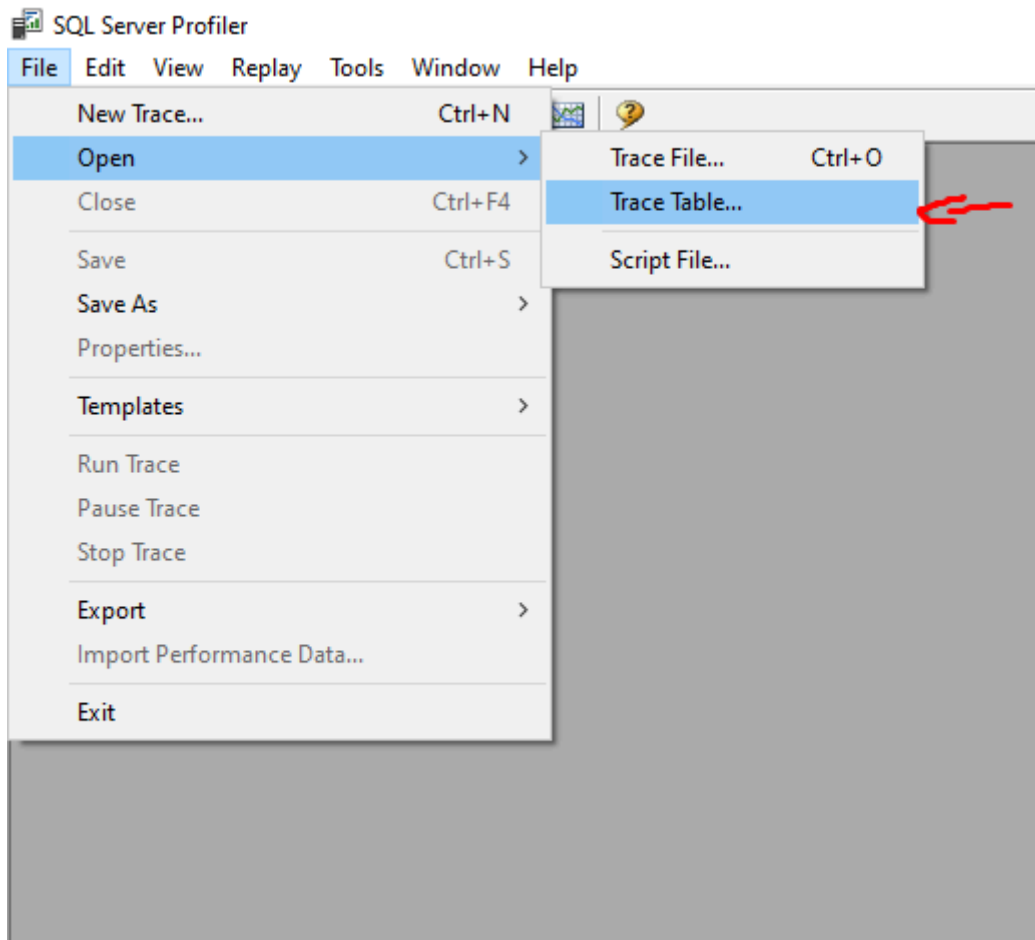
```

if not exists (select * from sys.dm_xe_sessions where name = 'telemetry_xeevents')
    alter event session telemetry_xeevents on server state=start
  
```

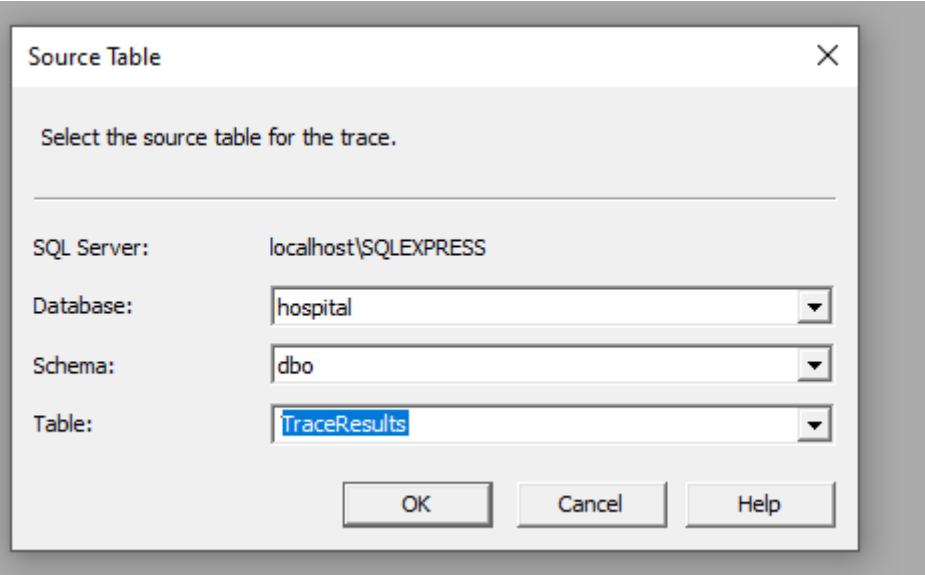
Trace is running. Ln 14, Col 1 Rows: 14

- Reproducir los resultados del rastreo

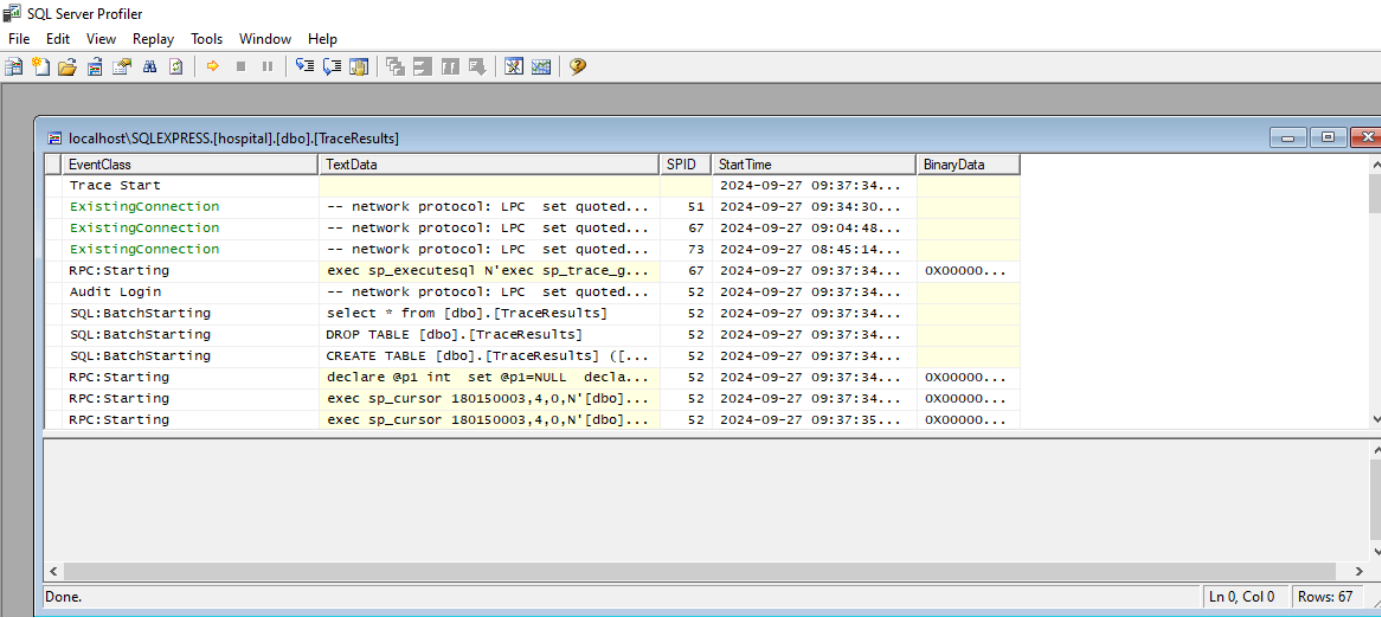
esto lo podemos lograr de manera relativamente facil, primero debemos abrir el trace table que creamos con el trace tsq1 al inicio



luego seleccionamos la tabla donde se estan guardando los registros, este caso llamada trace results



y de esta manera estaríamos reproduciendo el rastreo de manera exitosa y nos lo confirmara con un done



Redactado y escrito por:

angelgabrielortega