SMD. Práctica 4. Herramientas ETL
SSIS (*SQL Server Integration Services*)

José Samos Jiménez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada

Curso 2019-2020

Índice

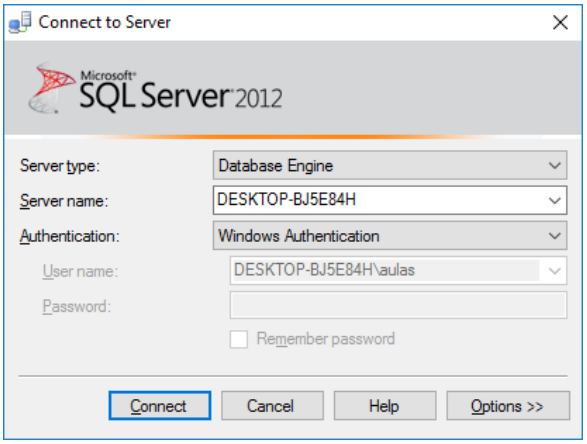

Figura 1: Conectar con el servidor.

Los objetivos de esta actividad son:

- Entender el componente ETL.
- Desarrollar los elementos del componente ETL para un caso sencillo (extracción, transformación y carga de datos).
- Usar una herramienta ETL profesional.
- Aprender el funcionamiento básico de la herramienta ETL *SSIS* (*SQL Server Integration Services*).

1. Introducción

Para trabajar con *SQL Server* usaremos principalmente dos herramientas:

- *SQL Server Management Studio* (*SSMS*) y
- *SQL Server Data Tools* (*SSDT*).

SSMS es una aplicación que permite administrar los servidores de BD que gestionamos, por ejemplo: *Database Engine*, el motor de la BD relacional; *Analysis Services*, el componente multidimensional; *Integration Services*, el componente ETL.

SSDT está basado en *Visual Studio* y permite desarrollar proyectos para los distintos componentes de *SQL Server*, en particular, en esta actividad nos interesará desarrollar paquetes de *SQL Server Integration Services* (*SSIS*), el componente ETL.

2. Operaciones sobre *SQL Server*

2.1. Crear BD

Para crear una BD, se puede utilizar *SSMS*. Al abrir esta herramienta, en primer lugar, nos permite seleccionar el tipo de servidor y el nombre del servidor con el que conectarnos. Para crear una BD, seleccionamos como «Server type» el tipo «Database Engine». Generalmente detecta el servidor que tenemos instalado y su forma de acceso por lo que en los campos «Server name» y «Authentication» podemos dejar las opciones por defecto, y pulsamos sobre el botón «Connect» (figura 1).

Una vez conectados, desplegamos los elementos del servidor pulsando sobre «+» a la izquierda de su nombre. Para crear una BD, en el menú contextual de «Databases», pulsamos sobre [«New Database»] (figura 2). En el campo «Database name» introducimos el nombre de la BD y el resto de datos se configuran en función del nombre; podemos dejar las opciones por defecto en el resto de campos y pulsar sobre el botón «OK» para llevar a cabo la operación. Al desplegar «Databases» (pulsando sobre «+») aparecerá la nueva BD.

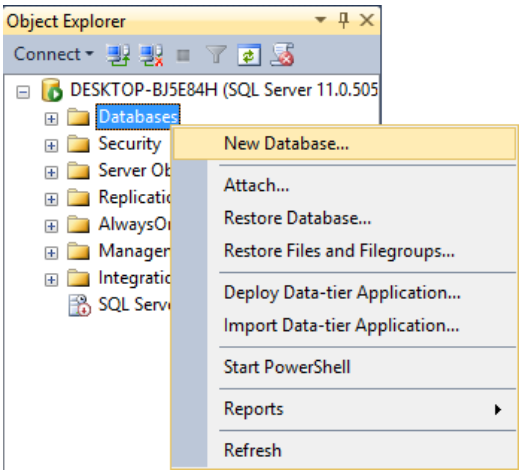

Figura 2: Crear una BD nueva.

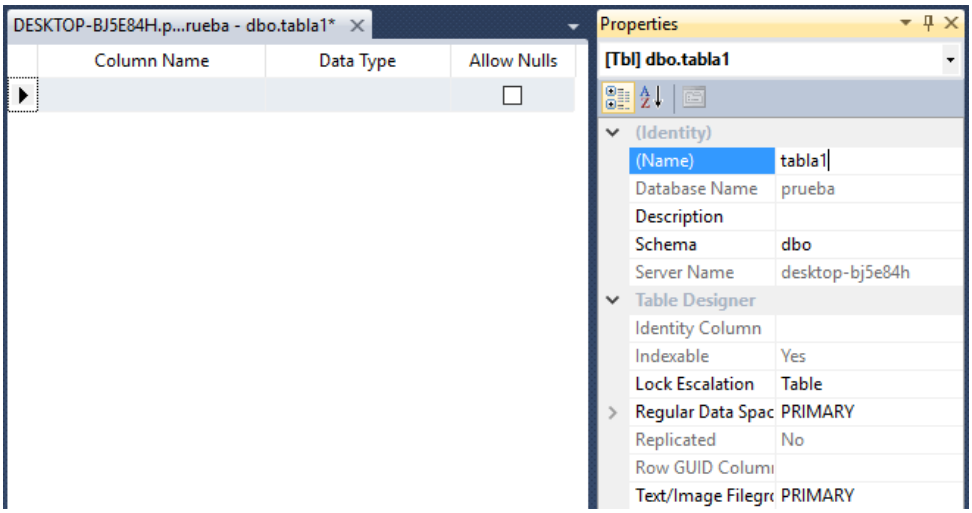

Figura 3: Crear una tabla desde *SSMS*.

2.2. Crear una tabla y definir su estructura

2.2.1. Desde *SSMS*

Desde *SSMS*, para crear una tabla en una BD, desplegamos los elementos de la BD (pulsando sobre «+» a la izquierda de su nombre) y, en el menú contextual asociado al elemento «Tables», pulsamos sobre [«New Table»]. En la ventana que se abre, introducimos el nombre de las columnas y seleccionamos su tipo. En el menú contextual de cada columna definida podemos definir varias características, en particular la llave primaria de la tabla. Si es una llave compuesta, podemos seleccionar las columnas que la componen y definirla en el menú contextual asociado.

Para asignar el nombre a la tabla, podemos usar el campo «Name» de la sección «Identity» en la ventana «Properties», situada a la derecha de la pantalla (figura 3) o bien, al salvar la definición (pulsando sobre los iconos «Save» o «Save All» de la barra de herramientas) abrirá una ventana donde podemos indicar el nombre.

Una vez creada una tabla, podemos definir o modificar posteriormente su estructura seleccionando la opción «Design» en el menú contextual de la tabla.

2.2.2. Desde *LibreOffice*

Para trabajar con *SQL Server* desde *LibreOffice*, en primer lugar, debemos definir una conexión creando una nueva BD en *LibreOffice* y definiendo que se corresponde con una BD *SQL Server*. Posteriormente, podemos trabajar en *LibreOffice* de manera que las transformaciones de lleven a cabo

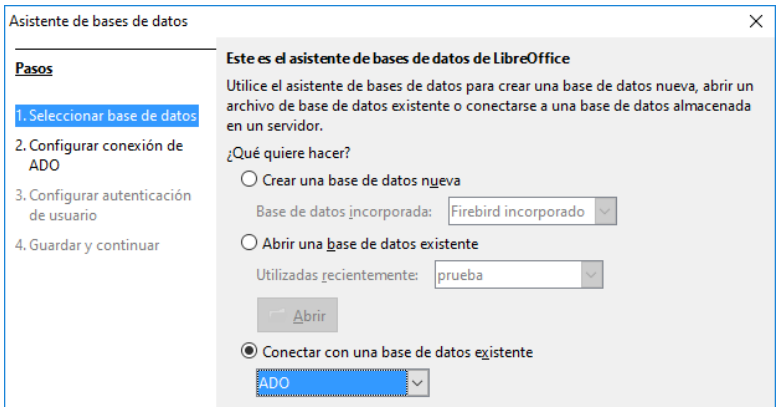

Figura 4: Conectar *LibreOffice* con *SQL Server*.

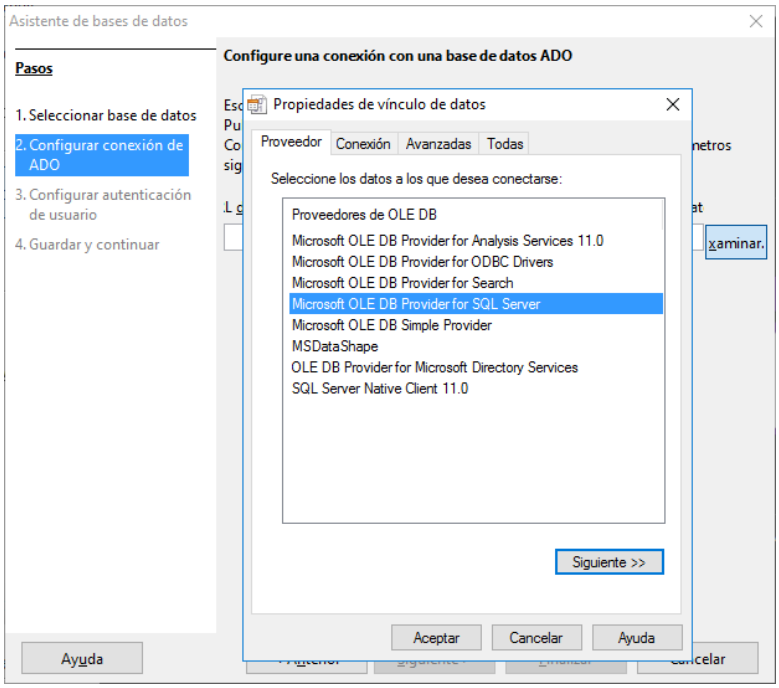

Figura 5: Configuración de la conexión.

en *SQL Server*.

Conectar *LibreOffice* con *SQL Server* En *LibreOffice* pulsamos sobre «Base de datos de Base» y, en la ventana del «Asistente de bases de datos», elegimos la opción «Conectar con una base de datos existente» y seleccionamos «ADO» (figura 4)

A continuación, configuramos la conexión definiendo la cadena de conexión, para ello pulsamos sobre el botón «xaminar» y, en la ventana que se abre, seleccionamos la opción «Microsoft OLE DB Provider for SQL Server» (figura 5) y pulsamos sobre el botón «Siguiente».

En la pestaña «Conexión» a la que accedemos, seleccionamos el nombre del servidor e indicamos para iniciar la sesión «Usar la seguridad integrada de Windows NT» que es la opción que se ha elegido al realizar la instalación del servidor; también seleccionamos el nombre de la BD con la que queremos trabajar (figura 6). Podemos probar la conexión pulsando sobre el botón «Probar conexión».

Por la forma de iniciar la sesión, no debemos indicar el nombre de usuario ni la contraseña. Si no la hemos probado, podemos probar la conexión desde la pantalla siguiente. Si pulsamos sobre el botón «Probar conexión», aparecerá una ventana indicando que se ha realizado correctamente (figura 7).

Por último, debemos seleccionar las opciones «Sí, registrar la base de datos» y «Abrir la base de datos para su edición» (figura 8) para trabajar en la BD *SQL Server* desde *LibreOffice*.

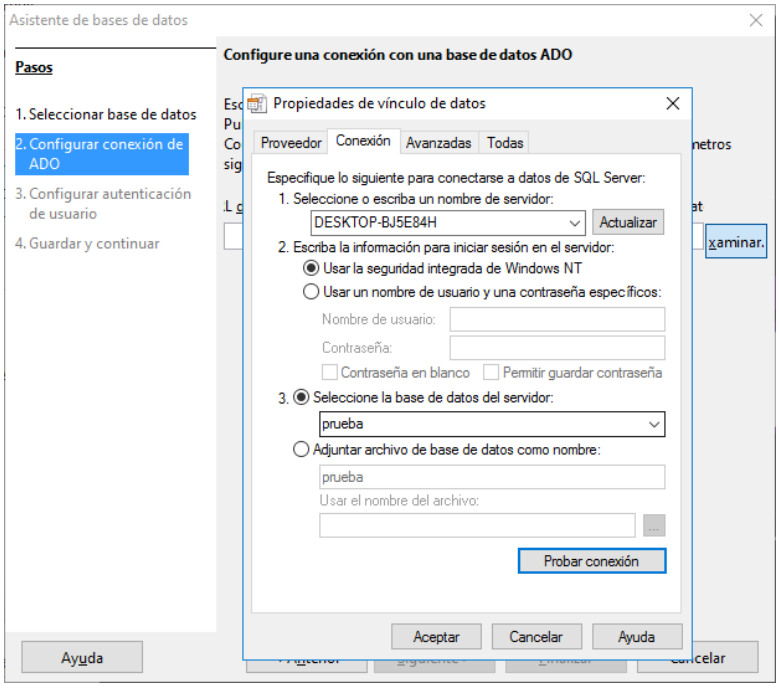

Figura 6: Detalles de la configuración de la conexión.

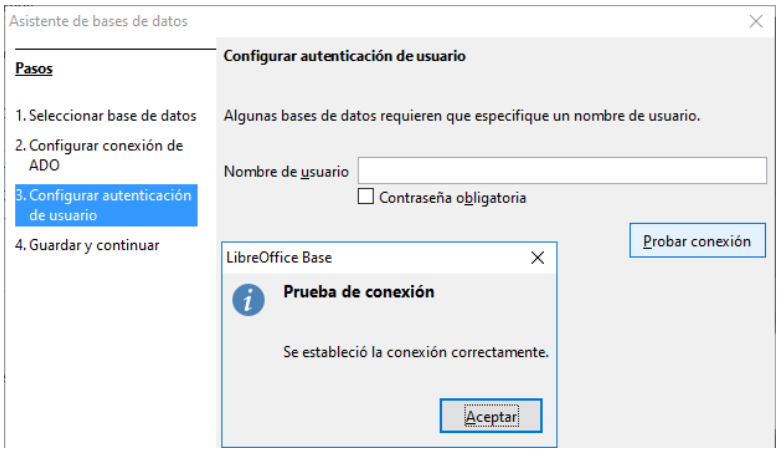

Figura 7: Probar la conexión.

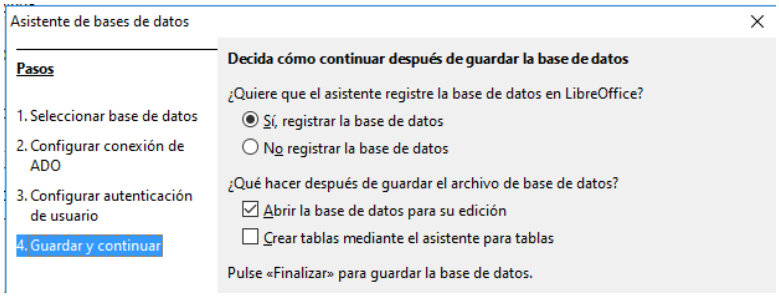

Figura 8: Registrar la BD.

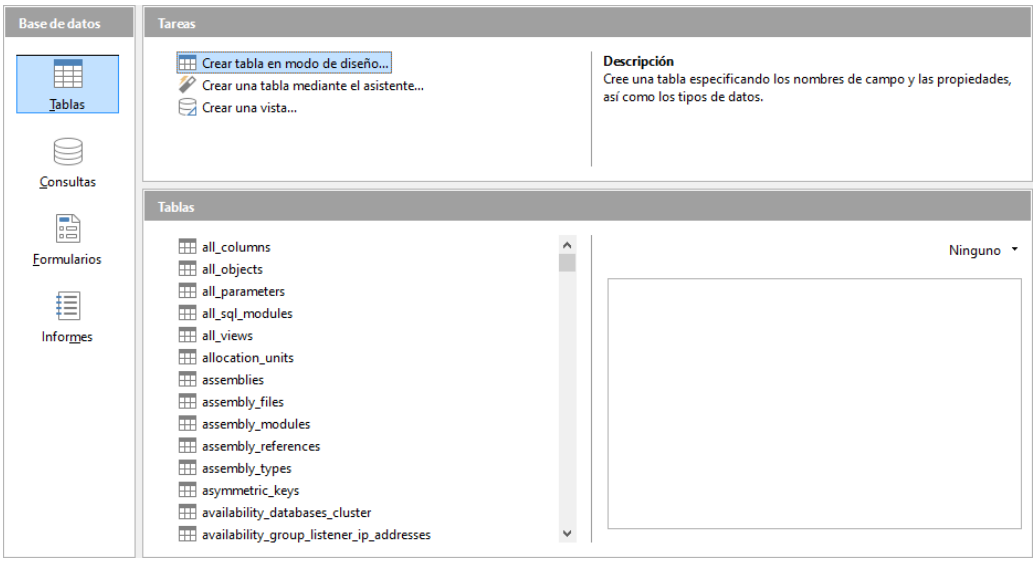

Figura 9: Crear tabla en modo diseño.

Crear una tabla y definir su estructura Una vez hemos conectado la BD actual con *SQL Server*, aparece una larga lista de tablas de sistema que no son visibles desde *SSMS*.

Para definir una nueva tabla, pulsamos sobre la tarea «Crear tabla en modo diseño» (figura 9). Se abre una ventana donde podemos definir la lista de los nombres de campo y sus tipos, así como los índices que nos interesen.

2.2.3. Importar datos en *SSMS*

A partir de una fuente de datos podemos importar su contenido en una BD creando las tablas que contendrán los datos y, si lo indicamos, genera automáticamente un paquete ETL de *SSIS* que realiza las operaciones necesarias.

En el menú contextual de cualquier BD pulsamos sobre la opción [«Tasks», «Import Data»] (figura 10).

En la ventana que se abre (figura 11), en el campo «Data source» definimos el tipo de fuente de datos, en este caso archivo *Excel*; la ubicación del archivo (seleccionamos el archivo *Excel* generado mediante *Power Query*, en mi caso el archivo es `granada-ETL-jsamos.xlsx`) y la versión de este; adicionalmente nos aseguramos que esté seleccionada la opción «First row has column names».

A continuación (figura 12), nos aseguramos que esté seleccionada la opción correspondiente a *SQL Server* («SQL Server Native Client 11.0») y el nombre del servidor, indicamos la forma de identificación para nuestra instalación («Use Windows Authentication»), y seleccionamos la BD en la que queremos importar los datos.

Podemos indicar mediante consultas una selección de los datos a importar, en este caso, nos interesa importar todos los datos disponibles, en la siguiente pantalla seleccionamos la opción «Copy data from one or more tables or views» (figura 13).

A continuación, podemos definir la correspondencia entre las hojas y las tablas de la BD (figura 14). Tanto para los nombres de las tablas como de los campos, usaremos minúsculas sin tilde y, en lugar del criterio *Camel Case* (`CamelCase`), usaremos el carácter «_» como separador (`camel_case`).

Podemos definir los nombres de los campos seleccionando correspondencia entre la hoja y la tabla, y pulsando sobre el botón «Edit Mappings» (figura 14).

Definimos los nuevos nombres de los campos según el criterio que vamos a usar (figura 15). En principio, se pueden cambiar los tipos de los campos, en las pruebas que he hecho siempre me ha dado error, por lo que recomiendo cambiar solo los nombres y la opción de si pueden tener el valor nulo o no («Nulable»), los tipos de los campos se pueden cambiar posteriormente editando las tablas generadas, por ejemplo, desde *SSMS* («Design»). Adicionalmente, podemos seleccionar la opción «Drop and re-create destination table» que genera las sentencias SQL necesarias para eliminar la tabla antes de

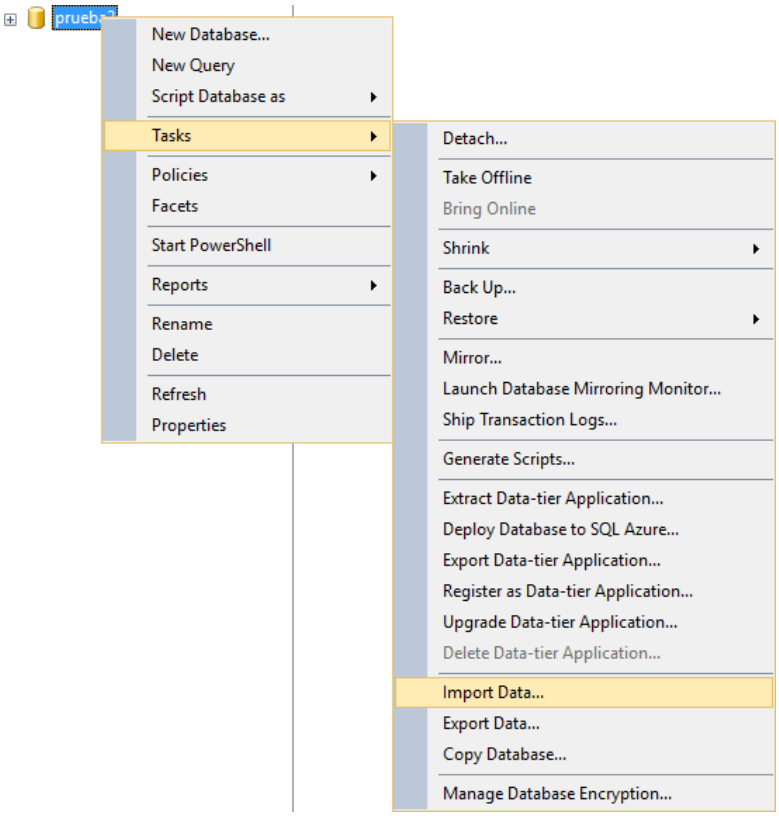

Figura 10: Importar datos a una BD.

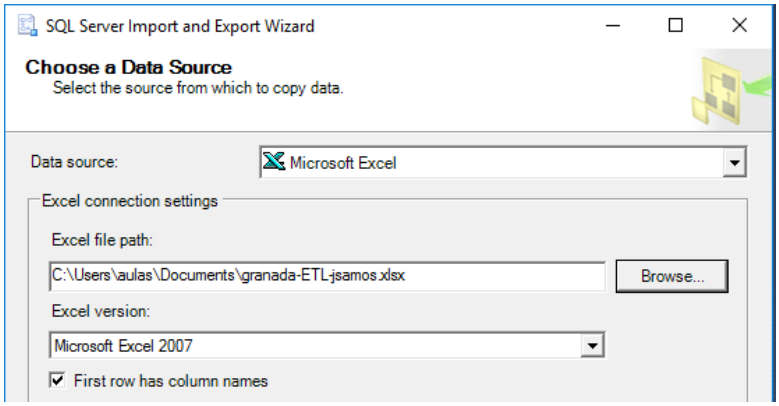

Figura 11: Definir la fuente de datos.

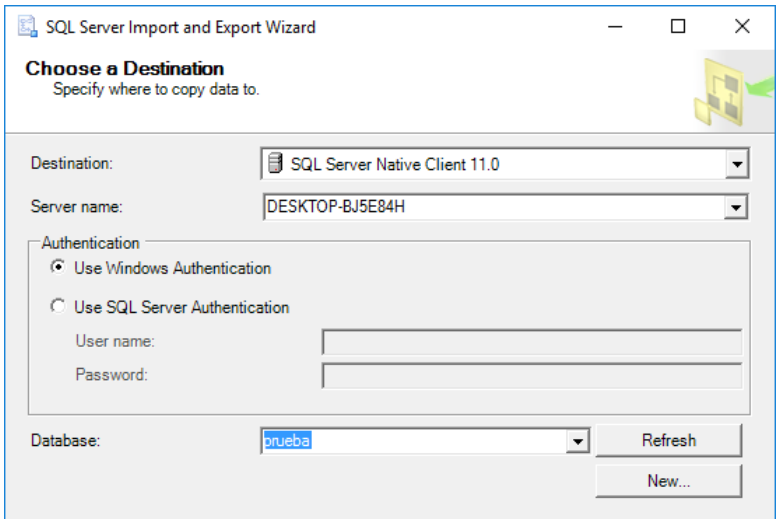

Figura 12: Definir el destino de los datos.

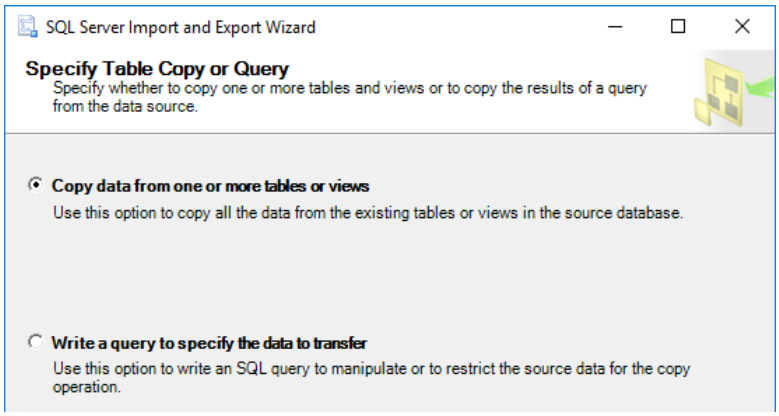

Figura 13: Copiar todos los datos disponibles en las hojas.

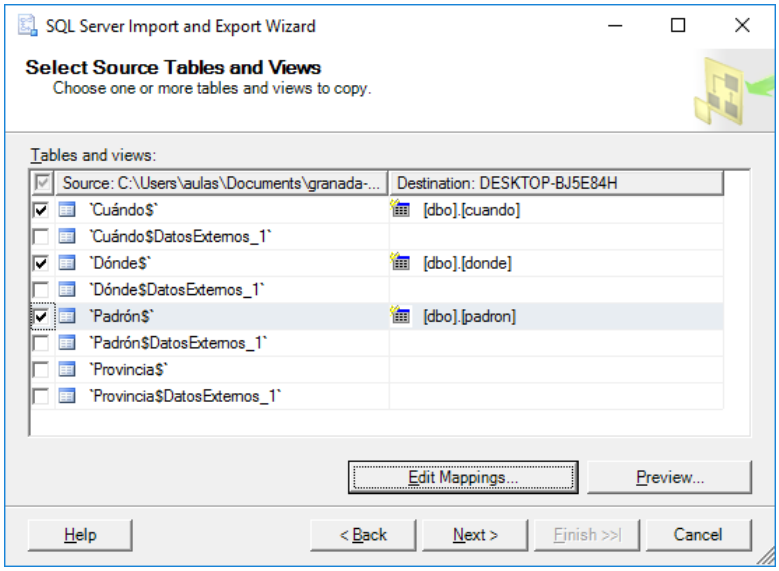

Figura 14: Correspondencia entre las hojas y las tablas.

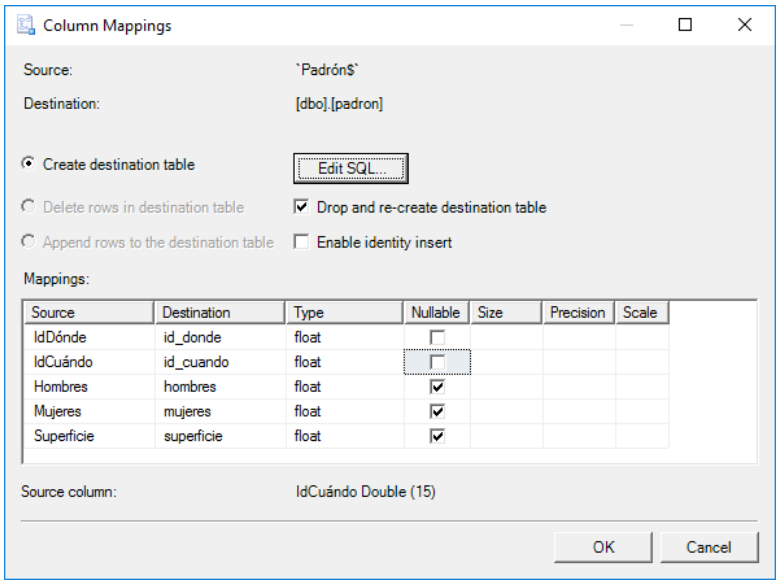

Figura 15: Correspondencia entre columnas y campos.

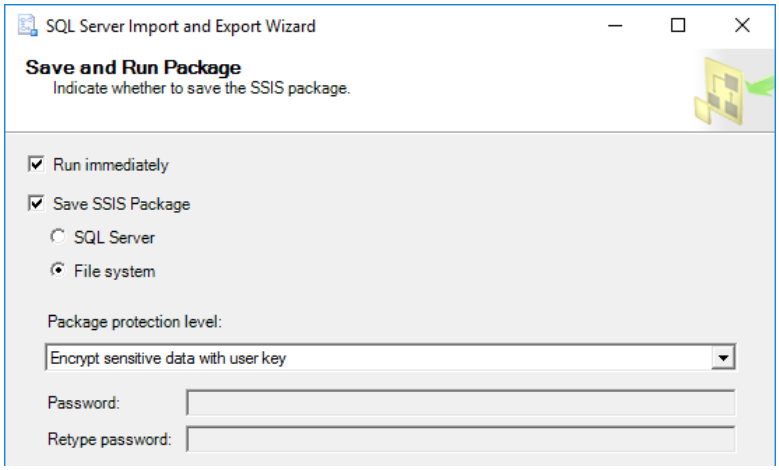

Figura 16: Ejecutar y guardar la transformación.

crearla; esta opción resulta de utilidad si queremos realizar varias pruebas de ejecución.

A continuación, podemos indicar que se ejecute inmediatamente el proceso de importación y, adicionalmente, que se genere una paquete *SSIS* con los elementos necesarios para realizarla. El paquete *SSIS*, se puede guardar en el servidor o bien como un archivo independiente, en este caso vamos a seleccionar que se almacene como un archivo seleccionando la opción «File system», también permite encriptar las claves que se usen, en este caso, podemos dejar la opción por defecto (figura 16). En la pantalla siguiente, podemos darle nombre al archivo donde se guardará el paquete e indicar en qué carpeta se almacenará (figura 17).

Por último, nos informa de las operaciones que va a realizar (figura 18) y, al ejecutarlas, el resultado de estas. Por ejemplo, si para las tablas hemos seleccionado la opción «Drop and re-create destination table» y estas no existen la primera vez que se ejecute, nos avisará del error pero no tendrá ninguna repercusión porque seguirá con las operaciones de creación de tablas.

1. Crea una BD *SQL Server* cuyo nombre sea `prueba` y tu nombre de usuario de correo UGR (en mi caso se llamará `prueba_jsamos`). Importa las hojas `Cuándo`, `Dónde` y `Padrón` del archivo obtenido con *Power Query* (en mi caso el archivo es `granada-ETL-jsamos.xlsx`). En los pasos de importación, **selecciona la opción «Save SSIS Package» y «File system»** para generar un paquete de *SSIS* en forma de archivo.

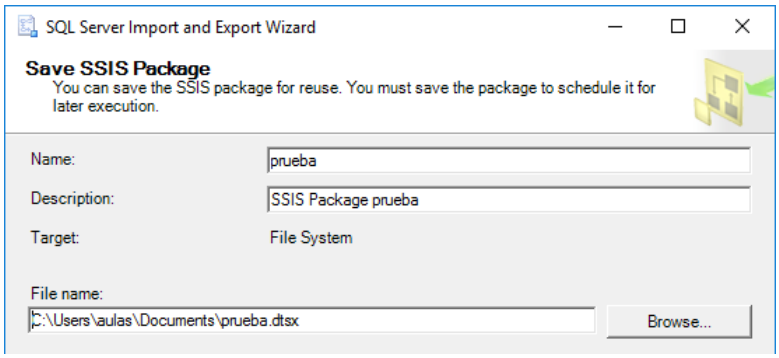

Figura 17: Guardar la transformación como archivo.

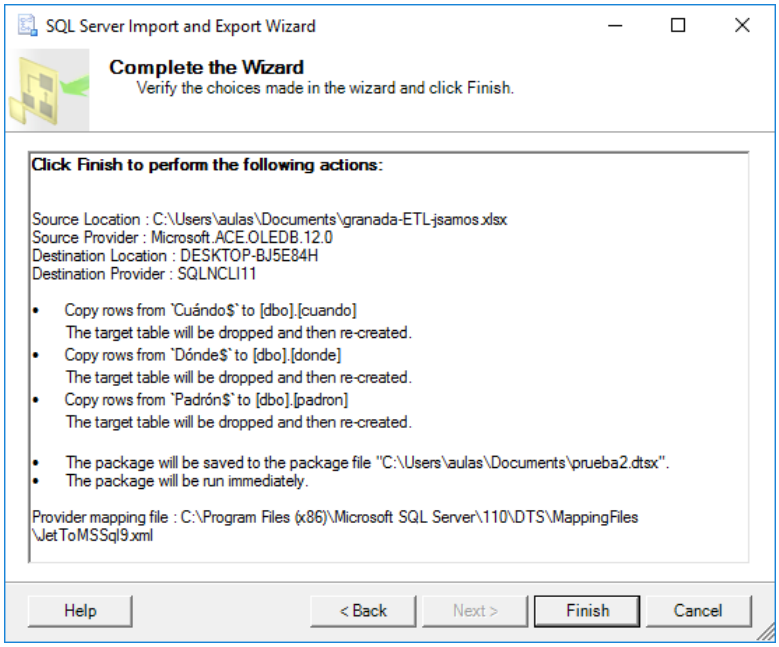

Figura 18: Operaciones a realizar para importar los datos.

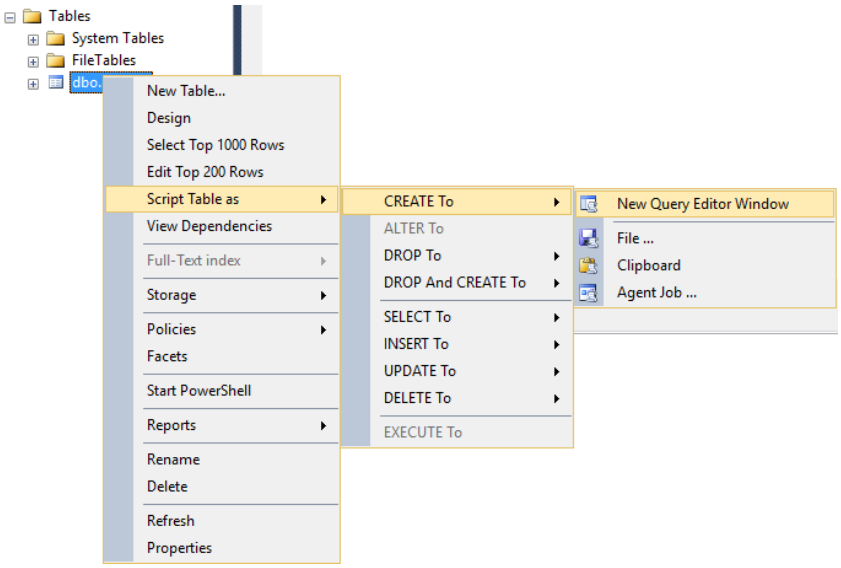

Figura 19: Generar sentencias SQL desde *SSMS*.

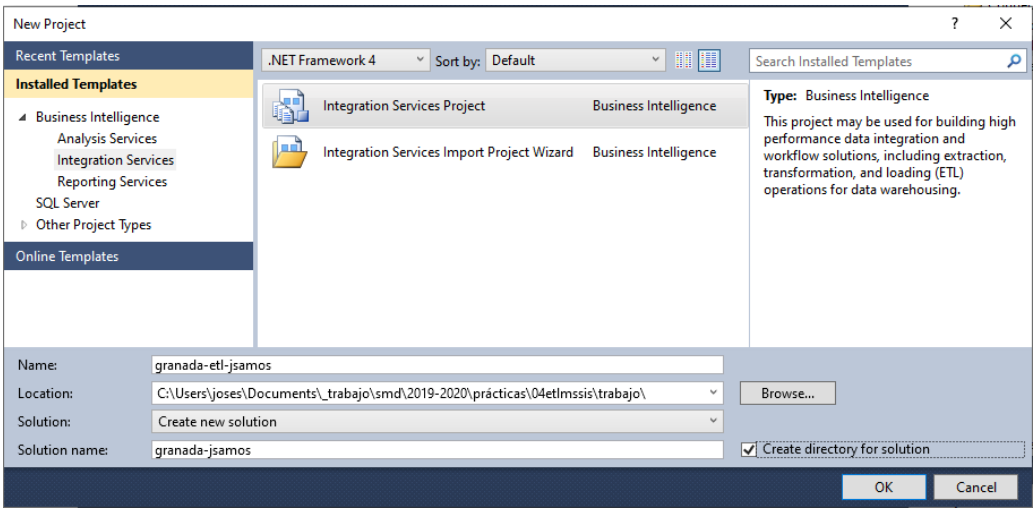

Figura 20: Crear un proyecto.

2.3. Generar sentencias SQL desde *SSMS*

Una vez tenemos las tablas definidas en la BD, podemos generar las sentencias SQL de diversas operaciones desde *SSMS*: en el menú contextual de cada tabla, pulsamos sobre la opción de la operación que necesitemos, por ejemplo [«Script Table as», «CREATE To», «New Query Editor Window»] (figura 19) para generar la sentencia de creación de tabla en una nueva ventana del entorno de *SSMS*.

3. Operaciones sobre *SSDT*

En esta sección, usaremos el paquete generado automáticamente en la operación de importar datos para introducir el uso de *SSDT* para trabajar con proyectos de *SSIS*. Se pueden encontrar más detalles de *SSIS* y *SSDT* en [SDKK12].

3.1. Paquetes, proyectos y soluciones

Para poder trabajar con un paquete *SSIS*, en primer lugar, debemos crear un proyecto en *SSDT* que contendrá los paquetes y otros elementos asociados a *SSIS*. Pulsamos sobre «New Project» en la ventana «Start Page» o bien sobre [«File», «New», «Project»], a continuación, pulsamos directamente sobre «Integration Services Project» o bien, primero pulsamos sobre «Integration Services» (en la

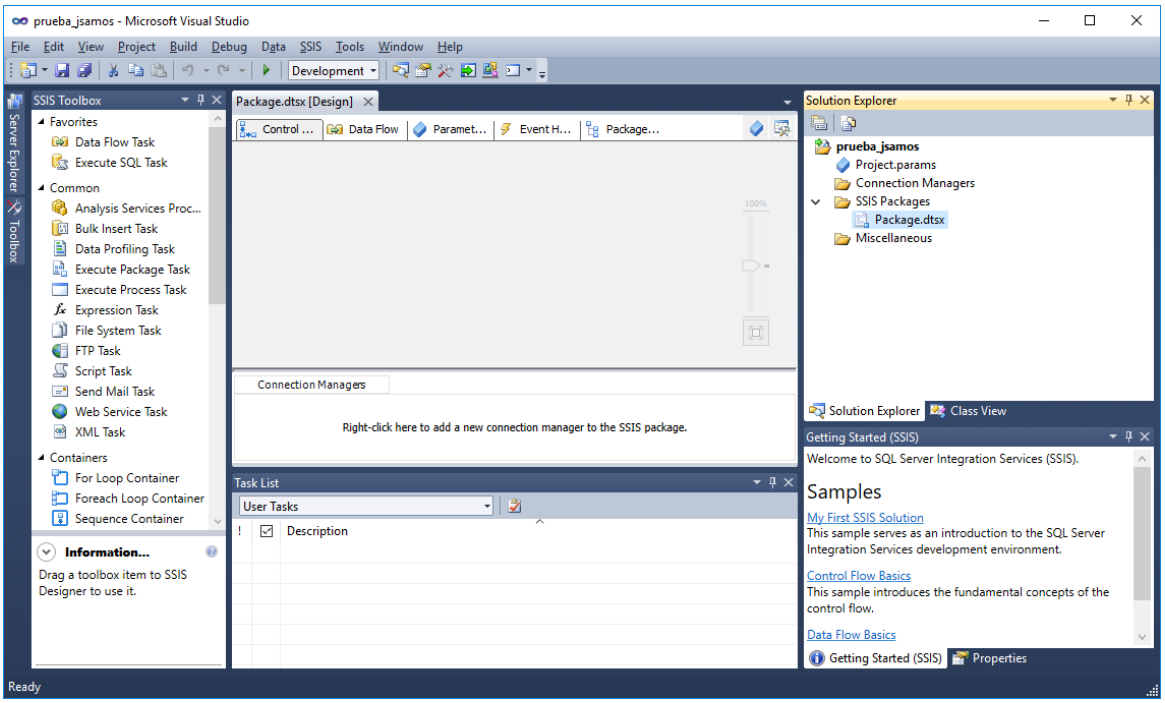

Figura 21: Entorno de trabajo de un nuevo proyecto.

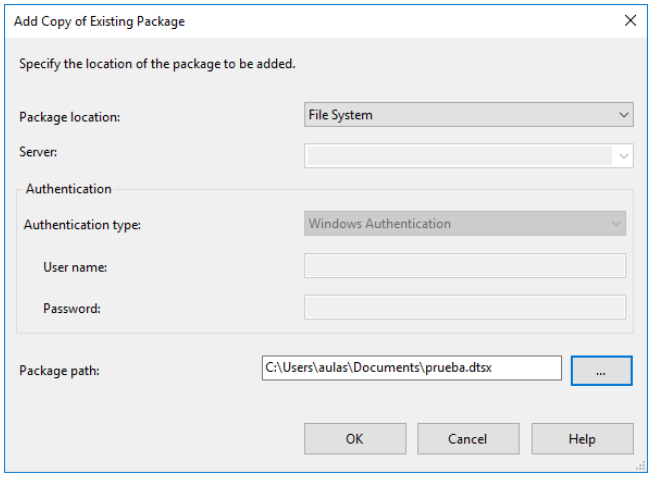

Figura 22: Selección del paquete almacenado como archivo.

columna de la izquierda) y después sobre «Integration Services Project». En la ventana del proyecto le asignamos un nombre y ubicación. Podemos tener varios proyectos asociados al mismo tema de trabajo, pueden ser proyectos de *SSIS* o de otros componentes de *SQL Server*; los proyectos se pueden agrupar bajo el concepto de *solución*, por eso, al crear un proyecto podemos seleccionar el nombre de una solución existente o bien indicar una nueva en el campo «Solution» para que se cree a la vez que el proyecto (figura 20). Cuando creamos nuestro primer proyecto, como todavía no tenemos ninguna solución creada, no ofrece la posibilidad de seleccionar una solución previa en la que ubicar el proyecto, tenemos que crear una nueva.

Al crear un proyecto, automáticamente se crea un paquete llamado `Package.dtsx`, podemos verlo en el entorno de trabajo de *SSDT* (figura 21). Para explicar mejor las partes del entorno, vamos a comenzar importando el paquete generado en el punto 1.

Para añadir un paquete al proyecto, en el menú contextual de de la carpeta «SSIS Packages» de la ventana «Solution Explorer» (parte derecha de la figura 21) seleccionamos la opción «Add Existing Package» y, en la ventana que se abre, seleccionamos el en campo «Package location» el valor «File System» y en el campo «Package path» el archivo que contiene el paquete generado (figura 22). Una

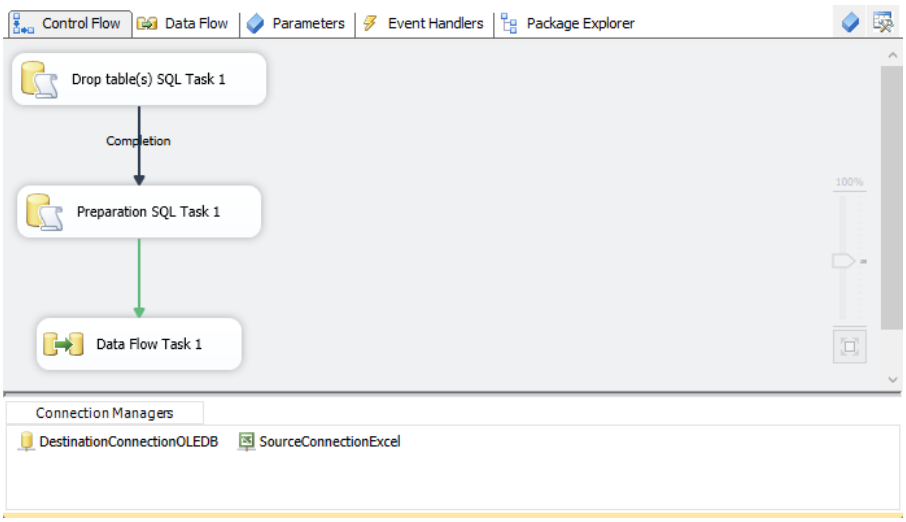

Figura 23: Flujo de control de un paquete.

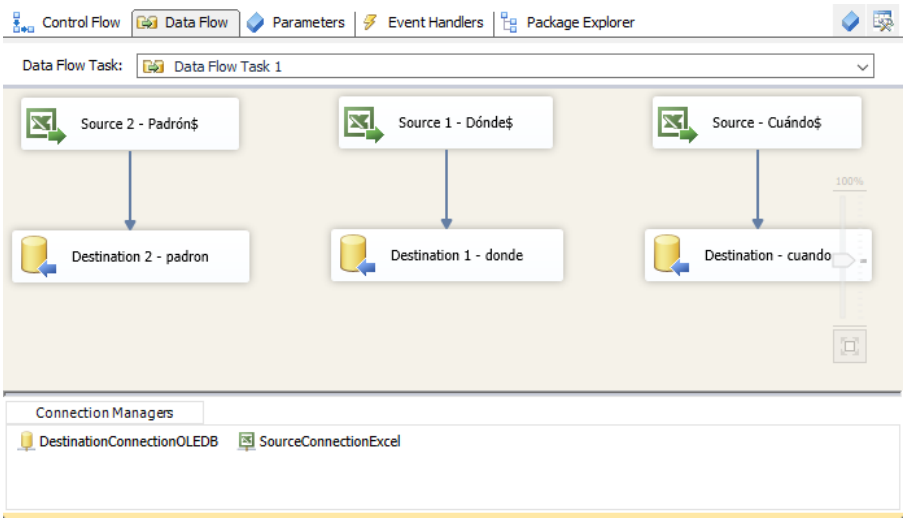

Figura 24: Flujo de datos de un paquete.

vez añadido el paquete bajo la carpeta «SSIS Packages», pulsamos «doble-clic» sobre su nombre para abrirlo.

2. Crea un proyecto *SSIS* en *SSDT* y una solución cuyos nombres sean `prueba` y tu nombre de usuario de correo UGR (en mi caso se llamarán `prueba_jsamos`), importa el paquete generado en el punto 1 y ábrelo.

3.2. Entorno de desarrollo de un proyecto

La ventana del proyecto tiene tres zonas en la parte superior:

- «Solution Explorer»: en la parte derecha, muestra los componentes del proyecto, uno de ellos son los paquetes.
- «Design»: en el centro, muestra los elementos del componente del proyecto con el que estamos trabajando, los que tenemos abiertos. Tiene varias pestañas que nos permiten diseñar los apartados de cada componente. En particular, para un paquete destacamos las pestañas:
  - «Control Flow»: permite diseñar el flujo de control del paquete (figura 23). Cada paquete solo puede tener un componente de flujo de control.

- «Data Flow»: permite diseñar los flujos de datos del paquete (figura 24). En un paquete puede haber varios componentes de flujo de datos, cada uno de ellos se corresponderá con un elemento «Data Flow Task» de la pestaña «Control Flow».

En la parte inferior se encuentra fija la pestaña «Connection Managers», donde se encuentran las conexiones a fuentes o destinos de los datos utilizados en el paquete.

- «SSIS Toolbox»: en la parte izquierda, se configura en función de la pestaña de la parte «Design» que tenemos seleccionada, muestra los elementos que podemos utilizar en su desarrollo. Está organizada en secciones dinámicas: podemos mover un elemento de una sección a otra desde su menú contextual, en particular, podemos ubicar en el apartado «Favorites» los de uso más frecuente.

Adicionalmente, en la parte inferior derecha hay una zona con dos pestañas en la parte inferior. Si pulsamos sobre la pestaña «Properties» se mostrará la ventana con las propiedades de cualquier elemento del proyecto sobre el que pulsemos.

3.3. Gestores de conexión

Se pueden definir «Connection Managers» particulares de un paquete o de un proyecto que pueden compartir todos los paquetes del mismo. Las conexiones actuales son particulares del paquete (figuras 23 y 24); se pueden transformar en conexiones de proyecto desde el menú contextual, «Convert to Project Connection» (pasaría a incluirse también en la carpeta «Connection Managers» de «Solution Explorer».

Se puede definir una nueva conexión de un paquete desde el menú contextual de esa parte de la ventana (pulsando sobre cualquier zona libre); también se puede definir a nivel de proyecto desde el menú contextual de la carpeta «Connection Managers».

3.4. Flujo de control

Asociado a cada paquete solo puede haber un componente de flujo de control («Control flow», figura 23). Los distintos elementos que se pueden usar para definirlo se encuentran en la ventana «SSIS Toolbox». En el ejemplo, está compuesto por dos elementos «Execute SQL Task» y un elemento «Data Flow Task». Cada elemento «Data Flow Task» tiene una pestaña particular asociada en la ventana «Design»: la ventana solo muestra una pestaña «Data Flow Task» pero esta muestra la definición del elemento correspondiente seleccionado en la pestaña «Control flow».

Los elementos se relacionan mediante conexiones que se definen «pulsando-arrastrando-soltando» la flecha que parte de cada elemento al pulsar sobre él. Una vez definida una conexión entre dos elementos, se puede definir su tipo en el menú contextual de la misma; admite tres tipos autodescriptivos: «Success», «Failure» y «Completion».

Para cada elemento que añadamos, podemos definir diferentes características. Por ejemplo, en el elemento «Execute SQL Task» de la figura 25 hay que definir la conexión que se usa y la sentencia SQL a ejecutar. La sentencia en concreto se puede ver pulsando sobre los «...» que aparecen a la derecha del valor del campo «SQLStatement». Para formular algunos tipos de consulta se puede utilizar la herramienta de ayuda que se abre pulsando sobre el botón «Build Query», sin embargo, para la consulta del tipo `CREATE TABLE` no se puede usar esta herramienta.

3.5. Flujos de datos

En los elementos de flujo de datos se definen las correspondencias entre las entradas y las salidas (figura 24). Las conexiones entre los elementos definen transferencia de datos o bien el tratamiento de errores.

Los dos elementos de uso más frecuente son «Source Assistant» y «Destination Assistant». Para cada uno de ellos podemos indicar la conexión asociada y los campos que se obtienen o se definen. En el elemento destino se pueden definir las correspondencias entre los campos fuente y destino (figura 26), se pueden definir «pulsando-arrastrando-soltando» un campo sobre otro.

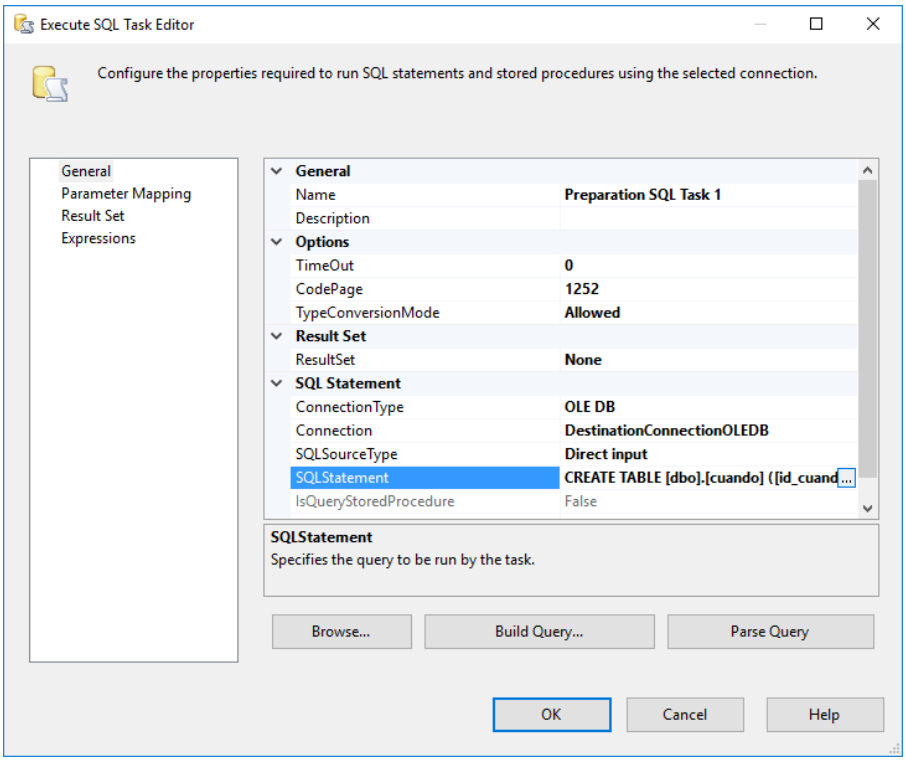

Figura 25: Configuración de la operación de ejecutar SQL.

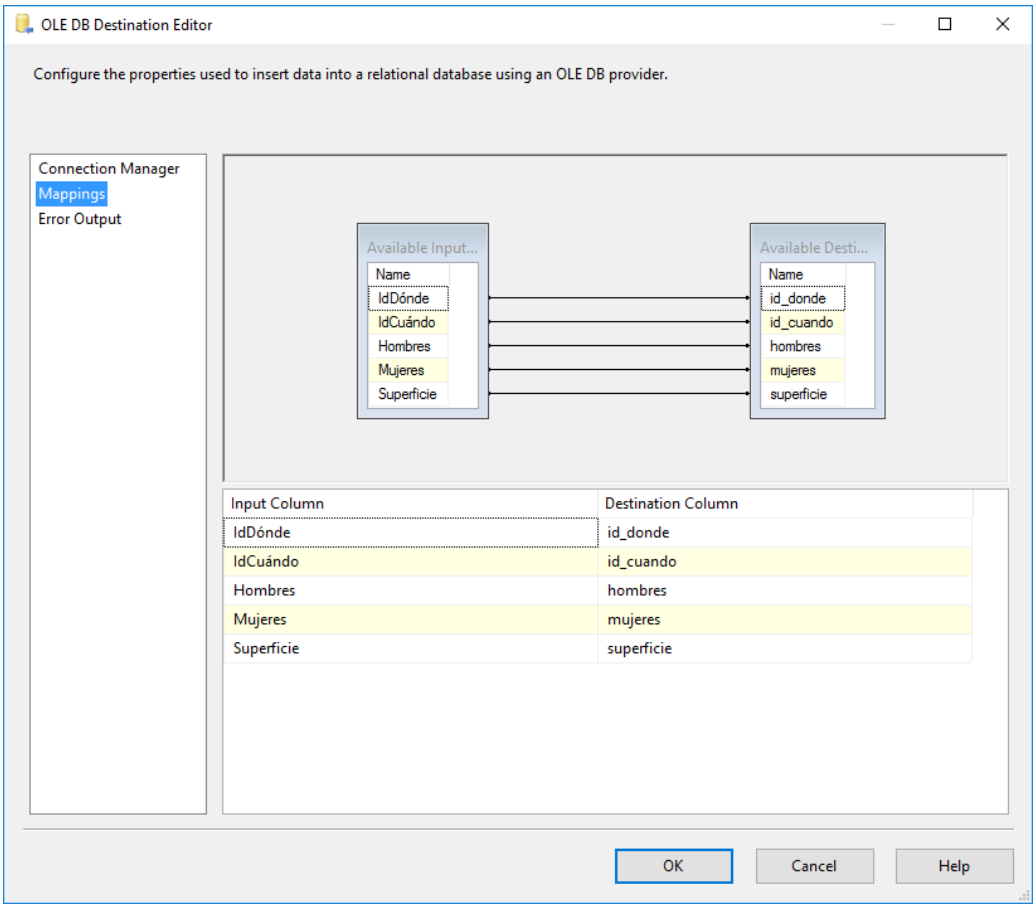

Figura 26: Definición de correspondencias entre fuente y destino.

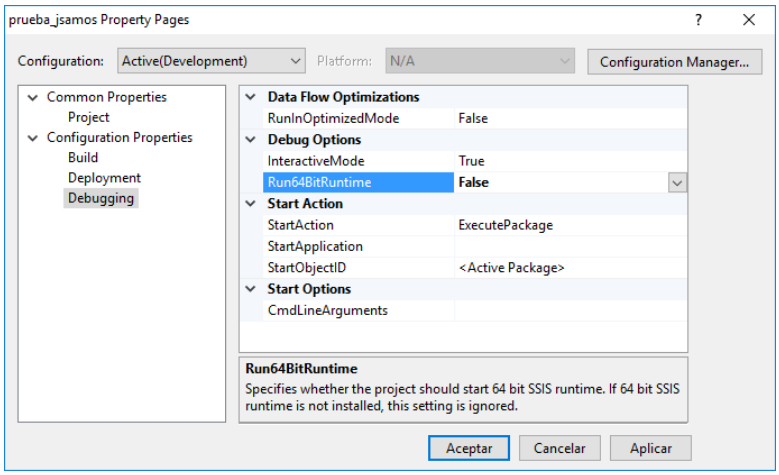

Figura 27: Cambiar las propiedades del proyecto.

3.6. Ejecución de un paquete

Podemos ejecutar un paquete desde el menú contextual del mismo, seleccionando la opción «Execute Package», también teniendo seleccionado el paquete y pulsando sobre el icono de la barra de herramientas «Start Debugging (F5)».

Para volver al modo de diseño, una vez acabada la ejecución, debemos pulsar sobre el icono de la barra de herramientas «Stop Debugging (Shift+F5)» o pulsando sobre [«Debug», «Stop Debugging»].

Si todo ha ido bien, lo indica con un tick sobre fondo verde asociado a cada elemento. Si se ha producido algún error lo indica con una equis sobre fondo rojo. Podemos ver los detalles de la ejecución en la pestaña «Progress» de la ventana «Design».

Si ejecutamos el paquete actual podemos observar que se producen errores en la obtención de los datos de las fuentes: en la pestaña «Progress» aparece el error «Error: The requested OLE DB provider Microsoft.ACE.OLEDB.12.0 is not registered. If the 64-bit driver is not installed, run the package in 32-bit mode.»

Resulta que la versión de *Office* instalada es de 32 bits y el resto de herramientas son de 64 bits. Para solucionar esto, tenemos que acceder a las propiedades del proyecto y cambiar el valor de la variable «Run64BitRuntime» a «False»[1].

Accedemos a las propiedades del proyecto pulsando sobre [«Project», «Properties»] y, en el apartado «Debugging» cambiamos el valor de la variable adecuadamente. Una vez realizados los cambios, si volvemos a ejecutar el paquete, todos los elementos se ejecutan con éxito.

3.7. Despliegue de un proyecto

Para poner el proyecto en producción, seleccionamos la opción «Deploy» del menú contextual del proyecto. Se abre una ventana con los pasos que tenemos que realizar. En primer lugar tenemos que indicar el servidor de *SSIS* y un catálogo. El catálogo deberíamos crearlo desde *SSMS*, conectando con el servicio «Database Engine», en el apartado «Integration Services Catalog».

En el catálogo se pueden definir entornos (por ejemplo, de test, de producción) donde se adapten las conexiones a la fuentes o destinos de datos. Adicionalmente, se puede programar la ejecución de paquetes mediante los servicios que ofrece *SQL Server*, en particular, mediante *SQL Server Agent*.

En nuestro caso nos limitaremos a ejecutar los proyectos en modo debug.

4. Transformaciones adicionales a realizar

Para definir las transformaciones utiliza *SSIS*.

[1] `https://www.sentryone.com/blog/how-to-run-ssis-packages-using-32-bit-drivers-on-a-64-bit-machine`

4.1. Transformaciones para obtener el resultado inmediato

En primer lugar, vamos a definir las transformaciones basadas en las que ya hemos realizado pero con unas características específicas.

3. Crea una BD *SQL Server* cuyo nombre sea el nombre de la provincia que tienes asignada y tu nombre de usuario de correo UGR (en mi caso se llamará `granada_jsamos`). Crea las tablas `cuando`, `donde` y `padron` añadiendo al nombre el sufijo de tu nombre de usuario de correo UGR (en mi caso se llamarán `cuando_jsamos`, `donde_jsamos` y `padron_jsamos`). La estructura de estas tablas ha de ser similar a la de las hojas correspondientes del archivo obtenido con *Power Query*, **usa estas hojas como origen de datos** (en mi caso el archivo es `granada-ETL-jsamos.xlsx`) pero, para los nombres de los campos, usa minúsculas sin tilde y, en lugar del criterio *Camel Case* (`CamelCase`), usa el carácter «_» como separador (`camel_case`).

   - El nombre de cada elemento definido (por ejemplo, proyecto, flujo de control, flujo de datos, y sus componentes) ha de tener como sufijo tu nombre de usuario de correo UGR.
   - Para cada tabla define un flujo de datos específico.

4.2. Transformaciones alternativas

4. Crea una BD *SQL Server* cuyo nombre sea `prueba2` y tu nombre de usuario de correo UGR (en mi caso se llamará `prueba2_jsamos`). Crea la tabla `cuando` añadiendo al nombre el sufijo de tu nombre de usuario de correo UGR (en mi caso se llamará `cuando_jsamos`). La estructura de esta tabla ha de ser similar a la de la hoja correspondiente del archivo obtenido con *Power Query* (en mi caso el archivo es `granada-ETL-jsamos.xlsx`) pero, para los nombres de los campos, usa minúsculas sin tilde y, en lugar del criterio *Camel Case* (`CamelCase`), usa el carácter «_» como separador (`camel_case`). Define el contenido de esa tabla mediante una transformación **usando como origen la hoja `Provincia`** del archivo generado mediante *Power Query*.

   - El nombre de cada elemento definido (por ejemplo, proyecto, flujo de control, flujo de datos, y sus componentes) ha de tener como sufijo tu nombre de usuario de correo UGR.

5. **Ejercicio libre**: para realizar la transformación 4, en lugar de la hoja `Provincia` del archivo generado con *Power Query*, considera el archivo original de tu provincia. Para transformar la tabla dinámica de partida se puede usar la transformación «Unpivot» ubicada en la carpeta «Other Transforms» de la ventana «SSIS Tools».

Bibliografía

[SDKK12] Wayne Snyder, Mike Davis, Devin Knight, and Brian Knight. *Knight's Microsoft SQL Server 2012 Integration Services 24-Hour Trainer*. Wrox Press, 2012.