

## Borrando datos, tablas y bases

Ya tenemos cubiertos Create, Read y Update, en este documento nos centraremos en la parte de Delete/Destroy. Vamos a aprender, de una forma controlada, como borrar datos de mis tablas y además como eliminar una tabla cuando ya no la necesito.

### Borrando registros, uso del Delete:

Como comenté en el anterior documento, así como el UPDATE, el DELETE no tiene dificultades importantes en la escritura de su sentencia, pero puede convertirse en un riesgo si no se toman los recaudos suficientes y así mitigar posibles errores involuntarios.

La sentencia DELETE nos va a permitir eliminar uno o más registros de una tabla, dependiendo de las condiciones que se incorporen en la misma. También permite eliminar todos los registros de una tabla si no se agrega ninguna condición.

A continuación, vemos la estructura de la sentencia, con sus respectivas cláusulas:

**DELETE FROM**[tabla](1)

**WHERE** [condiciones]; (2)

- 1) En [tabla] irá el nombre de la tabla que deseamos actualizar.
- 2) En [condiciones] luego de la cláusula WHERE pondremos todas las restricciones que requiramos para eliminar solo los registros que nos interesen. Se debe tener cuidado porque sin el WHERE la sentencia elimina todos los registros de la tabla.

Visto que la información eliminada por un DELETE no será fácil de recuperar, veremos algunos ejemplos y siempre procuramos utilizar lo visto en el anterior referido a “Ejecución controlada de Update” aplicándolo de igual forma a los DELETE.

Pasemos a un ejemplo para ver las variantes y como utilizarlo

#### Ejemplo 1

Supongamos que tenemos una tabla Rubro con la siguiente definición:

Rubro	
Pk	Cod_rubro
	Descripcion

Y requerimos eliminar el rubro con descripción “Electrodomésticos”, por no tener artículos disponibles para ese rubro.

Arrancamos por encontrar la consulta que nos permita encontrar el registro a eliminar siguiendo los pasos necesarios.

- 1) Buscamos el domino de mi consulta

**FROM** Rubro

- 2) Validamos si es necesario agregar condiciones a nivel de registro para agregar el **WHERE** en mi consulta. En este caso la descripción debe ser igual a “Electrodomésticos”.

**WHERE** descripcion = 'Electrodomésticos'

- 3) Por último, decidimos que columnas vamos a mostrar para agregar a la cláusula **SELECT**.

**SELECT** \*

Quedando la consulta de esta forma y dándonos el siguiente resultado:

```

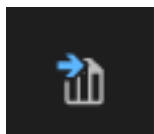
127  SELECT *
128  FROM Rubro
129  WHERE descripcion = 'Electrodomésticos'
130

```

Results		Messages
	COD_RUBRO	DESCRIPCION
1	9	Electrodomésticos

Ahora diseñamos la sentencia de DELETE, pero previo a eso veremos algunos pasos que nos facilitarán la recuperación de un registro en caso de un error involuntario.

Utilizamos la funcionalidad que nos brindan las aplicaciones para extraer la información en formato CSV.



Nos va a guardar un archivo que va a contener la siguiente información:

```
COD_RUBRO,DESCRIPCION
```

```
9,Electrodomésticos
```

Esta información la podemos transformar en un INSERT, para poder crear nuevamente el registro en nuestra base. Los pasos son relativamente fáciles:

- 1) Incorporamos la cláusula INSERT INTO y el nombre de la tabla por delante de los datos de la primera fila encerrando los atributos entre paréntesis

```
INSERT INTO Rubro (COD_RUBRO, DESCRIPCION)
```

```
9,Electrodomésticos
```

- 2) En la segunda fila incorporamos la cláusula VALUES y encerramos los valores entre paréntesis. Si tenemos algún valor que corresponde a cadena de caracteres lo debemos encerrar entre comillas simples.

```
INSERT INTO Rubro (COD_RUBRO, DESCRIPCION)
```

```
VALUES (9,'Electrodomésticos')
```

Y de esta forma obtenemos la sentencia INSERT que nos va a permitir recuperar el valor que hayamos borrado si así lo queremos. En caso de que nuestra consulta retorne más de una fila de valores a partir de la segunda reemplazamos VALUES por una coma como el siguiente ejemplo:

```
INSERT INTO Rubro (COD_RUBRO, DESCRIPCION)
VALUES (9,'Electrodomésticos')
, (12, 'Ejemplo')
```

Ahora si pasemos a la sentencia DELETE, con 2 pasos y haciendo uso de como conformamos la consulta usando el SELECT

- 1) Determinamos el dominio de la sentencia a realizar. Podemos usar el mismo dominio de la consulta.

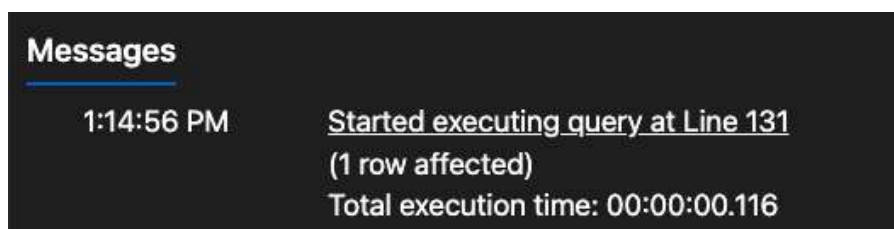
```
DELETE FROM Rubro
```

- 2) Agregamos las condiciones, en caso de no tener no es necesario agregar la cláusula WHERE, pero sabemos que en el DELETE borrará toda la información dentro de Rubro. En nuestro caso tenemos la condición de descripción y la tomamos el mismo WHERE que usamos para la consulta.

```
DELETE FROM Rubro
```

```
WHERE descripcion = 'Electrodomésticos'
```

Tal como ocurre en el UPDATE la devolución del motor es referida a la cantidad de registros que han sido afectados por la sentencia, incluyendo el tiempo que se demoró en la consulta.



Nos resta validar que efectivamente se haya borrado el registro, ejecutando la sentencia SELECT obtenemos el siguiente resultado:

```
127 SELECT *
128 FROM Rubro
129 WHERE descripcion = 'Electrodomésticos'
```

COD_RUBRO	DESCRIPCION
-----------	-------------

En el apartado de mensajes (Messages) se aprecia la leyenda:

Results Messages

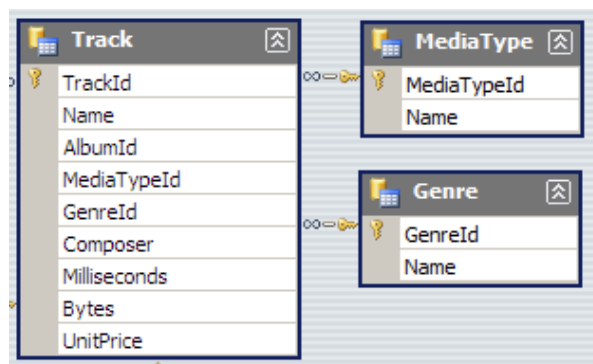
1:15:58 PM Started executing query at Line 127  
(0 rows affected)  
Total execution time: 00:00:00.037

Y de esta forma comprobamos que ya no existe el Rubro en cuestión solicitado por el ejemplo. Y en caso de error tenemos listo el INSERT para poder recuperar dicha información.

Veamos otro ejemplo donde tengamos más de una condición por agregar.

## Ejemplo 2

Dada las tablas Canción (track) y Género (Genre):



Se desea eliminar todas las canciones que pertenezcan al género “K-pop” (no sabemos del todo cómo está escrito en la base) que tenga un valor menor a “5”.

Arrancamos con las consultas que nos permitirán saber las canciones que serán borradas.

Primero descubrimos cuál es el id del género K-pop teniendo en cuenta que no sabemos del todo cómo está guardado en la base, para eso le daremos uso a lo aprendido en los módulos de lectura de registros en la base con caracteres.

```
SELECT *  
  
FROM Genre  
  
WHERE lower(name) like '%k_pop%'
```

Recordemos un poco sobre módulos anteriores:

- Utilizamos la función lower para comparar contra minúsculas.
- Like para comparar cadena de caracteres.
- “%” el comodín porcentaje que nos permite buscar cualquier carácter o espacio.
- “\_” el guion bajo que nos permite buscar un carácter o un espacio.

El resultado es:

Results		Messages
	GenreId ▾	Name ▾
1	26	K-pop

Con el Id de género (GenreId) 26 procedemos a realizar la consulta que nos retorne las canciones de ese género con el valor menor a 5 en la tabla Canción (Track).

```
SELECT *  
  
FROM Track  
  
WHERE GenreId = 26  
  
AND UnitPrice <5
```

El resultado es:

TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer	Milliseconds	Bytes	UnitPrice
3504	Blood Sweat & Tears	348	2	26	BTS	360000	32424243	2.09
3505	Spring Day	348	2	26	BTS	360000	32424243	2.09

No debemos olvidar guardar el archivo CSV o en el formato que nos sea cómodo para poder recuperar la información en caso de inconvenientes.

Ahora podemos proceder a armar el DELETE con la información que obtuvimos y utilizando la sentencia SELECT:

**DELETE FROM** Track

**WHERE** GenreId = 26

**AND** UnitPrice < 5;

Obtenemos el siguiente resultado:

Messages	
4:54:07 PM	Started executing query at Line 169 (2 rows affected) Total execution time: 00:00:00.122

Y comprobamos que los datos han sido eliminados al ejecutar nuevamente el SELECT sobre la tabla de canciones (Track).

163	SELECT *
164	FROM Track
165	WHERE GenreId = 26
166	AND UnitPrice < 5
167	
Results	
TrackId	Name

La sentencia DELETE no tiene muchas más complicaciones de las que hemos visto y, por lo general, es poco usada cuando se integra a desarrollos WEB, ya que se suele no borrar registros sino más bien actualizarlos (usando UPDATE) cambio su estado de visible a oculto o apagado utilizando una columna para tal fin. Y de esta forma evitamos el borrar involuntariamente algún registro que sea valioso.

### ¿Qué pasa cuando una tabla ya no la requiero, cómo la elimino?

Nos resta ver una de las sentencias que forman parte del agrupamiento DESTROY (en CRUD la D), la sentencia es DROP. DROP es una sentencia poco usada en relación directa con un desarrollo WEB, pero nos será útil cuando queramos responder la pregunta de este tópico con respecto a qué hacer si una tabla ya no la estoy usando por algún motivo.

Veremos el uso de la sentencia abarcando algunas formas que podemos utilizar para resguardar la información a eliminar.

Veamos cómo es la estructura de la sentencia:

**DROP TABLE** [table] (1)

- 1) En [table] completamos directamente con la tabla que queremos eliminar.

Al ejecutar la sentencia la información que se encuentra dentro de la tabla será eliminada junto con la definición de la tabla en nuestra base de datos.

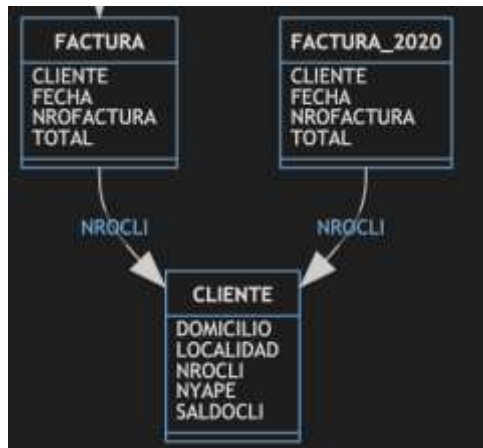
DROP puede ser usado para otros objetos de nuestra base es por esto que se acompaña de la palabra reservada TABLE cuando queremos eliminar una tabla.

Pasemos a ejemplos y entender cómo podemos resguardar nuestra información en caso que, tal como en el DELETE, exista algún involuntario.

#### Ejemplo 3

Suponemos que tenemos una tabla de factura que creamos con la intención de guardar las emitidas en el año 2020 tal como muestra el siguiente esquema:





En la tabla guardamos una copia de las facturas emitidas en el año 2020 que se encontraban en la tabla FACTURA, pero ya no la requerimos.

Vamos a dividir nuestros pasos para ejecutar el DROP en 3:

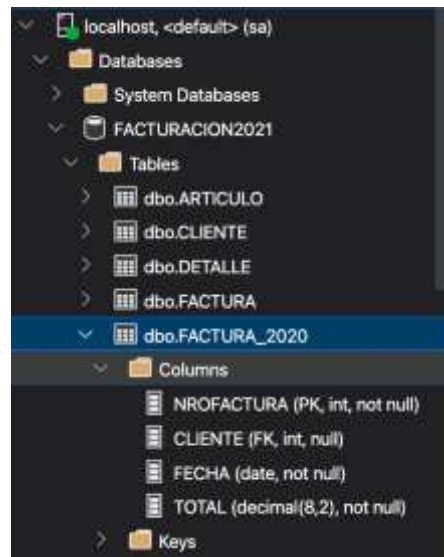
- 1) Backup de la sentencia de creación de la tabla a eliminar.
- 2) Backup de la información que se encuentra en la tabla.
- 3) Ejecución de la sentencia DROP.

### **Paso 1)**

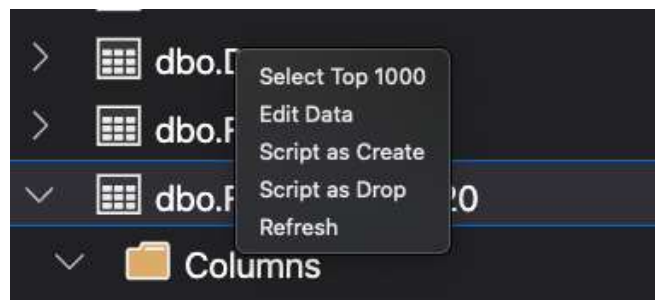
Comenzamos con el paso 1, realizar el backup de la SENTENCIA CREATE de la tabla.

La mayoría de los clientes que utilicemos tendrán una forma de obtener dicho archivo en el video de este módulo se muestra una forma utilizando MySQL como Base de Datos y MySQL Workbench como aplicación. En este documento veremos cómo hacerlo cuando contamos con un SQL Server como base de datos y la aplicación Azure Data Studio.

La aplicación nos permite ver en la barra lateral izquierda las conexiones disponibles, cuando desplegamos nos permite ver, dentro de la conexión, las bases de datos disponibles y al desplegar alguna de ellas podemos ver las tablas tal como se muestra en la siguiente imagen:



Vemos las columnas que contiene la tabla FACTURA\_2020, cuando hacemos un clic derecho sobre la tabla nos aparecen las siguientes opciones:



Al seleccionar “Script as Create” nos permitirá obtener la sentencia de CREATE que se debe utilizar para crear la tabla.

```
CREATE TABLE #m (FACTURA_2020)
(
  PROFACTURA #m NOT NULL,
  CLIENTE #m NULL,
  FECHA #m NOT NULL,
  TOTAL #decimal(7) NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE #m (FACTURA_2020) ADD PRIMARY KEY CLUSTERED
(
  PROFACTURA #m
)
```

```

)WITH (PAD_INDEX=OFF, STATISTICS_NORECOMPUTE=OFF, SORT_IN_TEMPDB=OFF, IGNORE_DUP_KEY=OFF,
ONLINE=OFF, ALLOW_ROW_LOCKS=ON, ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
GO
ALTER TABLE [dbo] [FACTURA_2020] ADDDEFAULT (getdate()) FOR [FECHA]
GO
ALTER TABLE [dbo] [FACTURA_2020] ADDDEFAULT ((0)) FOR [TOTAL]
GO
ALTER TABLE [dbo] [FACTURA_2020] WITHCHECKADDFOREIGN KEY([CLIENTE])
REFERENCES [dbo] [CLIENTE] ([NROCLI])
GO

```

Al guardar esta información nos garantizamos el poder volver a crear la tabla sin problemas.

## Paso 2)

Ahora debemos realizar un backup de la información que contiene la tabla. Para esto utilizaremos lo aprendido con UPDATE y DELETE, en este caso bastará con realizar un SELECT de toda la tabla sin aplicar ningún tipo de condición de la siguiente forma:

**SELECT \***

**FROM FACTURA\_2020**

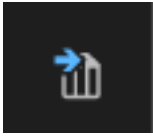
185    **Select \***  
186    **From FACTURA\_2020**

	NROFACTURA	CLIENTE	FECHA	TOTAL
1	1001	69	2020-02-02	42.00
2	1002	109	2020-02-02	50.00
3	1003	138	2020-02-04	1020.00
4	1004	160	2020-02-04	246.00
5	1005	179	2020-02-06	9242.20
6	1006	194	2020-02-06	25.00
7	1007	207	2020-02-08	600.00

n 186, Col 18   Spaces: 4   UTF-8   LF   SQL   49 rows   Choose SQL Language

Obteniendo de esta forma la totalidad de los registros que se encuentran en la tabla.

Debemos utilizar el icono que nos permite guardar la información como CSV:



Luego de que tengamos los registros en formato CSV podremos aplicar lo aprendido para la sentencia DELETE, incorporando algunas cláusulas en cada línea y el archivo se convertirá en un script de creación de registros en mi base.

### Paso 3)

Ahora podemos ejecutar el DROP sin problemas sabiendo que en caso de algún inconveniente vamos a contar con la información necesaria para recuperar la tabla y la información que se encontraba.

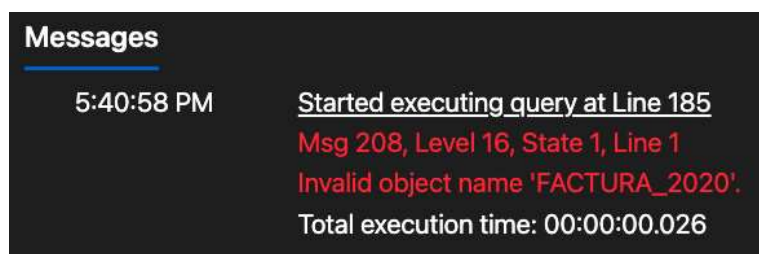
La sentencia quedaría de la siguiente forma:

**DROP TABLE** FACTURA\_2020

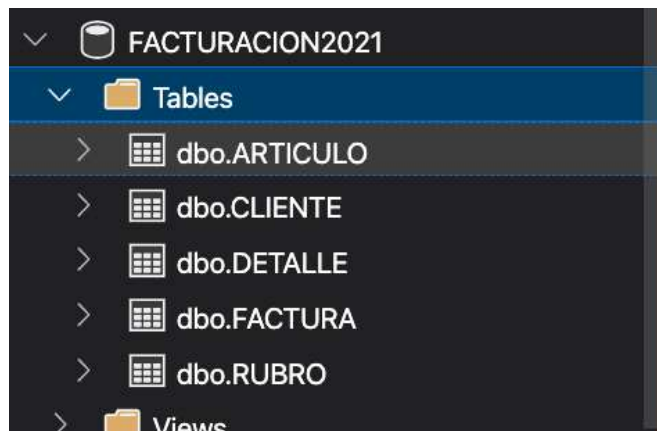
Y este sería el mensaje que nos retorna el motor:



Para comprobar que se haya realizado correctamente podemos ejecutar la consulta que nos permitió obtener todos los datos recibiendo el siguiente resultado:



Y comprobar al refrescar nuestra información de las bases disponibles, que ya no se encuentra la tabla FACTURA\_2020:



La sentencia no es complicada de utilizar, pero si es necesario tomar ciertos recaudos para evitar dolores de cabeza.

Veamos un ejemplo más, pero cuando queremos eliminar más de una tabla.

#### Ejemplo 4

Imaginamos el siguiente esquema, donde hemos duplicado la información de la tabla detalle agregando una llamada DETALLE\_2020 que cuenta con la información de los detalles de ese año y la tabla FACTURA\_2020 que trabajamos en el ejemplo anterior:



Nos interesa eliminar la tabla DETALLE\_2020 y FACTURA\_2020.

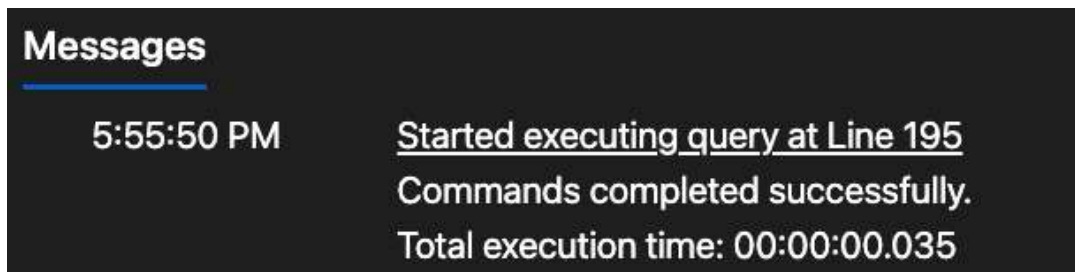
Debemos realizar los pasos 1 y 2 para cada una de las tablas a eliminar, para obtener:

- 1) Backup de la sentencia de creación de la tabla a eliminar.
- 2) Backup de la información que se encuentra en la tabla.

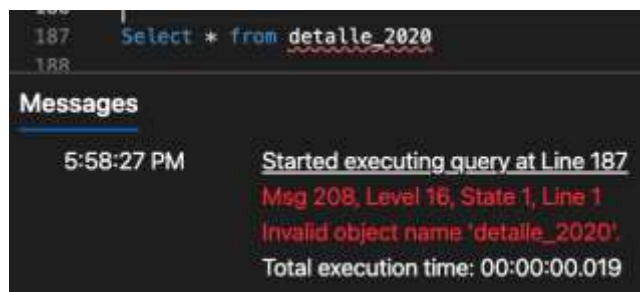
Es repetir de igual forma los que hicimos en el ejemplo anterior, lo que va a cambiar es el paso 3 al conformar el DROP TABLE. La sentencia nos permite eliminar más de una tabla al mismo tiempo separando por comas, quedando:

**DROP TABLE** DETALLE\_2020, FACTURA\_2020;

Tener en cuenta que es necesario ponerlas en orden de referencia, es decir primero las tablas que apuntan a la que va siguiente. En este caso DETALLE\_2020 posee una clave foránea que apunta/referencia a FACTURA\_2020. Y si todo esta correcto recibiremos un mensaje similar al siguiente:



Podremos validar con la ejecución de las sentencias que las tablas ya no existen en nuestra base de datos.



## Eliminando mi base de datos

Llegamos a la última sentencia que veremos en este curso, DROP DATABASE. En el tópico anterior vimos que DROP puede ser usado para borrar una tabla, de igual forma nos permite borrar una base de datos agregándole la palabra reservada DATABASE, la información de la base (tablas, registros, etc.) será eliminada.

Veamos la sentencia y después veremos la mejor forma de resguardar nuestra información por cualquier error involuntario:

**DROP DATABASE** [Database\_name]

En [Database\_name] debemos poner el nombre de la base que deseamos eliminar.

Vamos a ver un ejemplo y de qué forma podemos resguardar previo a ejecutar la sentencia la información que se encuentra en nuestra Base de Datos.

Para tener en cuenta, cada cliente de SQL que utilicemos cuenta con una funcionalidad similar al ejemplo que vamos a ver con Azure Data Studio.

### Ejemplo 5

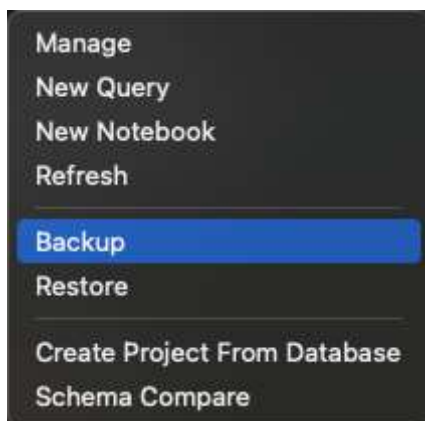
Dada nuestra base de datos FACTURACION2021



Procederemos a eliminarla, pero previo a eso nos encargaremos de resguardar la información necesaria para poder recuperarla.

### **Paso 1 – Backup de la base**

Presionado el botón derecho sobre el nombre de la base de datos se desplegarán las siguientes opciones:



Nos centraremos en backup, que luego de hacer un clic nos permitirá elegir varias opciones:



**Backup database - localhost:FACTURACION2021**

Backup name  
FACTURACION2021-Full-2021-11-14T21:20:08

Recovery model  
Simple

Backup type  
Full

☐ Copy-only backup

Backup files  
/var/opt/mssql/data/FACTURACION2021-20211114-18-20-8.bak

Compression  
Set backup compression  
Use the default server setting

Script Backup Cancel

Para un backup sencillo alcanza con presionar Backup, esto guardará un archivo en la ruta /var/opt/mssql/data/ con el nombre FACTURACION2021-20211114-18-20-8.bak. El nombre contiene la fecha de creación y la hora, lo que nos permitirá reconocerlo rápidamente cuando necesitemos usarlo.

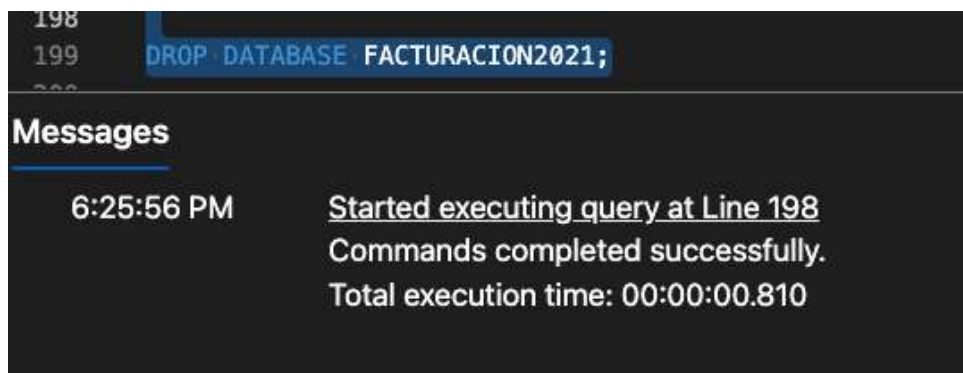
Es una buena práctica realizar un “Backup” periódico de nuestra base de datos, aun cuando no deseemos borrarla, solo por prevención.

## Paso 2 – Borrar la base

Ahora ejecutamos la consulta, debemos hacerlo desde nuestra conexión master es decir sin estar usando la base de datos que vamos a eliminar.

**DROP DATABASE** FACTURACION2021

Y obtenemos el siguiente resultado:

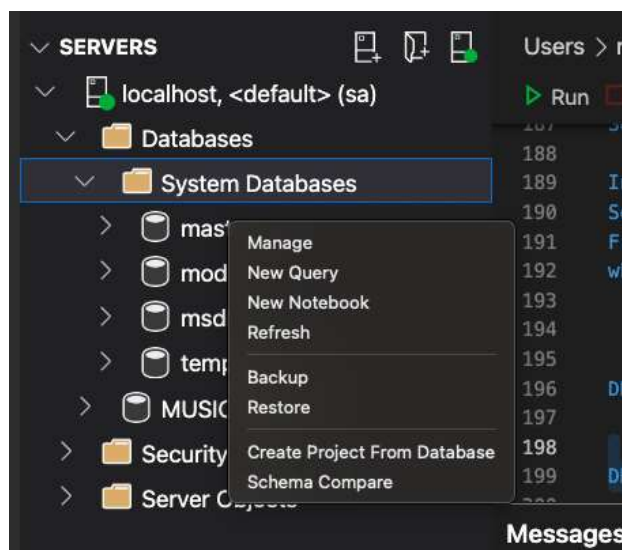


Ya no podremos acceder a nuestra base y al refrescar el árbol de base disponibles ya no estará disponible.

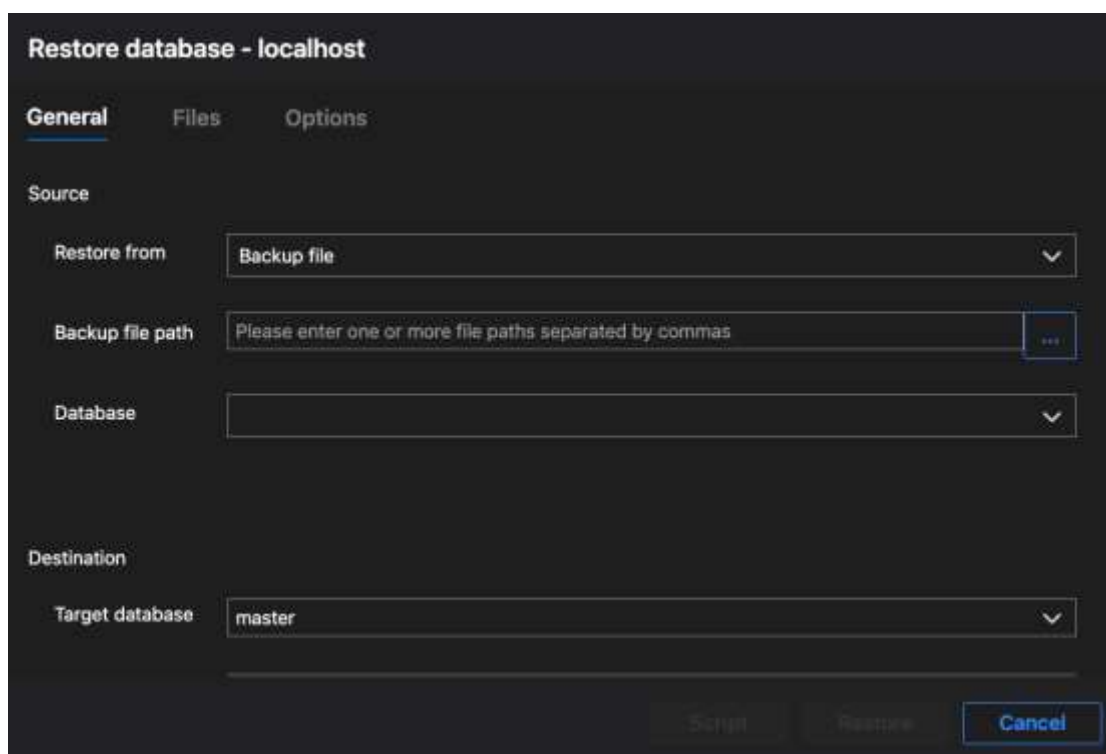
### Paso 3 – Recuperando la base

Este paso incluso lo podemos ejecutar para poder llevarnos la base a otro motor en otra máquina o incluso para compartir la información.

Para recuperar la base alcanza con pararnos en el árbol y presionar el botón derecho sobre la base master (la del sistema).



Nos aparecerá un listado de acciones disponibles, haremos clic en “Restore”.



**Restore database - localhost**

**General** Files Options

Source

Restore from: Backup file

Backup file path: Please enter one or more file paths separated by commas

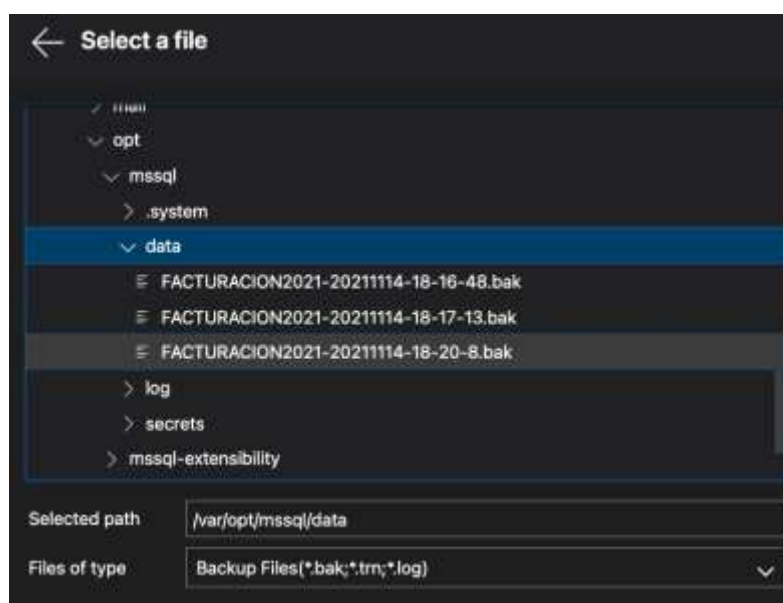
Database:

Destination

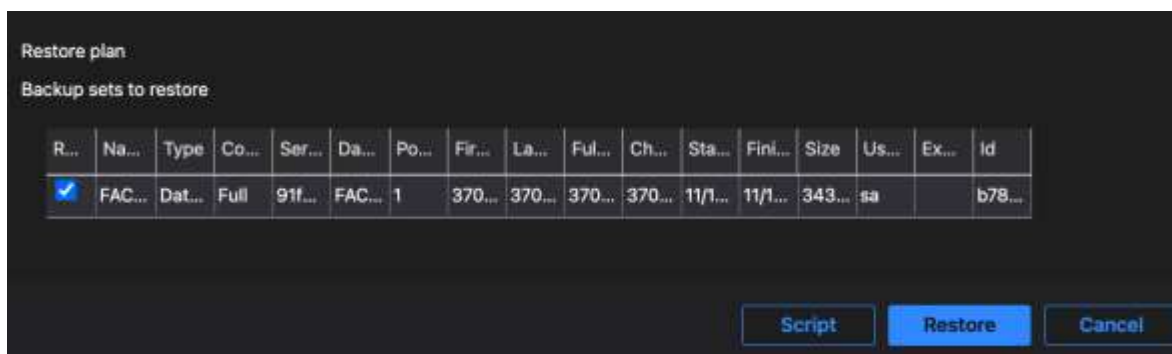
Target database: master

Script Restore Cancel

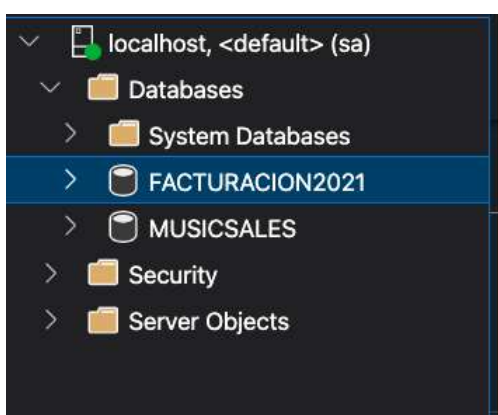
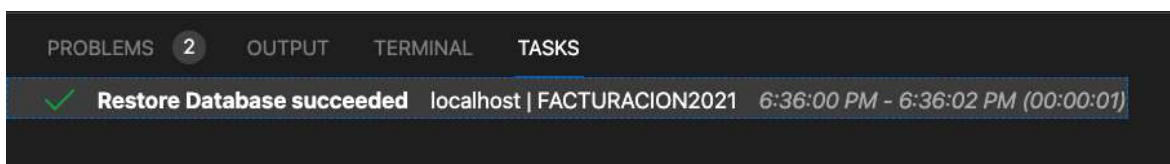
En “Restore from” elegimos “Backup file” y luego hacemos clic en los tres puntos que figuran al final de “Backup file path”.



Si no aparece directamente buscamos la ruta donde se encuentre nuestro archivo de backup que requerimos recuperar. Presionamos “Ok” y nos aparecerá la información de nuestra base de datos a recuperar:



Nos queda presionar “Restore” y podríamos ver nuestra base de datos en el árbol de bases de datos disponibles previo a visualizar un mensaje de que la base se recuperó:



Con esto, hemos visto cómo poder realizar el resguardo (Backup) de nuestra base e incluso poder recuperarla (Restore) en otra instancia de un motor de base de datos similar al que usamos. Cada motor de base de datos tiene una forma de poder realizar el “backup” y “restore” de las bases de datos. En caso de dudas, siempre se puede recurrir a los manuales en línea que posee el producto que estemos utilizando.