PLAN DE MIGRACIÓN DE BASES DE DATOS

PROYECTO: SOS DOCUMENTAL

INTEGRANTES:

Einer Obeimar Mendez

Juan Camilo Rairan Rodriguez

Diana Marcela Diaz Diaz

Angie BIbiana López Rivera

Cristhian Andres Rico Daza

BOGOTÁ 05 DE ABRIL DE 2019

SERVICIO NACIONAL DE APRENDIZAJE – SENA

CENTRO ELECTRICIDAD, ELECTRÓNICA Y TELECOMUNICACIONES – CEET

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN - ADSI

TRIMESTRE VI

**Plan de Migración de Bases de Datos**

**INTRODUCCIÓN**

Antes de empezar a explicar los pasos que se deben tener en cuenta para realizar una migración, es importante revisar qué es migrar.

La migración de BD es un proceso que se lleva a cabo para trasladar datos almacenados

previamente de un gestor de base de datos a otro o desde el mismo gestor pero a versiones

diferentes. Este es un proceso que nos sirve mucho como apoyo al momento de querer cambiar el

gestor de bases de datos utilizado en nuestro software.

En este documento se llevará a cabo el registro y paso a paso del proceso de migración de una base de datos creada en MySQL y su respectivo traspaso a SQL Server. Para el desarrollo de este proceso vamos a necesitar de una herramienta de software llamada Full Convert Enterprise.

Primero se va a evidenciar los registros que están contenidos en la base de datos y la estructura de la misma, para al final dar un veredicto del resultado de la ejecución de la migración.

También se va a realizar un estudio de factibilidad para lleva a cabo este proceso, de esta forma

veremos qué tan conveniente es hacer la migración y si el resultado que nos dará será el esperado.

JUSTIFICACIÓN

Es necesario realizar la migración en ocasiones ya que muchas veces se requiere combinar los sistemas para mantener la integridad de los datos y de dichos sistemas, mejorar la calidad de los datos y metadatos, es práctico a la hora en que necesitemos manejar nuestra base de datos en otro sistema operativo o lenguaje de programación. Además de todo esto, la migración nos convierte las definiciones de las columnas, es decir tipos de datos, restricciones y valores por defecto.

Convierte las restricciones de seguridad e integridad como las llaves primarias y foráneas, por otro lado nos convierte las palabras reservadas de dicho gestor de base de datos y sobretodo y lo

más importante nos transporta los datos almacenados

Antes de realizar la migración, se debe realizar un análisis previo del sistema gestor de base de

datos de MySQL que es el utilizado actualmente, y del sistema gestor de base de datos final el cual será SQL Server. Esto se realiza para tener una visión general de las ventajas y/o desventajas de

realizar dicha migración, además se estima el nivel de complejidad de llevar a cabo dicho

proceso. Estos elementos son:

Número de Registros filas:

La cantidad de registros en una base de datos es importante en este análisis previo, puesto que no

es lo mismo realizar una migración de cien registros a más de mil millones de registros. Tener en

cuenta que el número de registro de filas afecta tiempos de actualización, Back up, disparadores,

etc.

Número de Atributos (NA)

Otro de los indicadores de complejidad se calcula en base a la cantidad de campos de las tablas.

Se recomienda ejecutar consultas que permitan ver el promedio en general de las columnas, la cantidad máxima y mínima de columnas por tabla. No es lo mismo tener una tabla con 200 campos a una tabla con 2 campos, el manejo en cuanto a índices, restricciones, es diferente.

Número de claves foráneas. Referential Degree (RD)

Es importante conocer la cantidad de llaves foráneas, primarias, índices, constraints, existes en la

base de datos. Tener en cuenta que el manejo de las claves anteriores se relaciona con las tablas y la funcionalidad que estas desempeñan. Pueda que haya llaves innecesarias o por lo contrario que hagan falta índices o claves para mejor tiempos de consulta y actualización.

Número de Objetos de conversión no trivial.

En la base de datos MySQL hay objetos que no tienen correspondencia directa con respecto a la

base de datos Oracle por tanto dicho proceso se vuelve un poco complejo. Es por eso que están

importante conocer qué tan complejo es dicha conversión y así tomar medidas para prever el

impacto de la migración.

Los objetos de la base de datos cuya conversión a lenguaje Oracle, de nivel de complejidad bajo

son: tablas, índices, secuencias, vistas, sinónimos.

Otros aspectos a evaluar

Hay una serie de factores comunes a cualquier base de datos que influyen en la complejidad de la

migración y que hay que ponderar:

Cantidad y tipo de SQL propietario que se use.

Calidad de datos.

Existencia de documentación del sistema.

Requisitos de diseño tales como la alta disponibilidad y replicación.

Software de terceros dependencias.

Cambio en el sistema operativo y/o cambio de la plataforma de hardware como

consecuencia de la migración.

Cualificación y experiencia del personal involucrado en la migración.

La disponibilidad de un equipo dedicado para el desarrollo de la migración.

Imposibilidad de detener los cambios y los nuevos desarrollos sobre el código a migrar.

Tiempo máximo que se permite para la realización de la migración (ventana de corte).

# ANÁLISIS DE MIGRACIÓN

En este punto deben identificarse los cambios específicos que hay que realizar para transformar

cada elemento de la base de datos MySQL en un elemento de SQL Server que funcione de la misma manera. También hay que identificar los cambios a realizar en el código de la propia

aplicación para que funcione sobre una base de datos SQL Server.

Por otro lado, se va a mencionar algunos tipos de datos manejados en MySQL y su

equivalente en SQL Server:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de campo** | **MySQL** | **Especificaciones** | **SQL Server** | **Especificaciones** |
| **Numérico** | ***FLOAT*** | Almacena números de coma flotante, donde ‘m’ es el número de dígitos de la parte entera y ‘d’ el número de decimales. | ***FLOAT*** | Almacena números de coma flotante, donde ‘m’ es el número de dígitos de la parte entera y ‘d’ el número de decimales. |
|  | ***DOUBLE*** | Almacena número de coma flotante con precisión doble. Igual que FLOAT, la diferencia es el rango de valores posibles. | ***BIGINT*** | Ocupación de 8 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215. |
|  | ***BIGINT****:* | Ocupación de 8 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215. |  |  |
| **texto** | ***VARCHAR*** | Ocupación variable cuya longitud comprende de 1 a 255 caracteres. | ***VARCHAR*** | Caracteres no Unicode de longitud variable. n indica que el tamaño de almacenamiento máximo es de 231 - 1 bytes |
|  | ***TEXT*** | Una longitud máxima de 65.535 caracteres. Sirve para almecenar texto plano sin formato. Distingue entre minúculas y mayúsculas. | ***TEXT*** | En desuso, sustituido por varchar.    Datos no Unicode de longitud variable con una longitud máxima de 231 - 1 (2.147.483.647) caracteres |
|  | ***CHAR*** | almacena datos de carácter en un campo de longitud fija como una serie de letras, números y otros caracteres de un solo byte o de varios bytes que están soportados por el juego de códigos del entorno local de la base de datos | ***NCHAR*** | Datos de carácter Unicode de longitud fija, con *n*caracteres. *n* debe estar comprendido entre 1 y 4.000 |
| **tiempo** | ***DATE*** | Válido para almacenar una fecha con año, mes y día, su rango oscila entre ‘1000-01-01′ y ‘9999-12-31′. | ***DATE*** | El tipo de dato DATE sirve para almacenar una fecha con el formato YYYY-MM-DD (4 dígitos para el año, guion, 2 dígitos para el mes, guion, 2 dígitos para el día). |
|  | ***DATETIME****:* | Almacena una fecha (año-mes-día) y una hora (horas-minutos-segundos), su rango oscila entre ‘1000-01-01 00:00:00′ y ‘9999-12-31 23:59:59′. | ***TIMESTAMP*** | data type is used for values that contain both date and time parts.TIMESTAMP has a range of '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' |
|  | ***TIME*** | Válido para almacenar una hora (horas-minutos-segundos). Su rango de horas oscila entre -838-59-59 y 838-59-59. El formato almacenado es ‘HH:MM:SS’. |  |  |
| **imágenes** | ***LONGBLOB****:* | Una longitud máxima de 4.294.967.298 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas. | ***IMAGE*** | en desuso sustituido por varbinary |
|  | ***MEDIUMBLOB****:* | Una longitud máxima de 16.777.215 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas. |  |  |
|  | ***BLOB*** | Una longitud máxima de 65.535 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas. |  |  |

# **ESTRATEGIA DE MIGRACIÓN(PLAN DE MIGRACIÓN) ELABORAR PLAN DE EJECUCIÓN**

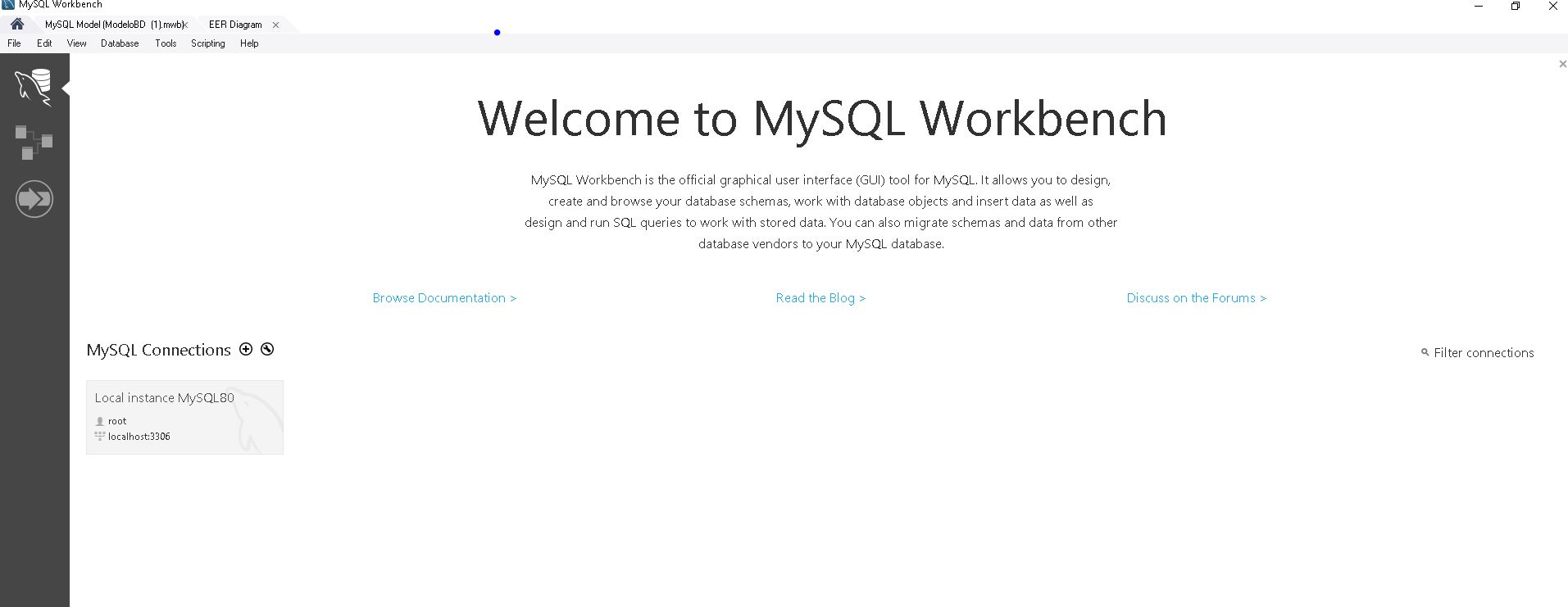
Este proceso se tiene pensado realizar iniciando la descarga de los gestores de base de datos, la migración de un gestor a otro se hará de manera completa es decir que se comenzará digitando código y por último exportando las tablas del gestor, esto lo veremos más adelante

# **PLAN DE MARCHA ATRÁS**

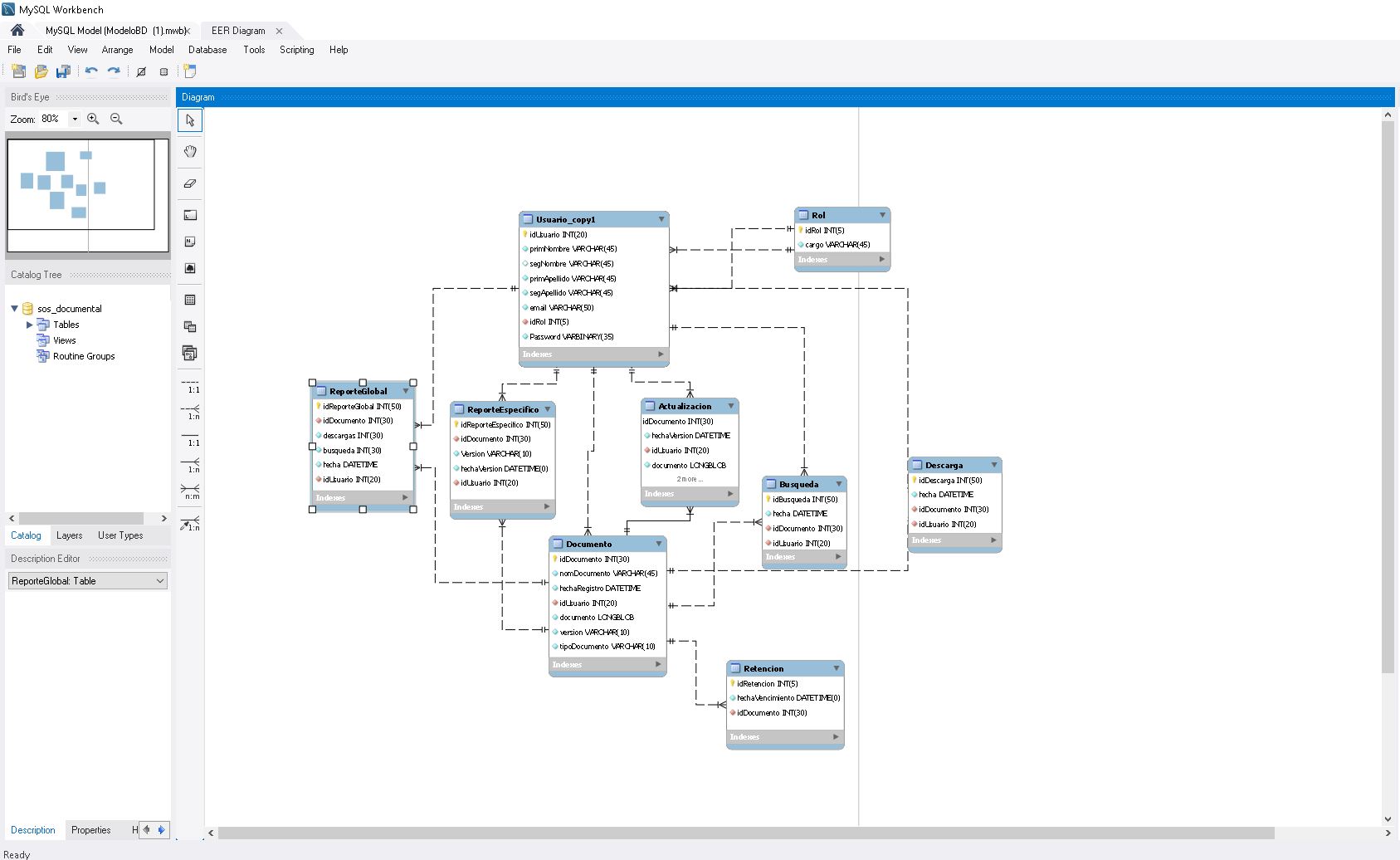
Este será un plan de respaldo, dado caso que en algún momento el plan de migración llegue a tener un fallo se tendrá un backup de la base de datos primaria, es decir que se tendrá una copia de la base de datos en este caso con el gestor de base de datos Mysql Workbench.

# **Copia de seguridad en MysqlWorkbench**

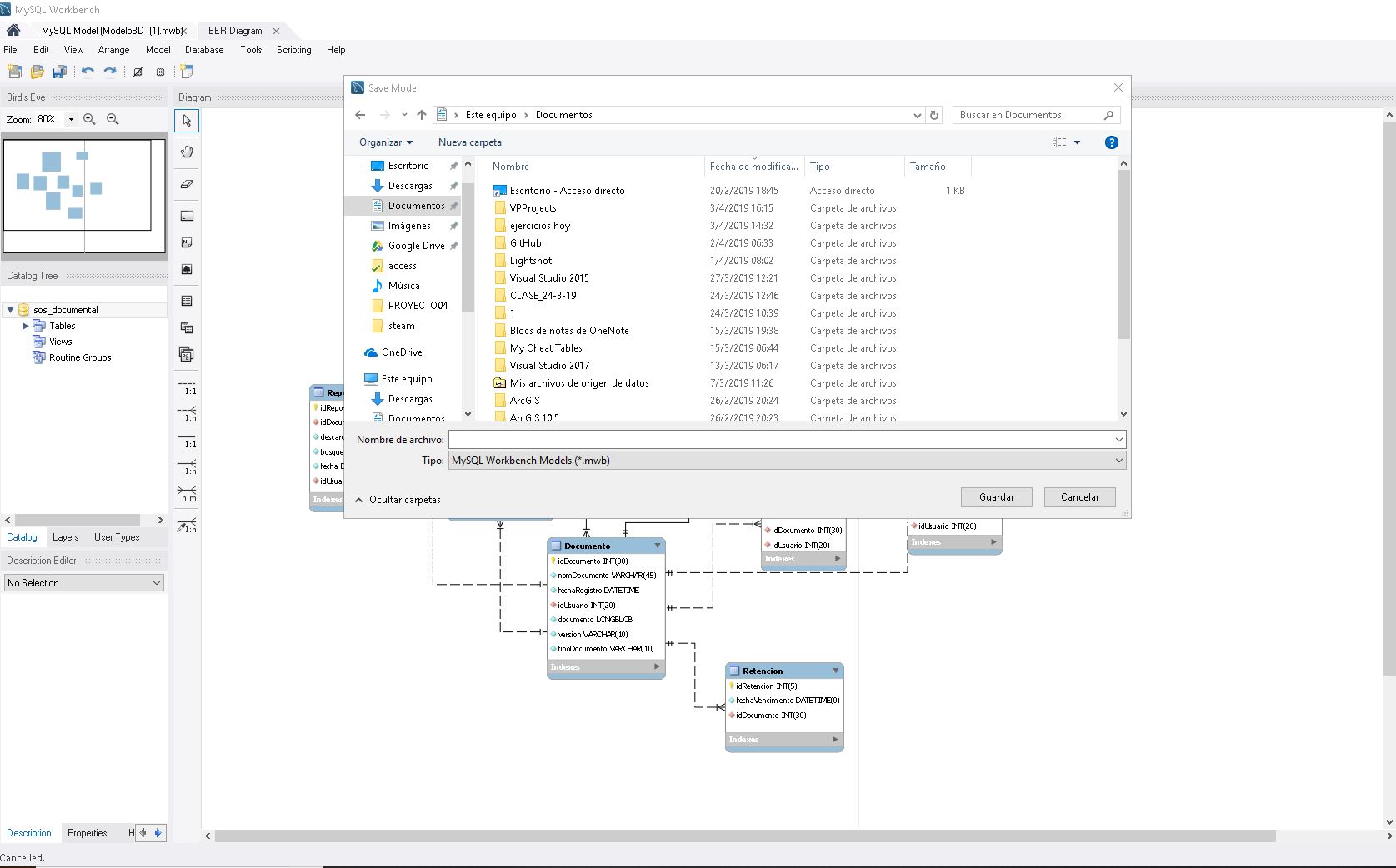
1. tendremos que iniciar abriendo MysqlWorkbench



2. crearemos nuestra base de datos



#### **3.** seleccionaremos en el apartado que dice **File** y le damos en **Save Model As…**



**4.**y quedará una copia de nuestra base de datos guardada, nos quedará un archivo como este

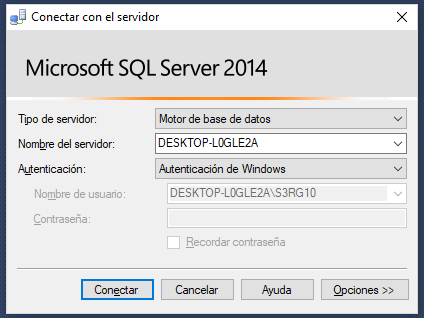


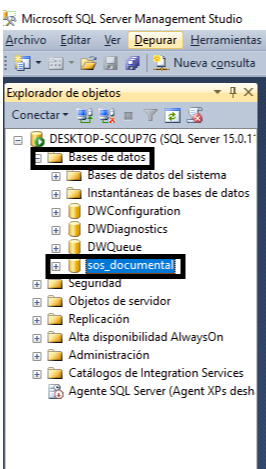
# **Copia de seguridad en SQL server**

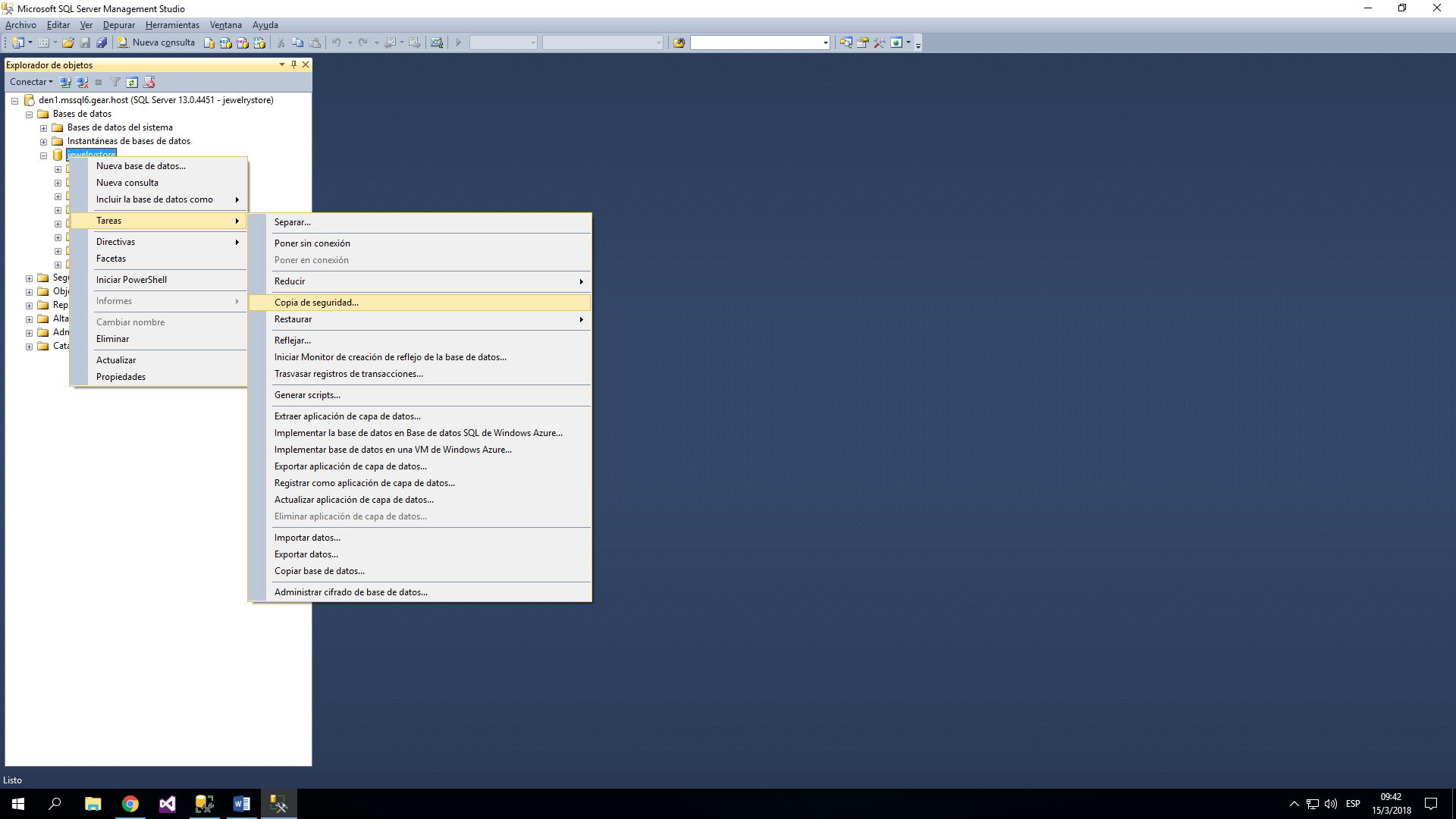
1. Se abre el Management Studio de SQL Server

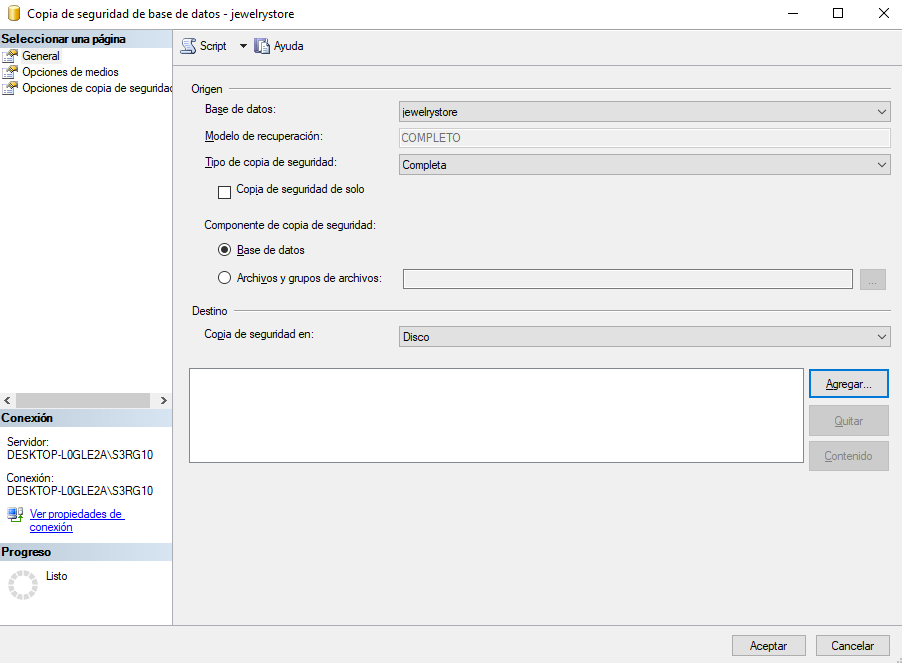


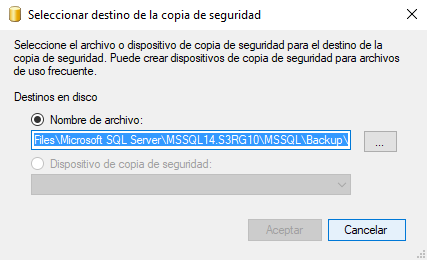
1. Se realiza la respectiva conexión con el servidor local donde está la base de datos:



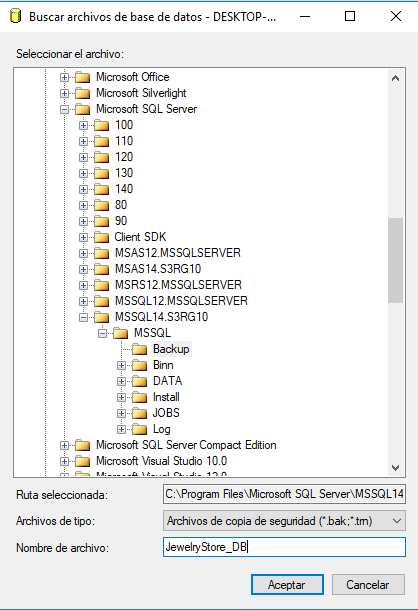
1. Buscar la base de datos a la cual se va a realizar la copia de seguridad
2. Clic derecho sobre la base de datos, buscar la opción ‘Tareas’ y luego seleccionar la opción ‘Copia de Seguridad’:



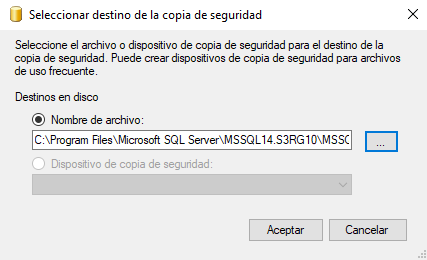
1. En la siguiente ventana seleccionar la opción ‘Agregar’:
2. Este paso es para agregar la carpeta de destino donde quedara guardada la copia de seguridad de la base de datos. Se recomienda dejar la dirección que aparece predeterminada. Clic en los puntos suspensivos:



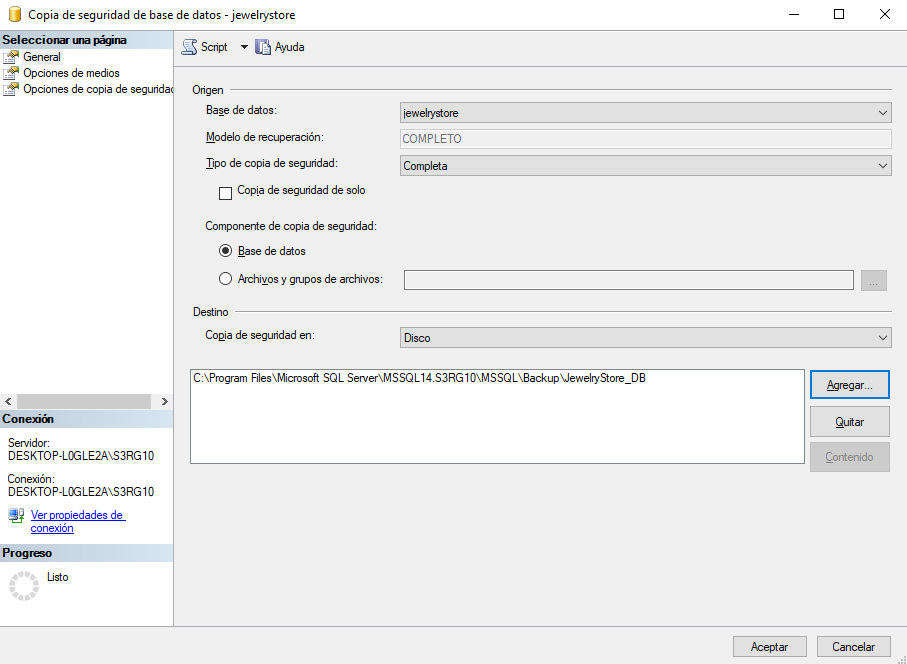
1. Escribir el nombre que se quiere para el archivo y después clic en ‘Aceptar’:



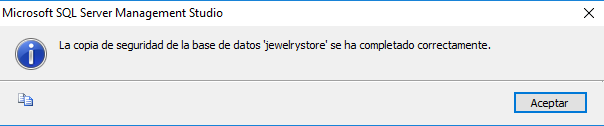
1. Como se aprecia en la siguiente imagen, el botón ‘Aceptar’ ya está disponible, dar clic de en este:



1. En este punto se puede notar que ya aparece la ruta donde quedara alojada la copia de seguridad o back up, dar clic en ‘Aceptar’:



1. Si el proceso se realizó correctamente aparecerá un mensaje como este. Clic en ‘Aceptar’:

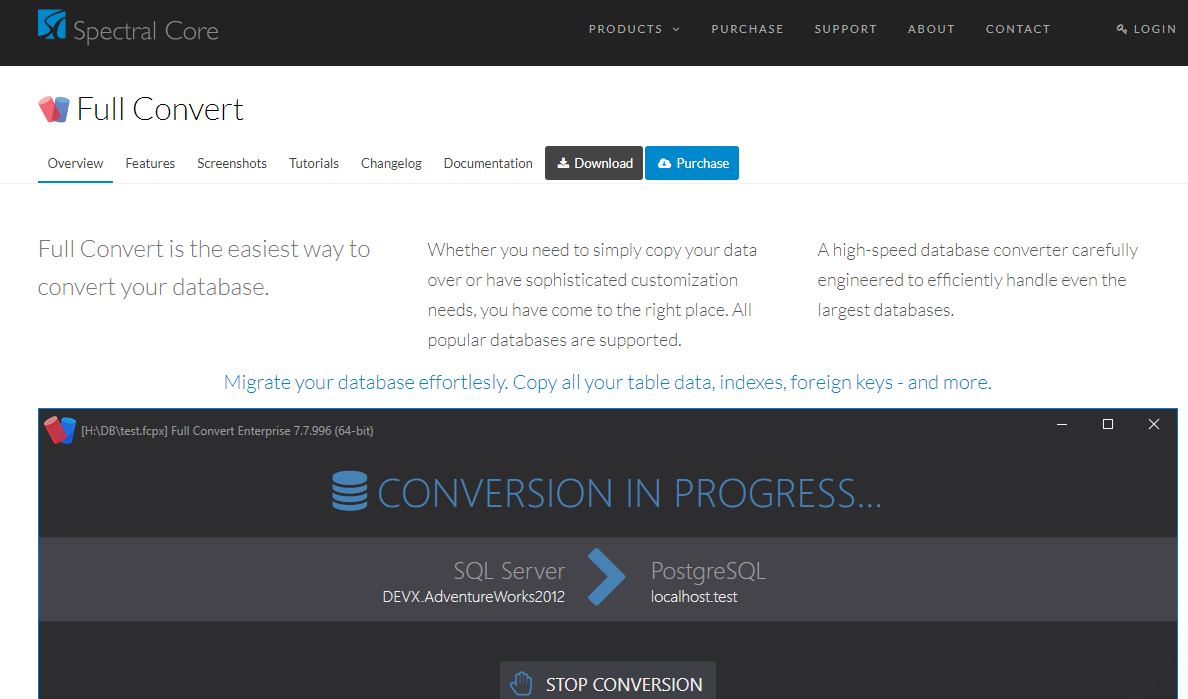


1. Para verificar que el archivo se creó correctamente, es necesario ir a la ruta que se especificó en el punto 9, en este caso fue:

*C:\Program Files\Microsoft SQL Server\MSSQL14.S3RG10\MSSQL\Back up*

#### 

Para empezar el proceso de migración de la base de datos vamos a necesitar de una herramienta llamada “Full Convert Enterprise”, para su descarga nos dirigiremos a su paina web: <https://www.spectralcore.com/fullconvert>

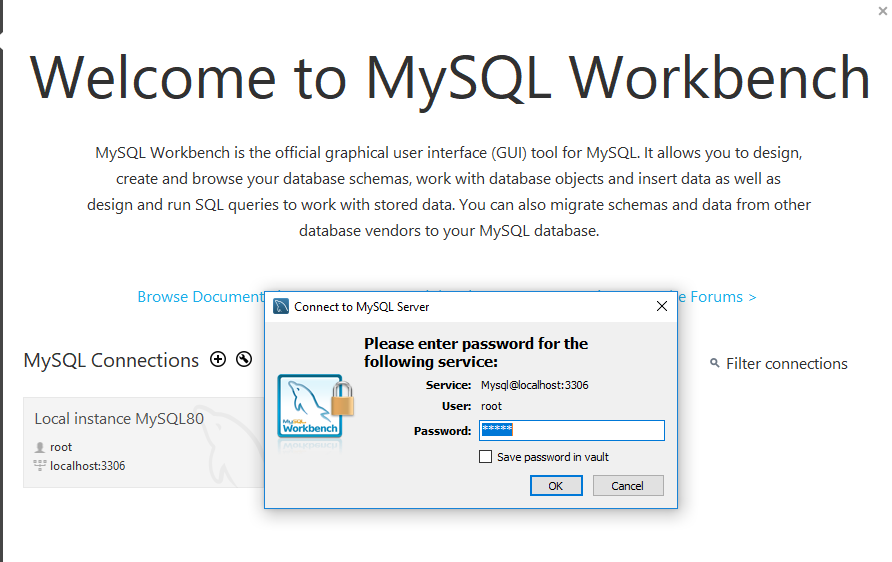


# **Proceso de ejecución**

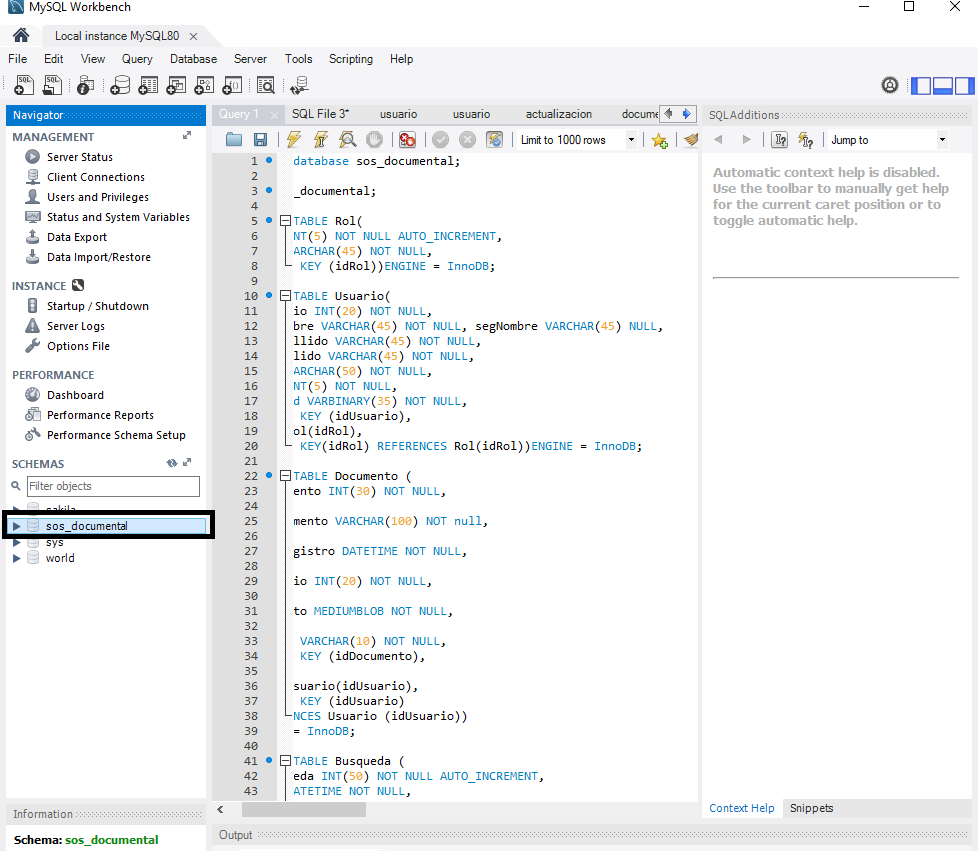
1. Abrimos mysql workbench ubicada en escritorio.



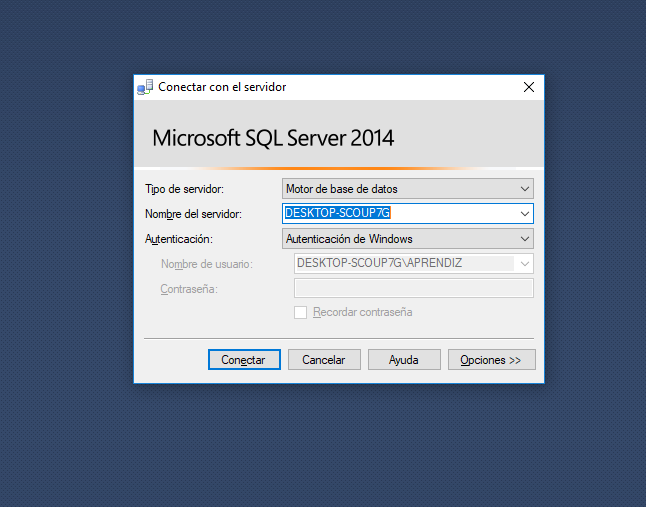
1. se realiza la respectiva conexión al servidor local



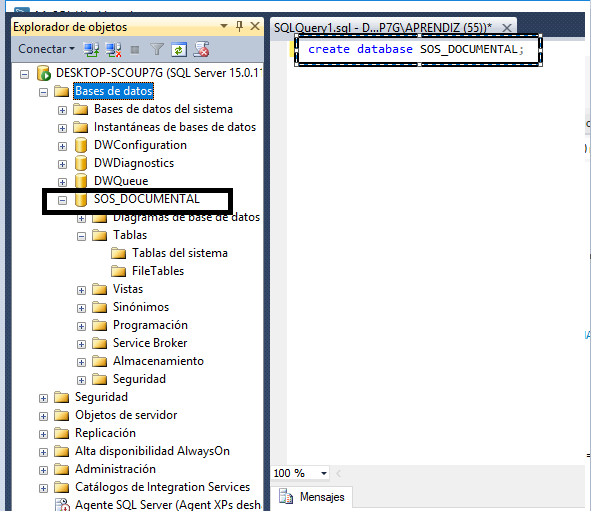
1. Una vez adentro, verificamos que la base de datos que queremos migrar se encuentre en el equipo:



1. iniciamos nuestro servido de SQL server en este caso la version 2014



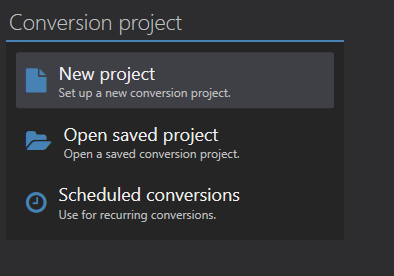
1. Una vez activo el servidor, abrimos la página de SQL server y creamos una base de datos donde vamos a migrar la informacion en nuestra caso la base de datos se va a llamar SOS\_DOCUMENTAL la cual va a estar en blanco:



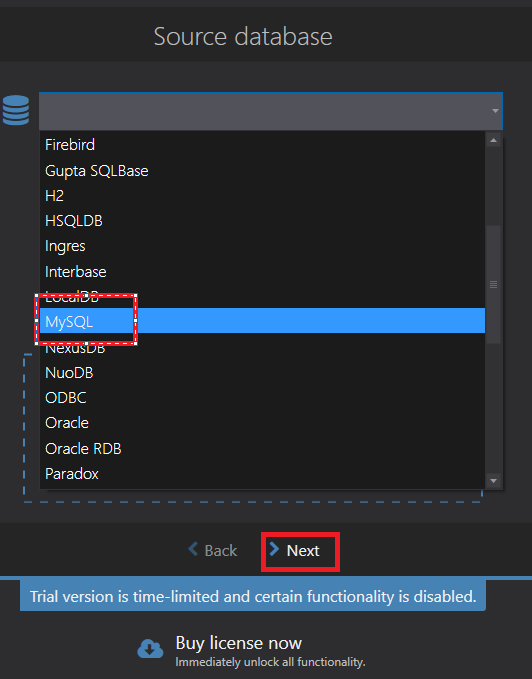
1. Una vez activo el servidor, abrimos la página de MySQL y creamos una base de datos de cotejamiento:



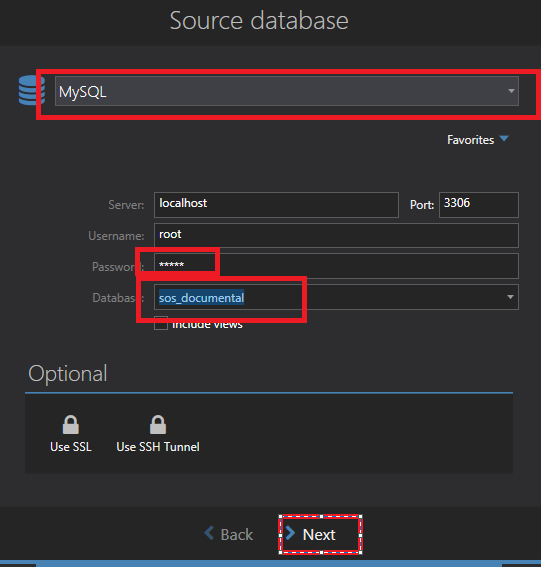
1. Dentro del programa damos click en “New project”



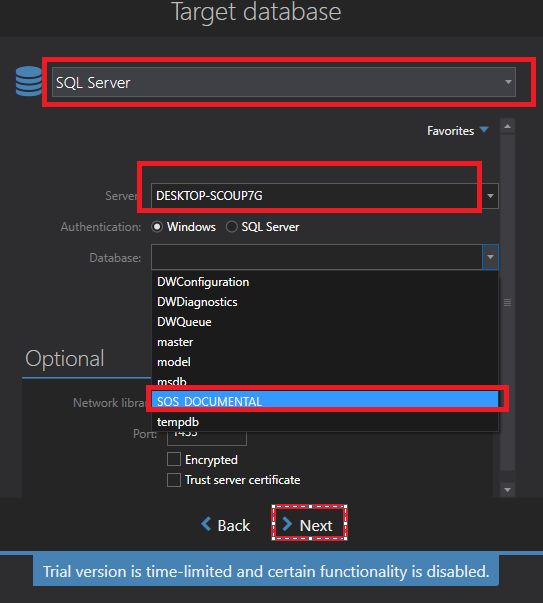
1. Se abre un cuadro de dialogo en el cual aparece los tipos de gestor de bases de datos desde los cuales podemos hacer la migración, damos click en MYSQL:



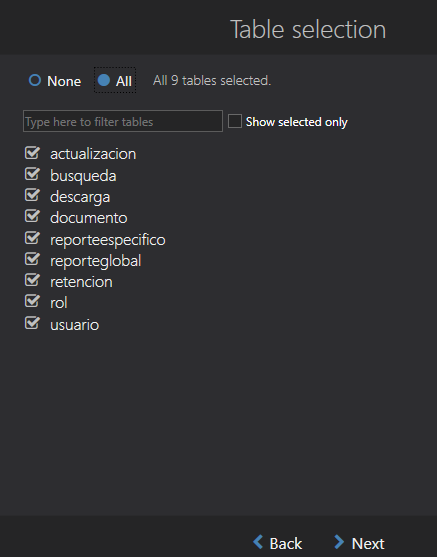
1. Se cargará el menú de autenticacion de MYSQL donde ingresamos las credenciales de administrador de servidor y nuestra base de datos, luego click en “Next” :



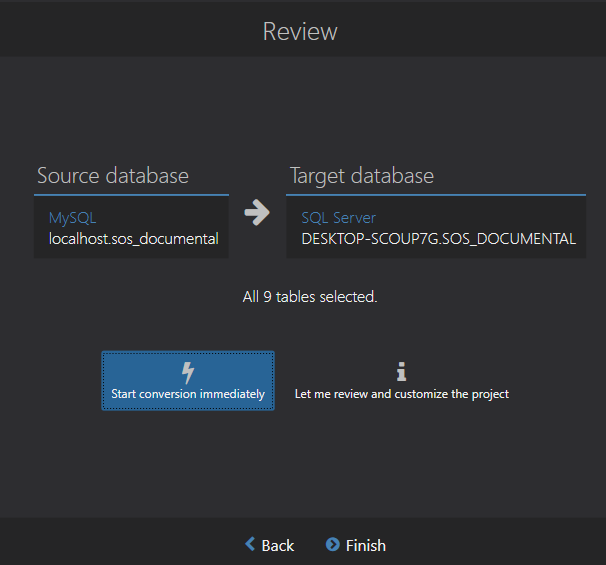
1. de igual manera seleccionamos el menú en el cual podremos escoger el destino de la migración, damos click en SQL server.En el menu de autenticación ingresamos nuestras credenciales y escogemos la base de datos creada anteriormente (SOS\_DOCUMENTAL), damos click en “Next”:



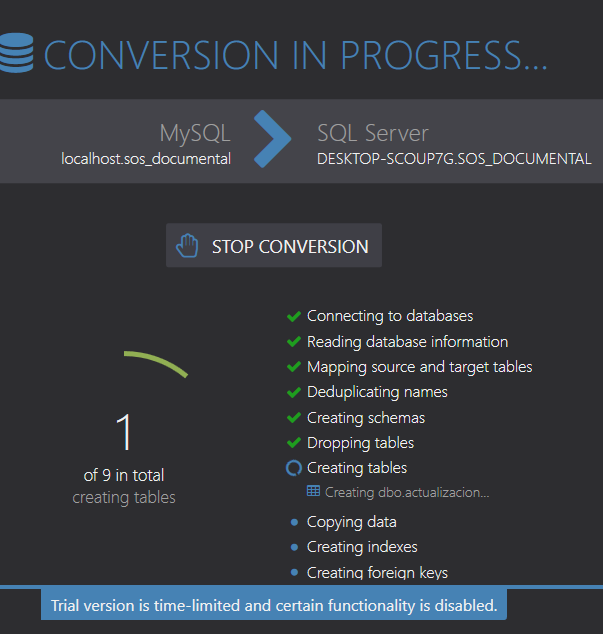
1. El programa nos muestra las tablas que vamos a migrar y podemos decidir cuales no convertir dando click en las flechas, tambien nos da varias opciones como contar los registros y personalizar las tablas a convertir, damos click en “Next”:



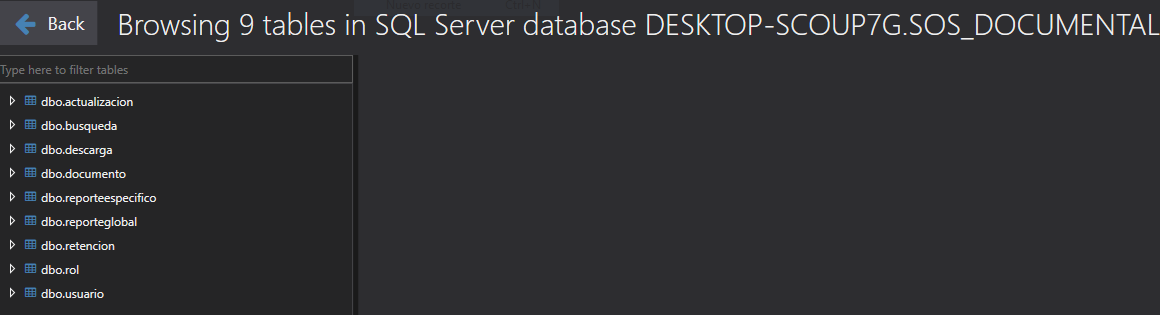
1. Por ultimos seleccionamos la opcion que queremos en este caso la primera “star conversion immediately” y finalmente damos click en “finish”



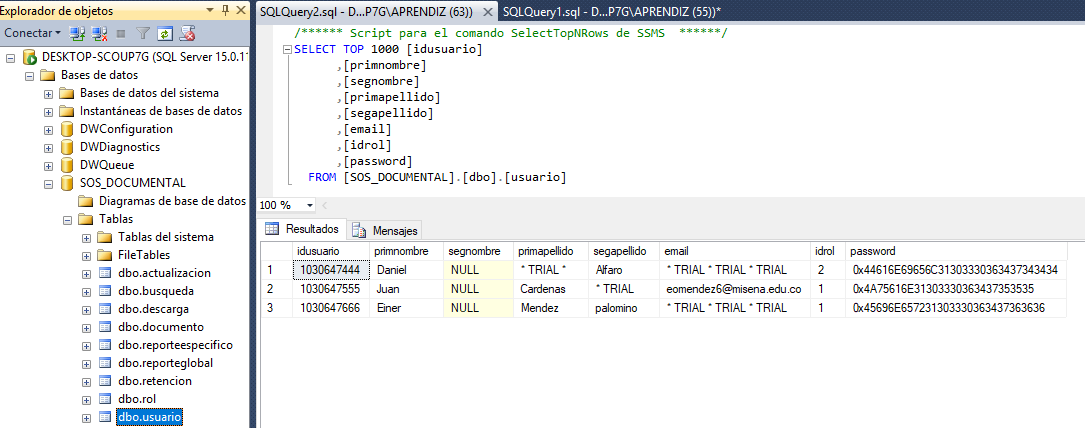
1. Se comenzara a realizar el proceso de migracion:



1. Una vez finalizado el proceso se mostrara una ventana con la informacion de la migracion, damos click en “Back” y cerramos el proceso:



1. solo queda revisar los datos ya migrador en la base de datos SQL server



**Plan de Soporte:**

#### En caso de que se presente algún altercado durante la ejecución de la migración de la base de datos no hay de qué preocuparse pues la base de datos no va a ser eliminada del motor de base de datos desde donde se ejecutará la migración, así que no se perderá la información, pero en caso de que pase por cualquier razón debemos seguir el plan de plan marcha atrás, con el cual podremos regresar la base de datos a una versión anterior.

#### 