

# ADMINISTRACIÓN DE SISTEMAS

## GESTORES DE BB.DD.

### **PROYECTO BB.DD. "BOY SCOUTS"**



AUTOR: Saúl Altoubah León (S.A.L.)

### **SEGURIDAD**

## Índice

- <b>1. Introducción .....</b>	pág. 4
- <b>2. Encriptación (Encryption) .....</b>	pág. 5
- <b>2.1. Tools (VeraCrypt, Let's Encrypt) .....</b>	pág. 6
- <b>2.2. Encriptación de Columnas de BD .....</b>	pág. 9
- <b>2.3. Encriptación de Backup de BD.....</b>	pág. 16
- <b>2.4. Encrip. BD TDE (Transparent Data Encryption).....</b>	pág. 23
- <b>2.5. Funciones .....</b>	pág.29
- <b>2.5.1. DDM (Dynamic Data Masking) .....</b>	pág. 31
- <b>2.5.2. Row Encryption (RLS, Row-Level Security).....</b>	pág. 45
- <b>2.6. Always Encrypted .....</b>	pág. 54
- <b>2.7. Tareas sobre BD en SSMS .....</b>	pág. 65
- <b>2.7.1. Data Discovery and Classification .....</b>	pág. 65
- <b>2.7.2. Vulnerability Assesment.....</b>	pág. 68
- <b>3. Auditoría (Audit).....</b>	pág. 73
- <b>3.1. Auditoría de Serv. y Especif. de auditoría de serv...pág. 74</b>	
- <b>3.2. Especificación de auditoría de BD.....</b>	pág. 79
- <b>3.3. Bonus Auditoría SQL Server .....</b>	pág. 83
- <b>4. Legislación (GDPR – General Data Prot. Reg. (EU)) ....pág. 92</b>	
- <b>5. Ataques .....</b>	pág. 94
- <b>5.1. DDoS .....</b>	pág. 95
- <b>5.2. Injection SQL.....</b>	pág. 97
- <b>5.3. Ransomware.....</b>	pág.108
- <b>5.4. Tools.....</b>	pág. 113
- <b>6. Docker .....</b>	pág. 119
- <b>6.1. Docker aplicado a BD .....</b>	pág. 126



## - **1. Introducción**

En este documento se tratará como una continuación del proyecto “**Boy Scouts**” entrando del proyecto en la parte de **Seguridad**, específicamente, en tratando el tema de **Encriptación**.

En esta segunda parte, se profundizará el tema de la **Encriptación** basada en Bases de datos y se explicará su uso y herramientas utilizando nuestra base de datos para aplicar sus ejemplos como encriptación a nivel de columnas, de copias de seguridad (*backups*), etc.

Se incluirán también métodos de seguridad tales como **Data Masking** y **Row-Level Encryption** para ocultar valores sensibles a nivel de filas.

Serán esenciales hacer uso de las **auditorías** para hacer control de los movimientos de los usuarios de nuestra base de datos, aplicación de la legislación (GDPR) y vulnerabilidades que se presentan en los servidores de bases de datos como **DDoS**, **Injection SQL**...

Y como última inserción será dar una explicación general sobre **Docker**, pero adaptado a bases de datos tanto relacionales como no relacionales (**NoSQL**, como por ejemplo **MongoDB**).

**\*SI SE HACE CLICK EN EL TÍTULO DE LOS APARTADOS, SE PUEDE VOLVER AL ÍNDICE.**

## - 2. Encriptación (*Encryption*)

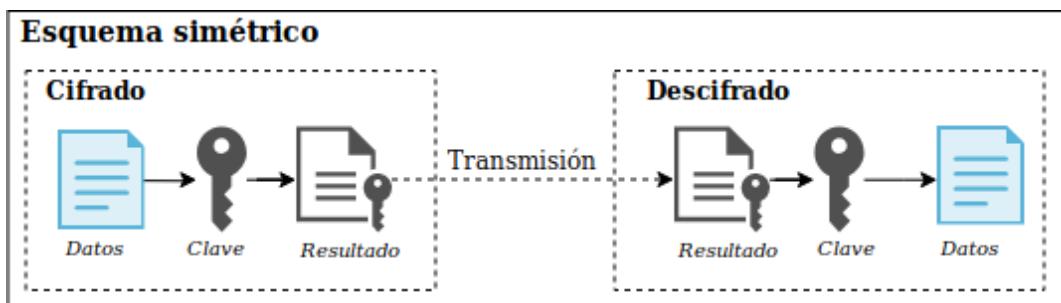
La **encriptación** es el procedimiento de seguridad que consiste en la alteración, mediante algoritmos, de los datos que componen un archivo. El objetivo es hacer que dichos datos se vuelvan ilegibles en caso de que un tercero los intercepte.

La encriptación es un recurso muy utilizado para garantizar una transferencia segura de datos y documentos. Si bien no se puede garantizar que no se sustraiga información sensible, sí puede evitar que se utilice para el perjuicio de sus dueños legítimos.

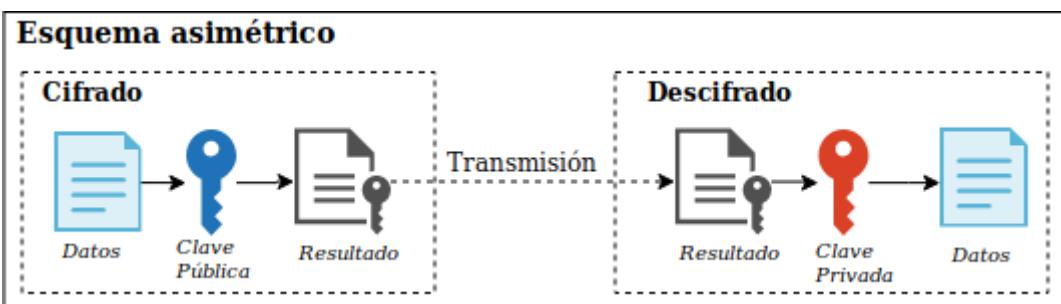
La banca y los comercios online usan la encriptación de datos para evitar el manejo inapropiado de información de sus clientes (números de tarjetas de crédito, información sobre transacciones, datos personales, etc.).

De la misma forma, muchos sistemas de mensajería recurren a esta herramienta para procurar comunicaciones más seguras y evitar que las conversaciones sean interceptadas.

Los métodos de encriptado se clasifican según sus claves y sus algoritmos:



Encriptación simétrica: la encriptación simétrica es aquella donde se utiliza la misma clave tanto para cifrar como para descifrar los datos. Algunos sistemas de encriptación simétricos más populares son **AES** y **Triple DES**.



Encriptación asimétrica: Consta de una clave pública para cifrar y una clave privada para descifrar. Los métodos más conocidos son **ElGamal**, **RSA**.

## • 2.1. Tools (VeraCrypt, Let's Encrypt)

~ VeraCrypt:



**VeraCrypt** es un software de código abierto para cifrar archivos, carpetas, unidades USB extraíbles, discos duros completos e incluso el disco duro donde se encuentra el propio sistema operativo instalado.

**VeraCrypt** es multiplataforma, actualmente es compatible con sistemas operativos **Windows**, cualquier sistema basado en **Linux** y

también es compatible con **macOS**. Este software está basado en el antiguo **TrueCrypt**.

Algunas de las principales características de **VeraCrypt** son las siguientes:

→ Creación de discos cifrados virtuales en un simple archivo: podremos crear un archivo cifrado a modo de contenedor, en el cual esté toda la información importante. Este archivo lo podremos montar para su lectura y escritura con **VeraCrypt**, este método es ideal para moverlo a cualquier sitio e incluso para enviarlo por email, subirlo a un servidor FTP o Samba y más. Gracias a que tenemos un simple archivo que contiene toda la información confidencial, podremos guardarlo a buen recaudo grabándolo en un CD o DVD, e incluso copiarlo en un pendrive.

→ Cifrado de dispositivos de almacenamiento extraíble como USB, tarjetas SD e incluso discos duros. En este caso, el dispositivo de almacenamiento extraíble estará completamente cifrado. Windows nos indicará que necesita formato del disco para poder leerlo, siempre debemos pinchar en cancelar y abrirlo con **VeraCrypt**, introduciendo la correspondiente clave de descifrado.

→ Cifrado de cualquier partición de estos dispositivos de almacenamiento extraíble.

→ Cifrado de la partición o disco completo donde Windows esté instalado. Esto nos permite hacer exactamente la misma función que **Bitlocker**, cifrará el disco duro o SSD por completo, para que tanto el sistema operativo como todos nuestros archivos estén a salvo frente a posibles robos.

→ El cifrado y descifrado si utilizamos **AES** se puede acelerar si el procesador del equipo soporta **AES-NI**, proporcionando una mayor velocidad de lectura y escritura.

→ Posibilidad de crear un volumen “oculto” para evitar que un posible atacante nos fuerce a revelar la contraseña del volumen (chantaje, extorsión, etc.)

The screenshot shows the official website for VeraCrypt (<https://www.veracrypt.fr/en/Home.html>). The page features the VeraCrypt logo, a navigation bar with links for Home, Source Code, Downloads, Documentation, Donate, and Forums. Below the navigation bar, a brief description states: "VeraCrypt is a free open source disk encryption software for Windows, Mac OSX and Linux. Brought to you by IDRIX (<https://www.idrix.fr>) and based on TrueCrypt 7.1a." A section titled "VeraCrypt main features:" lists several key capabilities, including creating virtual encrypted disks, encrypting partitions or storage devices, and providing plausibile deniability. Below this, there are links for "Donate to help the project" with icons for various payment methods like PayPal, Bitcoin, and Ethereum, and sections for "Release Notes / Changelog", "Frequently Asked Question", "Android & iOS Support", and "Contributed Resources & Downloads (Tutorials, PPA, ARM, Raspberry Pi...)". At the bottom, there are social media links for Twitter, Facebook, and Reddit, along with a "covertify passed" badge.

La descarga del programa la podremos hacer directamente desde la página oficial de **VeraCrypt**.

The screenshot shows the "Downloads" page of the VeraCrypt website (<https://www.veracrypt.fr/en/Downloads.html>). The top navigation bar has "Downloads" selected. The page includes a note to publishers about linking to the page instead of hosting files. It provides a PGP Public Key ([https://www.idrix.fr/VeraCrypt/VeraCrypt\\_PGP\\_public\\_key.asc](https://www.idrix.fr/VeraCrypt/VeraCrypt_PGP_public_key.asc)) and a Fingerprint (ID=0x680D16DE, Fingerprint=5069A233D55A0EEB174A5FC3821ACD02680D16DE). The "Latest Stable Release" section indicates releases for macOS 10.7 and later, and for other operating systems. Below this, there are download links for Windows, macOS, and Linux, each with specific file names and sizes. For example, the Windows section includes links for "VeraCrypt Setup 1.24-Update7.exe" (34.5 MB) and "VeraCrypt Portable 1.24-Update7.exe" (34.3 MB).

~ Let's Encrypt:



Let's Encrypt es una autoridad de certificación (AC, o CA por sus siglas en inglés *Certification Authority*) gratuita, automatizada y abierta que existe para el beneficio del público. Es un servicio provisto por el **Internet Security Research Group (ISRG)**. Distribuyen certificados digitales gratuitamente a personas que necesitan poder habilitar el uso del protocolo **HTTPS (SSL/TLS)** en sitios web asegurando la privacidad y la seguridad a los usuarios. Los principios claves detrás de **Let's Encrypt** son:

- Gratis: cualquier que posea un nombre de dominio puede usar **Let's Encrypt** para obtener un certificado de confianza sin costo alguno.
- Automático: un programa corriendo en un servidor de web puede interactuar con **Let's Encrypt** para obtener un certificado fácilmente, configurarlo de manera segura para uso, y automáticamente hacerse cargo de la renovación.
- Transparente: todos los certificados emitidos o revocados serán registrados públicamente y disponibles para que cualquiera los inspeccione.
- Abierto: la emisión automática y el protocolo de renovación serán publicados como un estándar abierto para que otros pueden adoptar.

Básicamente, su objetivo es hacer posible la configuración de un servidor HTTPS y hacer que obtenga automáticamente un certificado confiado por el navegador, sin ninguna intervención humana. Esto se logra ejecutando un agente de manejo de certificados en un servidor de web.

Hay dos pasos para este proceso. Primero, el agente le prueba al AC que el servidor de web controla el dominio. Luego, el agente puede pedir, renovar, y revocar certificados para ese dominio.

Para la validación del dominio, **Let's Encrypt** identifica el administrador del servidor por llave pública. La primera vez que el software del agente interactúa con **Let's Encrypt**, genera un nuevo par de llaves y demuestra al CA que el servidor controla uno o más dominios. Esto es similar al proceso tradicional de un AC de crear una cuenta y agregar dominios a esa cuenta.

## • 2.2. Encriptación de columnas de BD

La encriptación de columnas de nuestra base de datos se rige por una jerarquía de encriptación. Esta jerarquía utiliza una **DMK** (*Database Master Key*, Clave Maestra de Base de Datos), necesaria para realizar esta acción dentro de cada base de datos. Podemos tener claves maestras separadas cualquier base de datos del sistema. Dicho así, una *Master Key* es un objeto en SQL Server que funciona como base para la encriptación dentro de las bases de datos que permite asegurar las otras claves de encriptación.

Vamos a crear una base de datos contenida de práctica con usuario inventado, copiaremos la tabla **SAL\_FACTURA\_MATERIAL** escogiendo alguna columna como ejemplo para luego encriptarlas ejecutando las siguientes sentencias para la encriptación de columnas:

- Creamos la base de datos contenida **SCOUTS\_PRACTICE**:

```
USE master
GO

-- Controlamos la existencia de la BD contenida
DROP DATABASE IF EXISTS SCOUTS_PRACTICE
GO

CREATE DATABASE SCOUTS_PRACTICE
GO

ALTER DATABASE SCOUTS_PRACTICE
    SET CONTAINMENT = PARTIAL
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, a connection to 'SAL\_WS16\_SCOUTS (SQL Server 14.0.1000.1)' is selected, showing databases like System Databases, AdventureWorks2017, Northwind, pubs, SAL\_SCOUTS, SCOUTS\_PRACTICE, and WideWorldImporters. The central pane displays a query window titled 'encryption.sql - ...T\SAL\_SCOUTS (52)'. The script content is as follows:

```
1 USE master
2 GO
3
4 -- Controlamos la existencia de la BD contenida
5 DROP DATABASE IF EXISTS SCOUTS_PRACTICE
6 GO
7
8 CREATE DATABASE SCOUTS_PRACTICE
9 GO
10
11 ALTER DATABASE SCOUTS_PRACTICE
12     SET CONTAINMENT = PARTIAL
13 GO
14
15 USE SCOUTS_PRACTICE
16 GO
```

The status bar at the bottom indicates '127 %' and the message pane says 'Commands completed successfully.'

- Usamos la base de datos contenida **SCOUTS\_PRACTICE**, en ella creamos el *schema PRACTICE* con login del usuario de prueba llamado **PEPO** y la **Master Key** :

```

USE SCOUTS_PRACTICE
GO

DROP SCHEMA IF EXISTS PRACTICE
GO

CREATE SCHEMA PRACTICE
GO

CREATE LOGIN PEPO WITH PASSWORD = 'Abcd1234.'
GO

CREATE USER PEPO
GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Abcd1234.'
GO

```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which displays the database structure of the 'SAL\_WS16\_SCOUTS' database, including various system databases and user-defined databases like 'SCOUTS\_PRACTICE'. The main window contains a query editor with the script from above. The status bar at the bottom right shows a green checkmark and the message 'Query executed successfully.'

```

15 USE SCOUTS_PRACTICE
16 GO
17
18 DROP SCHEMA IF EXISTS PRACTICE
19 GO
20
21 CREATE SCHEMA PRACTICE
22 GO
23
24 CREATE LOGIN PEPO WITH PASSWORD = 'Abcd1234.';
25 go
26
27 CREATE USER PEPO
28 GO
29
30 CREATE MASTER KEY ENCRYPTION BY PASSWORD='Abcd1234.'
31 GO

```

- Creamos en la tabla **PRACTICE.SCOUTS\_ENCRYPT** junto con las filas de la tabla **SAL\_FACTURA\_MATERIAL**:

```

DROP TABLE IF EXISTS PRACTICE.SCOUTS_ENCRYPT
GO

SELECT fecha_compra, producto, cod_producto, cantidad, precio
    INTO SCOUTS_PRACTICE.PRACTICE.SCOUTS_ENCRYPT
    FROM SAL_SCOUTS.dbo.SAL_FACTURA_MATERIAL
GO
-- (10000 rows affected)

SELECT top 5 * FROM PRACTICE.SCOUTS_ENCRYPT
GO

-- fecha_compra      producto      cod_producto      cantidad      precio
-- 2020-04-05        Ziplock       83054           29          190,83
-- 2019-03-20        Ziplock       43417           12          177,08
-- 2020-06-20        Ice_chest     45414           42          70,30
-- 2021-01-13        Can_opener   64114           33          178,76
-- 2017-07-12        bed          40866           49          126,94

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "encryption.sql - SAL\_WS16\_SCOUTS.SCOUTS\_PRACTICE (SAL-SCOUTS\SAU\_SCOUTS (52)) - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, showing the database structure for "SAL\_WS16\_SCOUTS". The right pane contains the query results from the script. The script itself is as follows:

```

33  DROP TABLE IF EXISTS PRACTICE.SCOUTS_ENCRYPT
34  GO
35
36  SELECT fecha_compra, producto, cod_producto, cantidad, precio
37    INTO SCOUTS_PRACTICE.PRACTICE.SCOUTS_ENCRYPT
38    FROM SAL_SCOUTS.dbo.SAL_FACTURA_MATERIAL
39  GO
40  -- (10000 rows affected)
41
42  SELECT top 5 fecha_compra, producto, cod_producto, cantidad, precio FROM PRACTICE.SCOUTS_ENCRYPT
43  GO
44
45  -- fecha_compra producto      cod_producto      cantidad      precio
46  -- 2020-04-05  Ziplock       83054           29          190,83
47  -- 2019-03-20  Ziplock       43417           12          177,08
48  -- 2020-06-20  Ice_chest     45414           42          70,30
49  -- 2021-01-13  Can_opener   64114           33          178,76
50  -- 2017-07-12  bed          40866           49          126,94

```

The "Results" tab displays the output of the last two lines of the script, which is a table with five rows of data. The table has columns: fecha\_compra, producto, cod\_producto, cantidad, and precio. The data is as follows:

	fecha_compra	producto	cod_producto	cantidad	precio
1	2020-04-05	Ziplock	83054	29	190,83
2	2019-03-20	Ziplock	43417	12	177,08
3	2020-06-20	Ice_chest	45414	42	70,30
4	2021-01-13	Can_opener	64114	33	178,76
5	2017-07-12	bed	40866	49	126,94

At the bottom of the results pane, a message says "Query executed successfully."

- Concedemos permisos al usuario de pruebas **PEPO** y creamos el certificado **Scout\_CertPRACT** autorizando al mismo y generamos la clave simétrica llamada **SK\_SCOUT\_PRACTICE** encriptada por el certificado:

```

GRANT SELECT, INSERT, UPDATE, ALTER ON PRACTICE.SCOUTS_ENCRYPT TO PEPO
GO
CREATE CERTIFICATE Scout_CertPRACT AUTHORIZATION PEPO
    WITH SUBJECT = 'Certificate Practice', START_DATE='2021/05/05';
GO
SELECT name certName,
       certificate_id CertID,
       pvt_key_encryption_type_desc EncryptType,
       issuer_name Issuer
  from sys.certificates;
go
-- certName          CertID      EncryptType           Issuer
-- Scout_CertPRACT   256        ENCRYPTED_BY_MASTER_KEY Certificate Practice
CREATE SYMMETRIC KEY SK_SCOUT_PRACTICE
    WITH ALGORITHM = AES_256
    ENCRYPTION BY CERTIFICATE Scout_CertPRACT;
GO
SELECT name KeyName,
       symmetric_key_id KeyID,
       key_length KeyLength,
       algorithm_desc KeyAlgorithm
  FROM sys.symmetric_keys;
GO
-- KeyName          KeyID      KeyLength      KeyAlgorithm
-- ##MS_DatabaseMasterKey## 101        256          AES_256
-- SK_SCOUT_PRACTICE     256        256          AES_256
SELECT *
FROM sys.symmetric_keys
GO
-- name  principal_id  symmetric_id  key_length  key_algorithm  algorithm_desc
-- create_date modify_date key_guid
-- key_thumbprint provider_type  cryptographic_provider_guid
-- cryptographic_provider_algid
-- ##MS_DatabaseMasterKey## 1 101 256 A3 AES_256
-- 2021-05-05 11:53:20.653 2021-05-05 11:53:20.653 40F09500-548F-4E10-8E4D-
5F3565A651FA NULL NULL NULL NULL
-- SK_SCOUT_PRACTICE 1 256 256 A3 AES_256
-- 2021-05-05 11:53:26.950 2021-05-05 11:53:26.950 512AD600-6729-4BD6-89EB-
041592A2A561 NULL NULL NULL NULL

```

name	principal_id	symmetric_id	key_length	key_algorithm	algorithm_desc	create_date	modify_date
##MS_DatabaseMasterKey##	1	101	256	A3	AES_256	2021-05-05 11:53:20.653	2021-05-05 11:53:20.653
SK_SCOUT_PRACTICE	1	256	256	A3	AES_256	2021-05-05 11:53:26.950	2021-05-05 11:53:26.950

- Impersonamos como el usuario **PEPO** para proceder con la encriptación de columnas, para ello debemos abrir con nuestra clave simétrica, pero nos dará error indicando que dispone de permisos **VIEW** tanto en el certificado como en la clave simétrica:

```

EXECUTE AS USER = 'PEPO';
GO

PRINT USER
GO

OPEN SYMMETRIC KEY SK_SCOUT_PRACTICE
    DECRYPTION BY CERTIFICATE Scout_CertPRACT;
GO
-- Msg 15151, Level 16, State 1, Line 122
-- Cannot find the symmetric key 'SK_SCOUT_PRACTICE', because it does not exist or
you do not have permission.

REVERT
GO

PRINT USER
GO

GRANT VIEW DEFINITION ON CERTIFICATE::Scout_CertPRACT TO PEPO
go
--Cannot grant, deny, or revoke permissions to sa, dbo, entity owner,
information_schema, sys, or yourself.
GRANT VIEW DEFINITION ON SYMMETRIC KEY::SK_SCOUT_PRACTICE TO PEPO
go
-- Commands completed successfully

EXECUTE AS USER = 'PEPO';
GO

PRINT USER
GO

ALTER TABLE PRACTICE.SCOUTS_ENCRYPT
    add prod_encr varbinary(max);
GO

OPEN SYMMETRIC KEY SK_SCOUT_PRACTICE
    DECRYPTION BY CERTIFICATE Scout_CertPRACT;
GO
-- Commands completed successfully

UPDATE PRACTICE.SCOUTS_ENCRYPT
    SET prod_encr = EncryptByKey(Key_GUID('SK_SCOUT_PRACTICE'), producto);
GO
-- (10000 rows affected)

CLOSE SYMMETRIC KEY SK_SCOUT_PRACTICE;
GO

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "encryption.sql - SAL\_WS16\_SCOUTS.SCOUTS\_PRACTICE (SAL-SCOUTS\Sal\_SCOUTS (52)) - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, showing the database structure. The right pane is the Results window for the "encryption.sql" script. The script content is as follows:

```

128
129   ALTER TABLE PRACTICE.SCOUTS_ENCRYPT
130     add prod_encri varbinary(max);
131   GO
132
133   OPEN SYMMETRIC KEY SK_SCOUT_PRACTICE
134     DECRYPTION BY CERTIFICATE Scout_CertPRACT;
135   GO
136   -- Commands completed successfully
137
138   UPDATE PRACTICE.SCOUTS_ENCRYPT
139     SET prod_encri = EncryptByKey(Key_GUID('SK_SCOUT_PRACTICE'),producto);
140   GO
141
142   -- (10000 rows affected)
143
144   CLOSE SYMMETRIC KEY SK_SCOUT_PRACTICE;
145   GO
146

```

The "Messages" tab in the Results window shows "Commands completed successfully.". The status bar at the bottom right says "Query executed successfully."

- Comprobamos que se han encriptado los datos de la columna:

```

select top 5 * from PRACTICE.SCOUTS_ENCRYPT
go

-- fecha_compra    producto    cod_producto    cantidad    precio    prod_encri
-- 2020-04-05 Ziplock    83054            29        190,83
0x00E69B66F54C624380280ABD1A506F4102000000132909999DBF894A01D8B2F050ADD1D30EDC
684F2091F2FCE4EFE4846725F9A0716531A22C6CD53ED202359074CED44C
-- 2019-03-20 Ziplock    43417            12        177,08
0x00E69B66F54C624380280ABD1A506F410200000028C588E8DD4BE9C2826FAADDFF3B50B185D5
E5EFD7254638695D233870891184547422FBAD461AE7F2EA8EDD8A68BD5A
-- 2020-06-20 Ice_chest   45414            42        70,30
0x00E69B66F54C624380280ABD1A506F41020000005D8FFF91D444D82FB93E36A7EB9D11C7D4E0
B1C5AB833769AB7F224E7A08E87CED03A91F19E59FF8A0C062FDC7CC5D56
-- 2021-01-13 Can_opener 64114            33        178,76
0x00E69B66F54C624380280ABD1A506F41020000006B28616A19E388E661D88223C209FF7261E5
25D47AA5922FA46A9D2FC9E5C3F3CEA665C325058D484643B49CC0AF55BE
-- 2017-07-12 bed        40866            49        126,94
0x00E69B66F54C624380280ABD1A506F41020000001E5C698646FAB0FAEB2CC0B19A81BE47DA19
A493BEE37EB42A3F5B0B7738765E

```

```

146
147 select top 5 * from PRACTICE.Scouts_Encrypt
148 go
149
150 --- fecha_compra producto cod_producto cantidad precio prod_encri
151 -- 2020-04-05 Ziplock 83054 29 190,83 0x00E69B66F54C624380280ABD1A506F410200000013290999DBF894A01D082F050ADD1D30EDC684F2
152 -- 2019-03-20 Ziplock 43417 12 177,08 0x00E69B66F54C624380280ABD1A506F410200000028C588EB0D4BE9C2826FAADDFF3B50B185D5E5EF0
153 -- 2020-06-20 Ice_chest 45414 42 70,30 0x00E69B66F54C624380280ABD1A506F41020000005D8FF91D44D82FB93E36A7EB9011C7D4E0B1C5AE
154 -- 2021-01-13 Can_opener 64114 33 178,76 0x00E69B66F54C624380280ABD1A506F41020000006B28616A19E388E661D88223C209FF7261E525D47A
155 -- 2017-07-12 bed 40866 49 126,94 0x00E69B66F54C624380280ABD1A506F41020000001E5C698646FAB0FAEB2CC0B19A81BE47DA19A493BE

```

Query executed successfully.

- Si le retiramos (**REVOKE**) los permisos de **VIEW** sobre la clave simétrica al usuario, no podrá visualizar la columna y le saldrá todo en **NULL**:

```
REVERT
GO
```

```
REVOKE VIEW DEFINITION ON CERTIFICATE::Scout_CertPRACT to PEPO
go
--Cannot grant, deny, or revoke permissions to sa, dbo, entity owner,
information_schema, sys, or yourself.
REVOKE VIEW DEFINITION ON SYMMETRIC KEY::SK_SCOUT_PRACTICE to PEPO
go
```

```
EXECUTE AS USER = 'PEPO'
GO
```

```
PRINT USER
GO
```

```
SELECT      top      5      producto, CONVERT(VARCHAR, DECRYPTBYKEY(prod_encri))      as
PRODUCTO_ENcriptado  from PRACTICE.Scouts_Encrypt
GO
```

```

157 REVERT
158 GO
159
160 REVOKE VIEW DEFINITION ON CERTIFICATE::Scout_CertPRACT to PEPO
161 go
162 --Cannot grant, deny, or revoke permissions to sa, dbo, entity owner, information_schema, sys, or yourself.
163 REVOKE VIEW DEFINITION ON SYMMETRIC KEY::SK_SCOUT_PRACTICE to PEPO
164 go
165
166 EXECUTE AS USER = 'PEPO'
167 GO
168
169 PRINT USER
170 GO
171
172 SELECT top 5 producto,CONVERT(VARCHAR,DECRYPTBYKEY(prod_encri)) as PRODUCTO_ENcriptado from PRACTICE.Scouts_Encrypt
173 GO

```

Query executed successfully.

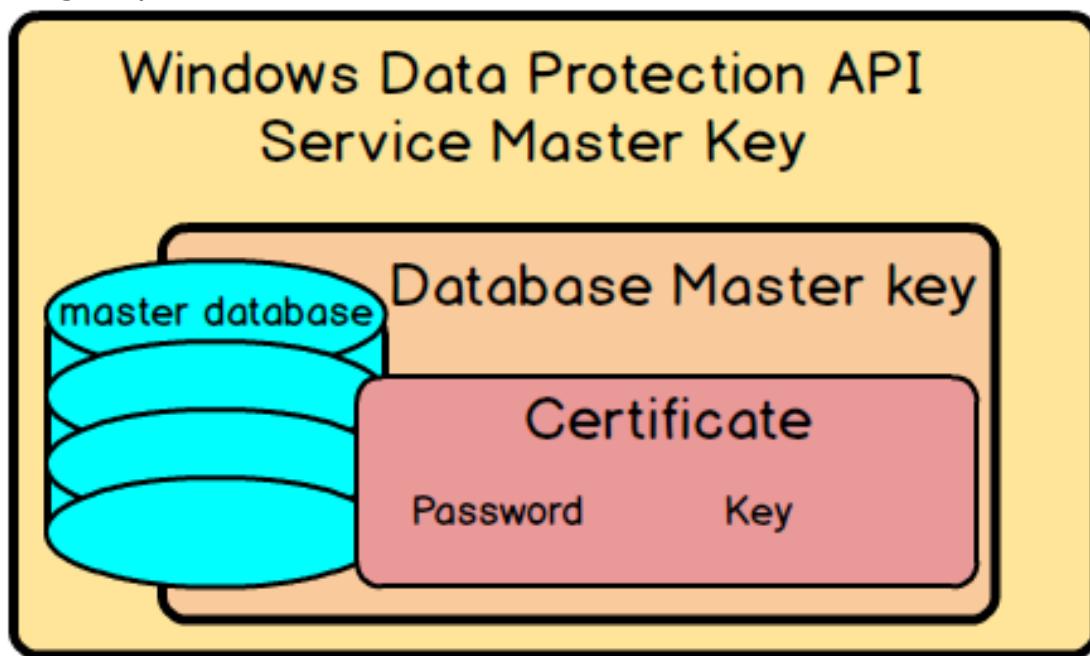
## • 2.3. Encriptación de backups de BD

Como se mencionó anteriormente, SQL Server tiene una infraestructura de encriptación en relación con las **DMKs** (*Database Master Key*). Esta infraestructura jerárquica de encriptación hace que cada capa en la jerarquía encripta la de abajo.

La primera capa de jerarquía es la **SMK** (*Service Master Key*). La Clave Maestra de Servicio es generada automáticamente durante la instalación de SQL Server y almacenada en la base de datos maestra del sistema. La **SMK** es la única para cada instancia SQL Server. La Clave Maestra de Servicios es encriptada basándose en las credenciales para la cuenta de servicio SQL Server y la clave *Windows Data Protection API (DPAPI)*.

La siguiente capa es nuestra Clave Maestra de Base de datos. Es única para cada base de datos maestra del sistema para cada instancia SQL Server.

El siguiente nivel en la jerarquía es un certificado que puede contener una clave privada que es protegido por la Clave Maestra de Bases de Datos, o una clave asimétrica.



La característica de encriptación de copias de seguridad de SQL Server provee encriptación de datos con los algoritmos **AES\_128**, **AES\_192**, AES\_256 y **Triple DES (3DES)** (estos algoritmos marcados en negrita, ya están en desuso).

Antes de realizar la encriptación de los backups de nuestra base de datos debemos verificar si disponemos de una Clave Maestra de Servicio y una Clave Maestra de Base de Datos en la base de datos *master*. Como la SMK ya se había creado durante la instalación de SQL Server, debería ya estar contenida en *master*. Para comprobarlo, ejecutamos la sentencia:

```

SELECT * FROM master.sys.symmetric_keys
GO

-- name           principal_id      symmetric_key_id key_length key_algorithm
-- algorithm_desc  create_date          modify_date
key_guid          key_thumbprint provider_type
cryptographic_provider_guid
-- ##MS_ServiceMasterKey##    1           102          256
A3                AES_256        2021-02-06 13:28:01.903   2021-02-06 13:28:01.903
E9A9DA63-5A9B-4FFF-AC5F-2ABAFAABACA5 NULL          NULL          NULL
NULL

```

	name	principal_id	symmetric_key_id	key_length	key_algorithm	algorithm_desc	create_date	modify_date	key_guid	key_thumbprint
1	##MS_ServiceMasterKey##	1	102	256	A3	AES_256	2021-02-06 13:28:01.903	2021-02-06 13:28:01.903	E9A9DA63-5A9B-4FFF-AC5F-2ABAFAABACA5	NULL

Query executed successfully.

Si la fila `##MS_ServiceMasterKey##` no existe, ejecutamos la sentencia:

```

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Abcd1234.'
GO

```

```

-- name           principal_id      symmetric_key_id key_length key_algorithm
-- key_algorithm  algorithm_desc    create_date          modify_date
key_guid          key_thumbprint provider_type
cryptographic_provider_guid
-- ##MS_DatabaseMasterKey##    1           101          256
A3                AES_256        2021-05-05 23:21:21.707   2021-
05-05 23:21:21.707 67DB5800-5090-43BA-916E-914BD70ACD1F NULL          NULL
NULL
-- ##MS_ServiceMasterKey##     1           102          256
A3                AES_256        2021-02-06 13:28:01.903   2021-
02-06 13:28:01.903 E9A9DA63-5A9B-4FFF-AC5F-2ABAFAABACA5 NULL          NULL
NULL

```

```

187
188  CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Abcd1234.'
189  GO
190
191  -- name          principal_id  symmetric_key_id  key_length  key_algorithm  algorithm_desc  create_date
192  --- ##MS_DatabaseMasterKey## 1           101            256          A3             AES_256        2021-05-05 2
193  --- ##MS_ServiceMasterKey## 1           102            256          A3             AES_256        2021-02-06 1

```

Results pane output:

name	principal_id	symmetric_key_id	key_length	key_algorithm	algorithm_desc	create_date
##MS_DatabaseMasterKey##	1	101	256	A3	AES_256	2021-05-05 23:21:21.707
##MS_ServiceMasterKey##	1	102	256	A3	AES_256	2021-02-06 13:28:01.903

Query executed successfully.

Después creamos el certificado ejecutando la sentencia:

```

CREATE CERTIFICATE SAL_SCOUTSCert
    WITH SUBJECT = 'CERT SAL';
GO

```

Es conveniente hacer un backup al propio certificado, incluso a la **Master Key**. Ejecutamos las sentencias:

```

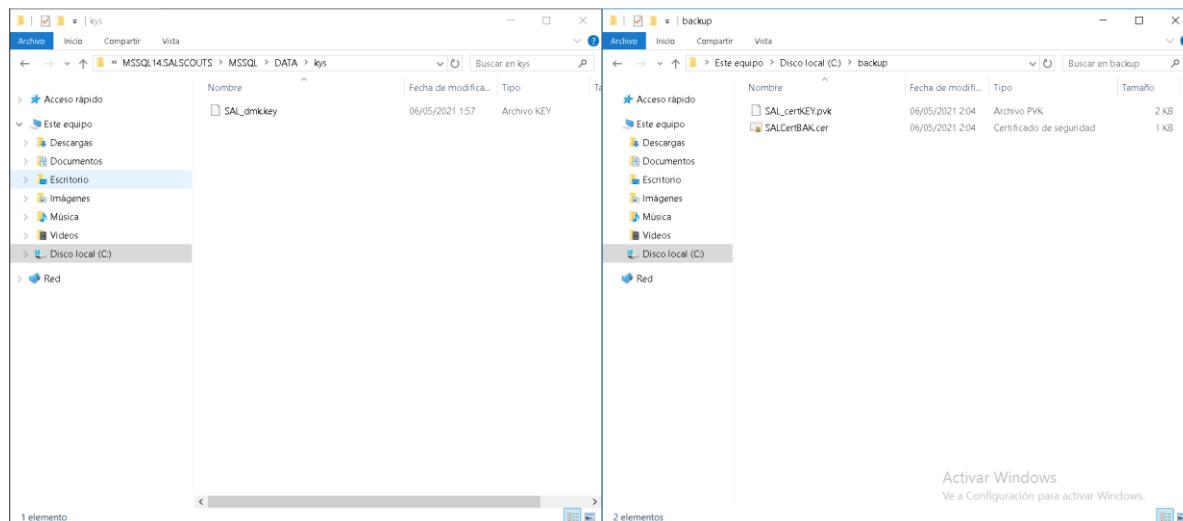
BACKUP      MASTER      KEY      TO      FILE      =      'C:\Program      Files\Microsoft      SQL
Server\MSSQL14.SALSCOUTS\MSSQL\DATA\kys\SAL_dmk.key'
    ENCRYPTION BY PASSWORD = 'Abcd1234.';
GO

```

```

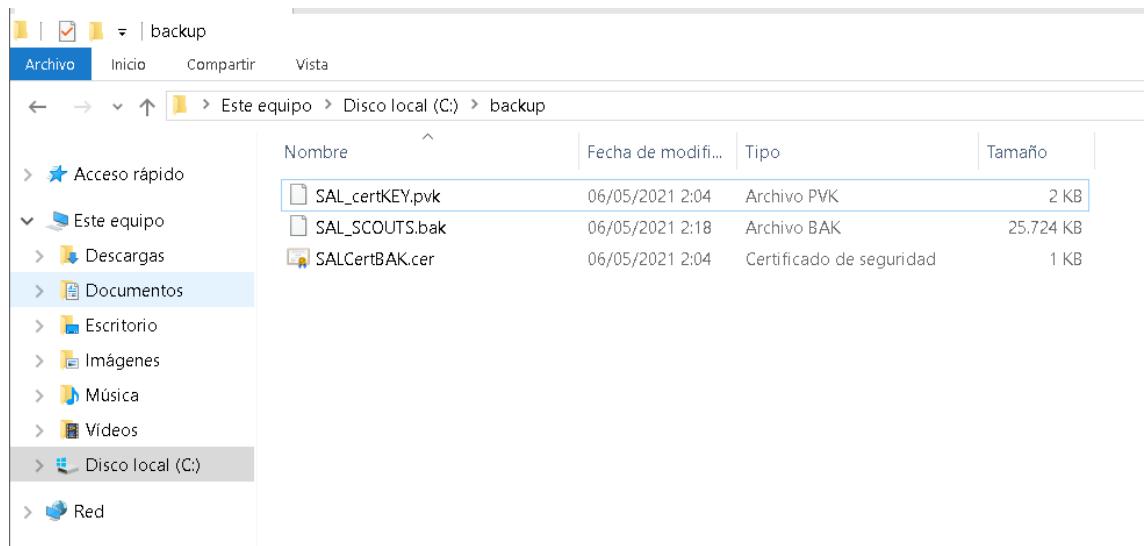
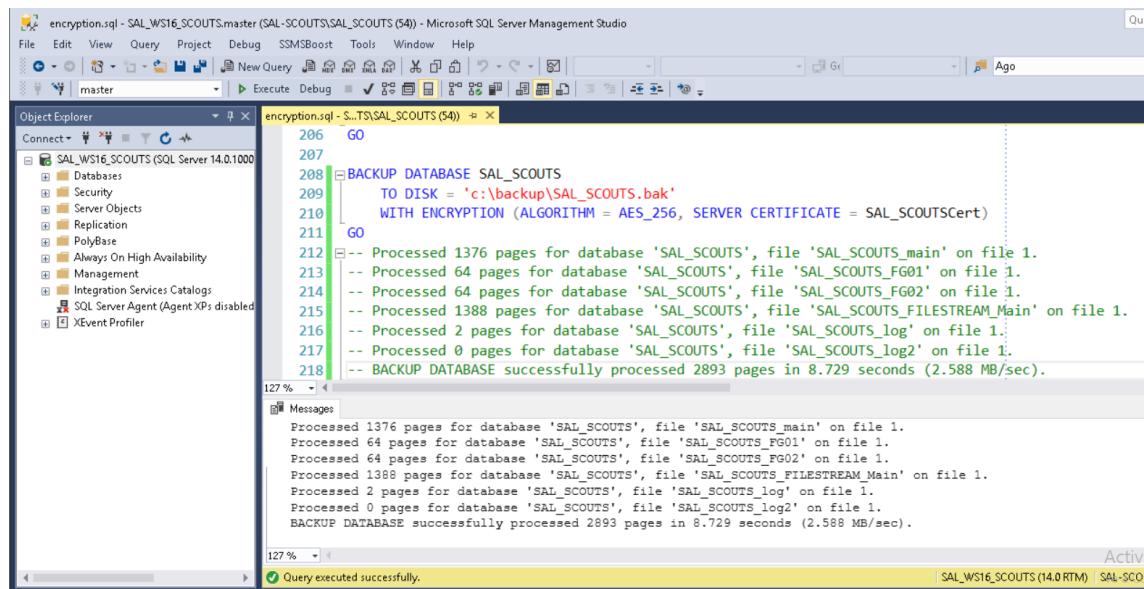
BACKUP CERTIFICATE SAL_SCOUTSCert
    TO FILE = 'C:\backup\SALCertBAK.cer'
    WITH PRIVATE KEY (
        FILE = 'C:\backup\SAL_certKEY.pvk',
        ENCRYPTION BY PASSWORD = 'Abcd1234.')
GO

```



Con todo respaldado ya podemos realizar una copia de seguridad a nuestra base de datos, ejecutamos la sentencia:

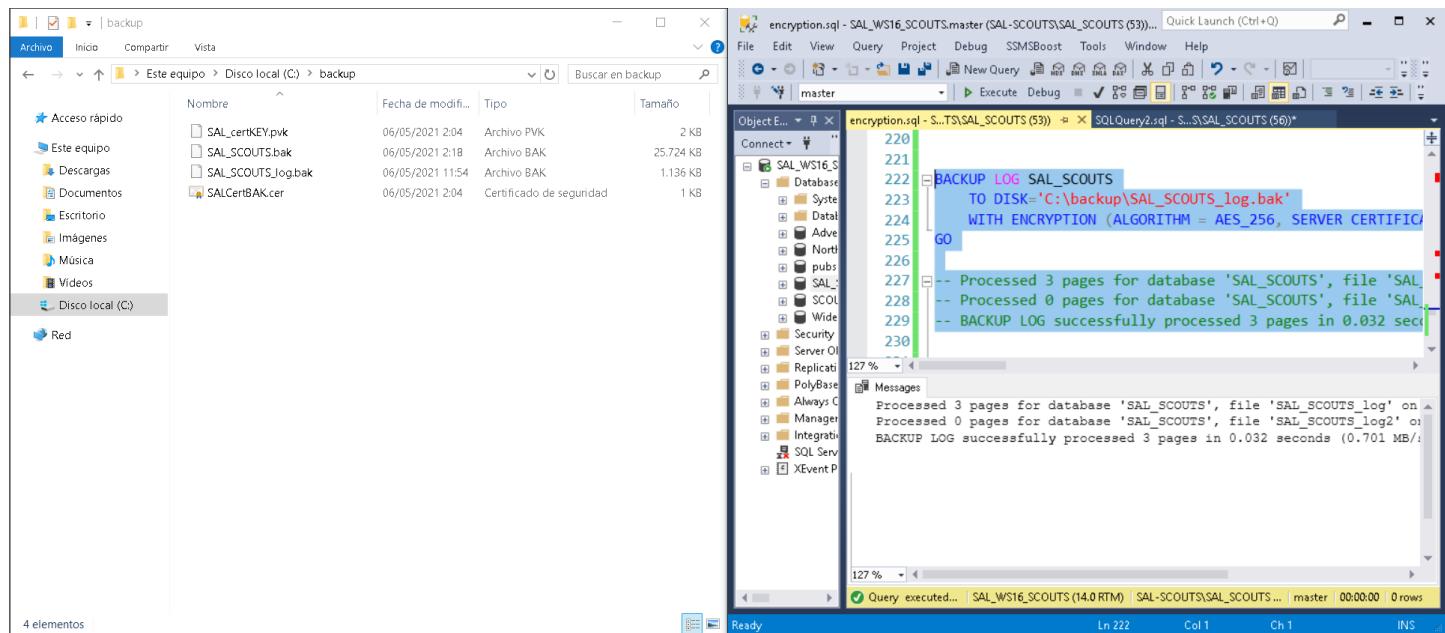
```
BACKUP DATABASE SAL_SCOUTS
    TO DISK = 'c:\backup\SAL_SCOUTS.bak'
    WITH ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE = SAL_SCOUTSCert)
GO
-- Processed 1376 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_main' on file
1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG01' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG02' on file 1.
--     Processed      1388      pages      for      database      'SAL_SCOUTS',      file
'SAL_SCOUTS_FILESTREAM_Main' on file 1.
-- Processed 2 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- BACKUP DATABASE successfully processed 2893 pages in 8.729 seconds (2.588
MB/sec).
```



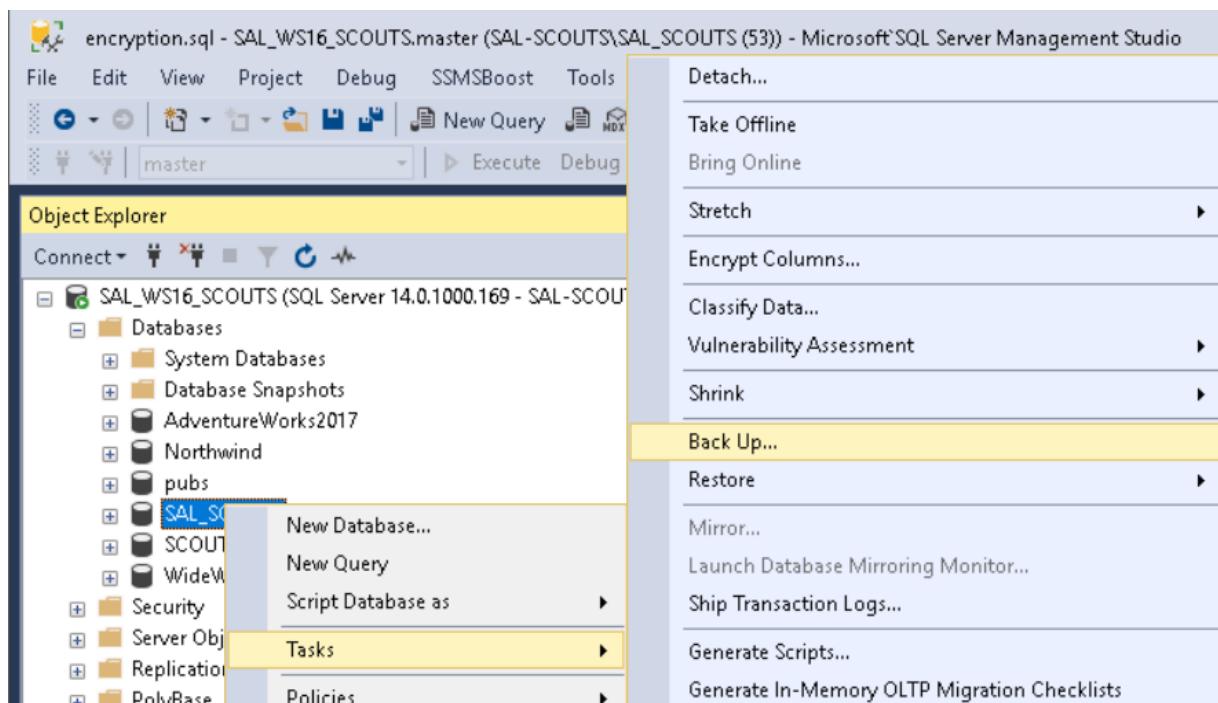
Y sin olvidarnos también de la copia de seguridad del *log*:

```
BACKUP LOG SAL_SCOUTS
    TO DISK='C:\backup\SAL_SCOUTS_log.bak'
    WITH ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE = SAL_SCOUTSCert);
GO

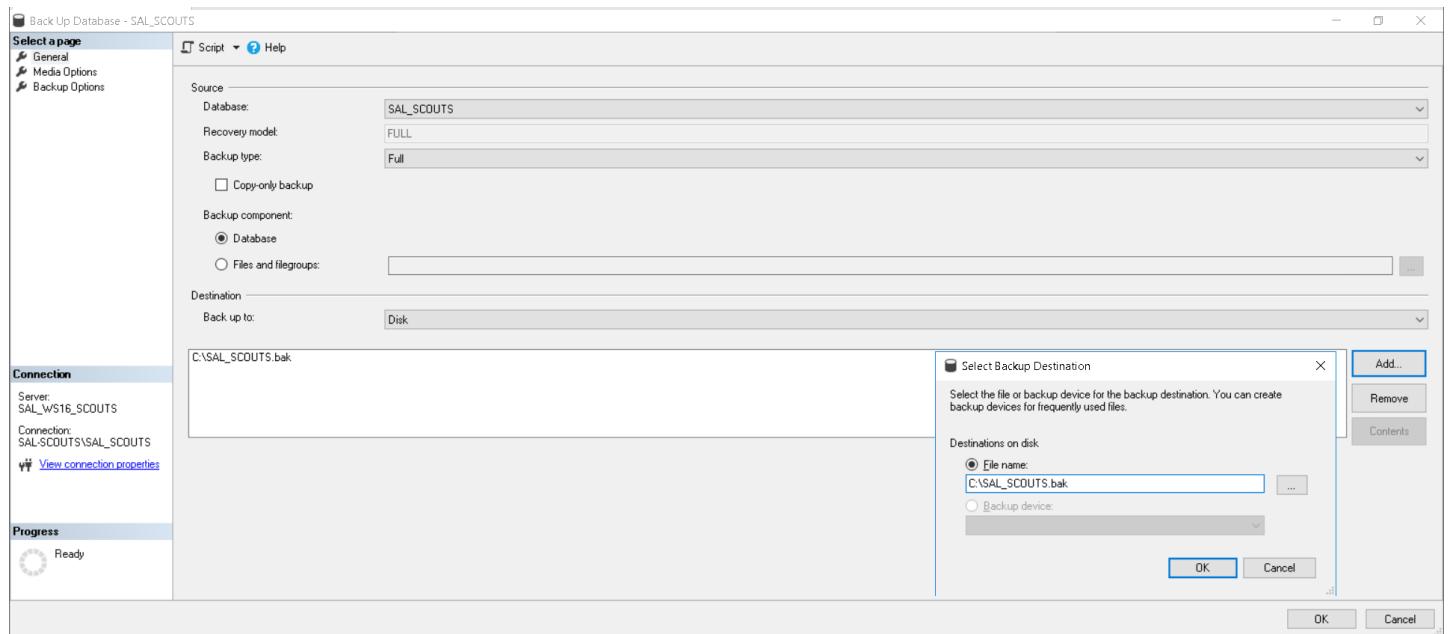
-- Processed 3 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- BACKUP LOG successfully processed 3 pages in 0.032 seconds (0.701 MB/sec).
```



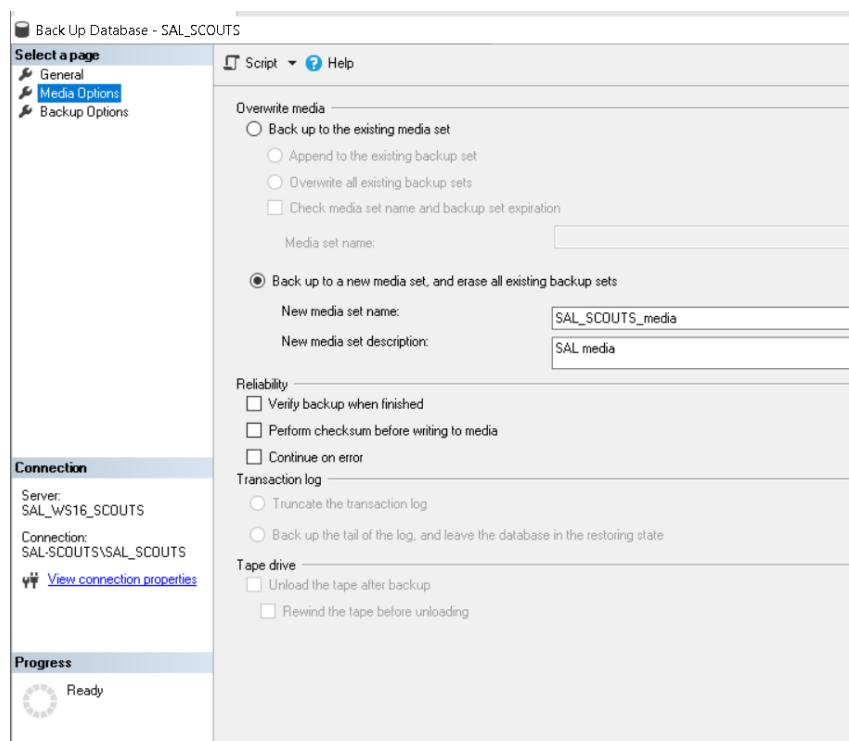
En el caso de no querer hacerlo mediante T-SQL, se puede hacer esto mismo desde GUI a través del *Object Explorer* haciendo click derecho > *Tasks* > *Back Up...*:



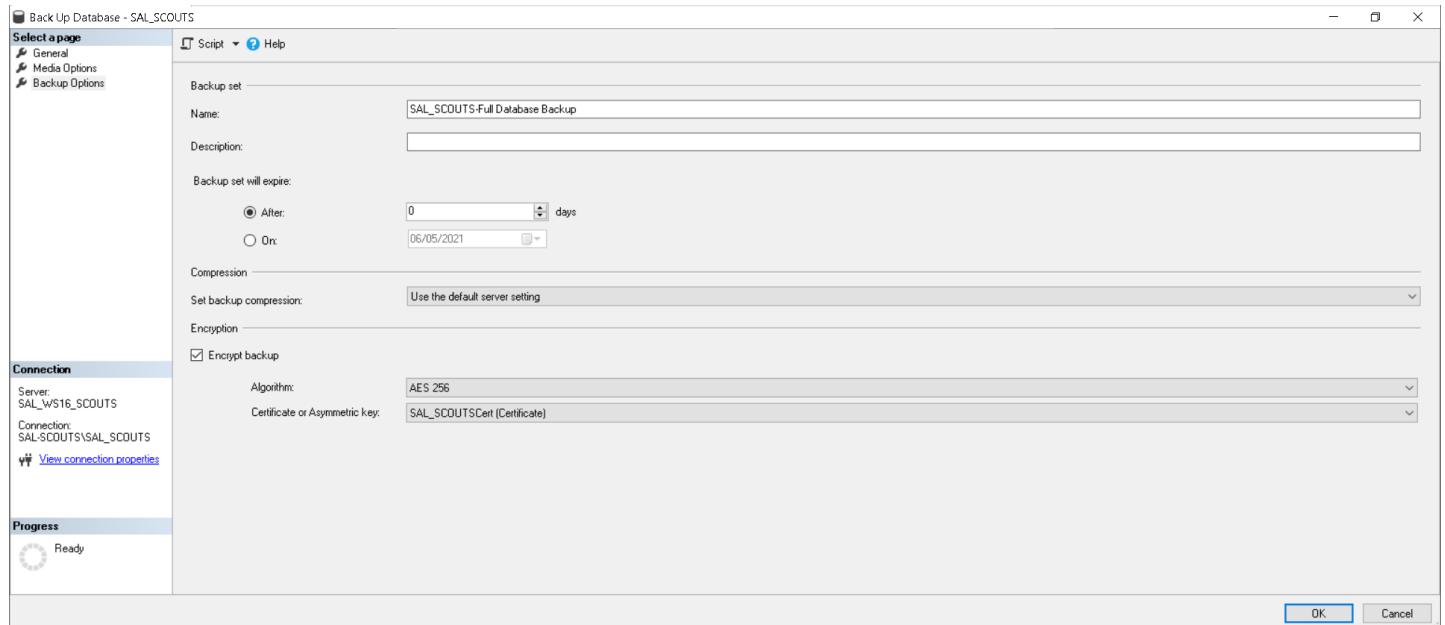
Después nos aparece el asistente de copias de seguridad, en vez de hacer una copia de seguridad normal, nuestro caso es hacerlo con nuestro certificado. Para ello realizamos las mismas acciones, seleccionamos destino en dónde almacenaremos el archivo **.bak**:



Aquí nos damos de cuenta que, al parecer, una de las restricciones de las copias de seguridad cifradas es que no pueden ser anexadas a un conjunto de copias de seguridad existente, por lo tanto, **SSMS** requiere configurar la base de datos para respaldarse a un nuevo conjunto de medios. Debajo de la pestaña **Media Options**, seleccionamos la opción **"Back up to a new media set, and erase all existing backup sets"** e ingresamos el nombre de un conjunto de medios y su descripción:



Luego debajo de la pestaña **Backup Options**, seleccionamos la opción **Encrypt Backup**, seleccionamos el algoritmo de encriptación y nuestro certificado:

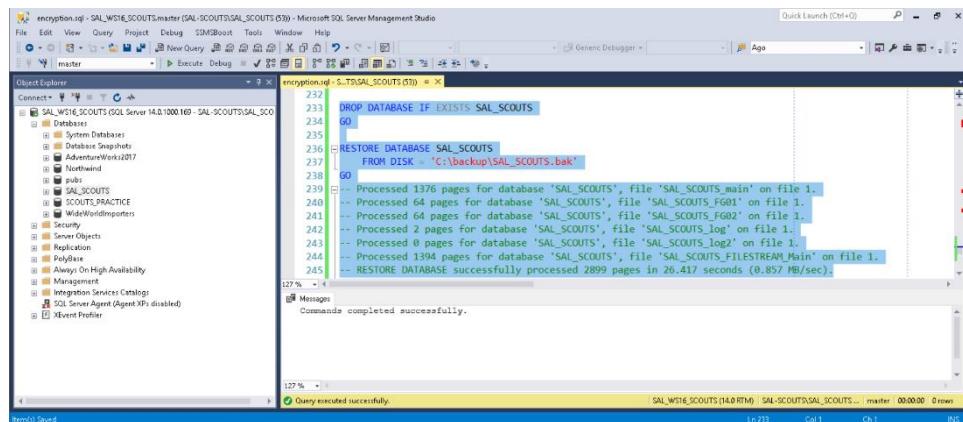


Y, por último, para realizar la restauración de nuestra base de datos, ejecutamos la sentencia:

```
DROP DATABASE IF EXISTS SAL_SCOUTS
GO
```

```
RESTORE DATABASE SAL_SCOUTS
    FROM DISK = 'C:\backup\ SAL_SCOUTS.bak'
GO

-- Processed 1376 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_main' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG01' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG02' on file 1.
-- Processed 2 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- Processed 1394 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FILESTREAM_Main' on file 1.
-- RESTORE DATABASE successfully processed 2899 pages in 26.417 seconds (0.857 MB/sec).
```



## • 2.4. Encrip. BD TDE (Transparent Data Encryption)

El **Cifrado/Encriptado de Datos Transparente (TDE, Transparent Data Encryption)** cifra/encrypta los archivos de datos de **SQL Server** y otras instancias, lo que se conoce como cifrado de datos en reposo.

En el caso de sufrir un robo de los medios físicos como las unidades de disco o similares o copias de seguridad se puede restaurar la base de datos o conectarse a ella y examinar sus datos.

Una solución sería cifrar los datos confidenciales en la base de datos y usar un certificado para proteger las claves con las que esos datos se cifran. Esta solución impide que alguien que carezca de las claves use los datos. Este tipo de protección se debe planear de antemano.

TDE realiza el cifrado y descifrado de I/O (*INPUT/OUTPUT*, entrada/salida) en tiempo real de los archivos de datos y de registro. Este cifrado usa una clave de cifrado de datos (DEK, *Data Encryption Key*). El registro de arranque de la base de datos almacena la clave para que esté disponible durante la recuperación. La DEK es una clave simétrica. Está protegida por un certificado que la base de datos maestra del servidor almacena, o por una clave asimétrica que un módulo EKM (*Extensible Key Management*) protege.

TDE protege los datos en reposo que son los archivos de datos y de registro. Permite cumplir muchas leyes, normativas y directrices establecidas en diversos sectores. Esto permite a los desarrolladores de software cifrar datos con algoritmos de cifrado **AES** y **3DES** (en desuso) sin cambiar las aplicaciones existentes.

Las ventajas de utilizar TDE son:

- Como administrador de seguridad tendremos la tranquilidad de que los datos confidenciales estén protegidos en caso de sustracción de los medios de almacenamiento o de los archivos de datos.
- La implementación de TDE ayuda a abordar los aspectos de cumplimiento reglamentario relacionados con la seguridad.
- No es necesario crear *triggers* ni *views* para descifrar los datos para una aplicación o usuario autorizados. Los datos de las tablas se descifran de forma transparente para la aplicación y el usuario de la base de datos.
- No hace falta modificar las aplicaciones para controlar los datos cifrados. La base de datos administra el cifrado y descifrado de datos.
- Las operaciones de administración de claves están automatizadas. El usuario o la aplicación no necesitan administrar claves de cifrado.

El procedimiento de encriptación es parecido al del punto anterior, solo que crearemos un certificado diferente y emplearemos la misma clave maestra. La diferencia que puede haber es que el archivo `.bak` va a depender del certificado (cosa que se ha olvidado de comprobar anteriormente). Creamos el nuevo certificado con la sentencia:

```
CREATE CERTIFICATE SAL_TDEscout
    WITH SUBJECT = 'SAL TDE SCOUT';
GO
-- Commands completed successfully
```

Comprobamos los certificados con la tabla del sistema `sys.certificates`:

```
SELECT TOP 1 *
FROM sys.certificates
ORDER BY name DESC
GO

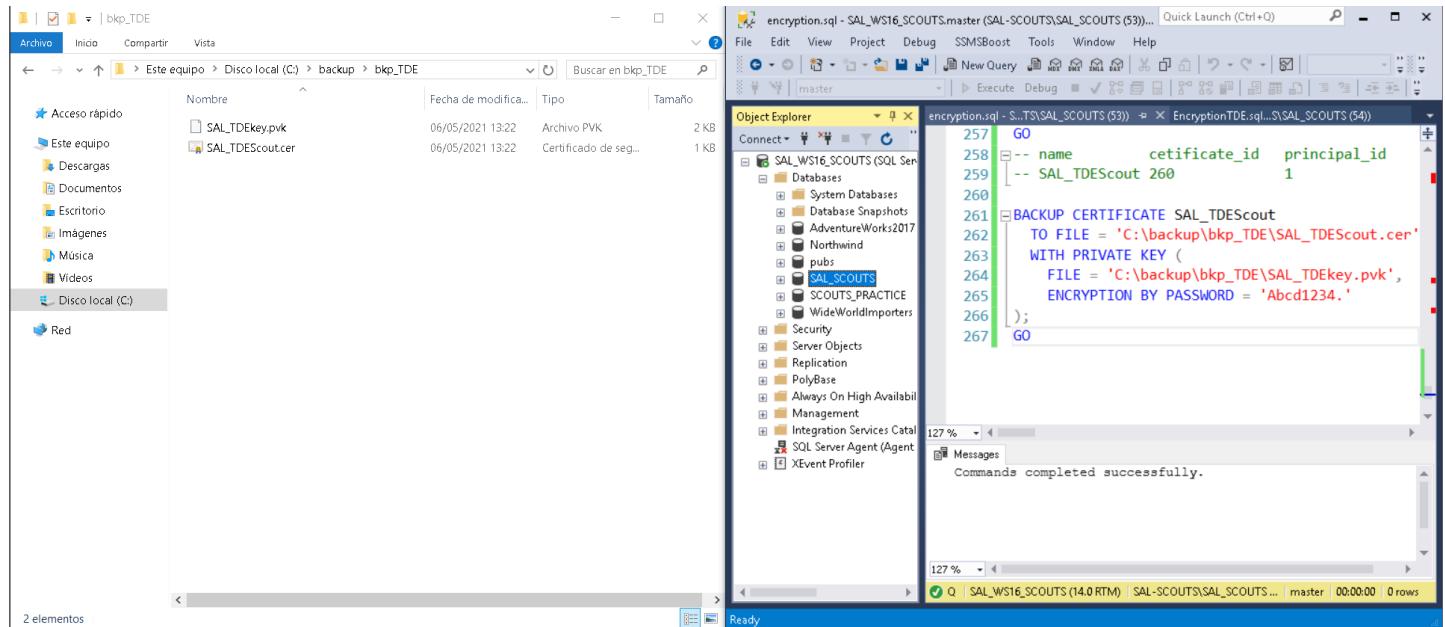
--name      certificate_id   principal_id   pvt_key_encryption_type
           pvt_key_encryption_type_desc  is_active_for_begin_dialog issuer_name
           cert_serial_number   sid   string_sid subject   expiry_date   start_date
           thumbprint attested_by   pvt_key_last_backup_date   key_length
--TDECert   277   1   MK   ENCRYPTED_BY_MASTER_KEY   1   TDE Cert for Test
   37   c5   aa   01   a2   ce   59   9d   44   7e   34   0e   1f   09   34   77
   0x010600000000000901000000BF05FDBA4584C56ACAE9AE38E0FF4EED74E7F83   S-1-9-1-
3137144255-1791329349-3818581194-4008972174-2206158551   TDE   Cert   for   Test
   2022-04-21 14:59:51.000   2021-04-21   14:59:51.000
   0xBF05FDBA4584C56ACAE9AE38E0FF4EED74E7F83   NULL   NULL   2048
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database `SAL_WS16_SCOUTS` is selected. In the center pane, a query window displays the creation of a certificate and its selection. Below the query results, a results grid shows the details of the created certificate. The status bar at the bottom indicates the query was executed successfully.

name	certificate_id	principal_id	pvt_key_encryption_type	pvt_key_encryption_type_desc	is_active_for_begin_dialog	issuer_name	cert_serial_number
SAL_TDEscout	260	1	MK	ENCRYPTED_BY_MASTER_KEY	1	SAL TDE SCOUT	7977ab28130b159e4f4f00c4c3a1b0

Ahora hacemos un *backup* al certificado y a la clave privada:

```
BACKUP CERTIFICATE SAL_TDEScout  
TO FILE = 'C:\backup\Sal_TDEScout.cer'  
WITH PRIVATE KEY (  
FILE = 'C:\backup\Sal_TDEkey.pvk',  
ENCRYPTION BY PASSWORD = 'Abcd1234.'  
);  
GO
```



Ahora utilizamos nuestra base de datos y creamos la clave de encriptación de la base de datos, ejecutamos la sentencia:

```
USE SAL_SCOUTS  
GO  
  
CREATE DATABASE ENCRYPTION KEY  
WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE SAL_TDEScout;  
GO
```

A partir de este momento ya tenemos todo preparado para poder encriptar la base de datos. Solamente debemos tener en cuenta de que si hay alguna conexión existente no se realizará la encriptación, así que volveremos a *master* y ejecutamos la sentencia:

```
USE master;  
GO  
  
ALTER DATABASE SAL_SCOUTS SET ENCRYPTION ON;  
GO
```

Y aquí es cuando comienza el proceso de encriptación de la base de datos. Para revisar el estado de encriptación se ejecuta la sentencia:

```
SELECT * FROM sys.dm_database_encryption_keys;
SELECT DB_Name(9);
SELECT DB_Name(database_id) AS 'Database', encryption_state
FROM sys.dm_database_encryption_keys;
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_WS16\_SCOUTS' is selected. In the center pane, a query window displays three SELECT statements. The first statement retrieves all columns from the sys.dm\_database\_encryption\_keys system view. The second statement uses DB\_Name(9) to identify the database with database\_id 9. The third statement joins sys.dm\_database\_encryption\_keys with sys.database to get the database name and its encryption state. The results pane shows two rows for the 'SAL\_SCOUTS' database, both with an encryption\_state of 3. The bottom status bar indicates the query was executed successfully.

database_id	encryption_state	create_date	regenerate_date	modify_date	set_date	opened_date	key_algorithm	key_length	encryptor_thumbprint
1	2	2021-05-06 11:40:49.560	2021-05-06 11:40:49.560	1900-01-01 00:00:00.000	2021-05-06 11:40:49.560	AE5	256	0x	
2	9	2021-05-06 11:32:12.567	2021-05-06 11:32:12.567	2021-05-06 11:32:12.567	2021-05-06 11:40:49.540	AE5	256	0x013BB7301FDF9ED16F0377F17D	

Database	encryption_state
tempdb	3
SAL_SCOUTS	3

Observamos en esta triple consulta que hay 2 bases de datos que tienen en el campo **encryption\_state = 3**. Con la consulta `SELECT DB_Name(9);` identificamos que la base de datos con `database_id = 9` es la nuestra, **SAL\_SCOUTS**. El valor numérico del campo **encryption\_state** puede variar dependiendo del valor que indique:

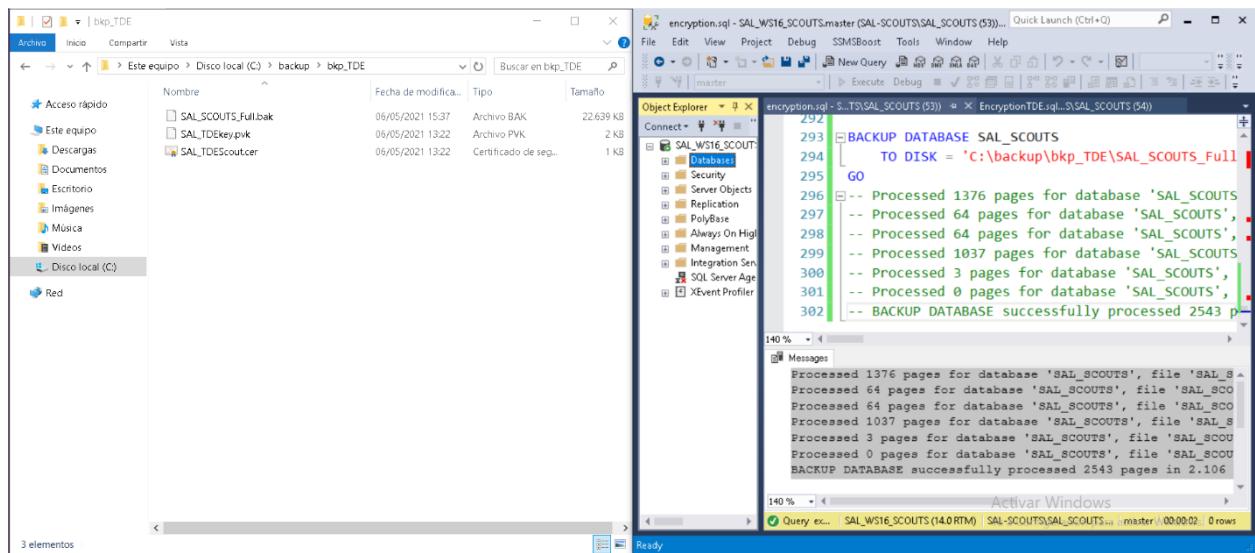
- 0 = No existe una clave de cifrado en la base de datos, no está cifrado.
- 1 = No cifrado.
- 2 = Se está cifrando la base de datos.
- 3 = Cifrado.
- 4 = Se está cambiando la clave de cifrado.
- 5 = Se está descifrando.
- 6 = Se está cambiando la protección. El certificado o la clave asimétrica que ha cifrado la clave de cifrado de base de dato se está modificando.

Ahora vamos a hacer una copia de seguridad a la base de datos, ejecutamos la sentencia:

```

BACKUP DATABASE SAL_SCOUTS
    TO DISK = 'C:\backup\bkp_TDE\SAL_SCOUTS_Full.bak';
GO
-- Processed 1376 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_main' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG01' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG02' on file 1.
-- Processed 1037 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FILESTREAM_Main' on file 1.
-- Processed 3 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- BACKUP DATABASE successfully processed 2543 pages in 2.106 seconds (9.430 MB/sec).

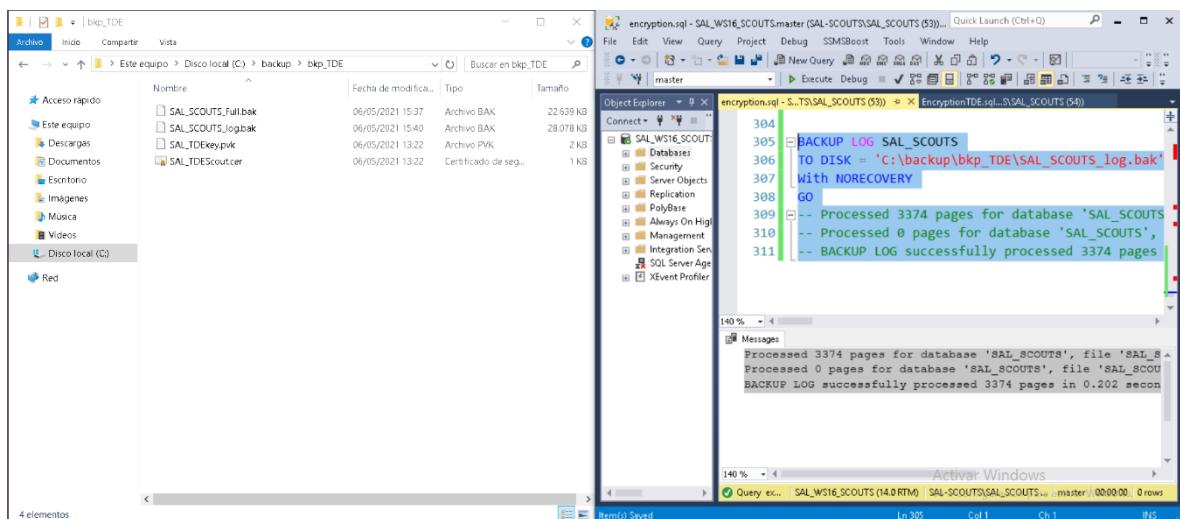
```



```

BACKUP LOG SAL_SCOUTS
TO DISK = 'C:\backup\bkp_TDE\SAL_SCOUTS_log.bak'
With NORECOVERY
GO
-- Processed 3374 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- BACKUP LOG successfully processed 3374 pages in 0.202 seconds (130.455 MB/sec).

```

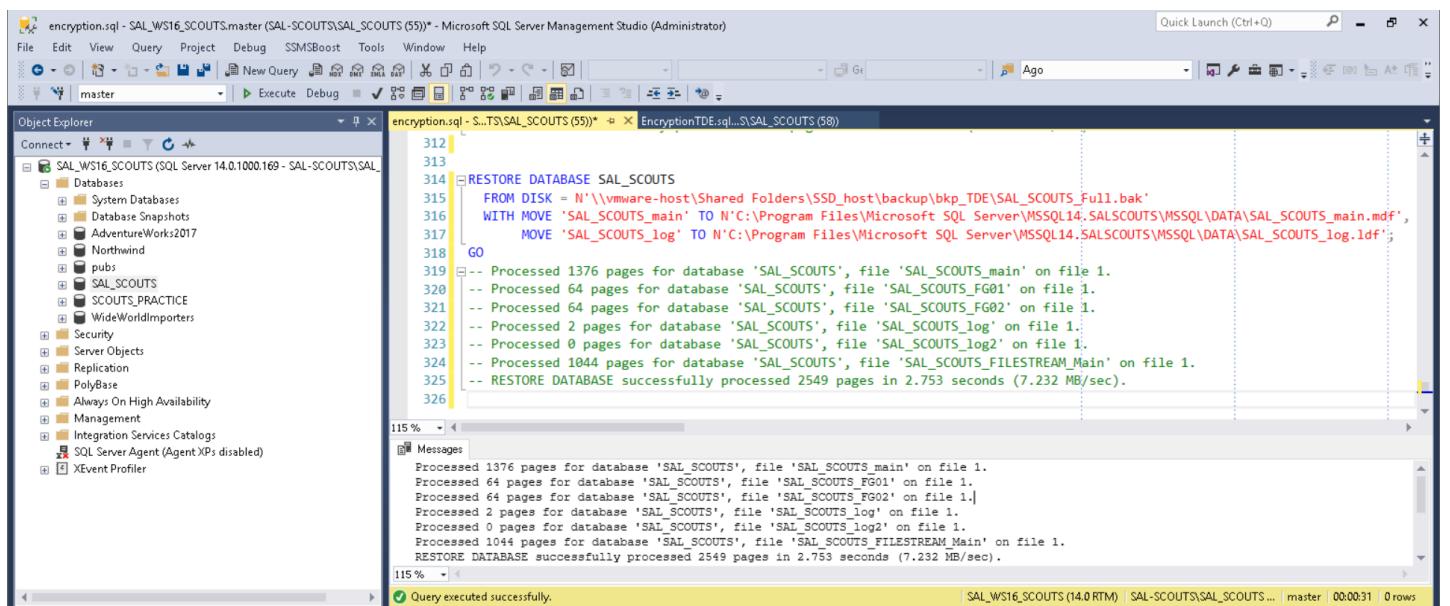


Hechos los *backups*, vamos a restaurar nuestra base de datos. Si restauramos **SAL\_SCOUTS** sin haber restaurado el certificado nos aparece este mensaje de error indicándonos que no puede encontrarlo:

```
--Msg 33111, Level 16, State 3, Line 20
--Cannot find server certificate with thumbprint
'0xD13BB7301FDF9ED16F0377F17D7CB414F2BCAE5A'.
--Msg 3013, Level 16, State 1, Line 20
--RESTORE DATABASE is terminating abnormally.
```

Pero como disponemos de su copia de seguridad, pues entonces podemos restaurarla, ejecutamos las sentencias:

```
RESTORE DATABASE SAL_SCOUTS
    FROM DISK = N'\\vmware-host\Shared Folders\SSD_host\backup\bkp_TDE\Sal_SCOUTS_Full.bak'
    WITH MOVE 'SAL_SCOUTS_main' TO N'C:\Program Server\MSSQL14.SALSCOUTS\MSSQL\DATA\Sal_SCOUTS_main.mdf',
        MOVE 'SAL_SCOUTS_log' TO N'C:\Program Server\MSSQL14.SALSCOUTS\MSSQL\DATA\Sal_SCOUTS_log.ldf';
GO
-- Processed 1376 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_main' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG01' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG02' on file 1.
-- Processed 2 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- Processed 1044 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FILESTREAM_Main' on file 1.
-- RESTORE DATABASE successfully processed 2549 pages in 2.753 seconds (7.232 MB/sec).
```



## • 2.5. Funciones

Una función es un conjunto de sentencias que operan como una unidad lógica.

Una función tiene un nombre, retorna un parámetro de salida y opcionalmente acepta parámetros de entrada. Las funciones de SQL Server no pueden ser modificadas, las funciones definidas por el usuario en sí.

SQL Server ofrece varios tipos de funciones para realizar distintas operaciones. Se pueden clasificar de la siguiente manera:

- De agregado: realizan operaciones que combinan varios valores y retornan un único valor. Son `COUNT()`, `SUM()`, `MIN()` y `MAX()`...

- Escalares: toman un solo valor y retornan un único valor. Pueden agruparse de la siguiente manera:

- ~ De configuración: devuelven información referida a la configuración, por ejemplo, `SELECT @@VERSION` → devuelve la fecha, versión y tipo de procesador de SQL Server.

- ~ De cursos: devuelven información sobre el estado de un cursor.

- ~ De fecha y hora: operan con valores `datetime` y `smalldatetime`. Reciben un parámetro de tipo fecha y hora y retornan un valor de cadena, numérico o de fecha y hora.

- ~ Matemáticas: realizan operaciones numéricas, geométricas y trigonométricas.

- ~ De metadatos: informan sobre las bases de datos y los objetos.

- ~ De seguridad: devuelven información referente a usuarios y funciones.

- ~ De cadena: operan con valores `CHAR`, `VARCHAR`, `NCHAR`, `NVARCHAR`, `BINARY` y `VARBINARY` y devuelven un valor de cadena o numérico.

- ~ Del sistema: informan sobre opciones, objetos y configuraciones del sistema, por ejemplo, `SELECT USER_NAME();`

- ~ Estadísticas del sistema: retornan información referente al rendimiento del sistema.

- ~ Texto e imagen: realizan operaciones con valor de entrada de tipo `TEXT` o `IMAGE` y retornan información referente al mismo.

- De conjuntos de filas: retornan conjuntos de registros. Se pueden emplear las funciones del sistema en cualquier lugar en el que se permita una expresión en una sentencia `SELECT`.

Básicamente, son sentencias las cuales hemos estado empleado a lo largo del curso y que funcionan de manera semejante a los procedimientos almacenados y a los *triggers*. Como ejemplo para funciones transformaremos la estructura del procedimiento almacenado `profit_lossCalc` para demostrar su versión como función *T-SQL* :

```
USE SAL_SCOUTS
GO

DROP FUNCTION IF EXISTS SAL_fn_profit_lossCalc
GO

CREATE OR ALTER FUNCTION SAL_fn_profit_lossCalc (
@date_1 AS date,
@date_2 AS date
) RETURNS MONEY
AS
BEGIN
    DECLARE @don MONEY -- total donaciones
    DECLARE @tot MONEY -- total totales gastos
    DECLARE @res MONEY -- total calculado

    SET @don = (SELECT SUM(importe) FROM SAL_DONACION
    WHERE fecha_donacion BETWEEN @date_1 AND @date_2);
    -- total de donaciones en el rango de fechas indicadas
    SET @tot = (SELECT SUM(total) FROM SAL_FACTURA_MATERIAL
    WHERE fecha_compra BETWEEN @date_1 AND @date_2);
    -- total de gastos en el rango de fechas indicadas

    SET @res = @don - @tot;

    RETURN @res;
END ;
GO

PRINT dbo.SAL_fn_profit_lossCalc('2017-01-01','2017-12-31')
GO
```

The screenshot shows the SSMS interface. The Object Explorer on the left lists databases like System Databases, AdventureWorks2017, Northwind, pubs, and SAL\_SCOUTS, along with their objects such as Tables, Views, and Stored Procedures. The query results window on the right displays a T-SQL script for calculating profit/loss between two dates. The output pane at the bottom shows the result of the execution.

```
-- total de gastos en el rango de fechas indicadas
22
23
24     SET @res = @don - @tot;
25
26     RETURN @res;
27 END;
28 GO
29
30 PRINT dbo.SAL_fn_profit_lossCalc('2017-01-01','2017-12-31');
31 GO
32
```

160 %

Messages

4101345880.25

160 %

Query executed successfully.

A continuación hablaremos sobre las funciones relacionadas con la encriptación como lo son: ***DDM, Dynamic Data Masking*** (Enmascaramiento Dinámico de Datos) y ***RLS, Row-Level Security*** (Seguridad a Nivel de Filas).

#### · 2.5.1. DDM (*Dynamic Data Masking*)

**DDM - Dynamic Data Masking** (Enmascaramiento Dinámico de Datos) es una función que trata de limitar u ocultar información sensible sin requerir cambios en las aplicaciones. Los datos en la base de datos realmente no se modifican, se alteran “al vuelo” de forma que cuando las consultas devuelven resultados se aplican las máscaras apropiadas. Esto hace que esta funcionalidad sea sencilla de implementar ya que no requiere cambios sustanciales y sea bastante transparente para las aplicaciones que utilizan los datos enmascarados.

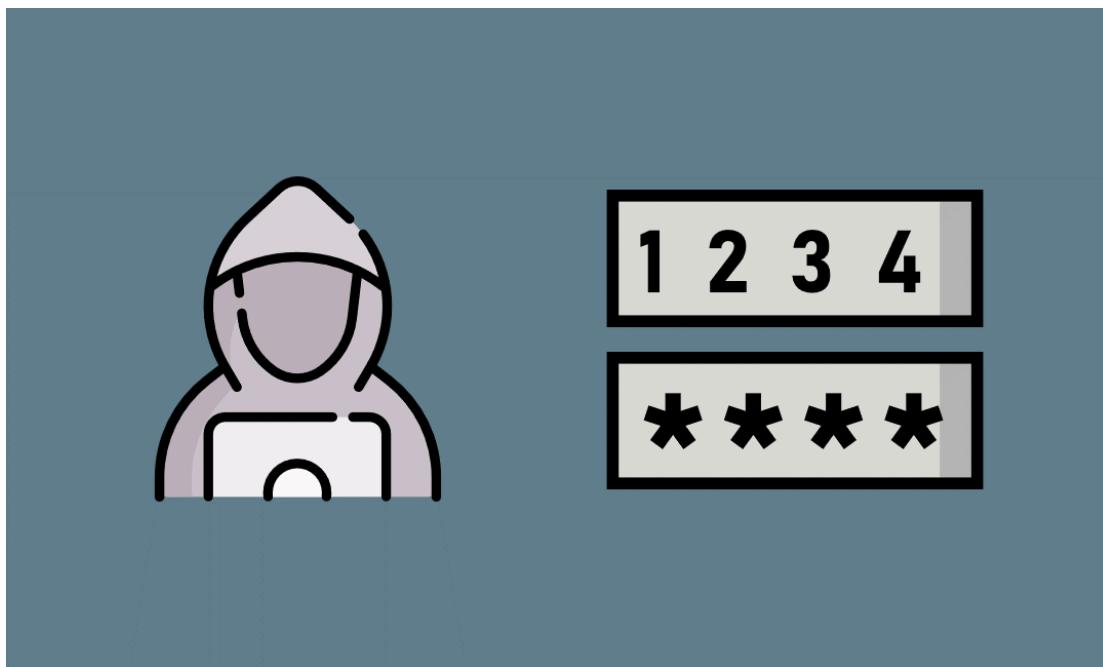
Una de las primeras cosas que debemos considerar al utilizar esta técnica es que no es una alternativa a la encriptación y que tiene ciertas limitaciones que pueden volverla “peligrosa” si se permite acceso al dato enmascarado de forma directa. Si no tenemos más remedio que utilizarla, se suele recomendar encarecidamente el uso de procedimientos almacenados o mecanismos de control sobre la tipología de operaciones que se pueden realizar sobre las tablas con datos enmascarados. Incluso pequeños extractos de datos necesitan ser creados con precaución, si son para consumo público, si son para información, para crear una muestra de datos como trabajo de desarrollo, o para propósitos de demostración y entrenamiento. Los datos sensibles pueden “ocultarse” en lugares inesperados.

Data Masking se le suele llamar algunas veces protección o desinfección de datos, es un término para la tecnología y procesos que son utilizados para **anonimizar** o **(p)seudominizar** datos personales, privados o sensibles.

· La **(p)seudonimización** trata de mantener parte de los datos, mientras se anonimizan los elementos de datos de identificación directa (es decir, los datos personales) de manera consistente, para permitir que los datos sean usados para propósitos de información o análisis sin revelar datos personales. Por ejemplo, la **(p)seudonimización** de una base de datos modificará el nombre y otros identificadores de individuos, pero dejará el resto de los datos, tales como posibles historiales de compra o intervenciones médicas intactas.

· Los datos **anonimizados** son simplemente datos de los que las personas, el “sujeto de datos” ya no puede ser identificado. Un proceso de **anonimización** lo hará imposible, o al menos extremadamente impráctico, de identificar el sujeto de datos, incluso mientras pretendan mantener la verosimilitud de los datos, para que así parezca real. Esto quiere decir que no es suficiente con solo enmascarar los elementos de datos de identificación directa tales como el nombre de la persona o su DNI. Requiere medidas adicionales para prevenir la identificación, el cual variará dependiendo en los datos y por qué necesitas anonimizarlos, pero generalmente involucrará barajar, revolver o cualquier otro cambio en las relaciones entre las personas y el resto de los datos. Por ejemplo, los hábitos de compra serán cambiados tal que así por los nombres, o las intervenciones médicas serán asignadas a otras personas.

La ventaja de la **(p)seudonimización** es que todos los enlaces relacionales permanecen intactos, y la distribución de datos está garantizada a que sea como los datos reales. La desventaja es que viene con implicaciones de seguridad obvias que limitarán su uso fuera de los entornos con protocolos de seguridad que coinciden con los de los datos reales. Con la **anonimización**, el dato resultante está seguro para su uso, pero si nos da la curiosidad de “jugueteear” con las referencias de las **FK** (*Foreign Keys*) para prevenir ataques de inferencia, llega a ser más difícil de mantener la distribución adecuada.



Muchas personas se preguntan el porqué de enmascarar los datos, viene a ser que, la custodia de los datos tiene un acto de balanceo difícil entre conformarse con las leyes de privacidad, la discreción y las obligaciones de confidencialidad, por un lado, y la necesidad de hacer ciertas partes de que esos datos disponibles para trabajo de desarrollo, entrenamiento, testeo, investigaciones médicas, gobierno abierto, seguridad y la Ley. Ya deja de ser legalmente posible hacer esto si los datos contienen información que deben ser protegidos. Los datos profesionales son enfrentados contra la tarea práctica de proveer datos, legalmente, en una manera en la que no pueda ser posible revelar, por ataques de inferencia, información privada sobre personas o información confidencial de entidades empresariales.

Esto es una tarea temiblemente compleja de anonimizar o (p)seudonimizar los datos de una base de datos entera, hacer su desarrollo o testeo. Mientras esto sea relativamente de (p)seudonimizar los contenidos de una única tabla, o incluso un puñado de tablas, entonces los datos reales no pueden ser recogidos por una simple inspección, es más difícil de conseguir una robusta (p)seudonimización que está a prueba contra las descodificaciones de una persona experta. Y de ahí ha habido varios casos en donde la información personal de gente ha sido extraída desde público, datos “(p)seudonimizados”.

Existen varios métodos de enmascarar datos dependiendo del tipo de dato a ser enmascarado y qué requisitos de enmascaramiento requieren.

- Sustitución completa: los datos en una o más columnas de la tabla pueden ser aleatoriamente sustituidos con valores de una lista apropiada o por un generador de datos que puede aportar valores creíbles.
- Barajeo de datos: los datos en una o más columnas pueden ser barajados aleatoriamente en contra de los campos clave. Esto puede requerir que ciertas filas sean respetadas o de muchas otras aplicaciones a ser aplicadas.
- Búsqueda y reemplazo de enmascarado: esto es usado para modificar datos de texto por significados de reglas o expresiones comunes. A menudo se suelen quitar nombres o identificadores desde un texto, sustituyendo una longitud aleatoria de “###”.
- Variación numérica: cada valor numérico o de fecha en una columna puede ser variado por un porcentaje aleatorio mientras mantenga la variante original, el rango y la distribución.

Esta funcionalidad de *Data Masking* estuvo disponible desde **SQL Server 2016** y por defecto se nos proporcionan 4 funciones de enmascarado:

- Función **`default()`**: esta función realiza distintos tipos de enmascarados según el tipo de dato. Si el dato a enmascarar es una cadena de texto se sustituirá por **XXXX**, si es un número se sustituirá por un 0.
- Función **`email()`**: esta función mantendrá la primera letra y el sufijo al final. Por ejemplo sal@scouts.com se ofuscaría como **sXX@XXXXXX.com**.
- Función **`random()`**: esta función generará un número aleatorio dentro de un rango como máscara al aplicar.
- Función **`partial()`**: con esta función definiremos cuántos caracteres al principio y al final de una cadena queremos dejar visibles y qué patrón utilizaremos para el resto. Por ejemplo, si utilizamos una función **`partial(0, 'XXXX-XXXX-XXXX', 4)`**, mostraríamos únicamente los 4 últimos de una tarjeta de crédito típica.

Para saber qué tablas han sido enmascaradas creamos un procedimiento almacenado llamado **`ShowMaskingStatus`**. **`ShowMaskingStatus`** realiza una consulta a las tablas de metadatos con **`sys.masked_columns`** realizando una **`JOIN`** con **`sys.tables`** que nos permite saber el nombre de la columna, a qué tabla pertenece, en qué estado se encuentra su enmascaramiento y qué función está utilizando. Éstas son las sentencias a ejecutar:

```

CREATE OR ALTER PROC ShowMaskingStatus
AS
BEGIN
SET NOCOUNT ON
SELECT c.name, tbl.name AS table_name, c.is_masked, c.masking_function
FROM sys.masked_columns AS c
JOIN sys.tables AS tbl
ON c.[object_id] = tbl.[object_id]
WHERE is_masked = 1;
END
GO

EXEC ShowMaskingStatus;
GO

-- name table_name is_masked masking_function
-- VACÍO

```

Como ejemplo para este apartado realizaremos un enmascaramiento a las tablas que contengan información privada, en este caso, datos como las donaciones, información de los scouts, el personal de la asociación, etc. Modificaremos las tablas: *SAL\_DONACION*, *SAL\_ENTIDAD*, *SAL\_FACTURA\_MATERIAL*, *SAL\_PERSONAL*, *SAL\_SCOUT*. Y una vez modificadas, comprobaremos si las filas de las columnas han sido enmascaradas iniciando sesión en las instancias clientes del dominio. Estas son las sentencias ejecutadas:

```

USE SAL_SCOUTS
go

CREATE OR ALTER PROC ShowMaskingStatus
AS
BEGIN
SET NOCOUNT ON
SELECT c.name, tbl.name AS table_name, c.is_masked, c.masking_function
FROM sys.masked_columns AS c
JOIN sys.tables AS tbl
ON c.[object_id] = tbl.[object_id]
WHERE is_masked = 1;
END
GO

EXEC ShowMaskingStatus;
GO

-- name table_name is_masked masking_function
-- VACÍO

-- SAL_DONACION

SELECT top 5 nombre_donante, importe FROM SAL_DONACION
GO

```

```

-- nombre_donante      importe
-- Ulla Sampson        364557,55
-- Holmes Hopkins      725784,49
-- Fletcher Johnson    749478,39
-- Iona Holder         338829,02
-- Mechelle Hernandez  306672,87

ALTER TABLE SAL_DONACION
    ALTER COLUMN nombre_donante
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_DONACION
    ALTER COLUMN importe
        ADD MASKED WITH (FUNCTION = 'random(-999999999,999999999)');
GO

EXEC ShowMaskingStatus;
GO

-- name                  table_name      is_masked  masking_function
-- nombre_donante        SAL_DONACION   1          default()
-- importe                SAL_DONACION   1          random(-1e+009, 1e+009)

-- SAL_ENTIDAD

SELECT top 5 CIF,nombre_entidad,direccion,pais,telefono from SAL_ENTIDAD
GO

-- CIF                   nombre_entidad  direccion       pais      telefono
-- U5194934   Nibh Corp.      1492 Integer Street  Romania  546488754
-- B6431137   Pede Praesent Limited 3372 Tincidunt Rd.  Pakistan 812831802
-- B9556348   Interdum Feugiat Inc. Ap #436-7321 Tellus Av. Switzerland
-- 874622896
-- D4427584   Pellentesque Tincidunt Tempus Associates 7749 Rhoncus Rd.
-- Bouvet Island 161824135
-- P9787672   Pellentesque Eget Dictum LLP     112-6972 Pharetra Rd. Montserrat
-- 556314139

ALTER TABLE SAL_ENTIDAD
    ALTER COLUMN CIF
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_ENTIDAD
    ALTER COLUMN nombre_entidad
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_ENTIDAD
    ALTER COLUMN direccion
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_ENTIDAD
    ALTER COLUMN pais
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_ENTIDAD
    ALTER COLUMN telefono
        ADD MASKED WITH (FUNCTION = 'default()');
GO

```

```

EXEC ShowMaskingStatus;
GO

-- name          table_name      is_masked  masking_function
-- nombre_donante SAL_DONACION  1          default()
-- importe        SAL_DONACION  1          random(-1e+009, 1e+009)
-- CIF           SAL_ENTIDAD   1          default()
-- nombre_entidad SAL_ENTIDAD   1          default()
-- direccion     SAL_ENTIDAD   1          default()
-- pais          SAL_ENTIDAD   1          default()
-- telefono      SAL_ENTIDAD   1          default()

-- SAL_FACTURA_MATERIAL

SELECT top 5 producto,cod_producto,num_lote,cantidad,precio,subtotal,total from
SAL_FACTURA_MATERIAL
GO

-- producto      cod_producto    num_lote      cantidad  precio    subtotal
-- total          83054          HTS95ELP5DG    29        190,83    5534,07
-- Ziplock        5534,07
-- Ziplock        43417          JFP74IIF8TP    12        177,08    2124,96
-- Ice_chest      2124,96
-- Ice_chest      45414          AOY36LYE7JS    42        70,30     2952,60
-- Can_opener     2952,60
-- Can_opener     64114          KVI87HBP3MK    33        178,76    5899,08
-- bed            5899,08
-- bed            40866          RHX43NRG6MJ    49        126,94    6220,06
-- bed            6220,06

ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN producto
    ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN cod_producto
    ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN num_lote
    ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN cantidad
    ADD MASKED WITH (FUNCTION = 'random(-9999,9999)');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN precio
    ADD MASKED WITH (FUNCTION = 'random(-9999,9999)');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN subtotal
    ADD MASKED WITH (FUNCTION = 'random(-9999,9999)');
ALTER TABLE SAL_FACTURA_MATERIAL
  ALTER COLUMN total
    ADD MASKED WITH (FUNCTION = 'random(-9999,9999)');
GO

```

```

EXEC ShowMaskingStatus;
GO

-- name          table_name      is_masked masking_function
-- nombre_donante SAL_DONACION  1        default()
-- importe       SAL_DONACION  1        random(-1e+009, 1e+009)
-- CIF           SAL_ENTIDAD   1        default()
-- nombre_entidad SAL_ENTIDAD   1        default()
-- direccion     SAL_ENTIDAD   1        default()
-- pais          SAL_ENTIDAD   1        default()
-- telefono      SAL_ENTIDAD   1        default()
-- producto      SAL_FACTURA_MATERIAL 1        default()
-- cod_producto  SAL_FACTURA_MATERIAL 1        default()
-- num_lote      SAL_FACTURA_MATERIAL 1        default()
-- cantidad      SAL_FACTURA_MATERIAL 1        random(-9999, 9999)
-- precio        SAL_FACTURA_MATERIAL 1        random(-9999, 9999)
-- subtotal      SAL_FACTURA_MATERIAL 1        random(-9999, 9999)
-- total         SAL_FACTURA_MATERIAL 1        random(-9999, 9999)

-- SAL_PERSONAL

SELECT nombre, apellido1, apellido2, DNI, fecha_nacimiento, telefono, direccion FROM
SAL_PERSONAL
GO

-- nombre          apellido1  apellido2      DNI    fecha_nacimiento      telefono
-- direccion
-- Juan            Robles      Martinez      87654321A  1967-06-18
-- 601234567  C/ Penacho 14 3° Dcha
-- Saúl            Altoubah    León          12345678B  1997-03-08
-- 600000000  Avda. MiCalle 1 1°A
-- Lucía           García      Caamaño      24688642C  1973-09-02
-- 612345678  C/ Estructura 100 4° IZQA.
-- Santiago        Descalzo    De La Torre    13579135D  1996-06-06
-- 698765432  fake St. 123
-- Alberto          Castaña     Manzana      66502374R  1986-12-28
-- 636963636  Avda. Pepe 7 2°B
-- Alma             Martillo    Puertas      47445203M  1981-06-08
-- 654254254  C/ Jonás 12 8° DCHA

ALTER TABLE SAL_PERSONAL
ALTER COLUMN nombre
ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_PERSONAL
ALTER COLUMN apellido1
ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_PERSONAL
ALTER COLUMN apellido2
ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_PERSONAL
ALTER COLUMN DNI
ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');

```

```

ALTER TABLE SAL_PERSONAL
    ALTER COLUMN fecha_nacimiento
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_PERSONAL
    ALTER COLUMN telefono
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_PERSONAL
    ALTER COLUMN direccion
        ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
GO

```

```

EXEC ShowMaskingStatus;
GO

```

	name	table_name	is_masked	maksing_function
--	nombre_donante	SAL_DONACION	1	default()
--	importe	SAL_DONACION	1	random(-1e+009, 1e+009)
--	CIF	SAL_ENTIDAD	1	default()
--	nombre_entidad	SAL_ENTIDAD	1	default()
--	direccion	SAL_ENTIDAD	1	default()
--	pais	SAL_ENTIDAD	1	default()
--	telefono	SAL_ENTIDAD	1	default()
--	producto	SAL_FACTURA_MATERIAL	1	default()
--	cod_producto	SAL_FACTURA_MATERIAL	1	default()
--	num_lote	SAL_FACTURA_MATERIAL	1	default()
--	cantidad	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
--	precio	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
--	subtotal	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
--	total	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
--	nombre	SAL_PERSONAL	1	partial(1, "*****", 0)
--	apellido1	SAL_PERSONAL	1	partial(1, "*****", 0)
--	apellido2	SAL_PERSONAL	1	partial(1, "*****", 0)
--	DNI	SAL_PERSONAL	1	partial(1, "*****", 0)
--	fecha_nacimiento	SAL_PERSONAL	1	default()
--	telefono	SAL_PERSONAL	1	default()
--	direccion	SAL_PERSONAL	1	partial(1, "*****", 0)

```
-- SAL_SCOUT
```

```

SELECT top 5
    nombre, apellido1, apellido2, DNI, fecha_nacimiento, direccion, telefono, edad
    FROM
SAL_SCOUT
GO

```

	nombre	apellido1	apellido2	DNI	fecha_nacimiento	direccion
--	telefono		edad			
--	Yeo	Mccarthy	Holland		85179121E	2005-07-02 Ap #939-4170 In Road
244528785		15				
--	Phillip	Banks	Manning		18644331L	1997-04-15 P.O. Box 714, 8603
Dolor Street	587837181		23			
--	Jael	Joyce	Lott	51768952W	2003-02-16	6081 Diam. Ave
956884469		17				
--	Meredith	Cohen	Ball	56435386H	2000-07-12	P.O. Box 698, 6047
Lacus St.	873854814		20			

```
-- Stone Mccoy      Hall      19386398S 2014-07-21          Ap #650-5440  DUI,
Street      997139111  6
```

```
ALTER TABLE SAL_SCOUT
    ALTER COLUMN nombre
        ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN apellido1
        ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN apellido2
        ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN DNI
        ADD MASKED WITH (FUNCTION = 'partial(1,"*****",0)');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN fecha_nacimiento
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN direccion
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN telefono
        ADD MASKED WITH (FUNCTION = 'default()');
ALTER TABLE SAL_SCOUT
    ALTER COLUMN edad
        ADD MASKED WITH (FUNCTION = 'default()');
GO
```

```
EXEC ShowMaskingStatus;
GO
```

name	table_name	is_masked	masking_function
nombre_donante	SAL_DONACION	1	default()
importe	SAL_DONACION	1	random(-1e+009, 1e+009)
CIF	SAL_ENTIDAD	1	default()
nombre_entidad	SAL_ENTIDAD	1	default()
direccion	SAL_ENTIDAD	1	default()
pais	SAL_ENTIDAD	1	default()
telefono	SAL_ENTIDAD	1	default()
producto	SAL_FACTURA_MATERIAL	1	default()
cod_producto	SAL_FACTURA_MATERIAL	1	default()
num_lote	SAL_FACTURA_MATERIAL	1	default()
cantidad	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
precio	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
subtotal	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
total	SAL_FACTURA_MATERIAL	1	random(-9999, 9999)
nombre	SAL_PERSONAL	1	partial(1, "*****", 0)
apellido1	SAL_PERSONAL	1	partial(1, "*****", 0)
apellido2	SAL_PERSONAL	1	partial(1, "*****", 0)
DNI	SAL_PERSONAL	1	partial(1, "*****", 0)
fecha_nacimiento	SAL_PERSONAL	1	default()
telefono	SAL_PERSONAL	1	default()
direccion	SAL_PERSONAL	1	partial(1, "*****", 0)
nombre	SAL_SCOUT	1	partial(1, "*****", 0)
apellido1	SAL_SCOUT	1	partial(1, "*****", 0)

```

-- apellido2      SAL_SCOUT          1      partial(1, "*****", 0)
-- DNI            SAL_SCOUT          1      partial(1, "*****", 0)
-- fecha_nacimiento SAL_SCOUT        1      default()
-- direccion       SAL_SCOUT          1      default()
-- telefono        SAL_SCOUT          1      default()
-- edad            SAL_SCOUT          1      default()

```

Como se puede observar, no hemos utilizado la función `email()` ya que en ninguna de las columnas se incluye un campo para correos electrónicos. Ahora el inconveniente es que los empleados, excepto el coordinador (`sysadmin`) y no podrán visualizar los valores reales de las tablas (Resp. Materiales no se ve afectado ya que sus tablas no fueron seleccionadas):

~ Tesorero (ACM\_SCOUTS) → ***SAL\_DONACIÓN – SAL\_FACTURA\_MATERIAL:***

The screenshot shows the Azure Data Studio interface with a query results grid. The connection is to SQLQuery\_1 - 192.168.1.100.master (ACM\_SCOUTS). The query executed is:

```

USE SAL_SCOUTS
GO
SELECT * FROM SAL_DONACION;
SELECT * FROM SAL_FACTURA_MATERIAL;
GO

```

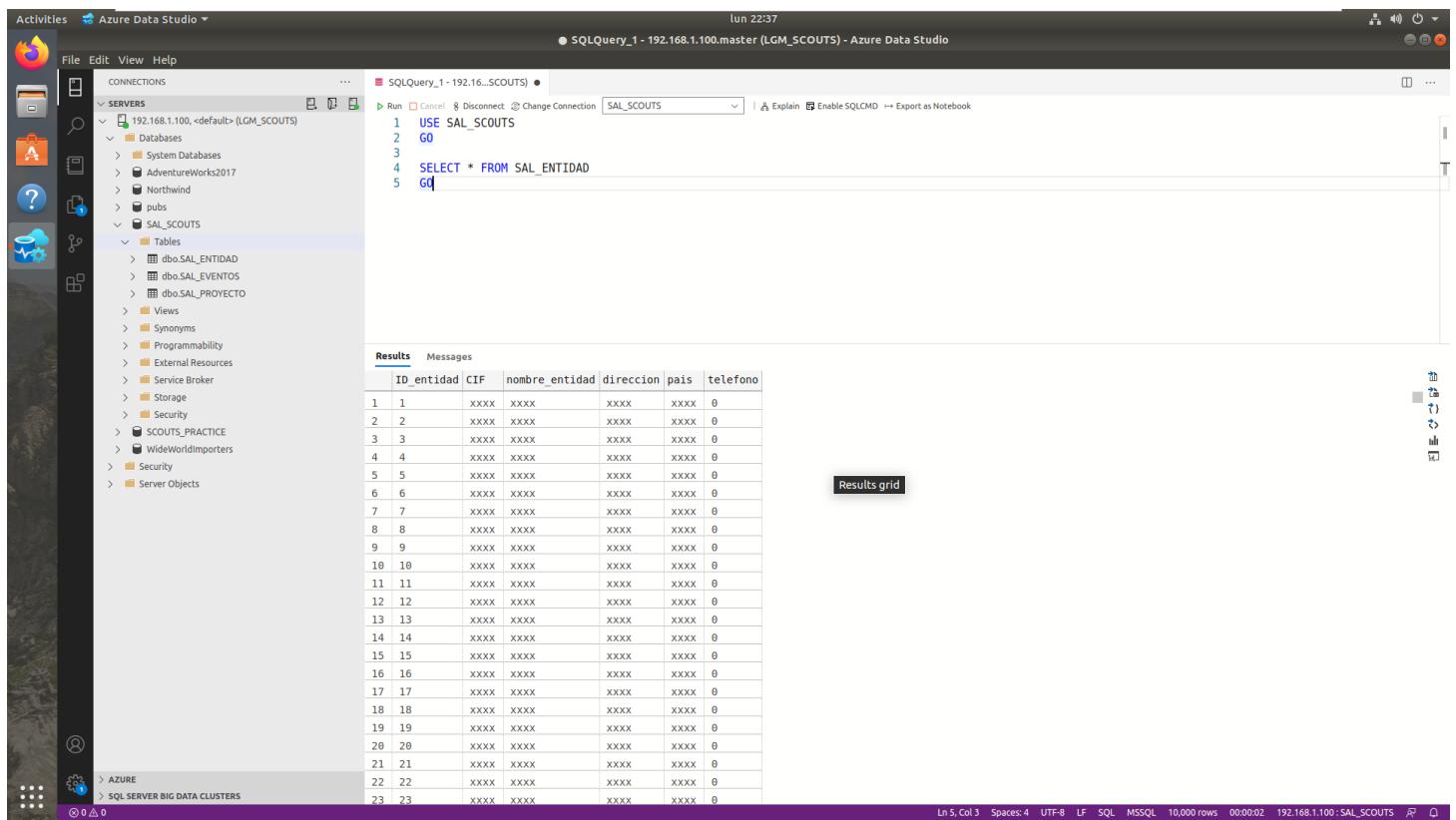
The results grid displays two tables:

ID_donacion	fecha_donacion	nombre_donante	importe	SAL_EVENTOS_ID_evento	SAL_TESORERO_ID_personal
1	2018-03-03	XXXX	-16814.1378	1	5
2	2017-05-18	XXXX	6996.2913	5	5
3	2017-03-04	XXXX	10547.6347	4	5
4	2017-07-20	XXXX	-93669.1060	6	5
5	2017-08-26	XXXX	94960.1705	1	5
6	2017-08-06	XXXX	-70598.1396	6	5
7	2017-06-08	XXXX	-120345.0138	3	5
8	2017-08-09	XXXX	-94582.6600	1	5
9	2017-02-04	XXXX	-17970.0458	2	5
10	2018-01-17	XXXX	-79581.1443	4	5
11	2017-09-10	XXXX	-116145.0449	1	5
12	2018-02-05	XXXX	25810.2028	3	5
13	2017-08-04	121653.1531	3	5	
14	2018-01-17	XXXX	-99647.0758	3	5

ID_factura	SAL_ENTIDAD_ID_entidad	fecha_compra	num_order	fecha_pedido	producto	cod_producto	num_lote	cantidad	precio	subtotal	total	
1	1	7916	571925	2020-04-05	XXXX	0	XXXX	430	8289.1115	1628.1027	-8734.4681	
2	2	1030	686421	2019-03-20	XXXX	0	XXXX	-115	8349.4344	8741.1523	-398.6854	
3	3	9510	712481	2020-06-20	XXXX	0	XXXX	9170	-256.9432	9856.1621	-3805.4034	
4	4	2121	2021-01-13	333369	2021-01-13	XXXX	0	9743	2118.2159	2774.2940	9494.9653	
5	5	6510	2017-07-12	547645	2017-07-12	XXXX	0	XXXX	283	3282.0208	-4665.7486	-9316.6981
6	6	7349	2019-08-17	774363	2019-08-17	XXXX	0	XXXX	-6403	1549.0473	8764.6245	-688.9481
7	7	5392	2020-02-17	269611	2020-02-17	XXXX	0	6161	3237.5962	-2688.1079	4601.5258	
8	8	1398	2018-07-03	707775	2018-07-03	XXXX	0	XXXX	-4618	829.6183	-2614.2889	5650.9076
9	9	1924	2018-04-26	124807	2018-04-26	XXXX	0	XXXX	-8999	4896.7786	4609.6909	-4938.5437
10	10	6663	2017-10-17	545214	2017-10-27	XXXX	0	XXXX	-2249	-133.0138	-3225.4406	5227.0999
11	11	9385	2019-03-06	808331	2019-03-06	XXXX	0	XXXX	-8138	-6715.6251	-3804.1596	-5993.8371
12	12	2589	2017-04-07	786680	2017-04-07	XXXX	0	XXXX	-2496	-9010.2049	4940.4433	4795.5149
13	13	9950	2019-02-02	642691	2019-02-02	XXXX	0	3842	3070.3828	599.4125	7732.9061	
14	14	3461	2020-05-09	257169	2020-05-09	XXXX	0	XXXX	-9018	3021.5091	1612.7383	-2901.0888

## ~ Resp. Com. RR.SS. (LGM SCOUTS) → SAL\_ENTIDAD:



The screenshot shows the Azure Data Studio interface. The left sidebar displays a tree view of database connections, showing '192.168.1.100, <default> (LGM\_SCOUTS)' with its databases, system databases, and tables. The 'Tables' section under 'SAL\_SCOUTS' is selected. The main pane contains a SQL query window with the following code:

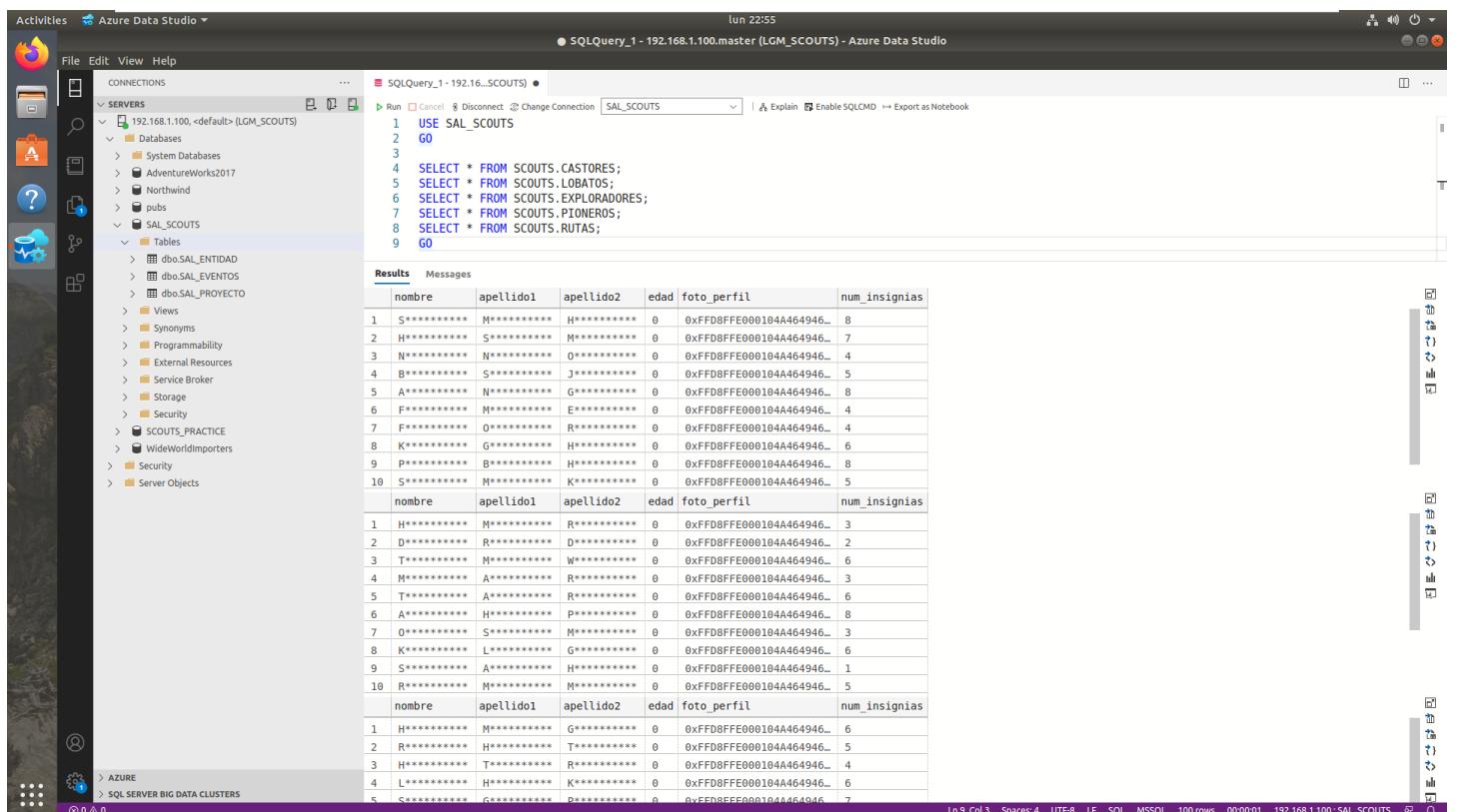
```

USE SAL_SCOUTS
GO
SELECT * FROM SAL_ENTIDAD
GO

```

The results pane shows a grid of data with columns: ID\_entidad, CIF, nombre\_entidad, direccion, pais, and telefono. The data consists of 23 rows, each with 'XXXX' values for most fields. The bottom status bar indicates the results grid has 10,000 rows.

## ~ SAL\_SCOUTS > SCOUTS.CASTORES/LOBATOS/EXPLOR./PIONEROS/RUTAS:



The screenshot shows the Azure Data Studio interface. The left sidebar displays a tree view of database connections, showing '192.168.1.100, <default> (LGM\_SCOUTS)' with its databases, system databases, and tables. The 'Tables' section under 'SAL\_SCOUTS' is selected. The main pane contains a SQL query window with the following code:

```

USE SAL_SCOUTS
GO
SELECT * FROM SCOUTS.CASTORES;
SELECT * FROM SCOUTS.LOBATOS;
SELECT * FROM SCOUTS.EXPLORADORES;
SELECT * FROM SCOUTS.PIONEROS;
SELECT * FROM SCOUTS.RUTAS;
GO

```

The results pane shows two grids of data. The first grid has columns: nombre, apellido1, apellido2, edad, foto\_perfil, and num\_insignias. It contains 10 rows of data with various names and birth years. The second grid also has columns: nombre, apellido1, apellido2, edad, foto\_perfil, and num\_insignias. It contains 5 rows of data. The bottom status bar indicates the first grid has 10,000 rows and the second grid has 100 rows.

## ~ Presidente (JRM\_SCOUTS) → SAL\_PERSONAL:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including the SAL\_WS16\_SCOUTS database and its tables. In the center, a query window titled 'SQLQuery1.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (JRM\_SCOUTS (56))' contains the following SQL code:

```
USE SAL_SCOUTS
GO

SELECT * FROM SAL_PERSONAL
GO
```

The results pane shows a table with 6 rows of data from the SAL\_PERSONAL table. The columns are: ID\_personal, nombre, apellido1, apellido2, DNI, fecha\_nacimiento, telefono, direccion, foto\_perfil, ID\_img, num\_insignias, and anio. The data is as follows:

ID_personal	nombre	apellido1	apellido2	DNI	fecha_nacimiento	telefono	direccion	foto_perfil	ID_img	num_insignias	anio
1	J-----	R-----	M-----	8-----	1900-01-01	0	C-----	0xFD8FFE100BC4578696600049492A00080000000600120...	88FB5A45-E4D8-43DA-B406-4AE9D0416E5B	15	20
2	S-----	A-----	L-----	1-----	1900-01-01	0	A-----	0xFD8FFE000104A46494600010000010001000FFDB00...	2B8CD23D-9420-4C9-8B7E-500F4DAEE0C2	5	3
3	L-----	G-----	C-----	2-----	1900-01-01	0	C-----	0xFD8FFE100BC4578696600049492A00080000000600120...	3FBC416C-676A-4C87-85CE-15201DAEFE52	15	17
4	S-----	D-----	D-----	1-----	1900-01-01	0	F-----	0x89504E470D0A1A0A00000004948445200000200000002...	3269C736-1685-49A3-991F-D0BDEE87B750	6	4
5	A-----	C-----	M-----	6-----	1900-01-01	0	A-----	0x89504E470D0A1A0A00000004948445200000200000002...	E32CE7F3-2B11-44FB-8D2C-831B6714FBC5	12	9
6	A-----	M-----	P-----	4-----	1900-01-01	0	C-----	0x89504E470D0A1A0A0000000494844520000020000002...	208206B9-00A8-4786-A860-EEA212F7CA83	8	11

At the bottom of the results pane, a message indicates: 'Query executed successfully.'

Una manera de poder visualizar los datos enmascarados sería garantizarles el permiso **UNMASK** al personal con la sentencia **GRANT UNMASK TO Presidente/Tesorero/Responsable** o crear un usuario que contenga esos permisos y que hagan la impersonación de tal usuario, el ejemplo se mostraría así:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including the SAL\_WS16\_SCOUTS database. In the center, a query window titled 'data\_masking.sql - ...\\SAL\_SCOUTS (53)' contains the following SQL code:

```
276 -- UNI          SAL_SCOUT
277 -- fecha_nacimiento SAL_SCOUT
278 -- direccion      SAL_SCOUT
279 -- telefono       SAL_SCOUT
280 -- edad           SAL_SCOUT
281
282
283
284 GRANT UNMASK TO Presidente;
285 GO
```

The results pane shows a message: 'Commands completed successfully.'

At the bottom of the results pane, a message indicates: 'Query executed successfully.'

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery1.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (IRM\_SCOUTS (56))\* - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, displaying the database structure for "SAL\_WS16\_SCOUTS (SQL Server 14.0.1000)". The right pane shows the results of a query:

```
USE SAL_SCOUTS
GO

SELECT * FROM SAL_PERSONAL
GO
```

The results grid displays the following data:

ID_personal	nombre	apellido1	apellido2	DNI	fecha_nacimiento	telefono	direccion	foto_perfil	ID_img
1	Juan	Robles	Martinez	87654321A	1967-06-18	601234567	C/ Penacho 14 3º Dcha	0xFFD8FFE100BC45786966000049492A0000800000000600120...	88F65A45-E4D8-43DA-B406-4AE9D4164E5B
2	Saúl	Altoubah	León	12345678B	1997-03-08	600000000	Avda. Micalle 1 11A	0xFFD8FFE00010446494600010100001000010000FFFB00...	288CD23D-9420-4C9-8B7E-500FA0AE0C2
3	Lucía	García	Caamaño	24688642C	1973-09-02	612345678	C/ Estatua 100 4º IzQZA.	0xFFD8FFE100BC45786966000049492A00008000000600120...	3FBC416C-676A-4C87-85CE-152010AEFE52
4	Santiago	Descalzo	De La Tome	13579135D	1996-06-06	698765432	Fake St. 123	0x89504E470D0A1A0A0000000D4948445200000200000002...	3269C736-1685-49A3-891F-0D8BEE87B750
5	Alberto	Castaña	Manzana	66502374R	1986-12-28	636963636	Avda. Pepe 7 2ºB	0x89504E470D0A1A0A0000000D494844520000020000002...	E32CE7F3-2B11-44FB-8D2C-831B6714FBC5
6	Alma	Marillo	Puertas	47445203M	1981-06-08	654254254	C/ Jonás 12 8º DCHA	0x89504E470D0A1A0A0000000D494844520000020000002...	208206B9-00A8-4786-A860-EEA21D7FCAB3

At the bottom, a message bar indicates "Query executed successfully."

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "Solution1 - Miscellaneous Files - data\_masking.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Project, Debug, SSMSBoost, Tools, Window, and Help. The toolbar contains various icons for database management tasks. The Object Explorer on the left shows the database structure for "SAL\_WS16\_SCOUTS (SQL Server 14.0.1000.169 - SAL)". The main pane displays a query window titled "data\_masking.sql - ...S\\$\\$AL\_SCOUTS (53)" containing the following T-SQL script:

```
286  
287 CREATE USER NoMask WITHOUT LOGIN;  
288 GO  
289  
290 REVOKE UNMASK TO Presidente;  
291 GO  
292  
293 GRANT UNMASK TO NoMask;  
294 GO
```

The status bar at the bottom indicates "161 %". The message bar at the bottom right says "Commands completed successfully."

SQLQuery1.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (JRM\_SCOUTS (55)) - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

Object Explorer SAL\_SCOUTS

SQLQuery1.sql - SA-(JRM\_SCOUTS (55))

```
EXECUTE AS USER = 'NoMask';
GO

PRINT USER
GO

-- NoMask

SELECT * FROM SAL_PERSONAL
GO
```

Results Messages

ID_personal	nombre	apellido1	apellido2	DNI	fecha_nacimiento	telefono	direccion	foto_perfil	ID_img
1	Juan	Robles	Martinez	87564321A	1967-06-18	601234567	C/ Perancho 14 3º Dcha	0xFF0F8FE100B457869660004954204000000000600120...	8BF545-E0D4-4320-B4D2-4E9D4154E5B
2	Señal	Abubrah	León	123456789	1997-03-08	600000000	Avda. Mcalle 11 RA	0xFF0F8FE0010446449460001010000100000000000000120...	28C0D30-9420-44C9-8875-500F4ADEC2
3	Lucía	García	Caamaño	24688640C	1973-09-02	612345678	C/ Estatua 100 4º IZQA.	0xFF0F8FE100B457869660004954204000000000600120...	3FB41C6-C76A-4C37-85C1-15201DA4EFF52
4	Santiago	Descalzo	De La Rosa	13579135D	1996-06-06	698765432	Fran. St. 123	0x89504E47000A10A0000000004948452000000000000000002...	329X736-165-494-381F-008DE6E87
5	Alberto	Castilla	Manzana	66502374R	1986-12-28	639636356	Avda. Pepe 7 2ºB	0x89504E47000A10A0000000004948445200000000000000002...	E32CE7F-73-11-44F-802C-831B6714FC5
6	Alma	Martillo	Puerta	47445203M	1981-06-08	654254254	C/ Jordas 12 3º DCHA	0x89504E47000A10A0000000004948445200000000000000002...	20820689-00A-47B6-A860-EEA212F7CA83

Query executed successfully.

File Edit View Help

CONNECTIONS

- ✓ SERVERS 192.168.1.100, <default> (LGM\_SCOUTS)
- ✓ Databases
- System Databases
- AdventureWorks2017
- Northwind
- pubs
- ✓ SAL\_SCOUTS
- Tables
- dbo.SAL\_ENTIDAD
- dbo.SAL\_EVENTOS
- dbo.SAL\_PROYECTO
- Views
- Synonyms
- Programmability
- External Resources
- Service Broker
- Storage
- Security
- SCOUTS\_PRACTICE
- WideWorldImporters
- Security
- Server Objects

SQLQuery\_1 - 192.168.1.100.master (LGM\_SCOUTS) - Azure Data Studio

Run Cancel Disconnect Change Connection SAL\_SCOUTS Explain Enable SQLCMD Export as Notebook

```
17
18 PRINT USER
19 GO
20 -- NoMask
21
22
23 SELECT * FROM SCOUTS.CASTORES;
24 SELECT * FROM SCOUTS.LOBATOS;
25 SELECT * FROM SCOUTS.EXPLORADORES;
26 SELECT * FROM SCOUTS.PIONEROS;
27 SELECT * FROM SCOUTS.RUTAS;
28 GO
```

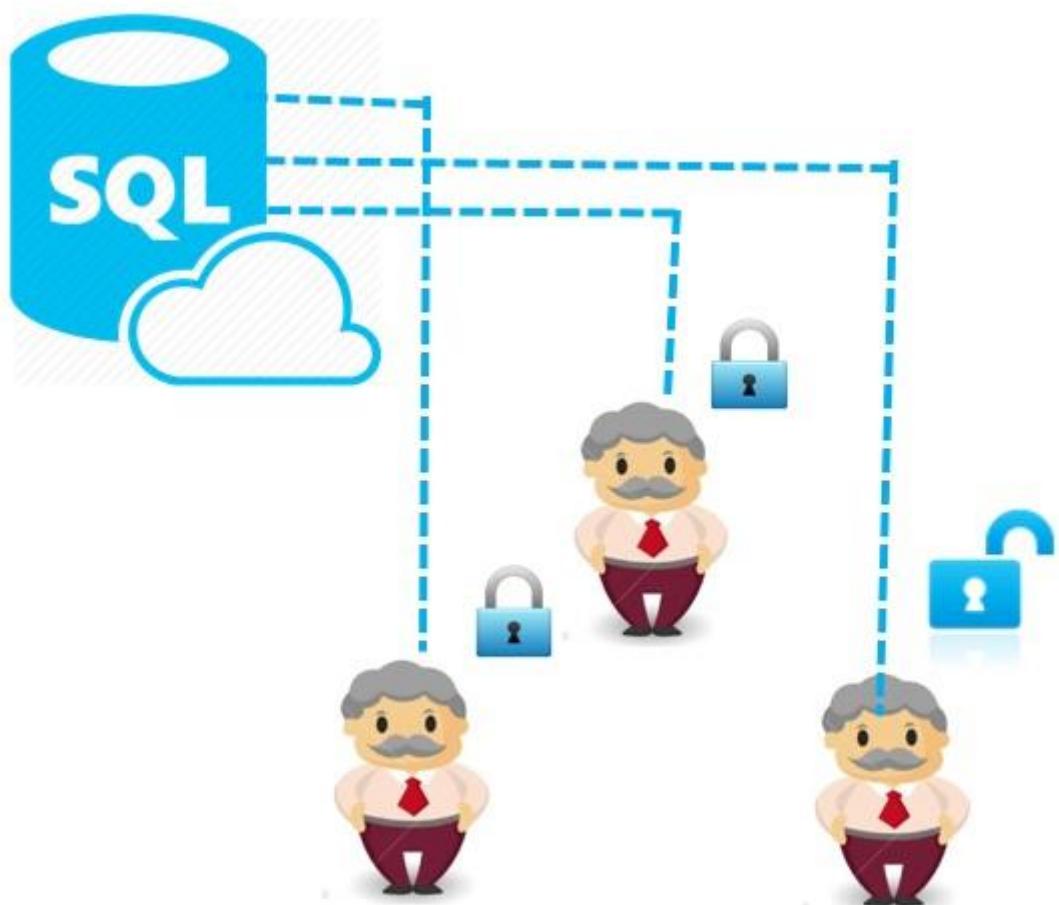
nombre	apellido1	apellido2	edad	foto_perfil	num_insignias
Stone	Mccoy	Hall	6	0xFFDBFFEE00B104A464946...	8
Hadassah	Snow	Mckee	8	0xFFDBFFEE00B104A464946...	7
Nayda	Nicholson	Ortega	6	0xFFDBFFEE00B104A464946...	4
Branden	Schultz	Jimenez	8	0xFFDBFFEE00B104A464946...	5
Austin	Nielsen	Gonzalez	7	0xFFDBFFEE00B104A464946...	8
Finn	Miranda	Eaton	7	0xFFDBFFEE00B104A464946...	4
Fuller	Ochoa	Riddle	7	0xFFDBFFEE00B104A464946...	4
Kermit	Garrett	Holloway	8	0xFFDBFFEE00B104A464946...	6
Paul	Barnett	Holland	6	0xFFDBFFEE00B104A464946...	8
Shellie	Manning	Keith	8	0xFFDBFFEE00B104A464946...	5

nombre	apellido1	apellido2	edad	foto_perfil	num_insignias
Hopie	Mckay	Robertson	10	0xFFDBFFEE00B104A464946...	3
Dalton	Roy	Dudley	9	0xFFDBFFEE00B104A464946...	2
Tana	Mckee	Whitehead	11	0xFFDBFFEE00B104A464946...	6
Malik	Avery	Rowlond	9	0xFFDBFFEE00B104A464946...	3
Thomas	Alston	Richmond	11	0xFFDBFFEE00B104A464946...	6

Y esto sería un ejemplo sobre el enmascaramiento dinámico de datos.

## • 2.5.2. Row Encryption (RLS, Row-Level Security)

La Seguridad a Nivel de Fila (**RLS, Row-Level Security**) es un mecanismo o característica de SQL Server que se utiliza para controlar/restringir la información de las tablas basándose en las autorizaciones de acceso al usuario que se encuentra actualmente en línea (*logged in*). Lo que significa que los datos almacenados de las tablas se podrán visualizar para aquellos usuarios que han aportado información a tales tablas. Básicamente se utiliza para que los usuarios visualicen la información que ellos mismos aportaron sin tener acceso al resto de datos de los demás dotando de privacidad en las inserciones.



La lógica de la restricción de acceso está ubicada en el nivel de base de datos en lugar de estar alejado de los datos en otro nivel de aplicación. El sistema de base de datos aplica las restricciones de acceso cada vez que se intenta acceder a los datos desde cualquier nivel. Esto hace que el sistema de seguridad resulte más sólido y confiable al reducir el área expuesta del sistema de seguridad.

El acceso a los datos de nivel de fila de una tabla está restringido por un predicado de seguridad que se define como una función con valores de tabla insertada. Luego, la función se invoca y una directiva de seguridad la aplica. Los predicados de filtro, la aplicación es consciente de las filas filtradas del conjunto de resultados. Si se filtran todas las filas, se devolverá un conjunto nulo. En el caso de los predicados de bloqueo, las operaciones que infrinjan el predicado generarán un error.

Los predicados de filtro se aplican al leer los datos desde la tabla base y afectan a todas las operaciones ***SELECT***, ***DELETE*** y ***UPDATE***. Los usuarios no se pueden seleccionar o eliminar las filas filtradas. El usuario no puede actualizar las filas filtradas. Pero, es posible actualizar las filas de tal manera que se filtren después. Los predicados de bloqueo afectan a todas las operaciones de escritura.

Los predicados de filtro y de bloqueo y las directivas de seguridad tienen el siguiente comportamiento:

- 1)** Puede definir una función de predicado que se combine con otra tabla o invoque una función. Si la directiva de seguridad se crea con ***SCHEMABINDING = ON***, entonces se puede acceder a la función o combinación desde la consulta, y funciona como se espera sin comprobaciones de permisos adicionales. Si la directiva de seguridad se crea con ***SCHEMABINDING = OFF***, entonces los usuarios necesitarán los permisos ***SELECT*** en estas funciones y tablas adicionales para consultar la tabla de destino.
- 2)** Puede emitir una consulta a una tabla que tenga un predicado de seguridad definido pero deshabilitado. Todas las filas que se han filtrado o bloqueado no se ven afectadas.
- 3)** Si el usuario ***dbo***, un miembro del rol ***db\_owner*** o el propietario de la tabla consulta una tabla que tiene una directiva de seguridad definida y habilitada, **las filas se filtran o bloquean** según indique la directiva de seguridad.
- 4)** Los intentos de modificar el *schema* de una tabla enlazada por una directiva de seguridad enlazada a un *schema* producirán un error. Sin embargo, se pueden modificar las columnas a las que el predicado no hace referencia.
- 5)** Los intentos de agregar un predicado a una tabla que ya tiene uno definido para la operación especificada producen un error. Esto ocurrirá tanto si el predicado está habilitado como si no.

**6)** Los intentos de modificar una función, que se usa como predicado en una tabla dentro de una directiva de seguridad enlazada a un esquema, producen un error.

**7)** Definir varias directivas de seguridad activar que contienen predicados no superpuestos, será correcto.

En resumen, los predicados de filtro **filtran** en modo silencioso las filas disponibles para leer operaciones con **SELECT**, **UPDATE** y **DELETE**.

Los predicados de bloqueo tienen el siguiente comportamiento:

**1)** Los predicados de bloqueo para **UPDATE** se dividen en operaciones independientes para **BEFORE** y **AFTER**. En consecuencia, no puede, por ejemplo, bloquear a los usuarios para que no actualicen una fila con un valor mayor que el actual. Si se requiere este tipo de lógica, debemos usar las tablas temporales **deleted** e **inserted** para hacer referencia a los valores antiguos y nuevos juntos.

**2)** El optimizador no comprobará un predicado de bloqueo **AFTER UPDATE** si no se ha cambiado ninguna de las columnas usadas por la función de predicado.

En resumen, los predicados de bloqueo **bloquean** explícitamente las operaciones de escritura (**AFTER INSERT**, **AFTER UPDATE**, **BEFORE UPDATE**, **BEFORE DELETE**) que infringen el predicado.

Dicho esto, como ejemplo de este apartado se dará la siguiente situación: han llegado 3 nuevos miembros al personal de la asociación:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_SCOUTS' is selected. In the center pane, a query window titled 'rls.sql - SAL\_WS16...\$SAL\_SCOUTS (52)' contains the following T-SQL code:

```
1 USE SAL_SCOUTS
2 GO
3
4 SELECT * FROM SAL_PERSONAL WHERE ID_personal >= 7;
5 INSERT INTO SAL_TESORERO VALUES (7), (8), (9);
6 SELECT * FROM SAL_TESORERO;
7 GO
```

The 'Results' tab displays two tables: 'ID\_personal' and 'ID\_tesorero'. The 'ID\_personal' table has three rows with IDs 7, 8, and 9. The 'ID\_tesorero' table has four rows with IDs 5, 7, 8, and 9. Below the results, a message says 'Query executed successfully.'

Estos 3 nuevos miembros (**Marta**, **David** y **Sheila**) formarán parte del departamento de Tesorería junto con **Alberto (ACM\_SCOUTS)**.

Si hacemos una consulta a la tabla **SAL\_DONACION** veremos los diferentes IDs de los tesoreros que realizaron inserciones:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_WS16\_SCOUTS' is selected. In the center pane, a query window displays the following SQL code:

```

3  SELECT * FROM SAL_TESORERO;
4  GO

```

The results grid shows 16 rows of data from the 'SAL\_TESORERO' table. The columns are:

	ID_donacion	fecha_donacion	nombre_donante	importe	SAL_EVENTOS_ID_evento	SAL_TESORERO_ID_personal
1	1	2017-08-19	Emalee Overington	2448,48	3	8
2	2	2017-04-06	Tyson Brook	4260,17	1	5
3	3	2017-05-31	Halli Selbach	4498,48	5	5
4	4	2017-02-24	Heinberto Gibson	5178,37	3	8
5	5	2017-08-10	Bidget O'Kearan	1263,37	3	9
6	6	2017-05-10	Nichole Grindall	4982,77	3	5
7	7	2018-01-06	Susy Janusz	3627,80	5	9
8	8	2018-01-19	Kelby Springle	5918,28	5	9
9	9	2017-05-31	Winni Kezman	6262,68	4	5
10	10	2017-10-31	Tobey Sterkhouse	5197,99	3	9
11	11	2017-10-10	Blisse Hart	5596,73	1	5
12	12	2017-07-11	Benetta Skepper	2226,02	3	9
13	13	2017-09-12	Randee Giusto	9920,08	5	5
14	14	2017-03-13	Vikky Kaveney	9265,35	5	7
15	15	2017-10-25	Truda Bussen	4117,95	3	9
16	16	2017-06-23	Anaiese Maynell	7610,10	2	9

At the bottom of the results grid, it says 'Query executed successfully.' and 'SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\|SAL\_SCOUTS... | SAL\_SCOUTS | 00:00:00 | 1000 rows'.

Lo importante de esto es que no queremos que se enteren cuántas inserciones llegaron a hacer cada uno, así que, vamos a crear una función con una directiva que evite esto mismo, pero antes, para disponer de una mejor identificación de usuarios de nuestra base de datos, vamos a modificar la tabla **SAL\_PERSONAL** añadiéndole una columna llamada **scout\_user** que mostrará los nombres de usuario del personal de la asociación:

```

ALTER TABLE SAL_PERSONAL
ADD scout_user VARCHAR(50);
GO

UPDATE SAL_PERSONAL SET scout_user = 'JRM_SCOUTS' WHERE ID_personal = 1;
UPDATE SAL_PERSONAL SET scout_user = 'SAL_SCOUTS' WHERE ID_personal = 2;
UPDATE SAL_PERSONAL SET scout_user = 'LGM_SCOUTS' WHERE ID_personal = 3;
UPDATE SAL_PERSONAL SET scout_user = 'SDDT_SCOUTS' WHERE ID_personal = 4;
UPDATE SAL_PERSONAL SET scout_user = 'ACM_SCOUTS' WHERE ID_personal = 5;
UPDATE SAL_PERSONAL SET scout_user = 'AMP_SCOUTS' WHERE ID_personal = 6;
UPDATE SAL_PERSONAL SET scout_user = 'MMO_SCOUTS' WHERE ID_personal = 7;
UPDATE SAL_PERSONAL SET scout_user = 'DGC_SCOUTS' WHERE ID_personal = 8;
UPDATE SAL_PERSONAL SET scout_user = 'SHAL_SCOUTS' WHERE ID_personal = 9;
GO
-- (9 rows affected)

SELECT ID_personal,scout_user FROM SAL_PERSONAL
GO

```

```
-- ID_personal      scout_user
-- 1                JRM_SCOUTS
-- 2                SAL_SCOUTS
-- 3                LGM_SCOUTS
-- 4                SDDT_SCOUTS
-- 5                ACM_SCOUTS
-- 6                AMP_SCOUTS
-- 7                MMO_SCOUTS
-- 8                DGC_SCOUTS
-- 9                SHAL_SCOUTS
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_WS16\_SCOUTS' is selected. In the center pane, a query window titled 'rls.sql - SAL\_WS16..\$SAL\_SCOUTS (54)' contains the following code:

```
16 UPDATE SAL_PERSONAL SET scout_user = 'SHAL_SCOUTS' WHERE ID_personal = 9;
17 GO
18 -- (9 rows affected)

20 SELECT ID_personal,scout_user FROM SAL_PERSONAL
21 GO
22
23 -- ID_personal  scout_user
24 -- 1            JRM_SCOUTS
25 -- 2            SAL_SCOUTS
```

The 'Results' tab shows the output of the last SELECT statement:

ID_personal	scout_user
1	JRM_SCOUTS
2	SAL_SCOUTS
3	LGM_SCOUTS
4	SDDT_SCOUTS
5	ACM_SCOUTS
6	AMP_SCOUTS
7	MMO_SCOUTS
8	DGC_SCOUTS
9	SHAL_SCOUTS

At the bottom of the screen, a yellow bar indicates: 'Query executed successfully.' and 'SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\sal\_scouts... | SAL\_SCOUTS | 00:00:00 | 9 rows'.

Después creamos una vista especial que incluya los IDs de los tesoreros que a su vez hagan relación con los nombres de usuario, esta sería la consulta:

```
SELECT d.* , p.scout_user AS scout_user
  FROM SAL_DONACION d
 JOIN SAL_TESORERO ON d.SAL_TESORERO_ID_personal = SAL_TESORERO.ID_tesorero
 JOIN SAL_PERSONAL p ON SAL_TESORERO.ID_tesorero = p.ID_personal;
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_WS16\_SCOUTS' is selected. In the center pane, a query window titled 'Query1.sql - S...\$SAL\_SCOUTS (54)' contains the following code:

```
27 -- 4           SDDT_SCOUTS
28 -- 5           ACM_SCOUTS
29 -- 6           AMP_SCOUTS
30 -- 7           MMO_SCOUTS
31 -- 8           DGC_SCOUTS
32 -- 9           SHAL_SCOUTS
33
34 SELECT d.* , p.scout_user AS scout_user
  FROM SAL_DONACION d
 JOIN SAL_TESORERO ON d.SAL_TESORERO_ID_personal = SAL_TESORERO.ID_tesorero
 JOIN SAL_PERSONAL p ON SAL_TESORERO.ID_tesorero = p.ID_personal;
35
36 GO
```

The 'Results' tab shows the output of the last SELECT statement:

ID_donacion	fecha_donacion	nombre_donante	importe	SAL_EVENTOS_ID_evento	SAL_TESORERO_ID_personal	scout_user
1	2017-08-19	Emilee Oveington	2448.49	3	8	DGC_SCOUTS
2	2017-04-06	Tyson Brook	4280.17	1	5	ACM_SCOUTS
3	2017-05-31	Hall Selbach	4498.48	5	5	ACM_SCOUTS
4	2017-02-24	Heriberto Gibson	5178.37	3	8	DGC_SCOUTS
5	2017-08-10	Bidget O'Kearan	1283.37	3	9	SHAL_SCOUTS
6	2017-05-10	Nichols Grindall	4882.77	3	5	ACM_SCOUTS
7	2018-01-06	Suzi Janusz	3627.80	5	9	SHAL_SCOUTS
8	2018-01-19	Katia Schenck	5918.28	5	9	SHAL_SCOUTS

At the bottom of the screen, a yellow bar indicates: 'Query executed successfully.' and 'SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\sal\_scouts... | SAL\_SCOUTS | 00:00:00 | 1000 rows'.

Teniendo en cuenta que ahora los tesoreros no podrán usar la tabla **dbo.SAL\_DONACION** para insertar información, les retiramos los permisos que tenían con la tabla anterior y les crearemos una nueva vista llamada **TREASURER.DONATION**, de esta manera las inserciones son privadas y **dbo** puede ver todo. Ejecutamos las siguientes sentencias:

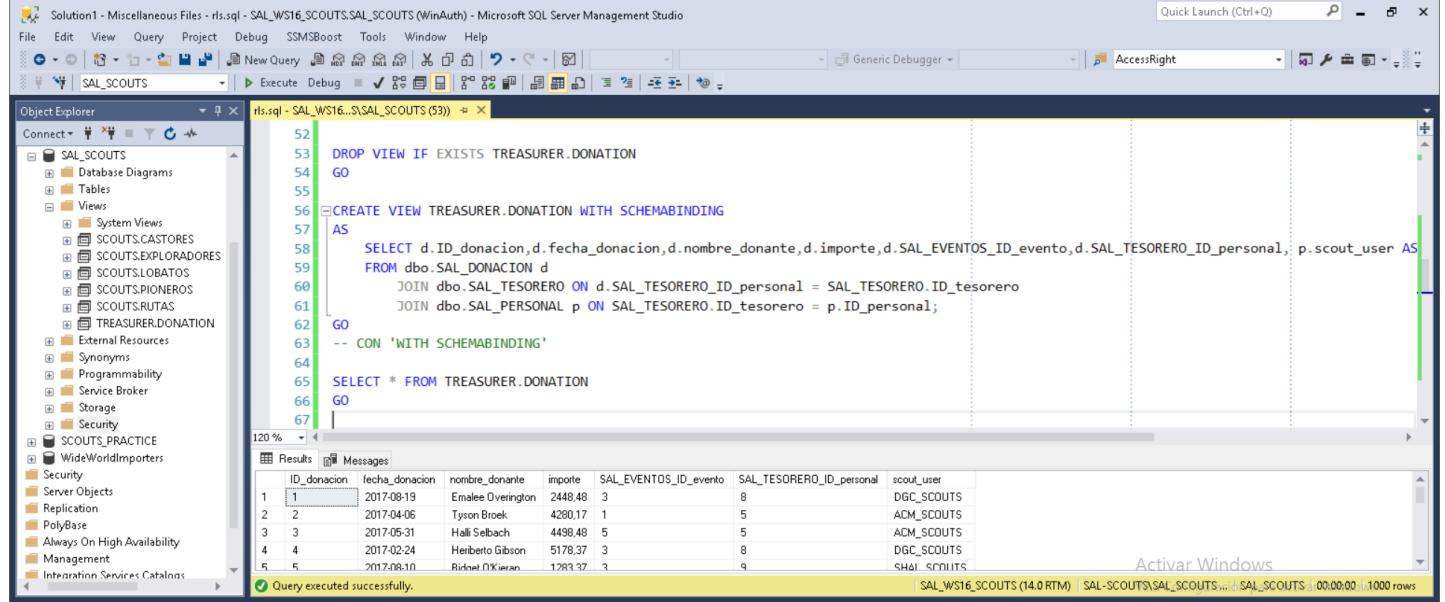
```
REVOKE SELECT,INSERT,UPDATE,DELETE ON SAL_DONACION TO Tesorero;
GRANT SELECT,INSERT,UPDATE,DELETE ON TREASURER.DONATION TO Tesorero;
--GRANT IMPERSONATE ON USER::[NoMask] TO [MMO_SCOUTS];
--GRANT IMPERSONATE ON USER::[NoMask] TO [DGC_SCOUTS];
--GRANT IMPERSONATE ON USER::[NoMask] TO [SHAL_SCOUTS];
GRANT UNMASK TO Tesorero; -- Para que los valores no se encuentren enmascarados
GO

DROP SCHEMA IF EXISTS TREASURER
GO

CREATE SCHEMA TREASURER
GO

DROP VIEW IF EXISTS TREASURER.DONATION
GO

CREATE VIEW TREASURER.DONATION WITH SCHEMABINDING
AS
    SELECT
        d.ID_donacion, d.fecha_donacion, d.nombre_donante, d.importe, d.SAL_EVENTOS_ID_evento
        , d.SAL_TESORERO_ID_personal, p.scout_user AS scout_user
    FROM dbo.SAL_DONACION d
    JOIN dbo.SAL_TESORERO ON d.SAL_TESORERO_ID_personal = SAL_TESORERO.ID_tesorero
    JOIN dbo.SAL_PERSONAL p ON SAL_TESORERO.ID_tesorero = p.ID_personal;
GO
-- CON 'WITH SCHEMABINDING'
```



Y por último queda crear la función que trabajará como predicado de filtro para que filtre el campo **scout\_user** según el usuario que se encuentre conectado, ejecutamos las siguientes sentencias:

```
-- Controlamos la existencia de la función
DROP FUNCTION IF EXISTS TREASURER.SALfnDON
GO

CREATE OR ALTER FUNCTION TREASURER.SALfnDON (@scout_user SYSNAME)
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN SELECT 1 AS treasurer_filter
WHERE @scout_user = USER_NAME()
OR (USER_NAME() IN ('dbo','dbo')); -- EN PRINCIPIO DEBERÍA SER
'ACM_SCOUTS', no otra vez 'dbo' | ¿ES UN BUG?
-- SIN ESTE APARTADO, dbo no PODRÍA VISUALIZAR NINGÚN CAMPO DE LA VISTA
TREASURER.DONATION
GO

-- Controlamos la existencia de la directiva de seguridad
DROP SECURITY POLICY IF EXISTS TREASURER.SAL_SPOscouts_don
GO

CREATE SECURITY POLICY TREASURER.SAL_SPOscouts_don
ADD FILTER PREDICATE TREASURER.SALfnDON(scout_user)
ON TREASURER.DONATION
WITH (STATE = ON);
GO
```

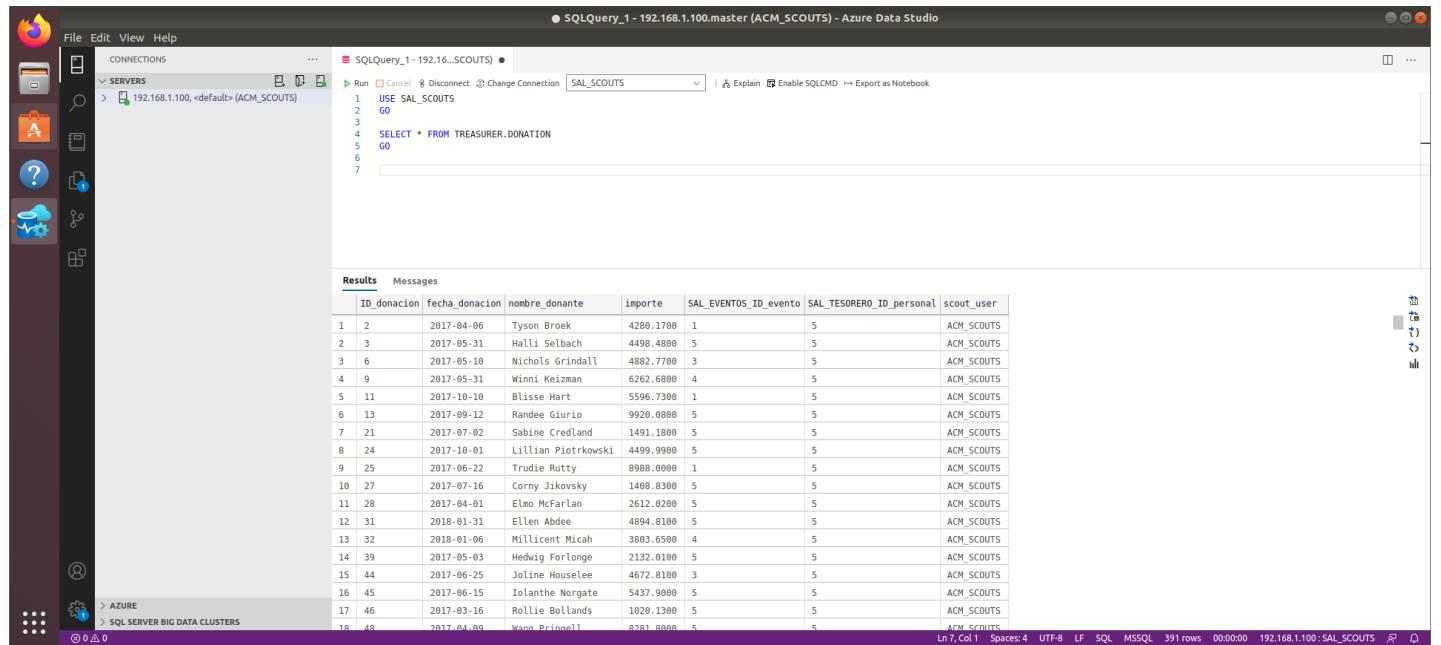
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left shows the database structure, including the 'SAL\_SCOUTS' database and its objects like Programmability, Security, and Tables. The central pane displays a query window with the following SQL code:

```
-- SIN ESTE APARTADO, dbo no PODRÍA VISUALIZAR NINGÚN CAMPO DE LA VISTA TREASURER.DONATION
79 GO
80
81 -- Controlamos la existencia de la directiva de seguridad
82 DROP SECURITY POLICY IF EXISTS TREASURER.SAL_SPOscouts_don
83 GO
84
85
86 CREATE SECURITY POLICY TREASURER.SAL_SPOscouts_don
87 ADD FILTER PREDICATE TREASURER.SALfnDON(scout_user)
88 ON TREASURER.DONATION
89 WITH (STATE = ON);
90 GO
91
92 SELECT * FROM TREASURER.DONATION
93 GO
```

The status bar at the bottom of the SSMS window indicates "Query executed successfully.".

Con la función creada, solamente nos queda comprobarlo desde el equipo cliente para ver si se ha ejecutado y funciona correctamente:

~ ACM\_SCOUTS:

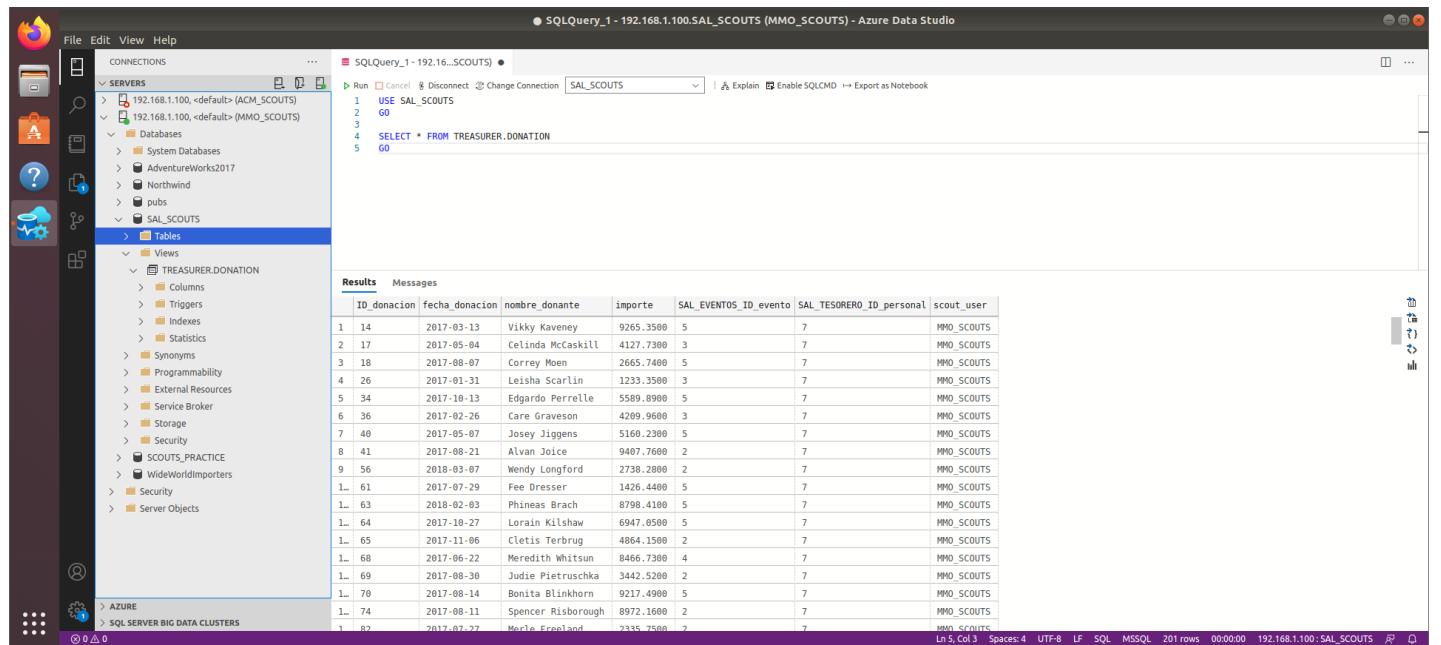


The screenshot shows the Azure Data Studio interface with a connection to SQLQuery\_1 - 192.168.1.100.master (ACM\_SCOUTS). The query window contains the following code:

```
USE SAL_SCOUTS
GO
SELECT * FROM TREASURER.DONATION
GO
```

The results pane displays a table with 391 rows of data from the TREASURER.DONATION table. The columns are: ID\_donacion, fecha\_donacion, nombre\_donante, importe, SAL\_EVENTOS\_ID\_evento, SAL\_TESORERO\_ID\_personal, and scout\_user. The data includes various names like Tyson Brook, Halli Selbach, Nichols Grindall, Winni Keizman, Blisse Hart, Randee Giurio, Sabine Credland, Lillian Piotrkowski, Trudie Ratty, Corny Jikovsky, Elmo McFarlan, Ellen Abdee, Millicent Mical, Hedwig Fortinge, Joline Houselee, Iolanthe Norgate, Rollie Bollands, Wang Prinnell, and many others. The scout\_user column consistently shows 'ACM\_SCOUTS'.

~ MMO\_SCOUTS:

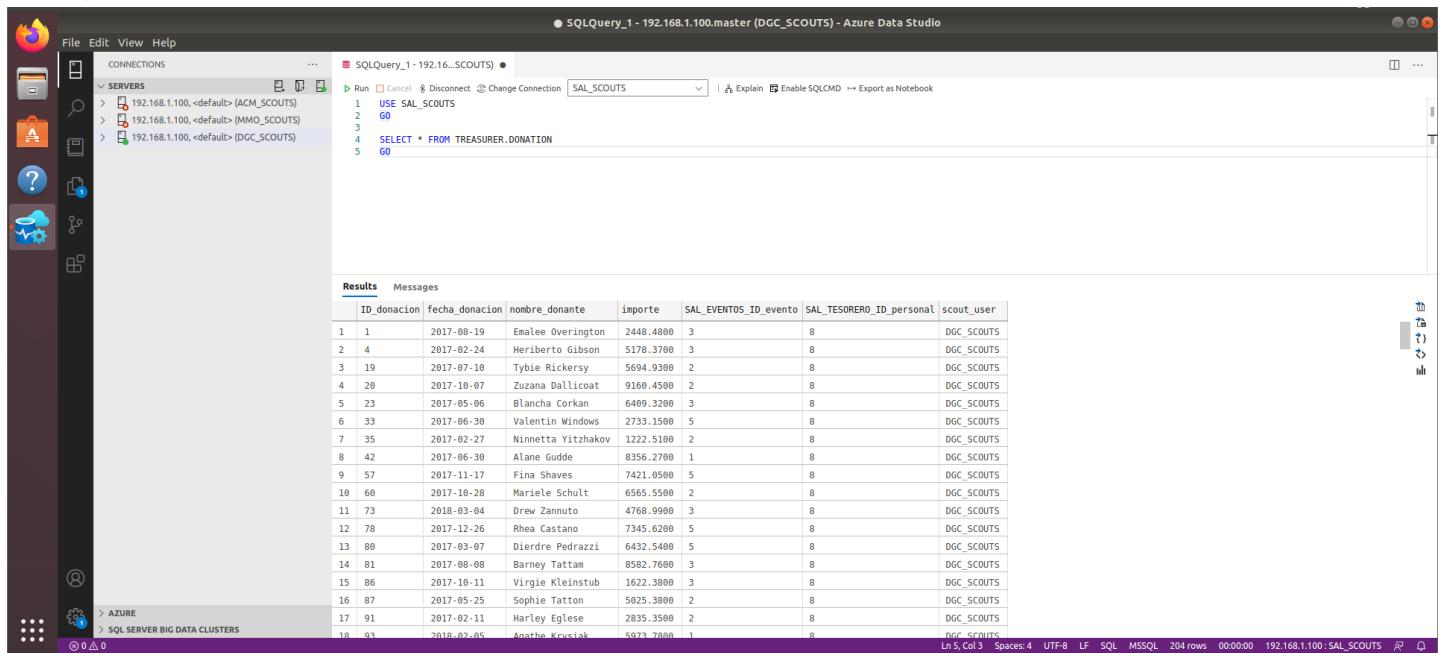


The screenshot shows the Azure Data Studio interface with a connection to SQLQuery\_1 - 192.168.1.100.SAL\_SCOUTS (MMO\_SCOUTS). The query window contains the same code as the previous screenshot:

```
USE SAL_SCOUTS
GO
SELECT * FROM TREASURER.DONATION
GO
```

The results pane displays a table with 201 rows of data from the TREASURER.DONATION table. The columns are: ID\_donacion, fecha\_donacion, nombre\_donante, importe, SAL\_EVENTOS\_ID\_evento, SAL\_TESORERO\_ID\_personal, and scout\_user. The data includes names like Vikky Kaveney, Celinda McCaskill, Correy Moen, Leisha Scarlin, Edgardo Perrelle, Care Graveson, Josey Jiggens, Alvan Joice, Wendy Longford, Fee Dresser, Phineas Brach, Lorain Kilshaw, Cletis Terbrug, Meredith Whitsut, Judie Pietruschka, Bonita Blinkhorn, Spencer Risborough, and Marle Freeland. The scout\_user column consistently shows 'MMO\_SCOUTS'.

## ~ DGC\_SCOUTS:



File Edit View Help

CONNECTIONS

Servers

SQLQuery\_1 - 192.168.1.100.master (DGC\_SCOUTS) - Azure Data Studio

Run Cancel Disconnect Change Connection SAL\_SCOUTS Explain Enable SQLCMD Export as Notebook

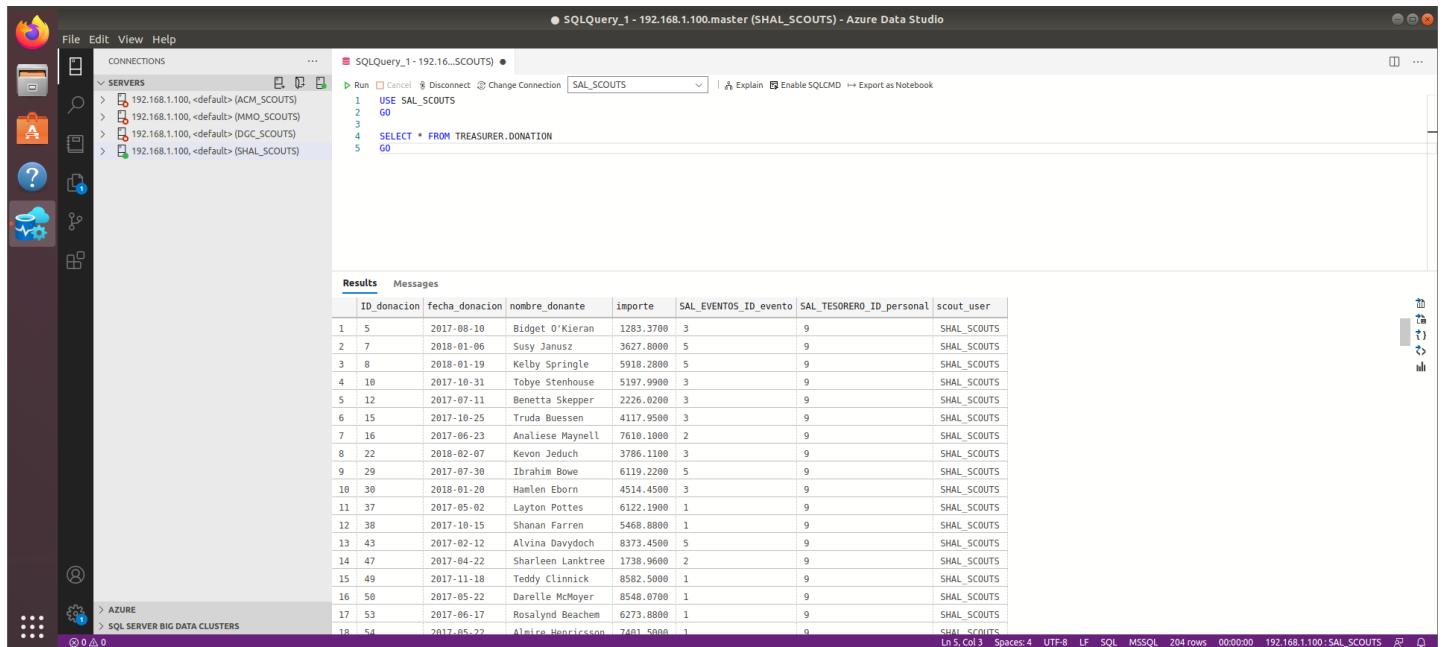
```
1 USE SAL_SCOUTS
2 GO
3
4 SELECT * FROM TREASURER.DONATION
5 GO
```

Results Messages

ID_donacion	fecha_donacion	nombre_donante	importe	SAL_EVENTOS_ID_evento	SAL_TESORERO_ID_personal	scout_user
1	2017-08-19	Emalee Overington	2448.4800	3	8	DGC_SCOUTS
2	2017-02-24	Heriberto Gibson	5178.3700	3	8	DGC_SCOUTS
3	19	2017-07-10	Tybie Rickersy	5694.9300	2	DGC_SCOUTS
4	20	2017-10-07	Zuzana Dallcoat	9160.4500	2	DGC_SCOUTS
5	23	2017-05-06	Blancha Corkan	6409.3200	3	DGC_SCOUTS
6	33	2017-06-30	Valentian Windows	2733.1500	5	DGC_SCOUTS
7	35	2017-02-27	Ninetta Yitzhakov	1222.5100	2	DGC_SCOUTS
8	42	2017-06-30	Alana Gudde	8356.2700	1	DGC_SCOUTS
9	57	2017-11-17	Fina Shaves	7421.8500	5	DGC_SCOUTS
10	60	2017-10-28	Marielle Schult	6565.5500	2	DGC_SCOUTS
11	73	2018-03-04	Drew Zannuto	4768.9900	3	DGC_SCOUTS
12	78	2017-12-26	Rhea Castano	7345.6200	5	DGC_SCOUTS
13	80	2017-03-07	Dierdre Pedraza	6432.5400	5	DGC_SCOUTS
14	81	2017-08-08	Barney Tattam	8582.7600	3	DGC_SCOUTS
15	86	2017-10-11	Virgie Kleinstub	1622.3800	3	DGC_SCOUTS
16	87	2017-05-25	Sophie Tatton	5025.3800	2	DGC_SCOUTS
17	91	2017-02-11	Harley Egelse	2835.3500	2	DGC_SCOUTS
18	93	2018-02-05	Anathe Krystiak	5073.7800	1	DGC_SCOUTS

Ln 5, Col 3 Spaces: 4 UTF-8 LF SQL MSSQL 204 rows 00:00:00 192.168.1.100 : SAL\_SCOUTS

## ~ SHAL\_SCOUTS:



File Edit View Help

CONNECTIONS

Servers

SQLQuery\_1 - 192.168.1.100.master (SHAL\_SCOUTS) - Azure Data Studio

Run Cancel Disconnect Change Connection SAL\_SCOUTS Explain Enable SQLCMD Export as Notebook

```
1 USE SAL_SCOUTS
2 GO
3
4 SELECT * FROM TREASURER.DONATION
5 GO
```

Results Messages

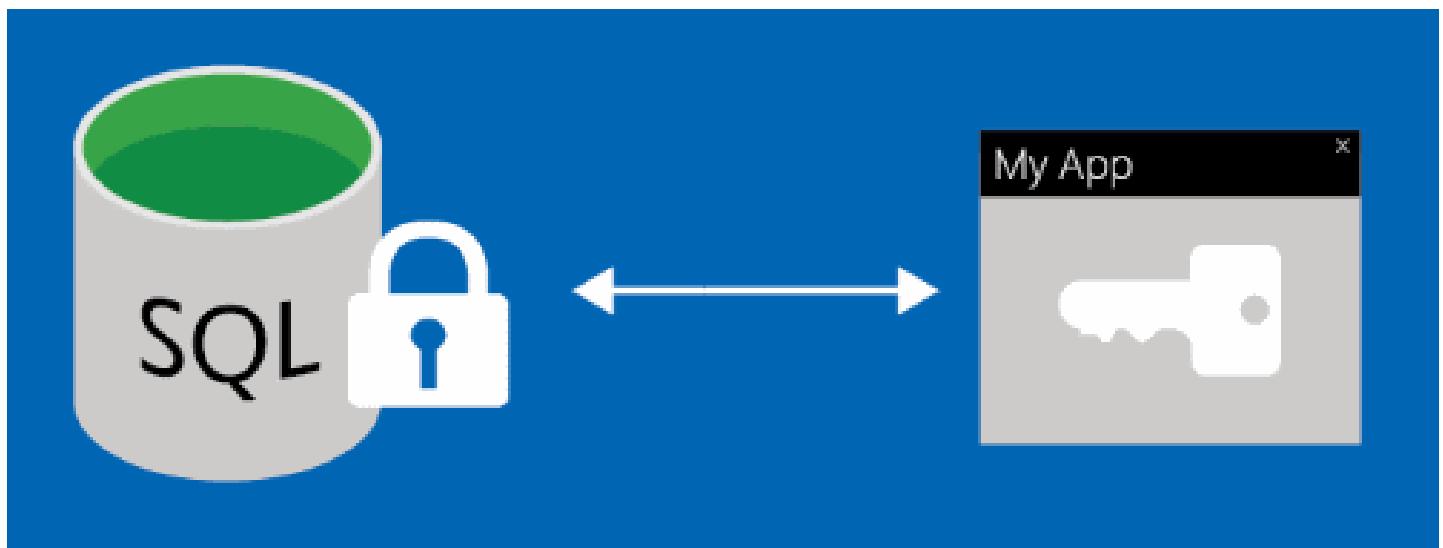
ID_donacion	fecha_donacion	nombre_donante	importe	SAL_EVENTOS_ID_evento	SAL_TESORERO_ID_personal	scout_user
1	5	2017-08-10	Bidget O'Kieran	1283.3700	3	SHAL_SCOUTS
2	7	2018-01-06	Susy Janusz	3627.8000	5	SHAL_SCOUTS
3	8	2018-01-19	Kelby Springle	5918.2800	5	SHAL_SCOUTS
4	10	2017-10-31	Tobye Stenhouse	5197.9900	3	SHAL_SCOUTS
5	12	2017-07-11	Benette Skepper	2226.0200	3	SHAL_SCOUTS
6	15	2017-10-25	Truda Buessen	4117.9500	3	SHAL_SCOUTS
7	16	2017-06-23	Analiese Maynell	7610.1000	2	SHAL_SCOUTS
8	22	2018-02-07	Kevon Jeduch	3786.1100	3	SHAL_SCOUTS
9	29	2017-07-30	Ibrahim Bowe	6119.2200	5	SHAL_SCOUTS
10	30	2018-01-20	Hamlen Eborn	4514.4500	3	SHAL_SCOUTS
11	37	2017-05-02	Layton Pottes	6122.1900	1	SHAL_SCOUTS
12	38	2017-10-15	Shanan Farren	5468.8800	1	SHAL_SCOUTS
13	43	2017-02-12	Alvina Davydoch	8373.4500	5	SHAL_SCOUTS
14	47	2017-04-22	Sharleen Lanktree	1738.9600	2	SHAL_SCOUTS
15	49	2017-11-18	Teddy Clinnick	8582.5000	1	SHAL_SCOUTS
16	50	2017-05-22	Darelle McMoyer	8548.0700	1	SHAL_SCOUTS
17	53	2017-06-17	Rosalyn Beachem	6273.8800	1	SHAL_SCOUTS
18	54	2017-05-22	Almire Henrirsson	7401.5000	1	SHAL_SCOUTS

Ln 5, Col 3 Spaces: 4 UTF-8 LF SQL MSSQL 204 rows 00:00:00 192.168.1.100 : SAL\_SCOUTS

## • 2.6. Always Encrypted

**Always Encrypted** (Siempre encriptado) se trata de una característica diseñada para proteger la información confidencial, como números de tarjeta de crédito o números de identificación nacional. Permite a los clientes cifrar información confidencial en aplicaciones cliente y nunca revelar las claves de cifrado en Motor de base de datos. Como resultado, **Always Encrypted** proporciona una separación entre aquellos que poseen los datos y pueden verlos, y aquellos que los administran, pero que no deberían tener acceso. Al asegurar que los administradores de base de datos local, los operadores de base de datos en la nube y otros con usuarios con privilegios elevados no autorizados no pueden obtener acceso a los datos cifrados, **Always Encrypted** permite a los clientes almacenar información confidencial de forma segura fuera de su control directo. Esto permite a las organizaciones almacenar sus datos en otras instancias como Azure, y permitir la delegación de la administración de la base de datos local a terceros, o reducir los requisitos de autorización de seguridad para su propio personal de administración de bases de datos.

**Always Encrypted** proporciona capacidades de computación confidencial al permitir que Motor de base de datos procese algunas consultas en datos cifrados y, al mismo tiempo, preserva la confidencialidad de los datos y brinda las ventajas de seguridad mencionadas anteriormente.



Realiza cifrado transparente en las aplicaciones. Un controlador habilitado para **Always Encrypted** instalado en el equipo cliente consigue esto al cifrar y descifrar automáticamente la información confidencial en la aplicación cliente, cosa que esto no nos influye en absoluto ni tendrá que ver en el proyecto. El controlador cifra los datos en columnas confidenciales antes de pasar los datos a Motor de base de datos y vuelve a escribir las consultas automáticamente para que conserve la semántica de la aplicación. De forma similar, el controlador descifra los datos de forma transparente, almacenados en columnas de bases de datos cifradas, incluidas en los resultados de la consulta.

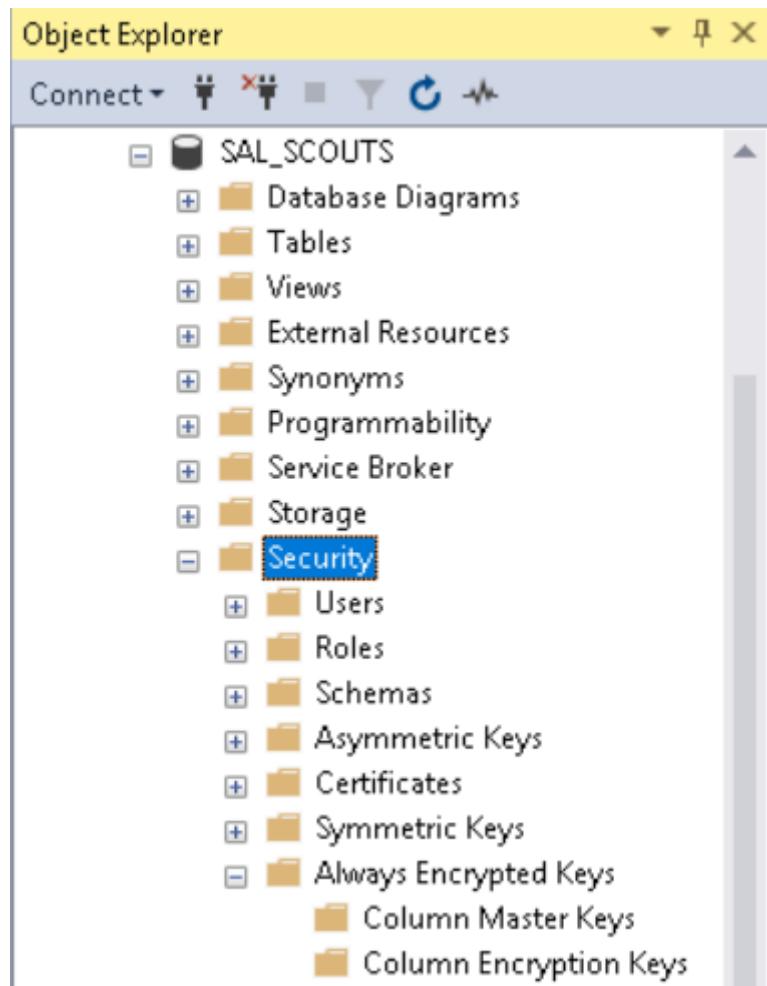
Para la configuración de esta característica se puede configurar de modo que las columnas individuales de la base de datos contengan la información confidencial. Al configurar el cifrado de una columna, debemos especificar la información sobre el algoritmo de cifrado y las claves criptográficas usadas para proteger los datos de la columna:

- Una clave de cifrado de columna (**CEK, Column Encryption Key**) se usa para cifrar los datos de una columna cifrada, siempre está situada en el lado del servidor de base de datos. Sin embargo, si alguien en por el lado de base de datos tiene acceso al **CEK**, puede descifrar los datos.
- Una clave maestra de columna (**CMK, Column Master Key**) es una clave de protección de claves que cifra una o varias claves de cifrado de columna, está situada en el lado del cliente o en el almacenamiento de aplicaciones de terceros. **CMK** es utilizada para proteger a la **CEK**, añadiendo una capa de seguridad adicional. Cualquier persona que tenga acceso al **CMK** puede actualmente desencriptar el **CEK** el cual puede ser usado para descifrar los datos actuales.

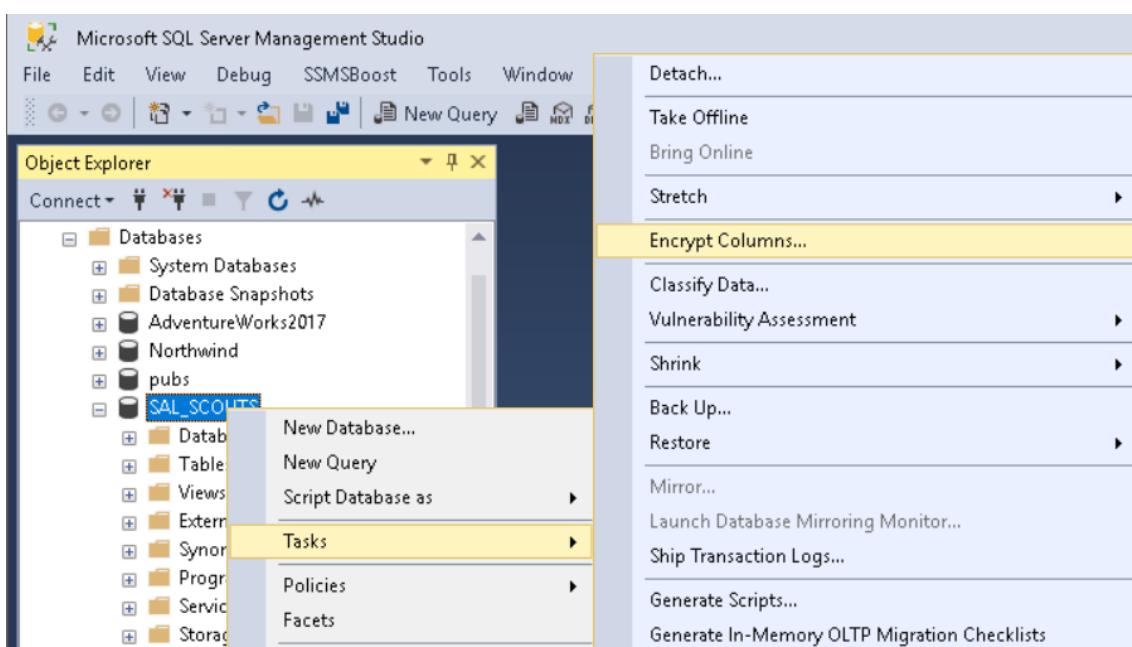
Existen dos tipos de encriptación, determinista (*deterministic*) y aleatorizado (*randomized*).

- Deterministic: este tipo de encriptación siempre generará un texto encriptado similar al mismo tipo de dato. Si queremos implementar búsquedas y agrupamientos en una columna de tabla, se recomienda utilizar encriptación determinista para tal columna.
- Randomized: una encriptación aleatorizada generará un texto encriptado diferente para el mismo tipo de dato. Si intentamos encriptar cualquier dato, se recomienda esta encriptación si la columna no es usada para implementar búsquedas y agrupamientos.

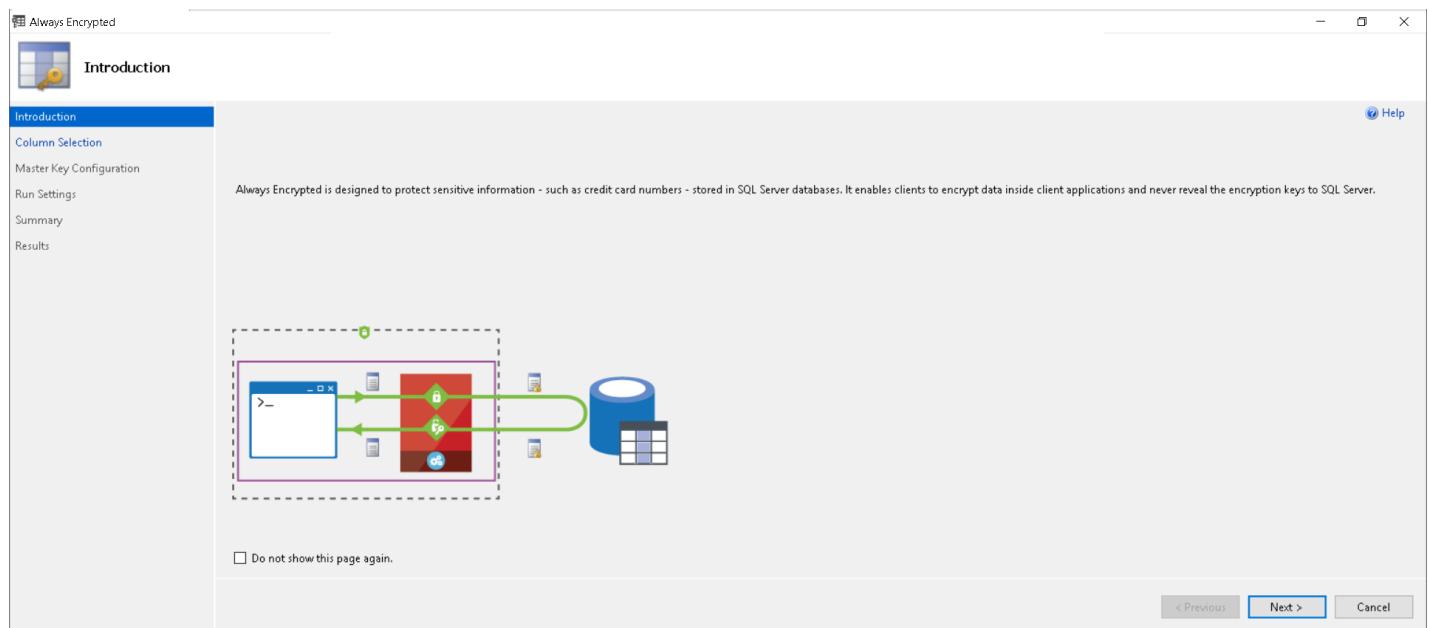
Ahora vamos a configurar SSMS para habilitar la característica **Always Encrypted**, para ello, primero vamos a comprobar si hay **CMKs** y **CEKs** existentes. Nos dirigimos a *nuestra base de datos > Security > Always Encrypted Keys > CMKs / CEKs*:



Y como podemos observar, no disponemos de ninguna clave, eso significa que procedemos a habilitar la característica, entonces hacemos click derecho en *nuestra base de datos > Tasks > Encrypt Columns...*



Y nos aparecerá este asistente:



Hacemos click en *Next* y nos saldrá una pantalla de selección de tablas de nuestra base de datos y vamos a seleccionar la tabla **SAL\_REUNION** y encriptaremos todas las columnas excepto las de los IDs:

The screenshot shows the 'Column Selection' step of the wizard. The left sidebar lists 'Introduction', 'Column Selection' (which is selected), 'Master Key Configuration', 'Run Settings', 'Summary', and 'Results'. The main pane displays a table of columns for the 'dbo.SAL\_REUNION' table. The columns are listed under the 'Name' column, and their properties are shown in the 'State', 'Encryption Type', and 'Encryption Key' columns. The 'Encryption Type' for all columns is set to 'Randomized', and the 'Encryption Key' dropdown for each column is set to 'CEK\_Auto1 (New)'. A checkbox 'Apply one key to all checked columns:' is checked. At the bottom, there is a checkbox 'Show affected columns only' and navigation buttons: '< Previous', 'Next >', and 'Cancel'.

Name	State	Encryption Type	Encryption Key
dbo.SAL_REUNION		Randomized	CEK_Auto1 (New)
ID_reunion		Randomized	CEK_Auto1 (New)
fecha_hora_inicio		Randomized	CEK_Auto1 (New)
fecha_hora_final		Randomized	CEK_Auto1 (New)
nom_reunion		Randomized	CEK_Auto1 (New)
descripcion		Randomized	CEK_Auto1 (New)
lugar		Randomized	CEK_Auto1 (New)
SAL_SCOUTER_ID_personal			
SAL_GRUPO_SCOUT_ID_grupo_scout			

Si salen señales amarillas de advertencia, significa que cambiará la codificación de las tablas a otra adecuada para la encriptación. Preparado todo, en el siguiente apartado aparece la configuración de la **Master Key**, dejamos la configuración por defecto:

Always Encrypted

## Master Key Configuration



Introduction  
Column Selection  
**Master Key Configuration**  
Run Settings  
Summary  
Results

To generate a new column encryption key, a column master key must be selected to protect it. The column master key is stored outside of the database.

Select column master key:

Auto generate column master key

Select the key store provider:

Windows certificate store (i)

Azure Key Vault (i)

Select a master key source:

Current User

Y al final nos aparecerá un sumario con todas las acciones preparadas a ejecutar:

Always Encrypted

## Summary



Introduction  
Column Selection  
Master Key Configuration  
Run Settings  
**Summary**  
Results

**Verify the choices made in this wizard.**

Click Finish to perform the operations with the following settings:

New encryption key: CEK_Auto1
Encrypt column fecha_hora_inicio
Table name: SAL_REUNION
Encryption key name: CEK_Auto1
Encryption type: Randomized
Encrypt column fecha_hora_final
Table name: SAL_REUNION
Encryption key name: CEK_Auto1
Encryption type: Randomized
Encrypt column nom_reunion
Table name: SAL_REUNION
Encryption key name: CEK_Auto1
Encryption type: Randomized
Encrypt column descripcion
Table name: SAL_REUNION
Encryption key name: CEK_Auto1
Encryption type: Randomized
Encrypt column lugar
Table name: SAL_REUNION
Encryption key name: CEK_Auto1
Encryption type: Randomized

Y al acabar de ejecutarse todo, aparece la pantalla final con las operaciones realizadas con éxito y vemos en el *Object Explorer* que se han creado las claves:

The screenshot shows the 'Always Encrypted' wizard log report. On the left, there's a sidebar with tabs: 'Introduction', 'Column Selection', 'Master Key Configuration', 'Run Settings', and 'Results'. The 'Results' tab is selected. The main area is titled 'Summary' and contains a table with three rows:

Task	Details
Generate new column master key CMK_Auto1 in Windows certificate store CurrentUser	Passed
Generate new column encryption key CEK_Auto1	Passed
Performing encryption operations	Passed

At the bottom of the summary table, there's a link 'Always Encrypted Wizard Log Report'. At the very bottom right, there are buttons for '< Previous', 'Next >', and 'Close'.

The screenshot shows the 'Object Explorer' window. The tree view displays the database structure of 'SAL\_SCOUTS'. Under the 'Security' node, the 'Always Encrypted Keys' node is expanded, showing two entries: 'Column Master Keys' and 'Column Encryption Keys'. Under 'Column Master Keys', there is one entry: 'CMK\_Auto1'. Under 'Column Encryption Keys', there is one entry: 'CEK\_Auto1'. The 'CEK\_Auto1' entry is highlighted with a blue selection bar at the bottom of the tree.

Si hacemos una consulta a la tabla observamos que están las columnas totalmente encriptadas:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_SCOUTS' is selected. In the center pane, a query window titled 'always\_encrypted.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio' contains the following script:

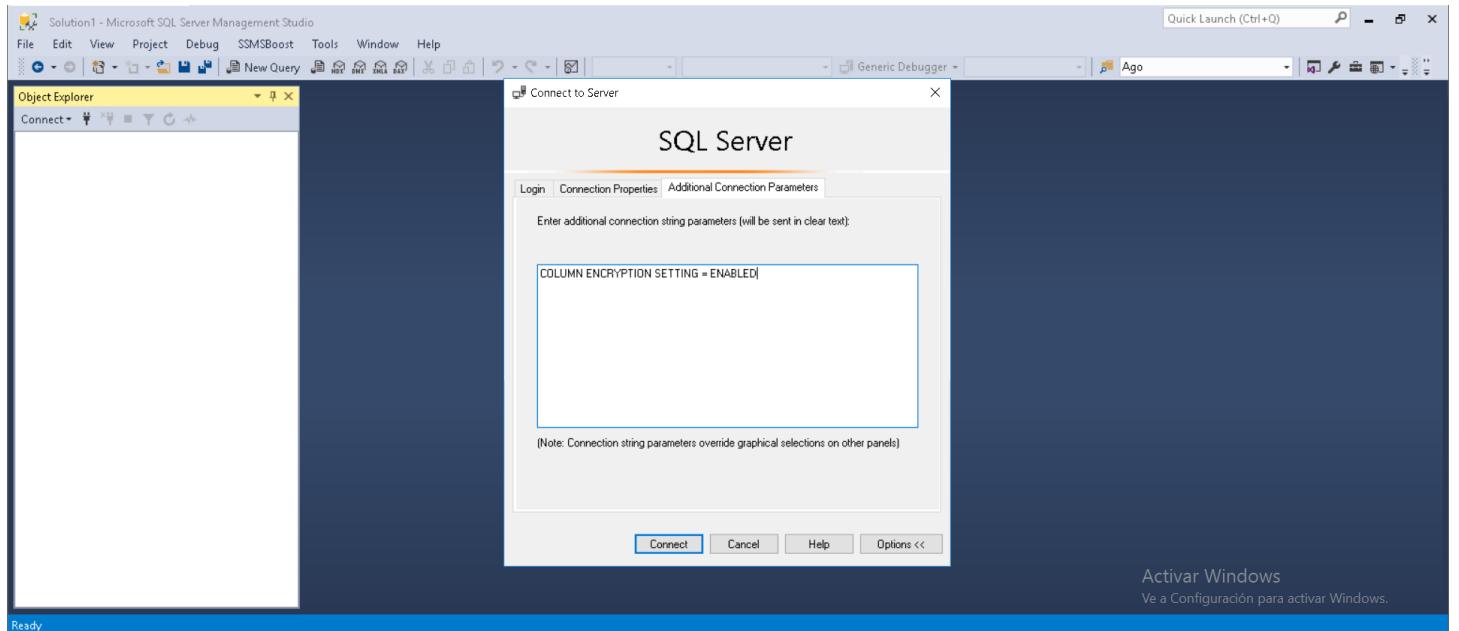
```

1 USE SAL_SCOUTS
2 GO
3
4 SELECT * FROM SAL_Reunion
5 GO

```

The results grid displays 9 rows of data. The columns are labeled: ID\_reunion, fecha\_hora\_inicio, fecha\_hora\_final, nom\_reunion, and descripcion. The data is heavily encrypted, appearing as long strings of hex digits. A status bar at the bottom indicates: 'Query executed successfully.' and 'SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\|SAL\_SCOUTS... | SAL\_SCOUTS | 00:00:01 | 961 rows'.

Para poder visualizar de nuevo las columnas tenemos que cerrar la sesión actual, y en el asistente de inicio de sesión, en la pestaña *Additional Configuration Parameters* debemos introducir la opción **COLUMN ENCRYPTION SETTING = ENABLED**:



Y de esta manera podemos visualizar de vuelta las filas de las columnas encriptadas:

```

1 USE SAL_SCOUTS
2 GO
3
4 SELECT * FROM SAL_Reunion
5 GO

```

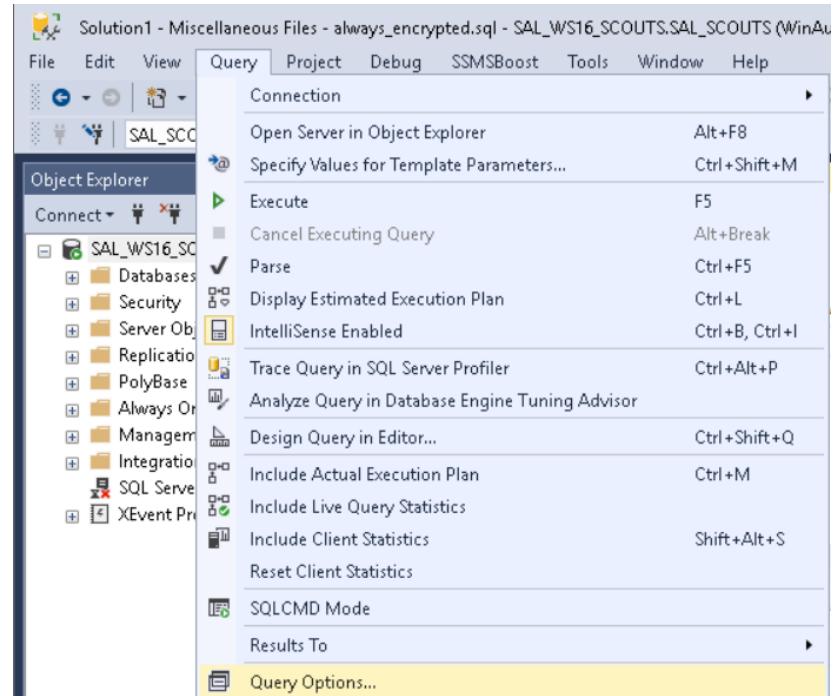
ID_reunion	fecha_hora_inicio	fecha_hora_final	nom_reunion	descripcion	lugar	SAL_SCOUTER_ID_personal	SAL_GRUPO_SCOUT_
1	2021-04-12 19:56:34.000	2021-03-02 18:32:59.000	SunZone Baby Sunscreen SPF-60	partuerit montes nascetur ridiculus mus vivamus ...	Spain	6	4
2	2020-10-30 11:54:25.000	2021-01-03 03:07:56.000	Perphenazine and Amitriptyline Hydrochloride	congue eget semper rutrum nulla nunc purus phas...	Spain	6	5
3	2020-06-13 08:13:18.000	2020-09-18 14:39:49.000	Hydrocodone Bitartrate and Acetaminophen	nisi at nibh in hac habitasse platea dictumst aliqua...	Spain	6	5
4	2020-09-21 15:26:11.000	2021-03-16 16:35:36.000	Carvedilol	volutpat convallis morbi odio odio elementum eu in...	Spain	6	4
5	2020-08-25 18:49:57.000	2021-03-11 11:22:44.000	Albuterol Sulfate	penatibus et magnis dis parturient montes nascetur...	Spain	6	2
6	2020-10-11 10:28:33.000	2021-05-15 02:17:36.000	DG Health Cold and Allergy	libero convallis eget eleifend luctus ultricies eu nib...	Spain	6	2
7	2021-01-16 15:45:09.000	2020-09-28 00:39:44.000	Carvedilol	condimentum curabitur in libero ut massa volutpat ...	Spain	6	5
8	2021-02-25 14:40:34.000	2020-05-27 04:43:07.000	ULMUS AMERICANA POLLEN	sed ante vivamus tortor duis mattis egestas metus ...	Spain	6	5
9	2020-08-24 14:10:14.000	2020-12-03 02:58:20.000	coated sennets nonstop sleep aid	malesuada eu in euodiet et commodo ut ultratac et est	Spain	6	5

A partir de ahora debemos tener en cuenta que el hecho de realizar inserciones no será lo mismo ya que requiere que los campos relacionados con la tabla han de estar parametrizados, lo que se llama en SQL Server "*Parametrization for Always Encrypted*" (Parametrización para *Always Encrypted*).

Esta característica está disponible en SSMS desde la versión 17.x en adelante que permite:

- La capacidad de ejecutar **INSERT INTO**, **UPDATE** y filtrar por valores almacenados en columnas encriptadas desde la ventana del *Query Editor* o mediante **T-SQL**.

Esta misma característica aparece como opción en el apartado *Query > Query Options...*:



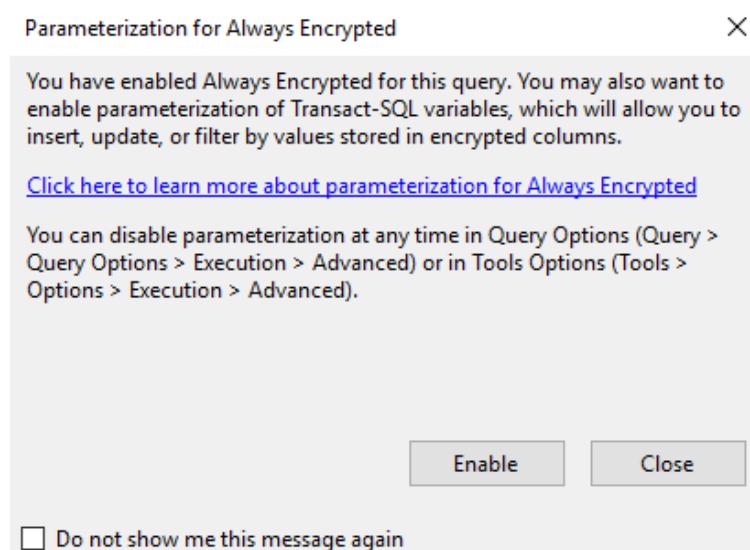
Cuando está habilitada, mapea las variables T-SQL a parámetros de consulta y refracta las consultas en sentencias parametrizadas, como si se tratase de un procedimiento almacenado, pero no lo es:

The screenshot shows a SQL Server Management Studio window with the following content:

- Query Editor:** Contains T-SQL code: `USE S\SQL_SCOUTS (56) GO SELECT GO`.
- Results Pane:** Displays a table with columns `ID_reunion` and `fecha`. The data is as follows:
 

	ID_reunion	fecha
1	1	202
2	2	202
3	3	202
4	4	202
5	5	202
- Query Options Dialog:**
  - Execution Tab:** Advanced section is selected.
  - Parameterization Settings:**  Enable Parameterization for Always Encrypted
  - Other Options:** SET XACT\_ABORT ON, SET TRANSACTION ISOLATION LEVEL (READ COMMITTED), SET DEADLOCK\_PRIORITY (Normal), SET LOCK TIMEOUT (-1 milliseconds), SET QUERY\_GOVERNOR\_COST\_LIMIT (0).

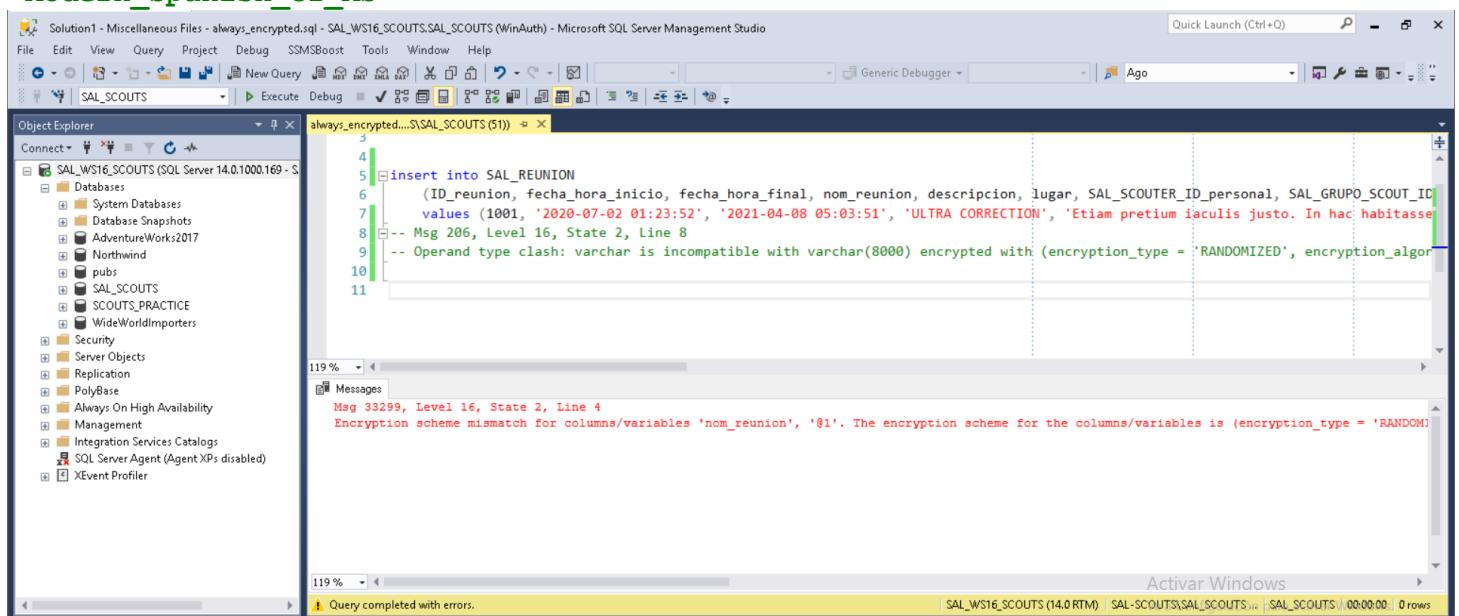
Por ejemplo, si realizamos una consulta sobre una sesión cuya conexión tiene la opción **COLUMN ENCRYPTION SETTING = ENABLED** y dicha parametrización encendida, el perfil de registro de SQL Server capturará en dos llamadas RPC (*Remote Procedure Call*, es un procedimiento almacenado interno en SQL Server que ejecuta remotamente una sentencia T-SQL desde una instancia SQL Server a otra instancia SQL Server usando un servidor enlazado, como ***sp\_executesql*** por ejemplo) en vez de una única sentencia.



Si nos aparece este mensaje, nos recomienda habilitar la parametrización (en el caso de no haberlo hecho), que como comentamos antes, nos ayudará a mejorar las inserciones de inserción y actualización en las columnas encriptadas.

Para empezar, si intentamos realizar una inserción sin parametrizarla antes y/o habilitar la opción, nos saldrá un error de conflicto con el tipo de dato:

```
insert into SAL_REUNION
    (ID_reunion, fecha_hora_inicio, fecha_hora_final, nom_reunion, descripcion,
lugar, SAL_SCOUTER_ID_personal, SAL_GRUPO_SCOUT_ID_grupo_scout)
    values (1001, '2020-07-02 01:23:52', '2021-04-08 05:03:51', 'ULTRA CORRECTION',
'Etiam pretium iaculis justo. In hac habitasse platea dictumst. Etiam faucibus
cursus urna. Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius.
Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt lacus
at velit. Vivamus vel nulla eget eros elementum pellentesque.', 'Spain', 6, 3);
-- Msg 206, Level 16, State 2, Line 8
-- Operand type clash: varchar is incompatible with varchar(8000) encrypted with
(encryption_type      =      'RANDOMIZED',      encryption_algorithm_name      =
'AEAD_AES_256_CBC_HMAC_SHA_256',      column_encryption_key_name      =      'CEK_Auto1',
column_encryption_key_database_name      =      'SAL_SCOUTS')      collation_name      =
'Modern_Spanish_CI_AS'
```



Y si parametrizamos las consultas también sin habilitar la opción, nos aparecerá otro error sobre incompatibilidad de esquemas/variables de encriptación:

```
DECLARE @name VARCHAR(50) = 'Carvedilol'
SELECT * FROM SAL_REUNION
WHERE nom_reunion = @name;
GO

-- Msg 33299, Level 16, State 2, Line 14
-- Encryption scheme mismatch for columns/variables '@name', 'nom_reunion'. The
encryption scheme for the columns/variables is (encryption_type = 'RANDOMIZED',
encryption_algorithm_name          =          'AEAD_AES_256_CBC_HMAC_SHA_256',
column_encryption_key_name = 'CEK_Auto1', column_encryption_key_database_name =
'SAL_SCOUTS')
-- and the expression near line '3' expects it to be (encryption_type =
'DETERMINISTIC') (or weaker).
```

Solution1 - Miscellaneous Files - always\_encrypted.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio

```

-- Operand type clash: varchar is incompatible with varchar(8000) encrypted with (encryption_type = 'RANDOMIZED', encryption_algorithm = 'AEAD_AES_256_CBC_HMAC_SHA_1_32')
DECLARE @name VARCHAR(50) = 'Carvedilol'
SELECT * FROM SAL_REUNION
WHERE nom_reunion = @name;
GO

```

-- Msg 33299, Level 16, State 2, Line 14  
-- Encryption scheme mismatch for columns/variables '@name', 'nom\_reunion'. The encryption scheme for the columns/variables is (encryption\_type = 'RANDOMIZED', encryption\_algorithm = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_1\_32')  
-- and the expression near line '3' expects it to be (encryption\_type = 'DETERMINISTIC') (or weaker).

Msg 33299, Level 16, State 2, Line 14  
Encryption scheme mismatch for columns/variables '@name', 'nom\_reunion'. The encryption scheme for the columns/variables is (encryption\_type = 'RANDOMIZED', encryption\_algorithm = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_1\_32')

Query completed with errors.

Si nos fijamos bien, estas columnas han sido encriptadas en la manera *Randomized*, que no permite realizar búsquedas ni agrupamiento de tablas, pero si las hubiera encriptado en la manera *Deterministic*, entonces podría realizar la consulta como tal, parametrizada:

Solution1 - Miscellaneous Files - always\_encrypted.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio

```

DECLARE @name VARCHAR(50) = 'Carvedilol';
SELECT * FROM dbo.SAL_REUNION
WHERE nom_reunion = @name;
GO

```

ID_reunion	fecha_hora_inicio	fecha_hora_final	nom_reunion	descripcion	lugar	SAL_SCOUTER_ID_personal	SAL_GRUPO_SCOUT_ID_grupo_scout
4	2020-09-21 15:26:11.000	2021-03-16 16:35:36.000	Carvedilol	volutpat convallis morbi odio odio elementum eu ...	Spain	6	4
7	2021-01-16 15:45:09.000	2020-09-28 00:39:44.000	Carvedilol	condimentum curabitur in libero ut massa volutp...	Spain	6	5
254	2021-04-13 13:36:00.000	2021-01-15 05:15:20.000	Carvedilol	mi sit amet lobortis sapien sapien non mi integer ...	Spain	6	2

Query executed successfully.

## • 2.7. Tareas sobre BD en SSMS

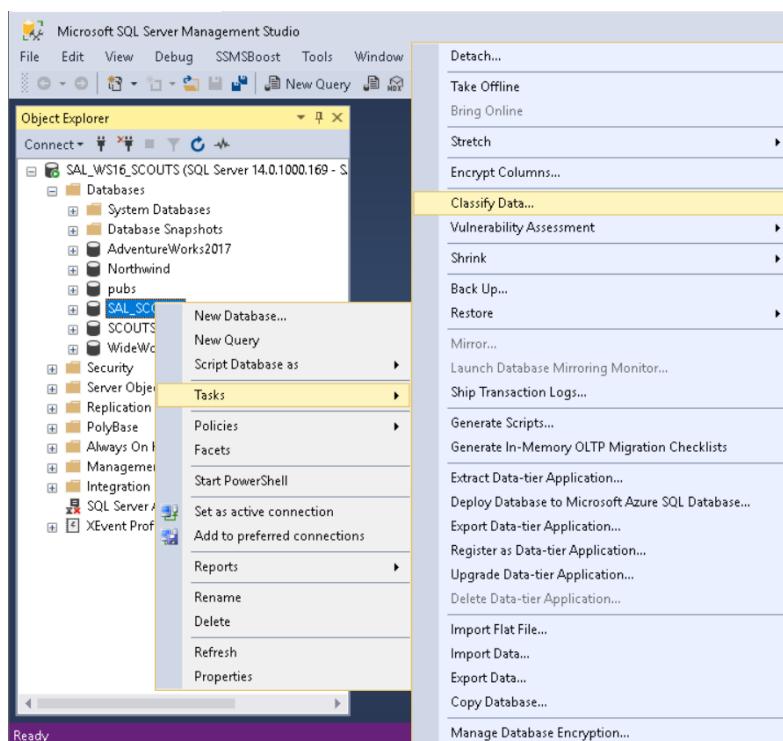
### • 2.7.1. Data Discovery and Classification

**Data Discovery and Classification** (Clasificación y Detección de Datos) es una herramienta integrada en **SSMS** para detectar, clasificar, etiquetar y notificar los datos confidenciales de las bases de datos. La detección y clasificación de la información más confidencial pueden desempeñar un papel fundamental en el estado de protección de la información de la organización con el objetivo de ayudar a cumplir con los estándares de privacidad de los datos y en supervisar el acceso a bases de datos o columnas que contengan datos altamente confidenciales.

El procedimiento de clasificación sigue de la siguiente manera: el motor de base de datos escanea primero a través de las columnas y detecta las columnas que pueden contener datos sensibles (una búsqueda realizada mediante sentencias **T-SQL** para analizarlas sintácticamente). Y después de eso, podemos revisarlas y aplicar las recomendaciones de clasificación adecuadas.

Las propiedades avanzadas nos permitirán especificar la etiqueta y el tipo de información de datos sensibles. El tipo de información puede ser el siguiente: búsquedas de Internet, contactos, credenciales, tarjetas de crédito, etc. Pero cabe indicar que la debilidad de esta clasificación se debe al idioma que se está utilizando en los nombres de las columnas, procedimientos almacenados, funciones, etc. ya que solo soporta el inglés y no está adaptado a otros idiomas como el español, francés, alemán, italiano, entre otros más.

Para llevar a la práctica esta herramienta, vamos a ejecutar esta tarea a nuestra base de datos, para ello hacemos *click derecho en ella > Tasks > Classify Data...*:



Una vez ejecutado obtenemos un informe de clasificación. En este informe, podemos ver las recomendaciones de clasificación tal que todavía no hay columnas clasificadas:

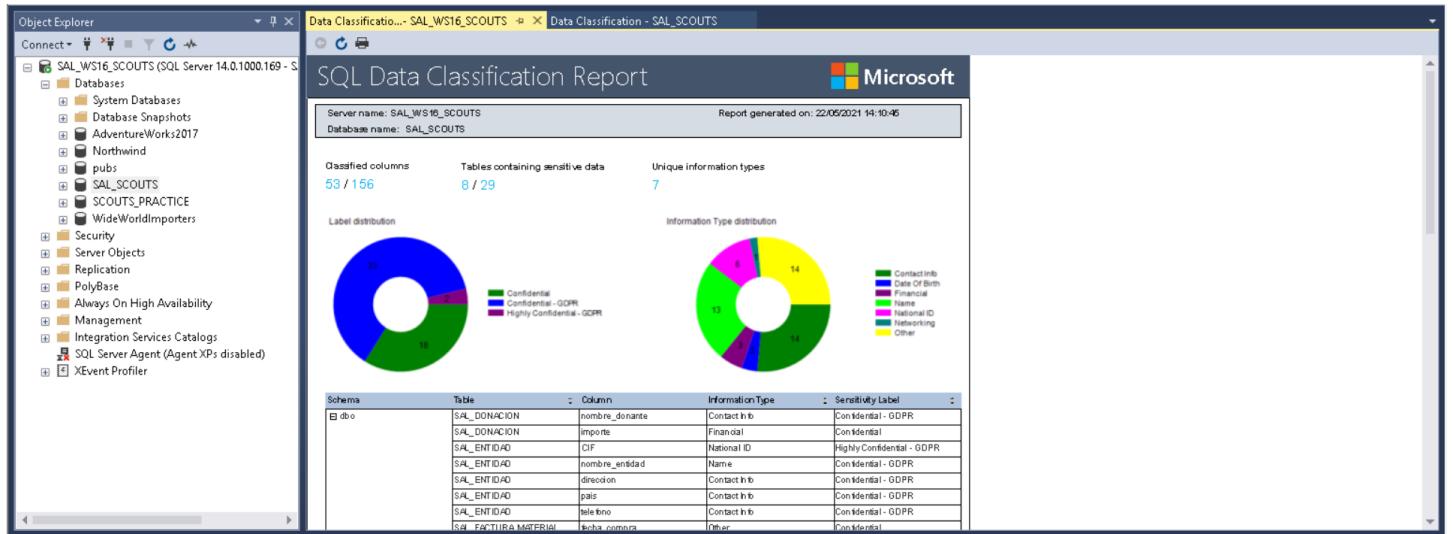
The screenshot shows the 'Data Classification - SAL\_SCOUTS' window. On the left, the Object Explorer displays the database structure. The main pane shows a message: '12 columns with classification recommendations (click to view)'. Below it, a table titled '0 classified columns' has columns: Schema, Table, Column, Information Type, and Sensitivity Label. A large icon of a document with a lock is centered. A message at the bottom says 'Currently you do not have any columns classified' with a 'Add Classification' button.

Para clasificarlas hacemos click en *Add Classification* y podemos ver las opciones de datos sensibles con las columnas personalizadas representadas a través del tipo de información y la etiqueta de sensibilidad o de recomendaciones aceptadas:

This screenshot shows the same interface after adding classifications. The message now says 'There are pending classification updates. Please save.' The table now lists 9 classified columns. An 'Add Classification' dialog is open on the right, showing fields for Schema (dbo), Table (SAL\_PERSONAL), Column (scout\_user), Information Type (Networking), and Sensitivity Label (Confidential). Buttons for 'Add' and 'Cancel' are at the bottom.

The final screenshot shows the interface after accepting recommendations. The message is '12 columns with classification recommendations (click to minimize)'. The table shows 12 classified columns across various tables like SAL\_DONACION, SAL\_EVENTOS, SAL\_INVENTARIO, and SAL\_PERSONAL, with details like 'Information Type' set to 'Other' and 'Sensitivity Label' set to 'Confidential'. A message at the bottom says 'Accept selected recommendations'.

Después de haber seleccionado las opciones adecuadas de sensibilidad, podemos aplicarlas y guardarlas. Y siguiendo esto podemos proceder ya con el **Informe de Clasificación de Datos SQL (SQL Data Classification Report)**:



Este informe nos muestra detalles de qué está marcado para clasificar y el total de cuántas columnas y tablas están marcadas basadas en las totales en la base de datos.

## • 2.7.2. *Vulnerability Assessment*

**Vulnerability Assessment** (evaluación de vulnerabilidades) es otra herramienta de **SSMS** (incluida en la versión 17.4) que permite detectar posibles vulnerabilidades en nuestra base de datos, realizar un seguimiento y corregirlas con el objetivo de mejorar la seguridad de la base de datos de manera proactiva.

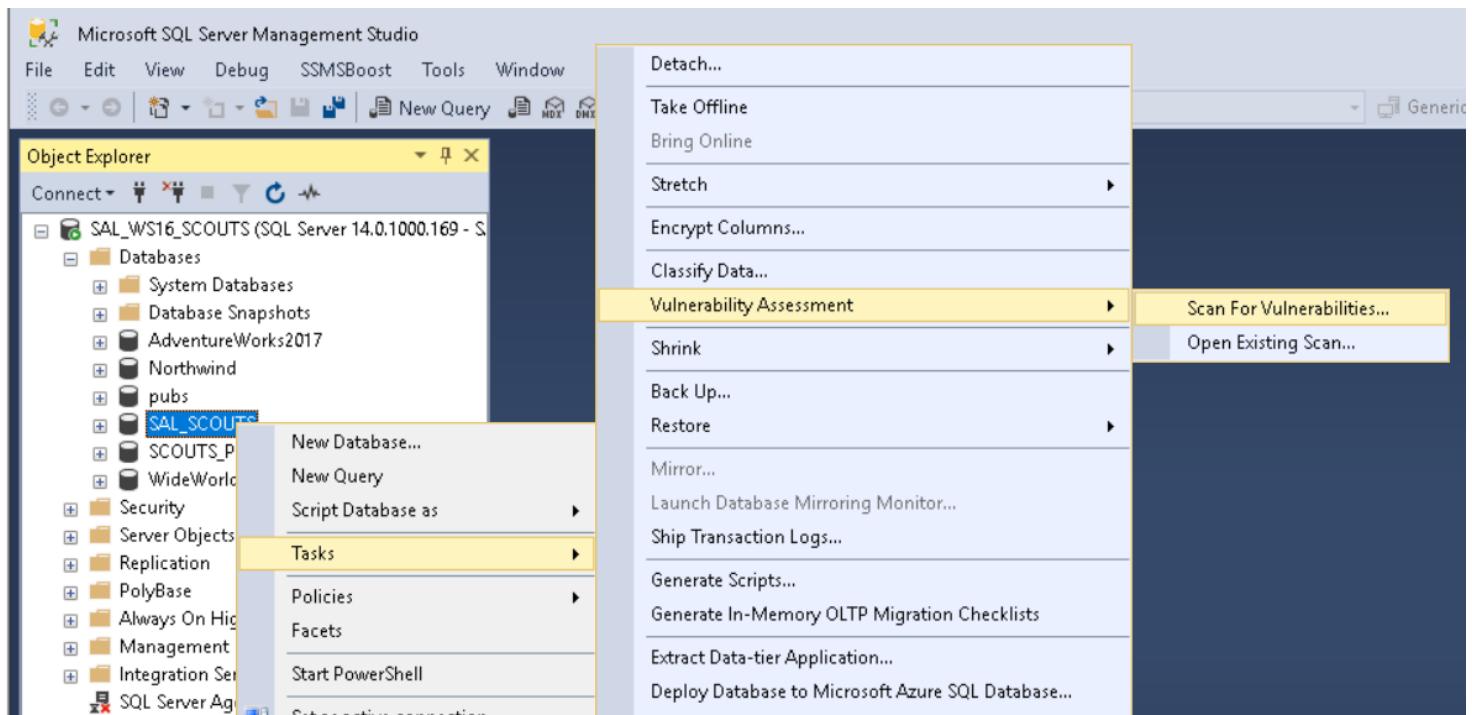
Las características de esta herramienta son proporcionar visibilidad sobre el estado de seguridad e incluir acciones recomendadas para resolver problemas de seguridad y mejorar la seguridad de la base de datos. En este sentido ayuda a:

- Satisfacer los requisitos de cumplimiento que requieren los informes de examen de base de datos.
- Cumplir los estándares de privacidad de los datos.
- Supervisar un entorno de base de datos dinámico donde resulta difícil realizar un seguimiento de los cambios.

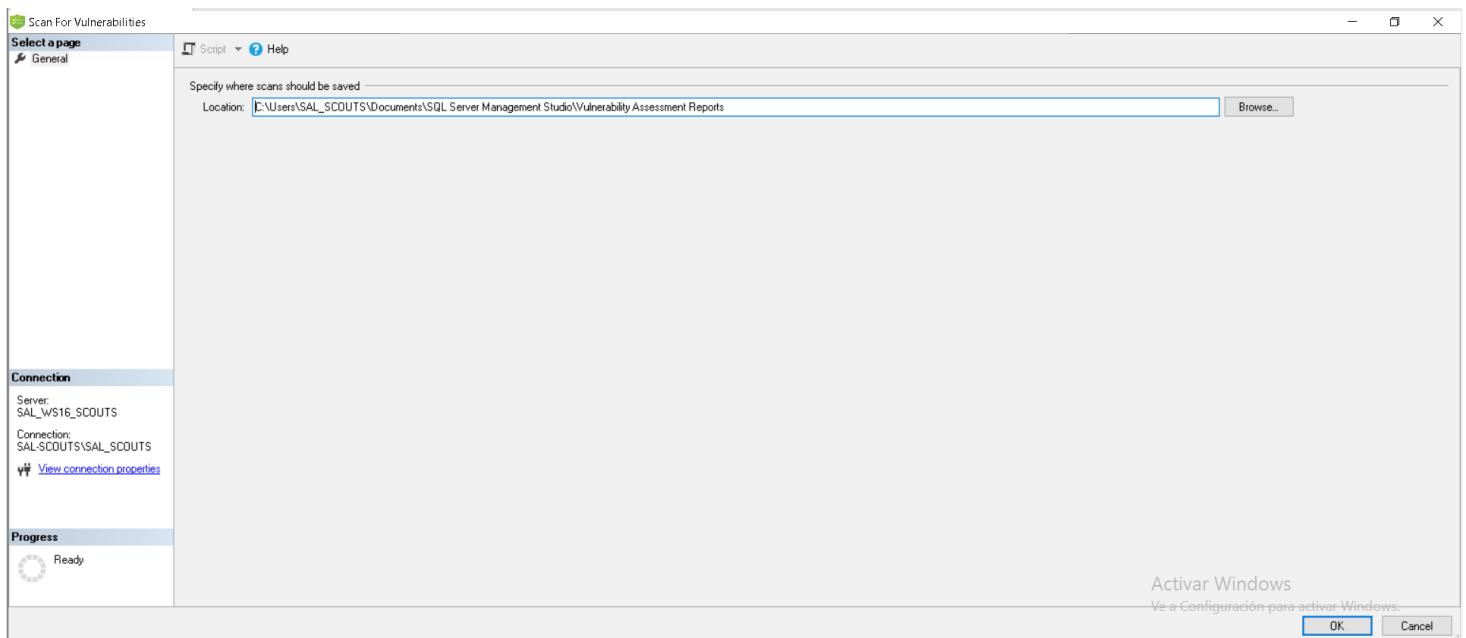
El servicio **VA** ejecuta un examen directamente en la base de datos. El servicio emplea una base de reglas de conocimientos que marcan las vulnerabilidades de seguridad y resaltan las desviaciones con respecto a los procedimientos recomendados, como errores de configuración, permisos excesivos y datos confidenciales sin protección. Las reglas se basan en procedimientos recomendados de Microsoft y se centran en los problemas de seguridad que presentan los riesgos más grandes para la base de datos y sus valiosos datos. Estas reglas también representan muchos de los requisitos que deben cumplir diversos organismos reguladores para satisfacer los estándares de cumplimiento.

Los resultados del examen incluyen pasos que requieren acción para corregir cada uno de los problemas y proporcionan scripts de solución personalizados donde sea aplicable. Un informe de evaluación se puede personalizar para el entorno y establecer una línea de base aceptable para las configuraciones de permisos, configuraciones de características y configuraciones de bases de datos.

Para ejecutar el examen de vulnerabilidades de nuestra base de datos hacemos *click derecho en ella > Tasks > Vulnerability Assessment > Scan for Vulnerabilities...*:



Nos aparecerá el asistente de escaneo de vulnerabilidades y nos pedirá el directorio para guardar los exámenes, hacemos click en **OK** y procederá con el escaneo:



Una vez finalizado el examen, se muestra automáticamente el informe de escaneo. El informe presenta información general sobre el estado de seguridad, cuántos problemas se encontraron y sus niveles de gravedad respectivos. Los resultados incluyen advertencias sobre las desviaciones con respecto a los procedimientos recomendados, así como una instantánea de la configuración relacionada con la seguridad, como las entidades de seguridad y los roles de base de datos y sus permisos asociados. El informe

de examen también proporciona un mapa de datos confidenciales detectados en la base de datos e incluye recomendaciones de los métodos integrados disponibles para protegerlo. Éste es el resultado que obtenemos de nuestra base de datos:

ID	Security Check	Category	Risk	Additional Information
VA1245	The dbo information should be consistent between the target DB and master	Surface Area Reduction	High	
VA1281	All memberships for user-defined roles should be intended	Auditing and Logging	Medium	No baseline set
VA1287	Sensitive data columns should be classified	Data Protection	Medium	No baseline set
VA2051	Minimal set of principals should be granted database-scoped VIEW DEFINITION permissions on schema	Authentication and Authorization	Medium	No baseline set
VA1069	Permissions to select from system tables and views should be revoked from non-sysadmins	Authentication and Authorization	Low	No baseline set
VA2030	Minimal set of principals should be granted database-scoped SELECT or EXECUTE permissions	Authentication and Authorization	Low	No baseline set
VA2031	Minimal set of principals should be granted database-scoped SELECT permission on objects or columns	Authentication and Authorization	Low	No baseline set
VA2032	Minimal set of principals should be granted database-scoped SELECT or EXECUTE permissions on schema	Authentication and Authorization	Low	No baseline set
VA2033	Minimal set of principals should be granted database-scoped EXECUTE permission on objects or columns	Authentication and Authorization	Low	No baseline set
VA2041	Minimal set of principals should be granted low impact database-scoped permissions on objects or columns	Authentication and Authorization	Low	No baseline set
VA1054	Excessive permissions should not be granted to PUBLIC role on objects or columns	Authentication and Authorization	Low	No baseline set
		Activar Windows		Ver configuración para activar Windows.

Hemos obtenido 13 errores: **1 de alto riesgo**, **3 de riesgo intermedio** y **9 de riesgo bajo**. Si hacemos click en el error de **alto riesgo**, por ejemplo, nos muestra la siguiente explicación:

**Name:** VA1245 - The dbo information should be consistent between the target DB and master

**Risk:** High

**Status:** Fail

**Description:** There is redundant information about the dbo identity for any database: metadata stored in the database itself and metadata stored in master DB. This rule checks that this information is consistent between the target DB and master.

**Impact:** Both copies of dbo metadata should match to avoid potential system problems such as permission problems when using some features such as CLR.

**Rule Query:**

```
SELECT CASE
    WHEN EXISTS (SELECT *
        FROM sys.database_principals dbprs,
```

- Nombre: **VA1245** – La información de **dbo** debería ser consistente entre la base de datos objetivo (refiriéndose a **SAL\_SCOUTS**) y **master**
- Riesgo: Alto
- Estado: Fallido
- Descripción: Hay información redundante sobre la identidad **dbo** para cualquier base de datos: metadatos almacenados en la propia base de datos y metadatos almacenados en la base de datos **master**. Esta regla revisa que esta información es consistente entre la base de datos objetivo (**SAL\_SCOUTS**) y **master**.
- Impacto: Ambas copias de los metadatos de **dbo** deberían coincidir para evitar problemas en el sistema potenciales tales como un permiso que cause problemas cuando se usen características como **CLR** (*Common Language Runtime*, Modelo de Seguridad de integración)
- Regla de consulta:

**SELECT CASE**

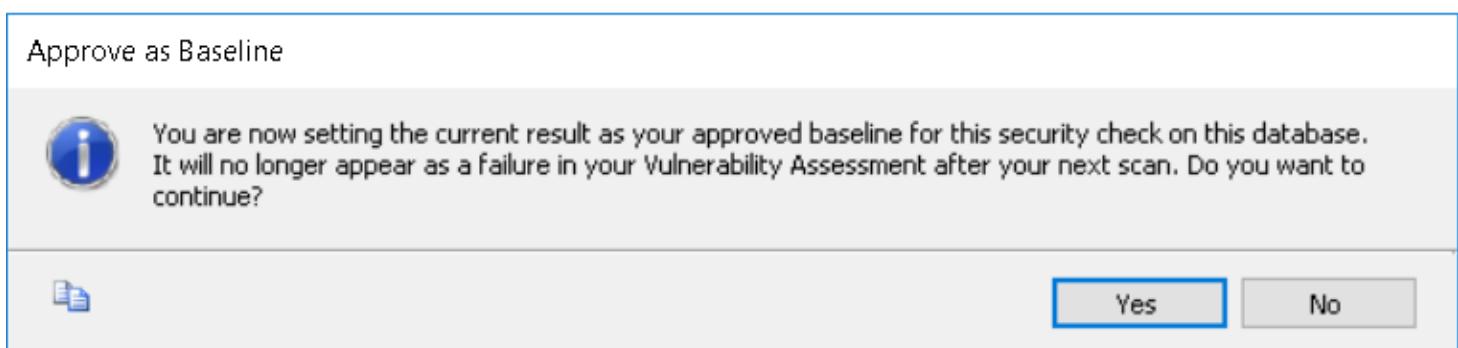
```

WHEN EXISTS (SELECT *
              FROM sys.database_principals dbprs,
                   sys.databases dbs
             WHERE dbprs.sid != dbs.owner_sid
               AND dbs.database_id = Db_id()
               AND dbprs.principal_id = 1) THEN 1
ELSE 0
END AS [Violation]

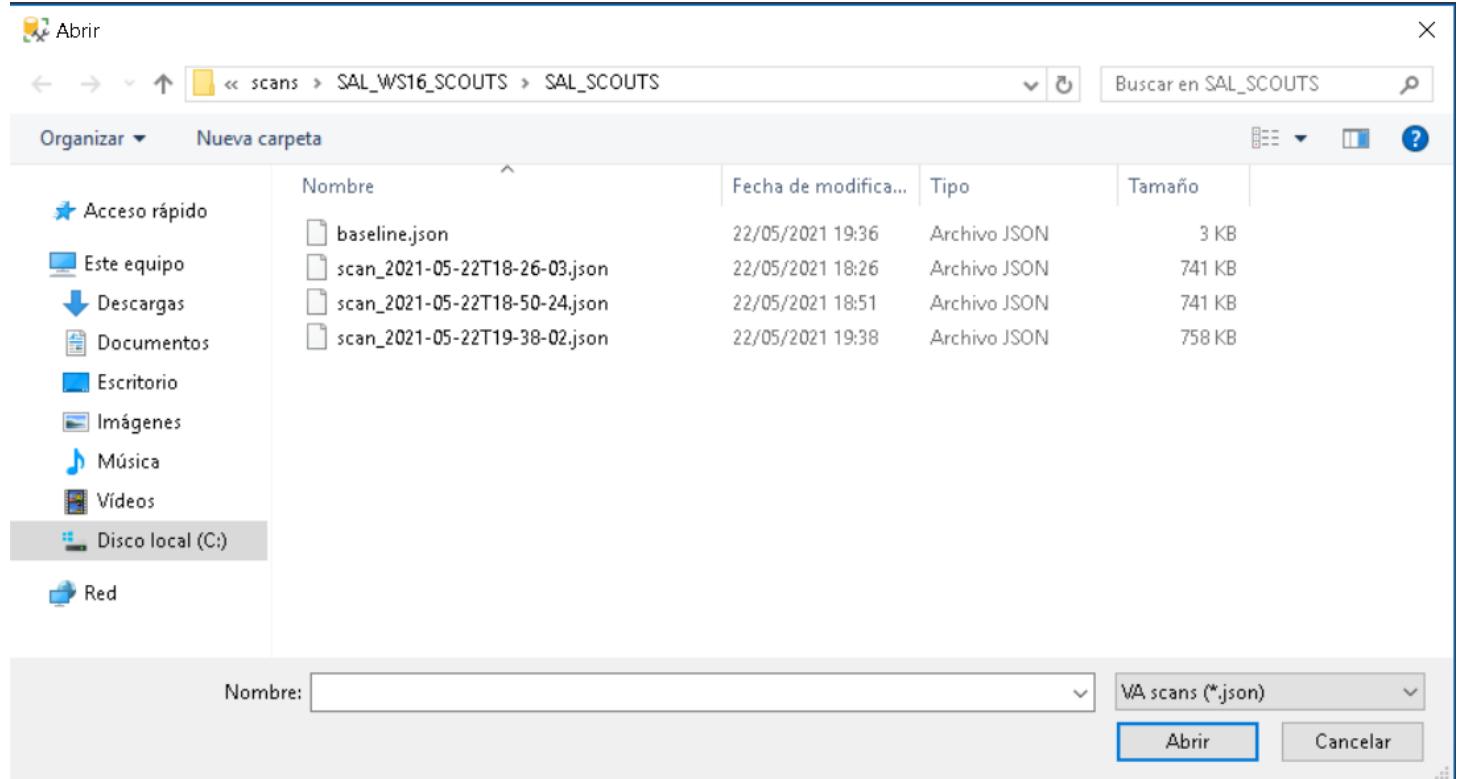
```

- Recomendación de Microsoft: Verdadero
- Línea base: No configurado
- Resultado actual: Falso
- Remedio: Utilizar **ALTER AUTHORIZATION DDL** para especificar al usuario que debería ser el **dbo** para la base de datos

Si hacemos click en **Approve as Baseline** (Aprobar como línea base) nos aparecerá un mensaje advirtiéndonos de que, si aceptamos tal resultado, no volverá a aparecer como fallo en nuestro examen de vulnerabilidades después de haber escaneado la base de datos:



Una vez aceptados y solucionados algún que otro riesgo importante, volvemos a escanear las vulnerabilidades de la base de datos. En el caso de disponer de algún resultado de un escaneo anterior, hacemos click en *Open Existing Scan...* y se nos aparece el explorador de archivos con el directorio de guardado de exámenes con archivos JSON:



Vulnerability Assess...22/05/2021 19:38:12 ✎ X

### Vulnerability Assessment Results

SAL\_WS16\_SCOUTS: SAL\_SCOUTS  
at 5/22/2021 7:38:12 PM

Total security checks	Total failing checks	High Risk	Medium Risk	Low Risk	
54	5	0	0	5	<div style="width: 100px; height: 10px; background-color: #0070C0;"></div>

✖ Failed (5) ✔ Passed (49)

ID	Security Check	Category	Risk	Additional Information
VA2031	Minimal set of principals should be granted database-scoped SELECT permission on objects or columns	Authentication and Authorization	Low	No baseline set
VA2041	Minimal set of principals should be granted low impact database-scoped permissions on objects or columns	Authentication and Authorization	Low	No baseline set
VA1054	Excessive permissions should not be granted to PUBLIC role on objects or columns	Authentication and Authorization	Low	
VA1094	Database permissions shouldn't be granted directly to principals	Authentication and Authorization	Low	
VA1286	Database permissions shouldn't be granted directly to principals (OBJECT or COLUMN)	Authentication and Authorization	Low	

### - 3. Auditoría (Audit)

La **auditoría** de base de datos es un proceso implementado por los auditores de sistemas con el fin de auditar los accesos a los datos, por lo general siguiendo bien una metodología basada en una lista de comprobación que contempla los puntos que se quieren comprobar o mediante la evaluación de riesgos potenciales.

En concreto, se realiza **un examen de los accesos** a los datos almacenados en las bases de datos con el fin de poder **medir, monitorear y tener constancia de los accesos a la información almacenada en las mismas**. Si bien el objetivo puede variar en función de la casuística, en todos los casos el fin último persigue, de uno u otro modo, la seguridad corporativa.

Una auditoría de base de datos, por lo tanto, facilita herramientas eficaces para conocer de forma exacta cuál es la relación de los usuarios a la hora de acceder a las bases de datos, incluyendo las actuaciones que deriven en una generación, modificación o eliminación de datos.

En la práctica, permite responder a muchas preguntas que pueden resultar relevantes a la hora de controlar y auditar. Desde determinar quién accede a los datos, cuándo se accedió o cuál es su **ubicación en la red** y desde **qué dispositivo o aplicación** lo hizo, hasta qué sentencia **SQL** fue ejecutada, así como el resultado del acceso.

En este sentido, la auditoría de base de datos es un control necesario, cuya dificultad aumenta de forma paralela a la creciente complejidad de las tecnologías de bases de datos. Asimismo, las amenazas de seguridad se han multiplicado, apareciendo nuevos riesgos e incrementándose los ya existentes, al tiempo que se amplía su alcance a través de la disciplina conocida como Gestión de Recursos de Información.

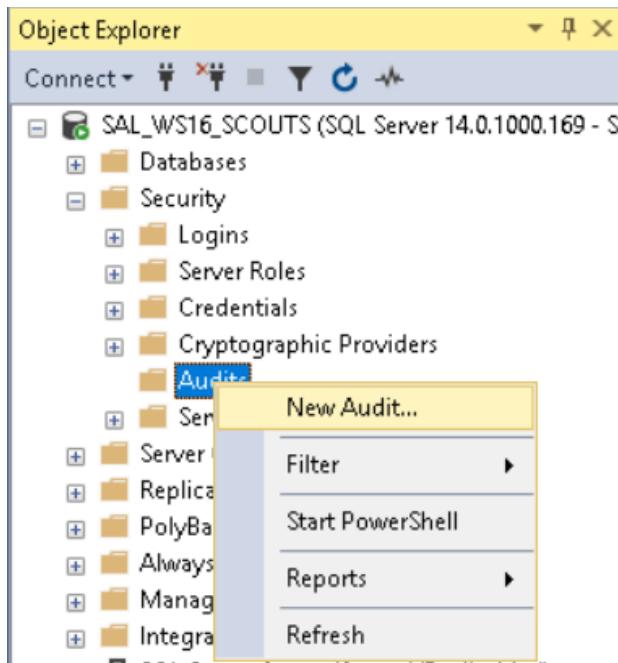
La auditoría de una instancia de **SQL Server** o de una base de datos de **SQL Server** implica el seguimiento y registro de los eventos que se producen en el sistema. El objeto **SQL Server Audit** recopila una única instancia de acciones y grupos de acciones de nivel de servidor o de nivel de base de datos para su supervisión. La auditoría se realiza en el nivel de instancia de **SQL Server**. Es posible tener varias auditorías por cada instancia de **SQL Server**.



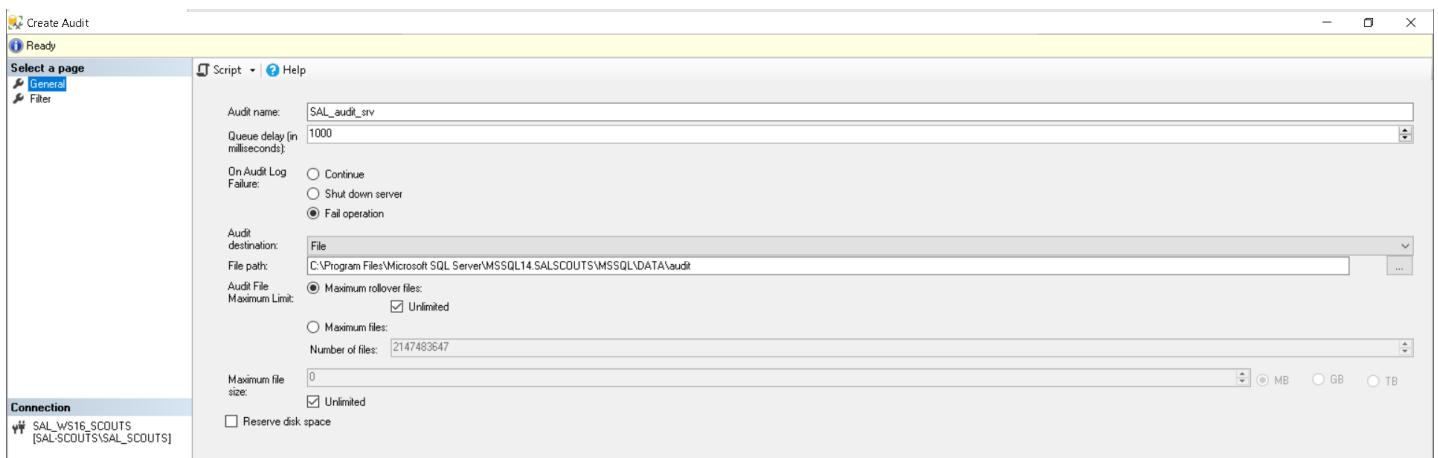
### • 3.1. Auditoría de serv. y especif. auditoría de serv.

Para poder crear una especificación de auditoría de servidor, debe existir la auditoría. Cuando se crea una especificación de auditoría de servidor, está en estado deshabilitado.

Para crear una nueva auditoría nos vamos al *Object Explorer* y hacemos *click derecho en Audits > New Audit...*:



Nos aparecerá un asistente de auditorías en donde nos pedirá los datos para crearla:



O, si no se quiere utilizar el asistente, se puede convertir estos parámetros en una sentencia **T-SQL**:

```

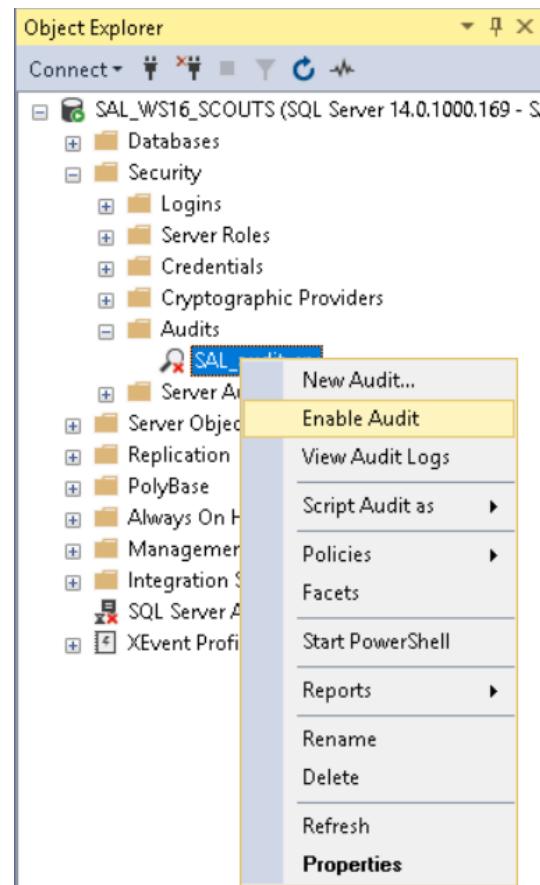
1 USE [master]
2 GO
3
4 CREATE SERVER AUDIT [SAL_audit_srv]
5 TO FILE ( FILEPATH = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SALSCOUTS\MSSQL\DATA\audit'
6   ,MAXSIZE = 0 MB
7   ,MAX_ROLLOVER_FILES = 2147483647
8   ,RESERVE_DISK_SPACE = OFF
9 ) WITH (
10   QUEUE_DELAY = 1000
11   ,ON_FAILURE = FAIL_OPERATION
12 )
13 GO
14

```

Messages  
Commands completed successfully.

Query executed successfully.

Ya creada nuestra auditoría de servidor observamos que está desactivada, para activarla podemos hacerlo desde el *Object Explorer* haciendo *click derecho en la auditoría > Enable Audit:*



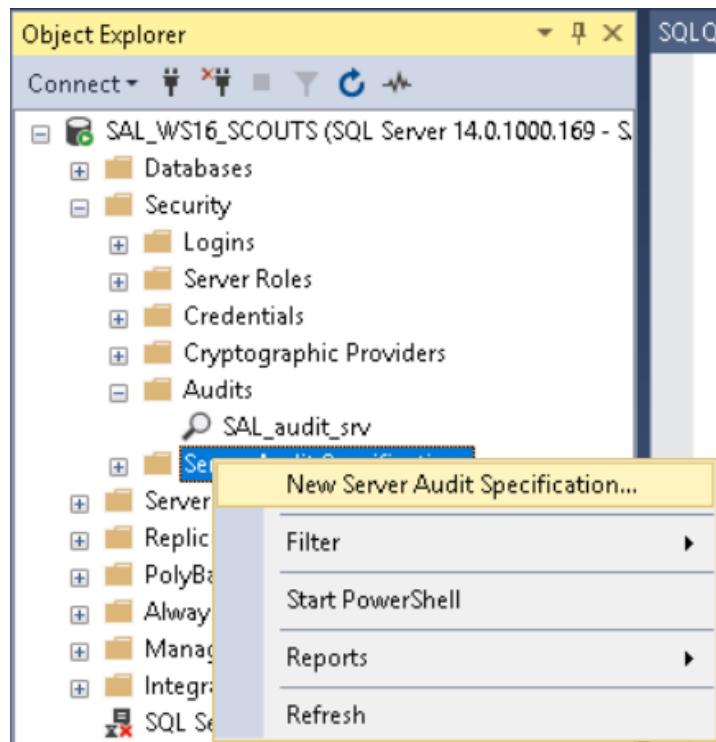
O también ejecutando una sentencia **T-SQL**:

```

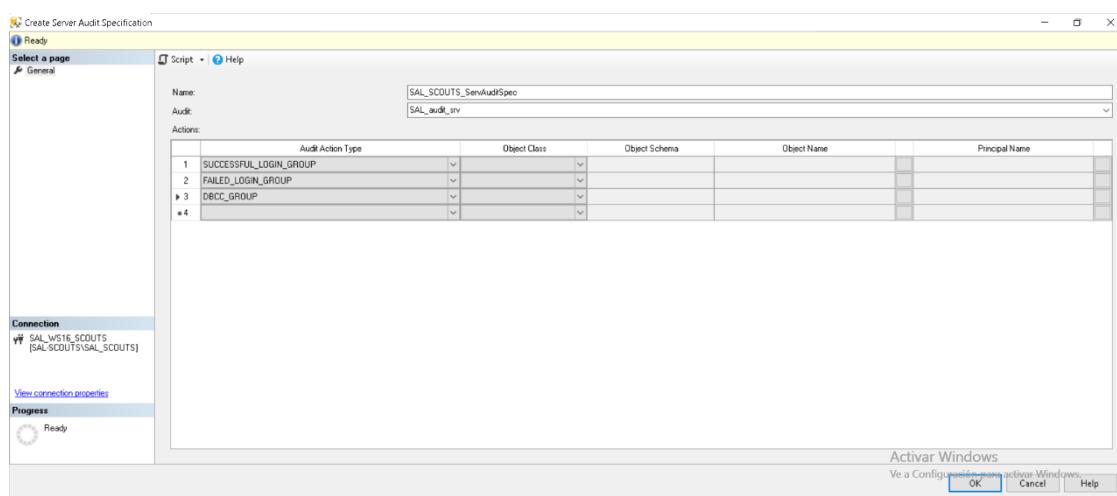
ALTER SERVER AUDIT [SAL_audit_srv] WITH (STATE = ON)
GO

```

Ya activada la auditoría ahora tenemos que crear una **especificación de auditoría de servidor**. La **especificación de auditoría de servidor** es usada para definir qué necesita ser auditado al nivel de servidor; sólo puede haber una **especificación de auditoría de servidor** por auditoría. Para crear la especificación seguimos los mismos pasos que al crear la auditoría, hacemos *click derecho en Server Audit Specifications > New Server Audit Specification...*:

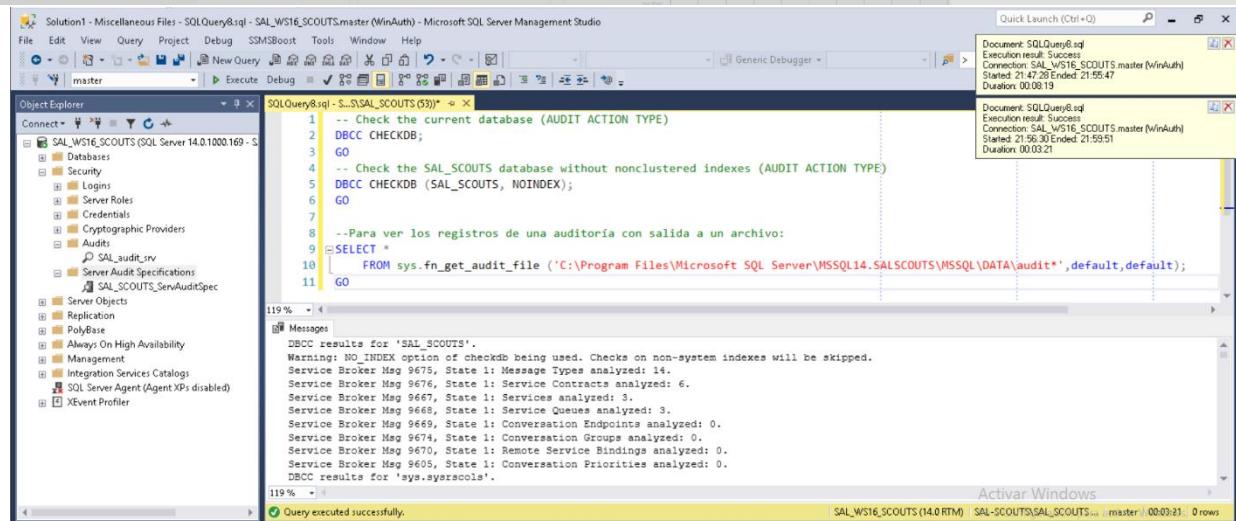
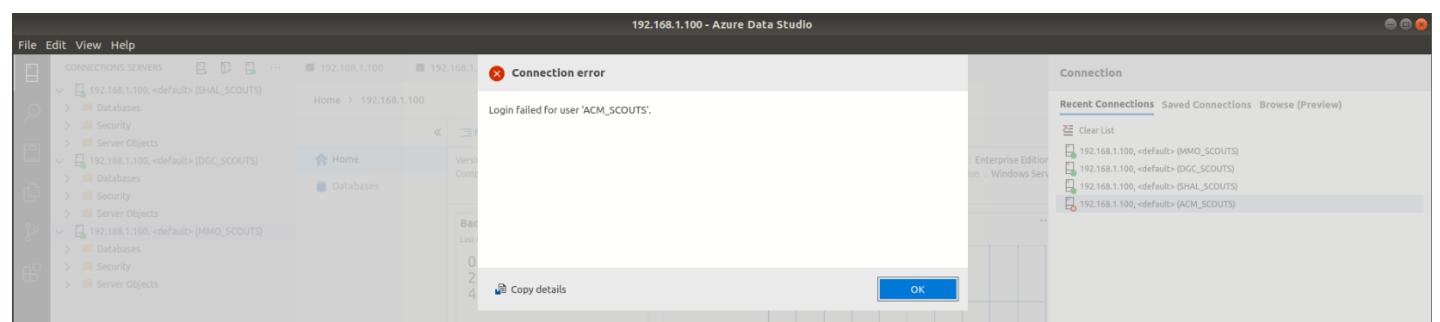
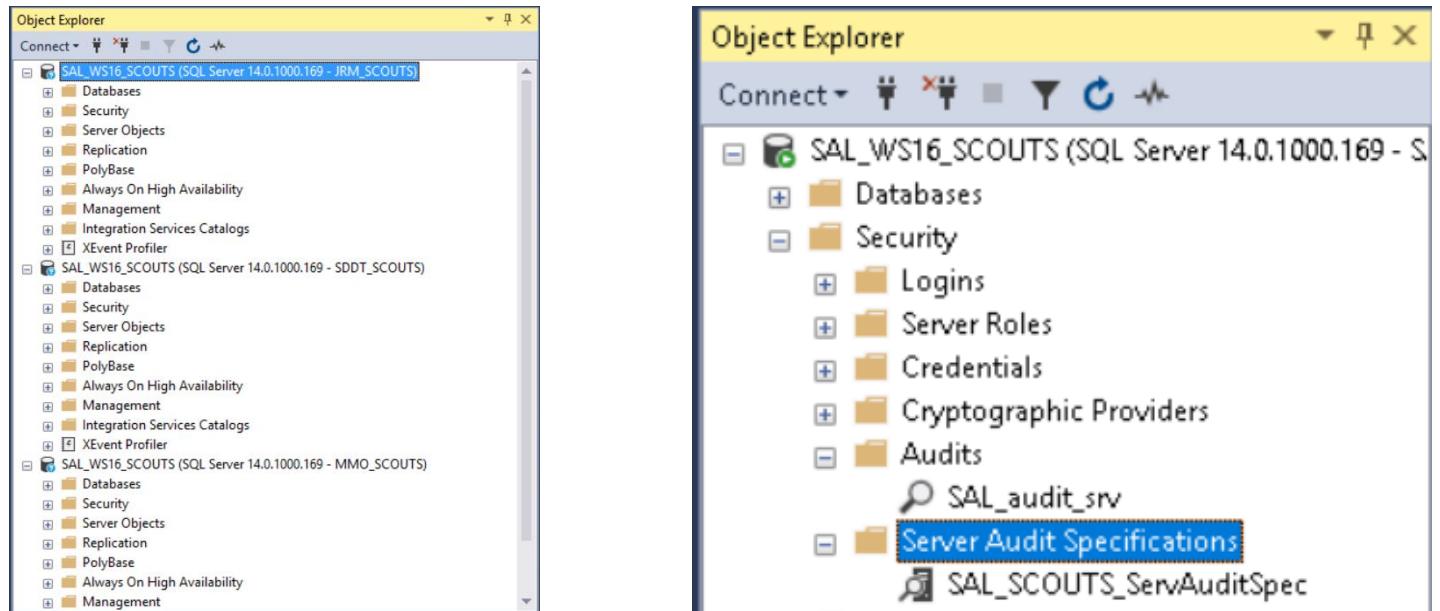


Nos aparecerá un asistente y nos pedirá tres cosas a especificar: el nombre de la especificación de auditoría de servidor (es opcional ya que por defecto se le asignará otro nombre), la auditoría de servidor (que define el objeto al que los eventos seleccionados deberían ser registrados) y el tipo de acción de auditoría (son los eventos que deberían ser auditados):



Como ejemplo usaremos 3 tipos de acción de auditoría: **SUCCESSFUL\_LOGIN\_GROUP** (indica que una entidad de seguridad ha iniciado una sesión de **SQL Server**), **FAILED\_LOGIN\_GROUP** (indica que una entidad de seguridad intentó iniciar una sesión de **SQL Server**, pero no lo consiguió) y **DBCC\_GROUP** (indica cuando una entidad de seguridad emite un comando **DBCC – DataBase Check Command**).

Creada y activada la especificación de auditoría de servidor, vamos a provocar los eventos que hemos indicado anteriormente. Para los *logins* usaremos los equipos clientes iniciando sesión con los usuarios del personal y, más tarde, ejecutaremos sentencias con **DBCC**.



Provocados los eventos podemos verlos haciendo *click derecho* en nuestra auditoría > *View Audit Logs*:

Event Time	Server Instance Name	Action ID	Class Type	Sequence Number	Succeeded	Permission Bit Mask	Column Permission	Session ID	Server Principal ID	Database
23/05/2021 20:47:06	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	51	260	0
23/05/2021 20:44:39	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:44:39	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:44:38	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:44:16	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:44:13	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:42:22	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:42:21	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:40:16	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:40:16	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:40:15	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	58	260	0
23/05/2021 20:38:10	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:38:09	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:38:00	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:38:00	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:38:00	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	57	260	0
23/05/2021 20:37:34	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	56	260	0
23/05/2021 20:36:30	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	55	260	0
23/05/2021 20:36:29	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	54	260	0
23/05/2021 20:36:29	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	51	260	0
23/05/2021 20:34:46	SAL_WS16_SCOUTS	LOGIN_SUCCEEDED	LOGIN	1	True	0x00000000000000000000000000000000	False	51	260	0

O podemos ejecutar la siguiente sentencia:

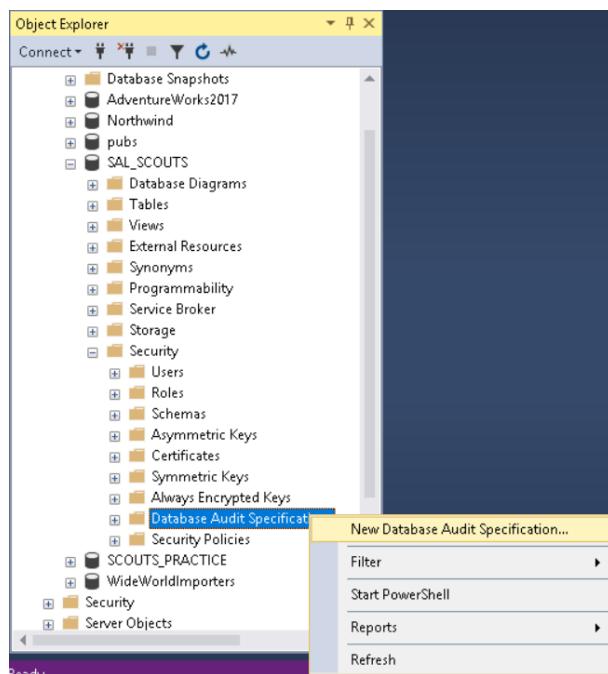
```
SELECT event_time, action_id, succeeded, database_principal_id, class_type,
session_server_principal_name, server_principal_name, server_instance_name,
database_name, statement, file_name, audit_file_offset, client_ip,
application_name
FROM sys.fn_get_audit_file ('C:\Program Files\Microsoft SQL Server\MSSQL14.SALSCOUTS\MSSQL\DATA\audit\*', default, default);
GO
```

event_time	action_id	succeeded	class_type	session_server_principal_name	server_principal_name	server_instance_name	database_name	statement	file_name
2021-05-23 17:53:18.8879621	AUSC	1	A	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_WS16_SCOUTS	C:\Program Files\Microsoft SQL Server\MSSQL14	--Para ver los registros de una auditoría con salida a un archivo:	
2021-05-23 19:38:13.6201927	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:38:14.2448002	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:43:16.1521508	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:43:16.7142311	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:47:13.4496092	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:47:13.5592257	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:47:14.8084679	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	
2021-05-23 19:47:15.8879620	LGIS	1	LX	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL-SCOUTS\SAL_WS16_SCOUTS\$	SAL_WS16_SCOUTS		-- network protocol: LPC set quoted_iden	

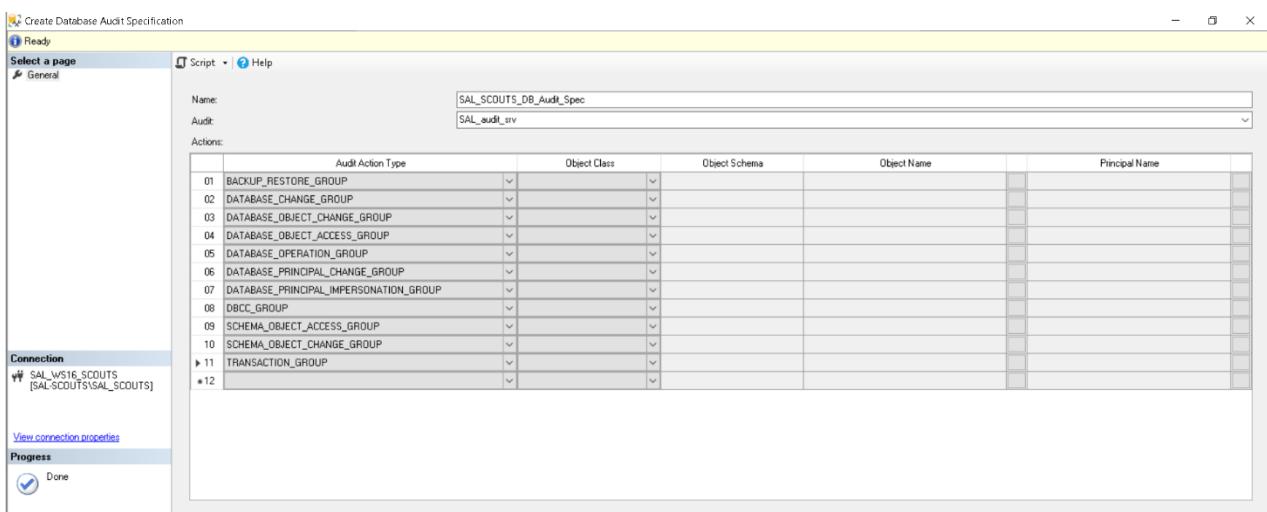
### • 3.2. Especificación de auditoría de BD

La **especificación de auditoría de base de datos** recopila acciones de auditoría de nivel de base de datos. Puede agrupar grupos de acciones de auditoría o eventos de auditoría a una especificación de auditoría de base de datos. Los eventos de auditoría son las acciones atómicas (**CREATE, DROP, UPDATE** – crear, eliminar y modificar/actualizar) que puede auditar el motor de SQL Server. Los grupos de acciones de auditoría son grupos predefinidos de acciones. Ambos están en el ámbito de la base de datos de SQL Server. Estas acciones envían a la auditoría, que las registra en el destino.

Ahora vamos a crear una auditoría a nuestra base de datos, para crearla nos vamos a *Security > click derecho en Database Audit Specification > New Database Audit Specification...:*



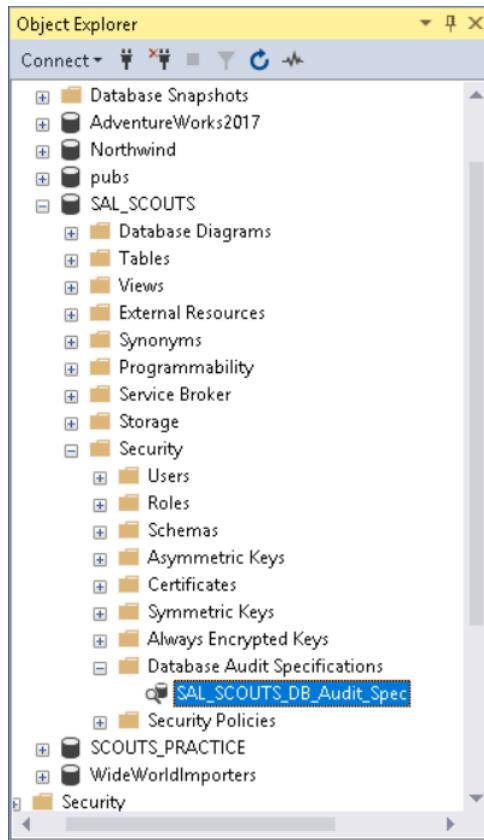
Nos aparecerá el asistente y nos pedirá las mismas cosas que en la anterior especificación:



Como ejemplo seleccionamos 11 tipos de acción:

- **BACKUP\_RESTORE\_GROUP**: este evento tiene lugar cuando se ejecuta un comando de copia de seguridad o de restauración.
- **DATABASE\_CHANGE\_GROUP**: este evento se desencadena al crear, modificar o quitar una base de datos cualquiera.
- **DATABASE\_OBJECT\_CHANGE\_GROUP**: este evento se desencadena al ejecutar una instrucción **CREATE**, **ALTER** o **DROP** en objetos de base de datos (esto puede crear bastantes registros de auditoría).
- **DATABASE\_OBJECT\_ACCESS\_GROUP**: este evento se desencadena para cualquier tipo de acceso a cualquier base de datos (esto puede generar registros de gran tamaño).
- **DATABASE\_OPERATION\_GROUP**: este evento tiene lugar en cualquier operación de base de datos y en cualquier base de datos.
- **DATABASE\_PRINCIPAL\_CHANGE\_GROUP**: este evento se provoca al crear, modificar o quitar entidades de seguridad, como usuarios, de una base de datos.
- **DATABASE\_PRINCIPAL\_IMPERSONATION\_GROUP**: este evento se desencadena cuando hay una suplantación (cuando se realiza una impersonación) en el ámbito de la base de datos.
- **DBCC\_GROUP**: este evento se desencadena cuando una entidad de seguridad emite un comando **DBCC**.
- **SCHEMA\_OBJECT\_ACCESS\_GROUP**: este evento se desencadena al usar un permiso de objeto en el esquema.
- **SCHEMA\_OBJECT\_CHANGE\_GROUP**: este evento se desencadena al realizar una operación **CREATE**, **ALTER** o **DROP** en un esquema.
- **TRANSACTION\_GROUP**: este evento se desencadena para las operaciones de **BEGIN TRANSACTION**, **ROLLBACK TRANSACTION** y **COMMIT TRANSACTION** (transacciones explícitas), tanto en llamada explícitas a esas instrucciones como en operaciones de transacciones implícitas (**ALTER TABLE**, **FETCH**, **REVOKE**, **CREATE**, **GRANT**, **SELECT**, **DELETE**, **INSERT**, **TRUNCATE TABLE**, **DROP**, **OPEN** y **UPDATE**).

Creada y activada la **especificación de auditoría de base de datos** vamos a ejecutar varias sentencias para provocar los eventos de auditoría. Serán provocados desde todos los equipos del dominio tanto equipo servidor como equipos clientes usando todos los usuarios del dominio para crear una diversidad de registros.



Aquí se muestra la auditoría...:

~ ...de las copias de seguridad:

SQLQuery2.sql - S..\\SAL\_SCOUTS (51)\* SQLQuery1.sql - S..\\SAL\_SCOUTS (55)\*

```

10 GO
11 =BACKUP CERTIFICATE SAL_TDEscout
12 TO FILE = 'C:\backup\bkp_TDE\SAL_TDEscout.cer'
13 WITH PRIVATE KEY (
14     FILE = 'C:\backup\bkp_TDE\SAL_TDEkey.pvk',
15     ENCRYPTION BY PASSWORD = 'Abc1234';
16 );
17 GO
18 =BACKUP DATABASE SAL_SCOUTS
19     TO DISK = 'C:\backup\bkp_TDE\SAL_SCOUTS_Full_TDE.bak'
20     WITH ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE = SAL_TDEscout)
21 GO
22 =BACKUP LOG SAL_SCOUTS
23     TO DISK = 'C:\backup\bkp_TDE\SAL_SCOUTS_log_TDE.bak'
24     WITH ENCRYPTION (ALGORITHM = AES_256, SERVER CERTIFICATE = SAL_TDEscout);
25 GO
26
27 =SELECT event_time, action_id, class_type, session_server_principal_name, server_principal_name, server_instance_name, database_name, statement, client_ip, application_name
28     FROM sys.fn_get_audit_file ('C:\Program Files\Microsoft SQL Server\140\SQL14\SALSCOUTS\MSQL\DATA\audit\*.sqlaudit', default,default)
29     WHERE event_time > '2021-05-24' AND server_principal_name = 'SAL-SCOUTS\SAL_SCOUTS' ORDER BY event_time DESC;

```

Results Messages

event_time	action_id	class_type	session_server_principal_name	server_principal_name	server_instance_name	database_name	statement	client_ip	application_name
2021-05-24 11:51:27.8580018	SL	V	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	SELECT param.is_readonly AS [IsRea...	local machine	Microsoft SQL Server Management Studio - Transact-SQL IntelliSense
2021-05-24 11:51:26.3427004	LGIS	LX	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS		-- network protocol LPC set quoted...	local machine	Microsoft SQL Server Management Studio - Transact-SQL IntelliSense
2021-05-24 11:45:41.5050783	BAL	DB	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	BACKUP LOG SAL_SCOUTS TO DI...	local machine	Microsoft SQL Server Management Studio - Query
2021-05-24 11:45:38.2793204	BA	DB	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	BACKUP DATABASE SAL_SCOUTS ...	local machine	Microsoft SQL Server Management Studio - Query
2021-05-24 11:45:36.1703173	BA	CR	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	BACKUP CERTIFICATE SAL_TDEsc...	local machine	Microsoft SQL Server Management Studio - Query
2021-05-24 11:45:36.1703173	TXCM	DB	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	BACKUP CERTIFICATE SAL_TDEsc...	local machine	Microsoft SQL Server Management Studio - Query
2021-05-24 11:45:36.1389131	TXBG	DB	SAL-SCOUTS\SAL_SCOUTS	SAL-SCOUTS\SAL_SCOUTS	SAL_W516_SCOUTS	SAL_SCOUTS	BACKUP CERTIFICATE SAL_TDEsc...	local machine	Microsoft SQL Server Management Studio - Query

81 %

Query executed successfully. | SAL\_W516\_SCOUTS (14.0 RTM) | SAL-SCOUTS\SAL\_SCOUTS ... | SAL\_SCOUTS | 00:00:00 | 383 rows

~ ...modificación de nuestra base de datos:

~ ...modificación y consulta de tablas:

```
SQLQuery2.sql - S...S\$\$AL_SCOUTS (51)* SQLQuery1.sql - S...S\$\$AL_SCOUTS (55)* x

41 USE $AL_SCOUTS
42 GO
43
44
45 UPDATE $AL_PERSONAL
46 SET direccion = 'C\ Mi Calle 1'
47 WHERE ID_personal = 2;
48 GO
49
50 SELECT * FROM $AL_PERSONAL
51 GO
52
53 SELECT event_time, action_id, class_type, session_server_principal_name, server_principal_name, server_instance_name, database_name, statement, client_ip, application_name
54 FROM sys.fn_get_audit_file ('C:\Program Files\Microsoft SQL Server\150\SSIS\LOGS\audit*.sqlaudit', default, default)
55 WHERE event_time > '2021-05-24' AND server_principal_name = '$AL-SCOUTS\\$AL_SCOUTS' ORDER BY event_time DESC;
56 GO
57
```

~ ...Etc. :

```
SQLQuery2.sql - S...$SAL_SCOUTS (51)* SQLQuery1.sql - S...$SAL_SCOUTS (55)* > x
50 SELECT * FROM sys.dm_exec_sessions
51 GO
52
53 --SELECT event_time, action_id, class_type, session_server_principal_name, server_principal_name, server_instance_name, database_name,statement, client_ip, application_name
54 --FROM sys.fn_get_audit_file ('C:\Program Files\Microsoft SQL Server\150\MSSQL14.SALSCOUTS\MSSQL\DATA\audit%',.saudit%',default,default)
55 ORDER BY event_time DESC;
56 GO
57
```

89 %

	event_time	action_id	class_type	session_server_principal_name	server_principal_name	server_instance_name	database_name	statement	client_ip	application_name
1	2021-05-24 15:17:39.7759869	TXBG	DB		sa	SAL_WS16_SCOUTS	SAL_SCOUTS		Unknown	
2	2021-05-24 15:17:39.7759869	TXCM	DB		sa	SAL_WS16_SCOUTS	SAL_SCOUTS		Unknown	
3	2021-05-24 15:17:32.2286103	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
4	2021-05-24 15:17:32.2286103	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
5	2021-05-24 15:17:32.2286103	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
6	2021-05-24 15:17:32.2286103	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
7	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
8	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
9	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
10	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
11	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
12	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
13	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService
14	2021-05-24 15:17:32.1972414	SL	V	ACM_SCOUTS	ACM_SCOUTS	SAL_WS16_SCOUTS	SAL_SCOUTS	SELECT SCHEMA_NAME(udl.schema_id) AS [Schema]..	192.168.1.3	azdata-languageService

### • 3.3. Bonus Auditoría SQL Server

Como bonus de auditoría en **SQL Server** se ha planteado la siguiente situación:

- Se ha votado entre los miembros de la asociación en crear un canal anónimo de denuncias que posibilita una comunicación anónima y segura de las denuncias que se presenten. El objetivo de este canal es el de cualquier persona o entidad pueda denunciar y/o comunicar cualquier posible conducta irregular, poco ética o inapropiada de cualquier persona que esté empleada, represente o participe de alguna forma en la asociación.
- Animando a cualquier persona o entidad a comunicarse con la asociación y a hacerle llegar toda sugerencia, denuncia o cualquier indicio o prueba que puede ser de utilidad en los procesos de investigación o estudio que se llegue a abrir.
- Los tipos de datos de los que se disponga (fechas, nombres, lugares, etc.) que puedan ayudar en un posible proceso de investigación interno, a fin de tener las máximas evidencias posibles.
- Se creará una tabla temporal. La tabla historial nos servirá de auditoría interna para auditar y procesar las denuncias teniendo en cuenta quién la recibió, en qué hora y fecha se realizó, saber en qué estado se encuentra la denuncia, etc. y se crearán otros aspectos en los apartados anteriores como repaso.

La estructura de la tabla será ejecutando la siguiente sentencia:

```
USE SAL_SCOUTS
GO

SELECT d.compatibility_level
  FROM sys.databases d
 WHERE d.name = DB_NAME();
GO
-- compatibility_level
-- 140

--ALTER DATABASE SAL_SCOUTS_JR
-- SET COMPATIBILITY_LEVEL = 140;
--GO

ALTER DATABASE SAL_SCOUTS
  SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT = ON;
GO
-- Commands completed successfully

-- ALTER DATABASE SAL_SCOUTS
-- ADD FILEGROUP SAL_SCOUTS_oltp
-- CONTAINS MEMORY_OPTIMIZED_DATA;
-- GO
-- NUESTRA BASE DE DATOS YA ESTUVO OPTIMIZADA EN LA 1a PARTE DEL PROYECTO
```

```

ALTER DATABASE SAL_SCOUTS
ADD FILE (
NAME = 'SAL_SCOUTS_oltp',
FILENAME          = 'C:\Program Files\Microsoft SQL Server\MSSQL14.SALSCOUTS\MSSQL\DATA\SAL_OLTP'
) TO FILEGROUP SAL_SCOUTS_oltp;
GO

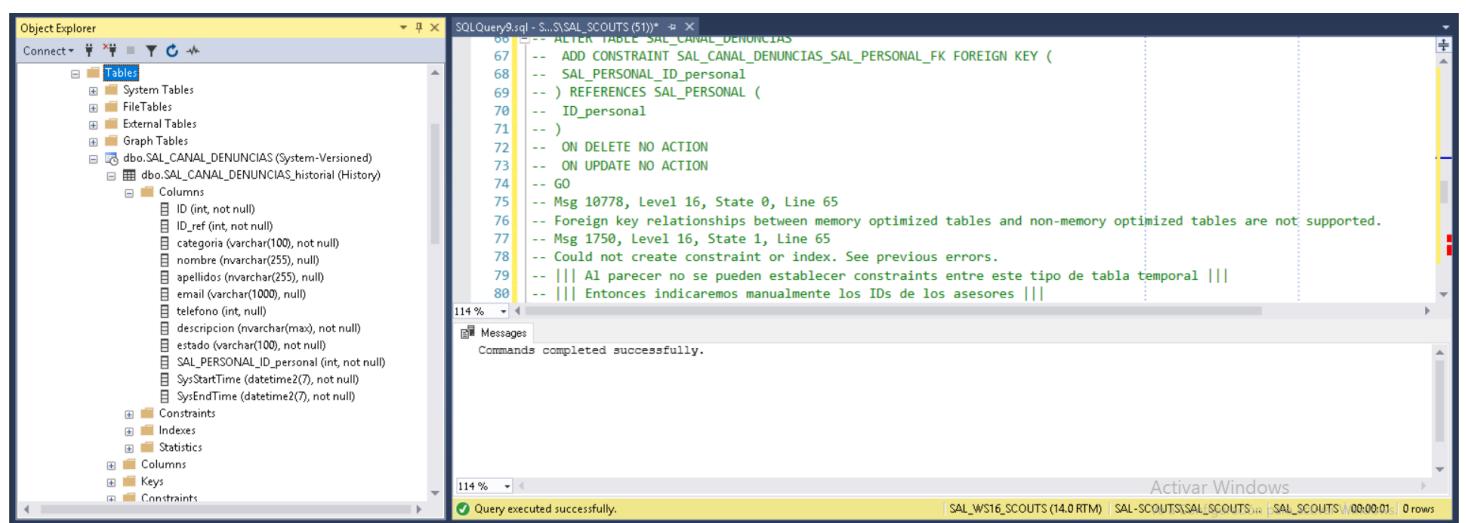
-- Controlamos la existencia de la tabla temporal, primero desactivando el sistema de versiones
-- y después eliminando la tabla
BEGIN
    IF ((SELECT temporal_type FROM SYS.TABLES WHERE object_id = OBJECT_ID('dbo.SAL_CANAL_DENUNCIAS', 'U')) = 2)
        BEGIN
            ALTER TABLE [dbo].[SAL_CANAL_DENUNCIAS] SET (SYSTEM_VERSIONING = OFF)
        END
    DROP TABLE IF EXISTS [dbo].[SAL_CANAL_DENUNCIAS];
    DROP TABLE IF EXISTS [dbo].[SAL_CANAL_DENUNCIAS_historial];
END
GO

CREATE TABLE [dbo].[SAL_CANAL_DENUNCIAS] (
    ID int IDENTITY(1,1) NOT NULL PRIMARY KEY NONCLUSTERED,
    ID_ref int NOT NULL,
    categoria VARCHAR(100) NOT NULL,
    nombre NVARCHAR(255) NULL,
    apellidos NVARCHAR(255) NULL,
    email VARCHAR(1000) NULL,
    telefono INT NULL,
    descripcion NVARCHAR(max) NOT NULL,
    SAL_PERSONAL_ID_personal INT NOT NULL,
    SysStartTime datetime2(7) GENERATED ALWAYS AS ROW START NOT NULL,
    SysEndTime datetime2(7) GENERATED ALWAYS AS ROW END NOT NULL,
    PERIOD FOR SYSTEM_TIME (SysStartTime,SysEndTime),
) WITH (
MEMORY_OPTIMIZED = ON,
DURABILITY = SCHEMA_AND_DATA,
SYSTEM_VERSIONING = ON (HISTORY_TABLE = [dbo].[SAL_CANAL_DENUNCIAS_historial])
);
GO

-- ALTER TABLE SAL_CANAL_DENUNCIAS
--     ADD CONSTRAINT SAL_CANAL_DENUNCIAS_SAL_PERSONAL_FK FOREIGN KEY (
--         SAL_PERSONAL_ID_personal
--     ) REFERENCES SAL_PERSONAL (
--         ID_personal
--     )
--     ON DELETE NO ACTION

```

```
-- ON UPDATE NO ACTION
-- GO
-- Msg 10778, Level 16, State 0, Line 65
-- Foreign key relationships between memory optimized tables and non-memory
optimized tables are not supported.
-- Msg 1750, Level 16, State 1, Line 65
-- Could not create constraint or index. See previous errors.
-- !!! Al parecer no se pueden establecer constraints entre este tipo de
tabla temporal !!!
-- !!! Entonces indicaremos manualmente los IDs de los asesores !!!
```



```
-- Creamos los perfiles de los asesores
EXEC SALnewPerson 2,10,'Juan Jose','Pau','Pernas','220133697H','1970-07-29',699840123,'Avda. Platanal' 90 3°
Izq.',51,'C:\SAL_imgs\sal.jpg',15,15,0,0;
UPDATE SAL_PERSONAL SET scout_user = 'JJPP_SCOUTS' WHERE ID_personal = 10;
CREATE LOGIN [JJPP_SCOUTS] WITH PASSWORD='Abcd1234.', DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english];
CREATE USER [JJPP_SCOUTS] FOR LOGIN [JJPP_SCOUTS] WITH DEFAULT_SCHEMA=[dbo];
EXEC SALnewPerson 2,11,'María','Martínez','Pérez','331244708I','1971-08-30',600951234,'Avda. do Pedral' 60 1°
Izq.',50,'C:\SAL_imgs\sal.jpg',10,10,0,0;
UPDATE SAL_PERSONAL SET scout_user = 'MMP_SCOUTS' WHERE ID_personal = 11;
CREATE LOGIN [MMP_SCOUTS] WITH PASSWORD='Abcd1234.', DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english];
CREATE USER [MMP_SCOUTS] FOR LOGIN [MMP_SCOUTS] WITH DEFAULT_SCHEMA=[dbo];
EXEC SALnewPerson 2,12,'Valeria','Suárez','Pérez','996652147I','1981-11-30',666998544,'C/ Negra 5 7° Dcha.' ,41,'C:\SAL_imgs\sal.jpg',10,10,0,0;
UPDATE SAL_PERSONAL SET scout_user = 'VSP_SCOUTS' WHERE ID_personal = 12;
CREATE LOGIN [VSP_SCOUTS] WITH PASSWORD='Abcd1234.', DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english];
CREATE USER [VSP_SCOUTS] FOR LOGIN [VSP_SCOUTS] WITH DEFAULT_SCHEMA=[dbo];
GO
```

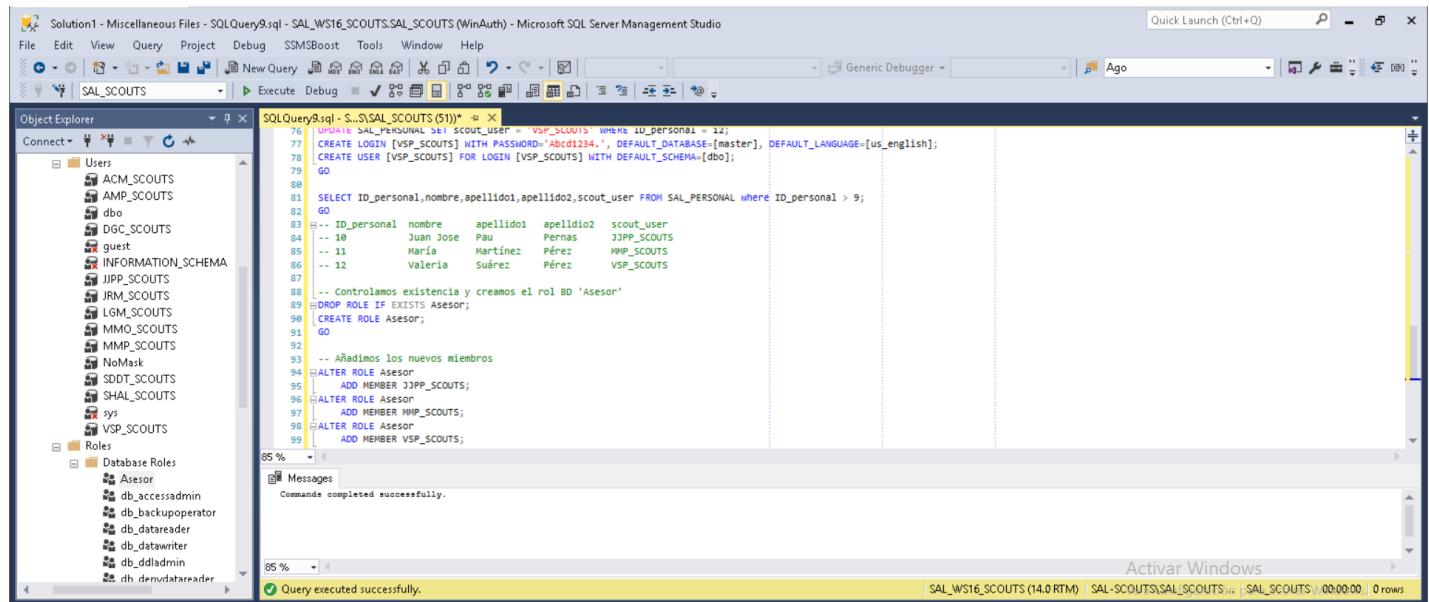
```

SELECT ID_personal,nombre,apellido1,apellido2,scout_user FROM SAL_PERSONAL
where ID_personal > 9;
GO
-- ID_personal      nombre          apellido1 apellido2 scout_user
-- 10                Juan Jose     Pau        Pernas    JJPP_SCOUTS
-- 11                María          Martínez   Pérez    MMP_SCOUTS
-- 12                Valeria       Suárez    Pérez    VSP_SCOUTS

-- Controlamos existencia y creamos el rol BD 'Asesor'
DROP ROLE IF EXISTS Asesor;
CREATE ROLE Asesor;
GO

-- Añadimos los nuevos miembros
ALTER ROLE Asesor
    ADD MEMBER JJPP_SCOUTS;
ALTER ROLE Asesor
    ADD MEMBER MMP_SCOUTS;
ALTER ROLE Asesor
    ADD MEMBER VSP_SCOUTS;
GO

```



```
-- Insertamos datos en la tabla y consultamos
SELECT * FROM SAL_CANAL_DENUNCIAS;
SELECT * FROM SAL_CANAL_DENUNCIAS_historial;
GO
```

Solution1 - Miscellaneous Files - bonus\_auditoria.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio

File Edit View Query Project Debug SSMSBoost Tools Window Help

Generic Debugger Ago

bonus\_auditoria.s...\\SAL\_SCOUTS (54)\* x

```
102
103 -- Insertamos datos en la tabla y consultamos
104 SELECT * FROM SAL_CANAL_DENUNCIAS;
105 [ SELECT * FROM SAL_CANAL_DENUNCIAS_historial;
106 GO
107
108
```

137 %

Results Messages

ID	ID_ref	categoria	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	SysStartTime	SysEndTime	
1	1390	Grupos scouts	Abusos	-	Fairley	vestibulum@nulla.org	696624783	Texto de ejemplo	Activa	11	2021-05-24 22:23:55.5175072	9999-12-31 23:59:59.9999999	
2	1392	2391	Trabajadores/as	Abusos	-	ultrices.duis@acuvestibulum.co.uk	626978981	Texto de ejemplo	Cerrada	10	2021-05-24 22:23:55.5331133	9999-12-31 23:59:59.9999999	
3	1396	2395	Voluntarios/as	Otro	Vivien	Hunt	dictum@integervitaeinlib.net	698895839	Texto de ejemplo	Activa	11	2021-05-24 22:23:55.5331133	9999-12-31 23:59:59.9999999
4	1414	2413	Trabajadores/as	Otro	-	aliquam.fringilla@fermentumrisusat.com	633887994	Texto de ejemplo	Activa	10	2021-05-24 22:23:55.5331133	9999-12-31 23:59:59.9999999	
5	1421	2420	Voluntarios/as	Discriminacion	Nolan	-	lobortis.class@pede.com	659279887	Texto de ejemplo	Activa	11	2021-05-24 22:23:55.5331133	9999-12-31 23:59:59.9999999
6	1431	2430	Familias	Otro	-	Allen	curabitur.sed@fringilla.org	683625867	Texto de ejemplo	Activa	10	2021-05-24 22:23:55.5487662	9999-12-31 23:59:59.9999999
7	1433	2432	Trabajadores/as	Abusos	-	urna.nec@nonapienmolestie.ca	661972954	Texto de ejemplo	Cerrada	10	2021-05-24 22:23:55.5487662	9999-12-31 23:59:59.9999999	

ID	ID_ref	categoria	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	SysStartTime	SysEndTime	
1	2	1001	Trabajadores/as	Incumplimiento del cod. etico	Xanthus	Gay	odio.tristique@nequevenenatis.edu	651623493	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.4575543	2021-05-24 22:26:21.5615337
2	8	1007	Familias	Otro	-	Keller	maius.vel@impunson.co.uk	618367465	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.4723393	2021-05-24 22:26:21.5615337
3	14	1013	Trabajadores/as	Incumplimiento de la norma...	-	Mcclain	sagittis@inciduntorciquis.net	696267453	Texto de ejemplo	Eliminada	10	2021-05-24 22:21:13.5040956	2021-05-24 22:26:21.5615337
4	15	1014	Trabajadores/as	Incumplimiento del cod. etico	Eleanor	-	hinduism@neicitellus.net	624638376	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.5040956	2021-05-24 22:26:21.5615337
5	19	1018	Trabajadores/as	Otro	Nyssa	Robles	aliquam@vellacuibus.net	692857936	Texto de ejemplo	Eliminada	10	2021-05-24 22:21:13.5199691	2021-05-24 22:26:21.5615337
6	25	1024	Familias	Incumplimiento del cod. etico	Pearson	elementum@semivit.ca	695852948	Texto de ejemplo	Eliminada	10	2021-05-24 22:21:13.5956504	2021-05-24 22:26:21.5615337	

✓ Query executed successfully.

SAL\_WS16\_SCOUTS (14.0 RTM) - SAL-SCOUTS\Sal-SCOUTS... - SAL\_SCOUTS : 00:00:00 | 1500 rows

-- Creamos schema CONSULTANT

```
DROP SCHEMA IF EXISTS CONSULTANT  
GO
```

## CREATE SCHEMA CONSULTANT

GO

```
SELECT cd.ID_ref,cd.categoría, cd.tipo, cd.nombre, cd.apellidos, cd.email,  
cd.telefono, cd.descripcion, cd.estado, cd.SAL_PERSONAL_ID_personal,  
p.scout_user AS scout_user  
    FROM SAL_CANAL_DENUNCIAS cd  
    JOIN SAL_PERSONAL p ON p.ID_personal = cd.SAL_PERSONAL_ID_personal;  
GO
```

-- Creamos vista CONSULTANT.REPORT

```
CREATE VIEW CONSULTANT.REPORT WITH SCHEMABINDING  
AS
```

```
SELECT cd.ID_ref, cd.categoría, cd.tipo, cd.nombre, cd.apellidos, cd.email,
cd.telefono, cd.descripcion, cd.estado, cd.SAL_PERSONAL_ID_personal,
p.scout_user AS scout_user
    FROM dbo.SAL_CANAL_DENUNCIAS cd
    JOIN dbo.SAL_PERSONAL p ON p.ID_personal = cd.SAL_PERSONAL_ID_personal;
GO
```

-- Controlamos la existencia de la función

```
DROP FUNCTION IF EXISTS CONSULTANT.SALfnREPORT
```

GO

```

CREATE OR ALTER FUNCTION CONSULTANT.SALfnREPORT (@scout_user SYSNAME)
    RETURNS TABLE
    WITH SCHEMABINDING
AS
    RETURN SELECT 1 AS consultant_filter
        WHERE @scout_user = USER_NAME()
        OR (USER_NAME() IN ('dbo', 'dbo'));
GO

-- Controlamos la existencia de la directiva de seguridad
DROP SECURITY POLICY IF EXISTS CONSULTANT.SAL_SPOscouts_report
GO

```

```

CREATE SECURITY POLICY CONSULTANT.SAL_SPOscouts_report
    ADD FILTER PREDICATE CONSULTANT.SALfnREPORT(scout_user)
    ON CONSULTANT.REPORT
    WITH (STATE = ON);
GO

```

GO

```

Solution1 - Miscellaneous Files - bonus_auditoria.sql - SAL_WS16_SCOUTS.SAL_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio
File Edit View Query Project Debug SSMSBoost Tools Window Help
SAL_SCOUTS Execute Debug
137 WHERE @scout_user = USER_NAME()
138     OR (USER_NAME() IN ('dbo', 'dbo'));
139 GO
140
141 -- Controlamos la existencia de la directiva de seguridad
142 DROP SECURITY POLICY IF EXISTS CONSULTANT.SAL_SPOscouts_report
143 GO
144
145 CREATE SECURITY POLICY CONSULTANT.SAL_SPOscouts_report
146     ADD FILTER PREDICATE CONSULTANT.SALfnREPORT(scout_user)
147     ON CONSULTANT.REPORT
148     WITH (STATE = ON);
149 GO
150
151 Messages
Commands completed successfully.

```

Query executed successfully.

```

Solution1 - Miscellaneous Files - bonus_auditoria.sql - SAL_WS16_SCOUTS.SAL_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio
File Edit View Query Project Debug SSMSBoost Tools Window Help
SAL_SCOUTS Execute Debug
168
169 SELECT * FROM CONSULTANT.REPORT;
170 SELECT * FROM SAL_CANAL_DENUNCIAS_historial;
171 GO

```

Results

ID_ref	categoria	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	scout_user
1	2391	Trabajadores/as	Abusos	-	ultrices.duis@arcuvestibulum.co.uk	628670881	Texto de ejemplo	Cerrada	10	JPP_SCOUTS
2	2413	Trabajadores/as	Otro	-	aliquam.fringilla@ermentumrisusat.com	633887894	Texto de ejemplo	Activa	10	JPP_SCOUTS
3	2430	Familias	Otro	Allen	curabitur.sed@fringilla.org	683625987	Texto de ejemplo	Activa	10	JPP_SCOUTS
4	2432	Trabajadores/as	Abusos	-	uma.nec@nonsapienmolestie.ca	661972954	Texto de ejemplo	Cerrada	10	JPP_SCOUTS
5	2445	Otros	Discriminacion	Barclay	risus.donec.nbr@diam.ca	615474263	Texto de ejemplo	Cerrada	10	JPP_SCOUTS
6	2499	Familias	Otro	-	uma.er@onarelectusante.net	65392479	Texto de ejemplo	Cerrada	10	JPP_SCOUTS
7	2089	Grupos scouts	Agresiones y amenazas	-	orci.phasellus.dapibus@ dolor.com	676785946	Texto de ejemplo	Activa	10	JPP_SCOUTS
8	2105	Educandos/as	Agresiones y amenazas	Cochran	lacinia.al.iaculis@nunc.org	67754664	Texto de ejemplo	Cerrada	10	JPP_SCOUTS

ID	ID_ref	categoria	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	SysStartTime	SysEndTime	
1	2	1001	Trabajadores/as	Incumplimiento del cod. etico	Xanthus	Gay	odio.tristique@nequevenenatis.edu	651623493	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.4575543	2021-05-24 22:26:21.5615337
2	8	1007	Familias	Otro	Keller	mauris.vel@tempusnon.co.uk	618367466	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.4723399	2021-05-24 22:26:21.5615337	
3	14	1013	Trabajadores/as	Incumplimiento de la normativa interna y/o la Ley	McClain	sagittis@linciduntorci quis.net	696267453	Texto de ejemplo	Eliminada	10	2021-05-24 22:21:13.5040956	2021-05-24 22:26:21.5615337	
4	15	1014	Trabajadores/as	Incumplimiento del cod. etico	Eleanor	-	tincidunt@necetelus.net	624638376	Texto de ejemplo	Eliminada	11	2021-05-24 22:21:13.5040956	2021-05-24 22:26:21.5615337
5	19	1018	Trabajadores/as	Otro	Nyssa	Robles	aliquam@velfaucibus.net	692857936	Texto de ejemplo	Eliminada	10	2021-05-24 22:21:13.5198937	2021-05-24 22:26:21.5615337

Query executed successfully.

Como podemos observar hemos creado un nuevo departamento en la asociación llamado Asesor el cual se encargará del canal de denuncias. Han llegado 3 nuevos miembros: **Juan José (JJPP\_SCOUTS)**, **María (MMP\_SCOUTS)** y **Valeria (VSP\_SCOUTS)**. Se les ha creado una directiva similar a la de los **Tesoreros** para llevar un seguimiento de quién se encarga de las denuncias respectivamente e identificándolas mediante el ID del personal que a su vez se le ha creado una vista que muestra el usuario de la asociación para una mejor identificación. Comentado anteriormente y observado en las capturas de pantalla, se trata de una tabla temporal, la cual con su tabla historial se puede llevar una pequeña auditoría interna sobre el estado de la denuncia, pero una de sus características principales es que está optimizada para la memoria. Esto nos permite mejorar en la carga de datos y poder en cuenta el período de tiempo de tales denuncias. La única inconveniencia existente es que, entre las especificaciones de auditoría de servidor con la de base de datos y la optimización de memoria de la tabla, pues no alcanza lo suficiente la memoria RAM (el equipo servidor solamente dispone de **1,5 GB** de RAM) y tenemos que desactivar la auditoría. Aún así el único remedio sería aumentarle un poco más la memoria de trabajo y no habría más problemas a la hora de realizar transacciones. En el caso de no querer tener la característica de optimización para la memoria y/o que fuera una tabla normal sería así su estructura:

```
DROP TABLE IF EXISTS [CONSULTANT].[SAL_CANAL_DENUNCIAS]
GO
CREATE TABLE [CONSULTANT].[SAL_CANAL_DENUNCIAS] (
    ID int NOT NULL PRIMARY KEY,
    ID_ref int NOT NULL,
    categoria VARCHAR(100) NOT NULL,
    tipo VARCHAR(100) NOT NULL,
    nombre NVARCHAR(255) NULL,
    apellidos NVARCHAR(255) NULL,
    email VARCHAR(1000) NULL,
    telefono INT NULL,
    descripcion NVARCHAR(max) NOT NULL,
    estado VARCHAR(100) NOT NULL,
    SAL_PERSONAL_ID_personal INT NOT NULL,
    fecha_inicio DATETIME2 NOT NULL DEFAULT GETDATE(),
    fecha_final DATETIME2 NULL
);
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'CONSULTANT' database, there is a table named 'SAL\_CANAL\_DENUNCIAS'. The table has the following structure:

```

CREATE TABLE [CONSULTANT].[SAL_CANAL_DENUNCIAS] (
    ID int NOT NULL PRIMARY KEY,
    ID_ref int NOT NULL,
    categoria VARCHAR(100) NOT NULL,
    tipo VARCHAR(100) NOT NULL,
    nombre NVARCHAR(255) NULL,
    apellidos NVARCHAR(255) NULL,
    email VARCHAR(1000) NULL,
    telefono INT NULL,
    descripcion NVARCHAR(max) NOT NULL,
    estado VARCHAR(100) NOT NULL,
    SAL_PERSONAL_ID_personal INT NOT NULL,
    fecha_inicio DATETIME2 NOT NULL DEFAULT GETDATE(),
    fecha_final DATETIME2 NULL
);

```

The 'Messages' pane at the bottom right indicates that the command was completed successfully.

Y la auditoría que podríamos realizar sería la de crear un trigger que nos permita apartar la fila que se encuentre en estado "En espera de ser procesada", básicamente sería apartarla de la tabla principal, copiarla la tabla de "denuncias en espera" y eliminarla de allí, pero si vuelve a estar activa, introducirla de nuevo a la principal. Ejecutamos las siguientes sentencias:

```

IF OBJECT_ID('CONSULTANT.SAL_trg_state_ctrl','TR') IS NOT NULL
    DROP TRIGGER CONSULTANT.SAL_trg_state_ctrl;
GO

CREATE OR ALTER TRIGGER CONSULTANT.SAL_trg_state_ctrl
    ON CONSULTANT.SAL_CANAL_DENUNCIAS
    AFTER UPDATE
AS
BEGIN
    IF UPDATE(estado)
        BEGIN
            UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS
            SET fecha_final = GETDATE() WHERE estado = 'Cerrada'
            -- Para cerrar la denuncia, se le añade la fecha actual de 'fecha_final'

            INSERT INTO CONSULTANT.SAL_STANDBY (
                ID, ID_ref, categoria, tipo, nombre, apellidos, email, telefono, descripcion, estado, SAL_PERSONAL_ID_personal, fecha_inicio, fecha_final
            ) SELECT *
            FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE estado = 'En espera de ser procesada' DELETE FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE estado = 'En espera de ser procesada'
            -- Se traslada la fila 'En espera de ser procesada' de SAL_CANAL_DENUNCIAS a SAL_STANDBY

            INSERT INTO CONSULTANT.SAL_CANAL_DENUNCIAS (
                ID, ID_ref, categoria, tipo, nombre, apellidos, email, telefono, descripcion, estado, SAL_PERSONAL_ID_personal, fecha_inicio, fecha_final
            ) SELECT *
            FROM CONSULTANT.SAL_STANDBY WHERE estado = 'Activa' DELETE FROM CONSULTANT.SAL_STANDBY WHERE estado = 'Activa'
            -- Se traslada la fila 'Activa' de SAL_STANDBY a SAL_CANAL_DENUNCIAS
        END
    END;
GO

```

Y comprobamos su funcionamiento:

```

bonus_auditoria.s...S\SCOUTS (61) -> X
229 GO
230
231 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 3;
232 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'Cerrada' where ID = 3 -- Done
233
234 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 4;
235 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'En espera de ser procesada' where ID = 4 -- Done
236
237 SELECT * FROM CONSULTANT.SAL_STANDBY WHERE ID = 4 ;
238 UPDATE CONSULTANT.SAL_STANDBY set estado = 'Activa' where ID = 4 -- Done
239
240

```

124 %

Results Messages

ID	ID_ref	categoría	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	fecha_inicio	fecha_final	
1	3	10002	Grupos scouts	Otro	_	Vazquez	nulla.inciidunt@purusgravida.com	622678245	Se burlaron de mis niñas del alma porque no sabi...	Cerrada	10	2020-02-03 19:11:54.0540000	2021-05-26 03:55:22.0433333

Activar Windows

Query executed successfully.

SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\SAL\_SCOUTS... | SAL\_SCOUTS | 00:00:00 | 1 rows

```

bonus_auditoria.s...S\SCOUTS (61) -> X
229 GO
230
231 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 3;
232 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'Cerrada' where ID = 3 -- Done
233
234 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 4;
235 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'En espera de ser procesada' where ID = 4 -- Done
236
237 SELECT * FROM CONSULTANT.SAL_STANDBY WHERE ID = 4 ;
238 UPDATE CONSULTANT.SAL_STANDBY set estado = 'Activa' where ID = 4 -- Done
239
240

```

124 %

Results Messages

ID	ID_ref	categoría	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	fecha_inicio	fecha_final	
1	4	10003	Trabajadores/as	Incumplimiento del cod. ético	Jescie	Jensen	anet.mass@moleste.net	637275473	Le metieron la mano en el bolsillo a mi hijo y ...	Activa	11	2019-08-17 10:32:01.0010000	NULL

Activar Windows

Query executed successfully.

SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\SAL\_SCOUTS... | SAL\_SCOUTS | 00:00:00 | 1 rows

```

bonus_auditoria.s...S\SCOUTS (61) -> X
229 GO
230
231 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 3;
232 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'Cerrada' where ID = 3 -- Done
233
234 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 4;
235 UPDATE CONSULTANT.SAL_CANAL_DENUNCIAS set estado = 'En espera de ser procesada' where ID = 4 -- Done
236
237 UPDATE CONSULTANT.SAL_STANDBY set estado = 'Activa' where ID = 4 -- Done
238 SELECT * FROM CONSULTANT.SAL_CANAL_DENUNCIAS WHERE ID = 4;
239
240

```

124 %

Results Messages

ID	ID_ref	categoría	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	fecha_inicio	fecha_final	
1	4	10003	Trabajadores/as	Incumplimiento del cod. ético	Jescie	Jensen	anet.mass@moleste.net	637275473	Le metieron la mano en el bolsillo a mi hijo y ...	Activa	11	2019-08-17 10:32:01.0010000	NULL

ID	ID_ref	categoría	tipo	nombre	apellidos	email	telefono	descripcion	estado	SAL_PERSONAL_ID_personal	fecha_inicio	fecha_final

Activar Windows

Query executed successfully.

SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\SAL\_SCOUTS... | SAL\_SCOUTS | 00:00:00 | 1 rows

## - 4. Legislación (GDPR – General Data Prot. Reg. (EU))

El **Reglamento General de Protección de Datos (RGPD, o GDPR – General Data Protection Regulation)** es una normativa que regula la protección de los datos de los ciudadanos que viven en la Unión Europea.

El reglamento entró en vigor el 24 de mayo de 2016, pero fue de obligado cumplimiento a partir del 25 de mayo de 2018.

El **GDPR** tiene un impacto significativo para las organizaciones y su forma de manejar, con sanciones potencialmente muy grandes para aquellas empresas que sufren una violación, llegando hasta un 4% de los ingresos globales.

Impacta directamente en el almacenamiento, procesamiento, acceso, transferencia y divulgación de los registros de datos de un individuo y afecta a cualquier organización a nivel mundial que procece datos personales de personas de la **Unión Europea**.

Su objetivo principal es dar control a los ciudadanos y residentes sobre sus datos personales y simplificar el entorno regulador de los negocios internacionales unificando la regulación dentro de la **UE**.

La UE ha ampliado sustancialmente la definición de datos personales en el marco del **GDPR**. Para reflejar los tipos de organizaciones de datos que ahora recopilan sobre personas, los identificadores online, como las direcciones IP, ahora son considerados como datos personales. Otros datos, como la información económica, cultural o de salud mental, también se consideran información de identificación personal.

**Microsoft** ha señalado su compromiso con el cumplimiento del **RGPD** y su apoyo a los clientes en el proceso de adaptación al mismo, que se extenderá por todo el entorno de las tecnologías y con ello por el entorno de **Microsoft SQL Server**. Por tanto, Microsoft recomienda a empresas empezar su proceso de adaptación al Reglamento centrándose en cuatro aspectos principales:

- **Detectar:** Determinar qué información personal está siendo gestionada y dónde reside la misma, identificando qué servidores o bases de datos contienen información personal o qué filas o columnas pueden marcarse como contenedoras de la misma. SQL Server dispone de varias herramientas para descubrir los datos, como la tabla de sistema **sys.columns**, índices **Full Text**, **Profiler** o **xevents**.

The screenshot shows the Microsoft SQL Server Management Studio interface with the following details:

- File Explorer:** Shows the database structure for "SAL\_W016\_SCOUTS" including "master", "tempdb", "model", "msdb", "SAL\_W016\_SCOUTS", and "SAL\_W016\_SCOUTS.dbo".
- Object Explorer:** Shows the schema and objects within the "SAL\_W016\_SCOUTS" database.
- SQL Query Editor:** Contains the following T-SQL query:

```
1 SELECT * FROM sys.columns
```
- Results Grid:** Displays the results of the query, which lists columns from various tables across the database. The columns include: object\_id, name, column\_id, system\_type\_id, user\_type\_id, max\_length, precision, scale, collation\_name, is\_nullable, is\_xml\_padded, is\_recomputed, is\_identity, is\_computed, is\_stored, is\_inited, and is\_ran\_id\_col.
- Status Bar:** Shows "Query executed successfully." and the session information "SAL\_W016\_SCOUTS (14297MB) SAL\_W016\_SCOUTS... master 08:08:25 1071 rows".

- Administrador: Supervisar cómo se puede acceder a esa información personal y cómo está procesada y utilizada, asegurándose de que los permisos otorgados a las personas que acceden a los datos son los mínimos necesarios para la realización de su cometido. Este punto se puede acometer con SQL Server controlando permisos con **SQL Server Authentication**, enmascarando datos con **DDM (Dynamic Data Masking)** o filtrando los datos que puede ver un usuario en una tabla con **RLS (Row-Level Security)**.
  - Proteger: Establecer controles de seguridad para prevenir, detectar y reaccionar a las debilidades e incumplimientos en la protección de datos. Esto requiere diferentes métodos para diferentes tipos de información y escenarios. Para proteger el dato, **SQL Server** dispone de varios mecanismos de encriptación a nivel físico y lógico, como encriptación de conexiones, **TDE (Transparent Data Encryption)**, **Always Encrypted**. También podemos controlar quien y cuando está accediendo a los datos mediante **SQL Server Audit**.
  - Monitorizar e informar: Guardar auditorías de todas las operaciones relacionadas con el manejo de información personal, gestionar las solicitudes de información u notificar cuando se produzca un incumplimiento del reglamento. Así como realizar un seguimiento de estos procesos y procedimientos para garantizar que se mantienen actualizados.
- Por último, en **SQL Server** podemos controlar el historial de cambios o accesos a una tabla con **System-Versioned Temporal Tables** y también con **SQL Server Audit**, y reportar esos fallos mediante **SQL Alerts** y **DB Mail** o paneles gráficos en tiempo real con **Power BI**.

## - 5. Ataques

La mayoría de información sensible del mundo está almacenada en sistemas gestores de bases de datos como **MySQL**, **Oracle**, **Microsoft SQL Server** entre otros. Toda esa información es la que hace que los hackers centren todo su esfuerzo en poder acceder a esa información por medio de alguna de las muchas vulnerabilidades que nos podemos encontrar referente a estos gestores, vulnerabilidades que o bien pueden ser debidos a problemas de seguridad en el software, en este caso es necesario tener siempre actualizada a la última versión para corregir posibles problemas de seguridad, y otras veces a la forma en la que está configurado su acceso o bien problemas en la programación de la aplicación, problemas que pueden causar el conocido ataque **SQL Injection**, uno de los ataques más comunes cuando de bases de datos se trata.

Hasta este momento, gran parte de esfuerzo para mejorar la seguridad de cualquier servicio informático se centraba en asegurar los perímetros de las redes por medio de **firewalls**, **IDS/IPS (Intruders Detection System/Intruders Prevention System – Sistema de Detección/Prevención de Intrusos)** y antivirus, pero cada vez las organizaciones están poniendo más esfuerzos en la protección de la seguridad de las bases de datos protegiéndolos de intrusiones y cambios no autorizados.

A continuación, explicaremos y mostraremos los ataques a las bases de datos que existen hoy en día con alguna demostración de por medio.

## • 5.1. **DDoS**

**DDoS (Distributed Denial of Service)**, **Denegación de Servicios Distribuído** es un ataque a un sistema de ordenadores o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad con la red de la víctima o sobrecarga de los recursos computacionales del sistema atacado. Los ataques **DDoS** se generan mediante la saturación de los puertos con múltiples flujos de información haciendo que el servidor se sobrecargue y no pueda seguir prestando su servicio. Por eso se denomina denegación, pues hace que el servidor no pueda atender la cantidad enorme de solicitudes. Esta técnica es usada por los *crackers* o piratas informáticos para dejar fuera de servicio servidores objetivo. A nivel global, este problema ha ido creciendo, en parte por la mayor facilidad para crear ataques y también por la mayor cantidad de equipos disponibles mal configurados o con fallos de seguridad que son explotados para generar estos ataques. Se ve un aumento en los ataques por reflexión y de amplificación por sobre el uso de **botnets** (equipos informáticos que se ejecutan de manera autónoma y automática y que puede controlar todos los ordenadores/servidores infectados de forma remota).

En **SQL Server**, el ataque está basado en la técnica que interfiere en las operaciones de **Microsoft SQL Server Resolution Protocol (Resolución de Protocolos de Microsoft SQL Server)** para el propósito de lanzar un ataque **DDoS de reflexión** que consiste en que los atacantes no requieren del uso de **malware** para infectar y controlar los **bots** de una **botnet**.

Este ataque ocurre cuando **SQL Server** responde a una consulta o petición de un cliente, atentando de explotar la **Microsoft SQL Server Resolution Protocol (MC-SQLR)**, escuchando por el puerto **1434 UDP**.

La **Microsoft SQL Server Resolution Protocol** se usa cada vez cuando un cliente necesita información en una instancia **SQL Server**. Cuando se está conectado al servidor de bases de datos, el servidor responde al cliente con una lista de instancias de bases de datos usando el protocolo **MC-SQLR** y asiste en identificar qué instancias de bases de datos están atentando en establecer conexión.

Los hackers pueden aprovecharse de los servidores **SQL Server** ejecutando peticiones de un script usando una dirección IP forjada para hacerle aparecer lo qué está viniendo desde el servidor objetivo. El número de instancias presentes en el **SQL Server** afectado determina el poder o el factor de amplificación del ataque.

Se recomiendan tomas las siguientes medidas para mitigar el ataque y trabajar con las aplicaciones comprometidas de acuerdo a:

- El uso de ingreso y regreso de filtros aplicados a los puertos **SQL Server** en los *firewalls*, *routers*, o dispositivos cercanos que puedan prevenir este ataque. Si hay una negociación para mantener el puerto **1434 UDP** abierto, debería estar filtrado para solo permitir direcciones IP confiadas.
- Bloquear conexiones de entrada desde Internet si los puertos no son usados para acceso externo o de administración.
- Los servidores que tienen solo una instancia de bases de datos que no utilice **Microsoft SQL Server Resolution Protocol**. Está deshabilitado por defecto desde Microsoft SQL Server 2008, sin embargo, está habilitado en versiones anteriores y en versiones de escritorio. Considera deshabilitar el servicio de **Microsoft SQL Server Resolution Protocol** para prevenir su abuso.

Si el uso de **Microsoft SQL Server Resolution Protocol** es necesario, añade una capa adicional de seguridad antes de que el servicio sea accedido, tales como métodos de autentificación seguros (SSH, VPN...) o por filtración.

## • 5.2. SQL Injection

La **inyección SQL (SQL Injection)** es un método de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.

El origen de la vulnerabilidad radica en la incorrecta comprobación o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o script que esté incrustado en otro.

Este tipo de intrusión normalmente es de carácter malicioso dañino o espía, por tanto, es un problema de seguridad informática, y debe ser tomado en cuenta por el programado de la aplicación para poder prevenirlo. Un programa elaborado con descuido, displicencia o con ignorancia del problema, podrá resultar ser vulnerable, y la seguridad de la base de datos podrá quedar eventualmente comprometida.

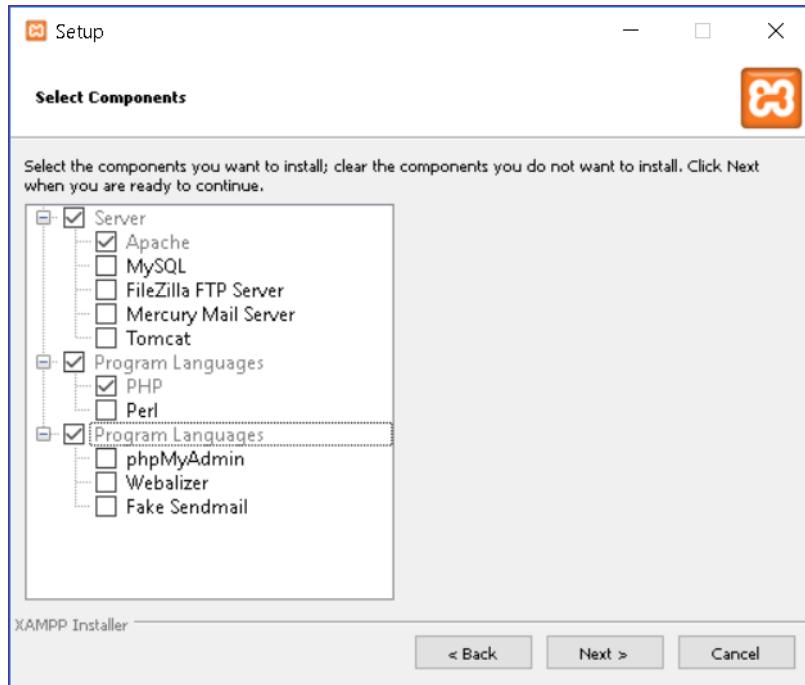
La intrusión ocurre durante la ejecución del programa vulnerable, ya sea, en ordenadores de escritorio o bien en sitios Web, en este último caso obviamente ejecutándose en el servidor que los aloja.

La vulnerabilidad se puede producir automáticamente cuando un programa une sentencia SQL en tiempo de ejecución, o bien durante la fase de desarrollo, cuando el programado explica la sentencia SQL a ejecutar en forma desprotegida. En cualquier caso, siempre que el programador necesite y haga uso de parámetros a ingresar por parte del usuario, a efectos de consultar una base de datos; ya que, justamente, dentro de los parámetros es donde se puede incorporar el código SQL intruso.

En **SQL Server**, la inyección de código SQL es un ataque en el que se inserta código malintencionado en cadenas que posteriormente se pasan a una instancia de **SQL Server** para su análisis y ejecución. Todos los procedimientos que generan instrucciones SQL deben revisarse en busca de vulnerabilidades de inyección de código, ya que **SQL Server** ejecutará todas las consultas recibidas que sean válidas desde el punto de vista sintáctico. Un atacante cualificado y con determinación puede manipular incluso los datos con parámetros.

Para hacer una demostración de inyección SQL en **SQL Server** haremos lo siguiente:

- Necesitamos un servidor web para realizar el ataque, para ello instalaremos **XAMPP** y solamente le añadiremos el servidor **Apache** y el servidor **PHP**:



- Como estamos utilizando bases de datos **SQL Server**, pues no nos servirá utilizar **phpMyAdmin** ya sirve solo para **MySQL/MariaDB**.
- Nos hará falta el lenguaje de programación **PHP** poder realizar un formulario con **HTML + PHP**.
- Debemos instalar drivers de Microsoft ODBC y conectores PHP para SQL Server en la página oficial de Microsoft, especialmente los archivos **sqlsvr32.exe** y **msodbcsql.msi**:

Microsoft® ODBC Driver 17 for SQL Server® - Windows, Linux, & macOS

Important! Selecting a language below will dynamically change the complete page content to that language.

Select Language: English

This page is no longer maintained. Please read the details below.

(+) Details  
(+) System Requirements  
(+) Install Instructions

Microsoft Drivers for PHP for SQL Server

Important! Selecting a language below will dynamically change the complete page content to that language.

Language: English

The Microsoft Drivers for PHP for SQL Server provide connectivity to Microsoft SQL Server from PHP applications.

(+) Details  
(+) System Requirements

- Cuando ejecutemos el archivo **SQLSRV32.EXE** nos descomprimirá varios archivos .dll, los movemos al directorio **C:\xampp\php\ext**:

Nombre	Fecha de modifica...	Tipo	Tamaño
SQLSRV_ThirdPartyNotices.rtf	27/02/2015 10:12	Documento de tex...	53 KB
PHP Drivers License Terms.rtf	04/03/2015 13:48	Documento de tex...	67 KB
SQLSRV Readme.htm	04/03/2015 13:48	Chrome HTML Do...	15 KB
php_pdo_sqlsrv_54_nts.dll	05/03/2015 10:14	Extensión de la ap...	163 KB
php_pdo_sqlsrv_54_ts.dll	05/03/2015 10:14	Extensión de la ap...	164 KB
php_pdo_sqlsrv_55_nts.dll	05/03/2015 10:14	Extensión de la ap...	178 KB
php_pdo_sqlsrv_55_ts.dll	05/03/2015 10:15	Extensión de la ap...	179 KB
php_pdo_sqlsrv_56_nts.dll	05/03/2015 10:15	Extensión de la ap...	178 KB
php_pdo_sqlsrv_56_ts.dll	05/03/2015 10:15	Extensión de la ap...	177 KB
php_sqlsrv_54_nts.dll	05/03/2015 10:15	Extensión de la ap...	180 KB
php_sqlsrv_54_ts.dll	05/03/2015 10:15	Extensión de la ap...	193 KB
php_sqlsrv_55_nts.dll	05/03/2015 10:15	Extensión de la ap...	196 KB
php_sqlsrv_55_ts.dll	05/03/2015 10:16	Extensión de la ap...	193 KB
php_sqlsrv_56_nts.dll	05/03/2015 10:16	Extensión de la ap...	196 KB
php_sqlsrv_56_ts.dll	05/03/2015 10:16	Extensión de la ap...	179 KB
release.txt	05/03/2015 10:24	Documento de tex...	3 KB
SQLSRV32.EXE	26/05/2021 17:35	Aplicación	547 KB
msodbcsql.msi	26/05/2021 17:35	Paquete de Windo...	4.576 KB

18 elementos

Nombre	Fecha de modifica...	Tipo	Tamaño
php_pdo_sqlite.dll	05/05/2021 3:10	Extensión de la ap...	28 KB
php_pdo_sqlsrv_54_nts.dll	05/03/2015 10:14	Extensión de la ap...	163 KB
php_pdo_sqlsrv_54_ts.dll	05/03/2015 10:14	Extensión de la ap...	164 KB
php_pdo_sqlsrv_55_nts.dll	05/03/2015 10:14	Extensión de la ap...	178 KB
php_pdo_sqlsrv_55_ts.dll	05/03/2015 10:15	Extensión de la ap...	179 KB
php_pdo_sqlsrv_56_nts.dll	05/03/2015 10:15	Extensión de la ap...	178 KB
php_pdo_sqlsrv_56_ts.dll	05/03/2015 10:16	Extensión de la ap...	179 KB
php_pgsql.dll	05/05/2021 3:10	Extensión de la ap...	105 KB
php_phplibg_webhelper.dll	05/05/2021 3:10	Extensión de la ap...	18 KB
php_shmop.dll	05/05/2021 3:10	Extensión de la ap...	18 KB
php_smp.dll	05/05/2021 3:10	Extensión de la ap...	410 KB
php_soap.dll	05/05/2021 3:10	Extensión de la ap...	245 KB
php_sockets.dll	05/05/2021 3:10	Extensión de la ap...	78 KB
php_sodium.dll	05/05/2021 3:10	Extensión de la ap...	75 KB
php_sqlite3.dll	05/05/2021 3:10	Extensión de la ap...	49 KB
php_sqlsrv_54_nts.dll	05/03/2015 10:15	Extensión de la ap...	177 KB
php_sqlsrv_54_ts.dll	05/03/2015 10:15	Extensión de la ap...	180 KB
php_sqlsrv_55_nts.dll	05/03/2015 10:15	Extensión de la ap...	193 KB
php_sqlsrv_55_ts.dll	05/03/2015 10:15	Extensión de la ap...	196 KB
php_sqlsrv_56_nts.dll	05/03/2015 10:16	Extensión de la ap...	193 KB
php_sqlsrv_56_ts.dll	05/03/2015 10:16	Extensión de la ap...	196 KB
php_sysvshm.dll	05/05/2021 3:10	Extensión de la ap...	19 KB
php_tidy.dll	05/05/2021 3:10	Extensión de la ap...	735 KB
php_xsl.dll	05/05/2021 3:10	Extensión de la ap...	285 KB
php_zend_test.dll	05/05/2021 3:10	Extensión de la ap...	27 KB

56 elementos 16 elementos seleccionados 2.25 MB

- Luego tenemos que modificar el archivo de configuración **php.ini** y añadir las extensiones **.dll** para el funcionamiento con **SQL Server** y reiniciar el servidor de **Apache** para aplicar los cambios:

```

937 ;extension=openssl
938 ;extension=pdo_firebird
939 ;extension=pdo_mysql
940 ;extension=pdo_oci
941 ;extension=pdo_odbc
942 ;extension=pdo_pgsql
943 ;extension=pdo_sqlite
944 ;extension=pgsql
945 ;extension=shmop
946
947 ; CONFIGURACIÓN SQL SERVER
948 extension=php_pdo_sqlsrv_7_nts_x64.dll
949 extension=php_pdo_sqlsrv_7_ts_x64.dll
950 extension=php_sqlsrv_7_nts_x64.dll
951 extension=php_sqlsrv_7_ts_x64.dll
952
953 ; The MIBS data available in the PHP distribution must be installed.
954 ; See http://www.php.net/manual/en/snmp.installation.php
955 ;extension=snmp
956
957 ;extension=soap
958 ;extension=sockets
959 ;extension=sodium
960 ;extension=sqlite3
961 ;extension=tidy
962 ;extension=xsl

```

MS ini file length : 75.636 lines : 1.999 Ln : 951 Col : 34 Pos : 37.695 Windows (CR LF) UTF-8 INS

- Creamos una tabla que contenga usuarios en nuestra base de datos ejecutando las siguientes sentencias:

```
USE SAL_SCOUTS
GO
```

```
DROP TABLE IF EXISTS SAL_USERS
GO
```

```
CREATE TABLE SAL_USERS (
    ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    username VARCHAR(100) NOT NULL,
    password VARCHAR(1000) NOT NULL,
    name VARCHAR(100) NOT NULL,
    surname VARCHAR(100) NOT NULL
);
GO
```

```
INSERT INTO SAL_USERS (username,password,name,surname)
values
('SAL_SCOUTS','Abcd1234. ','Saul','Altoubah Leon'),
('JRM_SCOUTS','Abcd1234. ','Juan','Robles Martinez'),
('LGC_SCOUTS','Abcd1234. ','Lucia','Garcia Caamanho'),
('SDDT_SCOUTS','Abcd1234. ','Santiago','Descalzo De La Torre'),
('ACM_SCOUTS','Abcd1234. ','Alberto','Castaña Manzana'),
('AMP_SCOUTS','Abcd1234. ','Alma','Martillo Puertas'),
('MMO_SCOUTS','Abcd1234. ','Marta','Manresa olivo'),
```

```

('DGC_SCOUTS', 'Abcd1234.', 'David', 'Gonzo Castanha'),
('SHAL_SCOUTS', 'Abcd1234.', 'Santiago', 'Abente Luares'),
('JJPP_SCOUTS', 'Abcd1234.', 'Juan Jose', 'Pau Pernas'),
('MMP_SCOUTS', 'Abcd1234.', 'Maria', 'Martinez Perez'),
('VSP_SCOUTS', 'Abcd1234.', 'Valeria', 'Suarez Perez');

```

GO

**SELECT \* FROM SAL\_USERS**

GO

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'SAL\_WST16\_SCOUTS' is selected. In the center pane, a query window titled 'SQLQuery1.sql - S...\$SAL\_SCOUTS (53)\*' contains the following code:

```

21   ('ACM_SCOUTS', 'Abcd1234.', 'Alberto', 'Castaña Manzana'),
22   ('AMP_SCOUTS', 'Abcd1234.', 'Alma', 'Martillo Puertas'),
23   ('MMO_SCOUTS', 'Abcd1234.', 'Marta', 'Manresa olivo'),
24   ('DGC_SCOUTS', 'Abcd1234.', 'David', 'Gonzo Castanha'),
25   ('SHAL_SCOUTS', 'Abcd1234.', 'Santiago', 'Abente Luares'),
26   ('JJPP_SCOUTS', 'Abcd1234.', 'Juan Jose', 'Pau Pernas'),
27   ('MMP_SCOUTS', 'Abcd1234.', 'Maria', 'Martinez Perez'),
28   ('VSP_SCOUTS', 'Abcd1234.', 'Valeria', 'Suarez Perez');
29 GO
30
31 SELECT * FROM SAL_USERS
32 GO

```

The results pane shows a table with 12 rows of data inserted into the SAL\_SCOUTS table:

ID	username	password	name	surname
1	SAL_SCOUTS	Abcd1234.	Saul	Alloubah Leon
2	JRM_SCOUTS	Abcd1234.	Juan	Robles Martinez
3	LGC_SCOUTS	Abcd1234.	Lucia	Garcia Caamano
4	SDDT_SCOUTS	Abcd1234.	Santiago	Descalzo De La Torre
5	ACM_SCOUTS	Abcd1234.	Alberto	Castaña Manzana
6	AMP_SCOUTS	Abcd1234.	Alma	Martillo Puertas
7	MMO_SCOUTS	Abcd1234.	Marta	Manresa olivo
8	DGC_SCOUTS	Abcd1234.	David	Gonzo Castanha
9	SHAL_SCOUTS	Abcd1234.	Santiago	Abente Luares
10	JPPP_SCOUTS	Abcd1234.	Juan.Jose	Pau Pernas
11	MMP_SCOUTS	Abcd1234.	Maria	Martinez Perez
12	VSP_SCOUTS	Abcd1234.	Valeria	Suarez Perez

At the bottom of the results pane, a message indicates: "Query executed successfully."

- Y por último unos archivos .php que actuarán como formulario, envío de datos del formulario y conexión a la base de datos:

~ *sql\_server\_conn.php*

The screenshot shows the Sublime Text editor with the file 'sql\_server\_conn.php' open. The code is as follows:

```

1 <?php
2     error_reporting(1);
3     //Function sal_scouts() for attach to action.php
4     function sal_scouts() {
5         $instance_host = "localhost"; //IP address
6         $login = array("Database"=>"SAL_SCOUTS", "UID"=>"sa", "PWD"=>"Abcd1234."); // PDO array for login
7         $conn = sqlsrv_connect($instance_host, $login); // Allows us to prepare the connection to our DB
8         if($conn == false) {
9             die(FormatErrors(sqlsrv_errors())); // If the connection goes wrong, it'll show us an error msg
10        }
11        return $conn;
12    }
13 ?>

```

## ~ form.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>SQL INJECTION TEST - SAL_SCOUTS</title>
    </head>
    <body>
        <h1>~ SQL INJECTION TEST - SAL_SCOUTS ~</h1>
        <form method="POST" action="action.php"><!-- A simple and small form -->
            <p> · Username: <input type="name" name="usr" /></p>
            <p> · Password: <input type="password" name="pwd" /></p>
            <input type="submit" name="submit" />
        </form>
    </body>
</html>
```

## ~ action.php

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>SQL INJECTION TEST - SAL_SCOUTS</title>
    </head>
    <body style="margin-top: 50px">
        <div class="container">
            <div class="row">
                <div class="span6">
                    <?php
                        require_once("sql_server_conn.php");
                        $usr = $_POST['usr'];
                        $pwd = $_POST['pwd'];
                        $query = "SELECT * FROM SAL_USERS WHERE username = '$usr' AND password = '$pwd' ";
                        $connDB = sal_scouts();
                        $result = sqlsrv_query($connDB,$query);
                        if (sqlsrv_has_rows($result) > 0) {
                            echo "<h4>-- User Information -- </h4>";
                            while ($row = sqlsrv_fetch_array($result)) {
                                echo "<p>". "Username <b>: "</$row[1]</b></p>";
                                echo "<p>". "Password : <b>:</$row[2]</b></p>";
                                echo "<p>". "Name : <b>:</$row[3]</b> Surname <b>:</$row[4]</b></p>";
                                echo "-----";
                            }
                        } else {
                            echo "Invalid user id or password";
                        }
                        sqlsrv_close($connDB);
                    ?>
                </div>
            </div>
        </div>
    </body>
</html>
```

Activar Windows  
Ve a Configuración para activar Windows.

Tab Size: 4

PHP

Ahora que tenemos todo preparado vamos a comprobar que funciona el formulario:

The screenshot shows a web browser window with two tabs. The active tab is titled "SQL INJECTION TEST - SAL\_SCOUTS" and the URL is "localhost/sql\_server/form.html". The page content is a form with two fields: "Username" containing "SAL\_SCOUTS" and "Password" containing ".....". Below the form is a button labeled "Enviar". The second tab is also titled "SQL INJECTION TEST - SAL\_SCOUTS" and the URL is "localhost/sql\_server/action.php".

~SQL INJECTION TEST - SAL\_SCOUTS~

· Username:

· Password:

SQL INJECTION TEST - SAL\_SCOUTS

← → ⌂ ① localhost/sql\_server/action.php

### -- User Information --

Username : SAL\_SCOUTS

Password : Abcd1234.

Name : Saul Surname : Altoubah Leon

-----

Observamos que funciona correctamente el formulario. Si nos fijamos en el código del archivo `action.php` en la parte de la sentencia de consulta:

```
$query = " SELECT * FROM SAL_USERS WHERE username = '$usr' AND password = '$pwd' ";
```

En el código anterior los datos `$usr` y `$pwd` tienen los valores del nombre de usuario y la contraseña del usuario. El intérprete ejecutará el comando (el cual estarán almacenado en `$result`) basado en las inserciones. Ahora si un atacante inserta `abcd` como nombre de usuario y `'anything' or 'x'='x'` como contraseña, entonces la consulta será construida como:

```
$query = "SELECT * FROM SAL_USERS WHERE username = 'abcd' AND password = 'anything' OR 'x'='x' ";
```

Basado en la precedencia del operador, la cláusula `WHERE` está `TRUE` por cada fila, por lo tanto, la consulta devolverá todos los registros. De esta manera, un atacante será capaz de visualizar toda la información personal de los usuarios. Comprobémoslo en el siguiente inicio de sesión:

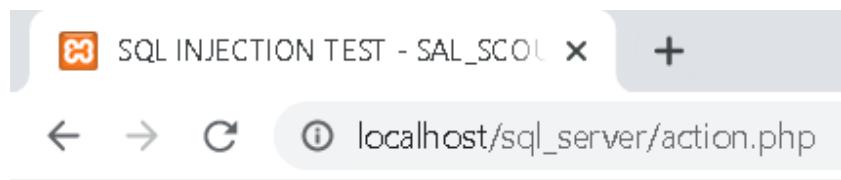
The screenshot shows a web browser window with the title "SQL INJECTION TEST - SAL\_SCOUTS". The address bar indicates the URL is "localhost/sql\_server/form.html". The page content is as follows:

# ~ SQL INJECTION TEST - SAL\_SCOUTS ~

· Username:

· Password:

*anything' or 'x'='x*



-- User Information --

Username : **SAL\_SCOUTS**

Password : **Abcd1234.**

Name : **Saul** Surname : **Altoubah Leon**

---

Username : **JRM\_SCOUTS**

Password : **Abcd1234.**

Name : **Juan** Surname : **Robles Martinez**

---

Username : **LGC\_SCOUTS**

Password : **Abcd1234.**

Name : **Lucia** Surname : **Garcia Caamanho**

---

Username : **SDDT\_SCOUTS**

Password : **Abcd1234.**

Name : **Santiago** Surname : **Descalzo De La Torre**

---

Username : **ACM\_SCOUTS**

Password : **Abcd1234.**

Name : **Alberto** Surname : **Castaña Manzana**

---



Username : AMP\_SCOUTS

Password : Abcd1234.

Name : Alma Surname : Martillo Puertas

-----  
Username : MMO\_SCOUTS

Password : Abcd1234.

Name : Marta Surname : Manresa olivo

-----  
Username : DGC\_SCOUTS

Password : Abcd1234.

Name : David Surname : Gonzo Castanha

-----  
Username : SHAL\_SCOUTS

Password : Abcd1234.

Name : Santiago Surname : Abente Luares

-----  
Username : JJPP\_SCOUTS

Password : Abcd1234.

Name : Juan Jose Surname : Pau Pernas

-----  
Username : MMP\_SCOUTS

Password : Abcd1234.

Name : Maria Surname : Martinez Perez

-----  
Username : VSP\_SCOUTS

Password : Abcd1234.

Name : Valeria Surname : Suarez Perez

Incluso haciéndolo desde **SSMS** hace el mismo resultado:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the top-left corner, there's a title bar for "Solution1 - Miscellaneous Files - SQLQuery1.sql - SAL\_WS16\_SCOUTS.SAL\_SCOUTS (WinAuth) - Microsoft SQL Server Management Studio". The main area contains a query editor window with the following SQL code:

```
SQLQuery1.sql - S...\\SAL_SCOUTS (53)*
21 ('ACM_SCOUTS','Abcd1234.', 'Alberto', 'Castaña Manzana'),
22 ('AMP_SCOUTS','Abcd1234.', 'Alma', 'Martillo Puertas'),
23 ('MMO_SCOUTS','Abcd1234.', 'Marta', 'Manresa olivo'),
24 ('DGC_SCOUTS','Abcd1234.', 'David', 'Gonzo Castanha'),
25 ('SHAL_SCOUTS','Abcd1234.', 'Santiago', 'Abente Luares'),
26 ('JPP_SCOUTS','Abcd1234.', 'Juan Jose', 'Pau Pernas'),
27 ('MMP_SCOUTS','Abcd1234.', 'Maria', 'Martinez Perez'),
28 ('VSP_SCOUTS','Abcd1234.', 'Valeria', 'Suarez Perez');
29 GO
30
31 SELECT * FROM SAL_USERS WHERE username = 'abcd' AND password = 'anything' or 'x'='x';
32 GO
```

Below the code, there's a results grid titled "Results" showing the following data:

ID	username	password	name	surname
1	SAL_SCOUTS	Abcd1234.	Saul	Altoubah Leon
2	JRM_SCOUTS	Abcd1234.	Juan	Robles Martinez
3	LGC_SCOUTS	Abcd1234.	Lucia	Garcia Caamano
4	SDDT_SCOUTS	Abcd1234.	Santiago	Descalo De La Torre
5	ACM_SCOUTS	Abcd1234.	Alberto	Castaña Manzana
6	AMP_SCOUTS	Abcd1234.	Alma	Martillo Puertas
7	MMO_SCOUTS	Abcd1234.	Marta	Manresa olivo
8	DGC_SCOUTS	Abcd1234.	David	Gonzo Castanha
9	SHAL_SCOUTS	Abcd1234.	Santiago	Abente Luares
10	JPP_SCOUTS	Abcd1234.	Juan Jose	Pau Pernas
11	MMP_SCOUTS	Abcd1234.	Maria	Martinez Perez
12	VSP_SCOUTS	Abcd1234.	Valeria	Suarez Perez

At the bottom of the SSMS window, the status bar displays: "Query executed successfully." and "SAL\_WS16\_SCOUTS (14.0 RTM) | SAL-SCOUTS\\SAL\_SCOUTS ... | SAL\_SCOUTS | 00:00:00 | 12 rows".

### • 5.3. **Ransomware**

El **ransomware** (*malware* de rescate) es un tipo de malware que impide a los usuarios acceder a su sistema o a sus archivos personales y que exige el pago de un rescate para poder acceder de nuevo a ellos. Las primeras variantes de **ransomware** se crearon al final de la década de los 80, y el pago debía efectuarse por correo postal. Hoy en día los creadores de **ransomware** piden que el pago se efectúe mediante criptomonedas o tarjetas de crédito.

Puede infectar nuestro ordenador de varias formas. Uno de los métodos más habituales actualmente es a través de spam malicioso, o **malspam**, que son mensajes no solicitados que se utilizan para enviar malware por correo electrónico. El mensaje de correo electrónico puede incluir archivos adjuntos trampa, como PDFs o documentos de Word. También puede contener enlaces a sitios web maliciosos.

El **malspam** usa ingeniería social para engañar a la gente con el fin de que abra archivos adjuntos o haga clic en vínculos que parezcan legítimos, aparentando que proceden de una institución de confianza o de un amigo. Los ciberdelincuentes emplean la ingeniería social en otros tipos de ataques de **ransomware**, por ejemplo, presentarse como el FBI para asustar a los usuarios y obligarles a pagar una suma de dinero por desbloquear los archivos.

Otro método de infección habitual, que alcanzó su pico en 2016, es la publicidad maliciosa. La publicidad maliciosa consiste en el uso de publicidad en línea para distribuir malware con poca interacción por parte del usuario o incluso ninguna. Mientras navegan por la web, incluso por sitios legítimos, los usuarios pueden ser conducidos a servidores de delictivos sin necesidad de hacer clic en un anuncio.

Estos servidores clasifican los detalles de los ordenadores de las víctimas y sus ubicaciones y, a continuación, seleccionan el malware más adecuado para enviarlo.

Si **SQL Server** tiene los archivos bloqueados o en uso, ¿cómo puede un **ransomware** encriptar esos archivos? Tradicionalmente los **ransomware** son suficientemente tortuosos para detectar cosas como **SQL Server** y detener los **SQL Services** (o reiniciarlos) de acuerdo en conseguir el acceso a la base de datos y los archivos de registro. En situaciones como es el **ransomware** ha sido capaz de encontrar copias de seguridad en la red local o en redes compartidas e incluso poder encriptarlas.

Lo peor que se puede hacer en un ataque como este es tener que pagar el secuestro en el caso de que no tengamos unas buenas copias de seguridad. Cuando una copia de seguridad es buena se refiere a los siguientes aspectos:

- Las copias de seguridad han sido examinadas regularmente y saber poder cómo restaurarlas.
- Las copias de seguridad están a la fecha y lo suficientemente cercanas al tiempo de ataque del **ransomware** para que no se pierdan bastantes datos.
- Las copias de seguridad son trasladadas a un lugar en donde el **ransomware** no pueda destruirlas. No solo en una red compartida a la que la base de datos tenga acceso, pero trasladarla a páginas de almacenamiento en la nube como **Azure** o **Amazon** donde es más difícil destruirlas.
- Las copias de seguridad incluyen nuestras bases de datos del sistema (**master** y **msdb**) que contienen cosas como los usuarios, los planes de mantenimiento y las tareas programadas (**jobs**).

Vamos a proponer esta situación: el equipo servidor ha sido atacado por **ransomware** y se encriptó casi todos los archivos almacenados:

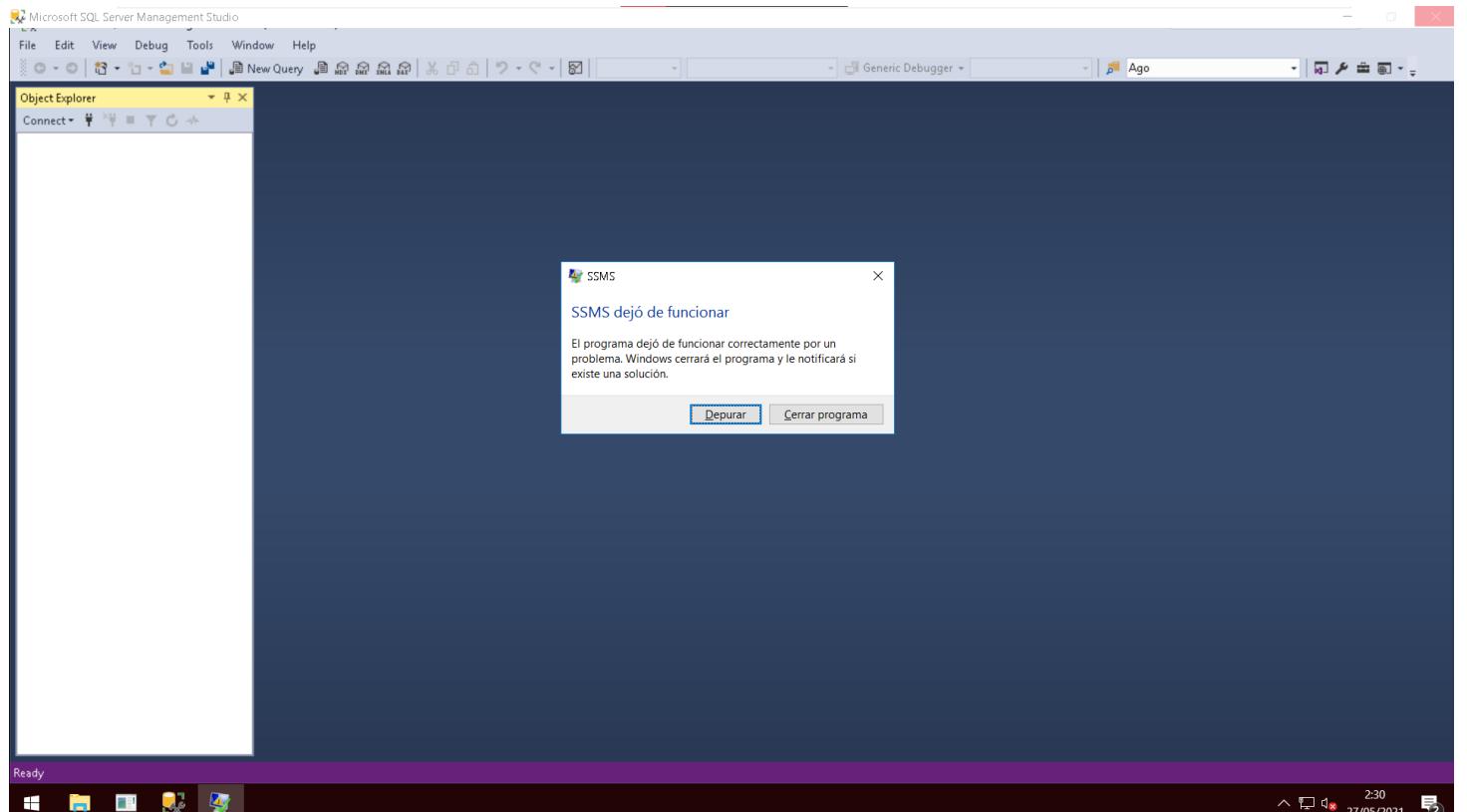


El escritorio del equipo en donde estaban los archivos `.sql` de haber realizado los apartados de este proyecto, están totalmente cifrados:

Y los archivos ***mdf***, ***ndf*** y ***ldf*** no pudieron siquiera aguantar el ataque:

Este equipo > Disco local (C:) > Archivos de programa > Microsoft SQL Server > MSSQL14.SALSCOUTS > MSSQL > DATA				
	Nombre	Fecha de modifica...	Tipo	Tamaño
★ Acceso rápido				
Este equipo	audit	27/05/2021 2:51	Carpetas de archivos	
Descargas	filestream	06/05/2021 19:36	Carpetas de archivos	
Desktop	kys	06/05/2021 2:15	Carpetas de archivos	
Documents	SAL OLTP	24/05/2021 20:52	Carpetas de archivos	
Music	xtp	27/05/2021 2:09	Carpetas de archivos	
Pictures	locked.SAL_SCOUTS_main (1).ndf	27/05/2021 2:47	SQL Server Database Secondary Data File	8.192 KB
Videos	locked.SAL_SCOUTS_main (2).ndf	27/05/2021 2:47	SQL Server Database Secondary Data File	8.192 KB
Disco local (C:)	locked.SAL_SCOUTS_main (3).ndf	27/05/2021 2:47	SQL Server Database Secondary Data File	8.192 KB
Red	locked.SAL_SCOUTS_main (4).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	73.728 KB
	locked.SAL_SCOUTS_main (4).mdf	27/05/2021 2:47	SQL Server Database Primary Data File	1.024.000 KB
	locked.SAL_SCOUTS_main (4).ndf	06/05/2021 19:36	SQL Server Database Secondary Data File	512.000 KB
	locked.SAL_SCOUTS_main (5).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	8.192 KB
	locked.SAL_SCOUTS_main (5).mdf	27/05/2021 2:36	SQL Server Database Primary Data File	8.192 KB
	locked.SAL_SCOUTS_main (5).ndf	06/05/2021 19:36	SQL Server Database Secondary Data File	512.000 KB
	locked.SAL_SCOUTS_main (6).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	61.696 KB
	locked.SAL_SCOUTS_main (6).mdf	27/05/2021 2:47	SQL Server Database Primary Data File	8.192 KB
	locked.SAL_SCOUTS_main (7).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	73.728 KB
	locked.SAL_SCOUTS_main (7).mdf	27/05/2021 2:47	SQL Server Database Primary Data File	8.192 KB
	locked.SAL_SCOUTS_main (8).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	8.192 KB
	locked.SAL_SCOUTS_main (8).mdf	06/02/2021 18:18	SQL Server Database Primary Data File	598.016 KB
	locked.SAL_SCOUTS_main (9).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	8.192 KB
	locked.SAL_SCOUTS_main (9).mdf	26/05/2021 17:28	SQL Server Database Primary Data File	270.336 KB
	locked.SAL_SCOUTS_main (10).ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	256.000 KB
	locked.SAL_SCOUTS_main (10).mdf	26/05/2021 17:28	SQL Server Database Primary Data File	8.192 KB
	locked.SAL_SCOUTS_main (11).mdf	26/05/2021 17:28	SQL Server Database Primary Data File	8.192 KB
	master.mdf	27/05/2021 2:47	SQL Server Database Primary Data File	5.504 KB
	masterlog.ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	1.792 KB
	model.mdf	27/05/2021 2:47	SQL Server Database Primary Data File	8.192 KB
	modellog.ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	8.192 KB
	MSDBData.mdf	26/05/2021 17:28	SQL Server Database Primary Data File	16.576 KB
	MSDBLog.ldf	27/05/2021 2:47	SQL Server Database Transaction Log File	5.184 KB
	SAL_SCOUTS_log2.ldf	06/05/2021 19:36	SQL Server Database Transaction Log File	256.000 KB

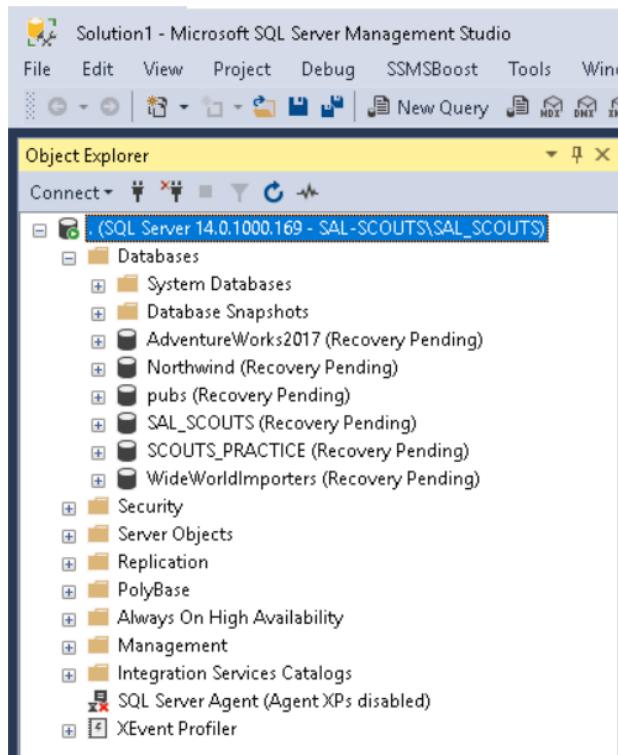
## Ni podemos iniciar SQL Server Management Studio:



Solamente nos queda probar de nuevo a entrar a **SSMS** reiniciando los servicios desde **Task Manager**:

Administrador de tareas					
Archivo Opciones Vista					
Procesos Rendimiento Usuarios Detalles Servicios					
Nombre	PID	Descripción	Estado	Grupo	
VMTools	2796	VMware Tools	En ejecución		
VGAuthService	2748	VMware Alias Manager and Ticket Se...	En ejecución		
vds	3176	Disco virtual	En ejecución		
VaultSvc	628	Administrador de credenciales	En ejecución		
UIODetect		Detección de servicios interactivos	Detenido		
UevAgentService		Servicio de virtualización de la experi...	Detenido		
TrustedInstaller		Instalador de módulos de Windows	Detenido		
TieringEngineService		Administración de capas de almacen...	Detenido		
SQLWriter	2604	SQL Server VSS Writer	En ejecución		
SQLTELEMETRY	4888	SQL Server CEIP service (MSSQLSERV...)	En ejecución		
SQLSERVERAGENT		SQL Server Agent (MSSQLSERVER)	Detenido		
SQLBrowser		SQL Server Browser	Detenido		
SQL Server Distributed Repl...		SQL Server Distributed Replay Contro...	Detenido		
SQL Server Distributed Repl...		SQL Server Distributed Replay Client	Detenido		
sppsvc		Protección de software	Detenido		
Spooler	2500	Cola de impresión	En ejecución		
SNMPTRAP		Captura SNMP	Detenido		
SensorDataService		Servicio de datos del sensor	Detenido		
SamSs	628	Administrador de cuentas de seguridad	En ejecución		
RSOPProv		Conjunto resultante de proveedor de ...	Detenido		
RpcLocator		Ubicador de llamada a procedimient...	Detenido		
PerfHost		DLL de host del Contador de rendimi...	Detenido		
Ntfrs		Replicación de archivos	Detenido		
NTDS	628	Servicios de dominio de Active Direct...	En ejecución		
NfsCInt	2696	Cliente para NFS	En ejecución		
NetTcpPortSharing		Servicio de uso compartido de puer...	Detenido		
Netlogon	628	Net Logon	En ejecución		
MSSQLSERVER		SQL Server (MSSQLSERVER)	Detenido		
msiserver		Windows Installer	Detenido		
MSDTC	3564	Coordinador de transacciones distrib...	En ejecución		
KeyIso	628	Aislamiento de claves CNG	En ejecución		
KdsSvc		Servicio de distribución de clave de ...	Detenido		
Kdc	628	Centro de distribución de claves Kerb...	En ejecución		
IsmServ	2672	Mensajería entre sitios	En ejecución		
gupdatem		Servicio de Google Update (gupdate...)	Detenido		

Se encuentran las bases de datos pendientes de recuperación:



Solamente nos queda eliminar la que se encuentra en la instancia y restaurarla de nuevo. Entonces ejecutamos las siguientes sentencias:

```
USE [master]
GO
DROP DATABASE IF EXISTS SAL_SCOUTS
GO
RESTORE DATABASE [SAL_SCOUTS]
FROM DISK = 'N:\backup\bkp_TDE\SAL_SCOUTS_Full_TDE.bak' WITH FILE = 1,
NOUNLOAD,
REPLACE, -- Para reemplazar los existentes
STATS = 5
GO
-- 8 percent processed.
-- 12 percent processed.
-- 92 percent processed.
-- 96 percent processed.
-- 100 percent processed.
-- Processed 1520 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_main' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG01' on file 1.
-- Processed 64 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FG02' on file 1.
-- Processed 2 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log' on file 1.
-- Processed 0 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_log2' on file 1.
-- Processed 1194 pages for database 'SAL_SCOUTS', file 'SAL_SCOUTS_FILESTREAM_Main' on file 1.
-- RESTORE DATABASE successfully processed 2843 pages in 20.054 seconds (1.107 MB/sec).
```

Lo dicho, si no se dispone de un buen *backup*, los ataques de **ransomware** son un claro y crítico riesgo para los negocios y planes de futuro cuya responsabilidad recae en el DBA.

## · 5.4. Tools

~ DDoS:

· **Incapsula**:



Es una herramienta muy fiable, que ofrece una protección completa contra todo tipo de ataques **DDoS** a nivel y de red y de aplicación. La herramienta filtra automáticamente el tráfico para una mitigación transparente y se basa en una red troncal de **2 Tbps** para un aprovisionamiento excesivo instantáneo.

Protege los sitios web contra los tipos más feroces y peligrosos de ataques **DDoS** sin

interrumpir su negocio. Gracias a su servicio basado en la nube, su negocio en línea estará funcionando incluso bajo ataque y sus visitantes no notarán nada inusual.

· **BeeThink anti-DDoS Guardian**:



**BeeThink**

Esta herramienta anti **DDoS** de **BeeThink** protege los servidores **Windows** contra la mayoría de los ataques **DDoS/DoS**, tales como **ataques SYN, inundación IP, inundación TCP, inundación UDP, inundación ICMP, ataques HTTP DDoS lentos, ataques de Capa 7, ataques de aplicaciones, ataques de adivinación de contraseñas de Escritorio Remoto de Windows**, y más.

Este software de protección **DDoS** es ligero y robusto, y puede desplegarse fácilmente en máquinas de servidor de sitios web de Windows.

· **Cloudbric:**



**Cloudbric** es una herramienta anti-DDoS que puede ser utilizada por cualquier persona con un sitio web y un dominio, independientemente, de las plataformas web que albergan su negocio en línea. Esta herramienta crea un escudo para filtrar los ataques maliciosos. *Penta Security System*, garantiza que **Cloudbric** puede proteger su sitio web contra todo tipo de ciberataques.

· **Cloudflare:**



**Cloudflare** hace que su ordenador sea a prueba de **DDoS**, protegiéndolo contra las amenazas que se dirigen a los protocolos **UDP** e **ICMP**, **SYN/ACK**, amplificación de **DNS** y **NTP** y **ataques de Capa 7**. **Cloudflare Inc**, la compañía que creó esta herramienta, confirma con orgullo que su software ha defendido a los usuarios contra ataques

sostenidos de más de **400 Gbps**. **Cloudflare** enruta automáticamente todo el tráfico de ataque a través de su red global de centros de datos, reduciendo el impacto en su sitio web. Una vez que el tráfico de ataque se desplaza, la herramienta aprovecha la importante capacidad global de la red y de la infraestructura en la que se basa para absorber las inundaciones de tráfico de ataque. Además, esta herramienta aprende de los ataques contra clientes individuales para proteger a todos sus clientes. Gracias a este sistema de aprendizaje automático, su sitio web está protegido contra las últimas amenazas.

- **StormWall Pro:**



**StormWall** Pro es una herramienta avanzada de protección anti-DDoS que puede defender su sitio web contra los ataques más severos. Esta herramienta puede bloquear todo tipo de ataques **DDoS**.

- ~ *SQL Injection:*

- **SQL Power Injection Injector:**

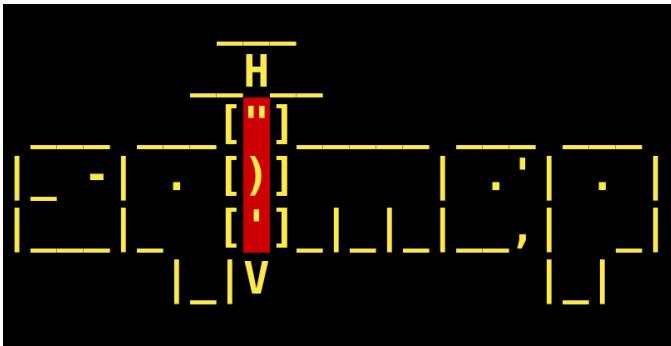


Su principal baza es la capacidad de automatizar inyecciones SQL pesadas utilizando para ello múltiples procesos. Sus principales características son:

- Es multiplataforma.
- Soporta SSL.
- Realiza la técnica de inyección ciega de SQL. Comparando las respuestas verdaderas y falsas de las páginas o de los resultados de las cookies y los retardos de tiempo.

· Soporta los motores de bases de datos: **Microsoft SQL Server, Oracle, MySQL, Sybase/Adaptive y DB2**.

- **SQLMap:**



Es una aplicación de automatización de inyección ciega de **SQL**, desarrollada en **Python**. Es capaz de generar una huella digital activa del sistema de administración de bases de datos. Sus principales características son:

- Al estar escrita en Python y es multiplataforma.

· El apoyo total para **MySQL, Oracle, PostgreSQL** y el servidor de **Microsoft SQL**.  
· Puede identificar también **Microsoft Access, DB2, Informix, Sybase e Interbase**.

· **SQLNinja:**



Es una herramienta para explotar vulnerabilidades de inyección SQL en aplicaciones web que utilizan Microsoft SQL Server como su motor de base de datos. Sus principales características son:

- Realiza *fingerprint* de servidores SQL.
- Realiza ataques de fuerza bruta a la cuenta del "sa".

- Escanea puertos **TCP/UDP** del servidor **SQL** que ataca, para encontrar los puertos permitidos por el cortafuegos de la red y utilizarlos para emplear un reverse **Shell**.

· **The Mole:**



Es una herramienta para la explotación automatizada de **inyección SQL** en páginas web. Solo necesita una URL vulnerable y una cadena válida en el sitio y puede detectar la vulnerabilidad y explotarla, ya sea mediante el uso de la técnica de unión o una consulta basada en técnicas booleanas.

**The Mole** utiliza una interfaz basada en comandos, lo que permite al usuario indicar la acción que quiere realizar

fácilmente. Con esta herramienta podemos auditar si un sitio web vulnerable a un ataque de **inyección SQL**, de una forma rápida y totalmente automatizada. Las características principales de **The Mole** son:

- Soporte para **MySQL**, **PostgreSQL**, **SQL Server** y **Oracle**.
- Interfaz de línea de comandos.
- Autocompletado para: comandos, argumentos de comandos y tablas y nombres de columnas en bases de datos.
- Explotación de **inyección SQL** automático utilizando la técnica de unión y técnicas booleanas.
- Explotación de inyección SQL ciega automática.
- Explota inyecciones **SQL** en: **GET**, parámetros de **POST** y **cookies**.
- Intérprete de comandos de gran alcance para simplificar su uso entre más características.

~ Ransomware:

· **Trend Micro Internet Security:**

**Trend Micro** es una de las pocas compañías en anunciar que, si está utilizando la herramienta de seguridad de Internet de la compañía, está seguro contra el ataque de **ransomware WannaCry**.



El software Internet Security de Trend Micro **lo protege de las estafas de correo electrónico y detiene el ransomware** y otras amenazas en línea antes de que lleguen a su PC.

**Trend Micro** afirma que el software bloquea más 250 amenazas diarias para un usuario. Junto con eso, la herramienta de **Trend Micro** trae características adicionales como proteger su privacidad en las redes sociales, restringir a los

niños de varias páginas web y la capacidad de optimizar el rendimiento. **Trend Micro** incluso ofrece una **solución gratuita si su PC ha sido afectada por un ransomware**. Existe la herramienta **Crypto Ransomware File Decryptor** y la herramienta **Lock Screen Ransomware**. Ambas herramientas están disponibles de forma gratuita y pueden detectar e incluso eliminar ciertos **crypto ransomware**.

· **Zemana Antimalware:**

**Zemana Antimalware** detecta y elimina spyware, adware y otros programas maliciosos hardcore diferentes. El software ofrece protección en tiempo real y características adicionales como la limpieza del navegador. Si bien **Zemana Antimalware** está disponible en una versión gratuita, tendrá que obtener la suscripción Premium para obtener protección en tiempo real y protección contra **ransomware**.



- **Malwarebytes:**



**Malwarebytes** está dirigido a *malware* que infecta PC. El software ofrece una detección más inteligente y cuenta con protección especializada contra ***ransomware***. **Malwarebytes** afirma que trae tecnología de próxima generación para proteger sus archivos del ***ransomware***. Gracias a su tecnología ***anti-malware*, anti-spyware y anti-rootkit**, la herramienta también detecta malware en tiempo real y también los elimina. También defiende los navegadores web y otro software que se conecta a Internet. También detecta sitios web falsos e infectados.

- **HitmanPro.Alert:**



Es un software ***anti-malware*** que puede detectar cualquier comportamiento de un ***ransomware*** en su sistema y se asegura de **eliminar o revertir sus efectos**. **HitmanPro.Alert** incluye paquetes en una tecnología ***CryptoGuard*** que pretende neutralizar cualquier ***ransomware*** en cierres en su sistema y restaurar los archivos antes de que se cifren. A diferencia de otros programas ***anti-ransomware***, mantiene un control sobre cualquier comportamiento malicioso para detectar cualquier malware o archivos maliciosos en el sistema. Aparte de eso, el software ofrece escaneado asistido por la nube, protección de la banca en línea y más.

## • 6. Docker



**Docker** es un proyecto de código libre que se ha convertido en uno de los términos de moda por las ventajas que proporciona, entre otros, a los profesionales del desarrollo web y de aplicaciones, o los administradores de sistemas, por la facilidad que supone el trabajar con el concepto de contenedores.

**Docker** está transformando la forma en que se desarrolla, distribuye y ejecuta el software. La ventaja es muy evidente, podemos encapsular todo el entorno de trabajo de manera que los desarrolladores saben que pueden estar trabajando en su servidor local, con la seguridad de que, al llegar el momento de ponerlo en producción, van a estar trabajando en su servidor local, con la seguridad de que, al llegar el momento de ponerlo en producción, van a estar ejecutándose con la misma configuración sobre la que se han hecho todas las pruebas.

De esta forma, vamos a poder reducir los tiempos de testeo y adaptaciones al hardware del que se dispone en el entorno de producción.

Empaque software en *containers* (contenedores) que incluyen en ellos todo lo necesario para que dicho software se ejecute, incluidas librerías. Con Docker se puede implementar y ajustar la escala de aplicaciones de una forma rápida en cualquier entorno con la garantía de que el código se ejecutará. A primera vista se piensa en Docker como una especie de máquina virtual “liviana”, pero en realidad no lo es. En **Docker** lo que se hace es usar las funcionalidades del Kernel para encapsular un sistema, de esta forma el proyecto que corre dentro de él no tendrá conocimiento que está en un contenedor. Los contenedores se encuentran aislados entre sí y se comportarán como máquinas independientes.

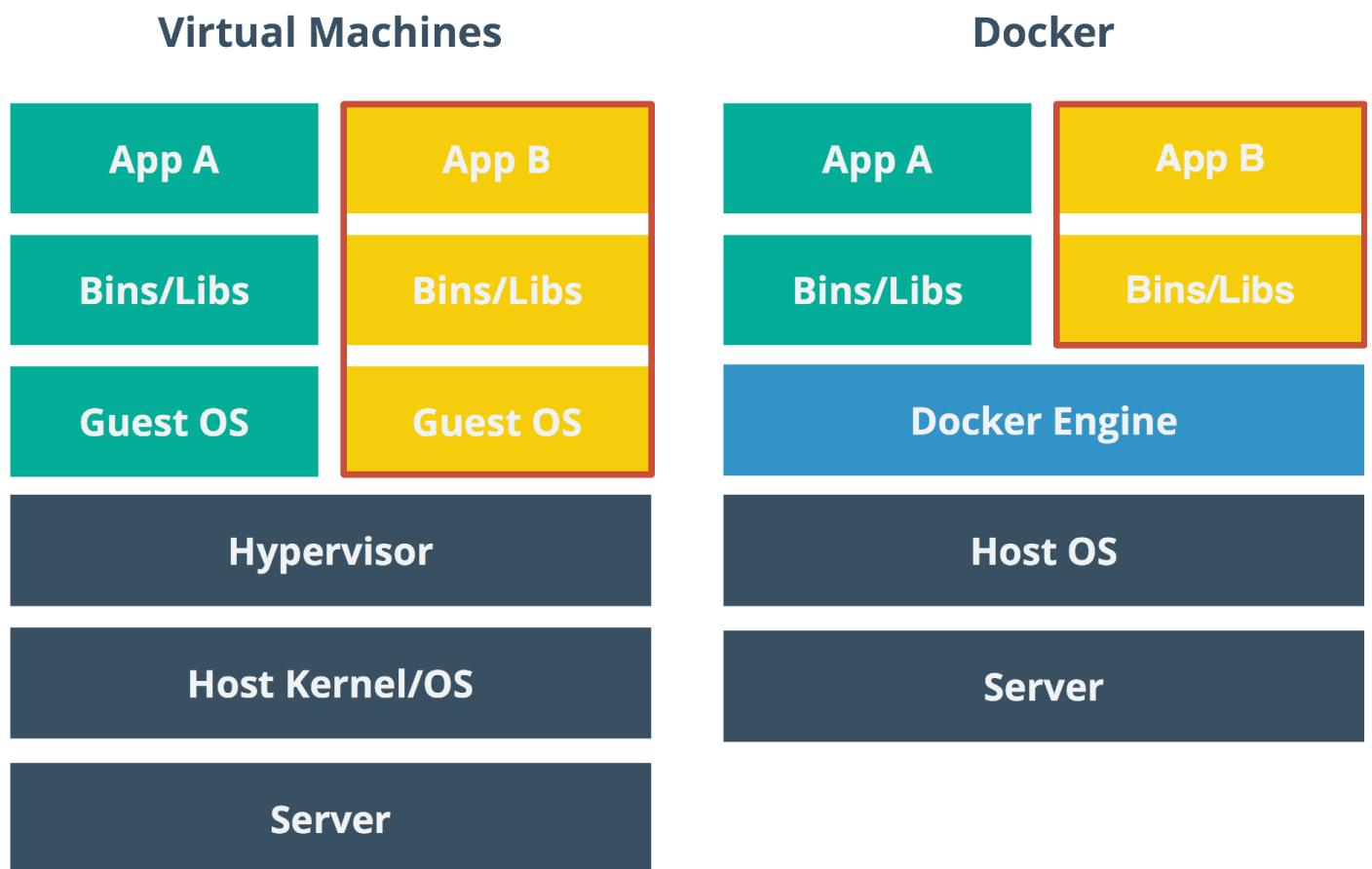
Iniciar un contenedor no tiene un gran impacto a diferencia de iniciar una máquina virtual ya que no tiene que iniciar un sistema operativo completo (desde cero). Gracias al uso de contenedores la demanda de recursos baja limitándose al consumo de la aplicación que contenga. Un contenedor inicia en milisegundos.

Docker trabaja con algo que se llama “**contenedores de Linux**”, estos son un conjunto de tecnologías que juntas forman un contenedor, este conjunto de tecnologías se llama:

- **Namespaces**: permite a la aplicación que corre en un contenedor de Docker tener una vista de los recursos del sistema operativo.
- **Cgroups**: permite limitar y medir los recursos que se encuentran disponibles en el sistema operativo.
- **Chroot**: permite tener en el contenedor una vista de un sistema “falso” para él mismo, es decir, crea su propio entorno de ejecución con su propio **root** y **/home**.

Algunas de las características más notables de un contenedor son:

- Los contenedores son más ligeros (ya que trabajan directamente sobre el *Kernel*) que las máquinas virtuales.
- No es necesario instalar un sistema operativo por contenedor.
- Menor uso de los recursos de la máquina.
- Mayor cantidad de contenedores por equipo físico.
- Mejor portabilidad.



Las diferencias entre Docker y las máquinas virtuales son pocas, aunque, por su naturaleza, se asemejan, pero lo que destaca de entre ellos son la eficiencia y la sencillez.

- Para empezar, los *containers* de **Docker** comparten recursos con el sistema operativo sobre el que se ejecutan. De esta manera podemos arrancar o parar el contenedor rápidamente, mientras que las MVs, como **VMware** o **VirtualBox**, se aislan del sistema operativo sobre el que trabajan y se comunican a través del *hypervisor* (hipervisor).
- La portabilidad de los contenedores hace que los problemas causados por cambiar el entorno donde está corriendo la aplicación se reduzcan a la mínima expresión.
- Si las MVs quieren simular el entorno diferente al nuestro, el *container* de **Docker** se centra en crear la aplicación y que se pueda portar todo el contenido de manera sencilla.

El **container engine** es el responsable de iniciar y parar los contenidos de una manera similar a como lo hace el *hypervisor* en una máquina virtual. Los procesos que corren en el contenido son equivalentes a los procesos nativos en el *host*.

~ **Docker hub:**



**Docker hub** es un servicio de registro de repositorios público en la nube, similar a **Github**, para distribuir los contenidos. Está mantenido por la propia empresa Docker y hay multitud de imágenes, de carácter gratuito, que se pueden descargar y así no tener que hacer el trabajo desde cero al poder aprovechar "plantillas".

Nos permite extraer y enviar imágenes de la ventana acoplable hacia y desde **Github**, donde obtenemos y enviamos nuestro código fuente, pero en el caso de **Docker hub**, descargamos un repositorio en línea basado en la nube que almacena ambos tipos de repositorios, es decir, el repositorio público y el privado. Los repositorios públicos son accesibles para todos, pero el privado es accesible para el propietario interesado de los repositorios. También hay un costo asociado si almacenamos más de un cierto número de repositorios como privado.

**Docker hub** tiene las siguientes características:

- Repositorios de imágenes:

The screenshot shows the Docker Hub interface with a search bar at the top containing 'sql server'. Below the search bar, there are navigation links for 'Explore', 'Repositories', 'Organizations', 'Get Help', and a user profile. The main content area displays search results for 'sql server' with 1 - 25 of 49,786 results. The results include three items: 'Microsoft SQL Server' by Microsoft, 'node' by nodejs, and 'redis' by redis. Each item has a thumbnail, name, publisher information, download count, and star count.

Nos ayuda a encontrar y extraer imágenes de contenedores de **Docker hub**. Y también nos ayuda a enviar imágenes como un repositorio privado o público.

- Equipo y Organizaciones:

The screenshot shows the Docker Hub interface with a search bar at the top containing 'no puedo crear más grupos, hay que pagar...'. Below the search bar, there are navigation links for 'Explore', 'Repositories', 'Organizations', 'Get Help', and a user profile. The main content area shows the 'Teams' section of an organization named 'salscouts'. It includes a team icon, the organization name, a description, and a 'Create Team' button. Below this, there is a table listing teams with columns for 'TEAM', 'DESCRIPTION', and 'MEMBERS'.

TEAM	DESCRIPTION	MEMBERS
owners	Full administrative access to the organization.	1
president	The President, only The President.	0

Nos permite crear grupos de trabajo e impulsar los repositorios como uno privado, que está disponible para su uso únicamente dentro de nuestra organización. De esta forma, hemos gestionado el acceso a nuestros repositorios privados de imágenes de contenedores.

- Integración de GitHub y BitBucket:

Permite la integración con repositorios de código fuente como **GitHub** y **BitBucket**.

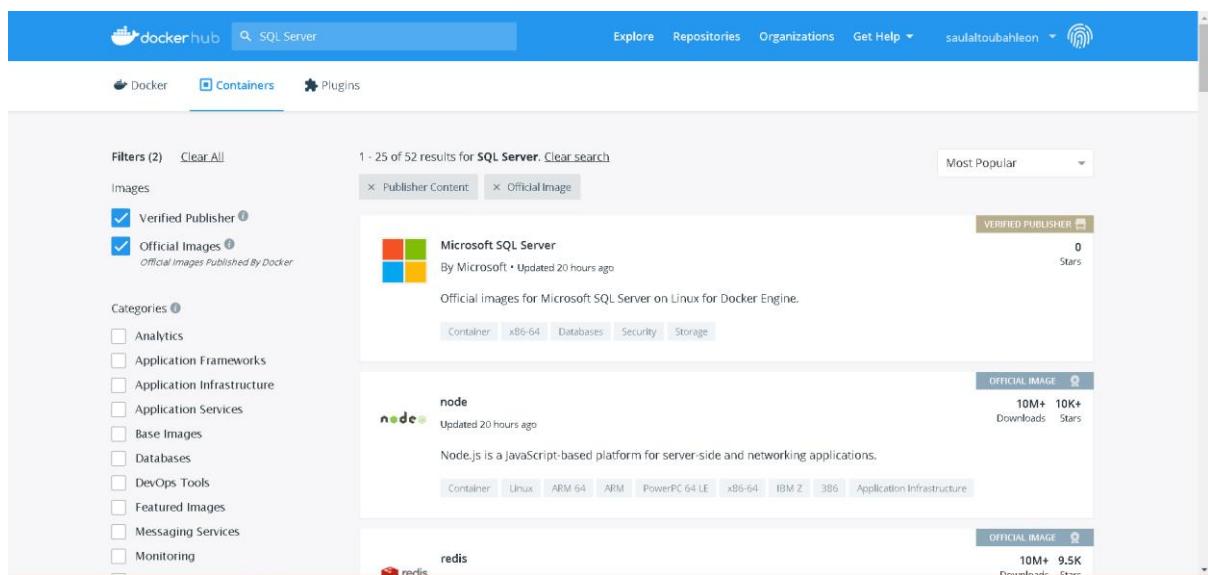
- Construcciones automatizadas:

Si se ha enviado algún cambio en el código fuente a los repositorios de código fuente, automáticamente detecta y crea imágenes de contenedor desde **GitHub** o **Bitbucket** y las envía a **Docker hub**.

- Webhooks:

Una vez que hemos enviado nuestras imágenes con éxito, con la ayuda de un *webhook*, desencadena una acción para integrar **Docker hub** con otros servicios.

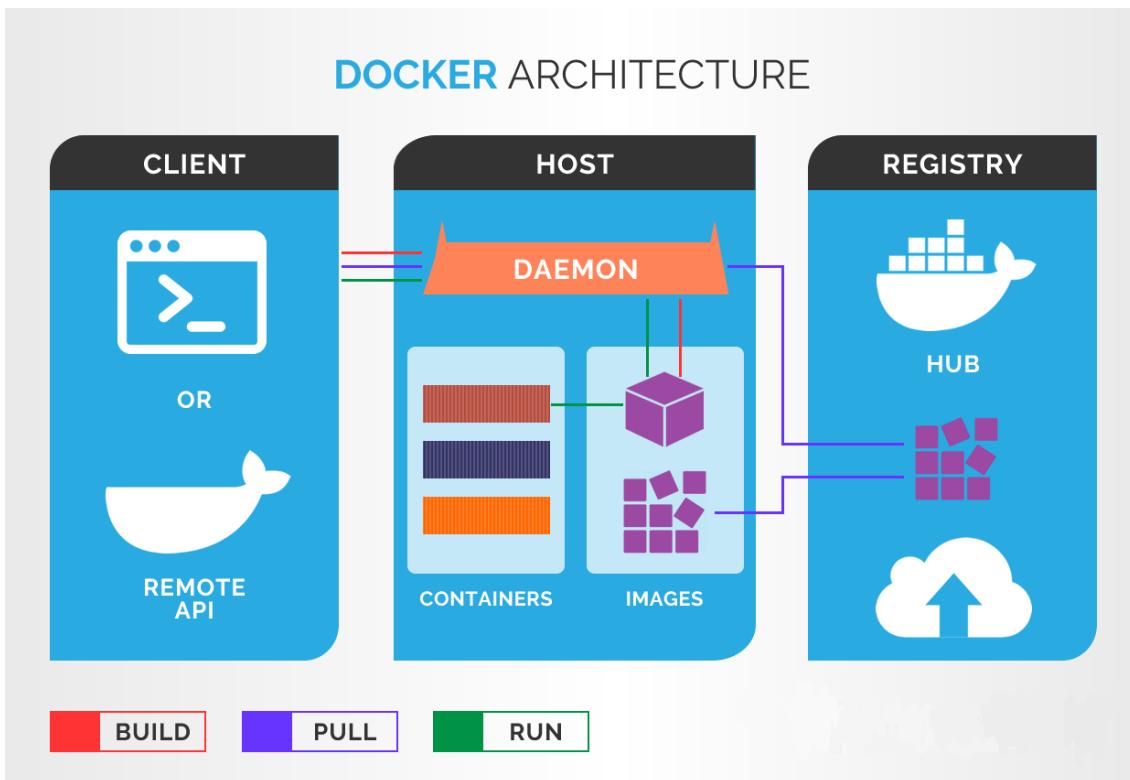
- Imágenes oficiales y del editor:



Las imágenes de alta calidad proporcionadas por los *dockers* se consideran "imágenes oficiales" y se pueden extraer y utilizar. Del mismo modo, las imágenes de alta calidad proporcionadas por proveedores externos son "imágenes del editor", también llamadas imágenes certificadas, que brindan soporte y garantía de compatibilidad con **Docker Enterprise**.

### ~ Arquitectura de Docker:

Docker usa la arquitectura cliente-servidor. El cliente de **Docker** "habla" con el *daemon* (los servicios) de **Docker**, el cual hace el pesado procedimiento de ejecución y procesamiento y distribución de nuestros *containers* de **Docker**. El cliente de **Docker** y el *daemon* pueden ejecutar en el mismo sistema o podemos conectar un cliente **Docker** a un *daemon* remoto.



El cliente y el demonio se comunican mediante una API remota o a través de interfaces de red. Otro cliente de **Docker** es **Docker Compose** que nos permite trabajar con aplicaciones que consisten en un set de contenedores.

### ~ Docker Daemon:

El demonio de **Docker** (`dockerd`) escucha las peticiones de la API y administra los objetos de **Docker** tales como imágenes, contenedores, redes y volúmenes. Un *daemon* puede comunicarse con otros *daemons* para administrar los servicios de **Docker**.

### ~ **Cliente de Docker:**

El cliente de **Docker** (`docker`) es el camino principal que muchos usuarios interactúan con **Docker**. Cuando ejecutamos comandos tales como `docker run`, el cliente envía esos comandos a `dockerd`, y se los lleva. El comando `docker` usa la API de **Docker**. El cliente de **Docker** puede comunicarse con más de un *daemon*.

### ~ **Registros de Docker:**

Los registros de **Docker** almacenan imágenes. Cuando ejecutamos los comandos `docker pull` o `docker run`, las imágenes necesarias son descargadas (*pulled*) desde nuestro registro configurado. Cuando ejecutamos el comando `docker push`, nuestra imagen es enviada (*pushed*) a nuestro registro configurado.

### ~ **Objectos de Docker:**

Cuando usamos **Docker**, estamos creando y usando imágenes, contenedores, redes, volúmenes, *plugins* y otros objetos:

- Imágenes: es una plantilla de solo lectura con instrucciones para crear un contenedor. A menudo, una imagen está basada en otra con unas cuantas personalizaciones. Podemos crear nuestras propias imágenes a partir de un *Dockerfile* con una simple sintaxis para definir los pasos necesarios para crear la imagen y ejecutarla. Cada instrucción dentro de un *Dockerfile* crea una capa en la imagen. Cuando cargamos el *Dockerfile* y reconstruimos la imagen, solo aquellas capas que se han cargado están reconstruidas. Esta es la parte en que hace las imágenes ligeras, pequeñas y rápidas cuando las comparamos con otras tecnologías de virtualización.

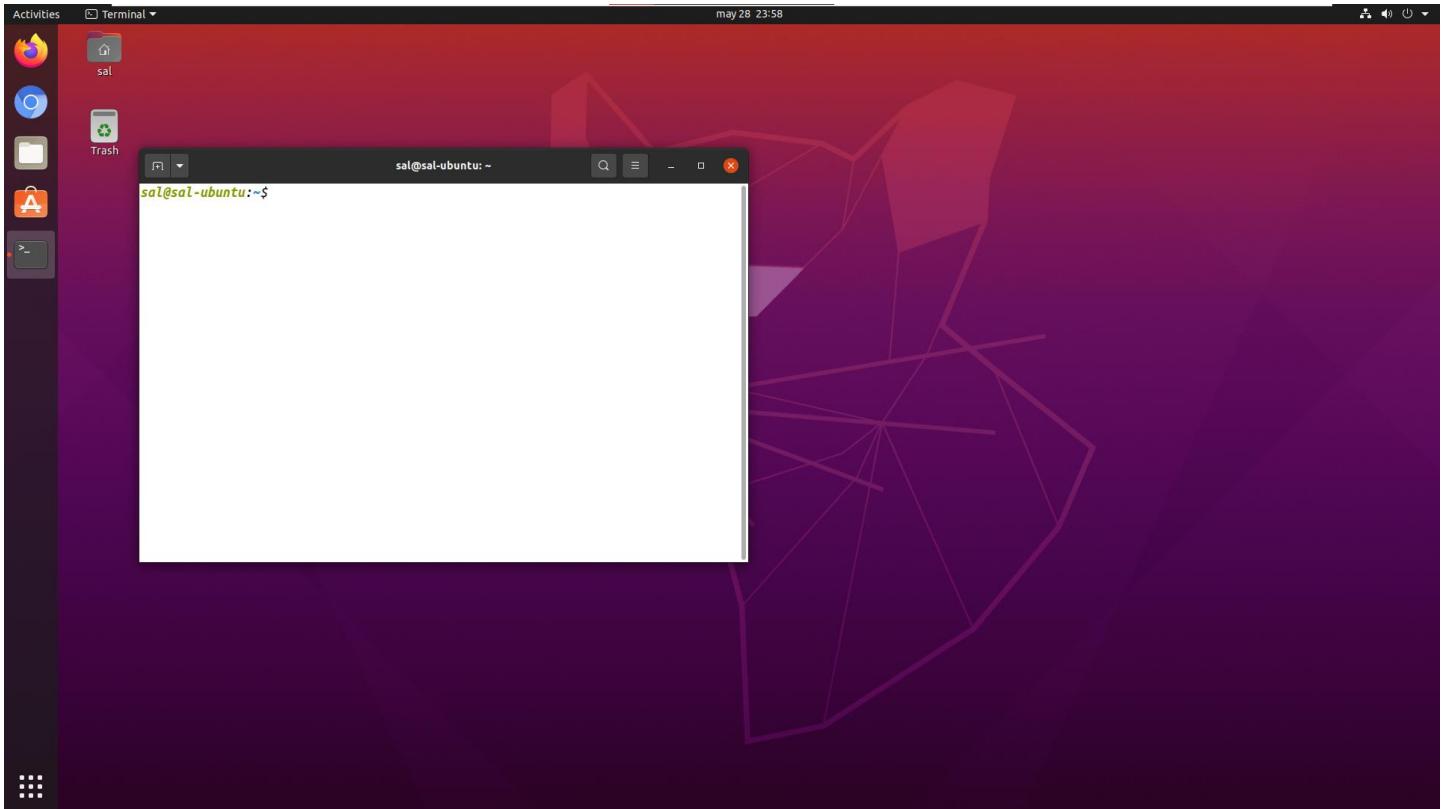
- Containers (contenedores): es una instancia ejecutable de una imagen. Podemos crear, iniciar, detener, mover o borrar un *container* usando la API o la línea de comandos. Cuando conectamos un contenedor a una o más redes, adjuntarle almacenamiento a él o incluso crear una nueva imagen basado en su estado actual.

Por defecto, un *container* está relativamente aislado de otros *containers* y su host. Podemos controlar cómo de aislada está la red de conexión, el almacenamiento o los sistemas inferiores está de los otros *containers* o desde el host.

Un contenedor está definido por su imagen al igual que cualquier opción de configuración que demostremos cuando lo creamos o lo iniciamos. Cuando un contenedor es eliminado, cualquier cambio a su estado que no se encuentre en el almacenamiento desaparece.

## • 6.1. Docker aplicado a BD

En este último apartado vamos a demostrar la instalación de **Docker** en una nueva máquina virtual con el sistema operativo **Ubuntu 20.04 Desktop**.



Es posible que la versión del paquete de instalación de **Docker** disponible en el repositorio oficial de **Ubuntu** no sea la más reciente. Para asegurarnos de contar con la versión más reciente, instalaremos **Docker** desde el repositorio oficial de **Docker**. Para hacerlo, agregaremos una nueva fuente de paquetes y la clave **GPG** de **Docker** para garantizar que las descargas sean válidas, y luego instalaremos el paquete.

```
sal@sal-ubuntu:~$ sudo apt update; sudo apt upgrade -y
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [274 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [323 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal-updates/universe DEP-11 48x48 Icons [204 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24,4 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [58,3 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Hit:13 http://packages.microsoft.com/repos/code stable InRelease
Hit:14 https://packages.microsoft.com/repos/vscode stable InRelease
Fetched 1.219 kB in 9s (135 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sal@sal-ubuntu:~$
```

Primero, actualizamos la lista de paquetes existente, ejecutamos el comando: **sudo apt update; sudo apt upgrade -y**

Después, instalamos los paquetes de requisitos previos para poder descargar a través de HTTPS, ejecutamos el comando: `sudo apt install apt-transport-https ca-certificates curl software-properties-common -y`

```
sal@sal-ubuntu:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20210119~20.04.1).
ca-certificates set to manually installed.
software-properties-common is already the newest version (0.98.9.5).
apt-transport-https is already the newest version (2.0.5).
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 161 kB/395 kB of archives.
After this operation, 1.115 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.5 [161 kB]
Fetched 161 kB in 5s (35.8 kB/s)
Selecting previously unselected package libcurl4:amd64.
(Reading database ... 162841 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.5_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.5) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.5_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.5) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.5) ...
Setting up curl (7.68.0-1ubuntu2.5) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
sal@sal-ubuntu:~$ |
```

Luego, añadimos la clave **GPG** para el repositorio oficial de **Docker** en nuestro sistema, ejecutamos el comando: `curl https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

```
sal@sal-ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
sal@sal-ubuntu:~$ |
```

Agregamos el repositorio de Docker a las fuentes de APT, ejecutamos el comando: `sudo add-apt-repository "deb https://download.docker.com/linux/ubuntu focal stable" [arch=amd64]`

```
sal@sal-ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [41,0 kB]
Hit:6 http://packages.microsoft.com/repos/code stable InRelease
Hit:7 https://packages.microsoft.com/repos/vscode stable InRelease
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [9.335 B]
Fetched 50,3 kB in 31s (1.620 B/s)
Reading package lists... Done
sal@sal-ubuntu:~$ |
```

Nos aseguramos qué última versión de *Docker* instalaremos en nuestro equipo, ejecutamos el comando: `apt-cache policy docker-ce`

```
sal@sal-ubuntu:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.6~3-0~ubuntu-focal
  Version table:
    5:20.10.6~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.5~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.4~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.3~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.2~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.1~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.0~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.15~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.14~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.13~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.12~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.11~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.10~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:19.03.9~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
sal@sal-ubuntu:~$
```

Y por último instalamos Docker, ejecutamos el comando: `sudo apt install docker-ce -y`

```
sal@sal-ubuntu:~$ sudo apt install docker-ce -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 113 MB of archives.
After this operation, 504 MB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57,4 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-lubuntu3.1 [884 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-lubuntu3.1 [4.557 kB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.4.6-1 [28,3 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [74,3 kB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:20.10.6~3-0~ubuntu-focal [41,4 MB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:20.10.6~3-0~ubuntu-focal [24,8 MB]
Get:9 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:20.10.6~3-0~ubuntu-focal [9.067 kB]
Get:10 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.7.0~ubuntu-focal [3.886 kB]
Fetched 113 MB in 1min 9s (1.629 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 162854 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.4.6-1_amd64.deb ...
Unpacking containerd.io (1.4.6-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../2-docker-ce-cli_5%3a20.10.6~3-0~ubuntu-focal_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.6~3-0~ubuntu-focal) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../3-docker-ce_5%3a20.10.6~3-0~ubuntu-focal_amd64.deb ...
Unpacking docker-ce (5:20.10.6~3-0~ubuntu-focal) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../4-docker-ce-rootless-extras_5%3a20.10.6~3-0~ubuntu-focal_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:20.10.6~3-0~ubuntu-focal) ...
Selecting previously unselected package docker-scan-plugin.
Preparing to unpack .../5-docker-scan-plugin_0.7.0~ubuntu-focal_amd64.deb ...
Unpacking docker-scan-plugin (0.7.0~ubuntu-focal) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../6-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../7-git-man_1%3a2.25.1-lubuntu3.1_all.deb ...
Unpacking git-man (1:2.25.1-lubuntu3.1) ...
Selecting previously unselected package git.
Preparing to unpack .../8-git_1%3a2.25.1-lubuntu3.1_amd64.deb ...
Unpacking git (1:2.25.1-lubuntu3.1) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_0.4.3-1_amd64.deb ...
Unpacking slirp4netns (0.4.3-1) ...
Setting up slirp4netns (0.4.3-1) ...
Setting up docker-scan-plugin (0.7.0~ubuntu-focal) ...
Setting up liberror-perl (0.17029-1) ...
Setting up containerd.io (1.4.6-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-ce-cli (5:20.10.6~3-0~ubuntu-focal) ...
Setting up pigz (2.4-1) ...
Setting up git-man (1:2.25.1-lubuntu3.1) ...
Setting up docker-ce-rootless-extras (5:20.10.6~3-0~ubuntu-focal) ...
Setting up docker-ce (5:20.10.6~3-0~ubuntu-focal) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up git (1:2.25.1-lubuntu3.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.6) ...
sal@sal-ubuntu:~$
```

Una vez instalado **Docker**, comprobamos que el *daemon* está activo, ejecutamos el comando: `sudo service docker status`

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-05-29 00:59:24 CEST; 3min 58s ago
     TriggeredBy: docker.socket
     Docs: https://docs.docker.com
 Main PID: 8423 (dockerd)
   Tasks: 12
    Memory: 59.4M
      CGroup: /system.slice/docker.service
             └─ 8423 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

may 29 00:59:23 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:23.49374565+02:00" level=warning msg="Your kernel does not support CPU realtime scheduler"
may 29 00:59:23 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:23.49374565+02:00" level=warning msg="Your kernel does not support cgroup blkio weight"
may 29 00:59:23 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:23.49378500+02:00" level=warning msg="Your kernel does not support cgroup blkio weight_device"
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.494061085+02:00" level=info msg="Loading containers: start."
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.494061085+02:00" level=info msg="Default bridge (dockero) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a preferred IP address"
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.494061085+02:00" level=info msg="Loading containerd: start"
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.494061085+02:00" level=info msg="Docker daemon commit=8728dd2 graphdriver(s)=overlay2 version=20.10.6"
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.818945381+02:00" level=info msg="Daemon has completed initialization"
may 29 00:59:24 sal-ubuntu systemd[1]: Started Docker Application Container Engine
may 29 00:59:24 sal-ubuntu dockerd[8423]: time="2021-05-29T00:59:24.859450965+02:00" level=info msg="API listen on /run/docker.sock"
_
_
```

Por defecto, el comando `docker` solo puede ser ejecutado por el usuario `root` o un usuario del grupo **Docker**, que se crea automáticamente durante el proceso de instalación de **Docker**. Para evitar problemas y tener que escribir `sudo` al ejecutar el comando `docker`, agregamos el nombre de usuario al grupo Docker, ejecutamos el comando: `sudo usermod -aG docker ${USER}`. Y para aplicar la nueva membresía de grupo, ejecutamos el comando: `su - ${USER}`

```
sal@sal-ubuntu:~$ sudo usermod -aG docker ${USER}
sal@sal-ubuntu:~$ su - ${USER}
Password:
sal@sal-ubuntu:~$ |
```

Para comprobar que el usuario se agregó al grupo **Docker**, ejecutamos el comando: `id -nG`

```
sal@sal-ubuntu:~$ id -nG
sal adm cdrom sudo dip plugdev lpadmin lxd sambashare
sal@sal-ubuntu:~$
```

Y con el usuario `root` hacemos lo mismo de antes:

```
root@sal-ubuntu:/home/sal# sudo usermod -aG docker ${USER}
root@sal-ubuntu:/home/sal# id -nG
root docker
root@sal-ubuntu:/home/sal# |
```

Y con esto ya tenemos instalado **Docker** en nuestro **Ubuntu 20.04**. Si ejecutamos el comando `docker` ó `docker --info` nos mostrará la lista de subcomandos seguida de argumentos:

```
sal@sal-ubuntu:~$ docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/home/sal/.docker")
  -c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with
                        "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"/"info"/"warn"/"error"/"fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string   Trust certs signed only by this CA (default "/home/sal/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/home/sal/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default "/home/sal/.docker/key.pem")
  --tlsverify          Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  app*               Docker App (Docker Inc., v0.9.1-beta3)
  builder            Manage builds
  buildx*            Build with BuildKit (Docker Inc., v0.5.1-docker)
  config              Manage Docker configs
  container           Manage containers
  context              Manage contexts
  image               Manage images
  manifest            Manage Docker image manifests and manifest lists
  network             Manage networks
  node                Manage Swarm nodes
  plugin              Manage plugins
  scan*               Docker Scan (Docker Inc., v0.7.0)
  secret              Manage Docker secrets
  service              Manage services
  stack               Manage Docker stacks
  swarm               Manage Swarm
  system              Manage Docker
  trust               Manage trust on Docker images
  volume              Manage volumes

Commands:
  attach              Attach local standard input, output, and error streams to a running container
  build               Build an image from a Dockerfile
  commit              Create a new image from a container's changes
  cp                  Copy files/folders between a container and the local filesystem
  create              Create a new container
  diff                Inspect changes to files or directories on a container's filesystem
  events              Get real time events from the server
  exec                Run a command in a running container
  export              Export a container's filesystem as a tar archive
  history             Show the history of an image
  images              List images
  import              Import the contents from a tarball to create a filesystem image
  info                Display system-wide information
  inspect             Return low-level information on Docker objects
  kill                Kill one or more running containers
  load                Load an image from a tar archive or STDIN
  login               Log in to a Docker registry
```

Para saber qué versión tenemos instalada de Docker ejecutamos el comando: `docker -v` ó `docker --version`

```
sal@sal-ubuntu:~$ docker -v; docker --version
Docker version 20.10.6, build 370c289
Docker version 20.10.6, build 370c289
sal@sal-ubuntu:~$ |
```

Para verificar si podemos acceder a imágenes y descargarlas de **Docker Hub**, probamos ejecutando el comando `docker run hello-world`:

```
root@sal-ubuntu:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:5122f6204b6a3596e048758cabba3c46b1c937a46b5be6225b835d091b90e46c
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

root@sal-ubuntu:~# |

Inicialmente, Docker no pudo encontrar la imagen de `hello-world` a nivel local. Por ello la descargó de **Docker Hub**, siendo el repositorio predeterminado. Una vez descargó la imagen, **Docker** creó un contenedor a partir de ella y de la aplicación dentro del contenedor ejecutado y mostró el mensaje.

Para buscar imágenes disponibles en **Docker Hub** usando ejecutando el comando `docker search`, por ejemplo, vamos a buscar la imagen de **Ubuntu**:

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	12299	[OK]	[OK]
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	534	[OK]	[OK]
websphere-liberty	WebSphere Liberty multi-architecture images ...	273	[OK]	[OK]
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offici...	251	[OK]	[OK]
consol/ubuntu-xfce-vnc	Ubuntu container with "headless" VNC session...	240	[OK]	[OK]
ubuntu-upstart	Upstart is an event-based replacement for th...	110	[OK]	[OK]
ansible/ubuntu14.04-ansible	Ubuntu 14.04 LTS with ansible	98	[OK]	[OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5	ubuntu-16-nginx-php-phpmyadmin-mysql-5	50	[OK]	[OK]
open-liberty	Open Liberty multi-architecture images based...	45	[OK]	[OK]
ubuntu-debootstrap	debootstrap --variant=minbase --components=...	44	[OK]	[OK]
i386/ubuntu	Ubuntu is a Debian-based Linux operating sys...	25	[OK]	[OK]
nuagebec/ubuntu	Simple always updated Ubuntu docker images w...	24	[OK]	[OK]
1and1internet/ubuntu-16-apache-php-5.6	ubuntu-16-apache-php-5.6	14	[OK]	[OK]
1and1internet/ubuntu-16-apache-php-7.0	ubuntu-16-apache-php-7.0	13	[OK]	[OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10	ubuntu-16-nginx-php-phpmyadmin-mariadb-10	11	[OK]	[OK]
1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4	ubuntu-16-nginx-php-5.6-wordpress-4	9	[OK]	[OK]
darksheer/ubuntu	Base Ubuntu Image -- Updated hourly	5	[OK]	[OK]
pivotaldata/ubuntu	A quick freshening-up of the base Ubuntu doc...	4	[OK]	[OK]
1and1internet/ubuntu-16-nginx-php-7.0	ubuntu-16-nginx-php-7.0	4	[OK]	[OK]
1and1internet/ubuntu-16-nginx-php-7.1-wordpress-4	ubuntu-16-nginx-php-7.1-wordpress-4	3	[OK]	[OK]
owncloud/ubuntu	ownCloud Ubuntu base image	3	[OK]	[OK]
smartentry/ubuntu	ubuntu with smartentry	1	[OK]	[OK]
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development	1	[OK]	[OK]
pivotaldata/ubuntu16.04-test	Ubuntu 16.04 image for GPDB testing	0	[OK]	[OK]
1and1internet/ubuntu-16-rspec	ubuntu-16-rspec	0	[OK]	[OK]
root@sal-ubuntu:~#				

La secuencia de comandos rastreará **Docker Hub** y mostrará una lista de todas las imágenes cuyo nombre coincide con la cadena de búsqueda.

En la columna **OFFICIAL**, **OK** indica una imagen creada y avalada por la empresa responsable del proyecto. Una vez que identifique la imagen que desearía usar, podemos descargarla ejecutando el comando `docker pull ubuntu`:

root@sal-ubuntu:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
345e3491a907: Pull complete
57671312ef6f: Pull complete
5e9250ddb7d0: Pull complete
Digest: sha256:adf73ca014822ad8237623d388cedf4d5346aa72c270c5acc01431cc93e18e2d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@sal-ubuntu:~#

Para revisar qué imágenes tenemos descargadas, ejecutamos el comando: `docker images`

```
root@sal-ubuntu:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest   7e0aa2d69a15  5 weeks ago  72.7MB
hello-world     latest   d1165f221234  2 months ago  13.3kB
root@sal-ubuntu:~# |
```

Ahora vamos a ejecutar un contenedor usando la imagen más reciente de Ubuntu, ejecutamos el comando `docker run -it ubuntu`

```
root@sal-ubuntu:~# docker run -it ubuntu
root@b5ffb332a315:/# |
```

Estamos dentro del contenedor de **Ubuntu** porque en vez de ver el *hostname* del equipo, vemos el ID del contenedor.

Para saber qué contenedores tenemos en funcionamiento, ejecutamos el comando `docker ps`:

```
root@sal-ubuntu:~# docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
root@sal-ubuntu:~# |
```

Como salimos del contenedor antes, pues ahora mismo está inactivo. Para saber qué contenedores tenemos en funcionamiento y cuáles están inactivos, ejecutamos el comando `docker ps -a`

```
root@sal-ubuntu:~# docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
621632ad7206      ubuntu      "/bin/bash"  About a minute ago  Exited (127) About a minute ago
b5ffb332a315      ubuntu      "/bin/bash"  8 minutes ago    Exited (0) 2 minutes ago
8fa662336afe      hello-world  "/hello"    47 minutes ago   Exited (0) 47 minutes ago
root@sal-ubuntu:~# |
```

Para iniciar un *container* detenido, ejecutamos el comando `docker start` seguido del ID del *container* o del nombre del *container*

```
root@sal-ubuntu:~# docker start 621632ad7206  
621632ad7206  
root@sal-ubuntu:~# |
```

Y comprobamos con `docker ps` para ver su estado

```
root@sal-ubuntu:~# docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
621632ad7206 ubuntu "/bin/bash" 7 minutes ago Up 30 seconds |  
root@sal-ubuntu:~# |
```

Para detener un *container* en funcionamiento, ejecutamos el comando Docker stop seguido del ID del *container* o del nombre del *container*

```
root@sal-ubuntu:~# docker stop festive_cohen  
festive_cohen  
root@sal-ubuntu:~# docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
621632ad7206 ubuntu "/bin/bash" 10 minutes ago Exited (0) 13 seconds ago |  
b5ffb332a315 ubuntu "/bin/bash" 17 minutes ago Exited (0) 11 minutes ago |  
8fa662336afe hello-world "/hello" 55 minutes ago Exited (0) 55 minutes ago |  
root@sal-ubuntu:~# |
```

Una vez que decidimos no necesitar más de un contenedor, ejecutamos el comando `docker rm` seguido del ID o del nombre del contenedor

```
root@sal-ubuntu:~# docker rm festive_cohen  
festive_cohen  
root@sal-ubuntu:~# |
```

Los contenedores pueden convertirse en imágenes que podremos usar para crear contenedores nuevos.

## ~ Contenedor con MySQL Server:

Vamos a crear un contenedor con **MySQL Server** sin persistencia de datos. Cuando un contenedor Docker que no tiene persistencia de datos quiere decir que cuando finalice la ejecución perderá todo el contenido que hayamos creado durante la ejecución, es decir, si durante la ejecución del contenedor hemos creado varias veces bases de datos en **MySQL Server**, éstas se perderán cuando el contenedor se detenga.

Para crear el contenedor con la imagen de **MySQL Server** ejecutaremos el comando:  
`docker run -d --rm --name SAL_mysql -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 mysql`

```
root@sal-ubuntu:~# docker run -d --rm --name SAL_mysql -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
69692152171a: Pull complete
1651b0be3df3: Pull complete
951da7386bc8: Pull complete
0f86c95aa242: Pull complete
37ba2dbbd4fe: Pull complete
6d278bb05e94: Pull complete
497efbd93a3e: Pull complete
f7fddfb10c2c2: Pull complete
16415d159dfb: Pull complete
0e530ffc6b73: Pull complete
b0a4a1a77178: Pull complete
cd90f92aa9ef: Pull complete
Digest: sha256:d50098d7fc25b1fc24e2d3247cae3fc55815d64fec640dc395840f8fa80969
Status: Downloaded newer image for mysql:latest
8e5b8c221af7fc10cc42af72d679938aa0f4ae527c937b10355b679601e5c159
root@sal-ubuntu:~# docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
mysql              latest        c0cdc95609f1   2 weeks ago       556MB
root@sal-ubuntu:~# docker ps
CONTAINER ID        IMAGE           COMMAND            CREATED            STATUS              PORTS               NAMES
8e5b8c221af7        mysql           "docker-entrypoint.s..."   13 seconds ago    Up 12 seconds      0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   SAL_mysql
root@sal-ubuntu:~#
```

- Con el parámetro `-d` nos permite ejecutar el contenedor en modo *detached*, es decir, en *background* (en segundo plano).
- Con el parámetro `--rm` hace que cuando salgamos del contenedor, ése se elimine y no ocupe espacio en nuestro almacenamiento interno.
- Con el parámetro `--name` nos permite asignarle un nombre a nuestro contenedor. Si no le asignamos uno, **Docker** nos generará un nombre automáticamente.
- El parámetro `-e` es para pasarle al contenedor una variable de entorno (*environment*). En este caso, estamos pasando la variable de entorno `MYSQL_ROOT_PASSWORD` con el valor de la contraseña que tendrá el usuario `root` para **MySQL Server**.
- El parámetro `-p` nos permite mapear los puertos entre nuestra máquina local y el contenedor. En este caso, estamos mapeando el puerto de **MySQL 3306/TCP** de nuestra máquina local con el puerto **3306/TCP** del contenedor.

Y ahora vamos a abrir un terminal en el contenedor para interaccionar con él, ejecutamos el comando: `docker exec -it SAL_mysql /bin/bash`

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sal_scouts     |
| sys            |
+-----+
5 rows in set (0.01 sec)

mysql> use sal_scouts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_sal_scouts |
+-----+
| sal_donacion        |
| sal_entidad          |
| sal_factura_material |
| sal_personal         |
| sal_scouter          |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

Iniciamos sesión con **MySQL** a través del terminal y hemos creado nuestra base de datos con alguna tabla.

Salimos del contenedor con `exit` y lo detenemos, no es necesario borrar el contenedor porque con la opción `--rm` provoca que al salir se borre automáticamente.

```
root@sal-ubuntu:/home/sal/Desktop# docker stop SAL_mysql
SAL_mysql
root@sal-ubuntu:/home/sal/Desktop# docker ps
CONTAINER ID   IMAGE    COMMAND   CREATED      STATUS      PORTS      NAMES
root@sal-ubuntu:/home/sal/Desktop# docker ps -a
CONTAINER ID   IMAGE    COMMAND   CREATED      STATUS      PORTS      NAMES
b5ffb332a315   7e0aa2d69a15   "/bin/bash"   14 hours ago  Exited (0) 14 hours ago
8fa662336afe   d1165f221234   "/hello"     15 hours ago  Exited (0) 15 hours ago
root@sal-ubuntu:/home/sal/Desktop#
```

Ahora vamos a hacer lo mismo, pero esta vez vamos a crear el contenedor con persistencia de datos, para ello vamos a crear un volumen de tipo *bind mount* donde montamos un directorio en nuestra máquina virtual local en un directorio dentro del contenedor. Ejecutamos el comando: `docker volume create SAL_mysql_data`

```
root@sal-ubuntu:~# docker volume create SAL_mysql_data
SAL_mysql_data
root@sal-ubuntu:~# |
```

Creado el volumen vamos mirar la lista de volúmenes de **Docker**, ejecutamos el comando `docker volume ls`:

```
root@sal-ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local        SAL_mysql_data
root@sal-ubuntu:~# |
```

Y ahora volvemos a crear de vuelta el contenedor con otro nombre e indicando la librería que hemos creado, ejecutamos el comando: `docker run -d --name SAL_mysql_vol -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 -v SAL_mysql_data:/var/lib/mysql mysql`

```
root@sal-ubuntu:~# docker run -d --name SAL_mysql_vol -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 -v SAL_mysql_data:/var/lib/mysql mysql
7b071dd31d8bf0b252cd88ed0345c5139ff25921f94aa8adb1420e945cbe911a
root@sal-ubuntu:~# docker ps
CONTAINER ID   IMAGE       COMMAND           CREATED          STATUS          PORTS          NAMES
7b071dd31d8b   mysql      "docker-entrypoint.s..."  33 seconds ago   Up 29 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   SAL_mysql_vol
root@sal-ubuntu:~# docker ps -a
CONTAINER ID   IMAGE       COMMAND           CREATED          STATUS          PORTS          NAMES
7b071dd31d8b   mysql      "docker-entrypoint.s..."  35 seconds ago   Up 31 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   SAL_mysql_vol
b5ffb332a315   7e0aa2d69a15   "/bin/bash"        16 hours ago    Exited (0) 16 hours ago
8fa662336afe   d1165f221234   "/hello"         17 hours ago    Exited (0) 17 hours ago
root@sal-ubuntu:~|
```

Abrimos un terminal en el contenedor para interaccionar con él, ejecutamos el comando:

```
docker exec -it SAL_mysql_vol /bin/bash
```

```
root@sal-ubuntu:~# docker exec -it SAL_mysql_vol /bin/bash
root@7b071dd31dbb:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database          |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sal_scouts         |
| sys                |
+--------------------+
5 rows in set (0.05 sec)

mysql> use sal_scouts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------+
| Tables_in_sal_scouts |
+----------------+
| sal_donacion        |
| sal_entidad          |
| sal_factura_material |
| sal_personal         |
| sal_scouter          |
+----------------+
5 rows in set (0.00 sec)

mysql> |
```

Le introducimos de nuevo nuestra base de datos y hacemos alguna que otra consulta. Salimos del contenedor y comprobamos que está en ejecución:

```
root@sal-ubuntu:~# docker ps
CONTAINER ID   IMAGE     COMMAND           CREATED      STATUS      PORTS          NAMES
7b071dd31dbb   mysql    "docker-entrypoint.s..."  9 minutes ago   Up 9 minutes   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   SAL_mysql_vol
root@sal-ubuntu:~# |
```

Detenemos el contenedor y volvemos a crear otro nuevo utilizando el volumen y le ponemos un nombre diferente, ejecutamos el comando: `docker run -d --rm --name SAL_mysql_vol2 -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 -v SAL_mysql_data:/var/lib/mysql mysql`

```
root@sal-ubuntu:~# docker run -d --rm --name SAL_mysql_vol2 -e MYSQL_ROOT_PASSWORD=Abcd1234. -p 3306:3306 -v SAL_mysql_data:/var/lib/mysql mysql
d68ed97d10b6e19a48244920a2546f2835e11002c733476cc092f222b33e5bd2
root@sal-ubuntu:~# |
```

Abrimos el terminal en el nuevo contenedor y comprobamos que nuestra base de datos sigue permaneciendo:

```
root@sal-ubuntu:~# docker exec -it SAL_mysql_vol2 /bin/bash
root@d68ed97d10b6:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sal_scouts     |
| sys            |
+-----+
5 rows in set (0.00 sec)

mysql> use sal_scouts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_sal_scouts |
+-----+
| sal_donacion        |
| sal_entidad          |
| sal_factura_material |
| sal_personal         |
| sal_scouter          |
+-----+
5 rows in set (0.01 sec)
```

## ~ **Adminer** **PhpMyAdmin** **Wordpress**:

Ahora vamos a crear un nuevo contenedor **MySQL** empleando el volumen del contenedor anterior que habíamos creado y crearemos otros dos contenedores que tienen dos SGBD llamados **Adminer** y **PhpMyAdmin**. La conexión entre contenedores lo haremos de dos maneras: una con el uso de banderas con `flag --link` y otra con *user-defined bridge network* (red puente definida por el usuario).

Primero preparamos el entorno borrando todos los contenedores existentes, ejecutamos el comando: `docker rm $(docker ps -qa)`

```
root@sal-ubuntu:~# docker ps -a
CONTAINER ID   IMAGE    COMMAND          CREATED     STATUS      PORTS          NAMES
d68ed97d10b6   mysql   "docker-entrypoint.s..."  35 minutes ago   Up 35 minutes   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   SAL_mysql_v02
b5ff332a15   7e0aa2d69a15   "/bin/bash"   17 hours ago   Exited (0) 17 hours ago
8fa662336afe  d1165f221234   "/hello"    18 hours ago   Exited (0) 18 hours ago
root@sal-ubuntu:~# docker rm $(docker ps -qa)
b5ff332a15
8fa662336afe
Error response from daemon: You cannot remove a running container d68ed97d10b6e19a48244920a2546f2835e11002c733476cc092f222b33e5bd2. Stop the container before attempting removal or force remove
root@sal-ubuntu:~# |
```

Nos borra los contenedores, pero nos da error en el contenedor que creamos antes porque está en funcionamiento, para forzar el cierre indicamos en el comando anterior el parámetro `-f`:

```
root@sal-ubuntu:~# docker rm -f $(docker ps -qa)
d68ed97d10b6
root@sal-ubuntu:~# |
```

Ahora ya podemos crear los contenedores, empezamos con el contenedor de **MySQL** para **Adminer**, ejecutamos el comando: `docker run -d --name mysqladminer -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Abcd1234. -v SAL_mysql_data:/var/lib/mysql mysql`

```
root@sal-ubuntu:~# docker run -d --name mysqladminer -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Abcd1234. -v SAL_mysql_data:/var/lib/mysql mysql
4c18d7f4c0d27f93229b09e0e59defc1d750a0023a74e8e77467c2fb656764aa
root@sal-ubuntu:~# |
```

Creado el contenedor de **MySQL**, ahora creamos el de Adminer y lo enlazamos, ejecutamos el comando: `docker run -d --rm --link mysqladminer -p 8080:8080 adminer`

```
root@sal-ubuntu:~# docker run -d --rm --link mysqladminer -p 8080:8080 adminer
80f4a4ff08ae7ec915a46190fda1bc9775ddcc57dc620b17782aef4fd0b7fdc2
root@sal-ubuntu:~# |
```

Comprobamos que los contenedores estén en ejecución:

```
root@sal-ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
80f4a4ff08ae        adminer            "entrypoint.sh docke..."   40 seconds ago    Up 37 seconds     0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
4c18d7f4cd2         mysql              "docker-entrypoint.s..."   4 minutes ago     Up 4 minutes      0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
root@sal-ubuntu:~#
```

Observamos que funcionan y ahora vamos a comprobar su funcionamiento en un buscador web con la URL: `localhost:8080`

The screenshot shows a web browser window with the following details:

- Title Bar:** Login - Adminer
- Address Bar:** localhost:8080
- Left Sidebar:** Adminer 4.8.0 4.8.1  
(MySQL) root@mysqladminer - sal\_scou
- Right Main Area:**
  - Login Form Fields:**

System	MySQL
Server	mysqladminer
Username	root
Password	[Redacted]
Database	sal_scouts
  - Buttons:** Login, Permanent login

Esta es la página de inicio de sesión de **Adminer a MySQL**, vamos a iniciar sesión con el servidor que será el nombre de nuestro contenedor, el usuario será `root` y la base de datos con la que iniciaremos será la nuestra ("en versión ligera"), **SAL\_SCOUTS** y este será su aspecto:

Database: sal\_scouts - my + [localhost:8080/?server=mysqladminer&username=root&db=sal\\_scouts](#)

Language: English MySQL » mysqladminer » Database: sal\_scouts

**Adminer 4.8.0 4.8.1**

DB: sal\_scouts [Import](#) [Export](#) [Create table](#)

```
select sal_donacion
select sal_entidad
select sal_factura_material
select sal_personal
select sal_scouter
```

**Tables and views**

Search data in tables (5)

<input type="checkbox"/>	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
<input type="checkbox"/>	sal_donacion	InnoDB	utf8_general_ci	81,920	32,768	0		~ 1,000	TRIAL
<input type="checkbox"/>	sal_entidad	InnoDB	utf8_general_ci	1,589,248	212,992	2,097,152		~ 10,000	TRIAL
<input type="checkbox"/>	sal_factura_material	InnoDB	utf8_general_ci	1,589,248	163,840	2,097,152		~ 9,994	TRIAL
<input type="checkbox"/>	sal_personal	InnoDB	utf8_general_ci	1,572,864	0	0		~ 11	TRIAL
<input type="checkbox"/>	sal_scouter	InnoDB	utf8_general_ci	16,384	0	0		~ 1	TRIAL
<b>5 in total</b>		InnoDB	utf8mb4_0900_ai_ci	4,849,664	409,600	0			

Selected (0) [Analyze](#) [Optimize](#) [Check](#) [Repair](#) [Truncate](#) [Drop](#)

Move to other database: sal\_scouts [Move](#) [Copy](#)  overwrite

[Create table](#) [Create view](#)

**Routines**

[Create procedure](#) [Create function](#)

**Events**

[Create event](#)

Select: sal\_donacion - my + [localhost:8080/?server=mysqladminer&username=root&db=sal\\_scouts&select=sal\\_donacion](#)

Language: English MySQL » mysqladminer » sal\_scouts » Select: sal\_donacion

**Adminer 4.8.0 4.8.1**

DB: sal\_scouts [Import](#) [Export](#) [Create table](#)

```
select sal_donacion
select sal_entidad
select sal_factura_material
select sal_personal
select sal_scouter
```

**Select: sal\_donacion**

Select data Show structure Alter table New item

Select Search Sort Limit  Text length  Action

**SELECT \* FROM `sal\_donacion` LIMIT 50 (0.002 s)** [Edit](#)

<input type="checkbox"/>	id_donacion	fecha_donacion	nombre_donante	importe	sal_eventos_id_evento	sal_tesorero_id_personal	trial320
<input type="checkbox"/>	1	2017-08-19	Emalee Overington	2448.48	3	8	T
<input type="checkbox"/>	2	2017-04-06	Tyson Broek	4280.17	1	5	T
<input type="checkbox"/>	3	2017-05-31	Halli Selbach	4498.48	5	5	T
<input type="checkbox"/>	4	2017-02-24	Heriberto Gibson	5178.37	3	8	T
<input type="checkbox"/>	5	2017-08-10	Bidget O'Kieran	1283.37	3	9	T
<input type="checkbox"/>	6	2017-05-10	Nichols Grindall	4882.77	3	5	T
<input type="checkbox"/>	7	2018-01-06	Susy Janusz	3627.8	5	9	T
<input type="checkbox"/>	8	2018-01-19	Kelby Springle	5918.28	5	9	T
<input type="checkbox"/>	9	2017-05-31	Winni Keizman	6262.68	4	5	T
<input type="checkbox"/>	10	2017-10-31	Tobye Stenhouse	5197.99	3	9	T
<input type="checkbox"/>	11	2017-10-10	Blisse Hart	5596.73	1	5	T
<input type="checkbox"/>	12	2017-07-11	Benetta Skepper	2226.02	3	9	T
<input type="checkbox"/>	13	2017-09-12	Randee Giurio	9920.08	5	5	T
<input type="checkbox"/>	14	2017-03-13	Vikky Kaveney	9265.35	5	7	T
<input type="checkbox"/>	15	2017-10-25	Truda Buessen	4117.95	3	9	T
<input type="checkbox"/>	16	2017-06-23	Analiese Maynell	7610.1	2	9	T
<input type="checkbox"/>	17	2017-05-04	Celinda McCaskill	4127.73	3	7	T

Page [1](#) [2](#) [3](#) [4](#) [5](#) ... [20](#) Whole result [Modify](#) Selected (0) [Export \(1,000\)](#)  1,000 rows [Save](#) [Edit](#) [Clone](#) [Delete](#)

Comprobado todo, vamos a preparar el entorno para **MySQL** con **PhpMyAdmin**, pero antes forzamos la eliminación de los contenedores:

```
root@sal-ubuntu:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
80f4a4ff08ae adminer "entrypoint.sh docke..." 19 minutes ago Up 19 minutes 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp bold_lovelace
4c18d7f4c0d2 mysql "docker-entrypoint.s..." 23 minutes ago Up 23 minutes 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp mysqladminer
root@sal-ubuntu:~# docker rm -f $(docker ps -qa)
80f4a4ff08ae
4c18d7f4c0d2
root@sal-ubuntu:~# |
```

Vamos a crear nuestra user-defined bridge network para los contenedores y comprobamos su existencia, ejecutamos el comando: `docker network create SALnet;` `docker network ls`

```
root@sal-ubuntu:~# docker network create SALnet; docker network ls
b8c6b42105169990861f036e9f991462a2f045b31dab2b2fb8a57f3c5c6c4a0b
NETWORK ID      NAME      DRIVER      SCOPE
b8c6b4210516    SALnet    bridge      local
378094eeb780    bridge    bridge      local
a0bc0704888e    host      host       local
1101d330a414    none      null       local
root@sal-ubuntu:~#
```

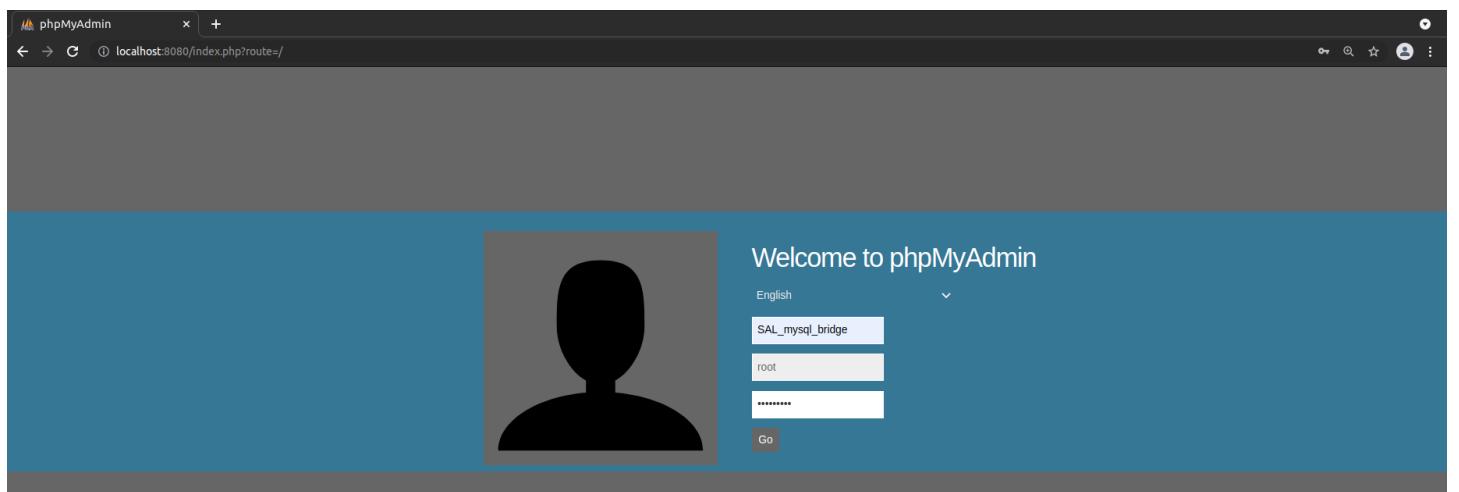
Creamos el contenedor **MySQL** con la red puente y con nuestro volumen primero, ejecutamos el comando: `docker run -d --rm --name SAL_mysql_bridge --network SALnet -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Abcd1234. -v SAL_mysql_data:/var/lib/mysql mysql`

```
root@sal-ubuntu:~# docker run -d --rm --name SAL_mysql_bridge --network SALnet -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Abcd1234. -v SAL_mysql_data:/var/lib/mysql mysql
9320ceed826ea52eb6da73617fe0e54d35cf284feb7e7b8fa64ddf0aba6b4f3e
root@sal-ubuntu:~#
```

Y segundo, creamos el contenedor de **PhpMyAdmin** y para que nos inicie en la página principal pasamos la variable de entorno `PMA_ARBITRARY=1`, ejecutamos el comando: `docker run -d --rm --network SALnet -e PMA_ARBITRARY=1 -p 8080:80 phpmyadmin/phpmyadmin`

```
root@sal-ubuntu:~# docker run -d --rm --network SALnet -e PMA_ARBITRARY=1 -p 8080:80 phpmyadmin/phpmyadmin
Unable to find image 'phpmyadmin/phpmyadmin:latest' locally
latest: Pulling from phpmyadmin/phpmyadmin
6f28985ad184: Pull complete
db883aae18bc: Pull complete
ffae70ea03a9: Pull complete
1e8027612378: Pull complete
3ec32e53dce5: Pull complete
3bb74037bf77: Pull complete
feda0fdbd85b1: Pull complete
b2244185b327: Pull complete
8852ae668073: Pull complete
985e21deb66e: Pull complete
f262da4e7afa: Pull complete
157f3d683e13: Pull complete
990684a56233: Pull complete
1a17ee268b78: Pull complete
1d4c31287d81: Pull complete
cebcede43b15c: Pull complete
887594e7bbfa: Pull complete
b0f7e76820a4: Pull complete
Digest: sha256:8911fb0cfef21dc9fb385ad02cc3254179cd7df87bab3d3a6fa04d1f0549463f
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
2a4538f8ae82300b3a04e2b1a44b2876e23f21369fba9e999d8c05a0461f3680
root@sal-ubuntu:~# |
```

Creado ya el contenedor, pues ahora nos vamos a un navegador web y nos conectamos a la base de datos con la URL: **localhost:8080**



**\*tiene aspecto diferente porque se comprobó con antelación la conexión y se cambió el tema de PhpMyAdmin)**

localhost:8080/SAL\_my... +

localhost:8080/index.php?route=/&route=%2F

PHPMYADMIN Server: SAL\_mysql\_bridge

**General settings**

- Change password
- Server connection collation: utf8mb4\_unicode\_ci
- More settings

**Database server**

- Server: SAL\_mysql\_bridge via TCP/IP
- Server type: MySQL
- Server connection: **SSL is not being used**
- Server version: 8.0.25 - MySQL Community Server - GPL
- Protocol version: 10
- User: root@172.18.0.3
- Server charset: UTF-8 Unicode (utf8mb4)

**Appearance settings**

- Language: English
- Theme: Metro Scheme: win

**Web server**

- Apache/2.4.38 (Debian)
- Database client version: libmysql - mysqld 7.4.16
- PHP extension: mysqli curl mbstring
- PHP version: 7.4.16

The phpMyAdmin configuration storage is not completely configured, some extended features have been deactivated. [Find out why.](#) Or alternately go to 'Operations' tab of any database to set it up there.

**phpMyAdmin**

- Version information: 5.1.0 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

localhost:8080/SAL\_my... +

localhost:8080/index.php?route=/database/structure&server=1&db=sal\_scouts

PHPMYADMIN Server: SAL\_mysql\_bridge > Database: sal\_scouts

**STRUCTURE** SQL SEARCH QUERY EXPORT IMPORT OPERATIONS PRIVILEGES ROUTINES EVENTS TRIGGERS DESIGNER

**Filters**

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
sal_donacion	Browse Structure Search Insert Empty Drop 1,000 InnoDB utf8_general_ci 112.0 Kib	1,000	InnoDB	utf8_general_ci	112.0 Kib	-
sal_entidad	Browse Structure Search Insert Empty Drop 10,000 InnoDB utf8_general_ci 1.7 MiB	10,000	InnoDB	utf8_general_ci	1.7 MiB	-
sal_factura_material	Browse Structure Search Insert Empty Drop 10,000 InnoDB utf8_general_ci 1.7 MiB	10,000	InnoDB	utf8_general_ci	1.7 MiB	-
sal_personal	Browse Structure Search Insert Empty Drop 11 InnoDB utf8_general_ci 1.5 MiB	11	InnoDB	utf8_general_ci	1.5 MiB	-
sal_scouter	Browse Structure Search Insert Empty Drop 1 InnoDB utf8_general_ci 16.0 Kib	1	InnoDB	utf8_general_ci	16.0 Kib	-
5 tables	Sum	21,012	InnoDB	utf8mb4_0900_ai_ci	5.0 MiB	0 B

Check all With selected:

**Create table**

Name: Number of columns: 4

Go

The screenshot shows the phpMyAdmin interface connected to the 'SAL\_mysql\_bridge' server, specifically the 'sal\_scouts' database and the 'sal\_factura\_material' table. The table has 10000 rows. The columns are: id\_factura, sal\_entidad\_id\_entidad, fecha\_compra, num\_orden, fecha\_pedido, producto, cod\_producto, num\_lote, cantidad, precio, subtotal, total, and trial378. The data includes various product details like Ziplock, Ice\_chest, Can\_opener, bed, No-stick\_pan, Stir\_spoon, Hammock, Toilet, and Tablecloth, along with their respective quantities, prices, and totals.

Como podemos observar todo funciona correctamente y sin ningún problema.

Cómo último ejemplo vamos a crear una instancia de 3 contenedores (**WordPress**, **PhpMyAdmin** y **MySQL**) bajo una misma red, esta vez no podremos utilizar nuestro volumen ya que debe de estar vacío sino no se podrían crear los usuarios debido a las variables de entorno de **WordPress**.

Vamos a seguir los siguientes pasos:

- Crear una nueva red puente llamada **SAL-wordpress-net**:

```
root@sal-ubuntu:~# docker network remove SAL-wordpress-net; docker network create SAL-wordpress-net; docker network ls
Error: No such network: SAL-wordpress-net
6b31bcbd5198c70bb9885a97cc72f20406cdd0e63ed4b41c089e808151a8a528
NETWORK ID     NAME      DRIVER      SCOPE
6b31bcbd5198   SAL-wordpress-net    bridge      local
b8c6b4210516   SALnet    bridge      local
378094eeb780   bridge    bridge      local
a0bc0704888e   host      host       local
1101d330a414   none     null       local
root@sal-ubuntu:~# |
```

- Crear nuevo volumen llamado ***SAL-wordpress-mysql-data***:

```
root@sal-ubuntu:~# docker volume remove SAL-wordpress-mysql-data; docker volume create SAL-wordpress-mysql-data; docker volume ls
Error: No such volume: SAL-wordpress-mysql-data
SAL-wordpress-mysql-data
DRIVER      VOLUME NAME
local      SAL-wordpress-mysql-data
local      SAL_mysql_data
root@sal-ubuntu:~# |
```

- Crear el contenedor **MySQL** con el nuevo volumen e introducir un usuario y una base de datos para **WordPress** con conexión a la nueva red puente:

```
root@sal-ubuntu:~# docker run -d --rm --name SAL_mysql-wp --network SAL-wordpress-net -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Abcd1234. -e MYSQL_DATABASE=wp_database -e MYSQL_USER=wp_user
-e MYSQL_PASSWORD=wp_password -v SAL-wordpress-mysql-data:/var/lib/mysql mysql
8diec6adae225a9ceeb5341ab55914ed34357b80690805b9b436e099a2a1a8bc
root@sal-ubuntu:~# |
```

- Crear el contenedor **PhpMyAdmin** con las mismas características del anterior ejemplo, pero con la nueva red puente:

```
root@sal-ubuntu:~# docker run -d --rm --network SAL-wordpress-net -e PMA_ARBITRARY=1 -p 8080:80 phpmyadmin/phpmyadmin
9c7051fd1d7769f3040e6f13ed3a61edad413071058e1456c604bb395d8748d6
root@sal-ubuntu:~# |
```

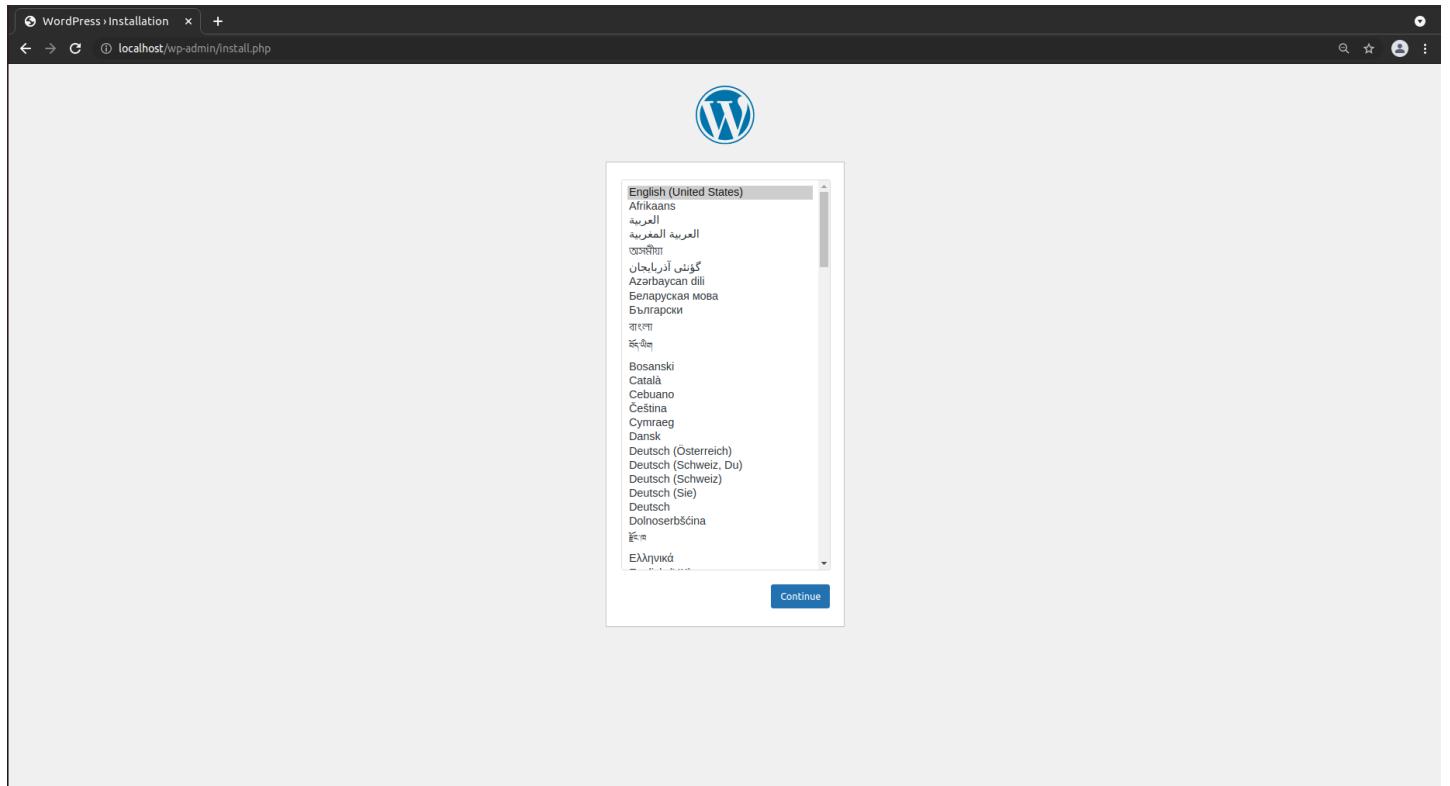
- Crear el contenedor **WordPress** con la nueva conexión puente junto indicando en las variables de entorno los valores del host, base de datos, usuario y contraseña de base de datos. Es necesario crear un volumen específico para este contenedor llamado ***wordpress-data*** e indicarlo en el directorio **/var/www/html**:

```
root@sal-ubuntu:~# docker volume remove wordpress-data; docker volume create wordpress-data; docker volume ls
wordpress-data
wordpress-data
DRIVER      VOLUME NAME
local      SAL-wordpress-mysql-data
local      SAL_mysql_data
local      wordpress-data
root@sal-ubuntu:~# docker run -d --rm --name wordpressc --network SAL-wordpress-net -p 80:80 -e WORDPRESS_DB_HOST=SAL_mysql-wp -e WORDPRESS_DB_NAME=wp_database -e WORDPRESS_DB_USER=wp_user
-e WORDPRESS_DB_PASSWORD=wp_password -v wordpress-data:/var/www/html wordpress
a6819cd431ecf237526b6eef0ec579a65c8a90f17f839a33381ed59d6389e467
root@sal-ubuntu:~# |
```

Creado todo comprobamos los contenedores:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2c8637333280	wordpress	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:80->80/tcp, :::80->80/tcp 0.0.0.0:8080->80/tcp, :::8080->80/tcp 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	wordpressc condescending_nobel SAL_mysql-wp
9c7051fd1d77	phpmyadmin/phpmyadmin	"/docker-entrypoint...."	30 minutes ago	Up 30 minutes		
8diec6adae22	mysql	"docker-entrypoint.s..."	33 minutes ago	Up 33 minutes		

Y solamente queda ir al navegador web y entrar en WordPress con la URL: **localhost**



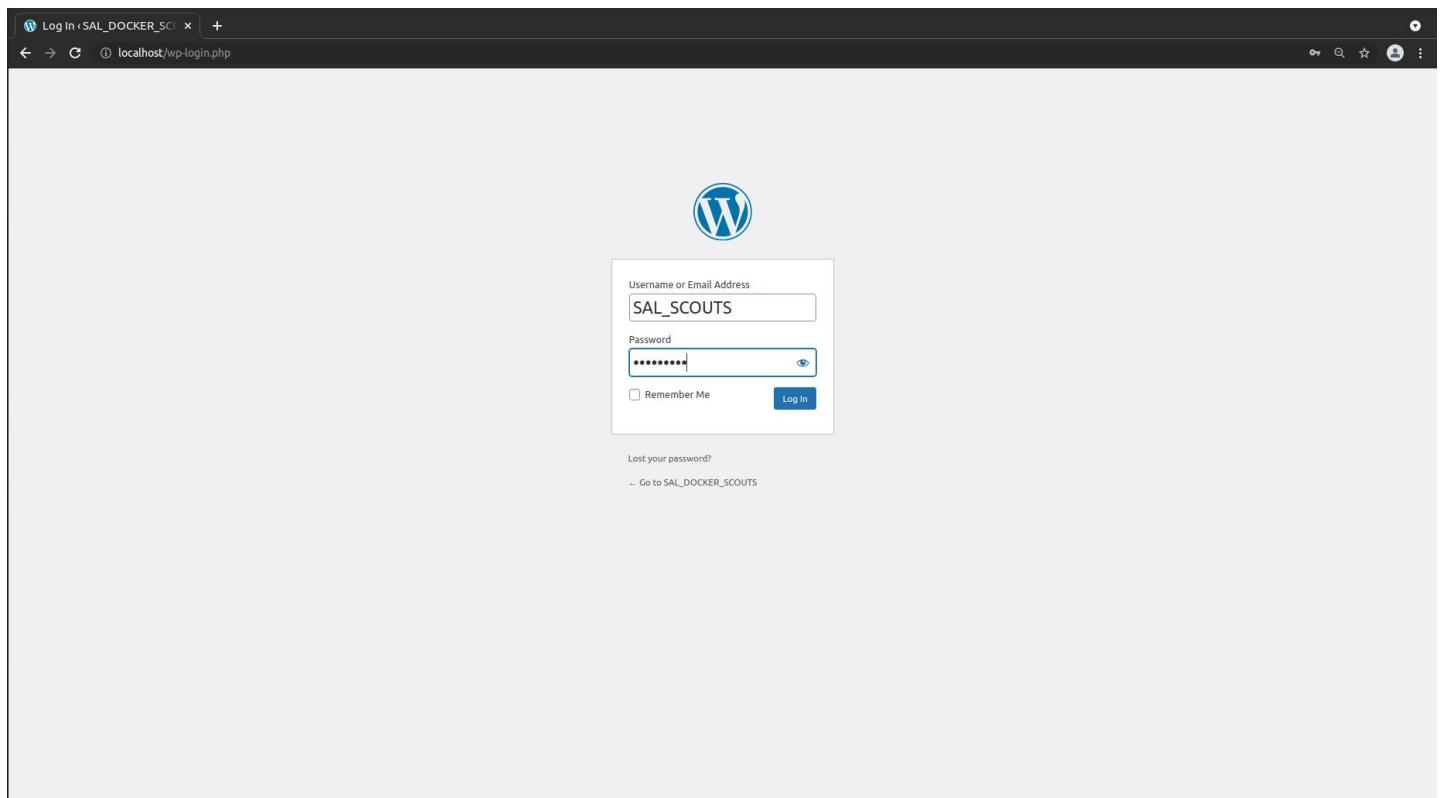
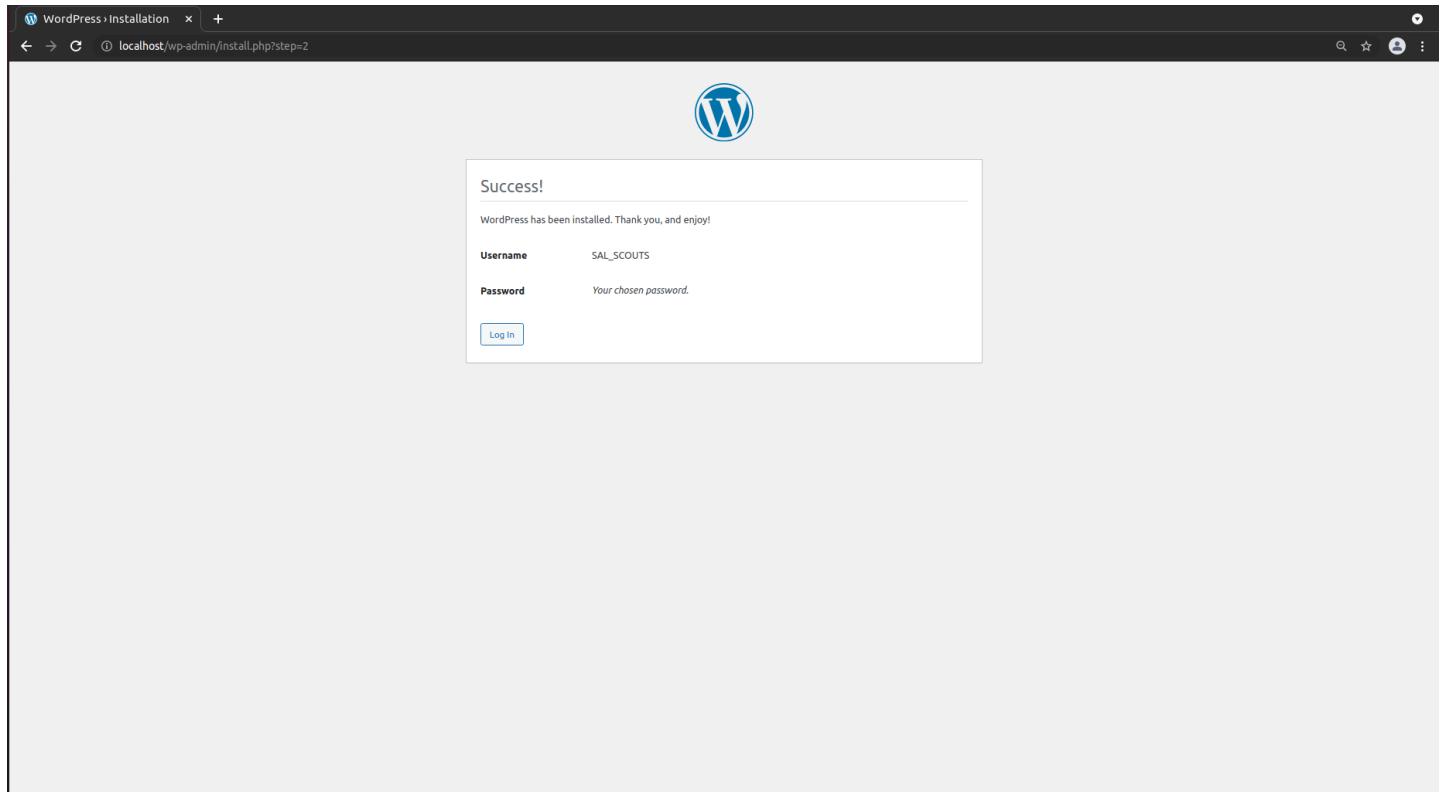
**Welcome**  
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

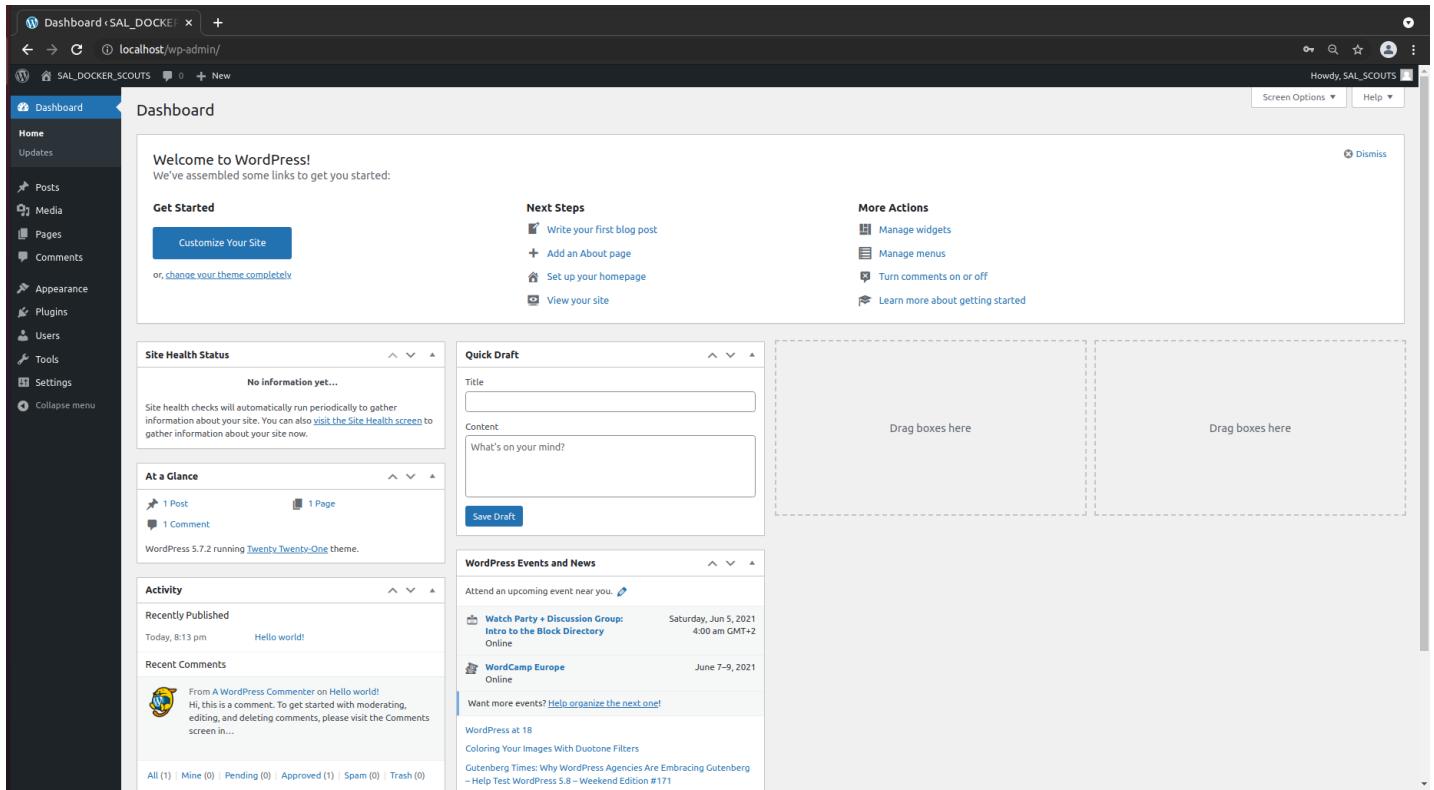
**Information needed**

Please provide the following information. Don't worry, you can always change these settings later.

<b>Site Title</b>	SAL_DOCKER_SCOUTS
<b>Username</b>	SAL_SCOUTS
<b>Password</b>	Abcd1234. Very weak Hide
<b>Confirm Password</b>	<input checked="" type="checkbox"/> Confirm use of weak password
<b>Your Email</b>	sal_scouts@scouts.com Double-check your email address before continuing.
<b>Search engine visibility</b>	<input type="checkbox"/> Discourage search engines from indexing this site It is up to search engines to honor this request.

**Install WordPress**





Y con esto tenemos 3 contenedores de Docker trabajando al mismo tiempo en la misma red.

### ~ PostgreSQL y pgAdmin:

Vamos a crear un contenedor **PostgreSQL 12** con el SGBD **pgAdmin 4**, en otro contenedor diferente. El método que realizaremos será el de crear un archivo de **docker-compose**, para ello tendremos que instalar **docker-compose**, ejecutamos el comando: `apt install docker-compose -y`

```
root@sal-ubuntu:~# apt install docker-compose -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-attr python3-cached-property python3-distutils python3-docker pykerberos python3-docopt python3-importlib-metadata python3-jsonschema
  python3-lib2to3 python3-more-itertools python3-pysistent python3-setuptools python3-texttable python3-websocket python3-zipp
Suggested packages:
  python3-attr-doc python3-jsonschema-doc python3-setuptools-doc
Recommended packages:
  docker.io
The following NEW packages will be installed:
  docker-compose python3-attr python3-cached-property python3-distutils python3-docker pykerberos python3-docopt python3-importlib-metadata
  python3-jsonschema python3-lib2to3 python3-more-itertools python3-pysistent python3-setuptools python3-texttable python3-websocket python3-zipp
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 993 kB of archives.
After this operation, 6,155 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-cached-property all 1.5.1-4 [10,9 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 pykerberos all 0.53.0-2ubuntu1 [32,3 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docker all 4.1.0-1 [83,8 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 pykerberos all 0.4.1-2 [11,1 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-docopt all 0.6.2-2.2ubuntu1 [19,7 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-attr all 19.3.0-2 [33,9 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 libzstd3 all 3.8.10-0ubuntu1-20.04 [76,3 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-distutils all 3.8.10-0ubuntu1-20.04 [141 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-setuptools all 45.2.0-1 [330 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-more-itertools all 4.2.0-1build1 [39,4 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-zipp all 1.0.0-1 [5,312 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu focal/main amd64 pykerberos all 0.53.0-2ubuntu1 [9,992 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-pysistent amd64 0.15.5-1build1 [52,1 kB]
Get:14 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-jsonschema all 3.2.0-0ubuntu2 [43,1 kB]
Get:15 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-texttable all 1.6.2-2 [11,0 kB]
Get:16 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 docker-compose all 1.25.0-1 [92,7 kB]
Fetched 993 kB in 5s (196 kB/s)
Selecting previously unselected package python3-cached-property.
(Reading database ... 165875 files and directories currently installed.)
Preparing to unpack .../00-python3-cached-property_1.5.1-4_all.deb ...
Unpacking python3-cached-property (1.5.1-4) ...
Selecting previously unselected package python3-websocket.
Preparing to unpack .../01-python3-websocket_0.53.0-2ubuntu1_all.deb ...
```

Instalado ya, crearemos una carpeta que contendrá el archivo `docker-compose.yml`:

```
root@sal-ubuntu:~# mkdir PostgreSQL_pgAdmin
root@sal-ubuntu:~# cd PostgreSQL_pgAdmin/
root@sal-ubuntu:~/PostgreSQL_pgAdmin# ls -l
total 0
root@sal-ubuntu:~/PostgreSQL_pgAdmin# |
```

Y creado el directorio vamos a crear el archivo `docker-compose.yml` en **Visual Studio Code** y tendrá la siguiente estructura:

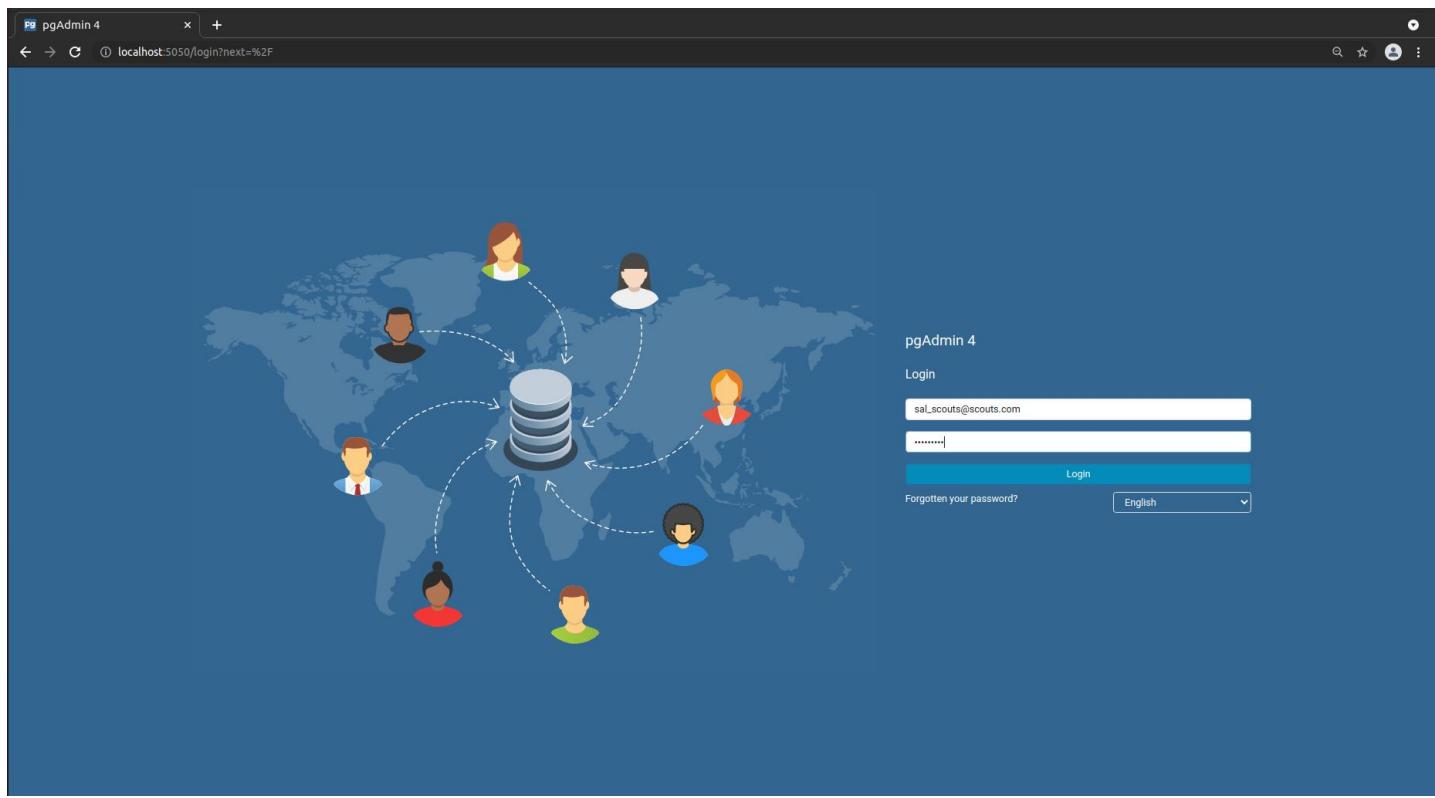
```
version: '3.3'
services:
  db:
    container_name: sal-postgres
    image: postgres
    restart: always
    environment:
      POSTGRES_USER: root
      POSTGRES_PASSWORD: Abcd1234.
      POSTGRES_DB: sal_scouts
    ports:
      - "5432:5432"
  pgadmin:
    container_name: sal-pgadmin4
    image: dpage/pgadmin4
    restart: always
    environment:
      PGADMIN_DEFAULT_EMAIL: sal_scouts@scouts.com
      PGADMIN_DEFAULT_PASSWORD: Abcd1234.
    ports:
      - "5050:80"
```

```
version: '3.3'
services:
  db:
    container_name: sal-postgres
    image: postgres
    restart: always
    environment:
      POSTGRES_USER: root
      POSTGRES_PASSWORD: Abcd1234.
      POSTGRES_DB: sal_scouts
    ports:
      - "5432:5432"
  pgadmin:
    container_name: sal-pgadmin4
    image: dpage/pgadmin4
    restart: always
    environment:
      PGADMIN_DEFAULT_EMAIL: sal_scouts@scouts.com
      PGADMIN_DEFAULT_PASSWORD: Abcd1234.
    ports:
      - "5050:80"
```

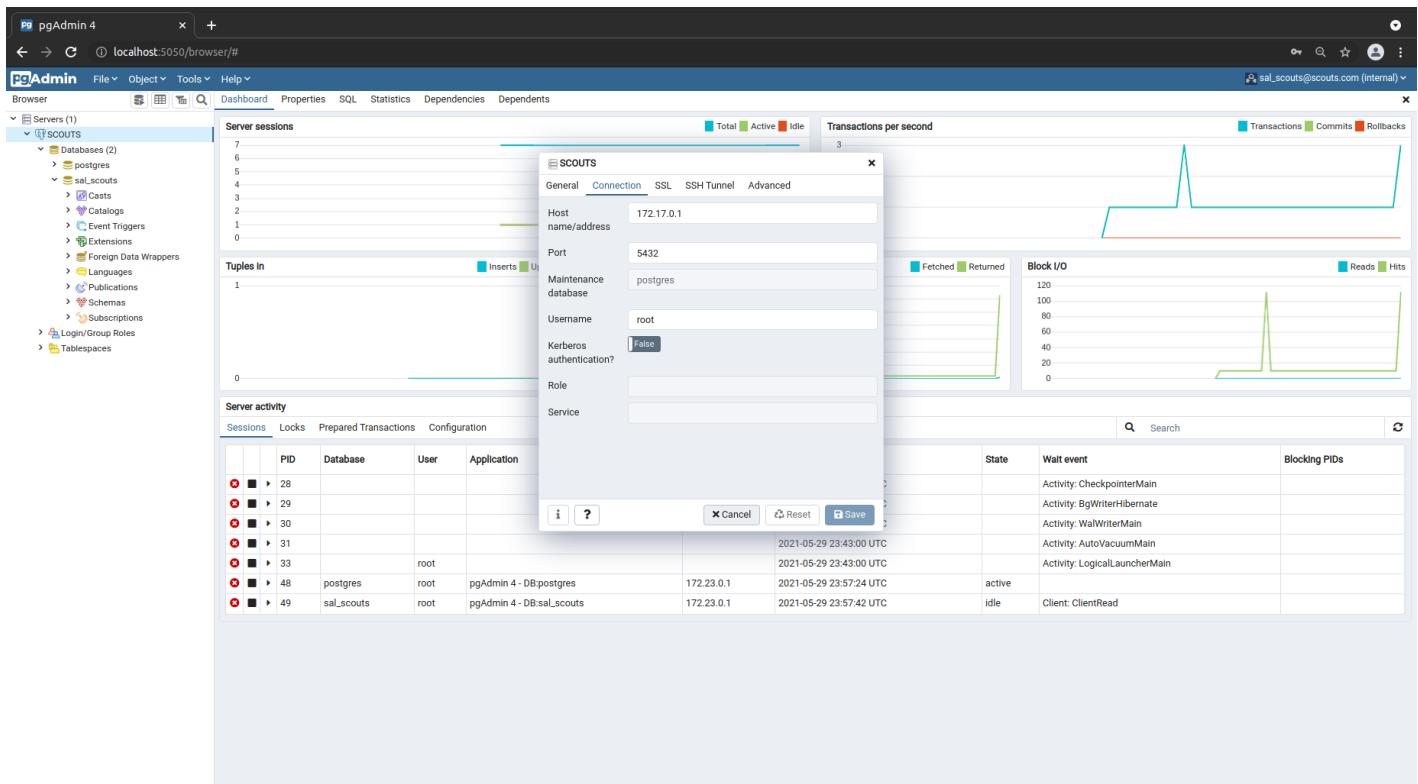
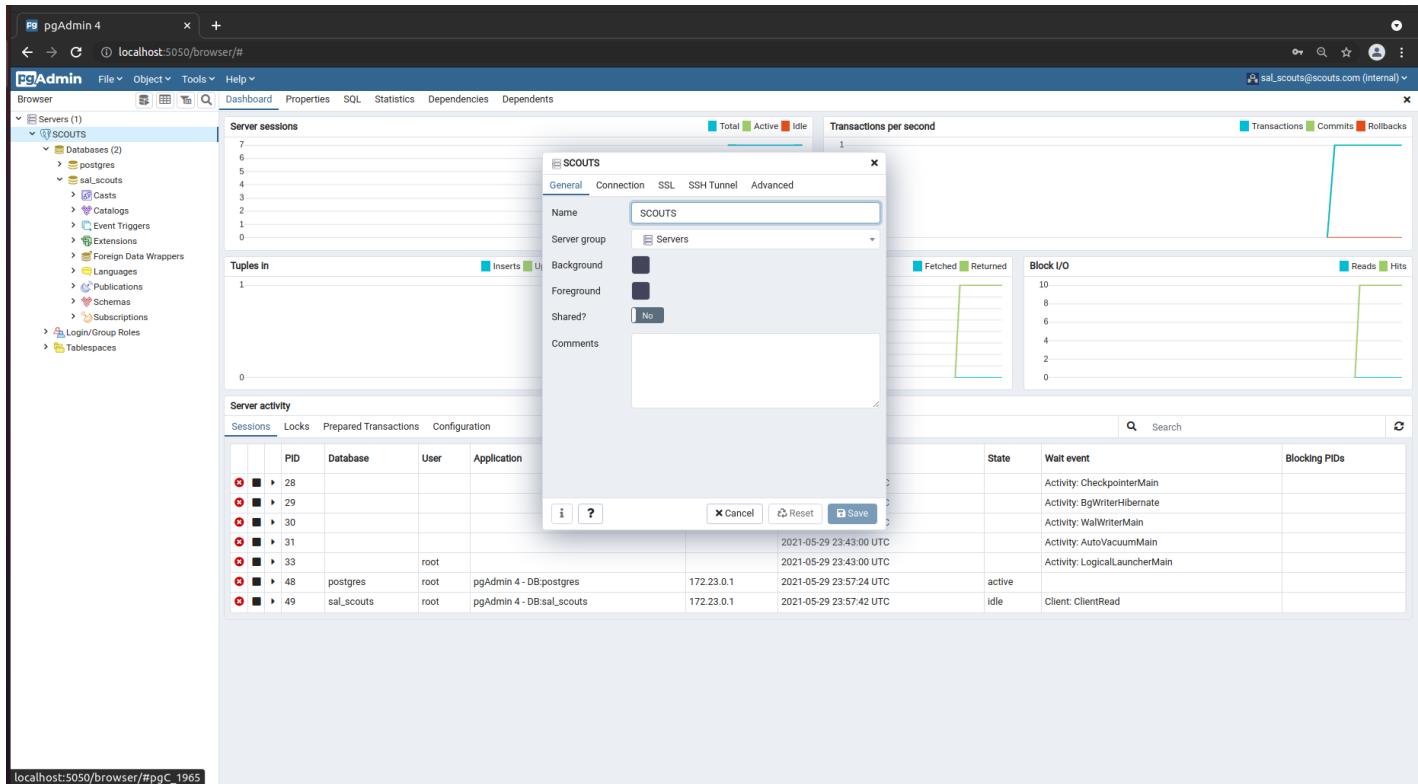
Creado el archivo ejecutamos el comando, estando dentro del directorio: **docker-compose up**

```
root@sal-ubuntu:~/postgreSQL_pgAdmin# docker-compose up
Recreating sal-pgadmin4 ... done
Starting sal-postgres ... done
Attaching to sal-postgres, sal-pgadmin4
sal-postgres |
sal-postgres | PostgreSQL Database directory appears to contain a database; Skipping initialization
sal-postgres |
sal-postgres | 2021-05-29 23:43:00.452 UTC [1] LOG:  starting PostgreSQL 13.3 (Debian 13.3-1.pgdg100+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
sal-postgres | 2021-05-29 23:43:00.456 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
sal-postgres | 2021-05-29 23:43:00.459 UTC [1] LOG:  listening on IPv6 address "::", port 5432
sal-postgres | 2021-05-29 23:43:00.462 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
sal-postgres | 2021-05-29 23:43:00.467 UTC [27] LOG:  database system was shut down at 2021-05-29 23:41:05 UTC
sal-postgres | 2021-05-29 23:43:00.476 UTC [1] LOG:  database system is ready to accept connections
sal-pgadmin4 | NOTE: Configuring authentication for SERVER mode.
sal-pgadmin4 |
sal-pgadmin4 | [2021-05-29 23:43:20 +0000] [1] [INFO] Starting gunicorn 20.1.0
sal-pgadmin4 | [2021-05-29 23:43:20 +0000] [1] [INFO] Listening at: http://[::]:80 (1)
sal-pgadmin4 | [2021-05-29 23:43:20 +0000] [1] [INFO] Using worker: gthread
sal-pgadmin4 | [2021-05-29 23:43:20 +0000] [93] [INFO] Booting worker with pid: 93
```

Y ahora el terminal se ha convertido en el log de ambos contenedores; comprobamos su funcionamiento en el navegador web con la URL: **localhost:5050**



A screenshot of the pgAdmin 4 browser interface. The top navigation bar includes "File", "Object", "Tools", "Help", and a user dropdown. The main content area has a "Welcome" header and a "pgAdmin" logo. It displays a brief introduction about pgAdmin being an open-source PostgreSQL management tool. Below this are sections for "Quick Links" (with "Add New Server" and "Configure pgAdmin" buttons) and "Getting Started" (with links to "PostgreSQL Documentation", "pgAdmin Website", "Planet PostgreSQL", and "Community Support").



Como podemos observar tenemos un par de contenedores ejecutándose con **docker-compose**.

### ~ Dockerfile con MongoDB:

En este apartado vamos a crear un archivo `dockerfile` que lo usaremos para crearnos una imagen. Primero iniciamos sesión en **Docker Hub** mediante CLI con el comando `docker login`:

```
root@sal-ubuntu:~# docker login -u saulaltoubahleon
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@sal-ubuntu:~# |
```

Iniciada la sesión creamos un nuevo directorio para guardar nuestro archivo `dockerfile`:

```
root@sal-ubuntu:~# mkdir mongo_dockerfile
root@sal-ubuntu:~# cd mongo_dockerfile/
root@sal-ubuntu:~/mongo_dockerfile# ls -la
total 8
drwxr-xr-x  2 root root 4096 may 30 02:23 .
drwx----- 11 root root 4096 may 30 02:23 ..
root@sal-ubuntu:~/mongo_dockerfile# |
```

Creamos el archivo `dockerfile` en **Visual Code Studio**, tendrá la siguiente estructura:

```
# Dockerfile for building a MongoDB service
```

```
# Pull base image.
```

```
FROM mongo
```

```
# Define mountable directories.
```

```
VOLUME ["/data/db"]
```

```
# Define working directory.
```

```
WORKDIR /data
```

```
# Define default command.
```

```
CMD ["mongod"]
```

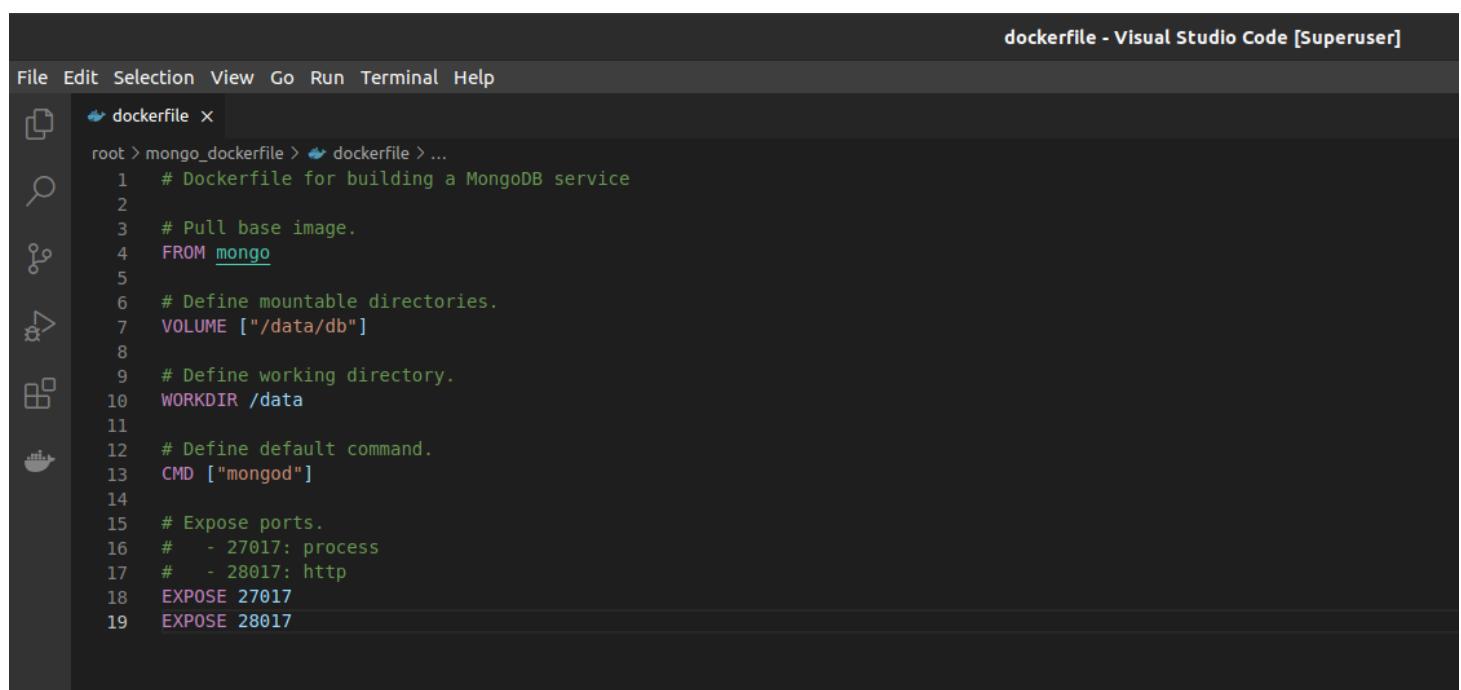
```
# Expose ports.
```

```
# - 27017: process
```

```
# - 28017: http
```

```
EXPOSE 27017
```

```
EXPOSE 28017
```



The screenshot shows a dark-themed instance of Visual Studio Code with a single file open: 'dockerfile'. The file contains a Dockerfile with the following content:

```
root > mongo_dockerfile > dockerfile > ...
1  # Dockerfile for building a MongoDB service
2
3  # Pull base image.
4  FROM mongo
5
6  # Define mountable directories.
7  VOLUME ["/data/db"]
8
9  # Define working directory.
10 WORKDIR /data
11
12 # Define default command.
13 CMD ["mongod"]
14
15 # Expose ports.
16 # - 27017: process
17 # - 28017: http
18 EXPOSE 27017
19 EXPOSE 28017
```

The code editor interface includes a top bar with 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help' menus. On the left, there's a sidebar with icons for file operations like opening, saving, and deleting files. The status bar at the bottom right displays 'dockerfile - Visual Studio Code [Superuser]'. The code itself is color-coded, with syntax highlighting for Docker instructions like 'FROM', 'CMD', and 'EXPOSE'.

Ya creado, construimos nuestra imagen de **Docker**, ejecutamos el comando, estando dentro del directorio: `docker build -t saulaltoubahleon/mongobuild` .

```

root@sal-ubuntu:~/mongo_dockerfile# docker build -t saulaltoubahleon/mongobuild .
Sending build context to Docker daemon 2.048kB
Step 1/6 : FROM mongo
latest: Pulling from library/mongo
01bf7da0a88c: Pull complete
f3b4a5f15c7a: Pull complete
57ffbe87baa1: Pull complete
77d5e5c7eab9: Pull complete
43798cf18b45: Pull complete
67349a81f435: Pull complete
590845b1f17c: Pull complete
1f2ff17242ce: Pull complete
6f11b2ce0594: Pull complete
91532386f4ec: Pull complete
705ef0ab262e: Pull complete
e6238126b609: Pull complete
Digest: sha256:8b35c0a75c2dbf23110ed2485fec a567ec9ab743feee7a0d7a148f806daf5e86
Status: Downloaded newer image for mongo:latest
    --> 07630e791de3
Step 2/6 : VOLUME ["/data/db"]
    --> Running in 4e29420754ce
Removing intermediate container 4e29420754ce
    --> 66a2451b116f
Step 3/6 : WORKDIR /data
    --> Running in ec4c0988c8bf
Removing intermediate container ec4c0988c8bf
    --> 7de970039d69
Step 4/6 : CMD ["mongod"]
    --> Running in e2548f3eb422
Removing intermediate container e2548f3eb422
    --> 1c55f39941df
Step 5/6 : EXPOSE 27017
    --> Running in a761d7d884e3
Removing intermediate container a761d7d884e3
    --> 6f892ef8f7f9
Step 6/6 : EXPOSE 28017
    --> Running in a8e8faab0a7f
Removing intermediate container a8e8faab0a7f
    --> df8b9334c2b6
Successfully built df8b9334c2b6
Successfully tagged saulaltoubahleon/mongobuild:latest
root@sal-ubuntu:~/mongo_dockerfile#

```

Si revisamos nuestras imágenes con `docker images` aparecerá la que hemos creado:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
saulaltoubahleon/mongobuild	latest	df8b9334c2b6	2 minutes ago	449MB
wordpress	latest	0adda6ed742f	5 days ago	551MB
dpage/pgadmin4	latest	df872ce2bc9e	11 days ago	244MB
postgres	latest	293e4ed402ba	2 weeks ago	315MB
mysql	latest	c0cdc95609f1	2 weeks ago	556MB
mongo	latest	07630e791de3	2 weeks ago	449MB
adminer	latest	365268e7ce46	3 weeks ago	89.8MB
phpmyadmin/phpmyadmin	latest	72000eb04892	2 months ago	477MB

Creamos el contenedor de **MongoDB** con nuestra imagen ejecutamos el comando:  
**docker run -p 28017:28017 -name mongoSAL saulaltoubahleon/mongobuild**, y se iniciarán los servicios de **Mongo**:

```
root@sal-ubuntu:~/mongo dockerfile# docker run -p 28017:28017 --name mongoSAL saulaltoubahleon/mongobuild
{"t":{"$date": "2021-05-30T00:46:40.757+00:00"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols none"}
{"t":{"$date": "2021-05-30T00:46:40.760+00:00"}, "s": "W", "c": "ASIO", "id": 22601, "ctx": "main", "msg": "No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date": "2021-05-30T00:46:40.761+00:00"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
{"t":{"$date": "2021-05-30T00:46:40.762+00:00"}, "s": "I", "c": "STORAGE", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 1, "port": 27017, "dbPath": "/data/db", "architecture": "64-bit", "host": "d59ea016a644"}}
{"t":{"$date": "2021-05-30T00:46:40.762+00:00"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "4.4.6", "gitVersion": "72e66213c2c3eab37d9358d5e78ad7f5c1d0dd07", "openSSLVersion": "OpenSSL 1.1.1 11 Sep 2018", "modules": [], "allocator": "tcmalloc", "environment": {"distmod": "ubuntu1804", "distarch": "x86_64", "target_arch": "x86_64"}}, "os": {"name": "Ubuntu", "version": "18.04"}}
{j: "Terminal"
{"t":{"$date": "2021-05-30T00:46:40.762+00:00"}, "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Ubuntu", "version": "18.04"}}, "j": "Terminal"
{"t":{"$date": "2021-05-30T00:46:40.762+00:00"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"bindIp": "*"}}, "j": "Terminal"
{"t":{"$date": "2021-05-30T00:46:40.768+00:00"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startupWarnings"]}
{"t":{"$date": "2021-05-30T00:46:40.768+00:00"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": {"create,cache_size=467M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress]"}}
{"t":{"$date": "2021-05-30T00:46:41.299+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1622335601:299742][1:0x7ff1bbe03a0c], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}, "j": "Terminal"
{"t":{"$date": "2021-05-30T00:46:41.299+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1622335601:299814][1:0x7ff1bbe03a0c], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}, "j": "Terminal"
{"t":{"$date": "2021-05-30T00:46:41.307+00:00"}, "s": "I", "c": "STORAGE", "id": 4795906, "ctx": "initandlisten", "msg": "WiredTiger opened", "attr": {"durationMillis": 539}}
{"t":{"$date": "2021-05-30T00:46:41.307+00:00"}, "s": "I", "c": "RECOVERY", "id": 23987, "ctx": "initandlisten", "msg": "WiredTiger recoveryTimestamp", "attr": {"recoveryTimestamp": {"$timestamp": {"$t": 0, "i": 0}}}
{"t":{"$date": "2021-05-30T00:46:41.322+00:00"}, "s": "I", "c": "STORAGE", "id": 4366408, "ctx": "initandlisten", "msg": "No table logging settings modifications are required for existing WiredTiger tables", "attr": {"loggingEnabled": true}}
{"t":{"$date": "2021-05-30T00:46:41.323+00:00"}, "s": "I", "c": "STORAGE", "id": 22262, "ctx": "initandlisten", "msg": "Timestamp monitor starting"}
{"t":{"$date": "2021-05-30T00:46:41.327+00:00"}, "s": "W", "c": "CONTROL", "id": 22120, "ctx": "initandlisten", "msg": "Access control is not enabled for the database. Read and write access to data and configuration is unrestricted", "tags": ["startupWarnings"]}
{"t":{"$date": "2021-05-30T00:46:41.327+00:00"}, "s": "I", "c": "INDEX", "id": 20320, "ctx": "initandlisten", "msg": "createCollection", "attr": {"namespace": "admin.system.version", "uuidDisposition": "provided", "uuid": {"$uuid": "7ed4ce6d3-8c9d-4646-bd5f-49cd9d50c55e"}}, "options": {"$uuid": "7ed4ce6d3-8c9d-4646-bd5f-49cd9d50c55e"}}
{"t":{"$date": "2021-05-30T00:46:41.340+00:00"}, "s": "I", "c": "INDEX", "id": 20345, "ctx": "initandlisten", "msg": "Index build: done building", "attr": {"buildUUID": null, "namespace": "admin.system.version", "index": {"id": "commitTimestamp", "timestamp": {"$t": 0, "i": 0}}}
{"t":{"$date": "2021-05-30T00:46:41.340+00:00"}, "s": "I", "c": "COMMAND", "id": 20459, "ctx": "initandlisten", "msg": "Setting featureCompatibilityVersion", "attr": {"newVersion": "4.4"}}
{"t":{"$date": "2021-05-30T00:46:41.341+00:00"}, "s": "I", "c": "STORAGE", "id": 20536, "ctx": "initandlisten", "msg": "Flow Control is enabled on this deployment"}
{"t":{"$date": "2021-05-30T00:46:41.342+00:00"}, "s": "I", "c": "STORAGE", "id": 20320, "ctx": "initandlisten", "msg": "createCollection", "attr": {"namespace": "local.startup_log", "uuidDisposition": "generated", "uuid": {"$uuid": "ad4e00d-a684-47aa-b5bf-47abff8bb00"}}, "options": {"$capped": true, "size": 10485760}}
{"t":{"$date": "2021-05-30T00:46:41.352+00:00"}, "s": "I", "c": "INDEX", "id": 20345, "ctx": "initandlisten", "msg": "Index build: done building", "attr": {"buildUUID": null, "namespace": "local.startup_log", "index": {"id": "commitTimestamp", "timestamp": {"$t": 0, "i": 0}}}
{"t":{"$date": "2021-05-30T00:46:41.353+00:00"}, "s": "I", "c": "FDC", "id": 20625, "ctx": "initandlisten", "msg": "Initializing full-time diagnostic data capture", "attr": {"dataDirect": "/data/db/diagnostic.data"}, "j": "Terminal"
{"t":{"$date": "2021-05-30T00:46:41.355+00:00"}, "s": "I", "c": "STORAGE", "id": 20320, "ctx": "LogicalSessionCacheRefresh", "msg": "createCollection", "attr": {"namespace": "config.system.sessions", "uuidDisposition": "generated", "uuid": {"$uuid": "29c8de21-8c4d-4eb8-97b1-c134faf5fceee"}}, "options": []}
{"t":{"$date": "2021-05-30T00:46:41.355+00:00"}, "s": "I", "c": "CONTROL", "id": 20712, "ctx": "LogicalSessionCacheReap", "msg": "Sessions collection is not set up; waiting until next sessions rear interval", "attr": {"error": "NamespaceNotFound: config.system.sessions does not exist"}, "j": "Terminal"
}
```

Para seguir operando abrimos otra terminal, comprobamos el contenedor si está en funcionamiento e iniciamos un terminal como cliente **Mongo**:

```
root@sal-ubuntu:~# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d59ea016a644 saulaltoubahleon/mongobuild "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 27017/tcp, 0.0.0.0:28017->28017/tcp, :::28017->28017/tcp mongoSAL

root@sal-ubuntu:~# docker exec -it mongoSAL bash
root@d59ea016a644:/data# mongo
MongoDB shell version v4.4.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fad38908-ebf9-4035-858f-02496c42a219") }
MongoDB server version: 4.4.6
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
...
The server generated these startup warnings when booting:
2021-05-30T00:46:40.768+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-05-30T00:46:41.327+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
...
...
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-->
```

### ~ **MongoDB & Mongo Express con Docker-Compose:**

Vamos a crear otro contenedor de **MongoDB**, pero lo haremos con **Docker-Compose**. El archivo `.yaml` contendrá dos contenedores, uno para **MongoDB** y otro para la interfaz gráfica de **Mongo Express**. Creamos un directorio nuevo para alojar el archivo; esta es la estructura del archivo:

```
# Use root/example as user/password credentials
version: '3.1'
```

```
services:
```

```
mongo:
```

```
  image: mongo
  restart: always
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: Abcd1234.
```

```
mongo-express:
```

```
  image: mongo-express
  restart: always
  ports:
    - 8081:8081
  environment:
    ME_CONFIG_MONGODB_ADMINUSERNAME: root
    ME_CONFIG_MONGODB_ADMINPASSWORD: Abcd1234.
```

File Edit Selection View Go Run Terminal Help

```

docker-compose.yml x
root > mongo_compose > docker-compose.yml
1  # Use root/example as user/password credentials
2  version: '3.1'
3
4  services:
5
6    mongo:
7      image: mongo
8      restart: always
9      environment:
10        MONGO_INITDB_ROOT_USERNAME: root
11        MONGO_INITDB_ROOT_PASSWORD: Abcd1234.
12
13    mongo-express:
14      image: mongo-express
15      restart: always
16      ports:
17        - 8081:8081
18      environment:
19        ME_CONFIG_MONGODB_ADMINUSERNAME: root
20        ME_CONFIG_MONGODB_ADMINPASSWORD: Abcd1234.

```

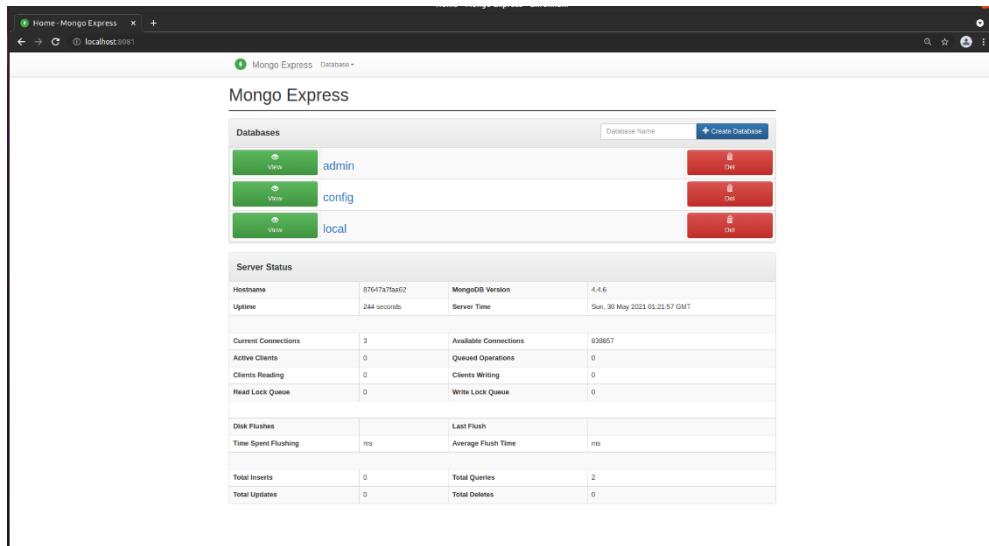
Ya creado, lo ejecutamos y comprobamos el funcionamiento de los contenedores:

```

root@sal-ubuntu:~/mongo_compose# docker-compose up -d
Creating network "mongo_compose_default" with the default driver
Pulling mongo-express (mongo-express:... latest: Pulling from library/mongo-express
ddad3d7c1e96: Pull complete
3a8370f05d5d: Pull complete
71a8563b7fea: Pull complete
119c7e14957d: Pull complete
c06612553eeef: Pull complete
931f05f69fdde: Pull complete
2766ec5ce375: Pull complete
a60269e588ca: Pull complete
Digest: sha256:1df4d44b722aadb31335105972c62e9c971e015f83e68623cec24e6a1a3f0d38
Status: Downloaded newer image for mongo-express:latest
Creating mongo_compose_mongo_1 ... done
Creating mongo_compose_mongo-express_1 ... done
root@sal-ubuntu:~/mongo_compose# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
87647a7faa62 mongo "docker-entrypoint.s..." About a minute ago Up About a minute 27017/tcp mongo_compose_mongo_1
eb7e7ecc2b10 mongo-express "tini -- /docke... About a minute ago Up About a minute 0.0.0.0:8081->8081/tcp, :::8081->8081/tcp mongo_compose_mongo-express_1
root@sal-ubuntu:~/mongo_compose#

```

Y ya solo nos queda conectarnos al navegador web:



## ~ **Creación de Docker propias y subiéndolas a Docker Hub:**

En este último apartado vamos a crear nuestras propias imágenes y las subiremos a nuestro repositorio de **Docker Hub**. Vamos a crear un contenedor con **SQL Server** que contenga nuestra base de datos.

Iniciamos sesión en **Docker Hub** a través de CLI, revisamos que imágenes tenemos descargadas y qué contenedores disponemos. Si no tenemos nada, creamos un contenedor con la imagen de Ubuntu y abrimos un terminal dentro de él:

```
root@sal-ubuntu-docker:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: saulaltoubahleon
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@sal-ubuntu-docker:~# docker images; docker ps -a
REPOSITORY TAG IMAGE ID CREATED SIZE
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@sal-ubuntu-docker:~#
```

Revisado todo, creamos el contenedor, ejecutamos el comando: `docker run -e ACCEPT_EULA=Y -e SA_PASSWORD=Abcd1234. -p 1433:1433 --name sql_server -h sql_server -d mcr.microsoft.com/mssql/server:2019-latest`

```
root@sal-ubuntu-docker:~# docker run -e ACCEPT_EULA=Y -e SA_PASSWORD=Abcd1234. -p 1433:1433 --name sql_server -h sql_server -d mcr.microsoft.com/mssql/server:2019-latest
Unable to find image 'mcr.microsoft.com/mssql/server:2019-latest' locally
2019-latest: Pulling from mssql/server
04a5f4cda3ee: Pull complete
ff496a88c8ed: Pull complete
0ce83f459fe7: Pull complete
18147e02582b: Pull complete
32f0c0acc4b8: Pull complete
e036a29020fa: Pull complete
4ee3d995ce58: Pull complete
Digest: sha256:dde9e587abb7ca1a09e89b4ec4a48b36ab299bfbd1460c64e44ea9bdcb003ac
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2019-latest
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2019-latest
08b23ba91c0f6280201809c06ce95fd738367430bd17b8cb12b3b61f7b0a00b4
root@sal-ubuntu-docker:~#
```

```
root@sal-ubuntu-docker:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
08b23ba91c0f mcr.microsoft.com/mssql/server:2019-latest "/opt/mssql/bin/perm..." 6 minutes ago Up 6 minutes 0.0.0.0:1433->1433/tcp, :::1433->1433/tcp sql_server
root@sal-ubuntu-docker:~#
```

Creado y comprobado el contenedor de **SQL Server**, abrimos un terminal para revisar, ejecutamos el comando: `docker exec -it sql_server /bin/bash`

Revisado todo, copiamos una carpeta llamada *backup* que tenemos en nuestro equipo local al interior del contenedor para poder restaurar nuestra base de datos en la carpeta */home*, ejecutamos el comando *docker cp <carpeta\_local> sql\_server:/home*

```
root@sal-ubuntu-docker:~# docker exec -it sql_server /bin/bash
mssql@sql_server:$ ls -l
total 52
lrwxrwxrwx 1 root root 7 Mar 25 16:58 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Apr 15 2020 boot
drwxr-xr-x 5 root root 340 May 30 15:51 dev
drwxr-xr-x 1 root root 4096 May 30 15:51 etc
drwxr-xr-x 2 root root 4096 Apr 15 2020 home
lrwxrwxrwx 1 root root 7 Mar 25 16:58 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Mar 25 16:58 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Mar 25 16:58 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Mar 25 16:58 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4096 Mar 25 16:58 media
drwxr-xr-x 2 root root 4096 Mar 25 16:58 mnt
drwxrwxr-x 1 root root 4096 Apr 3 03:26 opt
dr-xr-xr-x 376 root root 0 May 30 15:51 proc
drwx----- 2 root root 4096 Mar 25 17:08 root
drwxr-xr-x 1 root root 4096 Mar 28 00:23 run
lrwxrwxrwx 1 root root 8 Mar 25 16:58 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Mar 25 16:58 srv
dr-xr-xr-x 13 root root 0 May 30 15:51 sys
drwxrwxrwt 1 root root 4096 Apr 3 03:26 tmp
drwxrwxr-x 1 root root 4096 Apr 3 03:23 usr
drwxr-xr-x 1 root root 4096 Mar 25 17:08 var
mssql@sql_server:$ exit
exit
root@sal-ubuntu-docker:~# docker cp /home/sal/Desktop/backup sql_server:/home
root@sal-ubuntu-docker:~#
```

Entramos de nuevo en la base de datos y movemos la carpeta */home/backup* a la carpeta de *sqlcmd*:

```
mssql@sql_server:/var/opt/mssql/data/backup$ ls -l
total 25848
-rw----
```

-	rwxr--r--	1	mssql	root	25291776	May 30 16:20	SAL_SCOUTS_Full_TDE.bak
-	rwxr--r--	1	mssql	root	1162752	May 30 16:20	SAL_SCOUTS_log_TDE.bak
-	rwxr--r--	1	mssql	root	700	May 30 16:20	SAL_TDEScout.cer
-	rwxr--r--	1	mssql	root	1212	May 30 16:20	SAL_TDEkey.pvk
-	rwxr--r--	1	mssql	root	208	May 30 16:20	SAL_dmk.key

```
mssql@sql_server:/var/opt/mssql/data/backup$
```

Iniciamos sesión con `SA` en `sqlcmd`, ejecutamos el comando: `/opt/mssql-tools/bin/sqlcmd -S localhost -U SA`

```
mssql@sql_server:/ $ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA  
Password:  
1> [REDACTED]
```

Antes de restaurar la base de datos, debemos restaurar la `master key` y después el certificado:

```
mssql@sql_server:/ $ ls -l /var/opt/mssql/data/backup/  
total 25848  
-rwxr--r-- 1 mssql root 25291776 May 30 16:20 SAL_SCOUTS_Full_TDE.bak  
-rwxr--r-- 1 mssql root 1162752 May 30 16:20 SAL_SCOUTS_log_TDE.bak  
-rwxr--r-- 1 mssql root 700 May 30 16:20 SAL_TDEScout.cer  
-rwxr--r-- 1 mssql root 1212 May 30 16:20 SAL_TDEkey.pvk  
-rwxr--r-- 1 mssql root 208 May 30 16:20 SAL_dmk.key  
mssql@sql_server:/ $ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA  
Password:  
1> DROP MASTER KEY;  
2> GO  
Msg 15151, Level 16, State 1, Server sql_server, Line 1  
Cannot find the symmetric key 'master key', because it does not exist or you do not have permission.  
1> RESTORE MASTER KEY FROM FILE = '/var/opt/mssql/data/backup/sal_dmk.key' DECRYPTION BY PASSWORD = 'Abcd1234.' ENCRYPTION BY PASSWORD = 'Abcd1234.';  
2> GO  
1> CREATE CERTIFICATE [SAL_TDEScout]  
2> FROM FILE = '/var/opt/mssql/data/backup/SAL_TDEScout.cer'  
3> WITH PRIVATE KEY (DECRYPTION BY PASSWORD='Abcd1234.'  
4> , FILE = '/var/opt/mssql/data/backup/SAL_TDEKey.pvk');  
5> GO  
Msg 15581, Level 16, State 7, Server sql_server, Line 1  
Please create a master key in the database or open the master key in the session before performing this operation.  
1> OPEN MASTER KEY DECRYPTION BY PASSWORD = 'Abcd1234.';  
2> GO  
1> CREATE CERTIFICATE [SAL_TDEScout]  
2> FROM FILE = '/var/opt/mssql/data/backup/SAL_TDEScout.cer'  
3> WITH PRIVATE KEY (DECRYPTION BY PASSWORD='Abcd1234.'  
4> , FILE = '/var/opt/mssql/data/backup/SAL_TDEKey.pvk');  
5> GO
```

```
mssql@sql_server:/var/opt/mssql/data$ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA  
Password:  
1> use master;  
2> go  
Changed database context to 'master'.  
1> RESTORE DATABASE [SAL_SCOUTS]  
2> FROM DISK = N'/var/opt/mssql/data/SAL_SCOUTS_Full_TDE.bak' WITH FILE = 1  
3> , MOVE N'SAL_SCOUTS_main' TO N'/var/opt/mssql/data/SAL_SCOUTS_main.mdf'  
4> , MOVE N'SAL_SCOUTS_FG01' TO N'/var/opt/mssql/data/SAL_SCOUTS_FG01.ndf'  
5> , MOVE N'SAL_SCOUTS_FG02' TO N'/var/opt/mssql/data/SAL_SCOUTS_FG02.ndf'  
6> , MOVE N'SAL_SCOUTS_log' TO N'/var/opt/mssql/data/SAL_SCOUTS_log.ldf'  
7> , MOVE N'SAL_SCOUTS_log2' TO N'/var/opt/mssql/data/SAL_SCOUTS_log2.ldf'  
8> , MOVE N'SAL_SCOUTS_FILESTREAM_Main' TO N'/var/opt/mssql/data/filestream'  
9> , NOUNLOAD, STATS = 5;  
10> GO  
Msg 5135, Level 16, State 2, Server sql_server, Line 1  
The path '/var/opt/mssql/data/filestream' cannot be used for FILESTREAM files. For information about supported paths, see SQL Server Books Online.  
Msg 3156, Level 16, State 3, Server sql_server, Line 1  
File 'SAL_SCOUTS_FILESTREAM_Main' cannot be restored to '/var/opt/mssql/data/filestream'. Use WITH MOVE to identify a valid location for the file.  
Msg 3119, Level 16, State 1, Server sql_server, Line 1  
Problems were identified while planning for the RESTORE statement. Previous messages provide details.  
Msg 3013, Level 16, State 1, Server sql_server, Line 1  
RESTORE DATABASE is terminating abnormally.  
The path '/var/opt/mssql/data/filestream' cannot be used for FILESTREAM files. For information about supported paths, see SQL Server Books Online.  
1> [REDACTED]
```

La inconveniencia que ha surgido ahora es que **SQL Server 2019 en Linux no soporta Filestream**, entonces no podemos restaurar nuestra base de datos.

Supongamos que hemos restaurado nuestra base de datos correctamente, salimos del terminal del contenedor y aplicamos los cambios de la imagen ejecutando el comando:

```
docker commit -m "SQL SERVER 2019 SAL_SCOUTS Docker" -a "SAL" 67ef83f44c6a saulaltoubahleon/sal_scouts:v0.1
```

```
root@sal-ubuntu-docker:~# docker commit -m "SQL SERVER 2019 SAL_SCOUTS Docker" -a "SAL" 67ef83f44c6a saulaltoubahleon/sal_scouts:v0.1
sha256:05675a8c24fc373a23da19c110c6e326f625f3356961c6547359538cf0e3205a
root@sal-ubuntu-docker:~# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
saulaltoubahleon/sal_scouts    v0.1      05675a8c24fc  9 seconds ago  1.65GB
mcr.microsoft.com/mssql/server 2019-latest 62c72d863950  8 weeks ago   1.49GB
root@sal-ubuntu-docker:~#
```

Y ahora lo que vamos hacer es subirlo a nuestro perfil de **Docker Hub** creándose así un nuevo repositorio. Ejecutamos el comando: `docker push saulaltoubahleon/sal_scouts:v0.1`

```
root@sal-ubuntu-docker:~# docker push saulaltoubahleon/sal_scouts:v0.1
The push refers to repository [docker.io/saulaltoubahleon/sal_scouts]
06fe6dd36847: Pushed
4d84977d47ec: Pushed
f0331eeb6d21: Pushed
850bffe5ed57: Pushed
208ca9993fcf: Pushed
d4dfaa212623: Pushed
cba97cc5811c: Pushed
0c78fac124da: Pushed
v0.1: digest: sha256:8656e50bfc665f0cb252d8634eec0a88666dcdf9b83eb5c699f01ea80840bac1 size: 1999
root@sal-ubuntu-docker:~#
```

The screenshot shows the Docker Hub interface for the repository `saulaltoubahleon/sal_scouts`. The repository has one tag, `v0.1`, which was pushed a minute ago. There is no description or README provided. The repository is set up for vulnerability scanning, which is currently disabled.