



Universitat Oberta
de Catalunya

Máster universitario de Ciencia de Datos

Prueba de Evaluación Continua – PEC1

**Diseño y uso de bases de datos analíticas – Introducción
a las bases de datos analíticas y toma de contacto con la
máquina virtual (VDI)**

Autor:

Mario Ubierna San Mamés

Índice de Contenido

Índice de Contenido	3
Índice de ilustraciones	4
1. Parte teórica.....	7
1.1. Pregunta1	7
1.2. Pregunta 2	7
1.3. Pregunta 3	8
1.4. Pregunta 4	8
1.5. Pregunta 5	8
2. Parte práctica	10
2.1. Ejercicio 1: Configuración del entorno VDI	10
2.2. Ejercicio 2: Validación de la BBDD.....	18
2.2.1. Conectar al servidor de base de datos SQL Server	18
2.2.2. Configurar la base de datos DB_loginuoc.	19
2.2.3. Crear una tabla con el nombre T_PEC1	20
2.3. Ejercicio 3: Validación de la aplicación de ETL	22
2.3.1. Análisis preliminar del origen de datos.....	25
2.3.2. Crear una tabla en la BBDD con el nombre T_DATOS	26
2.3.3. Crear una transformación con <i>Spoon</i>	27
2.3.4. Crear un <i>job</i> que ejecute la transformación anterior	35
2.4. Ejercicio 4: Microsoft SQL <i>Analysis Services</i>	37
2.4.1. Crear una vista de origen de datos (<i>Data Source View</i>)	37
2.4.2. Crear un cubo con una única tabla de hecho T_DATOS.	40
2.4.3. Procesar e implementar el cubo en el servidor.	46
3. Bibliografía	48

Índice de ilustraciones

Ilustración 1 - Descarga de Citrix Workspace App [3].....	10
Ilustración 2 - Vista principal de la VDI asignada.	11
Ilustración 3 - Pantalla general VDI.....	11
Ilustración 4 - Información técnica sobre la VDI.....	12
Ilustración 5 - Permisos de lectura y escritura entre la VDI y el equipo local.	12
Ilustración 6 - Menú Archivo > Nuevo > Proyecto.	13
Ilustración 7 - Analysis Services > Multidimensional > Importar del servidor.....	13
Ilustración 8 - Asistente para importar la base de datos en Visual Studio.	14
Ilustración 9 - Propiedades del proyecto Visual Studio.	15
Ilustración 10 - Configuración destino de datos en Visual Studio.	15
Ilustración 11 - Ventana Origen de Datos.....	16
Ilustración 12 - Modificación Conexión de Datos.	17
Ilustración 13 - Versión de Microsoft SQL Server Management Studio 2017.....	18
Ilustración 14 - Pantalla de conexión inicial de la consola SQL Server Management Studio (SSMS).	19
Ilustración 15 - Explorador de objetos del SSMS.	19
Ilustración 16 - Modificación Recovery model a simple.	20
Ilustración 17 - Modificación actualización automática de estadísticas.	20
Ilustración 18 - Creación de la tabla T_PEC1.	21
Ilustración 19 - Comprobación de la tabla T_PEC1.....	21
Ilustración 20 - Versión de Spoon.	22
Ilustración 21 - Botón Connect.	22
Ilustración 22 - Ventana Repository Manager.....	23
Ilustración 23 - Repositorio de tipo file.	24
Ilustración 24 - Creación Repositorio file Spoon.....	25

Ilustración 25 - Estructura de los datos.	26
Ilustración 26 - Vista de la base de datos.	26
Ilustración 27 - Menú creación de tabla.	26
Ilustración 28 - Creación de la tabla T_DATOS.	27
Ilustración 29 - Creación de conexión desde Spoon a SQL Server.....	28
Ilustración 30 - Test conexión de Spoon con SQL Server.....	28
Ilustración 31 - Script de la transformación en Spoon.	29
Ilustración 32 - CSV file input.....	29
Ilustración 33 - String operations.....	30
Ilustración 34 - Filter rows.	30
Ilustración 35 - Null if.	31
Ilustración 36 - Replace in string.....	32
Ilustración 37 - Select values.....	32
Ilustración 38 - Table output.....	33
Ilustración 39 - Group by.	34
Ilustración 40 - Data validator.....	35
Ilustración 41 - Creación de un job.	35
Ilustración 42 - Script del job.	36
Ilustración 43 - Transformation.	36
Ilustración 44 - Menú creación vista de origen de datos.	37
Ilustración 45 - Asistente de creación de vista de origen de datos.	38
Ilustración 46 - Tablas y Vistas en el asistente de creación de vistas.....	39
Ilustración 47 - Nombre de la vista.....	40
Ilustración 48 - Finalización de la creación de la vista.	40
Ilustración 49 - Creación de cubo.	41
Ilustración 50 - Selección de vista en la creación de cubos.....	42
Ilustración 51 - Selección de medidas.....	43
Ilustración 52 - Selección de dimensiones.....	44
Ilustración 53 - Selección de nuevas dimensiones.	45
Ilustración 54 - Nombre del cubo.	46
Ilustración 55 - Finalización de la creación del cubo.	46
Ilustración 56 - Procesamiento del cubo.	47

Ilustración 57 - Implementación del cubo. 47

1. Parte teórica

1.1. Pregunta1

La respuesta que considero que es la correcta es:

- Las principales características de un *Data Warehouse* son: que esté orientado al tema, que los datos estén integrados, y que la información sea histórica y no volátil.

Considero que ésta es la única opción correcta, ya que en un *data warehouse* no podemos conocer los requerimientos desde un principio, por lo tanto no podemos estructurar el *data warehouse* según su funcionalidad, sino que hay que dividirla por los temas de interés.

Respecto a la integración de datos, esto tiene que ser así ya que el objetivo de un *data warehouse* es que los analistas puedan hacer uso de todos los datos y estructuras heterogéneas que tiene una empresa.

En cuanto a que la información sea histórica y no volátil, como bien se menciona antes, los analistas necesitan poder hacer estudios de la evolución del negocio y garantizar que no se pierden datos.

1.2. Pregunta 2

La respuesta que considero que es la correcta es:

- Los Maestros son entidades con información generalmente crítica, que intervienen en la mayoría de consultas para poder realizar análisis mucho más precisos. Algunos ejemplos de datos maestros pueden ser el detalle de compras, el inventario de *stock*, o el históricos de campañas de *marketing* realizado a cada uno de los clientes de la empresa.

Considero que esta es la respuesta correcta, ya que hay que saber discernir entre entidades maestras (cuya definición se correspondería con la primera opción), y datos maestros (son listas de datos que son compartidos y utilizados por diferentes aplicaciones que componen un sistema [1]). Por lo que entidades maestras serían por ejemplo: clientes, productos... Y datos maestros siguiendo el ejemplo anterior serían: el detalle de compras de cada cliente, el stock que hay de cada producto...

1.3. Pregunta 3

La respuesta que considero la correcta es:

- Las bases de datos operacionales tienen una estructura relacional y bastante estática para reforzar su estabilidad, mientras que los *data warehouse* destacan por su estructura multidimensional y dinámica, que les permite adaptarse rápidamente a las necesidades del negocio.

La anterior opción resume de la mejor forma posible los diferentes objetivos que tienen ambas bases de datos. Por un lado, las bases de datos operaciones se caracterizan por tener una estructura de datos estática, es decir, ésta no suele cambiar con frecuencia. Sin embargo, los almacenes de datos sí que buscan adaptarse rápidamente a las necesidades del negocio para mejorar la toma de decisiones, por lo que tienen que ser muy dinámicas.

1.4. Pregunta 4

La respuesta que considero la correcta es:

- Una dimensión conformada es una entidad del modelo de datos que puede ser compartida entre varios almacenes de datos, como puede ser la entidad calendario o cliente.

La anterior opción es la definición de lo que es una dimensión conformada, es por ello que fue la elegida.

1.5. Pregunta 5

La respuesta que considero la correcta es:

- Debido a los problemas que aparecen a la hora de construir el almacén de datos corporativo, se recomienda hacerlo siempre de forma simultánea con los almacenes de datos departamentales, planificando la construcción de la FIC desde un inicio.

Aunque se puede construir el almacén de datos corporativo después de los almacenes de datos departamentales, se recomienda construir dicho almacén antes o como muy tarde a la par que se construyen los departamentales. Para ello, se recomienda planificar la construcción de la FIC desde el comienzo.

2. Parte práctica

2.1. Ejercicio 1: Configuración del entorno VDI

Configurar la conexión al escritorio VDI que se utilizará durante todo el curso, tanto para realizar esta PEC como las siguientes actividades evaluables del curso.

Antes de comenzar cabe destacar que los pasos que voy a seguir a continuación, están determinados por los apuntes que se nos han proporcionado en esta asignatura [2]. Para poder realizar la configuración del entorno VDI (*Virtual Desktop Infrastructure*), lo primero que tenemos que hacer es instalar *Citrix Workspace App*, tal y como se muestra en la siguiente captura:



Ilustración 1 - Descarga de Citrix Workspace App [3].

Una vez descargado e instalado ya podemos acceder al portal de la UOC para así entrar en la máquina virtual.

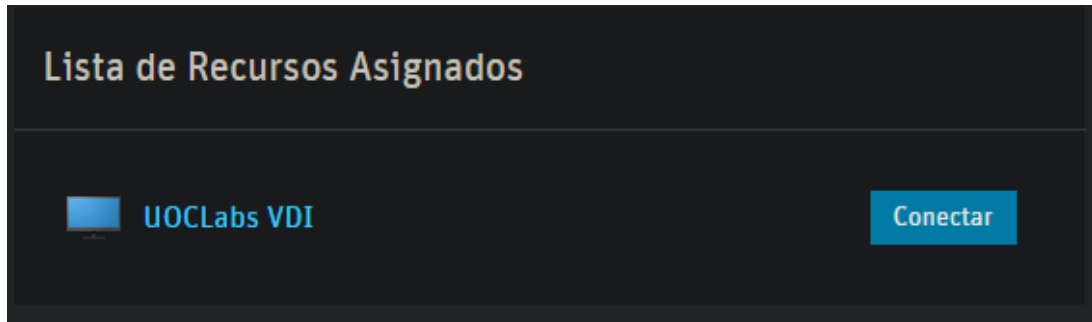


Ilustración 2 - Vista principal de la VDI asignada.

Como podemos apreciar en la anterior ilustración, vemos que tenemos una VDI asignada con el fin de realizar estas prácticas, haciendo click sobre el botón “Conectar” accedemos a la máquina virtual:

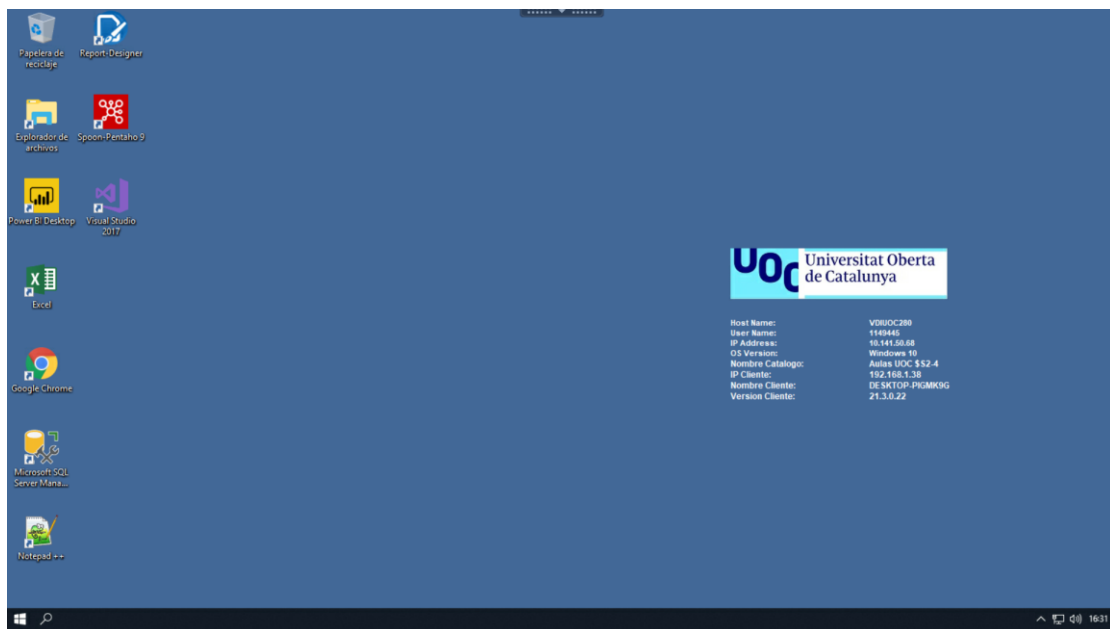


Ilustración 3 - Pantalla general VDI.

En la siguiente ilustración podemos ver de forma detalla la información técnica sobre esta VDI:



Ilustración 4 - Información técnica sobre la VDI.

Una vez que ya estamos dentro de la máquina virtual, damos acceso de lectura y escritura para que sea más fácil intercambiar archivos entre el equipo local y la máquina virtual.

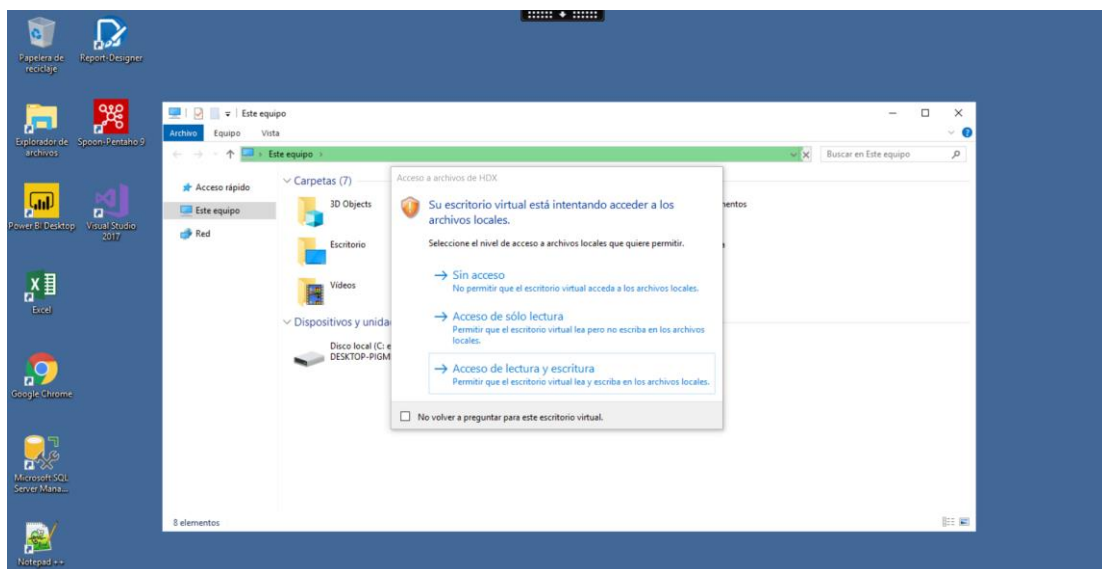


Ilustración 5 - Permisos de lectura y escritura entre la VDI y el equipo local.

Posterior a este punto tenemos que realizar la configuración de la base de datos, para ello por no hacer dos veces el mismo ejercicio, queda explicado en el ejercicio 2.

Para concluir, hay que configurar *Visual Studio*, lo primero que debemos hacer es importar la estructura de base de datos. Abrimos *Visual Studio*, seleccionamos del menú la opción *Archivo* y clickamos sobre *Nuevo proyecto*. Nos aparecerá la siguiente ventana:

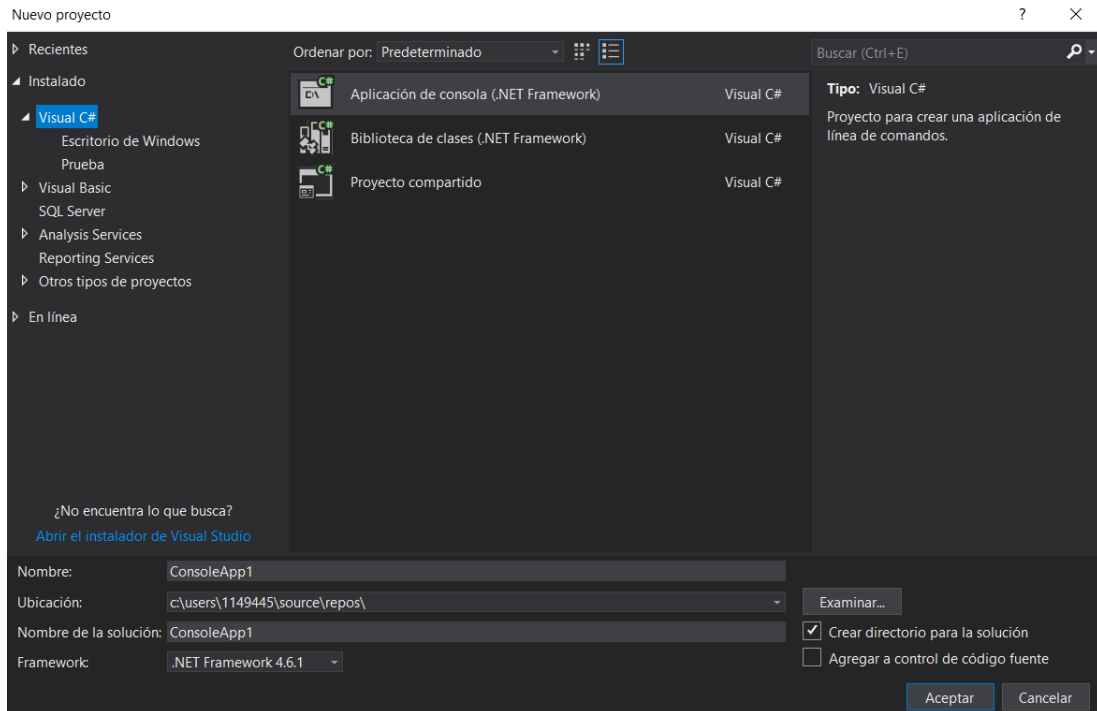


Ilustración 6 - Menú Archivo > Nuevo > Proyecto.

Seleccionamos *Analysis Services > Multidimensional > Importar del servidor* tal y como vemos en la siguiente captura:

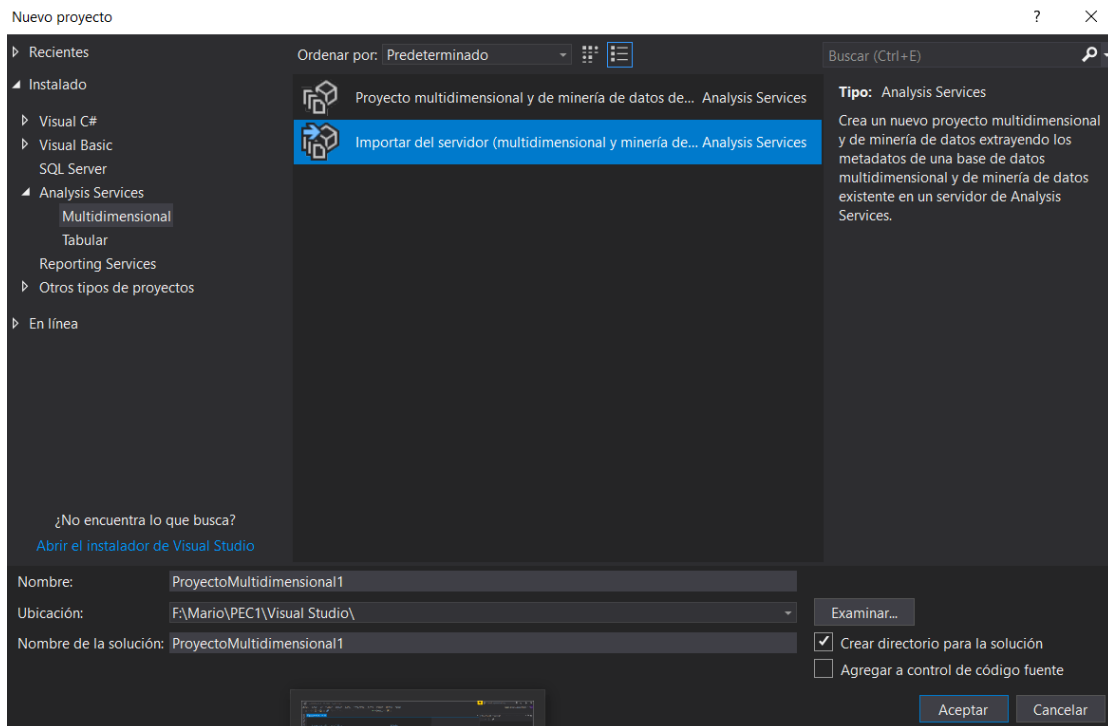


Ilustración 7 - Analysis Services > Multidimensional > Importar del servidor.

Nos aparecerá un asistente para importar la base de datos:

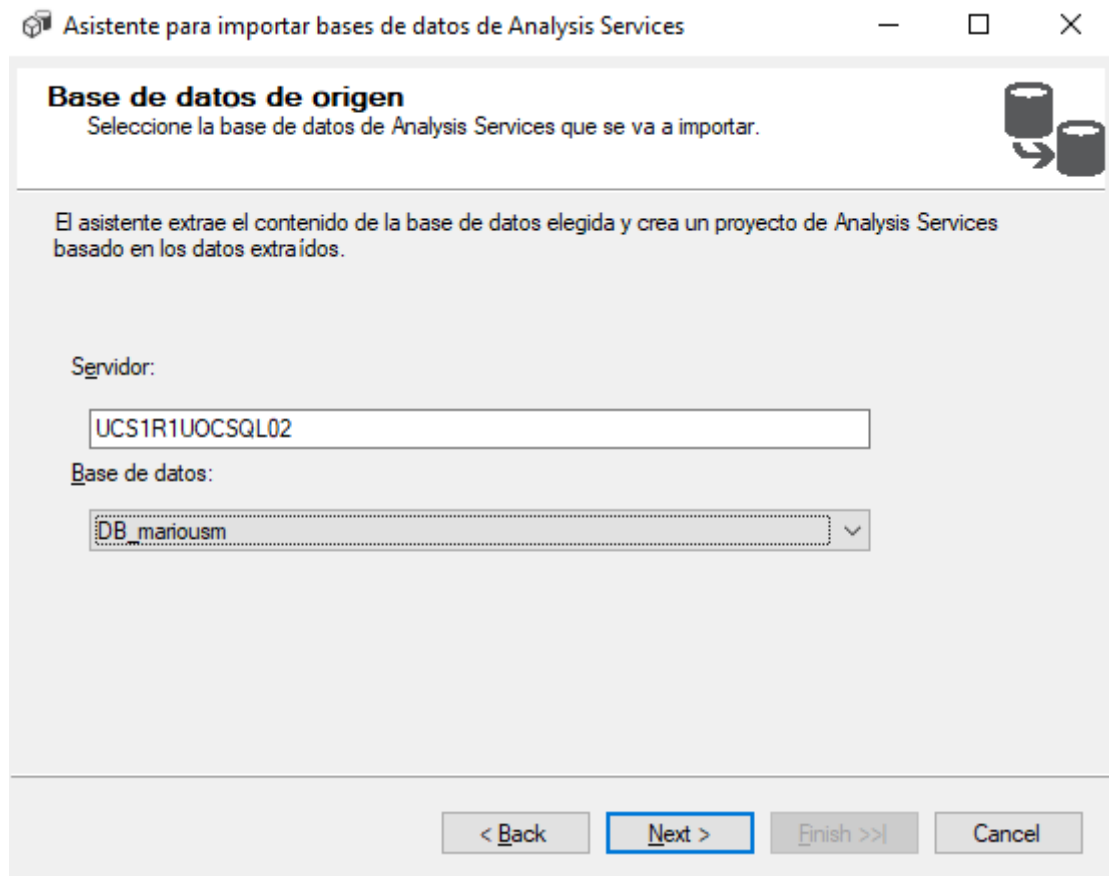


Ilustración 8 - Asistente para importar la base de datos en Visual Studio.

Escribiendo el nombre del servidor nos permitirá en el desplegable seleccionar la base de datos. Finalmente, pulsamos el botón *Next* hasta finalizar el asistente.

Una vez importada la estructura de la base de datos a nuestro proyecto, hay que configurar el destino de los datos. Para ello, nos dirigimos a las propiedades del proyecto:

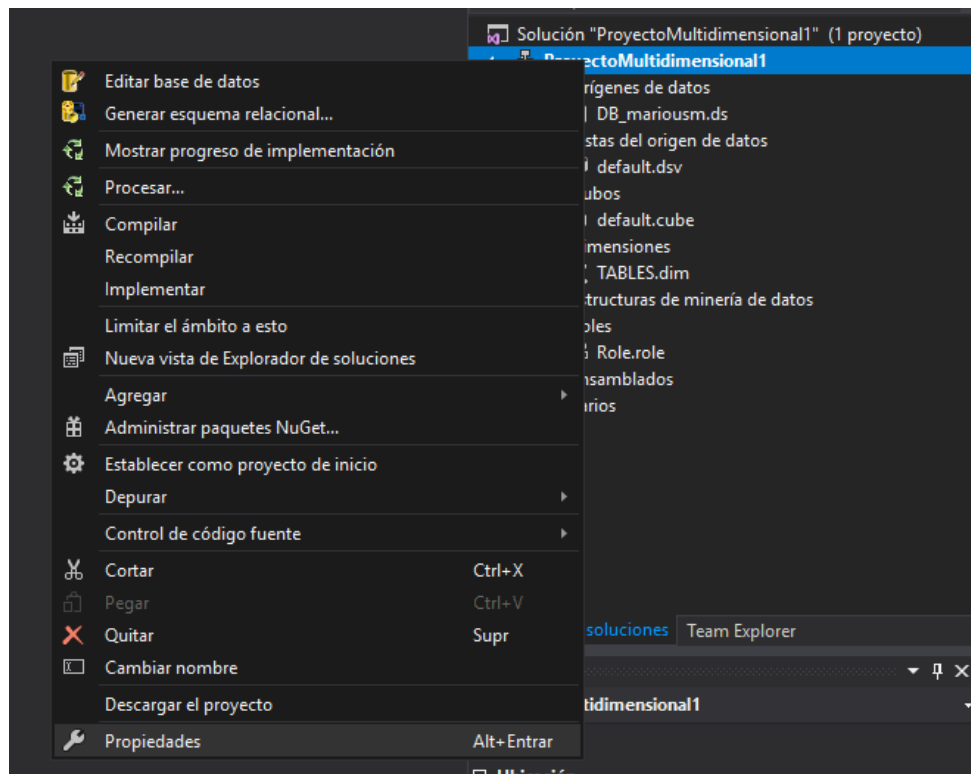


Ilustración 9 - Propiedades del proyecto Visual Studio.

En las propiedades nos vamos al apartado implementación, y en él cambiamos el nombre del servidor tal y como se muestra en la siguiente captura:

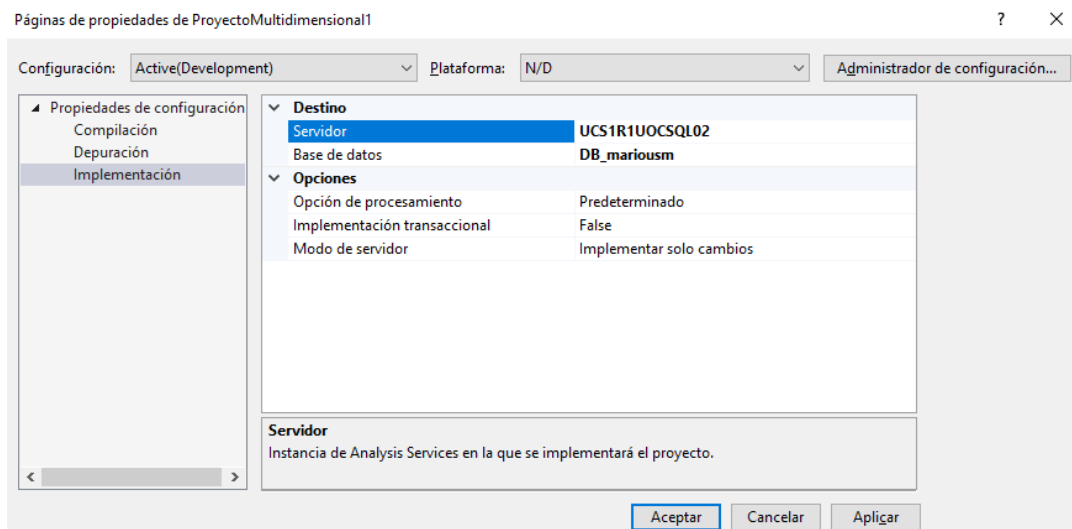


Ilustración 10 - Configuración destino de datos en Visual Studio.

Finalmente, hay que configurar el origen de los datos, para ello sobre el nombre de la base de datos en el explorador de soluciones, hacemos doble click y nos mostrará la siguiente ventana:

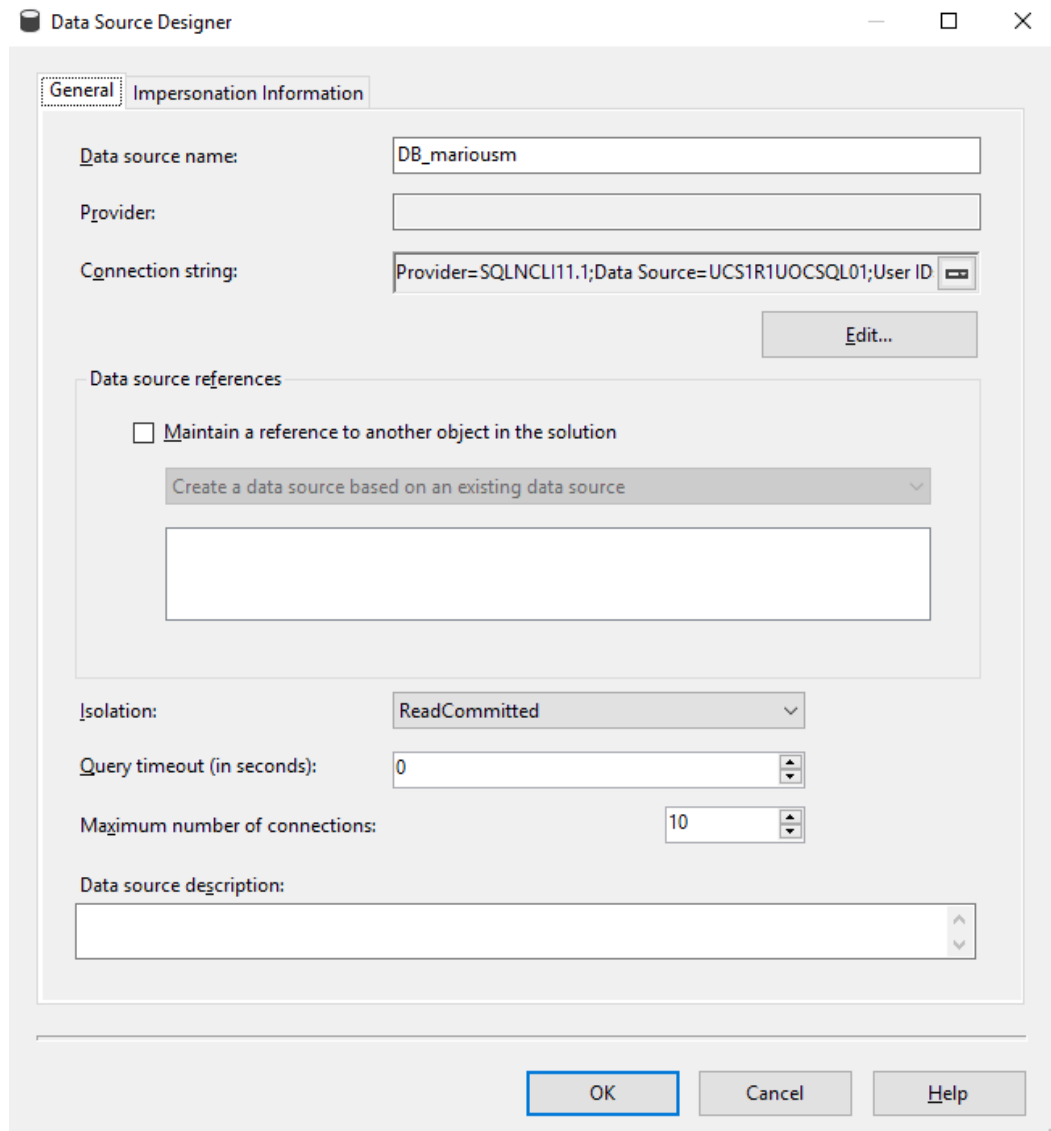


Ilustración 11 - Ventana Origen de Datos.

Pulsamos sobre el botón *Edit* para cambiar la conexión a la base de datos, una vez hemos pulsado veremos la siguiente pantalla:

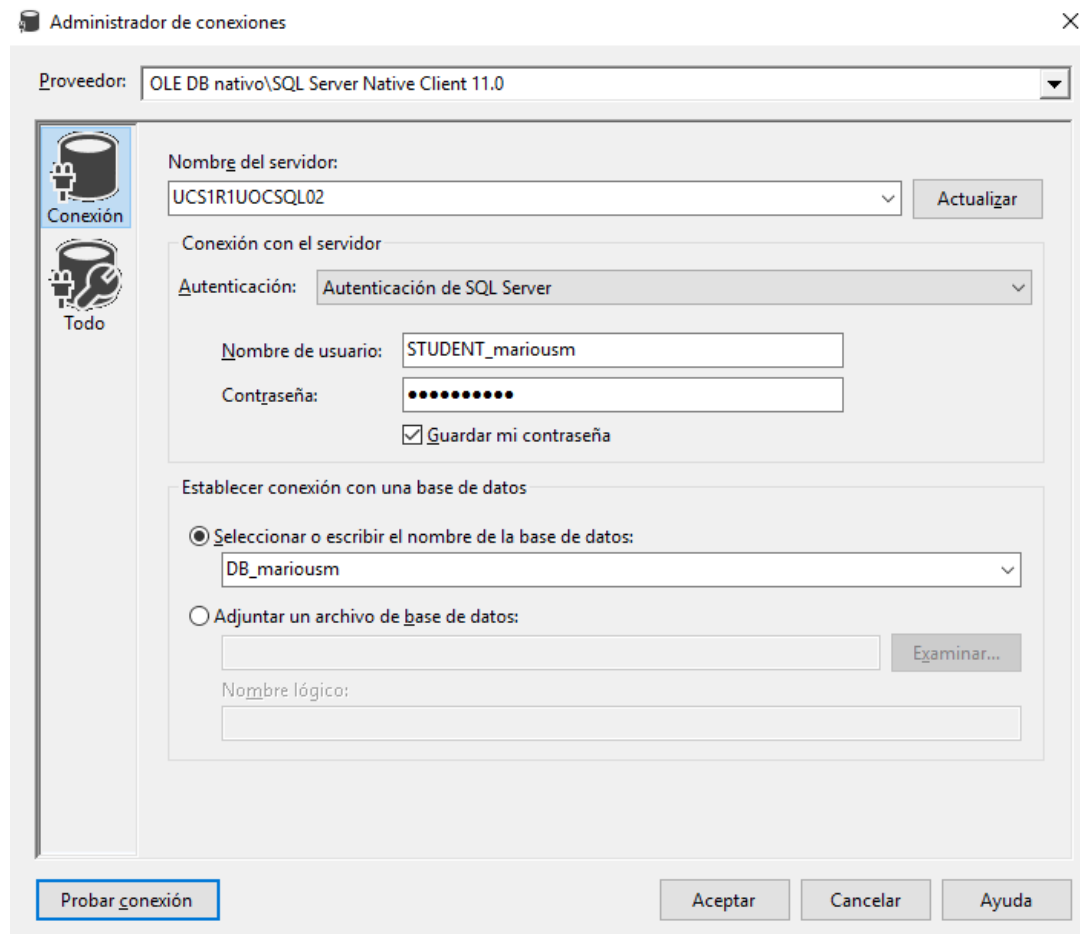


Ilustración 12 - Modificación Conexión de Datos.

Introducimos el servidor que se nos ha proporcionado, y el nombre de usuario y contraseña que usamos para acceder a *Microsoft SQL Server Management Studio 2017*.

Una vez realizado todos estos pasos podemos dar por concluido este ejercicio, ya que hemos configurado todo lo necesario para hacer la práctica.

2.2. Ejercicio 2: Validación de la BBDD

Toma de contacto con el Sistema Gestor de Bases de Datos facilitado con la máquina VDI (SQL Server 2017), configuración de una conexión con el servidor y creación de una base de datos simple con una tabla.

2.2.1. Conectar al servidor de base de datos SQL Server

La versión de *Microsoft SQL Server Management Studio 2017* la podemos ver una vez que hemos entrado en el gestor de bases de datos, para ello en la base de datos asignada ejecutamos la siguiente consulta:

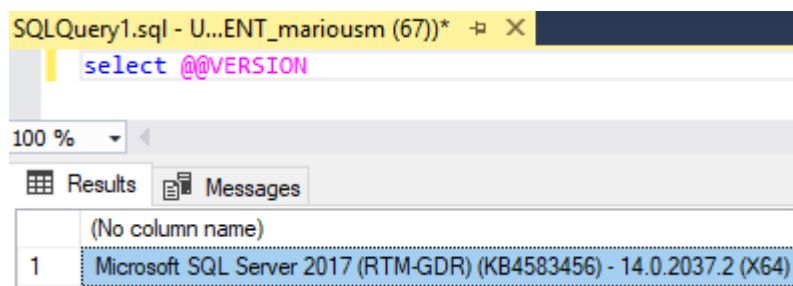


Ilustración 13 - Versión de Microsoft SQL Server Management Studio 2017.

Como podemos observar, vamos a trabajar con la versión del 2017, la cual se encuentra en la versión 14.0.2037.2.

Por otro lado, la pantalla de conexión inicial de la consola *SQL Server Management Studio (SSMS)* es la siguiente, en ella tenemos que introducir el nombre del servidor, seleccionamos la autenticación a través de SQL, por lo que haremos *logging* con nuestro usuario (*STUDENT_usuariouoc*) y la contraseña (*\$d3f4ult\$*):

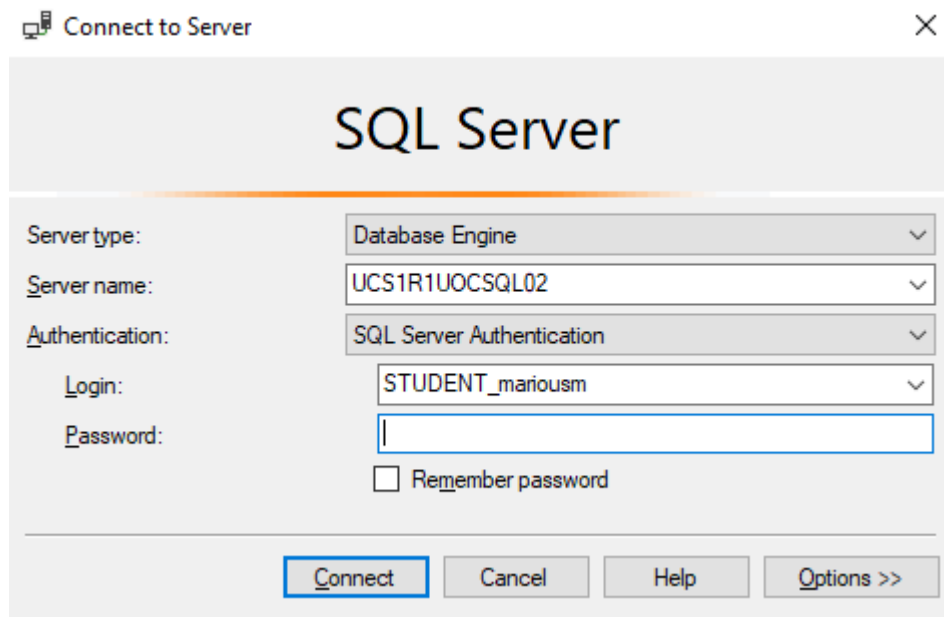


Ilustración 14 - Pantalla de conexión inicial de la consola SQL Server Management Studio (SSMS).

Finalmente, una vez iniciada la sesión podemos ver el explorador de objetos del SSMS:

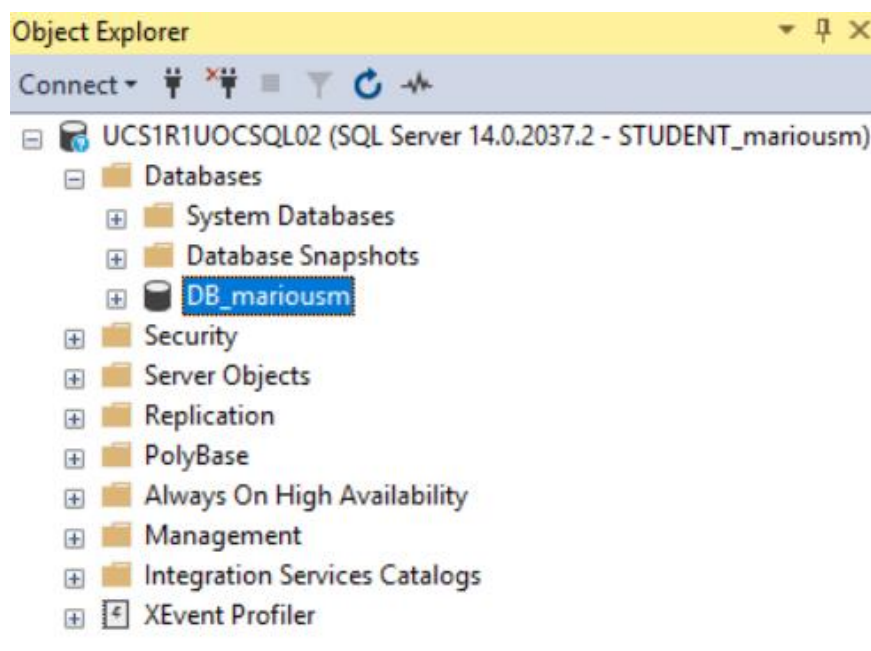


Ilustración 15 - Explorador de objetos del SSMS.

2.2.2. Configurar la base de datos DB_loginuoc.

Lo primero que debemos realizar es configurar el parámetro *Recovery Model* como “simple”, para ello accedemos a la base de datos y haciendo click derecho sobre la

instancia de la misma, nos aparecerá un menú contextual que pondrá “Options”, haciendo click sobre él nos aparecerá la siguiente ventana:

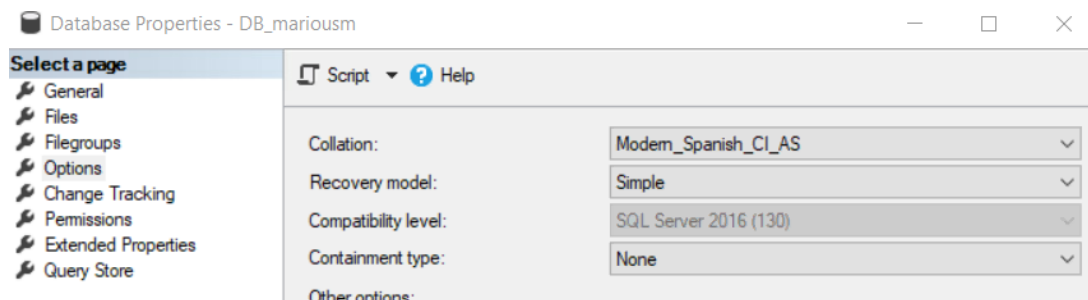


Ilustración 16 - Modificación Recovery model a simple.

Desactivamos la actualización automática de estadísticas:

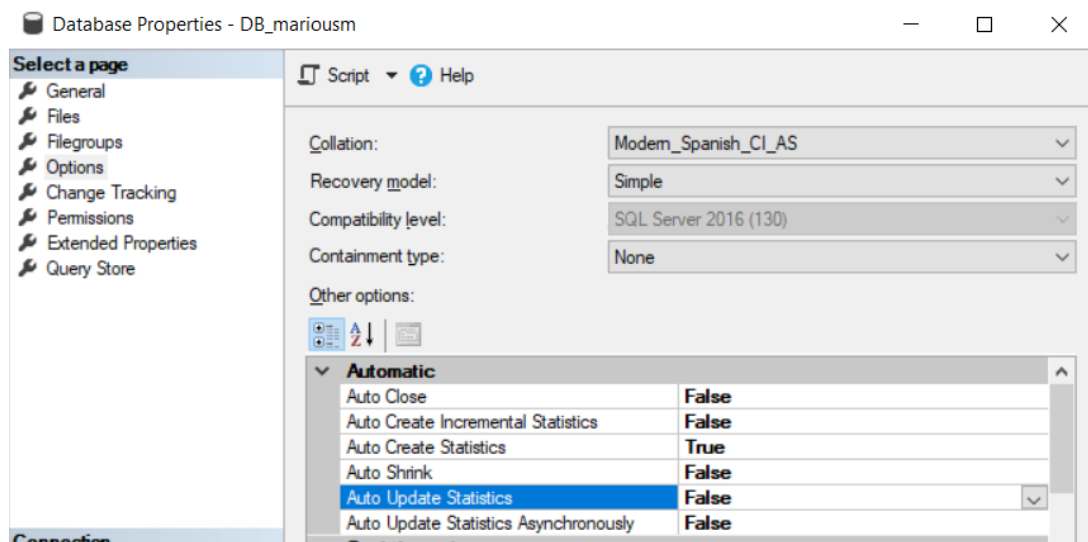
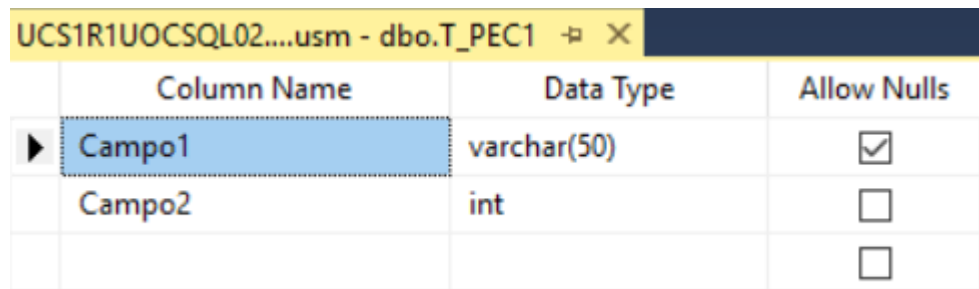


Ilustración 17 - Modificación actualización automática de estadísticas.

2.2.3. Crear una tabla con el nombre T_PEC1

Creamos una nueva tabla en la base de datos que se nos proporciona, para ello Campo1 tiene que ser de tipo *varchar(50)* y permitir valores nulos, por otro lado, el Campo2 será de tipo *int* y no permitirá valores nulos. Accedemos a la base de datos y haciendo click derecho sobre ella pulsamos sobre la acción de crear tabla, y nos aparecerá la siguiente ventana:



UCS1R1UOCSQL02....usm - dbo.T_PEC1			
	Column Name	Data Type	Allow Nulls
▶	Campo1	varchar(50)	<input checked="" type="checkbox"/>
	Campo2	int	<input type="checkbox"/>
			<input type="checkbox"/>

Ilustración 18 - Creación de la tabla T_PEC1.

Para comprobar que se ha creado correctamente hacemos una consulta *select* de la tabla, como es lógico no va a tener valores ya que no hemos hecho ningún *insert*:

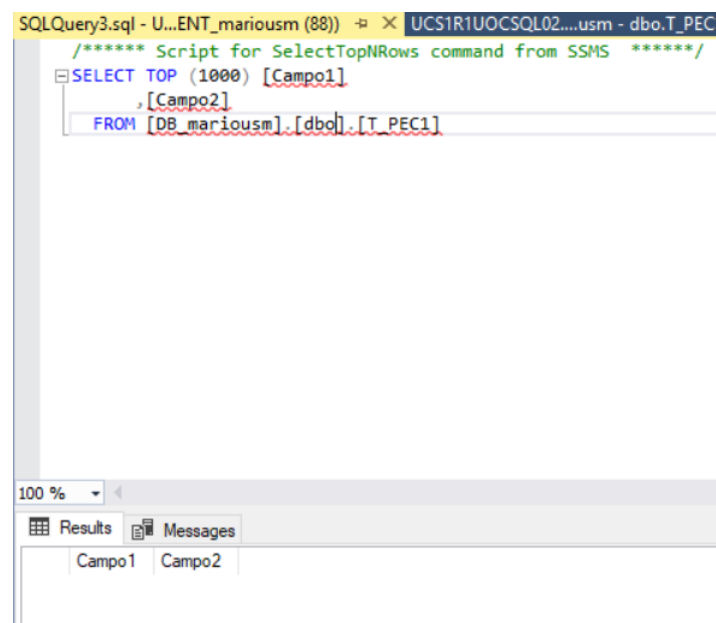


Ilustración 19 - Comprobación de la tabla T_PEC1.

2.3. Ejercicio 3: Validación de la aplicación de ETL

Toma de contacto con la aplicación de ETL suministrada en la máquina VDI (Pentaho Data Integration) y capacitar al estudiante para iniciar las actividades del curso.

Antes de comenzar con el ejercicio en sí, hay que abrir *Spoon* y capturar su versión:

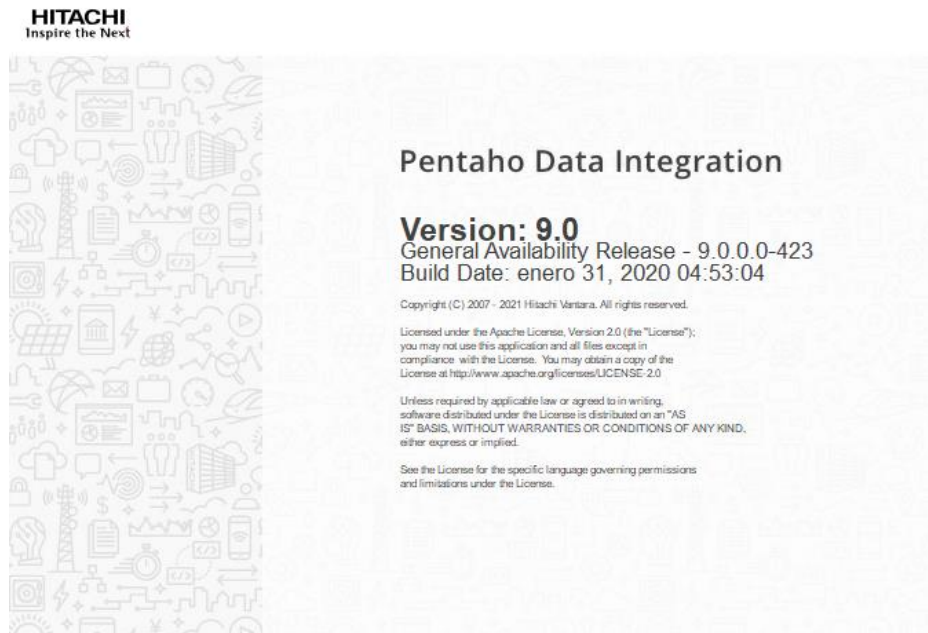


Ilustración 20 - Versión de Spoon.

Una vez que ya sabemos la versión de *Spoon* que vamos a utilizar, podemos continuar con el ejercicio. Pero antes de ello tenemos que crear un repositorio de tipo file, para ello pulsamos sobre el botón *Connect*, se encuentra a la derecha del todo como podemos apreciar en la siguiente imagen:



Ilustración 21 - Botón Connect.

Posteriormente, seleccionamos la opción *Repository Manager*, como resultado veremos la siguiente ventana:

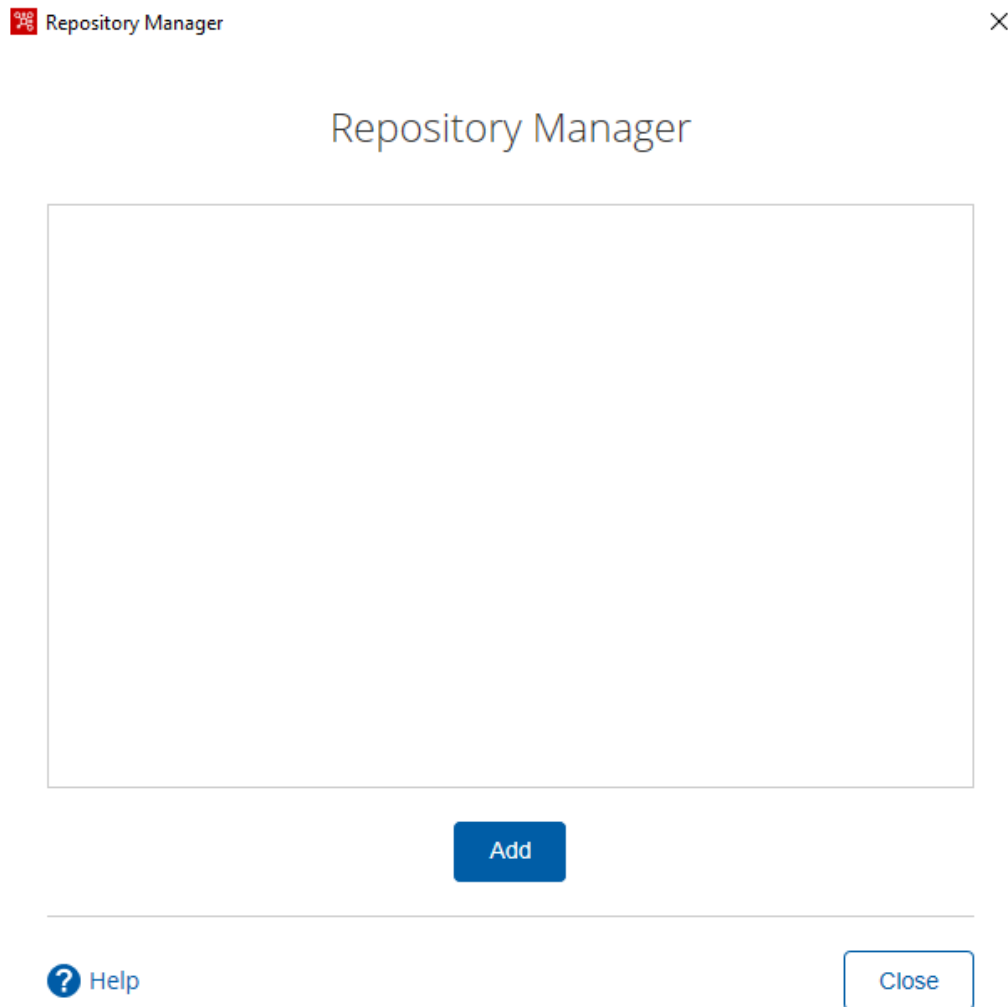


Ilustración 22 - Ventana Repository Manager.

Pulsamos sobre el botón añadir y seleccionamos la opción repositorio de tipo *file*:

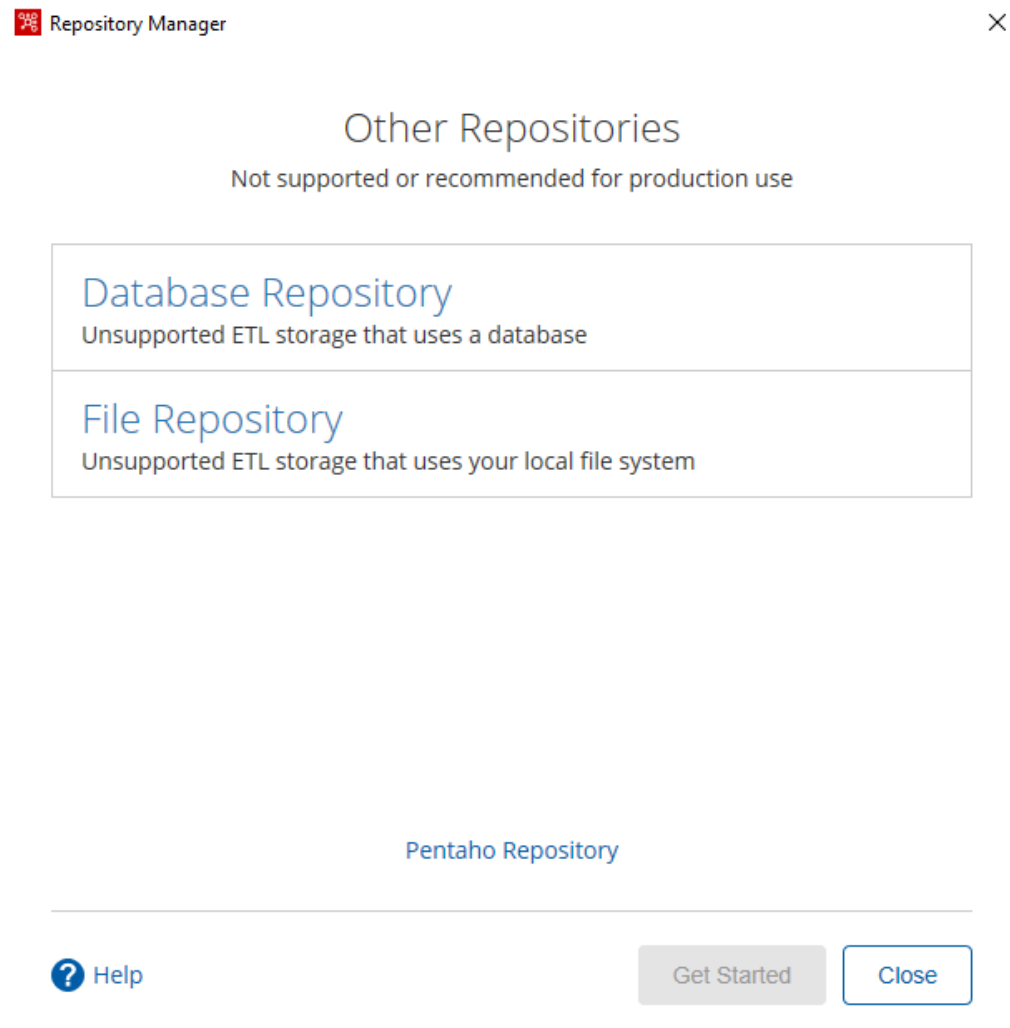


Ilustración 23 - Repositorio de tipo file.

Rellenamos los datos como en la siguiente captura y pulsamos *Finish*, con ello ya tendremos el repositorio:

New Repository Connection ×

Repository Details

Display Name

Location

☐ Don't modify items in this location

☐ Show hidden folders

Description

☐ Launch connection on startup


 [Help](#)

Ilustración 24 - Creación Repositorio file Spoon.

2.3.1. Análisis preliminar del origen de datos

Lo primero que debemos de hacer es analizar el fichero de datos que se nos proporciona. En un primer análisis debemos ver los campos que se van a cargar:

- **PracNo:** variable de tipo *int*, dicha variable no debe admitir nulos, ya que es un identificador único para cada registro, por lo que debe ser la *primary key*.
- **PracticeName:** variable de tipo *nvarchar*, ya que la información que contiene este campo es texto. Viendo algunos registros, sí que debe permitir valores nulos en dicho campo.
- **Address1:** variable de tipo *nvarchar*, estos datos también son de tipo texto y no pueden ser nulo.
- **Address2:** variable de tipo *nvarchar* y no debe admitir nulos.
- **Address3:** variable de tipo *nvarchar*, sí que debe permitir valores nulos.
- **PostCode:** variable de tipo *nvarchar*, ya que son datos alfanuméricos y no debe permitir nulos.

- **LCG**: variable de tipo *nvarchar*, y no debe admitir nulos.
- **RegisteredPacientes**: variable de tipo *int*, ya que es un valor numérico entero y no debe permitir nulos.

PracNo,PracticeName,Address1,Address2,Address3,Postcode,LCG,Registered Patients
1,DR. D IRWIN,HOLYWOOD ARCHES HEALTH CENTRE,WESTMINSTER AVENUE,BELFAST,BT4 1NS,Belfast,1328

Ilustración 25 - Estructura de los datos.

2.3.2. Crear una tabla en la BBDD con el nombre T_DATOS

Para crear la tabla nos vamos a *Microsoft SQL Server Management Studio 2017*, una vez hemos iniciado sesión tal y como se ha explicado tendremos acceso a la base de datos, en nuestro caso se llama “DB_marioum”:

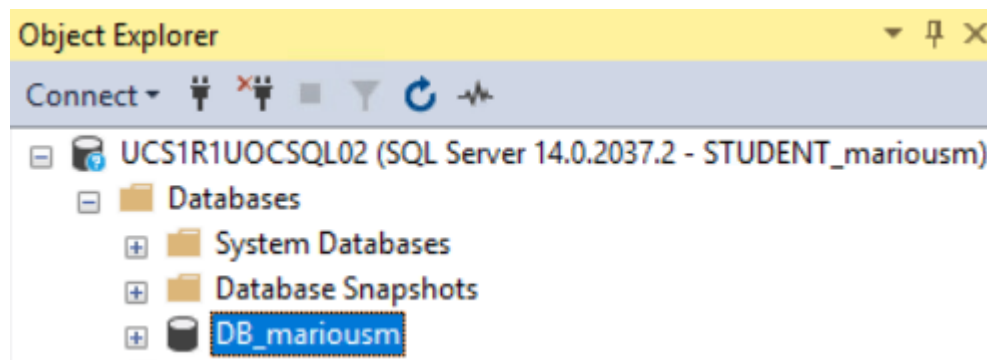


Ilustración 26 - Vista de la base de datos.

Posteriormente, hacemos click con el botón derecho del ratón sobre la opción *Table* y seleccionamos *New > Table*:

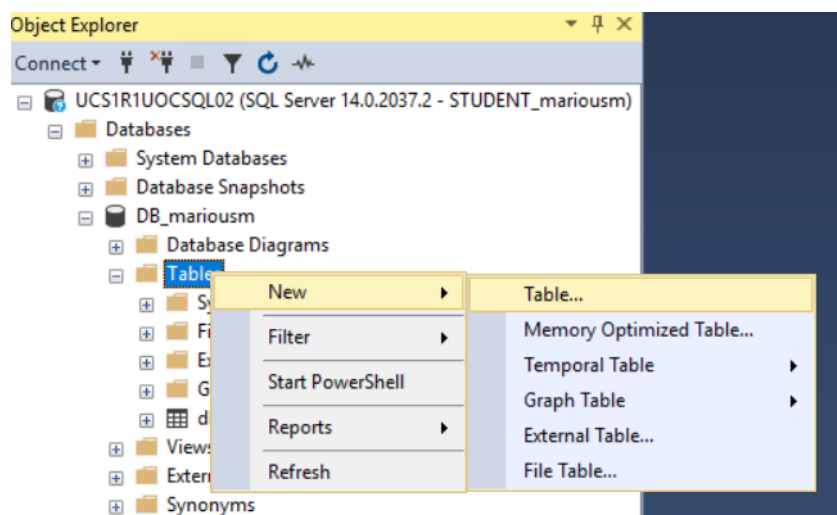
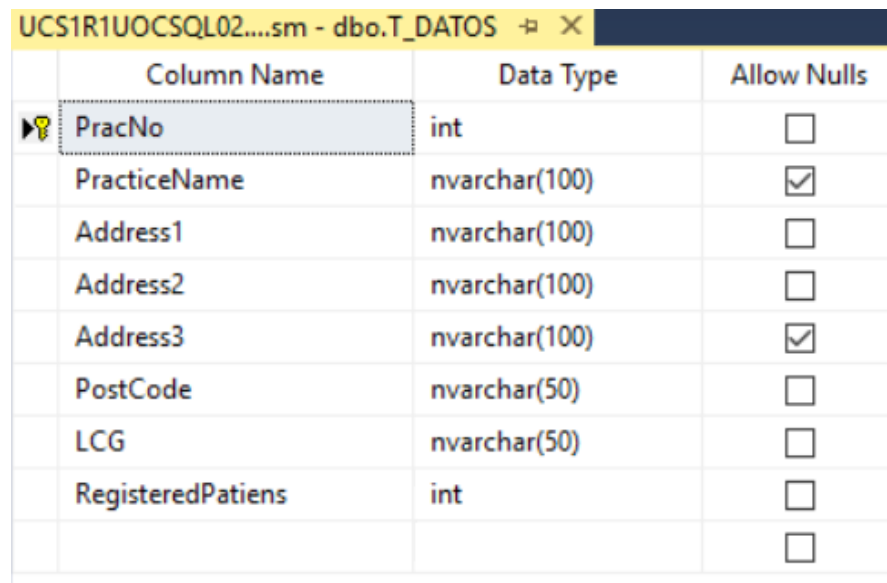


Ilustración 27 - Menú creación de tabla.

Una vez realizado el paso anterior, nos aparecerá la siguiente ventana, en la cual podemos diseñar la tabla según los datos que hemos analizado en el punto anterior:




	Column Name	Data Type	Allow Nulls
	PracNo	int	<input type="checkbox"/>
	PracticeName	nvarchar(100)	<input checked="" type="checkbox"/>
	Address1	nvarchar(100)	<input type="checkbox"/>
	Address2	nvarchar(100)	<input type="checkbox"/>
	Address3	nvarchar(100)	<input checked="" type="checkbox"/>
	PostCode	nvarchar(50)	<input type="checkbox"/>
	LCG	nvarchar(50)	<input type="checkbox"/>
	RegisteredPatiens	int	<input type="checkbox"/>
			<input type="checkbox"/>

Ilustración 28 - Creación de la tabla T_DATOS.

2.3.3. Crear una transformación con *Spoon*

Antes de comenzar, tenemos que realizar una conexión con la base de datos para así posteriormente tener acceso [4], para ello nos vamos a *File > New > Transformation*. Una vez se nos crea la transformación, podemos establecer una conexión a la base de datos, para ello volvemos a seleccionar desde el menú *File > New > Database Connection*, como resultado nos encontraremos con la siguiente pantalla:

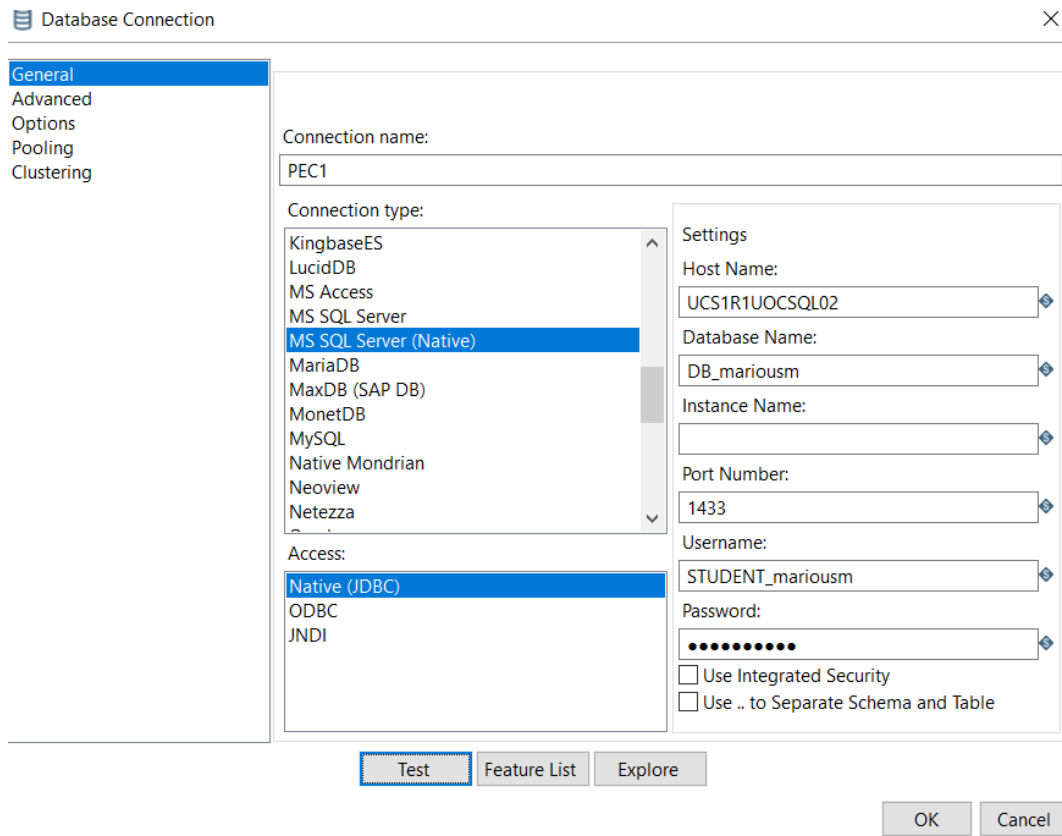


Ilustración 29 - Creación de conexión desde Spoon a SQL Server.

Lo siguiente que debemos hacer es, dar un nombre a la conexión, seleccionar el tipo de conexión (en nuestro caso *MS SQL Server (Native)*), luego escribimos el nombre del *host* (se nos ha proporcionado ese nombre en la práctica), introducimos el nombre de la base de datos, y finalmente el nombre de usuario y la contraseña. Pulsamos sobre el botón *Test* para comprobar si todo ha salido correctamente:

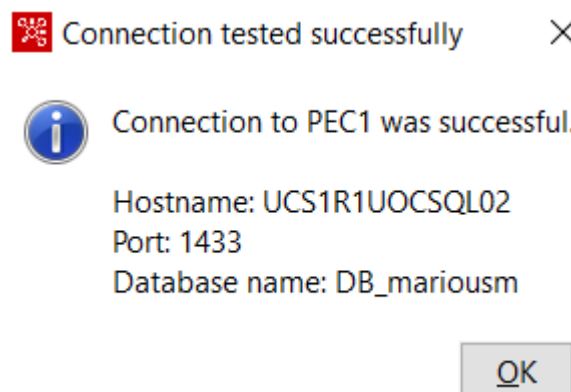


Ilustración 30 - Test conexión de Spoon con SQL Server.

Como podemos apreciar en la anterior imagen, la conexión se ha testado de forma correcta, por lo que pulsamos el botón *Ok* y dicha conexión será establecida.

Ahora sí podemos continuar con la transformación en *Spoon*, como podemos comprobar nos ha quedado el siguiente *script*:

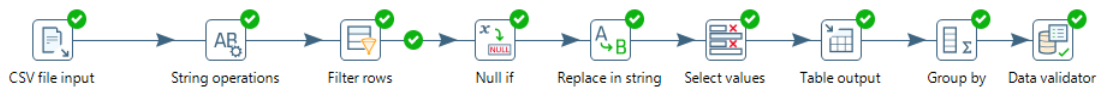


Ilustración 31 - Script de la transformación en Spoon.

Lo primero que debemos hacer es extraer la información del archivo CSV que se nos proporciona, para ello hemos usado el componente *CSV file input*, en él hemos tenido que indicar el fichero del que se iba a hacer la lectura. Además, hemos hecho una detección automática de las columnas pulsando sobre el botón *Get Fields*. Cabe destacar que el atributo *Registered Patients* se detecta como *int*, pero lo hemos transformado a *string* para solucionar una inconsistencia en los datos (en un registro aparecía una “A” en vez de un número). En la siguiente captura se puede ver el proceso descrito:

Step name: CSV file input

Filename: F:\Mario\PECT1\data\datos.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	PracNo	Integer	#	15	0	\$,	.	none
2	PracticeName	String		28		\$,	.	none
3	Address1	String		46		\$,	.	none
4	Address2	String		37		\$,	.	none
5	Address3	String		31		\$,	.	none
6	Postcode	String		8		\$,	.	none
7	LCG	String		13		\$,	.	none
8	Registered Patients	String	#	15	0	\$,	.	none

Buttons: Help, OK, Get Fields, Preview, Cancel

Ilustración 32 - CSV file input.

Después, tal y como se comentaba en la práctica se tenía que transformar todos los datos de los campos de tipo texto a mayúsculas, para ello hemos usado el componente *String operations*, gracias a él podemos indicar que las columnas que son texto sus

valores se transformen a mayúsculas, tan solo hay que establecer *Upper* en la columna *Lower/Upper*:

String operations

Step name:

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	PracticeName		none	upper	none			N	None	none	none
2	Address1		none	upper	none			N	None	none	none
3	Address2		none	upper	none			N	None	none	none
4	Address3		none	upper	none			N	None	none	none
5	Postcode		none	upper	none			N	None	none	none
6	LCG		none	upper	none			N	None	none	none

Help OK Get fields Cancel

Ilustración 33 - String operations.

Una vez realizada la anterior transformación, podemos corregir las inconsistencias en los datos que hemos encontrado, básicamente había tres problemas y para cada uno de ellos hemos usado un componente diferente:

- Al final del fichero encontramos dos filas con todos los valores nulos, lo cual no es correcto. En este caso hemos usado *Filter rows*, ya que podemos indicar que si uno o varios campos son nulos ignore dichas filas. Como la tabla que hemos creado tiene una *primary key* que es *PracNo*, hemos considerado solamente comprobar que dicho campo tenga valor.

Filter rows

Step name:

Send 'true' data to step:

Send 'false' data to step:

The condition:

☐ +

PracNo IS NOT NULL

Help OK Cancel

Ilustración 34 - Filter rows.

- Otro problema que nos hemos encontrado es que hay dos columnas que permiten valores nulos (*PracticeName*, *Address3*), estos valores no se

guardaban en base de datos como nulos sino que se almacenaban como *string* cuyo valor era *NULL*. Para ello hemos usado el componente *Null if*, nos permite convertir esos valores a nulos.

[illegible]

Ilustración 35 - Null if.

- El último error que nos hemos encontrado es que la columna *Registered Patients* tenía un registro con un valor “A” en vez de tener un número, lo cual sería lo correcto. Para ello hemos usado dos componentes, el primero *Replace in string* (conseguimos modificar ese valor “A” por un 0), y el segundo *Select values* (con él transformamos la columna de *string* a *int*). En las siguientes capturas podemos ver la configuración de ambos componentes:

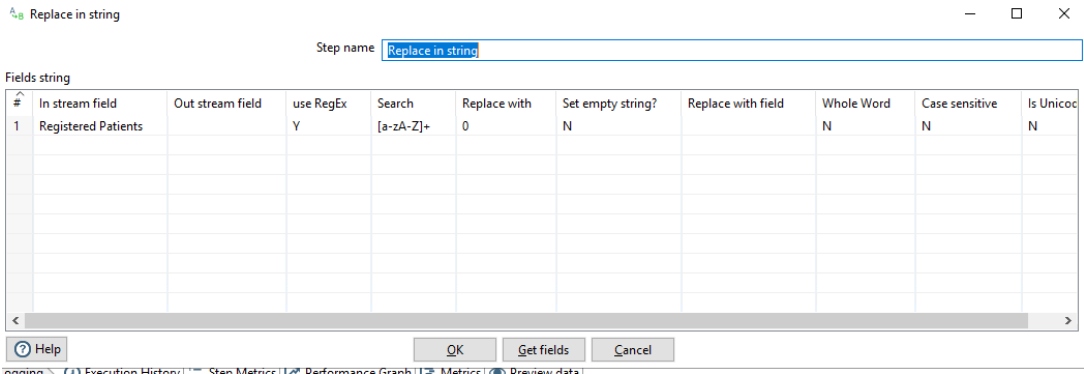


Ilustración 36 - Replace in string.

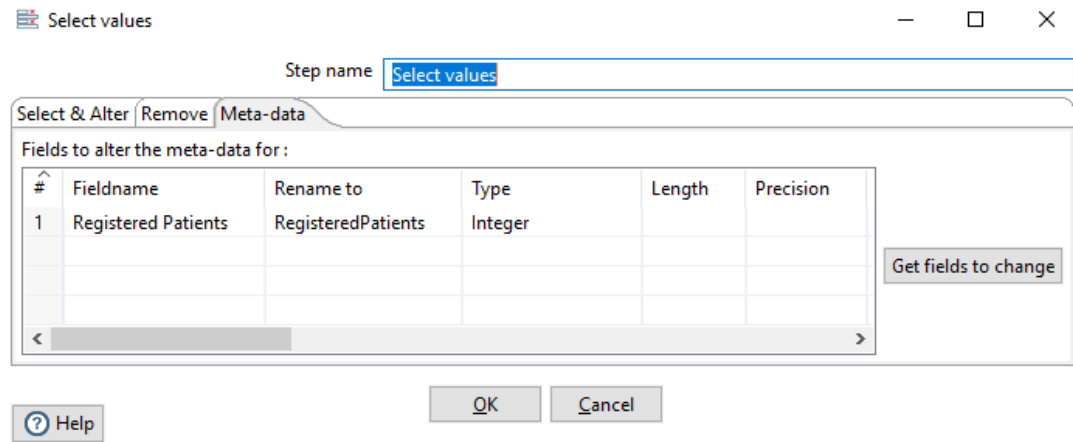


Ilustración 37 - Select values.

Una vez solucionados estos problemas, podemos hacer la carga de la información en la base de datos. Haciendo uso del componente *Table output* conseguimos que la salida de todos los pasos anteriores se guarde en la tabla de la base de datos correspondiente, en nuestro caso *T_DATOS*.

Table output

Step name: Table output

Connection: PEC1

Target schema: dbo

Target table: T_DATOS

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options Database fields

Partition data over tables: ☐

Partitioning field: [dropdown]

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table: [dropdown]

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: [dropdown]

Help OK Cancel SQL

Ilustración 38 - Table output.

Finalmente, hay que validar que la información se ha guardado de forma correcta en la base de datos. Con el fin de alcanzar este objetivo hemos usado los componentes *Group by* y *Data validator*, con el primero conseguimos contar el número de filas que hay en la tabla (solamente hay que indicar que el tipo a devolver sea el número de filas), y con el segundo validamos que el resultado obtenido es el deseado (para ello establecemos que la variable de salida del componente anterior sea la entrada de este y validamos que sea igual a 349 en este caso).

[illegible]

Ilustración 39 - Group by.

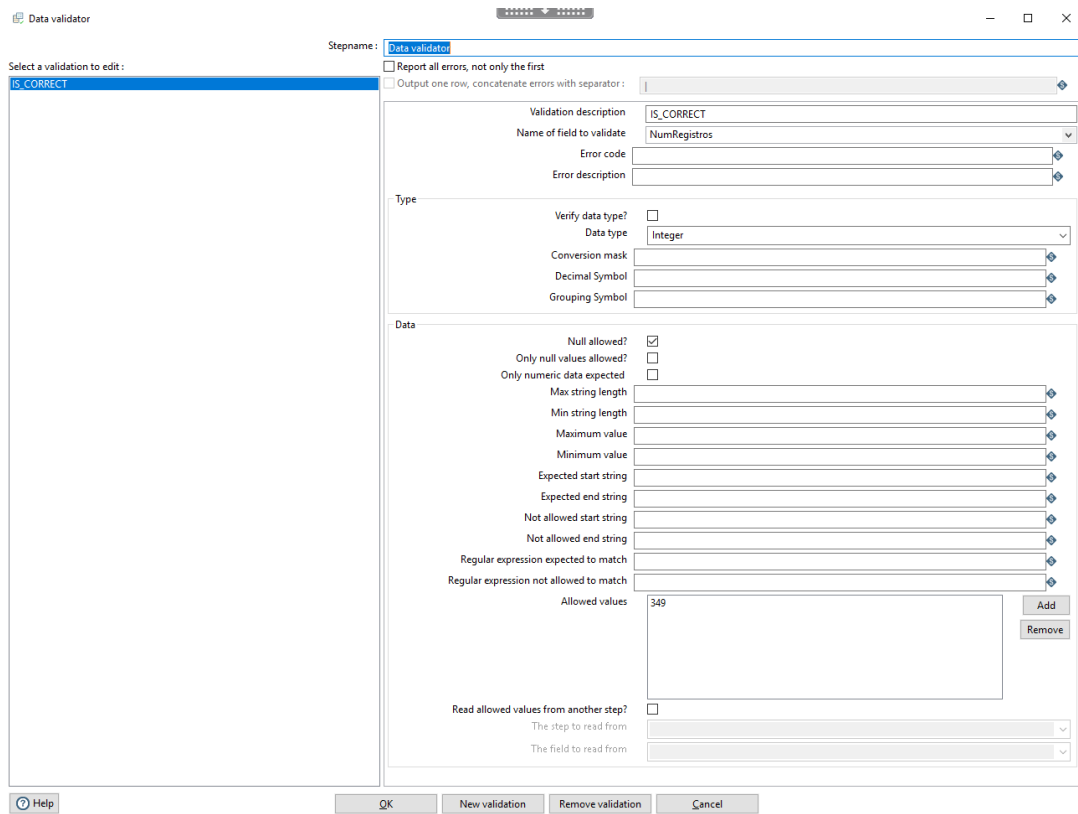


Ilustración 40 - Data validator.

Como resultado de todos los procesos anteriores conseguimos que se guarde la información en la base de datos de una forma correcta, y validamos que es verdad que se ha almacenado dicha información.

2.3.4. Crear un *job* que ejecute la transformación anterior

Para crear un *job* lo primero que debemos hacer es dirigirnos a *File > New > Job*:

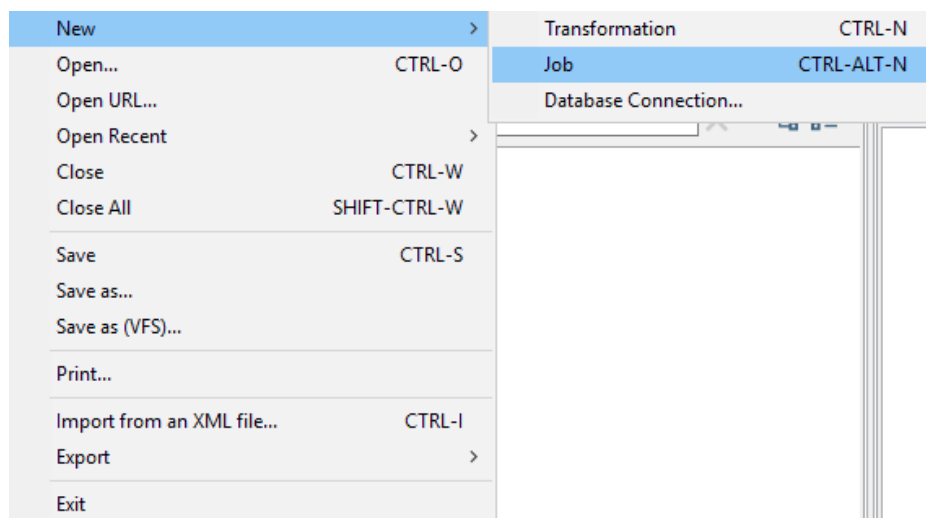


Ilustración 41 - Creación de un job.

Una vez que nos abre la pestaña, podemos realizar el *job*, en nuestro caso nos queda el siguiente *script*:



Ilustración 42 - Script del job.

Como podemos apreciar en la anterior captura tenemos tres componentes:

- *Start*: es el encargado de comenzar el *job*, en este caso no hemos tenido que cambiar ningún parámetro.
- *Success*: permite indicar al *job* que ha terminado el trabajo, al igual que con el *start* no hemos tenido que cambiar ningún parámetro.
- *Transformation*: este componente se encarga de incorporar la transformación a nuestro *job*. Para ello, tenemos que indicar por parámetro la transformación a la que va a hacer referencia el componente tal y como se ve en la siguiente ilustración:

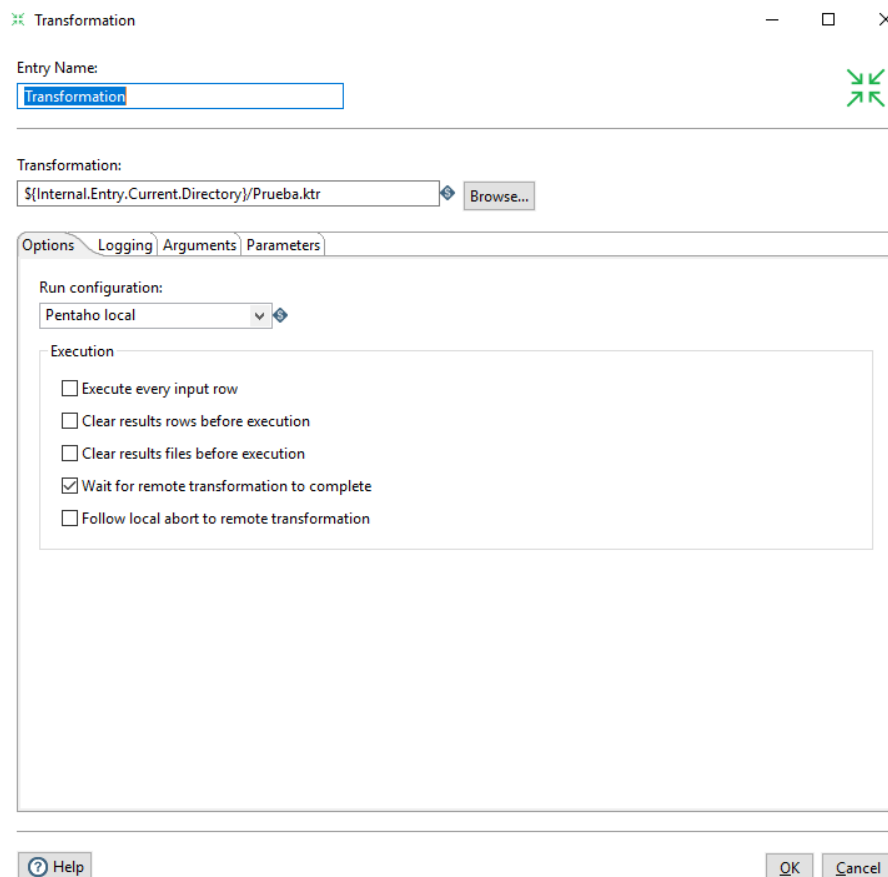


Ilustración 43 - Transformation.

2.4. Ejercicio 4: Microsoft SQL Analysis Services

Toma de contacto con la aplicación MOLAP suministrada en la máquina VDI (Microsoft SQL Analysis Services) y capacitar al estudiante para iniciar las actividades del curso.

Cabe destacar que los tres primeros apartados se han explicado en el ejercicio 1 de esta práctica, por lo partimos desde esos puntos para continuar con este ejercicio.

2.4.1. Crear una vista de origen de datos (*Data Source View*)

Para crear una vista de origen de datos tenemos que pulsar botón derecho sobre vistas del origen de datos > Nueva vista del origen de datos, tal y como se muestra en la siguiente captura:

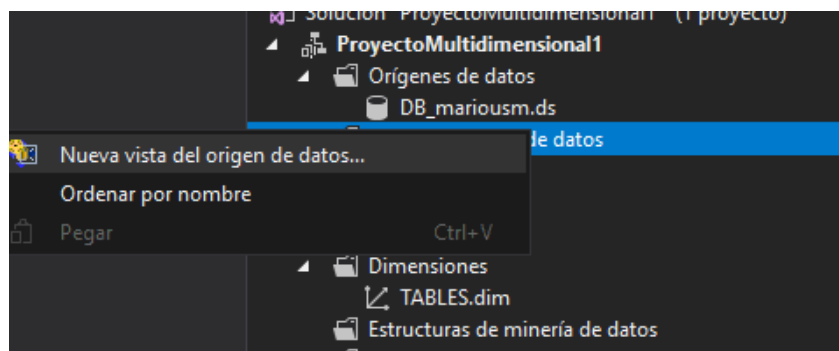


Ilustración 44 - Menú creación vista de origen de datos.

Una vez seleccionada la opción del menú, veremos la siguiente ventana:

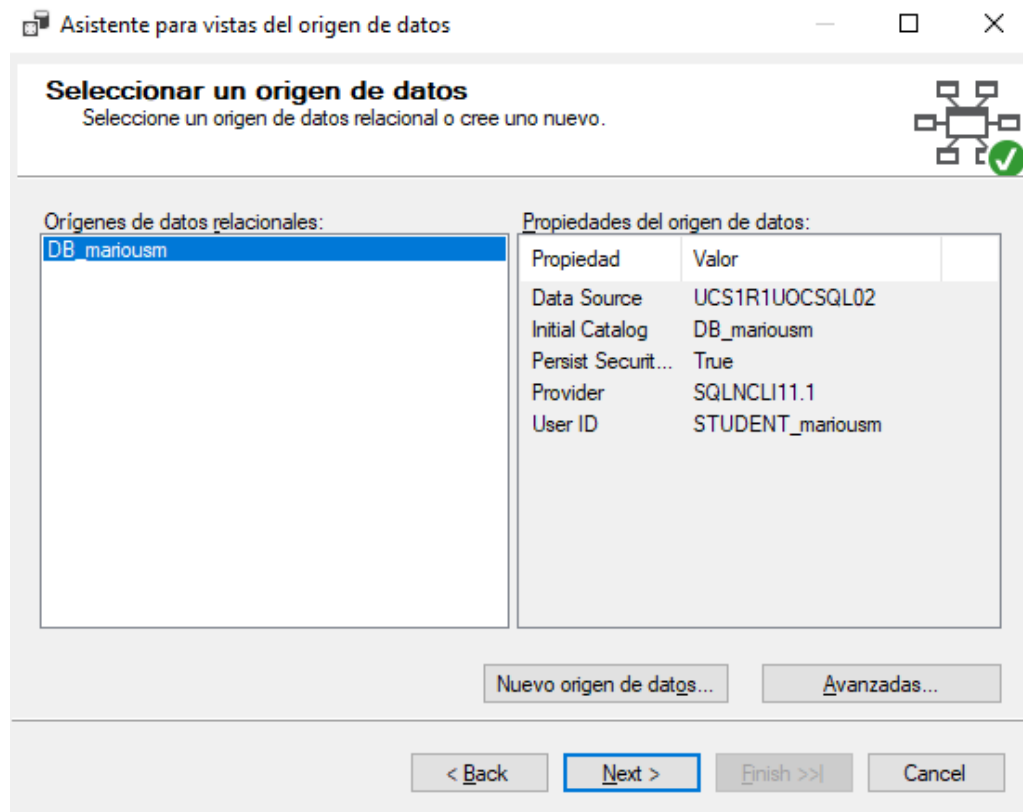


Ilustración 45 - Asistente de creación de vista de origen de datos.

Pulsando sobre el botón *Next*, podremos ver las diferentes tablas y vistas que tiene nuestra base de datos:

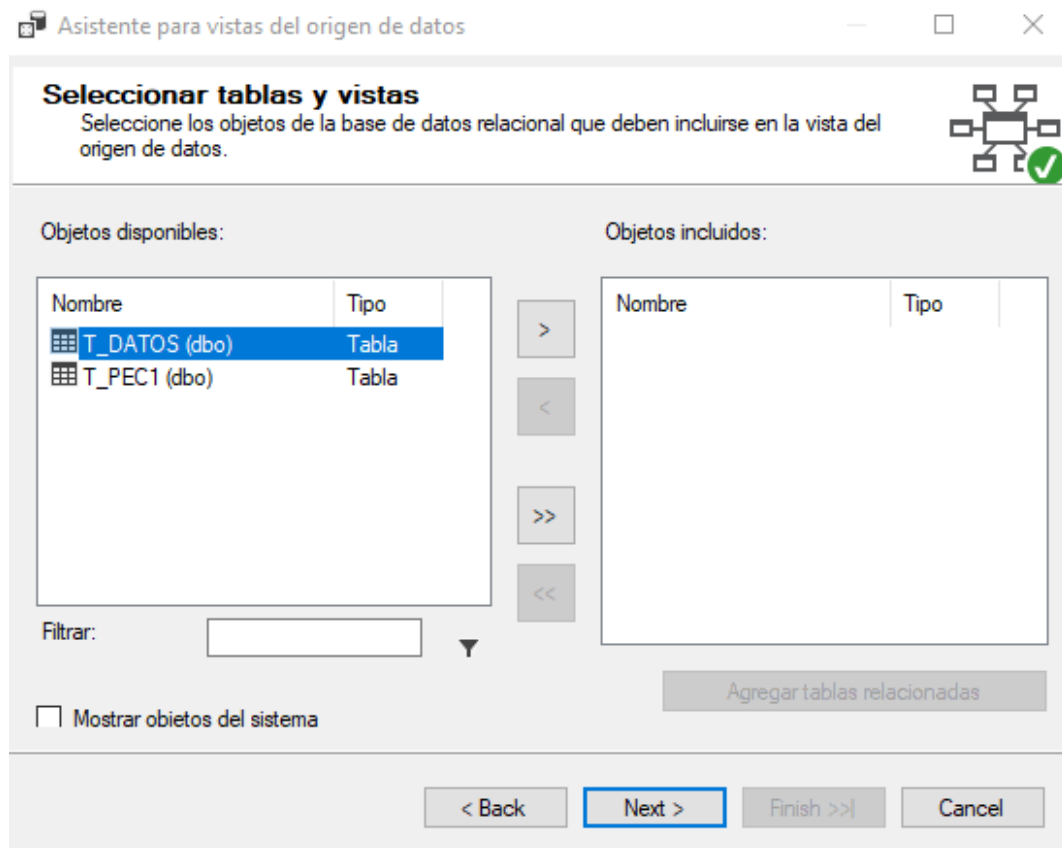


Ilustración 46 - Tablas y Vistas en el asistente de creación de vistas.

Seleccionamos la tabla *T_DATOS*, que es la que hemos tenido que crear para el ejercicio anterior y pulsamos sobre “>”, para así finalmente clicar en el botón *Next*.

Después nos permitirá dar un nombre a la vista:

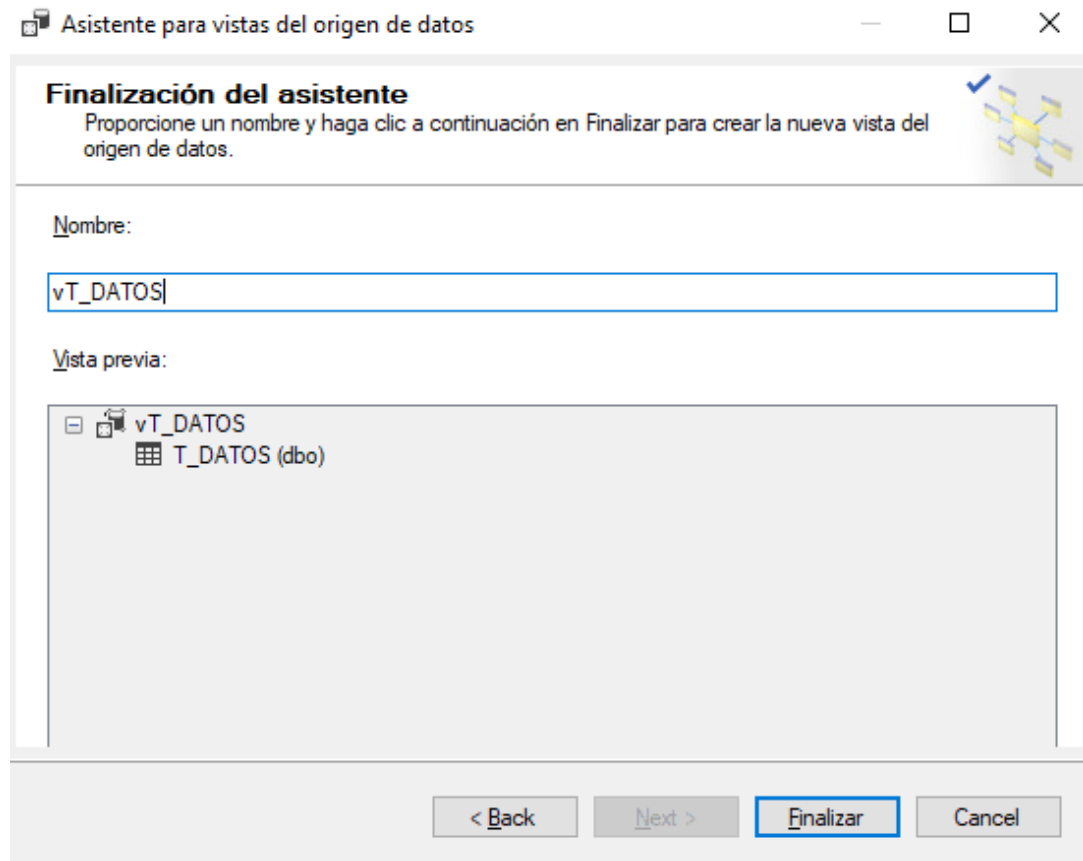


Ilustración 47 - Nombre de la vista.

Para terminar, pulsamos sobre el botón *Finalizar* para así salir del asistente y que se cree la vista, tal y como podemos observar a continuación:

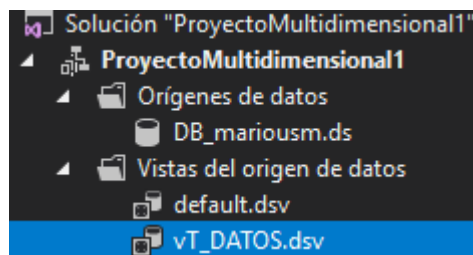


Ilustración 48 - Finalización de la creación de la vista.

2.4.2. Crear un cubo con una única tabla de hecho T_DATOS.

Para crear un cubo nos dirigimos al explorador de soluciones, y en el apartado cubos pulsamos botón derecho y creamos nuevo cubo, de tal forma nos aparecerá la siguiente ventana:

Asistente para cubos

Seleccionar método de creación
Se pueden crear cubos usando tablas existentes, creando un cubo vacío o generando tablas en el origen de datos.

¿Cómo desea crear el cubo?

☒ Usar tablas existentes

☐ Crear un cubo vacío

☐ Generar tablas en el origen de datos

Plantilla:

(Ninguno)

Descripción:

Cree un cubo basado en una o varias tablas de un origen de datos.

< Back Next > Finish >>| Cancel

Ilustración 49 - Creación de cubo.

Indicamos que vamos a usar tablas existentes, ya que se nos indicará hacer uso de la tabla T_DATOS. Una vez hecho eso pulsamos *Next* y seleccionamos la vista creada en el punto anterior:

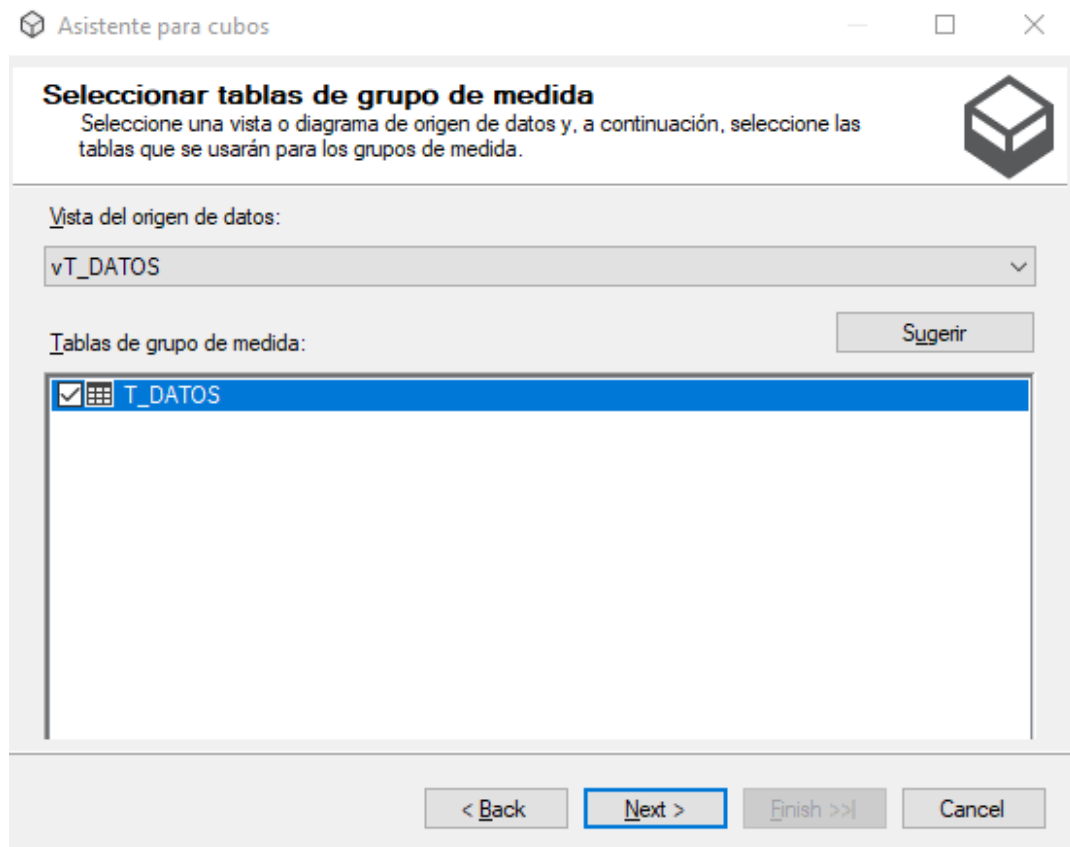


Ilustración 50 - Selección de vista en la creación de cubos.

Luego seleccionamos las medidas que vamos a utilizar para crear el cubo:

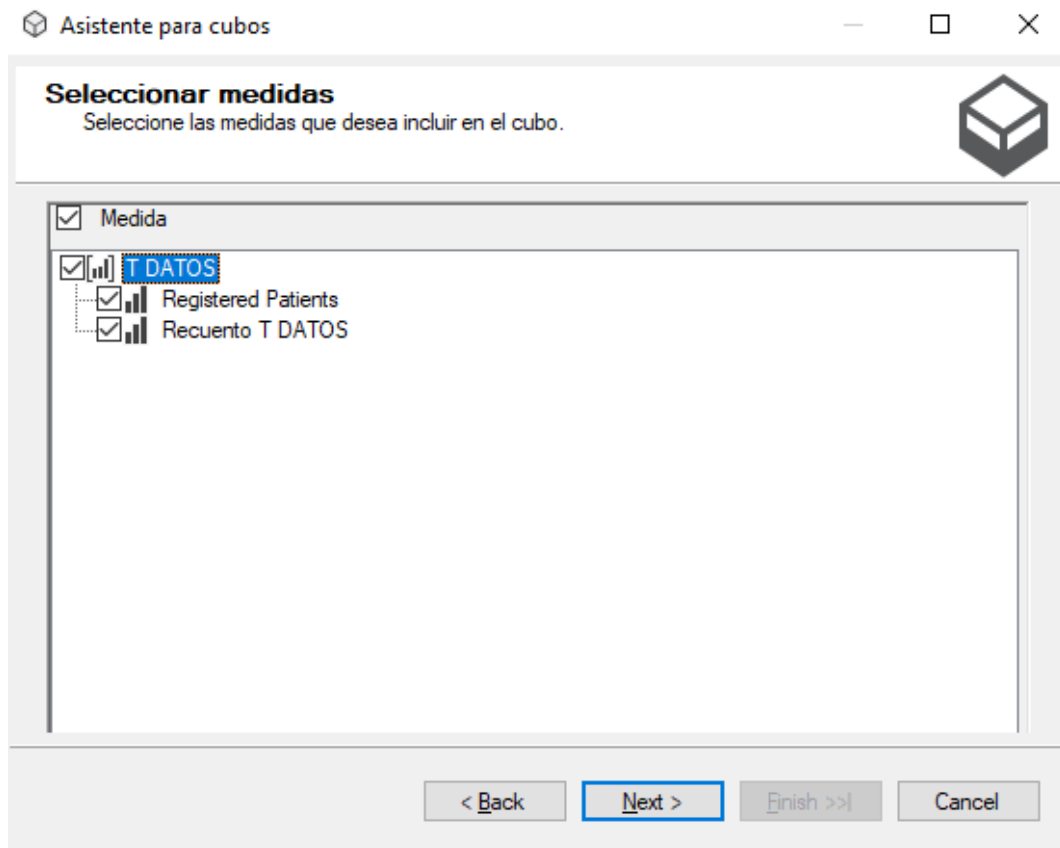


Ilustración 51 - Selección de medidas.

Posteriormente, seleccionamos las dimensiones de nuestro cubo:

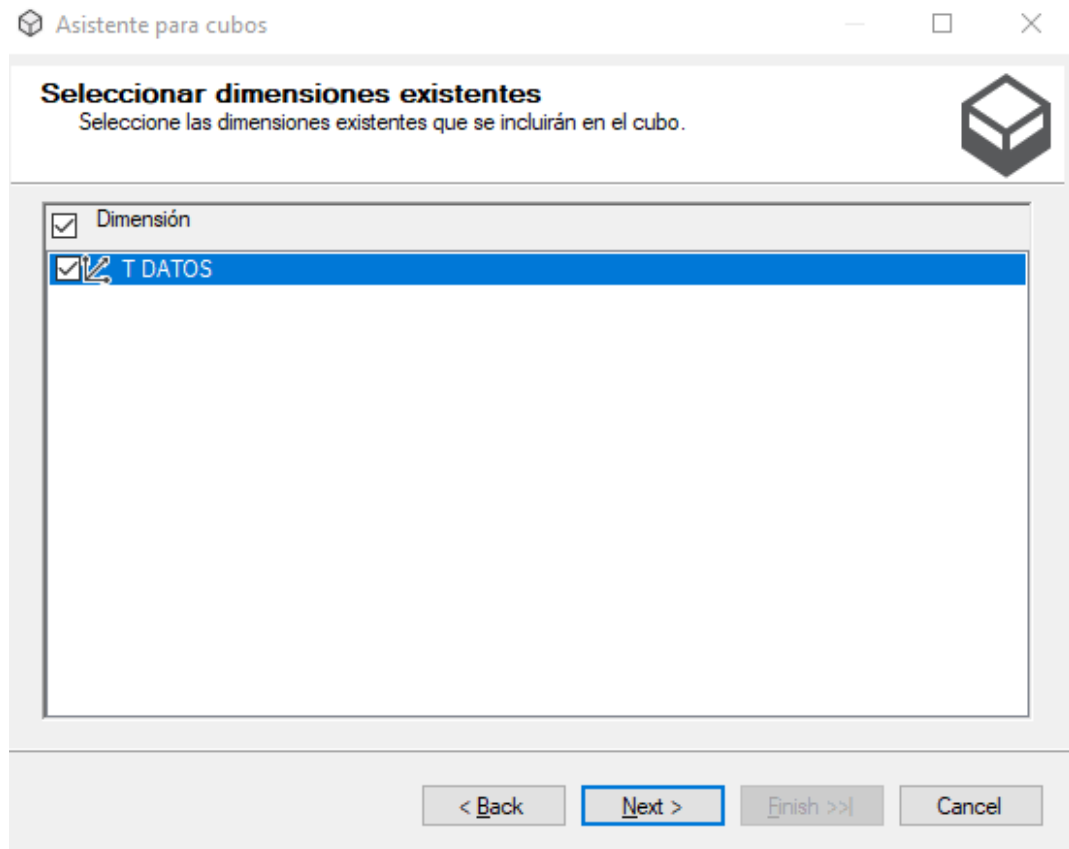


Ilustración 52 - Selección de dimensiones.

También hay que seleccionar las nuevas dimensiones:

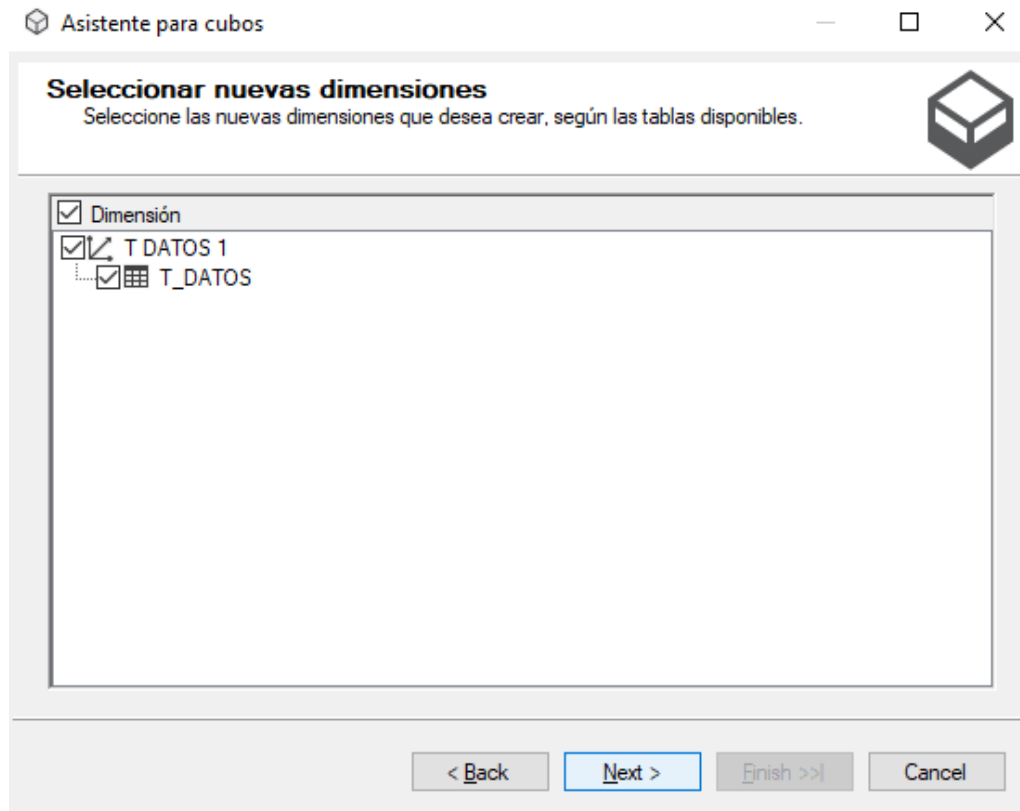


Ilustración 53 - Selección de nuevas dimensiones.

Pulsamos *Next*, y le damos un nombre al cubo. Para terminar con este punto, damos al botón *Finalizar* y ya se nos habrá creado el cubo.

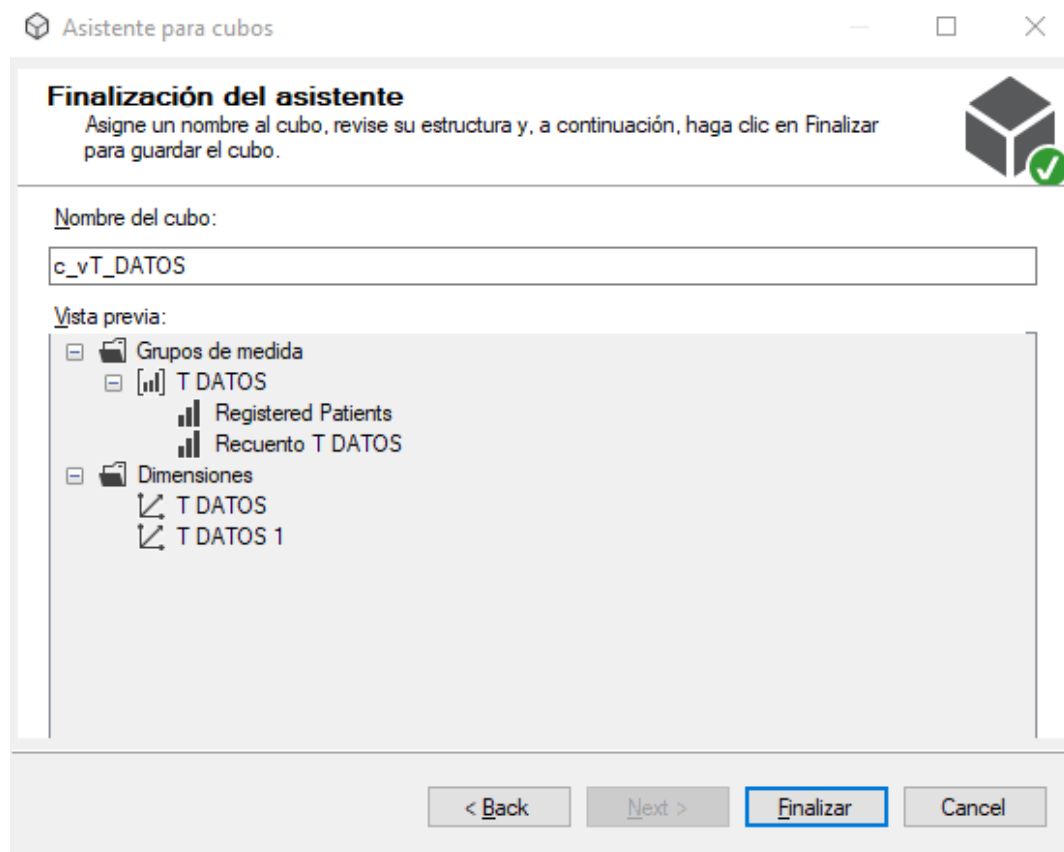


Ilustración 54 - Nombre del cubo.

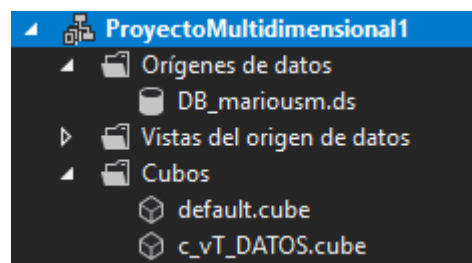


Ilustración 55 - Finalización de la creación del cubo.

2.4.3. Procesar e implementar el cubo en el servidor.

Una vez que ha sido creado el cubo, hacemos click derecho en el explorador de soluciones sobre el cubo y seleccionamos la opción procesar:

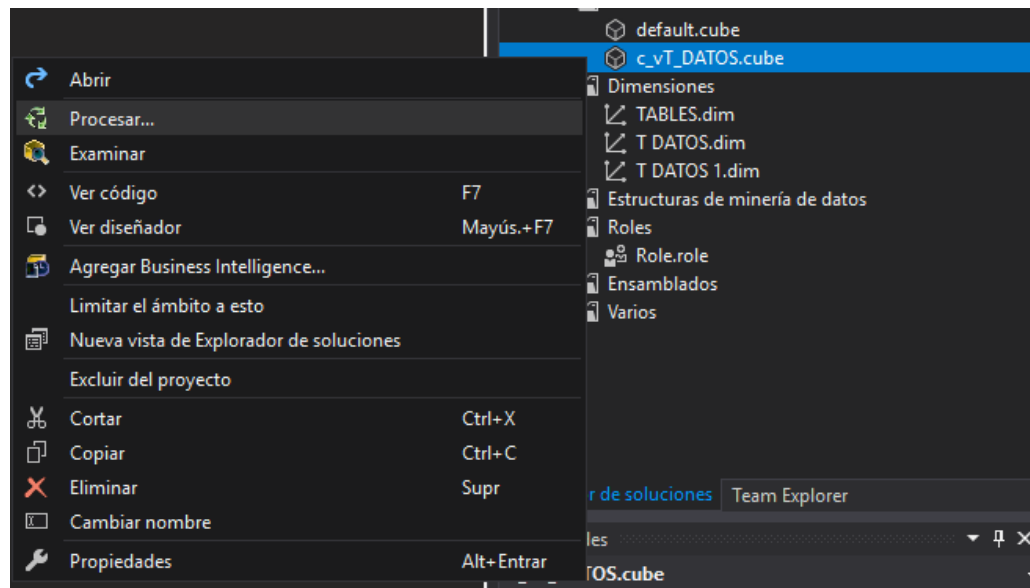


Ilustración 56 - Procesamiento del cubo.

Y si todo ha ido bien, tal y como vemos en la siguiente captura se habrá procesado e implementado el cubo de forma correcta:

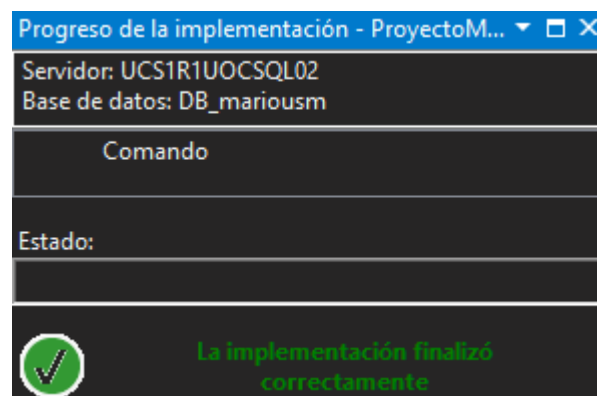


Ilustración 57 - Implementación del cubo.

3. Bibliografía

- [1] G. PowerData, «MDM (Master Data Management). ¿Qué es y cómo debes implementarlo en tu empresa?» <https://www.powerdata.es/mdm> (accedido mar. 19, 2021).

- [2] «Entorno_Virtual_cast.pdf». Accedido: mar. 17, 2021. [En línea]. Disponible en: https://ra.biblioteca.uoc.edu/prestatgeries/22510/Entorno_Virtual_cast.pdf.

- [3] «Citrix Workspace app 2103 for Windows - Citrix Spain», *Citrix.com*. <https://www.citrix.com/es-es/downloads/workspace-app/windows/workspace-app-for-windows-latest.html> (accedido mar. 17, 2021).

- [4] «Crear conexión a bases de datos con Spoon - YouTube». <https://www.youtube.com/watch?v=P-TvOqSb7RQ> (accedido mar. 17, 2021).