BI4SME

boosting business intelligence skills for SME growth

UNIT 3: ETL

BI4SME - R2 – Training materials

GRANT AGREEMENT 2021-1-ES01-KA220-VET-000033132

Co-funded by the European Union

Sommario

Co-funded by
the European Union

Public

Licence

Co-funded by
the European Union

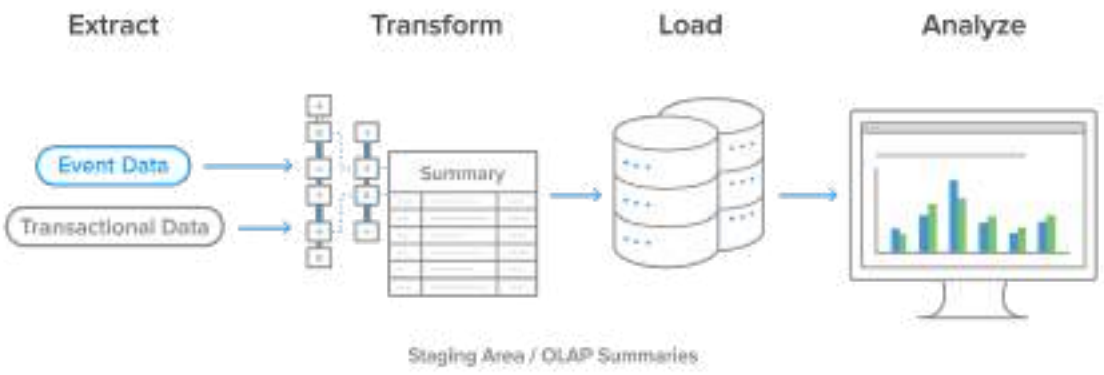
UNIT 3: ETL

3.1. Qué es el proceso de extracción, transformación y carga (ETL)?

3.1.1. Introducción

ETL significa **extraer, transformar y cargar.** Este concepto se refiere a un proceso de integración de datos que combina datos de una o múltiples fuentes, transforma el conjunto de datos original y persiste el nuevo conjunto de datos en otro sistema de destino.

El concepto de ETL se estableció en la década de 1970, cuando las bases de datos ganaron popularidad. Eventualmente, se convirtió en la forma principal de procesar datos para proyectos de almacenamiento de datos. Las empresas han utilizado durante mucho tiempo ETL como el estándar para el almacenamiento y análisis de datos. Sin embargo, a medida que avanzamos, debemos considerar ETL en el contexto de una integración empresarial amplia y resultados comerciales mejorados, no solo como un microcosmos de procesos de preparación de datos dentro de una organización. Históricamente y en su forma simple, el proceso de ETL ha sido así:



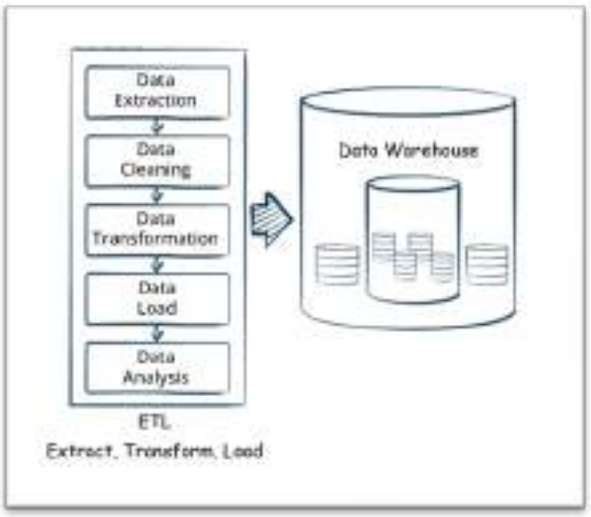
Fuente: https://www.stitchdata.com/etldatabase/etl-process/

Un sistema ETL, que ha sido construido adecuadamente, extrae datos de los sistemas fuente, aplica criterios de tipo de datos y validez de datos, y asegura que los datos cumplan estructuralmente con los requisitos de salida.

El objetivo de este proceso es proporcionar datos valiosos y bien estructurados para ser consumidos por otros usuarios. El conjunto de datos resultante puede ser utilizado para una variedad de tareas, incluyendo herramientas de reporte, aplicaciones, procesos de análisis de datos, procesos de aprendizaje automático, almacenamiento de datos u otros sistemas que puedan ayudar a los usuarios finales a tomar decisiones comerciales.

Por ejemplo, una empresa podría usar ETL para:

- Extraer datos de sistemas heredados.ù
- Limpiar los datos para mejorar la calidad de los datos y establecer consistencia.
- Cargar datos en una base de datos de destino.



Fuente: https://www.stitchdata.com/etldatabase/etl-process/

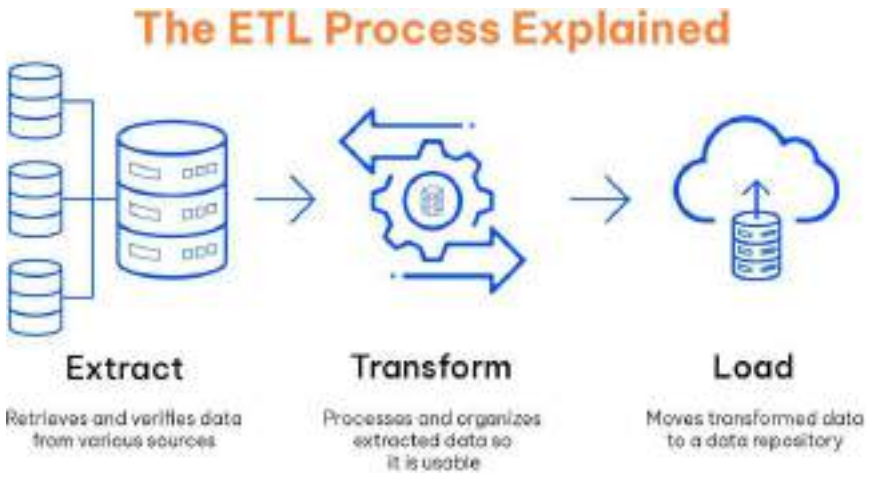
El paso final en el análisis de datos es explorar, visualizar y sacar

conclusiones de los datos utilizando enfoques estadísticos u otros. Esto podría implicar actividades como la compilación de informes, el desarrollo de modelos de previsión, la ejecución de análisis de regresión y la observación de tendencias o patrones en los datos.

En esta sección, nos vamos a centrar en el proceso ETL en sí. Los procesos y herramientas de análisis de datos en torno a los datos maestros obtenidos en el proceso ETL se tratarán en otras secciones.

3.2. Fases de un proceso ETL

Ya hemos visto las 3 etapas principales de ETL: **extracción de datos, transformación de datos y carga de datos.** Cada paso se realiza de manera secuencial. La naturaleza precisa de cada fase, cuyo formato es necesario para la base de datos de destino, depende de las demandas y especificaciones únicas de la organización.



Fuente: https://gemvietnam.com/big-data/etl-in-data-warehouse-the-definitive-guide/

Dentro de estas etapas podemos tener diferentes técnicas, múltiples tipos de formatos de datos, diferentes tipos de transformaciones, etc. Vamos a detallar cada uno de estos 3 bloques principales.

3.2.1. Fase de Extracción

El proceso ETL comienza con la extracción. Los sistemas ETL deben ser capaces de manejar varios formatos de datos para extraer datos de diferentes fuentes. Esto implica adquirir datos de numerosas fuentes como bases de datos, archivos planos y APIs.

Algunos de estos sistemas tienen soporte para la Captura de Datos de Cambio (CDC), que implica capturar solo los cambios en los datos de origen, en lugar de cargar todo el conjunto de datos cada vez.
Los datos se extraen de los sistemas de origen a lo largo del proceso de extracción y, a menudo, se mantienen en una ubicación de preparación (un área de almacenamiento intermedio donde los datos se almacenan temporalmente después de haber sido extraídos de su sistema de origen).
Los datos extraídos con frecuencia son no estructurados y están en un formato que dificulta su uso para el análisis. Para hacer que los datos sean más manejables y comprensibles, esto puede implicar acciones como filtrar, ordenar o agregar los datos.

Cómo se pueden extraer los datos?

A través de varias técnicas, como:

- **Perfilado de Datos:** Analizar los datos para entender la estructura, distribución y restricciones de los datos.
- **Validación de Datos:** Comprobar los datos contra un conjunto de reglas o restricciones para asegurar que sean precisos y completos.
- **Estandarización de Datos:** Convertir los datos en un formato consistente.
- **Desduplicación de Datos:** Identificar y eliminar datos duplicados.

Fuentes de datos heterogéneas

Las diversas fuentes de datos son un desafío común en el mundo impulsado por datos de hoy en día. Estas fuentes se refieren a información de varios sistemas, cada uno con su propia estructura, formato y tecnología.

Trabajar con fuentes de datos heterogéneas es uno de los desafíos más difíciles, ya que los datos pueden necesitar ser transformados y limpiados antes de ser cargados en el sistema de destino.

Por ejemplo, un campo en los datos de origen podría tener un tipo de dato que es incompatible con el sistema de destino, o un campo de fecha podría estar en un formato que el sistema de destino no requiere. Si los datos no se transforman adecuadamente, pueden volverse inútiles o causar errores en el sistema de destino.

Los datos de fuentes heterogéneas provienen de sistemas con estructuras, formatos y tecnologías diversas, como bases de datos, archivos planos o APIs.
Estas fuentes pueden incluir:

- Bases de datos relacionales, como MySQL, Oracle o SQL Server
- Bases de datos no relacionales, como MongoDB o Cassandra
- Archivos planos, como CSV o Excel
- APIs, como REST o SOAP
- Plataformas de redes sociales, como Twitter o Facebook
- Dispositivos IoT

Herramientas ETL como Talend e Informatica se utilizan comúnmente para extraer y transformar datos de múltiples fuentes para abordar estos desafíos. Estas herramientas frecuentemente incluyen una variedad de conectores y capacidades de transformación de datos que pueden manejar una amplia gama de tipos y formatos de datos.

Archivos Planos

Un archivo plano es una colección de datos almacenados en una base de datos bidimensional donde la información se mantiene como registros en una tabla. Las columnas de la tabla representan una dimensión de la base de datos, mientras que cada fila es un registro separado. Las bases de datos de archivos planos todavía son muy útiles como archivos de datos fáciles de crear y mantener para información de acceso común, como archivos de nombres y direcciones, listas de miembros o listas de clases. Las aplicaciones de hojas de cálculo como Excel o Google Sheets pueden usarse para crear y manipular bases de datos de archivos planos.

Tipos de Archivos

Varios tipos de archivos se utilizan comúnmente en los procesos ETL. El tipo de archivo elegido se determinará por varios factores, incluyendo el tamaño de los datos, la complejidad de la estructura de los datos y los requisitos de rendimiento y compresión. A continuación se presentan algunos de los tipos de archivos más comunes:

- **JSON (JavaScript Object Notation):** es un formato de intercambio de datos simple que es fácil de leer, escribir, analizar y producir tanto para humanos como para máquinas. Se utiliza comúnmente como un formato de almacenamiento de datos y para el intercambio de datos entre aplicaciones web y servicios web.

  Hay varios métodos para extraer datos JSON dependiendo de la fuente de los datos. Si los datos se almacenan en un archivo JSON, se puede utilizar un lenguaje de programación como Python o Java para leerlo y analizarlo en una estructura de datos utilizable.

  Después de que los datos se han extraído, deben modificarse para

coincidir con la estructura y el formato del sistema de destino. Esto puede incluir operaciones como aplanar estructuras jerárquicas, cambiar tipos de datos y mapear campos JSON a campos del sistema de destino.

- **CSV (Valores Separados por Comas):** es un formato de archivo basado en texto simple que separa los campos de datos usando una coma. Cada línea en un archivo CSV representa un solo registro, y los valores dentro de cada registro están separados por comas. Esto hace que sea fácil ver y editar los datos usando un editor de texto simple o software de hojas de cálculo. Los archivos CSV se utilizan comúnmente en los procesos ETL para importar y exportar datos entre diferentes sistemas.
- **TSV (Valores Separados por Tabulaciones):** es similar al CSV, pero en lugar de usar comas para separar los campos de datos, utiliza tabulaciones. Los archivos TSV también son archivos basados en texto simples que pueden verse y editarse fácilmente usando un editor de texto o software de hojas de cálculo.
- **Formato de archivo de ancho fijo:** es un formato de archivo de texto donde a cada campo se le asigna un ancho específico, y los datos se alinean en consecuencia. En este formato, los datos se almacenan en una estructura similar a una tabla, con cada campo ocupando un número específico de caracteres. Esto facilita la importación y exportación de datos entre sistemas, ya que los anchos de los campos se conocen de antemano. Sin embargo, los archivos de ancho fijo pueden ser más difíciles de editar y ver, ya que no son tan legibles para los humanos como los archivos CSV o TSV.
- **Excel:** es un formato de archivo binario utilizado por Microsoft Excel, que puede almacenar una gran cantidad de datos estructurados. Los archivos de Excel pueden abrirse y editarse fácilmente usando software de hojas de cálculo, lo que los convierte en una opción popular para los procesos ETL que implican datos que se actualizan o

manipulan regularmente.

- **XML (Lenguaje de Marcado Extensible):** es un lenguaje de marcado que se utiliza para almacenar e intercambiar datos estructurados. Los archivos XML contienen datos que están encerrados en etiquetas, las cuales definen la estructura de los datos. Esto facilita la representación de estructuras de datos complejas de manera organizada, y la importación y exportación de datos entre diferentes sistemas.

3.2.2. Fase de Transformación

La segunda etapa del proceso ETL implica el proceso de transformación. Los datos deben ser limpiados, estandarizados y transformados durante la etapa de transformación a un formato apropiado para el análisis y la generación de informes. La información se modifica a lo largo del proceso de transformación desde su estado bruto a un formato de archivo que puede ser introducido en el sistema de destino y utilizado para el análisis.

La transformación de datos puede involucrar tareas como:

- **Limpieza de datos:** Identificación y corrección de inexactitudes, inconsistencias y errores en los datos.
- **Estandarización de datos:** Conversión de datos a un formato consistente.
- **Integración de datos:** Combinación de datos de múltiples fuentes en un solo formato.
- **Enriquecimiento de datos:** Adición de datos a los datos existentes.
- **Agregación de datos:** Combinación de datos de múltiples registros en un solo registro.

Independientemente de dónde ocurran los cambios en el proceso, es una fase crucial en el flujo de trabajo para el análisis. Los datos se transforman para que las transformaciones estén listas para el análisis. A continuación

se presentan algunas de las transformaciones básicas más frecuentes según lo recuperado de la ETL Database:

Transformaciones Básicas:

Limpieza: La limpieza, también conocida como depuración de datos, es un proceso de identificación y corrección de inexactitudes, inconsistencias y errores en los datos. Limpiar los datos implica eliminar duplicados, corregir errores y convertirlos a un formato estándar. Esto se refiere al proceso de identificar y resolver problemas de calidad de datos como valores faltantes (NULL), valores inconsistentes o problemas de formato.

Por ejemplo, si una columna en un conjunto de datos tiene valores faltantes, podríamos reemplazar esos valores NULL con 0. De manera similar, podríamos estandarizar valores a lo largo de una columna mapeando "Male" a "M" y "Female" a "F". Los formatos de fecha podrían estandarizarse a un formato consistente para el análisis, como convertir "MM/DD/YYYY" a "YYYY-MM-DD".

Desduplicación: Esto implica identificar y eliminar registros duplicados de un conjunto de datos. Por ejemplo, si un conjunto de datos contiene múltiples entradas para el mismo cliente o transacción, podríamos eliminar todas menos una de esas entradas para crear un conjunto de datos más limpio y conciso. La desduplicación puede ayudar a reducir errores e inconsistencias en el análisis de un conjunto de datos, así como mejorar la precisión de los resultados.

Revisión de Formato: Esto implica convertir datos de un formato o unidad a otro. Por ejemplo, si un conjunto de datos incluye mediciones en unidades métricas, podríamos convertir esas mediciones a unidades imperiales para mantener la consistencia. Los conjuntos de caracteres

podrían convertirse a un formato de codificación estándar, como UTF-8, para asegurar la compatibilidad con otros sistemas. Las fechas y horas podrían convertirse de un formato a otro, como convertir del formato de marca de tiempo Unix a un formato legible por humanos.

Reestructuración de Claves: Esto se refiere al proceso de establecer relaciones clave entre tablas. Por ejemplo, si un conjunto de datos contiene información sobre clientes y sus pedidos, podríamos necesitar establecer una relación clave entre el ID del cliente en la tabla de clientes y el ID del cliente en la tabla de pedidos. Esto nos permite unir las dos tablas y analizar los datos de una manera más significativa. La reestructuración de claves es un paso importante en la integración de datos y puede ayudar a asegurar que los datos sean precisos y consistentes en diferentes fuentes.

Formato de Conversión de Campos: El formato de conversión de campos es el proceso de convertir un campo de un tipo de dato a otro. Esto se utiliza comúnmente durante la etapa de transformación de los procesos ETL. Las razones más comunes para la conversión de formato de campos son:

A. **Compatibilidad de Tipo de Dato:** En algunos casos, un campo en los datos de origen puede tener un tipo de dato que no es compatible con el sistema de destino. Por ejemplo, si los datos de origen se almacenan como una cadena de texto y el sistema de destino requiere que sea un número entero. En este caso, el campo debe convertirse al tipo de dato apropiado.
B. **Consistencia de Datos:** Algunos sistemas pueden requerir que los campos tengan un tipo de dato específico, como un campo de fecha que debe estar en un formato específico como 'YYYY-MM-DD'. Esto ayuda a asegurar la consistencia e integridad de los datos.
C. **Calidad de Datos:** Al convertir el formato de los campos, podemos verificar la calidad de los datos. Por ejemplo, un campo de número de teléfono puede convertirse a un número y verificar si tiene la longitud

correcta.

Consulta las siguientes fuentes para obtener más información sobre el tema:

SQL Server Data Type Conversion Methods and performance comparison

CAST() and CONVERT() in SQL Server

Transformaciones Avanzadas:

- **Derivación:** Aplicar reglas de negocio a tus datos que deriven nuevos valores calculados a partir de datos existentes, por ejemplo, crear una métrica de ingresos que reste los impuestos.
- **Filtrado:** Seleccionar solo ciertas filas y/o columnas.
- **Unión:** La unión de datos es el proceso de combinar filas de dos o más tablas basadas en una columna común entre ellas. Los datos de múltiples fuentes, como bases de datos o archivos planos, deben combinarse en un único conjunto de datos para esta tarea. Podemos revisar cómo funcionan los diferentes tipos de uniones (INNER JOIN, LEFT JOIN, RIGHT JOIN y FULL OUTER JOIN) en el módulo de SQL. Enlazar datos de múltiples fuentes, por ejemplo, agregando datos de gastos en anuncios a través de múltiples plataformas, como Google Adwords y Facebook Ads. Podemos combinar información de múltiples fuentes en un solo conjunto de datos unificado utilizando la unión de datos.
- **Referencias Cruzadas:** Referenciar cruzadamente datos es el proceso de comparar datos de una tabla con datos de otra tabla para identificar relaciones o patrones. Esto se realiza con una declaración SELECT de SQL y una cláusula WHERE que especifica las condiciones que deben cumplirse. Las referencias cruzadas pueden usarse para identificar duplicados, datos faltantes o datos que violan reglas de negocio específicas. Las referencias cruzadas de datos nos permiten identificar relaciones, patrones e inconsistencias en los datos.
- **División:** Dividir una sola columna en múltiples columnas.
- **Validación de Datos:** Validación de datos simple o compleja, por

ejemplo, si las tres primeras columnas en una fila están vacías, entonces rechazar la fila del procesamiento.

- **Resumen:** Los valores se resumen para obtener cifras totales que se calculan y almacenan en múltiples niveles como métricas de negocio, por ejemplo, sumar todas las compras que un cliente ha realizado para construir una métrica de valor de por vida del cliente (CLV).
- **Agregación:** Los elementos de datos se agregan de múltiples fuentes de datos y bases de datos.
- **Integración:** Dar a cada elemento de dato único un nombre estándar con una definición estándar. La integración de datos reconcilia diferentes nombres y valores de datos para el mismo elemento de datos.[1]

3.2.3. Fase de Carga

El paso final del proceso ETL es la carga. Los datos extraídos, limpiados y convertidos deben luego cargarse en un sistema de destino, como un almacén de datos o un lago de datos. De acuerdo con las necesidades del sistema, la carga tiene como objetivo garantizar que los datos se almacenen en un formato que sea sencillo de acceder y que pueda ser consultado para análisis y generación de informes.

La carga de datos puede llevarse a cabo a través de:

- **Validación de datos:** Para verificar que los datos sean correctos y completos, también puede ser necesario validar los datos, lo que implica comparar los datos con un conjunto de criterios o restricciones. Verificar los datos con un conjunto de reglas o restricciones para asegurar que sean precisos y completos. Los datos deben limpiarse antes de ser cargados en el sistema de destino

[1] Fuente: https://www.stitchdata.com/etldatabase/etl-transform/

porque los datos defectuosos o inconsistentes pueden resultar en conclusiones o decisiones incorrectas. Es un procedimiento que se realiza en la fase de transformación del proceso ETL.

- **Conciliación de datos:** Comparar los datos en el sistema de origen con los datos en el sistema de destino para asegurar que sean los mismos.
- **Archivado de datos**: Mantener una copia de los datos para referencia histórica y propósitos de cumplimiento.
- **Indexación de datos:** Crear un índice en los datos para mejorar el rendimiento de las consultas.

Tipos de Carga:

- Carga inicial: Incluye todas las tablas del almacén de datos.
- Carga incremental: En este tipo, puedes aplicar cambios continuos cuando sea necesario de vez en cuando.
- Actualización completa: Borra el contenido de una o más tablas y las recarga con nuevos datos[2]

Entender el trabajo que estás requiriendo del entorno de destino durante el proceso de carga es un asunto crucial, independientemente de tu objetivo final. Durante la carga de datos, puedes tener un impacto negativo en el sistema de alojamiento dependiendo del volumen, la estructura, el objetivo y el tipo de carga.

Hay dos métodos principales para cargar datos en un almacén:

- **Carga completa:** volcado de datos completo que se realiza la primera vez que se carga una fuente de datos en el almacén
- **Carga incremental:** el delta entre los datos de destino y de origen se vuelca a intervalos regulares. La última fecha de extracción se almacena para que solo se carguen los registros añadidos después de esta fecha. Las cargas incrementales vienen en dos variantes que varían según el volumen de datos que estás cargando:

[2] https://www.stitchdata.com/etldatabase/etl-load/

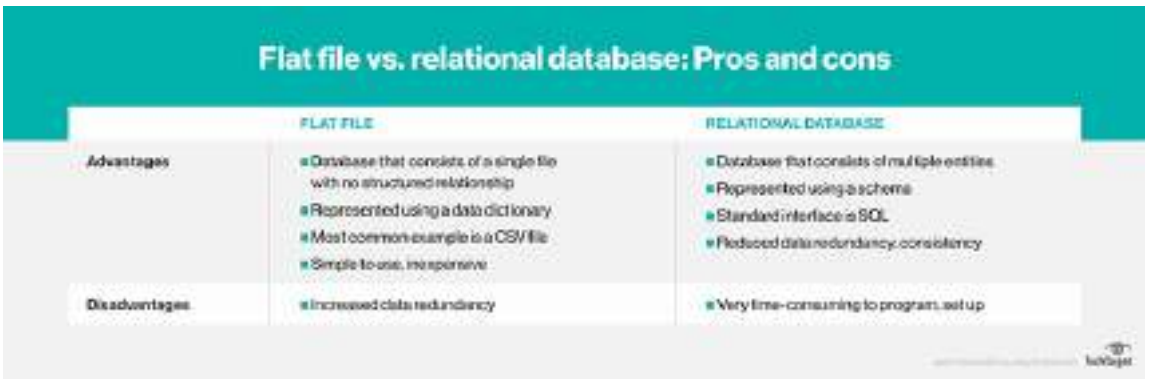
- Carga incremental por streaming: mejor para cargar volúmenes pequeños de datos.
- Carga incremental por lotes: mejor para cargar volúmenes grandes de datos.

Bases de Datos

Veamos los diferentes tipos de bases de datos que prevalecen. Un archivo plano a veces se refiere como una base de datos relacional, pero los dos tipos de bases de datos son significativamente diferentes tanto en forma como en función.

Un **archivo plano** consiste en una única tabla de datos. Permite al usuario especificar atributos de datos, como columnas y tipos de datos tabla por tabla, y almacena esos atributos separados de las aplicaciones. Este tipo de archivo se utiliza comúnmente para importar datos en proyectos de almacenamiento de datos.

En las **bases de datos relacionales,** un archivo plano a veces se usa como sinónimo de una "relación". Una base de datos relacional contiene múltiples tablas de datos que se relacionan entre sí y permite al usuario especificar información sobre múltiples tablas y las relaciones entre esas tablas, permitiendo mayor flexibilidad y control sobre las restricciones de la base de datos



Cargar datos en una base de datos relacional en SQL Server

Hay varias maneras de cargar datos en una base de datos relacional como SQL

Server. Algunos métodos comunes son:

- **Usando SQL Server Management Studio (SSMS):** Puedes usar el Asistente de Importación y Exportación en SSMS para importar datos desde varias fuentes, como Excel, CSV u otras bases de datos, a una base de datos de SQL Server.
- **Usando la utilidad bcp:** La utilidad bcp es una herramienta de línea de comandos que copia datos en bloque entre una instancia de SQL Server y un archivo de datos. Puede usarse para importar datos en una tabla de SQL Server desde un archivo de datos, o para exportar datos desde una tabla de SQL Server a un archivo de datos.
- **Usando SQL Server Integration Services (SSIS):** SSIS es una poderosa herramienta ETL que puede usarse para extraer, transformar y cargar datos en una base de datos de SQL Server. Puede usarse para cargar datos desde varias fuentes, como Excel, CSV u otras bases de datos, y para realizar transformaciones complejas de datos antes de cargar los datos en SQL Server.
- **Usando las funciones OPENROWSET o OPENDATASOURCE:** Estas funciones te permiten importar datos desde una variedad de fuentes de datos directamente en una consulta de SQL Server.
- **Usando Python u otros lenguajes de programación:** Puedes usar Python u otros lenguajes de programación para conectarte a la base de datos de SQL Server y cargar datos en ella. Librerías populares como Pyodbc y SQLAlchemy pueden usarse para crear una conexión, y luego puedes usar las declaraciones insert o bulk insert para insertar datos en la base de datos.

En el mundo de la gestión de datos, el proceso de extracción, transformación y carga (ETL) de datos es crucial. Existen una multitud de métodos y herramientas disponibles para llevar a cabo tareas ETL, y entre ellas, SQL Server Integration Services (SSIS) se destaca como una opción poderosa y versátil.

Parte del conjunto de herramientas de Microsoft SQL Server, SSIS es particularmente hábil para cargar datos en una base de datos de SQL Server desde una amplia variedad de fuentes, incluyendo Excel, CSV u otras bases de datos.

En esta guía, nos enfocaremos en usar SSIS para cargar datos en una base de datos de SQL Server. Te guiaremos a través del proceso de creación de un paquete SSIS, diseñando el proceso ETL y ejecutando el paquete para cargar datos. Este es solo un ejemplo de lo que puedes hacer con SSIS; el mejor método dependerá del tamaño y la complejidad de tus datos y de los requisitos específicos de tu proyecto.

Comenzaremos introduciendo los conceptos básicos de SSIS y explicando la estructura de un paquete SSIS típico. Luego, te guiaremos a través del proceso de creación de un nuevo proyecto de Integration Services en Visual Studio y el diseño de un paquete SSIS. Aprenderás cómo configurar el adaptador de origen y el adaptador de destino, y cómo conectarlos para crear un flujo de datos. Finalmente, te mostraremos cómo ejecutar el paquete y monitorear el proceso de carga de datos.

Cada sección de esta guía proporciona instrucciones detalladas, paso a paso, junto con explicaciones para ayudarte a comprender el propósito de cada paso. Al final de esta guía, tendrás una comprensión completa de cómo usar SSIS para cargar datos en una base de datos de SQL Server.

Cargar datos en la base de datos de SQL Server con SQL Server Integration Services (SSIS)

Crea un paquete de SQL Server Integration Services (SSIS) para cargar datos en SQL Server. Opcionalmente, puedes reestructurar, transformar y limpiar los datos a medida que pasan a través del flujo de datos de SSIS.

Esta guía te muestra cómo hacer las siguientes cosas:

- Crear un nuevo proyecto de Integration Services en Visual Studio.
- Configurar la conexión desde la fuente hasta el destino.
- Ejecutar el paquete SSIS para cargar los datos.

Conceptos Básicos

En SSIS, la unidad fundamental de trabajo es un 'paquete'. Los paquetes que comparten un propósito común se agrupan en 'proyectos'. La creación de estos proyectos y el diseño de paquetes se llevan a cabo en Visual Studio utilizando SQL Server Data Tools. El proceso de diseño es altamente visual, involucrando arrastrar y soltar componentes desde la caja de herramientas a la superficie de diseño, conectarlos y configurar sus propiedades. Una vez que tu paquete esté completo, tienes la opción de ejecutarlo directamente o desplegarlo en SQL Server o SQL Database. Desplegar el paquete permite una gestión completa, monitoreo y seguridad mejorada.

Sobre la Solución

La solución en la que nos enfocamos en esta guía es un paquete típico que emplea una tarea de flujo de datos que contiene una fuente y un destino. Este enfoque es versátil, soportando una amplia gama de fuentes de datos, incluyendo SQL Server y Azure SQL Database. Para los propósitos de esta guía, utilizaremos SQL Server como la fuente de datos, que puede ejecutarse tanto en las instalaciones como en una máquina virtual de Azure.

Para establecer una conexión con SQL Server y SQL Database, puedes usar un administrador de conexiones ADO.NET y origen y destino, o un administrador de conexiones OLE DB y origen y destino. En este tutorial usaremos ADO.NET debido a su simplicidad y menores opciones de configuración, aunque OLE DB puede ofrecer un rendimiento ligeramente mejor.

Como atajo, puedes utilizar el Asistente de Importación y Exportación de SQL Server para crear el paquete básico. Después de guardar el paquete, puedes abrirlo en Visual Studio o SSDT para verlo y personalizarlo más.

Para más información, consulta Importar y Exportar Datos con el Asistente de

Importación y Exportación de SQL Server.

Requisitos Previos

Para seguir este tutorial, necesitas lo siguiente:

1. **SQL Server Integration Services (SSIS):** SSIS es un componente de SQL Server y es esencial para crear y ejecutar paquetes ETL. Necesitarás una versión con licencia, o la versión para desarrolladores o de evaluación, de SQL Server. Para obtener una versión de evaluación de SQL Server, consulta Evaluar SQL Server.
2. **Visual Studio (opcional):** Visual Studio proporciona el entorno donde crearás y gestionarás tus proyectos SSIS. La edición gratuita Visual Studio Community Edition es suficiente para nuestros propósitos. Si prefieres no instalar Visual Studio, puedes instalar solo SQL Server Data Tools (SSDT), que instala una versión de Visual Studio con funcionalidad limitada.
3. **SQL Server Data Tools para Visual Studio (SSDT):** SSDT es un conjunto de herramientas que te permiten crear, probar y desplegar paquetes SSIS. Para obtener SQL Server Data Tools para Visual Studio, consulta Descargar SQL Server Data Tools (SSDT).
4. **Una instancia de SQL Server:** Aquí es donde se desplegarán y ejecutarán tus paquetes SSIS. SQL Server puede ejecutarse en las instalaciones o en una máquina virtual de Azure. Para descargar una edición de evaluación gratuita o para desarrolladores de SQL Server, consulta Descargas de SQL Server.
5. **Datos de muestra:** Este tutorial utiliza datos de muestra almacenados en SQL Server en la base de datos de ejemplo AdventureWorks como datos fuente. Tener datos de muestra para trabajar te permitirá entender mejor el proceso ETL. Para obtener la base de datos de ejemplo AdventureWorks, consulta Bases de Datos de Muestra AdventureWorks.
6. **Regla de firewall si estás cargando datos en SQL Database:** Si estás cargando datos en SQL Database, necesitarás crear una regla de firewall en SQL Database con la dirección IP de tu computadora

local. Esto asegura la transferencia segura de datos al SQL Database.

7. **Una regla de firewall si estás cargando datos en SQL Database:** Si estás cargando datos en SQL Database, necesitarás crear una regla de firewall en SQL Database con la dirección IP de tu computadora local. Esto asegura la transferencia segura de datos al SQL Database.

Crear un nuevo proyecto de Integration Services

1. **Crear un nuevo proyecto:** Ve al menú Archivo y selecciona Nuevo | Proyecto. Esto abrirá un cuadro de diálogo donde podrás crear un nuevo proyecto.

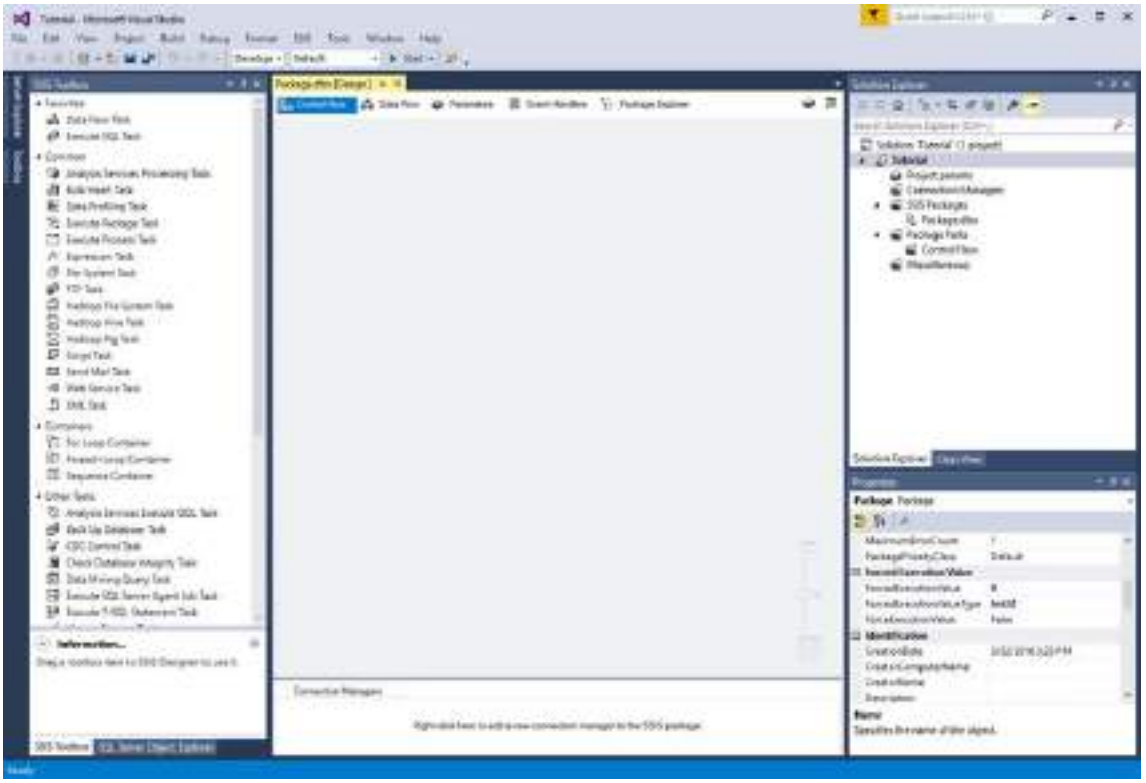
2. **Seleccionar el tipo de proyecto:** Navega a Instalado | Plantillas | Inteligencia de Negocios | Tipos de proyectos de Integration Services. Esto filtrará las plantillas de proyectos disponibles para mostrar solo aquellas relevantes para Integration Services.

3. **Configurar el proyecto:** Selecciona 'Proyecto de Integration Services'. Luego, deberás proporcionar un nombre para tu proyecto y elegir dónde se guardará en tu computadora. Una vez hecho esto, selecciona OK.

Después de completar estos pasos, Visual Studio creará un nuevo proyecto de Integration Services (SSIS) y abrirá el diseñador. El diseñador es donde crearás tu paquete SSIS (Package.dtsx). Verás varias áreas en la pantalla:

- A la izquierda, verás la Caja de herramientas, que contiene los componentes SSIS que puedes usar en tu paquete.
- En el centro, verás la superficie de diseño. Esta tiene múltiples pestañas, pero típicamente usarás al menos las pestañas de Flujo de control y Flujo de datos.
- A la derecha, verás el Explorador de soluciones y los paneles de Propiedades. El Explorador de soluciones muestra los archivos en tu proyecto, y el panel de

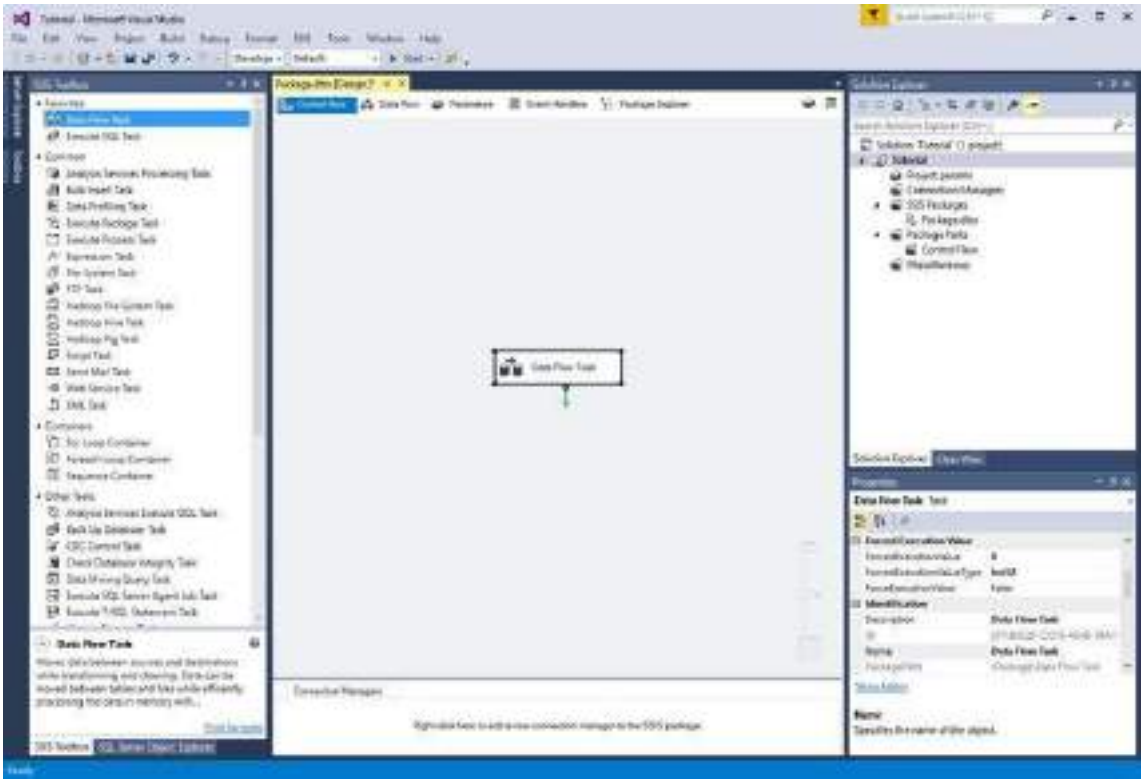
Propiedades muestra las propiedades del elemento actualmente seleccionado.



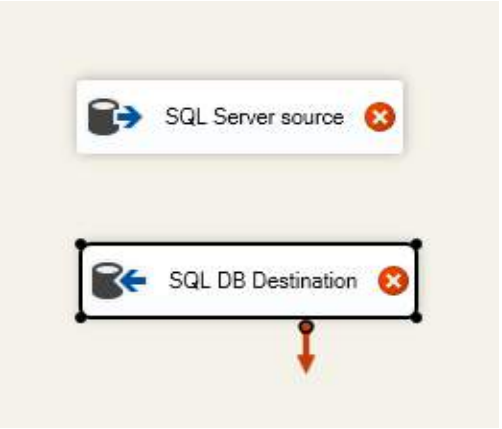
Crear el flujo de datos básico

1. **Comienza con una tarea de flujo de datos:** En la pestaña de Flujo de Control, arrastra una Tarea de Flujo de Datos desde la Caja de herramientas al centro de la superficie de diseño. Esta tarea servirá como la base para tu flujo de datos.

Co-funded by
the European Union



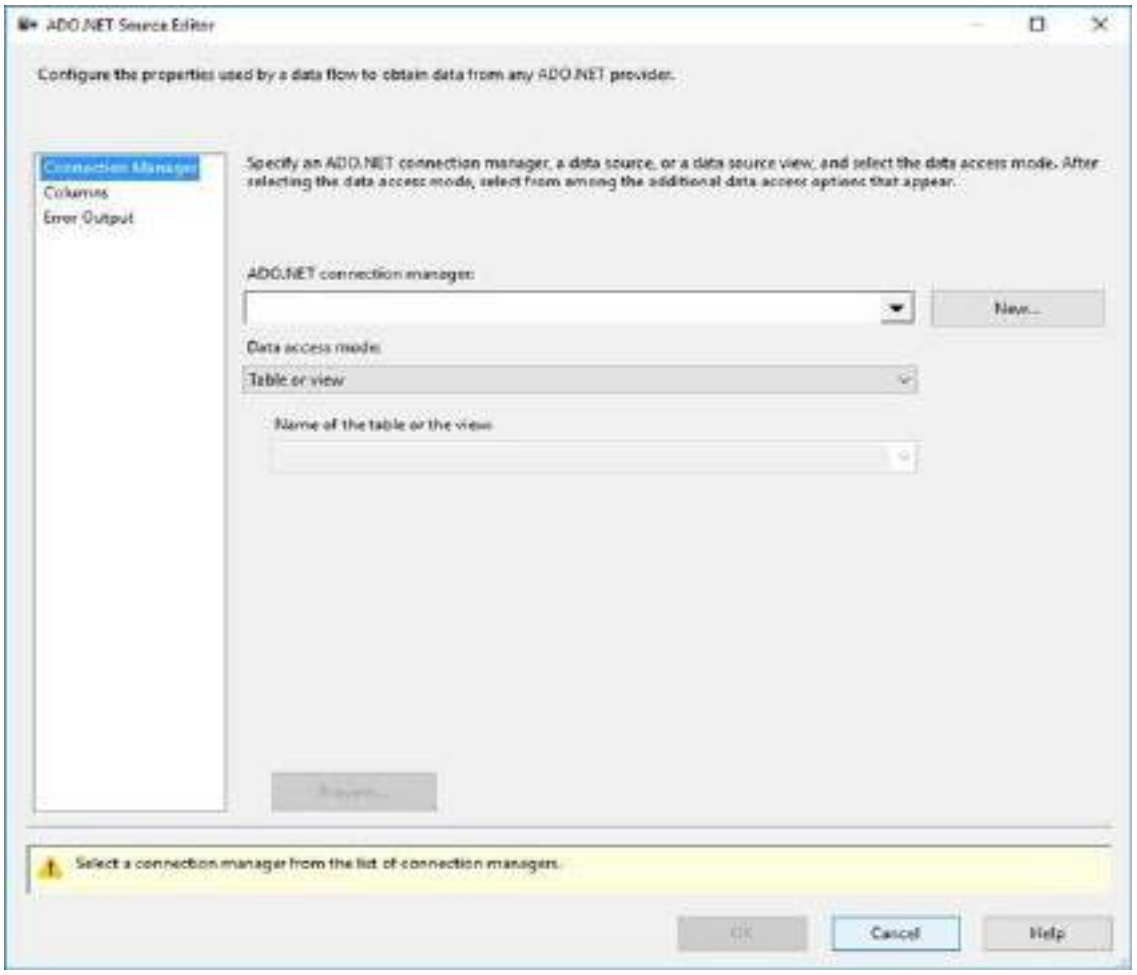
2. **Cambia a la pestaña de Flujo de Datos:** Haz doble clic en la Tarea de Flujo de Datos que acabas de agregar. Esto cambiará tu vista a la pestaña de Flujo de Datos, donde configurarás los detalles específicos de tu flujo de datos.
3. **Agrega una fuente:** Desde la lista de Otras Fuentes en la Caja de herramientas, arrastra una Fuente ADO.NET a la superficie de diseño. Esta será la fuente de tus datos. Con el adaptador de fuente aún seleccionado, ve al panel de Propiedades y cambia su nombre a 'Fuente SQL Server'. Esto ayuda a clarificar el rol de este componente en tu flujo de datos.
4. **Agrega un destino:** Desde la lista de Otros Destinos en la Caja de herramientas, arrastra un Destino ADO.NET a la superficie de diseño, colocándolo debajo de la Fuente ADO.NET. Este será el destino de tus datos. Con el adaptador de destino aún seleccionado, ve al panel de



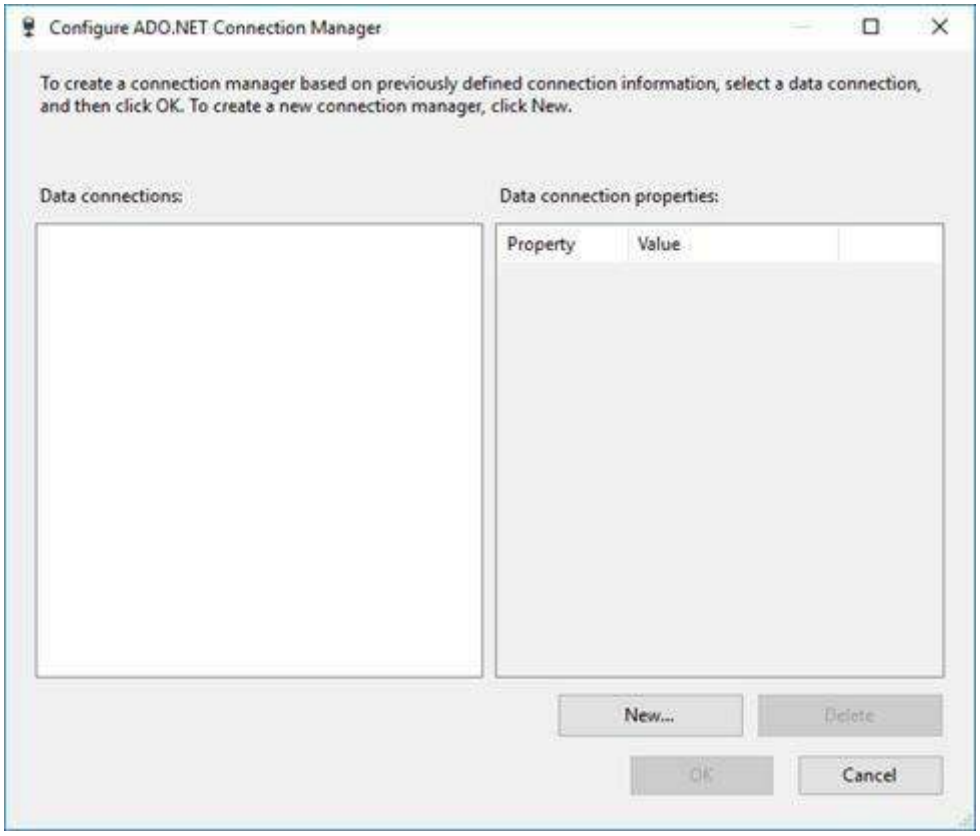
Propiedades y cambia su nombre a 'Destino SQL'. Esto ayuda a clarificar el rol de este de datos.

Configurar el adaptador de origen

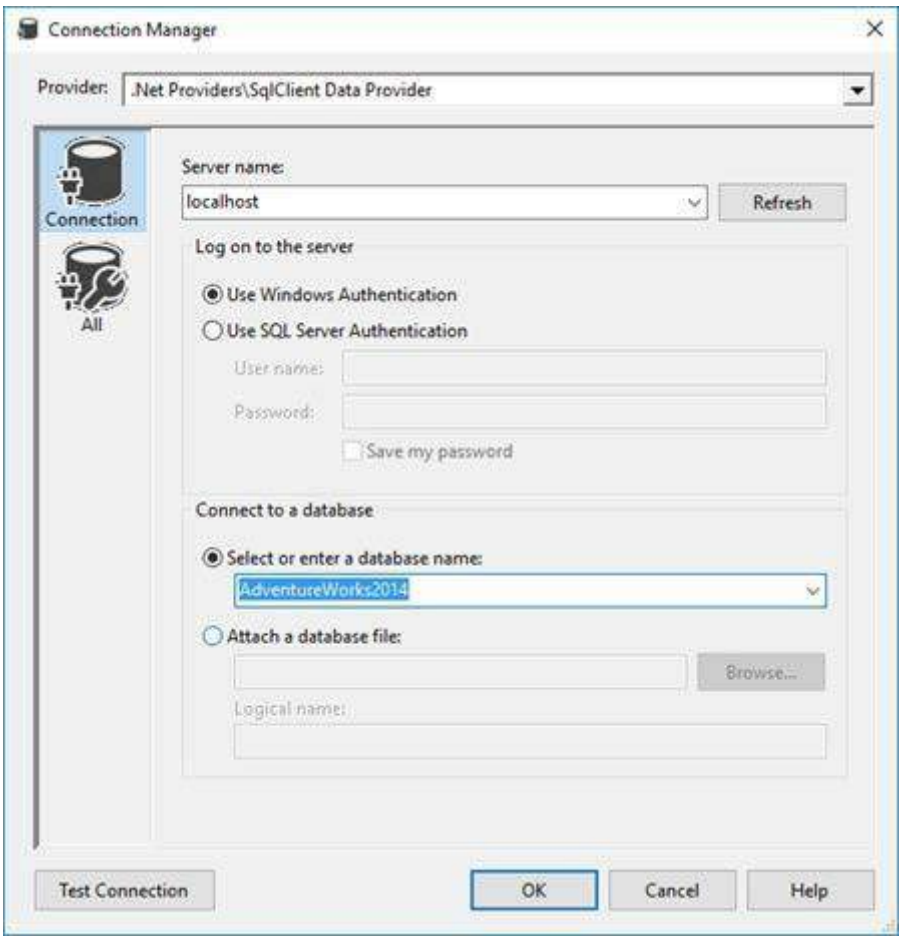
1. **Abre el Editor de Fuente ADO.NET:** Haz doble clic en el adaptador de fuente para abrir el Editor de Fuente ADO.NET. Aquí es donde configurarás los ajustes para tu fuente de datos.



2. **Crear configuraciones de conexión: En** la pestaña Administrador de Conexiones del Editor de Fuente ADO.NET, haz clic en el botón Nuevo al lado de la lista de administradores de conexiones ADO.NET. Esto abrirá el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET, donde crearás configuraciones de conexión para la base de datos de SQL Server desde la cual este tutorial cargará datos.



3. **Crear una nueva conexión de datos:** En el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET, haz clic en el botón Nuevo. Esto abrirá el cuadro de diálogo Administrador de Conexiones, donde crearás una nueva conexión de datos.



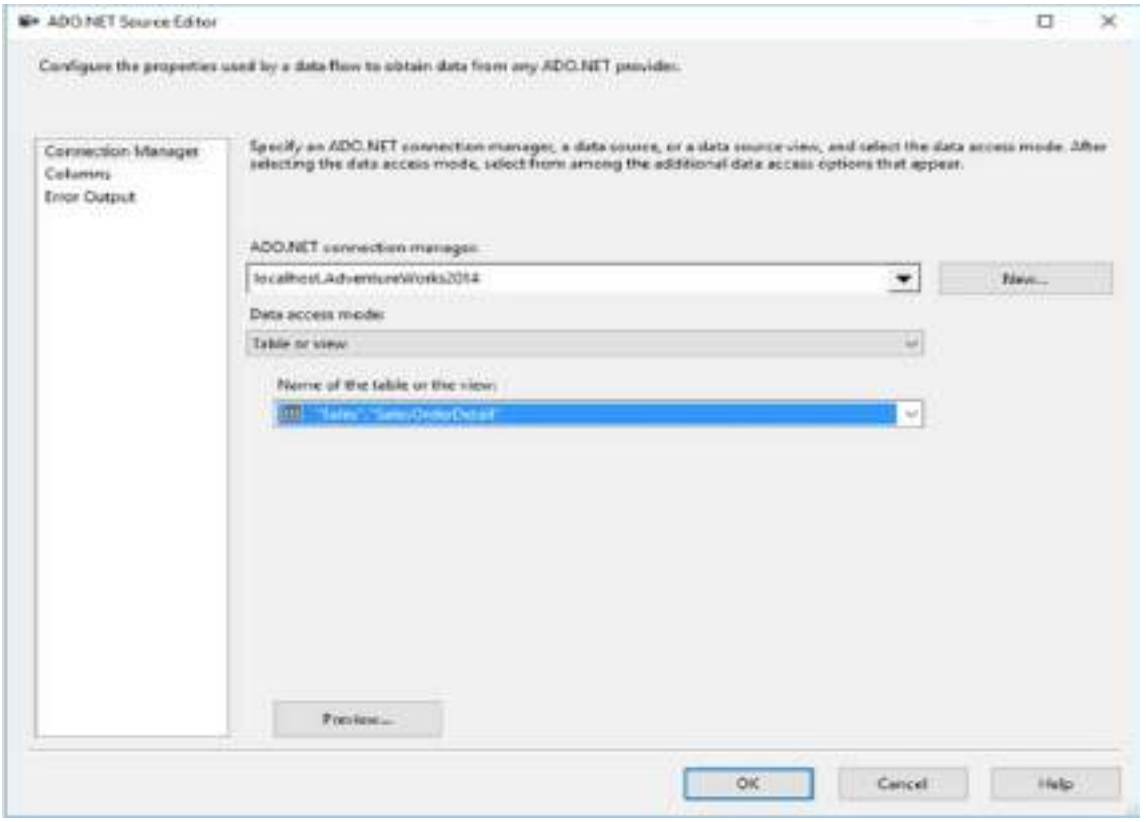
4. **Configurar la conexión de datos:** En el cuadro de diálogo Administrador de Conexiones, haz lo siguiente:
    a. Para Proveedor, selecciona el Proveedor de Datos SqlClient.
    b. Para Nombre del servidor, ingresa el nombre del servidor SQL.
    c. En la sección Iniciar sesión en el servidor, selecciona o ingresa la información de autenticación.
    d. En la sección Conectar a una base de datos, selecciona la base de datos de muestra AdventureWorks.
    e. Haz clic en Probar Conexión para asegurar que tus configuraciones sean correctas.

Co-funded by
the European Union

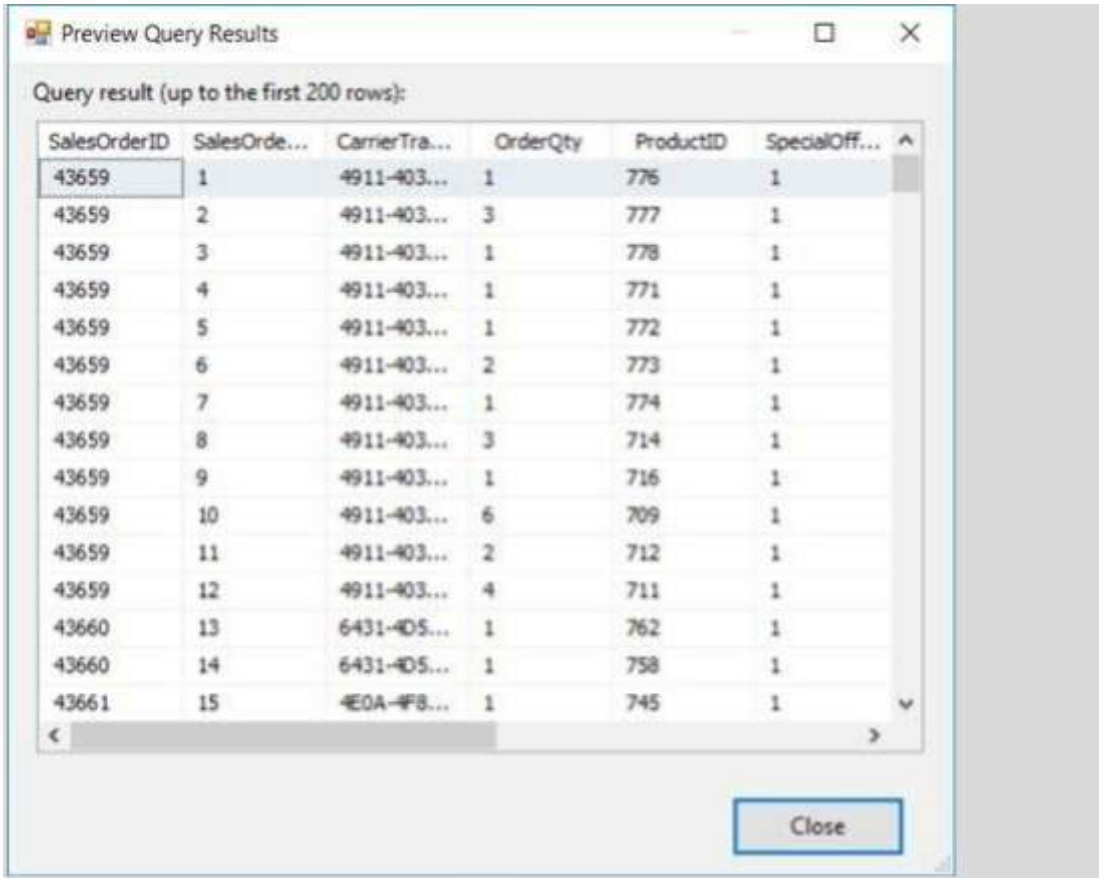
f. En el cuadro de diálogo que informa los resultados de la prueba de conexión, haz clic en OK para volver al cuadro de diálogo Administrador de Conexiones.

g. En el cuadro de diálogo Administrador de Conexiones, haz clic en OK para volver al cuadro de diálogo Configurar Administrador de Conexiones ADO.NET.

5. **Volver al Editor de Fuente ADO.NET:** En el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET, haz clic en OK para volver al Editor de Fuente ADO.NET.
6. **Seleccionar la tabla de origen:** En el Editor de Fuente ADO.NET, en la lista Nombre de la tabla o vista, selecciona la tabla Sales.SalesOrderDetail. Esta es la tabla de la que cargarás datos.



7. **Vista previa de los datos de origen**: Haz clic en Vista previa para ver las primeras 200 filas de datos en la tabla de origen en el cuadro de diálogo Resultados de la consulta de vista previa. Esto puede ayudarte a verificar que estás trabajando con los datos correctos.



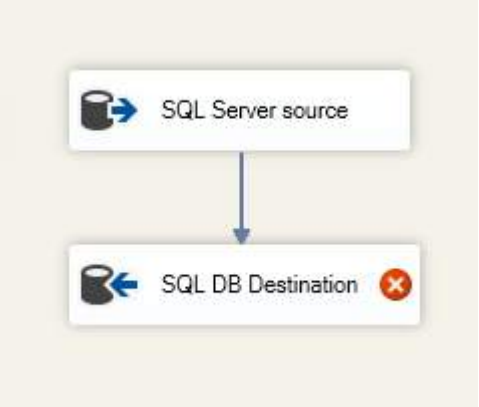
8. **Cerrar la vista previa:** En el cuadro de diálogo Resultados de la consulta de vista previa, haz clic en Cerrar para volver al Editor de Fuente ADO.NET.
9. **Terminar de configurar la fuente de datos:** En el Editor de Fuente ADO.NET, haz clic en OK para terminar de configurar la fuente de datos.

Conectar el adaptador de origen al adaptador de destino

1. **Seleccionar el adaptador de origen:** En la superficie de diseño, haz clic en el adaptador de origen para seleccionarlo. Este es el componente que configuraste en los pasos anteriores para extraer datos de tu fuente.

2. **Conectar la fuente al destino:** Verás una flecha azul que se

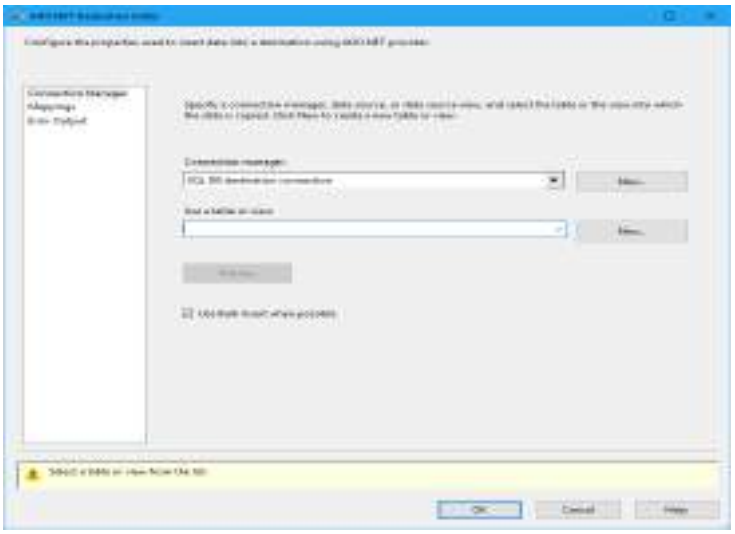
extiende desde el adaptador de origen. Haz clic y arrastra esta flecha hasta el adaptador de destino. A medida que arrastras, verás una línea que conecta los dos componentes. Suelta el botón del ratón cuando la línea se ajuste en su lugar. Esto crea una ruta de flujo de datos desde la fuente hasta el destino.



En un paquete típico de SSIS, utilizas varios otros componentes de la Caja de herramientas de SSIS entre la fuente y el destino para reestructurar, transformar y limpiar tus datos a medida que pasan a través del flujo de datos de SSIS. Para mantener este ejemplo lo más simple posible, estamos conectando la fuente directamente al destino.

Configurar el adaptador de destino

1. **Abre el Editor de Destino ADO.NET:** Haz doble clic en el adaptador de destino para abrir el Editor de Destino ADO.NET. Aquí es donde configurarás los ajustes para tu destino de datos.



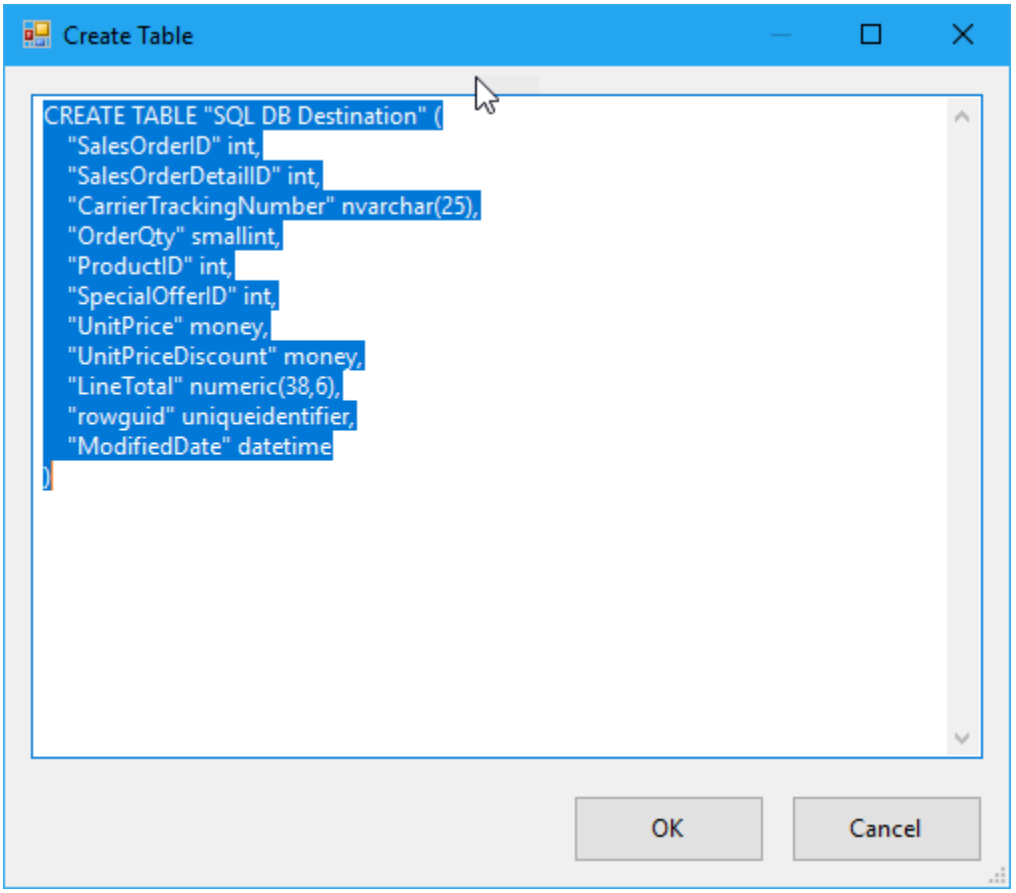
2. En la pestaña Administrador de Conexiones del Editor de Destino

ADO.NET, haz clic en el botón Nuevo al lado de la lista de administradores de conexiones para abrir el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET y crear configuraciones de conexión para la base de datos en la que este tutorial cargará datos.

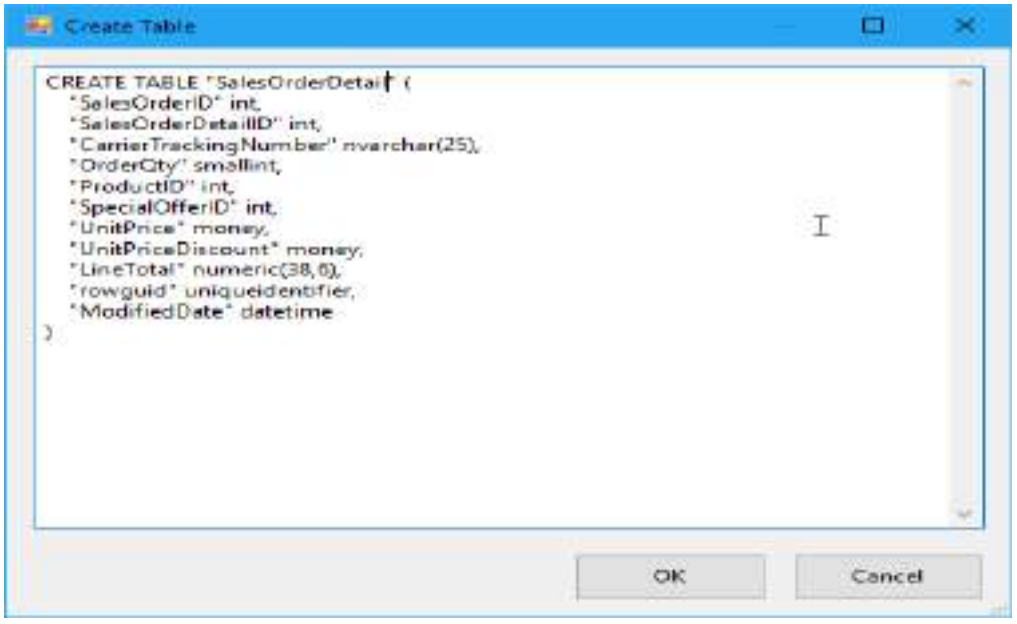
3. En el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET, haz clic en el botón Nuevo para abrir el cuadro de diálogo Administrador de Conexiones y crear una nueva conexión de datos.
4. **Configurar la conexión de datos:** En el cuadro de diálogo Administrador de Conexiones, haz lo siguiente:
    a. Para Proveedor, selecciona el Proveedor de Datos SqlClient.
    b. Para Nombre del servidor, ingresa el nombre del servidor SQL o del servidor de la base de datos SQL.
    c. En la sección Iniciar sesión en el servidor, selecciona Usar autenticación de SQL Server e ingresa la información de autenticación.
    d. En la sección Conectar a una base de datos, selecciona una base de datos existente.
    e. Haz clic en Probar Conexión para asegurar que tus configuraciones sean correctas.
    f. En el cuadro de diálogo que informa los resultados de la prueba de conexión, haz clic en OK para volver al cuadro de diálogo Administrador de Conexiones.
    g. En el cuadro de diálogo Administrador de Conexiones, haz clic en OK para volver al cuadro de diálogo Configurar Administrador de Conexiones ADO.NET.

5. **Volver al Editor de Destino ADO.NET:** En el cuadro de diálogo Configurar Administrador de Conexiones ADO.NET, haz clic en OK para volver al Editor de Destino ADO.NET.

6. **Crear una nueva tabla de destino**: En el Editor de Destino ADO.NET, haz clic en Nuevo al lado de la lista Usar una tabla o vista. Esto abrirá el cuadro de diálogo Crear Tabla, donde crearás una nueva tabla de destino con una lista de columnas que coincida con la tabla de origen.

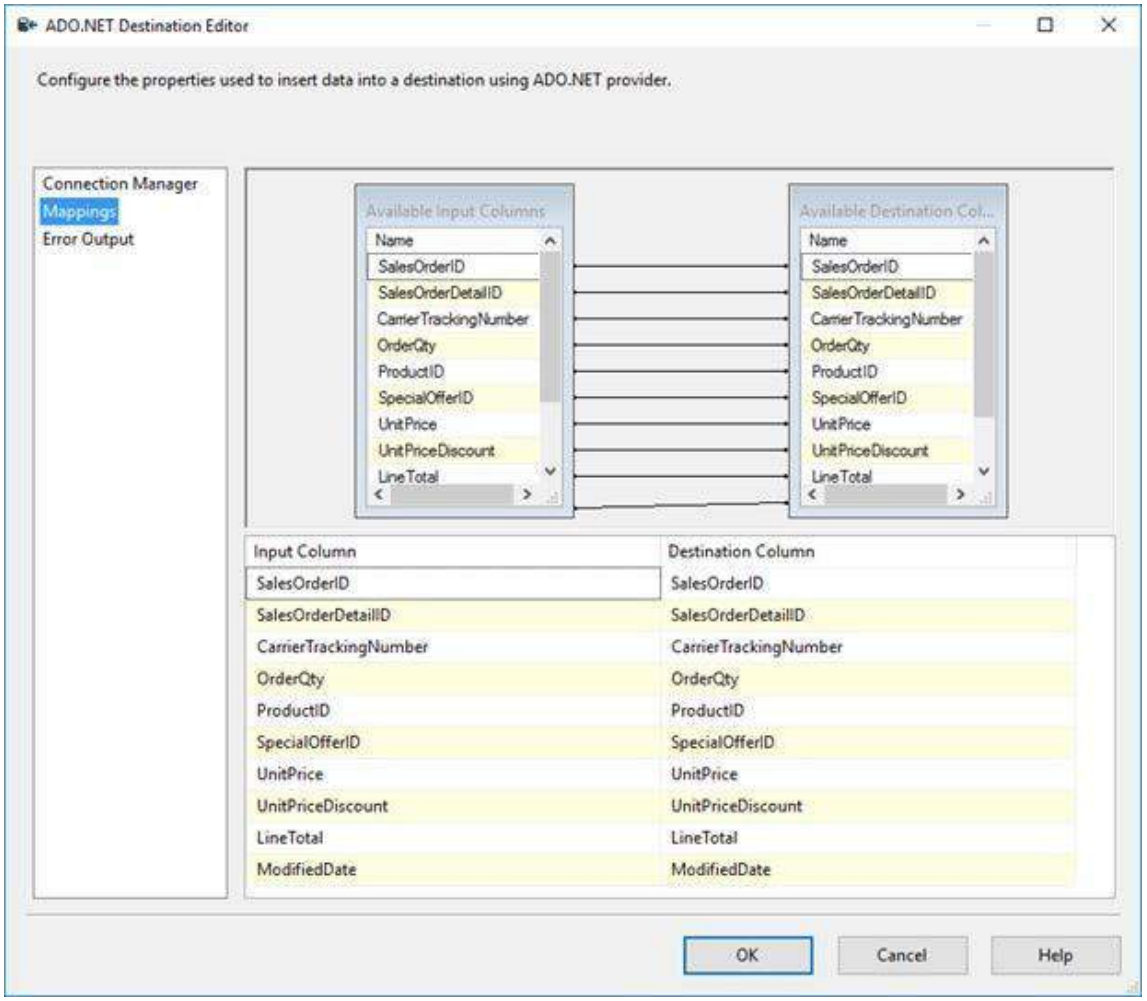


7. **Configurar la tabla de destino:** En el cuadro de diálogo Crear Tabla, haz lo siguiente:
   i. Cambia el nombre de la tabla de destino a SalesOrderDetail.



ii. Haz clic en OK para crear la tabla y volver al Editor de Destino ADO.NET.

8. **Verificar las asignaciones de columnas:** En el Editor de Destino ADO.NET, selecciona la pestaña Asignaciones. Esto te mostrará cómo las columnas en la fuente están asignadas a las columnas en el destino. Asegúrate de que todas las asignaciones sean correctas.



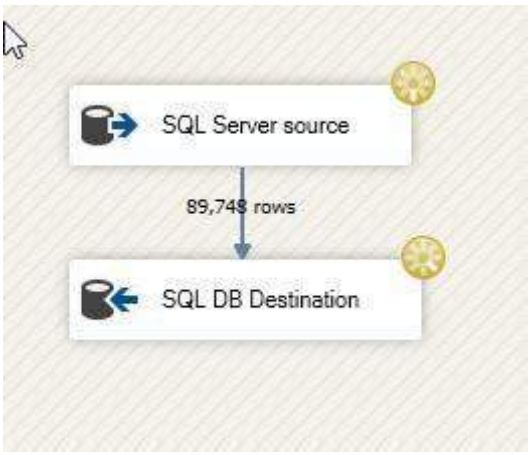
9. **Terminar de configurar el destino:** En el Editor de Destino ADO.NET, haz clic en OK para terminar de configurar el destino de los datos.

Ejecutar el paquete para cargar los datos

Ejecutar el Paquete SSIS: Ahora que has configurado los adaptadores de origen y destino y has establecido un flujo de datos entre ellos, es

momento de ejecutar el paquete. Puedes hacerlo haciendo clic en el botón Iniciar en la barra de herramientas o seleccionando una de las opciones de Ejecutar en el menú Depurar.

Monitorear el Proceso de Ejecución: A medida que el paquete comienza a ejecutarse, verás ruedas giratorias amarillas en la superficie de diseño. Estas indican que el paquete está actualmente en ejecución. También verás un conteo del número de filas que han sido procesadas hasta el momento. Esto te brinda una vista en tiempo real del proceso de transferencia de datos.



Revisar los Resultados de la Ejecución: Una vez que el paquete haya terminado de ejecutarse, las ruedas giratorias amarillas serán reemplazadas por marcas de verificación verdes. Estas indican que el paquete se ha ejecutado exitosamente. También verás el número total de filas de datos que fueron cargadas desde la fuente hasta el destino. Esto te permite confirmar que todos los datos esperados han sido transferidos.

Felicidades! Has utilizado con éxito SQL Server Integration Services para cargar datos en la base de datos de SQL Server.

Para más información, consulta los siguientes enlaces:

ETL (Extract, Transform, Load) ETL Database
BigQuery for data warehouse professionals

The Key Steps in the ETL Data Integration Process
ETL in Data Warehouse: The Definitive Guide

3.3. ETL Process in a Business Intelligence Project

ETL and BI

ETL TOOLS	BI TOOLS
Los datos se extraen de diversas fuentes, se transforman y se cargan en un sistema de almacenamiento de datos utilizando tecnologías ETL.	Las tecnologías de BI se utilizan para crear informes personalizados e interactivos para los usuarios finales, así como visualizaciones de datos para reuniones trimestrales, anuales y mensuales de la junta directiva.

Una de las técnicas más importantes de Integración de Datos es **ETL BI**
Para obtener valiosos conocimientos empresariales, el proceso ETL BI implica extraer datos de diversas fuentes, transformarlos en un formato estándar y luego colocar los datos transformados en un nuevo Almacén de Datos.
Típicamente, ETL BI se implementa utilizando herramientas ETL BI, que proporcionan a los desarrolladores la capacidad de construir scripts ETL y llevar a cabo varias tareas de gestión y desarrollo.

Con el auge de la **Nube (CLOUD),** muchas organizaciones están buscando utilizar el proceso ETL BI para mover sus datos desde sistemas de origen heredados a plataformas en la Nube.

El rendimiento y la escalabilidad suelen faltar en las organizaciones que utilizan fuentes de datos históricas como los sistemas de gestión de bases de datos relacionales (RDBMS), almacenes de datos (DW) y otros.

Por lo tanto, las empresas están moviendo sus datos a tecnologías en la nube como Amazon Web Services, Google Cloud Platform, Microsoft Azure, Nubes Privadas y muchas más, para mejorar el rendimiento, la escalabilidad y la tolerancia a fallos.

Los sistemas y procedimientos de Inteligencia Empresarial (BI) utilizados hoy en día dependen en gran medida de ETL. Organizaciones de todos los tamaños han utilizado el proceso ETL BI para ayudarlas a obtener conclusiones reveladoras de sus enormes silos de datos. Al ayudar con la integridad de los datos, el proceso ETL BI permite que una empresa tome decisiones precisas y más efectivas.

Qué implica ETL?

La solución creada y probada se despliega y se pone en uso durante la fase de implementación de un proyecto ETL. Esto puede requerir la configuración de la infraestructura necesaria para habilitar el proceso ETL, así como la instalación y configuración del software ETL.

Dependiendo de los requisitos del negocio, el proceso a menudo se configura para ejecutarse en un horario regular durante la fase de implementación, como diariamente o cada hora. La mayoría de los procesos están automatizados para que puedan funcionar sin interacción humana.

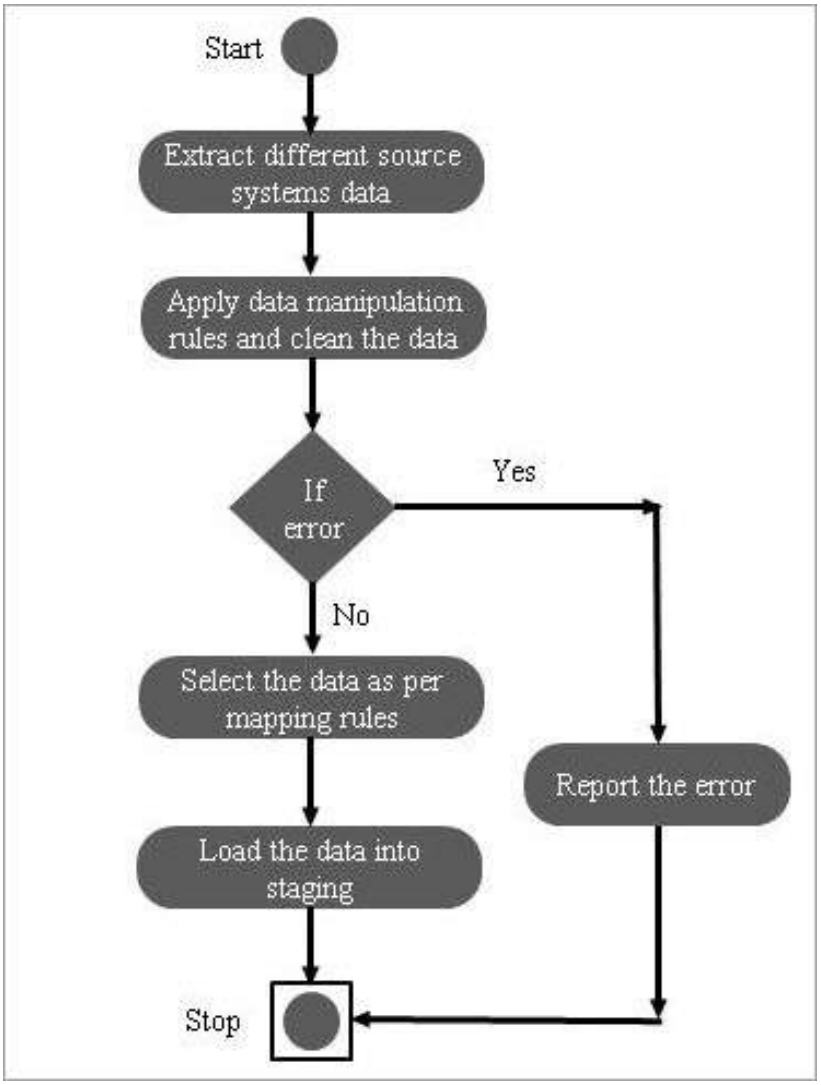
Sigamos un ejemplo de la fase de implementación de un proyecto ETL:

1. **Definir los requisitos:** El primer paso es definir claramente los requisitos para el proceso ETL, **incluyendo las fuentes de datos, los sistemas de destino y el formato deseado de los datos.**
2. **Diseñar la arquitectura ETL:** Basado en los requisitos, el siguiente paso es diseñar la arquitectura ETL, **incluyendo los componentes de extracción, transformación y carga de datos, así como los componentes de calidad de datos, manejo de errores y programación.**
3. **Extraer datos de las fuentes:** El siguiente paso es extraer datos de las diversas fuentes y cargarlos en el sistema ETL. Esto implica configurar la capa de extracción de datos para recuperar datos **de diferentes formatos de archivos, bases de datos, APIs y otras fuentes.**
4. **Transformar los datos:** Los datos extraídos se transforman luego en el formato deseado, lo cual depende de los requisitos, **incluyendo la limpieza, normalización y transformación de los datos para ajustarse a las necesidades específicas del sistema de destino.**
5. **Cargar los datos en el sistema de destino:** Los datos transformados se cargan luego en el sistema de destino, como un almacén de datos o un data mart, utilizando las técnicas apropiadas de carga de datos, **como carga en bloque, carga incremental o transmisión en tiempo real.**
6. **Validar los datos:** Finalmente, los datos se validan para asegurar su **precisión y calidad,** y para asegurar que todos los datos se hayan cargado exitosamente en el sistema de destino.
7. **Desplegar la solución ETL:** Una vez que la implementación esté completa y los datos hayan sido validados, la solución ETL se despliega en un entorno de producción y los componentes de programación y orquestación se configuran **para ejecutar el proceso ETL en intervalos específicos.**

3.3.1 Flujo de datos: Extracción

La extracción de datos del sistema de origen al área de preparación es parte de este proceso. Cualquier modificación puede realizarse en el área de preparación sin afectar el rendimiento del sistema de origen.

Antes de cargar y eliminar físicamente los datos, necesitas un mapa lógico de datos. El mapa de datos representa la relación entre las fuentes y los datos de destino.



Fuente: https://www.softwaretestinghelp.com/etl-process-in-data-warehouse/#Data_Extraction

Un mapa de datos es un componente importante en el proceso de extracción de una tubería ETL. Es un mapeo entre los datos de origen y los

datos de destino que define cómo los datos serán transformados de un formato a otro. ¿Cuáles son los pasos para construir un mapa de datos para el proceso de extracción de un ETL?

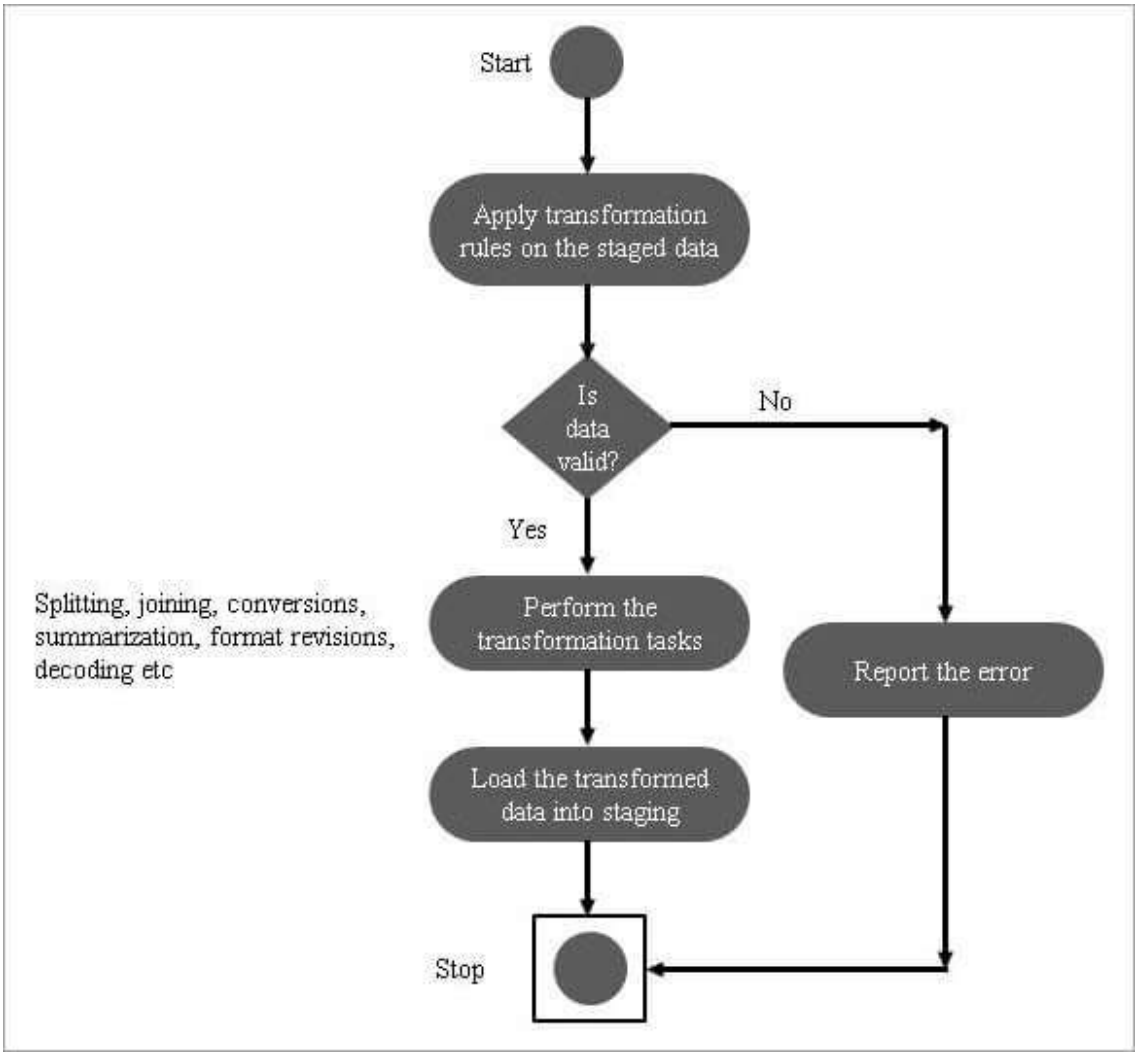
1. **Definir los sistemas de origen y destino**: Antes de crear un mapa de datos, necesitas definir el sistema de origen, de donde se extraerán los datos, y el sistema de destino, donde se cargarán los datos. Esto puede implicar definir las estructuras de datos y los tipos de datos en ambos sistemas.
2. **Analizar los datos:** Una vez que hayas definido los sistemas de origen y destino, necesitas analizar los datos para determinar qué datos necesitan ser extraídos y qué datos serán transformados. Esto puede implicar revisar muestras de datos y realizar un perfilado de datos para identificar patrones y anomalías en los datos.
3. **Crear el mapa de datos:** Basado en el análisis de los datos, puedes crear un mapa de datos que defina las relaciones entre los datos de origen y los datos de destino. Esto puede implicar crear un mapeo entre columnas en los sistemas de origen y destino, definir transformaciones de datos y crear declaraciones condicionales para manejar excepciones de datos.
4. **Probar el mapa de datos:** Una vez que hayas creado el mapa de datos, necesitas probarlo para asegurar que es preciso y que los datos serán transformados correctamente. Esto puede implicar ejecutar datos de prueba a través del mapa de datos y revisar los resultados para asegurar que los datos se transforman como se espera.
5. **Implementar el mapa de datos:** Una vez que hayas probado el mapa de datos y realizado las modificaciones necesarias, puedes implementarlo como parte de la tubería ETL. Esto puede implicar integrar el mapa de datos en la herramienta ETL, como Talend o Informatica, o implementarlo como parte de un proceso ETL personalizado

.

3.3.2 Flujo de datos: Limpieza y conformación

Los datos extraídos del servidor de origen son insuficientes e inutilizables tal como están. Por lo tanto, debes purificarlos, mapearlos y cambiarlos. Este es el paso más crucial en el proceso ETL donde los datos se mejoran y transforman para generar informes de BI intuitivos.

En la segunda etapa, los datos extraídos se someten a una serie de funciones, y este proceso depende completamente del uso de los datos:

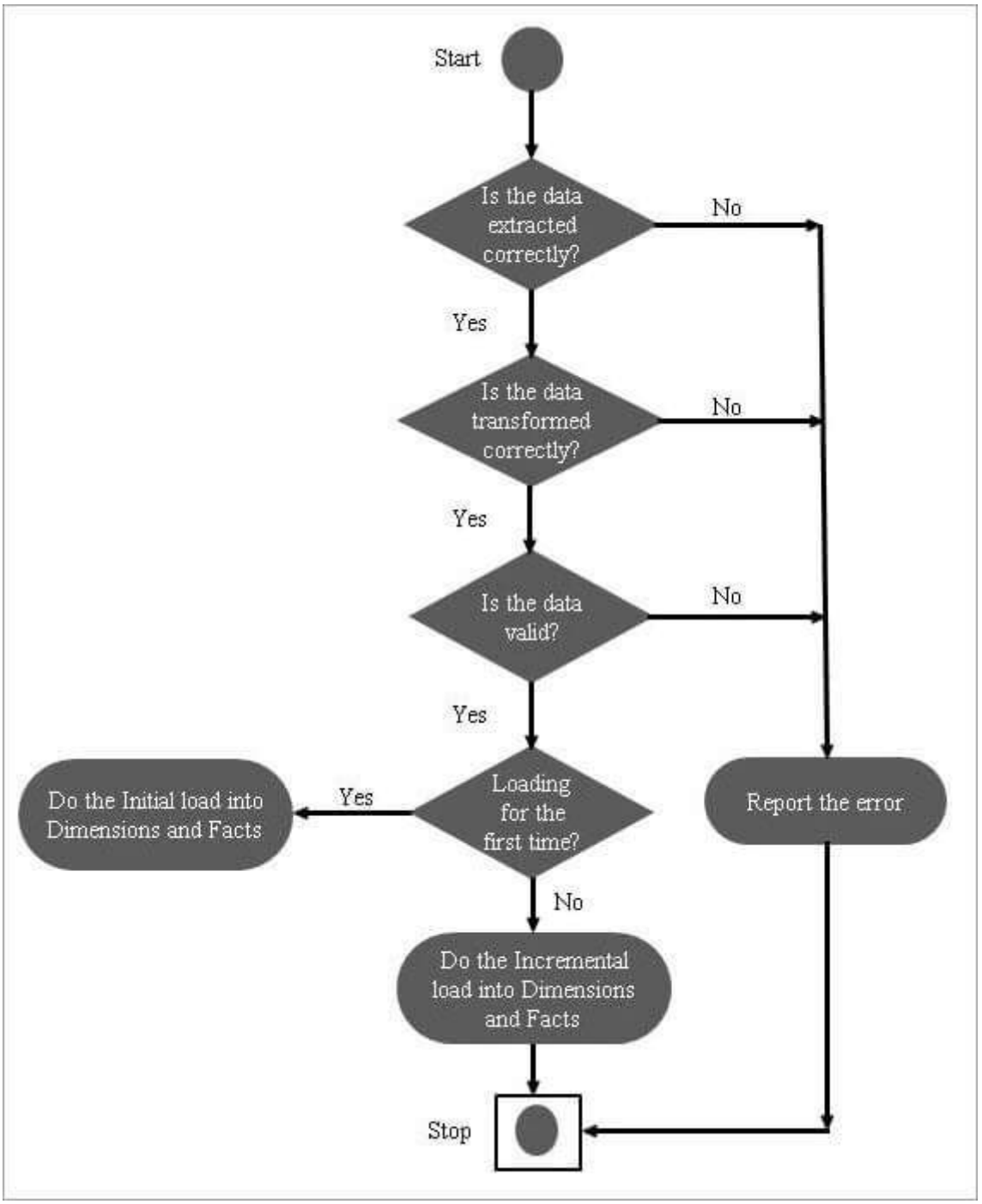


Fuente: https://www.softwaretestinghelp.com/etl-process-in-data-warehouse/#Data_Extraction

Co-funded by the European Union

3.3.3 Flujo de datos: Carga o entrega

La carga de datos en la base de datos de destino del almacén de datos es la fase final del proceso ETL. Grandes cantidades de datos deben cargarse en un almacén de datos típico en un período de tiempo muy corto. Por lo tanto, el procedimiento de carga debe optimizarse para el rendimiento.



Source: https://www.softwaretestinghelp.com/etl-process-in-data-warehouse/#Data_Extraction

Co-funded by the European Union

Para más información, consulta los siguientes enlaces:

ETL Testing

What is ETL?

Understanding ETL BI: 6 Comprehensive Aspects

What Is ETL (Extract, Transform, Load) Process In Data Warehouse?

Implementing an Effective Extract, Transform, Load Process for Your Data Warehouse

3.4. ETL tools

Existen muchas herramientas ETL disponibles en el mercado, cada una con sus propias características y capacidades únicas. Algunas de las herramientas ETL más populares son:

Talend

Talend es una herramienta ETL de código abierto conocida por su facilidad de uso y flexibilidad. Puede extraer datos de una amplia gama de fuentes, incluyendo bases de datos, archivos planos y plataformas de big data, y puede cargarlos en varios sistemas de destino

Pentaho

Pentaho es una herramienta ETL poderosa y flexible que es adecuada para una variedad de proyectos de inteligencia empresarial e integración de datos. Puede extraer datos de una amplia gama de fuentes, incluyendo bases de datos, archivos CSV y plataformas de big data, y puede cargarlos en varios sistemas de destino.

Pentaho también ofrece un conjunto completo de características de BI que

te permiten mejorar el rendimiento y la eficiencia empresarial[3]. Pentaho admite la creación de informes en varios formatos como HTML, Excel, PDF, Texto, CSV y XML, y puede aceptar datos de diferentes fuentes de datos, incluyendo bases de datos SQL, fuentes de datos OLAP e incluso la herramienta de integración de datos Pentaho Data Integration ETL.

"Pentaho es una herramienta de inteligencia empresarial que proporciona una amplia gama de soluciones de inteligencia empresarial a los clientes. Es capaz de realizar informes, análisis de datos, integración de datos, minería de datos, etc.

SQL Server Integration Services (SSIS)

Este es un componente de Microsoft SQL Server que proporciona una plataforma para la integración de datos y la automatización de flujos de trabajo. Es una herramienta poderosa que permite a las organizaciones extraer, transformar y cargar datos de diversas fuentes en una base de datos o almacén de datos de destino. Con SSIS, las organizaciones pueden mejorar la eficiencia de sus procesos de análisis de datos y optimizar sus operaciones de gestión de datos.

La plataforma SQL Server Integration Services (SSIS) *"es una plataforma para diseñar soluciones de integración de datos de alto rendimiento, incluyendo paquetes de extracción, transformación y carga (ETL) para almacenamiento de datos."* Los servicios SSIS recopilan datos de varias fuentes, incluyendo fuentes de datos relacionales, archivos planos, archivos XML, archivos CSV y otras fuentes, modifican los datos según sea necesario y luego los cargan en las ubicaciones respectivas.

Pentaho proporciona capacidades de clase mundial de OLAP (procesamiento analítico en línea), minería de datos, generación de informes y ETL (extracción, transformación y carga). Pentaho es una plataforma de inteligencia empresarial líder que permite a la empresa adquirir datos rápidamente, prepararlos y analizarlos utilizando interfaces

[3] https://www.tutorialspoint.com/pentaho/pentaho_overview.htm

simples y fáciles de usar. SQL Server Integration Services es una plataforma para crear soluciones de integración y transformación de datos a nivel empresarial.

El uso de SSIS puede contribuir a la gestión de objetos y datos de SQL Server, copiar o descargar archivos, cargar almacenes de datos, limpiar y minar datos, y resolver desafíos empresariales complicados.

Informatica
Informatica es una herramienta ETL líder conocida por su potencia y escalabilidad. Puede extraer datos de una amplia gama de fuentes, incluyendo bases de datos, archivos planos y plataformas de big data, y puede cargarlos en varios sistemas de destino.

Apache Nifi
Esta es una herramienta ETL de código abierto diseñada para entornos de big data. Es conocida por su escalabilidad y capacidad para manejar grandes volúmenes de datos en tiempo real.

3.4.1 Por qué elegir SSIS?

La capacidad de SSIS para extraer datos de numerosas fuentes, incluyendo bases de datos, archivos planos, servicios en línea y más, es una de sus principales ventajas. Esto permite a las empresas reunir todos sus datos en un solo lugar para facilitar su uso y análisis. SSIS también ofrece una variedad de componentes de transformación, lo que permite a los usuarios modificar, limpiar y preparar los datos para cargarlos en el destino.

Las organizaciones pueden usar SSIS para automatizar sus procedimientos de integración de datos, lo que reduce la probabilidad de errores y disminuye la necesidad de intervención manual. Esto puede ayudar a las empresas a ahorrar tiempo, aumentar la productividad y reducir costos. Además, SSIS ofrece una interfaz fácil de usar que facilita la creación,

gestión y ejecución de paquetes SSIS.

SSIS puede ayudarte a extraer más valor de tus datos al centralizarlos, limpiarlos y convertirlos, y automatizar el proceso de integración de datos, independientemente de si estás trabajando con un conjunto de datos pequeño o un almacén de datos masivo.

Características y beneficios

El método para la extracción, transformación y carga de datos en almacenes de datos es el objetivo principal de los paquetes de SQL Server Integration Services (SSIS).

Aquí hay algunas características y beneficios adicionales de SQL Server Integration Services (SSIS) que podrían considerarse para una empresa:

1. Entorno de desarrollo visual: SSIS incluye un entorno de desarrollo visual que te permite diseñar, construir y desplegar soluciones de integración de datos sin escribir ningún código.
2. Soporte de plataforma: SSIS puede ejecutarse en una variedad de plataformas, incluyendo Windows, Linux y contenedores Docker, lo que lo convierte en una plataforma de integración de datos flexible y portátil
3. Wrangling de datos: SSIS incluye capacidades de wrangling de datos que te permiten extraer, transformar y cargar (ETL) datos de una manera visual e intuitiva, utilizando una interfaz de arrastrar y soltar.
4. Integración en la nube: SSIS puede conectarse a una variedad de fuentes de datos basadas en la nube, incluyendo Azure SQL Database y Azure Data Lake, lo que facilita la integración de datos desde la nube.
5. Lineage de datos: SSIS incluye capacidades de lineage de datos que te permiten rastrear el flujo de datos desde la fuente hasta el destino y entender cómo se transforman los datos en el camino.
6. Servicios de calidad de datos: SSIS incluye Data Quality Services (DQS) para la limpieza y coincidencia de datos, así como la Tarea de Perfilado de Datos para entender la calidad y estructura de tus datos.

7. Mejora de la productividad: SSIS puede ayudar a mejorar la productividad de los desarrolladores de integración de datos al proporcionar un entorno de desarrollo visual, un conjunto rico de componentes preconstruidos y un diseño flexible que permite la personalización y extensibilidad.

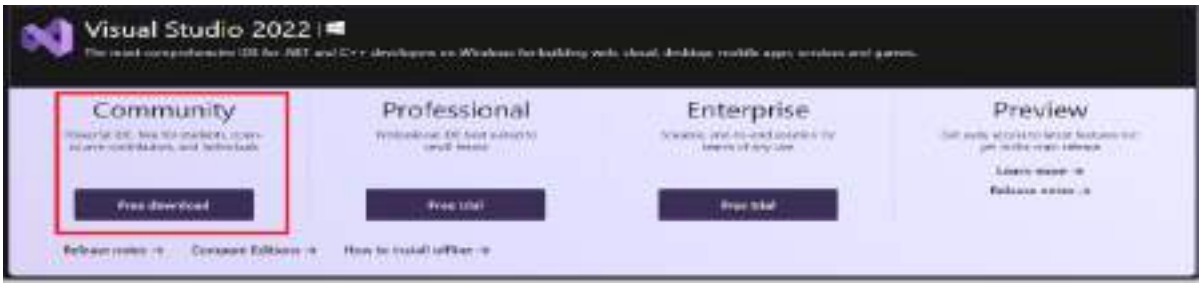
3.4.2 Guías de instalación

Un proyecto SSIS puede crearse en un entorno de desarrollo visual gracias a la extensión SSIS para Visual Studio. Visual Studio de Microsoft es un entorno de desarrollo integrado (IDE) que se utiliza para desarrollar programas informáticos, incluidos sitios web, aplicaciones web, servicios web y aplicaciones móviles.

Necesitas instalar la extensión SSIS en Visual Studio 2022 para desplegarla en proyectos. Hay un instalador independiente de SQL Server Data Tools (SSDT) con el entorno de programación SSIS disponible para versiones anteriores de Visual Studio. En lugar de ver código, verás cajas que representan tareas y conexiones entre ellas.

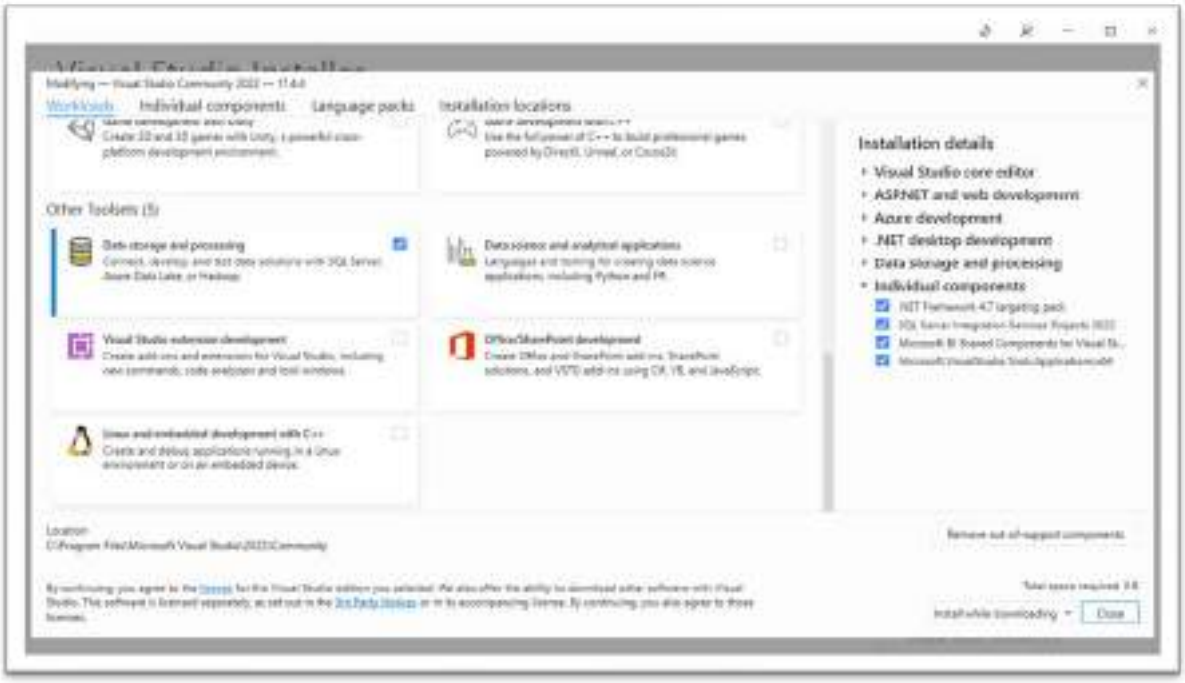
3.4.2.1 Manual de instalación de Visual Studio 2022

Para descargar Visual Studio 2022, utilizaremos el siguiente enlace. Para nuestro ejemplo, descargaremos la edición gratuita Community de este producto:



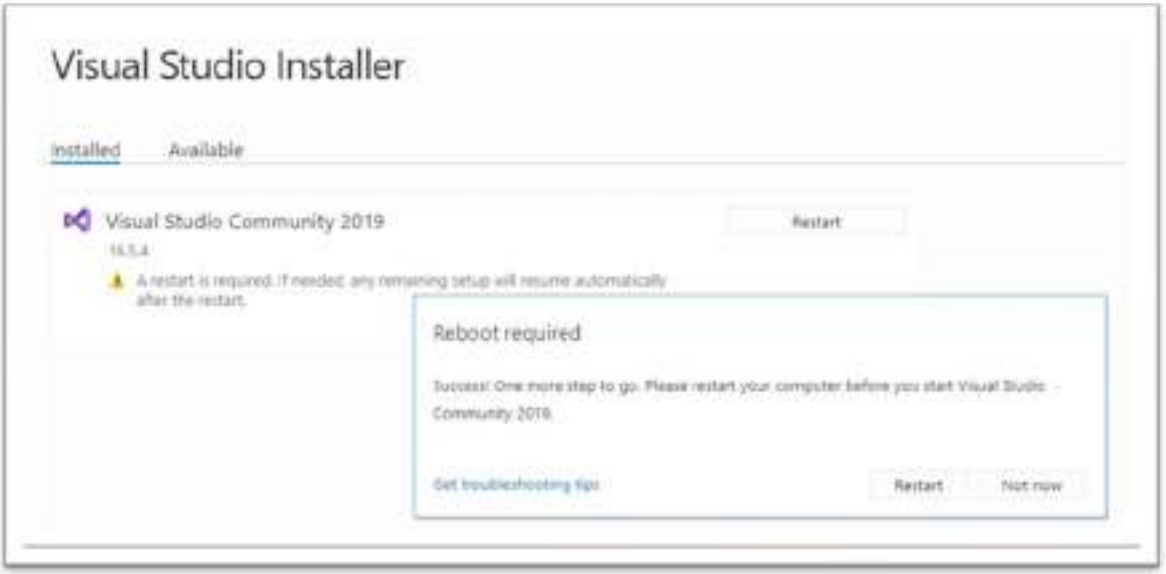
Luego, haremos clic en ""Free download". ". Después de descargar el

archivo .exe, haremos doble clic en él para iniciar el proceso.
El instalador de Visual Studio se inicia y, después de un tiempo, se abre la siguiente ventana:

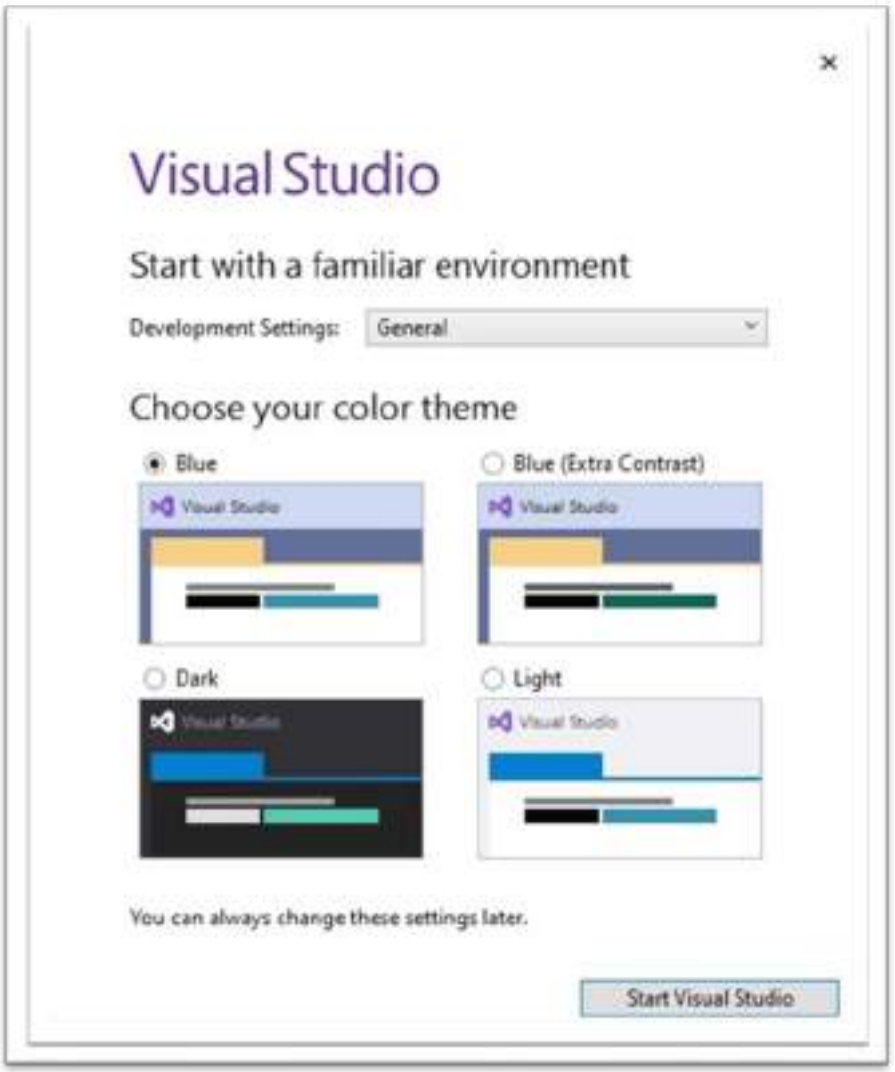


Para agregar SQL Server Data Tools (SSDT), desplazamos hacia abajo y seleccionamos **"Almacenamiento y procesamiento de datos".** Luego, presionamos el botón **"Instalar"** y esperamos a que se complete la instalación.

Después de la instalación, se requiere reiniciar, por lo que presionamos **"Reiniciar".**



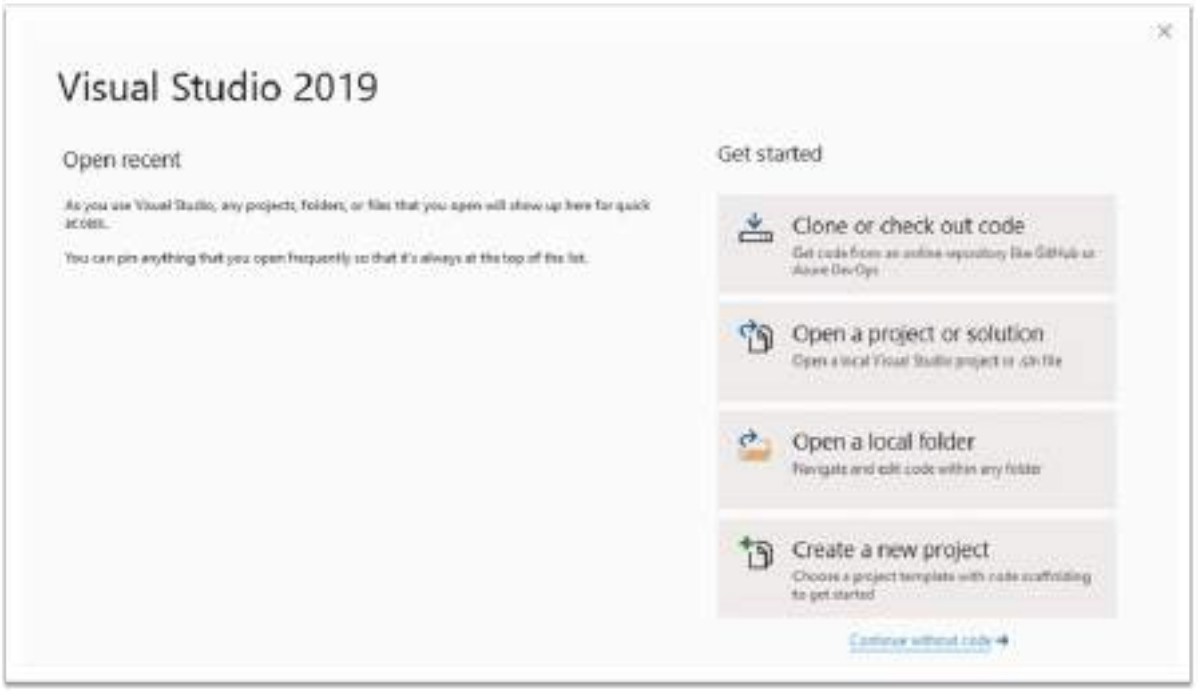
Después de reiniciar nuestra computadora, iniciamos Visual Studio. En la siguiente pantalla, para mantener nuestro ejemplo simple, simplemente hacemos clic en **"No ahora, tal vez más tarde"** en lugar de iniciar sesión:
Después de eso, elegimos el tema y hacemos clic en **"Iniciar Visual Studio":**



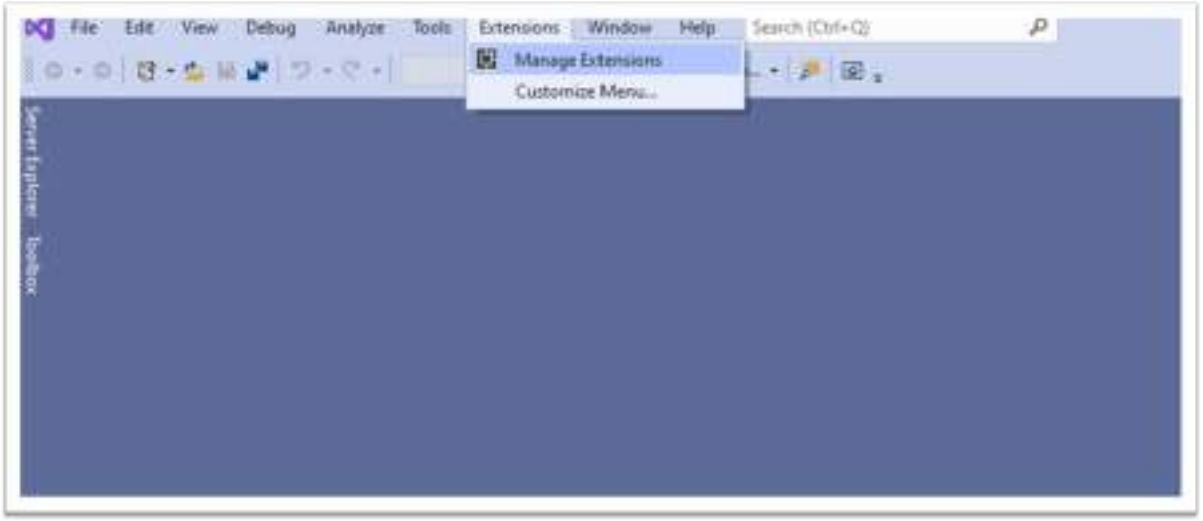
Bueno, hemos instalado Microsoft Visual Studio 2022 en nuestra computadora. Ahora, es momento de agregar la extensión para desarrollar proyectos de Integration Services.

3.4.2.2 Manual de instalación de la extensión SSIS

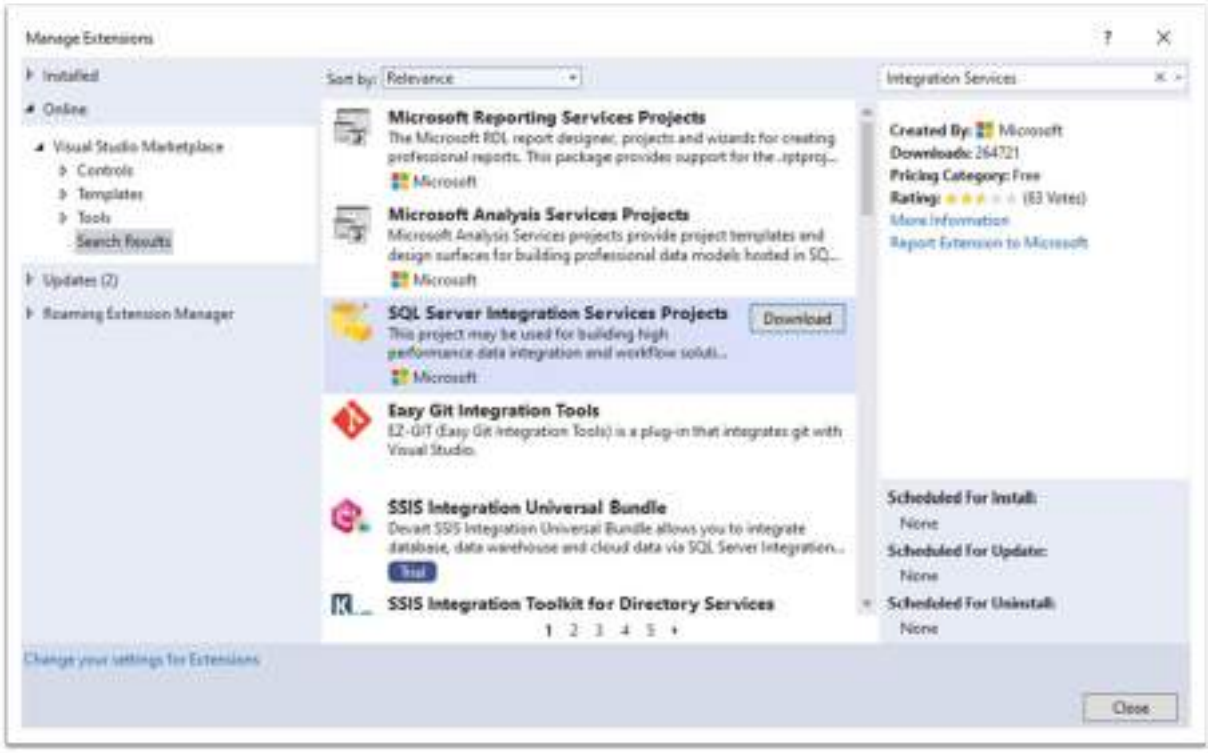
Cuando se abre Visual Studio, hacemos clic en "Continuar sin código" para agregar la extensión necesaria:



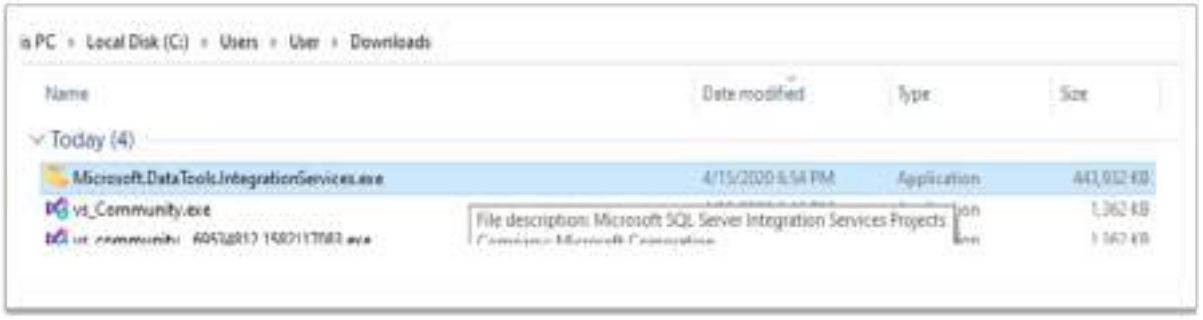
En esta ventana, hacemos clic en **"Extensions" > "Manage Extensions":**



En la barra de búsqueda de la ventana abierta, escribimos **"Integration Services"** para localizar fácilmente la extensión. De la lista que aparece, elegimos **"SQL Server Integration Services Projects" y** presionamos **"Descargar":**



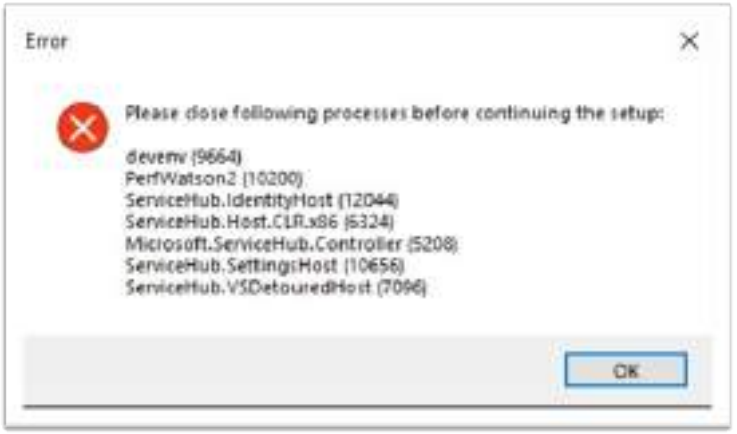
Luego, ejecutaremos el archivo descargado **.exe file:**



La instalación de la extensión ha comenzado. Ahora, seguiremos algunos pasos simples. En la siguiente ventana, hacemos clic en **"OK":**

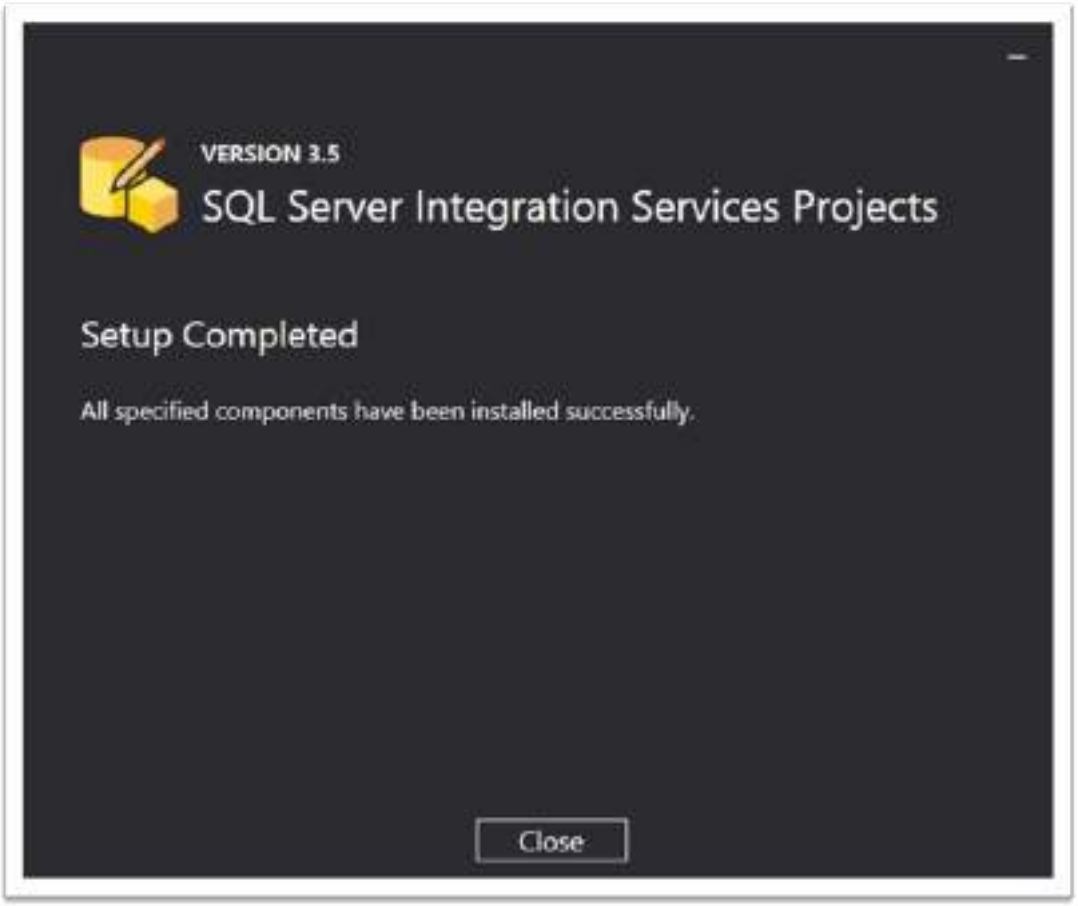
D

Si recibes el siguiente mensaje, probablemente tengas SQL Server Management Studio abierto. Cierra la aplicación e inténtalo de nuevo.



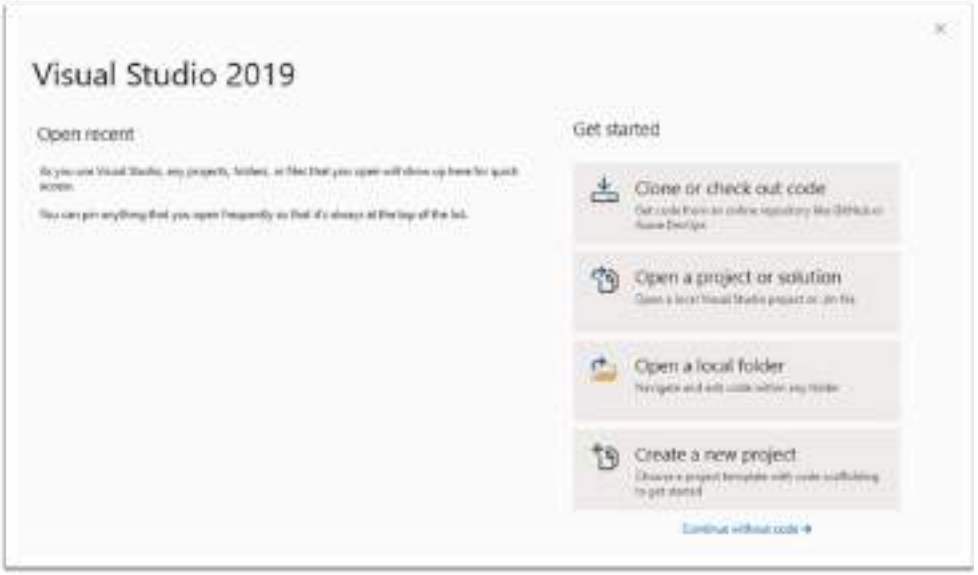
Ciérralo y haz clic en **"OK".** El proceso debería continuar.

Finalmente, la configuración está completa y tenemos nuestra extensión instalada.

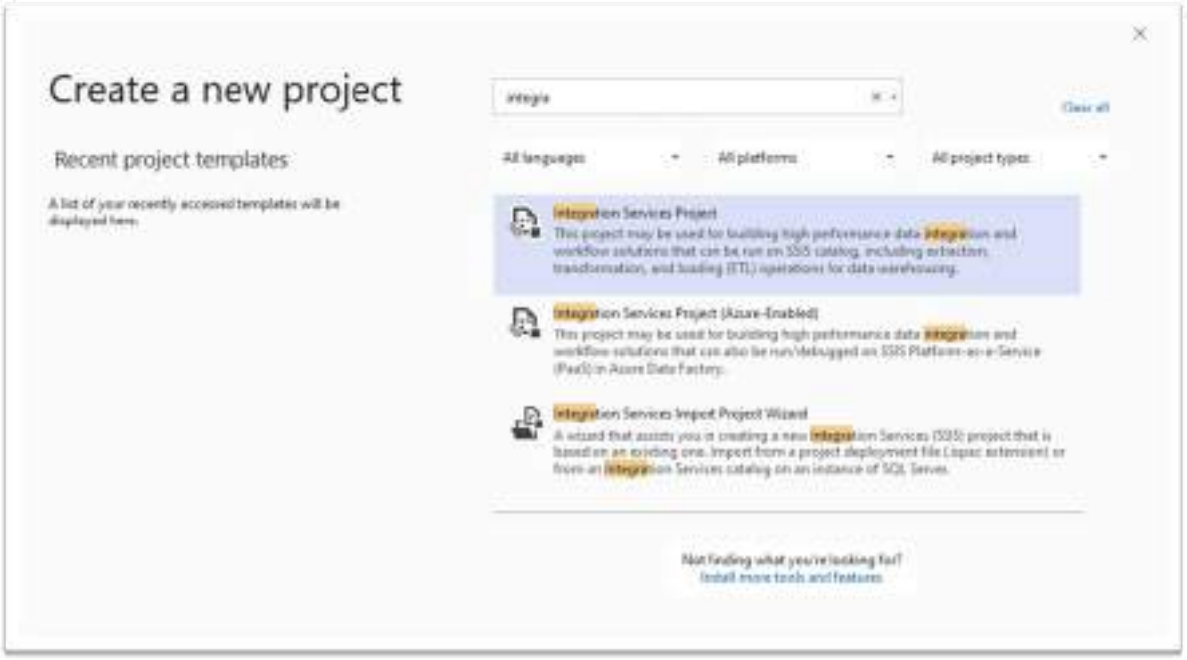


3.4.3 Creación de un proyecto SSIS

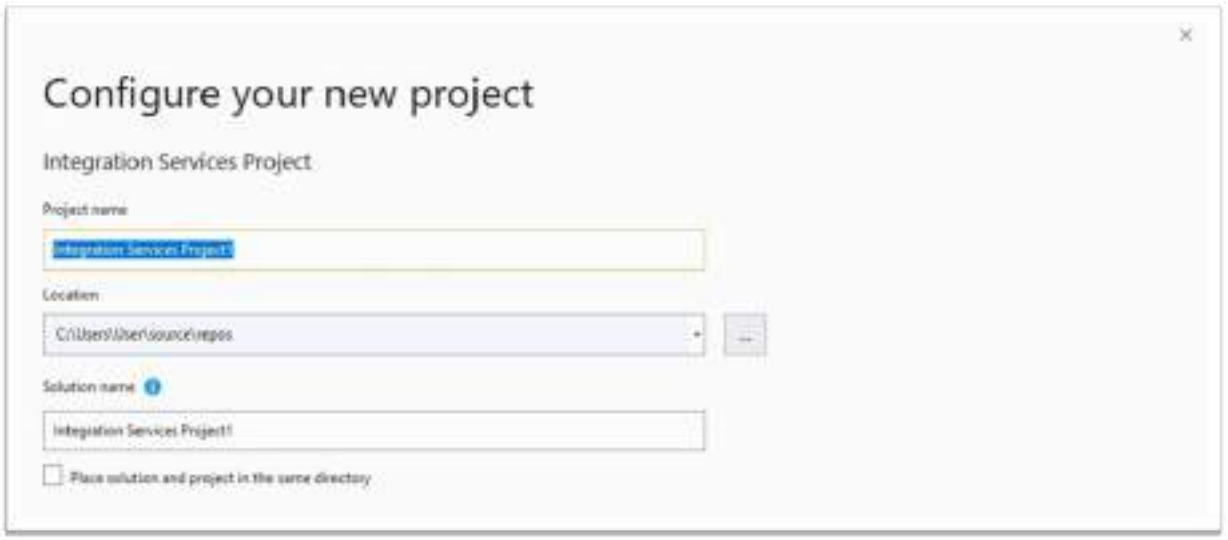
Ahora, estamos listos para crear proyectos de Integration Services. En Visual Studio, elegimos **"Crear un nuevo proyecto":**



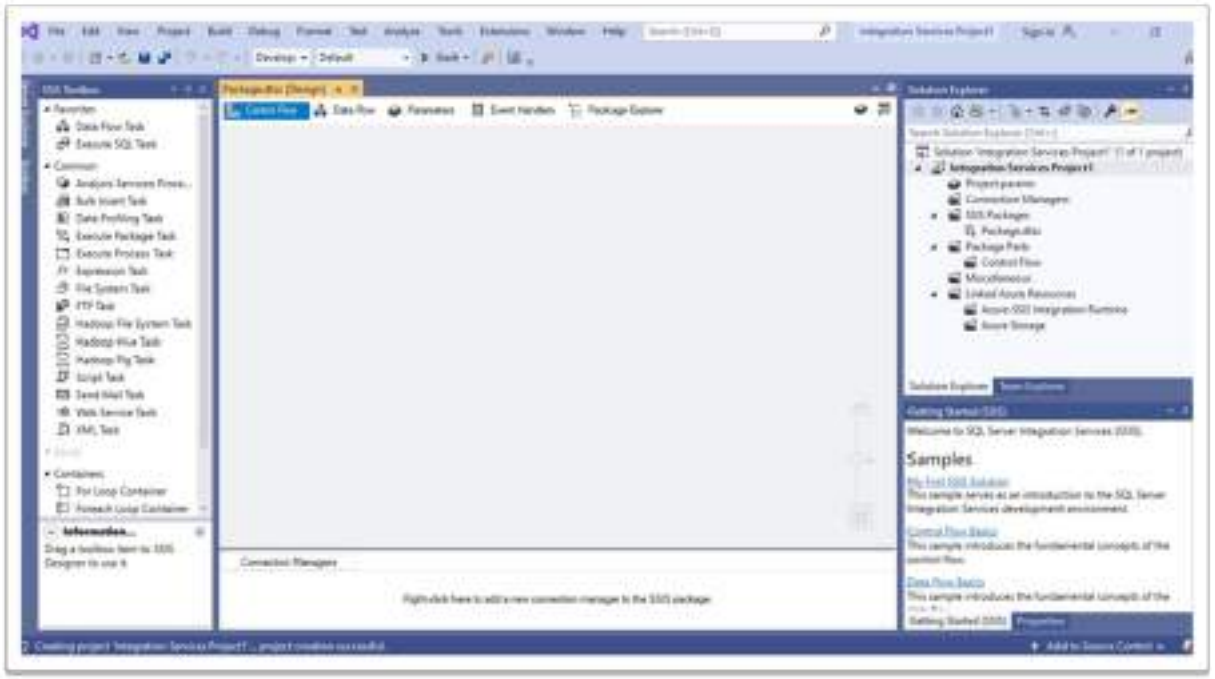
En la siguiente ventana, escribimos "integration" para encontrar **"Proyecto de Integration Services"** y hacemos clic en él:



Elegimos un nombre para nuestro proyecto:



Y así, está listo! Abrimos la interfaz donde podemos diseñar y desarrollar paquetes SSIS 2022



En el fototutorial anterior, revisamos las actualizaciones relacionadas con la herramienta para el desarrollo de proyectos de SQL Server Integration Services en Visual Studio 2022. Como hemos visto, en esta versión de Visual Studio, la herramienta para diseñar proyectos SSIS se instala como una

extensión de Visual Studio. También hemos explorado cómo instalar Visual Studio 2022 y cómo agregar la extensión de Integration Services Projects.

Visión general de la consola SSIS

La consola SSIS, una interfaz gráfica de usuario (GUI) que permite a los desarrolladores y administradores gestionar y monitorear paquetes SSIS, es uno de los componentes principales de SSIS. La interfaz principal en SQL Server Integration Services (SSIS) para gestionar y supervisar proyectos y paquetes de integración de datos es la consola SSIS.

Ofrece a los usuarios una vista detallada de sus proyectos y paquetes SSIS, con una vista de árbol de los componentes principales, una representación gráfica del diseño del paquete, información completa sobre la ejecución del paquete, registros generados durante la ejecución y estadísticas históricas de ejecución. La visión general de la consola facilita a los usuarios moverse entre proyectos, encontrar información crucial rápidamente y gestionar y seguir eficientemente sus operaciones de integración de datos.

La consola SSIS ofrece varias características y funcionalidades clave, incluyendo:

1. **Gestión de paquetes:** La consola SSIS permite a los desarrolladores y administradores crear, editar y gestionar paquetes SSIS. Proporciona una interfaz fácil de usar para crear y modificar paquetes SSIS, lo que facilita el trabajo con tareas complejas de integración de datos.
2. **Ejecución de paquetes**: La consola SSIS proporciona la capacidad de ejecutar paquetes SSIS bajo demanda o en un horario establecido. Permite a los desarrolladores y administradores ejecutar paquetes SSIS en una variedad de entornos, incluyendo entornos locales, remotos y basados en la nube.
3. **Monitoreo de paquetes:** La consola SSIS proporciona monitoreo y

registro en tiempo real de la ejecución de paquetes SSIS. Permite a los desarrolladores y administradores monitorear el progreso de los paquetes SSIS, ver estadísticas de ejecución y solucionar cualquier problema que pueda surgir durante la ejecución del paquete.

4. **Despliegue de paquetes:** La consola SSIS permite a los desarrolladores y administradores desplegar paquetes SSIS en diferentes entornos. Proporciona la capacidad de crear manifiestos de despliegue y desplegar paquetes en diferentes entornos, como entornos de desarrollo, prueba y producción. Esto facilita la gestión y el mantenimiento de los paquetes SSIS en diferentes entornos.
5. **Seguridad:** La consola SSIS proporciona una variedad de características de seguridad que se pueden utilizar para asegurar paquetes SSIS y datos. Permite a los desarrolladores y administradores asegurar paquetes y datos utilizando encriptación y protocolos seguros, así como configurar controles de acceso de usuarios y permisos.
6. **Informes:** La consola SSIS proporciona la capacidad de generar informes sobre la ejecución de paquetes, el estado de los paquetes y otras métricas de rendimiento. Estos informes pueden ser utilizados para obtener información sobre el rendimiento de los paquetes SSIS y para identificar cualquier problema que necesite ser abordado.

Perspectivas de Trabajo

SQL Server Integration Services (SSIS) es un robusto programa de integración de datos y flujos de trabajo que permite a las empresas gestionar y transformar datos de manera efectiva. Como resultado, al emplear SSIS, se pueden considerar diversas perspectivas de trabajo.

En primer lugar, SSIS es una herramienta útil para el almacenamiento e integración de datos. Permite a las empresas recopilar, modificar y cargar datos de diversas fuentes. SSIS también ofrece la capacidad de realizar transformaciones de datos sofisticadas, incluyendo normalización de datos,

filtrado de datos y limpieza de datos.

Además, SSIS es una herramienta fantástica para automatizar tareas de integración de datos. Esto permite a las empresas automatizar el movimiento y la transformación de datos, lo que puede reducir significativamente el tiempo y el esfuerzo necesarios para gestionar datos.

SSIS es una herramienta efectiva para la generación de informes e inteligencia empresarial. Las organizaciones pueden usar esto para comprender sus datos y tomar decisiones basadas en datos.

La migración de datos, o el proceso de trasladar datos de un sistema a otro, es abordada por SSIS como una solución útil. En las empresas, esta es una tarea común, especialmente cuando se actualizan o reemplazan sistemas.

En conclusión, SSIS proporciona una amplia variedad de perspectivas de trabajo que pueden ser utilizadas para optimizar la velocidad y la precisión de las operaciones para el almacenamiento de datos, integración de datos, automatización, inteligencia empresarial y transferencia de datos. Las organizaciones pueden beneficiarse enormemente de estas perspectivas de trabajo al mejorar la disponibilidad de datos, tomar decisiones más informadas y utilizar los recursos de manera más eficiente.

Perspectivas de Diseño para "Integration Services"

SQL Server Integration Services (SSIS) ofrece una variedad de actividades integradas para administrar una instancia de SQL Server además de ser un producto ETL. Aunque la arquitectura interna de SSIS ha sido diseñada para ofrecer un alto grado de paralelismo y eficiencia, aún existen ciertas prácticas recomendadas para maximizar aún más el rendimiento. A continuación se enumeran algunas de ellas que es bueno considerar al trabajar con SSIS:

I. **Escalabilidad:** Para asegurar que un sistema SSIS pueda manejar grandes volúmenes de datos mientras mantiene el rendimiento a medida que aumenta el volumen de datos, se debe tener en cuenta la escalabilidad al construir la solución. Esto puede implicar adoptar métodos de carga de alto rendimiento, como la inserción en bloque, o diseñar el sistema para que pueda expandirse a través de numerosos servidores.
II. **Reutilización:** Hacer que los paquetes y componentes SSIS sean reutilizables en muchas circunstancias es generalmente beneficioso. Esto puede acortar el proceso de desarrollo y hacer que la solución sea más fácil de mantener.
III. **Calidad de los datos:** El éxito de cualquier solución de integración de datos depende de la confiabilidad e integridad de tus datos. Considera cómo manejarás la limpieza de datos y los controles de calidad de datos al crear un sistema SSIS para asegurar que los datos sean precisos y consistentes.
IV. **Manejo de errores:** Es crucial pensar en cómo manejarás los errores y excepciones que podrían ocurrir cuando se ejecute un paquete SSIS. Esto podría implicar la creación de estrategias para manejar errores y reiniciar procesos fallidos, así como configurar el manejo de errores y el registro para rastrear y solucionar problemas.
V. **Seguridad:** Al crear un sistema SSIS, ten en cuenta la seguridad de los datos y la seguridad de la solución. Es posible que sea necesario encriptar datos sensibles, establecer conexiones seguras con las fuentes y destinos de datos, y poner en práctica controles de seguridad para evitar el acceso no autorizado.
VI. **Mantenibilidad:** Al crear un sistema SSIS, considera cómo será soportado y mantenido en el futuro. Esto puede implicar hacer que la solución sea modular y adaptable también.

3.4.4 Construcción de transformaciones y trabajos en un proyecto SSIS

Integration Services (SSIS) es una plataforma para construir soluciones de integración de datos y transformaciones a nivel empresarial. Usa Integration Services para resolver problemas comerciales complejos copiando o descargando archivos, cargando almacenes de datos, limpiando y minando datos, y gestionando objetos y datos de SQL Server.

Transformaciones: Los datos se modifican y alteran a medida que se toman de una o más fuentes y se importan en un sistema de destino utilizando los componentes principales de transformaciones de SSIS. La limpieza de datos, normalización, enriquecimiento y mapeo son algunos ejemplos de las transformaciones que se pueden aplicar a los datos a medida que se mueven a través de una tubería de datos.

Trabajos: La ejecución de paquetes SSIS se gestiona y automatiza utilizando trabajos. La ejecución de muchos paquetes puede gestionarse de manera coordinada mediante el uso de trabajos, que son colecciones de uno o más paquetes que se pueden ejecutar como una sola unidad. Los trabajos ofrecen un mecanismo para planificar, ejecutar y monitorear la ejecución de paquetes, así como para manejar el registro y el manejo de errores.

Mientras que las tareas se usan para automatizar y controlar la ejecución de paquetes SSIS, las transformaciones se usan para editar y modificar datos cuando se toman de una o más fuentes y se cargan en un sistema de destino. Estas ideas trabajan juntas para crear una plataforma sólida y adaptable para la integración de datos y la transformación de datos, y son partes esenciales de cualquier arquitectura de datos.

Para más información, consulta los siguientes enlaces:

Data Conversion Transformation

SQL Server Integration Services

How to Download and Install SQL Server for Windows

SSIS How to Create an ETL Package

Top 14 ETL Tools for 2023

A List of The 16 Best ETL Tools And Why To Choose Them

SSIS Tutorial

What is the difference between Pentaho and Microsoft SQL Server Integration Services?

Pentaho Data Integration Tutorial: What is, Pentaho ETL Tool

Pentaho - Overview

3.5. Mis primeras transformaciones

Cuando trabajes en tus primeras conversiones ETL, es fundamental comprender el procedimiento y las prácticas recomendadas. Una comprensión completa del proceso puede ayudarte a asegurar que tus operaciones ETL sean eficientes, precisas y confiables. Las transformaciones ETL son un paso crítico en el proceso de gestión y transformación de datos.

Comprender los datos con los que trabajarás es el primer paso en el proceso ETL. Esto incluye determinar la estructura, el formato y la calidad de los datos, así como localizar cualquier dato faltante o redundante. También es crucial identificar cualquier dato que esté en el formato o tipo de dato incorrecto.

Es fundamental definir los requisitos de los datos de salida. Esto incluye definir la estructura, el formato y el contenido ideales de los datos de salida. La siguiente etapa es extraer datos de varias fuentes. Esto se puede lograr utilizando muchas herramientas y técnicas, como consultas SQL, importaciones de archivos planos y llamadas a servicios web.

Después de que los datos han sido recopilados, el siguiente paso es transformarlos en el formato deseado. Esto incluye tareas como la limpieza y validación de datos, la conversión de tipos de datos y el filtrado de datos.

Durante el proceso de transformación, los datos se filtran, normalizan y limpian. Este procedimiento puede incluir la realización de cálculos o agregaciones y la eliminación de cualquier dato innecesario. Los principales objetivos del proceso de transformación son la precisión de los datos y la preparación para la carga en el sistema de destino. Los datos pueden cargarse en el sistema de destino. Esto puede implicar la importación de datos en un archivo plano, una base de datos u otra ubicación.

Es vital verificar la información después de que se ha cargado. Esto incluye comparar los datos con la fuente original, así como con las especificaciones establecidas. Esta etapa asegura que los datos sean correctos y que el procedimiento ETL se haya completado con éxito.

Es importante monitorear y gestionar los datos para asegurar que sean correctos y estén actualizados. Establecer operaciones ETL rutinarias, resolver problemas e implementar actualizaciones según sea necesario, son todas partes de esto. Este proceso asegura que los datos permanezcan precisos a lo largo del tiempo y que cualquier problema sea identificado y resuelto rápidamente.

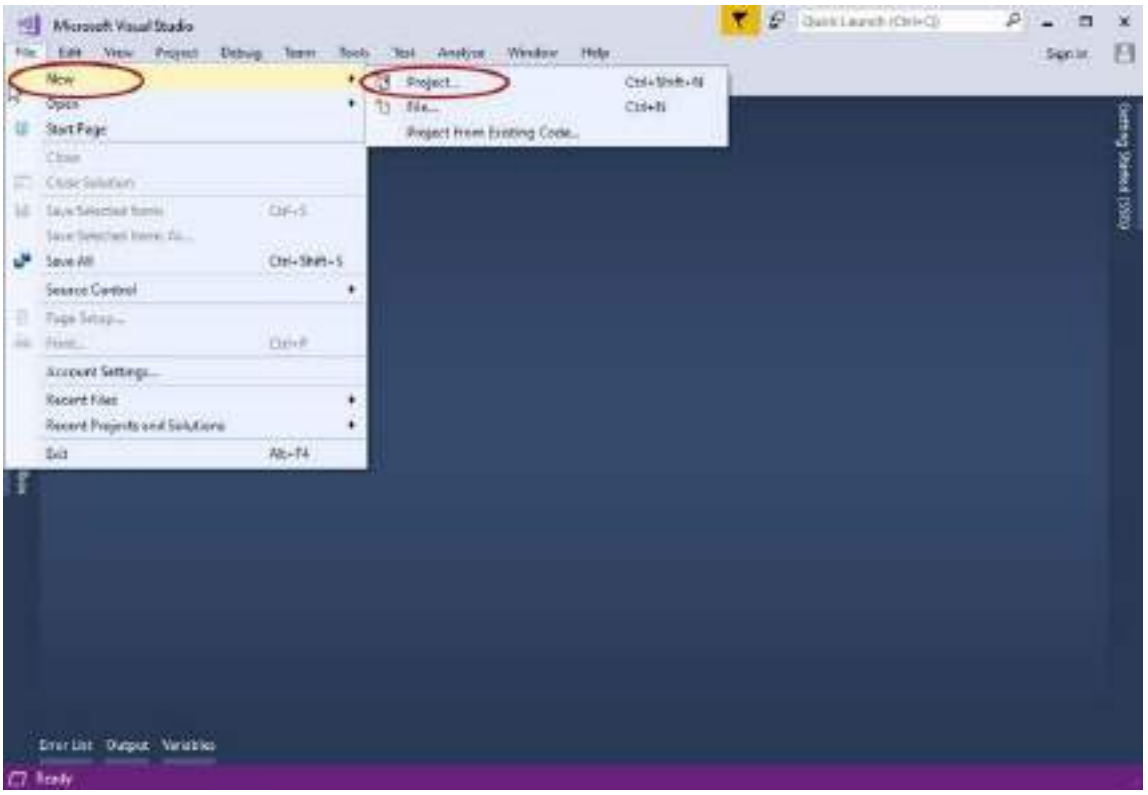
3.5.1. Automatización de la reconstrucción de índices en SQL Server: Desde la creación de paquetes SSIS hasta la ejecución

programada

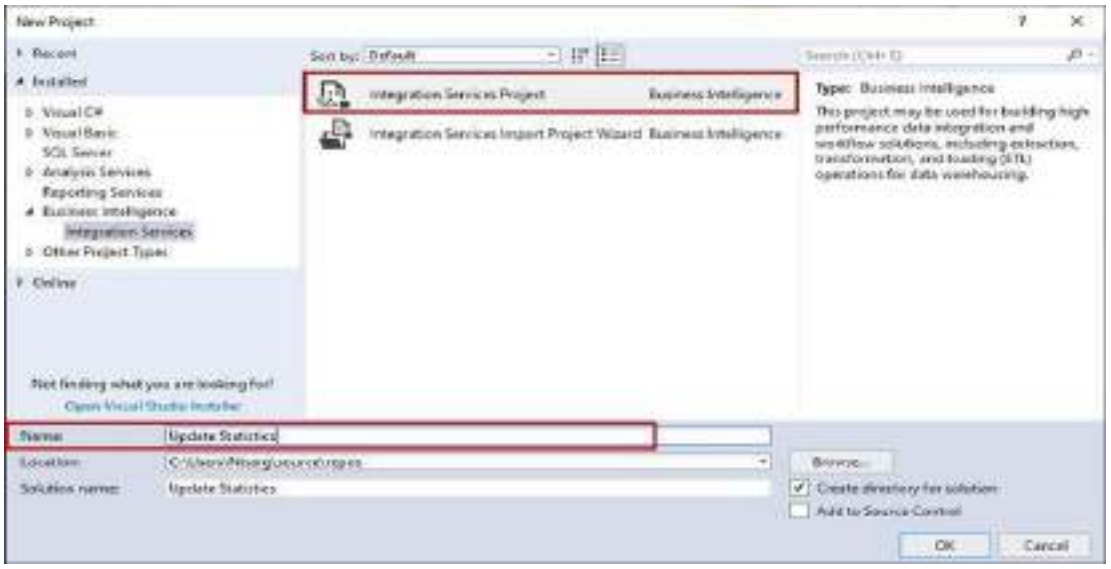
Para demostrar el proceso, realizaremos todos los siguientes trabajos en la práctica:

1. Crear un paquete SSIS que reconstruya todos los índices de las tablas de la base de datos del usuario.
2. Desplegar el paquete SSIS en el catálogo de Integration Services.
3. Crear un trabajo del Agente de SQL Server para automatizar la ejecución del paquete SSIS.

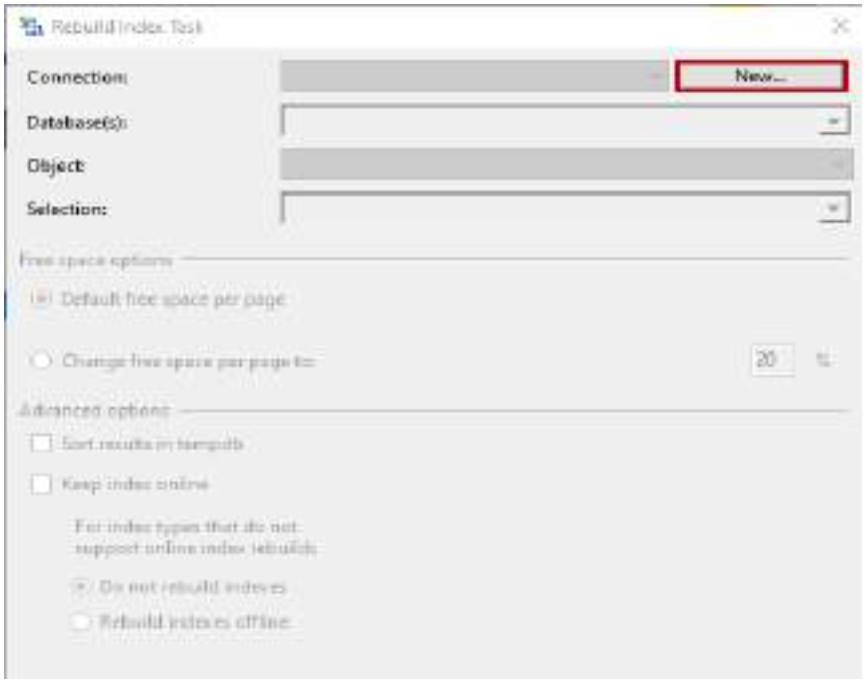
Podemos crear el paquete de servicios de integración usando **SSDT para Visual Studio 2022. Puedes descargarlo aquí.**
Una vez que esté descargado e instalado, crea un nuevo proyecto SSIS siguiendo los pasos aquí. Las directrices para instalar SSIS están disponibles aquí.
Open **SQL Server Data Tools > File > New > Project**



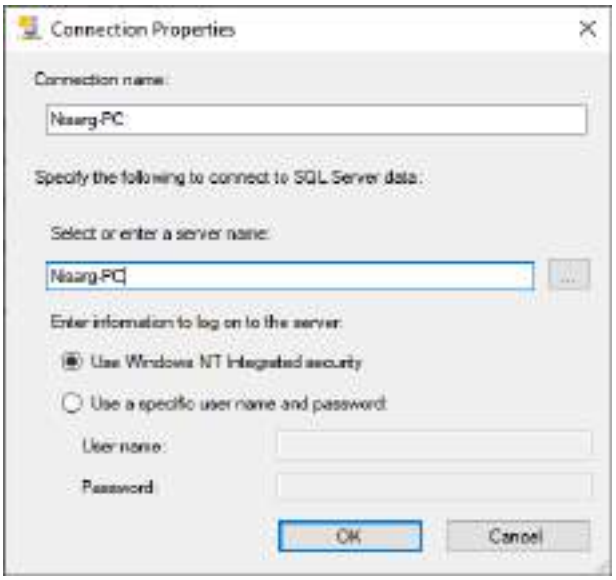
En el cuadro de **diálogo Nuevo Proyecto,** selecciona **Proyecto de Integration Services** y nombra el proyecto:



Aquí necesitamos configurar la tarea de reconstrucción de índices. Haz doble clic en ella y se abrirá la ventana de diálogo **Tarea de Reconstrucción de Índices**. Primero, haz clic en **Nuevo** para crear una nueva conexión al servidor.



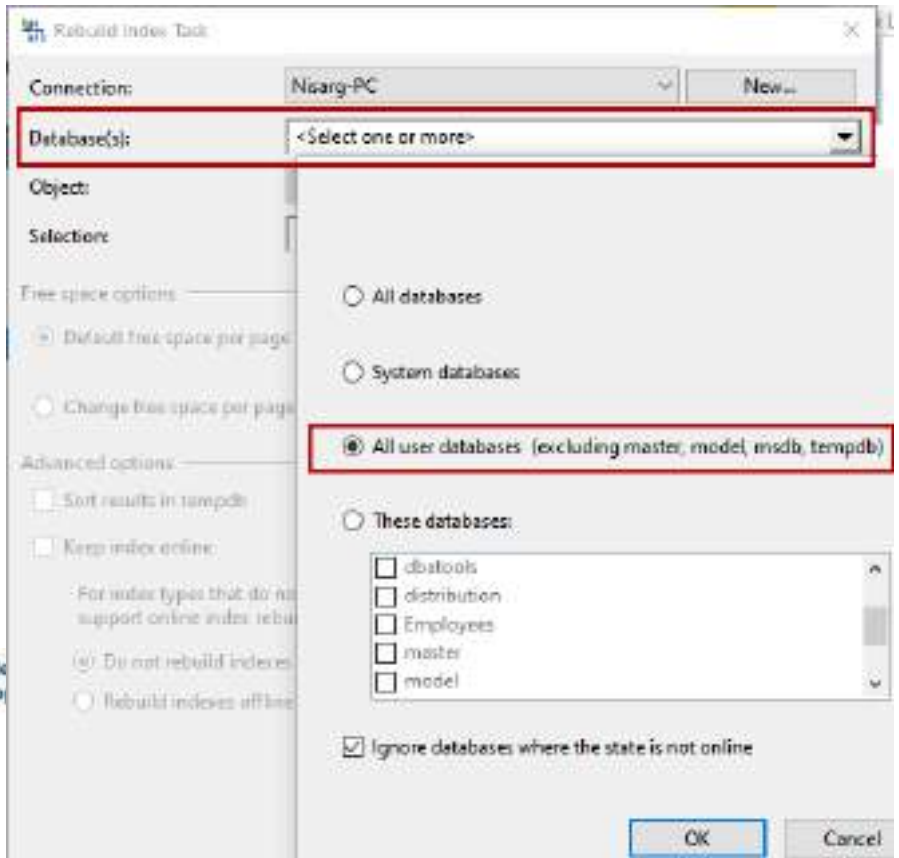
Se abrirá la ventana de diálogo Propiedades de Conexión. Especifica el nombre de la conexión, el nombre del servidor y el método de autenticidad para iniciar sesión en el servidor.



Haz clic en OK para guardar la configuración y cerrar la ventana. Una vez establecida la conexión, la lista de bases de datos se llenará en el cuadro desplegable de **Base de datos**.

En esta demostración, el paquete debe reconstruir el índice de todas las bases de datos. Haz clic en Base de datos y selecciona **Todas las bases de datos** de usuario del menú desplegable. Queremos excluir las bases de datos que no están en línea. Por lo tanto, **marca la casilla Ignorar bases de datos cuyo estado no esté en línea** en la misma ventana.

Después de elegir las bases de datos, tendrás acceso a la configuración.



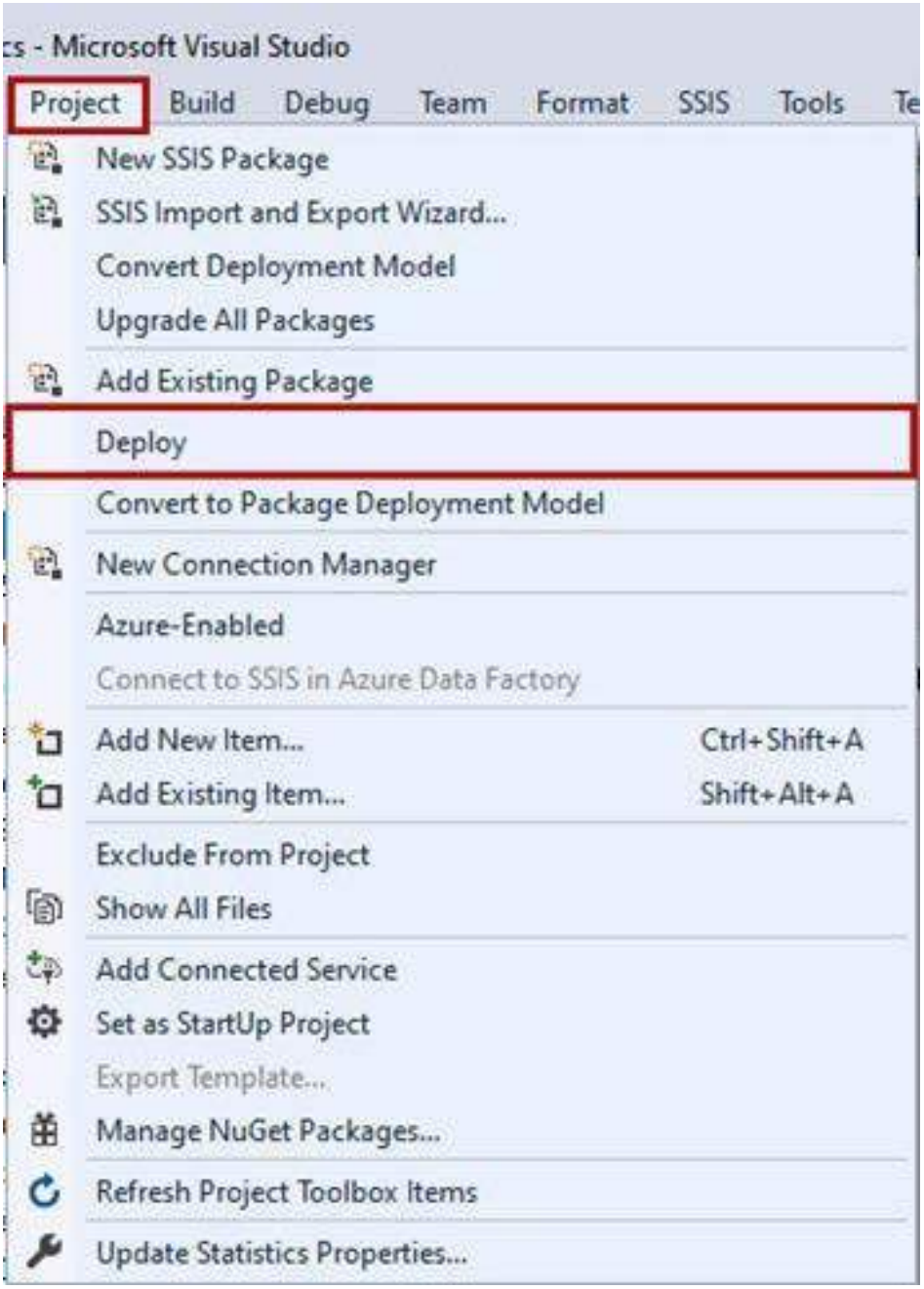
Co-funde
the Euro

Después de elegir las bases de datos, tendrás acceso a las opciones de configuración. Las opciones predeterminadas son razonables y adecuadas para nuestros objetivos, pero puedes configurar ajustes adicionales según tu caso:

1. Opciones de espacio libre:
   - **Espacio libre predeterminado por página:** elimina y recrea el índice de la tabla con el factor de llenado (FILL) predeterminado.
   - **Cambiar el espacio libre por página a:** elimina y recrea el índice de la tabla con el factor de llenado especificado por el usuario.
2. **Opciones avanzadas:**
   - Ordenar TempDB: Ordenar TempDB puede ser un paso importante de ajuste de rendimiento para entornos de SQL Server que hacen un uso intensivo de tablas temporales y otros objetos. TempDB es una base de datos del sistema utilizada para almacenar datos temporales, como variables de tabla, tablas temporales y otros objetos que son creados por SQL Server durante el procesamiento de consultas.
   - Mantener el índice en línea durante la operación de reconstrucción del índice. (Compatible con la edición Enterprise)
   - Establecer el valor de MAXDOP para ejecutar la reconstrucción del índice en múltiples hilos.
3. **Puedes optimizar la operación de reconstrucción del índice configurando las siguientes opciones:**
   - Reconstruir el índice solo si la fragmentación (la condición donde los datos se almacenan en ubicaciones no contiguas o dispersas en un dispositivo de almacenamiento, en lugar de en una manera secuencial y organizada) es mayor que el valor especificado en el cuadro de texto **fragmentation> textbox.**
   - Reconstruir el índice solo si el recuento de páginas es mayor que el valor especificado en el cuadro de texto **Page Count > textbox value.**
   - Reconstruir el índice solo si se ha utilizado antes del número de días especificado en el cuadro de texto **Used in the last textbox**.

Ten en cuenta que puedes seleccionar una de estas opciones o combinar varias de ellas. Haz clic en **OK** para guardar la configuración.

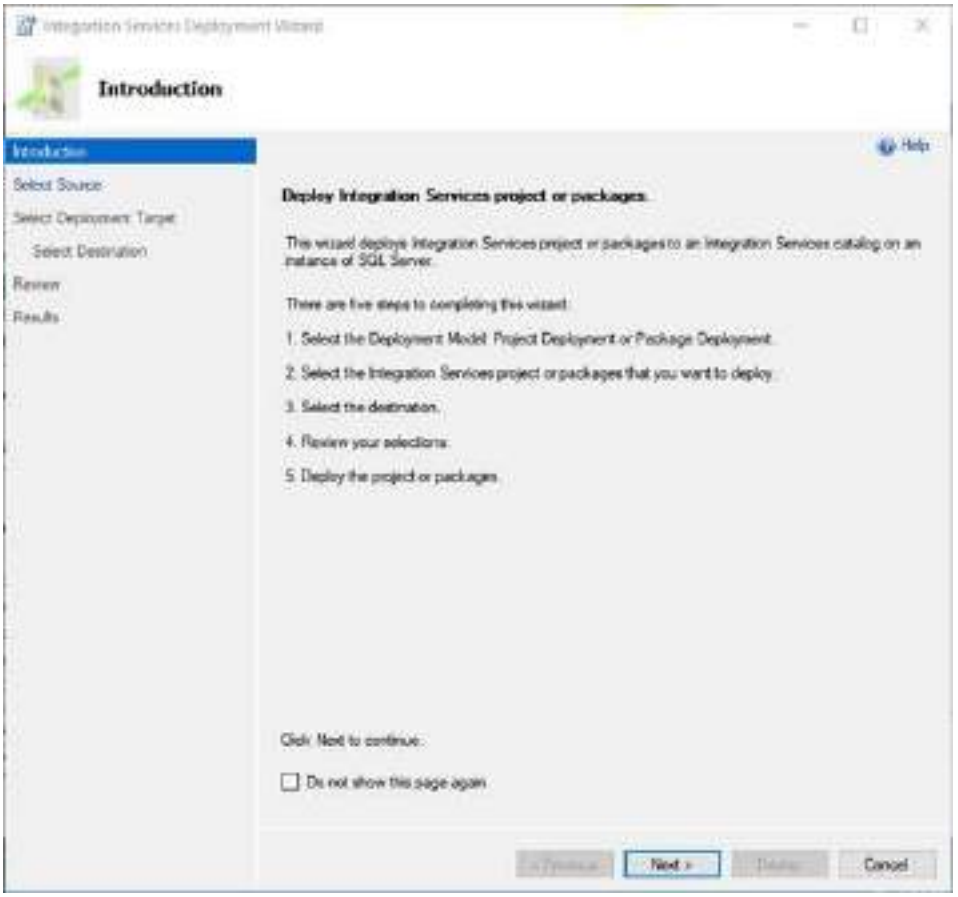
Ahora, vamos a desplegar el paquete SSIS. Coloca el cursor sobre **Project > Deploy:**



El Integration Services Deployment Wizard se inicia.

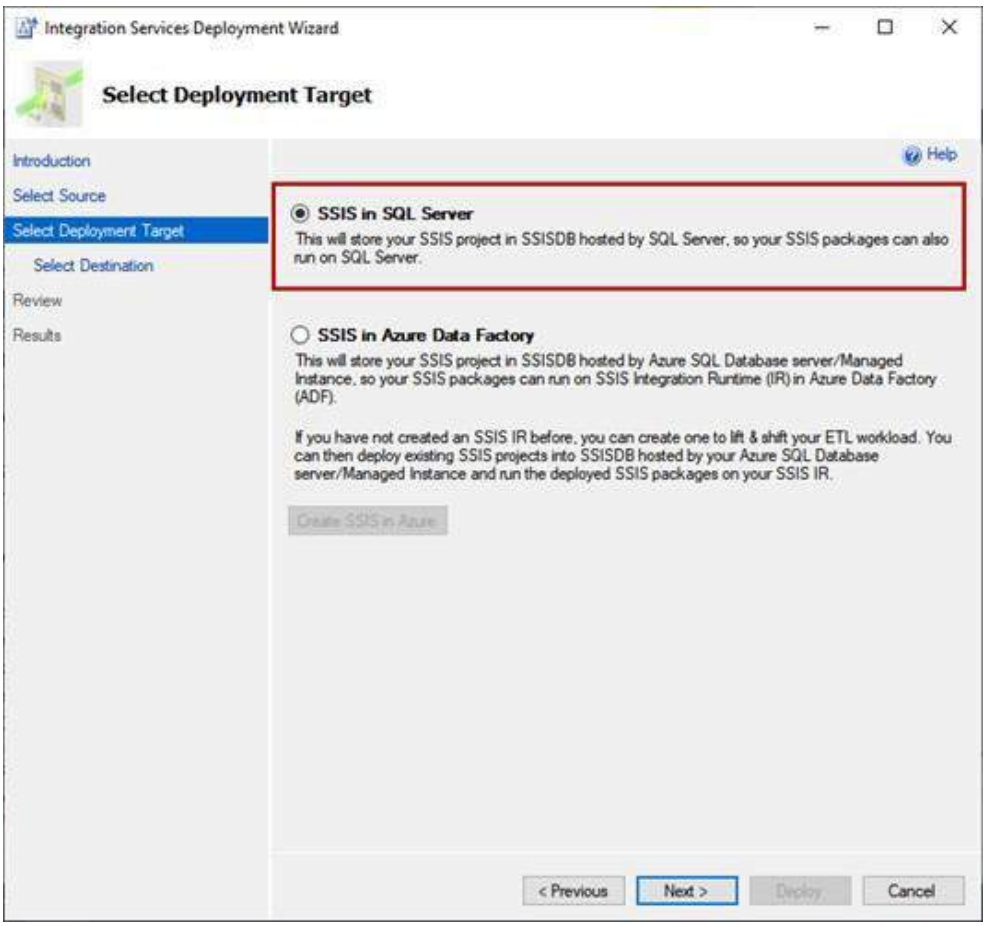
En la primera pantalla, puedes ver la lista de tareas que el asistente realizará. Puedes omitirla: marca la casilla **No mostrar esta página nuevamente** en la parte inferior de la ventana y procede a **Siguiente.**

Co-funded by
the European Union



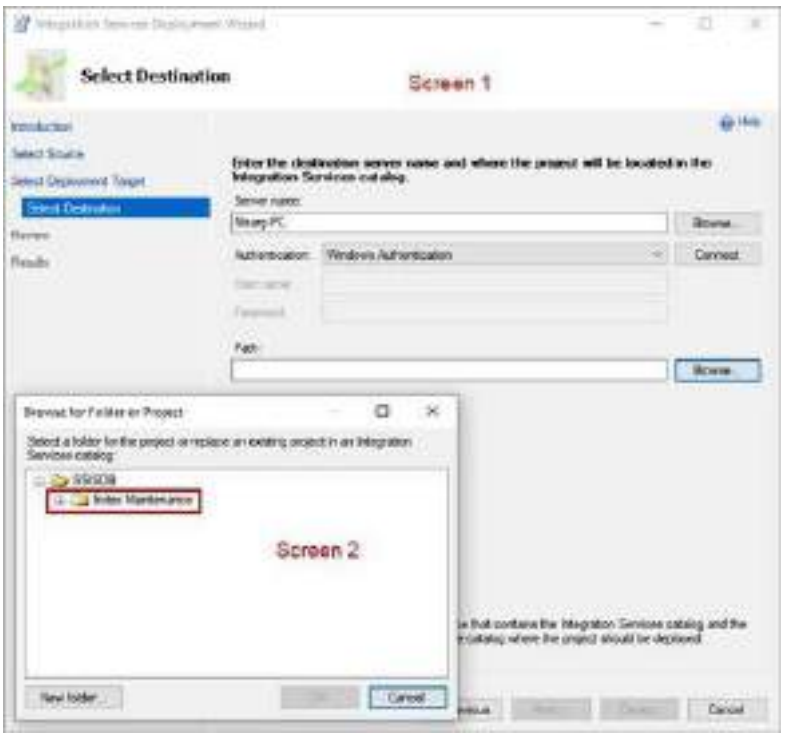
En la pantalla **Seleccionar destino de despliegue**, especifica el destino.

Puedes elegir entre los servicios de integración de SQL Server alojados en el servidor local o la opción de servicios de integración en Azure Data Factory. Hemos instalado los servicios de integración en un SQL Server local. Por lo tanto, selecciona **SSIS en SQL Server.** Haz clic en **Siguiente.**



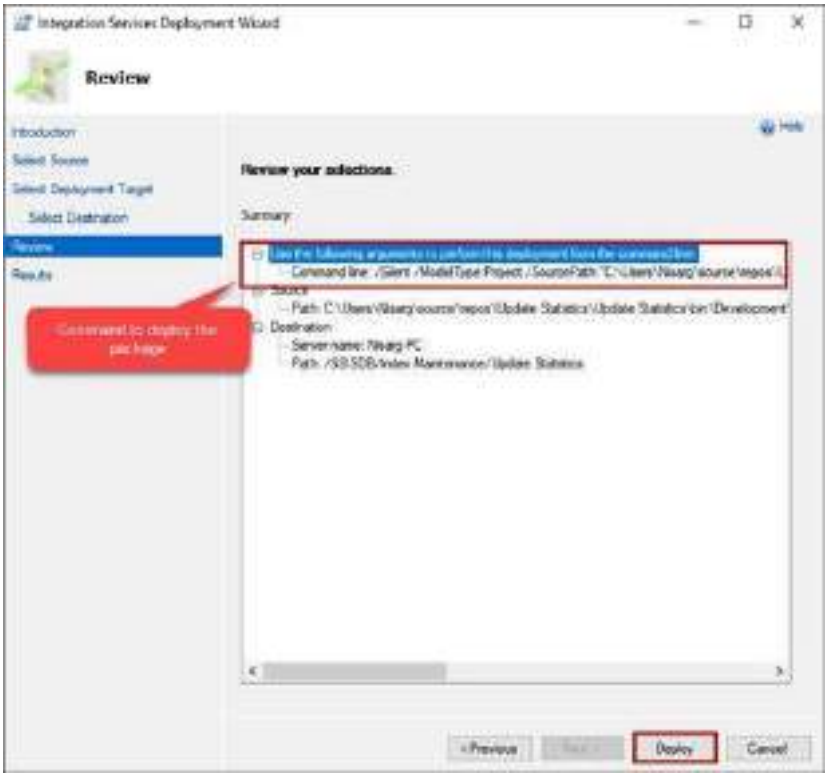
Especifica el nombre del servidor donde has instalado los servicios de integración de SQL Server y proporciona las credenciales de conexión **(Pantalla 1**). Ten en cuenta que en esta etapa debes especificar la ruta al proyecto de servicios de integración en el catálogo SSISDB.

Haz clic en **Examinar (Pantalla 1)** para ubicar el proyecto en el catálogo SSISDB y navega hasta la carpeta SSISDB **(Pantalla 2).** Haz clic en **OK.**

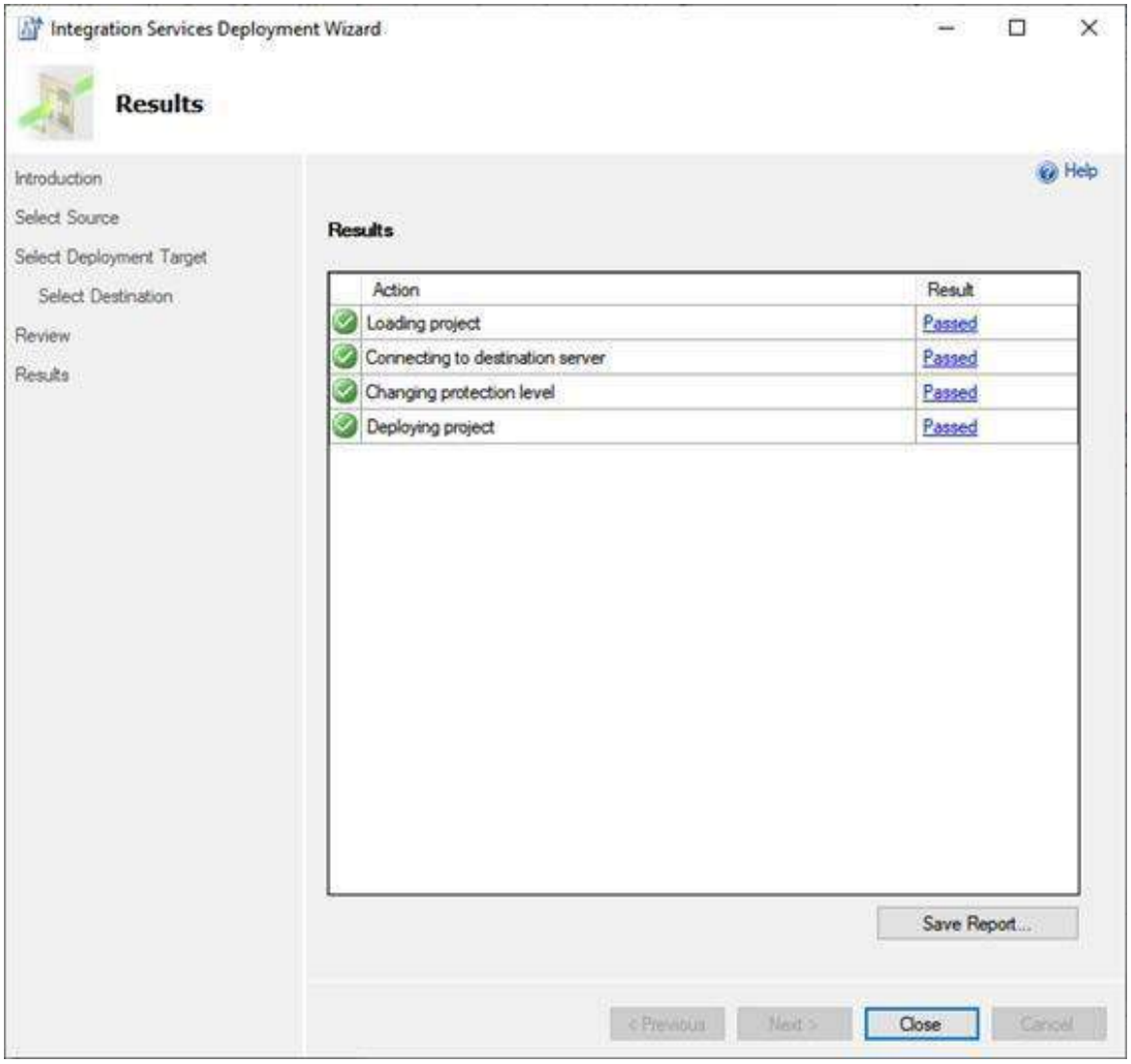


Haz clic en **Siguiente** y procede a la ventana de **Revisión**.

Aquí, puedes verificar los detalles de la fuente y el destino. Después de eso, haz clic en Desplegar.



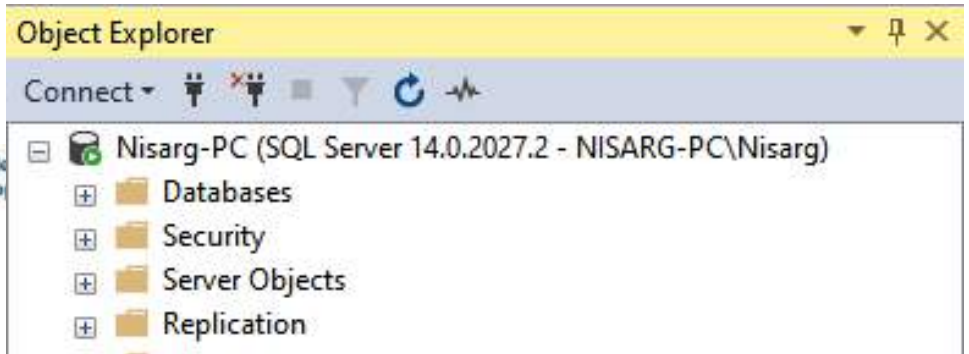
Una vez que el proyecto se despliegue con éxito, haz clic en **Cerrar** para completar la tarea en el Integration Services Deployment Wizard.



Puedes ver el paquete en la carpeta del catálogo de servicios de integración de SQL Server Management Studio.

Abre **SQL Server Management Studio** > Conéctate al **motor de la base de datos** > expande la instancia de la base de datos > Expande **Catálogos de Integration Services**.

Luego, **expande SSISDB > Mantenimiento de índices**.

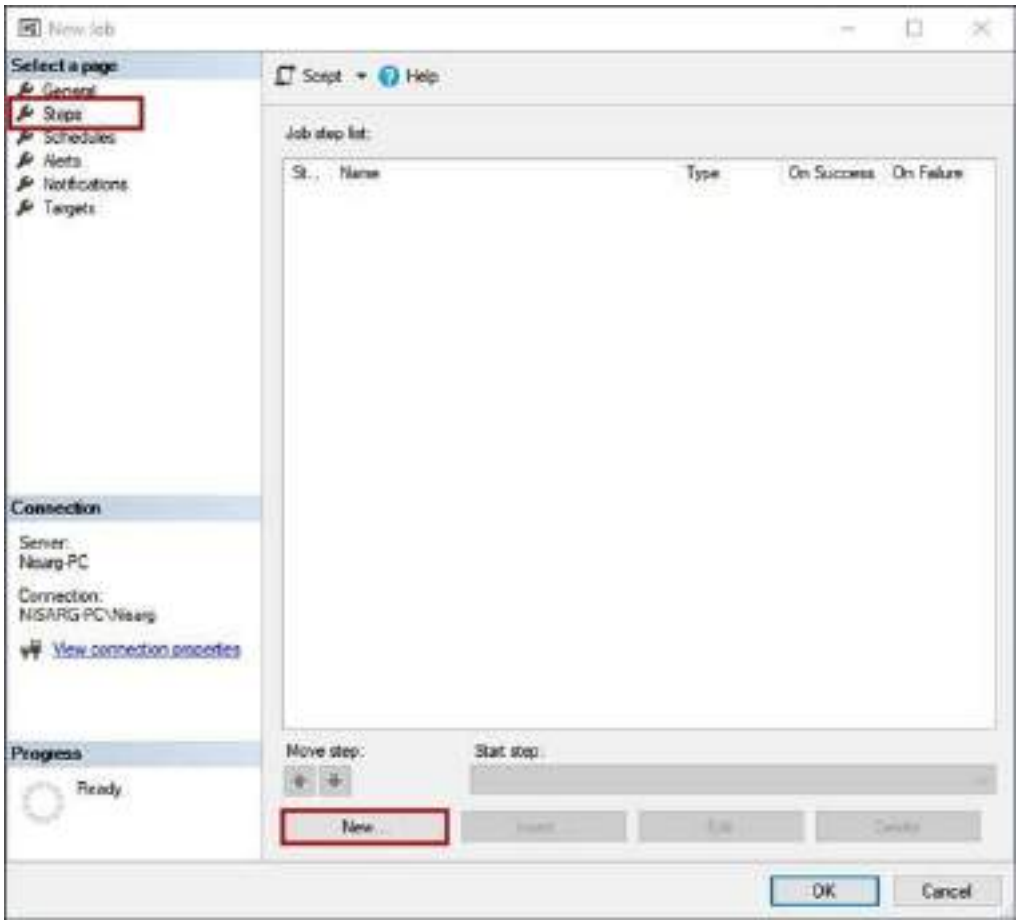


La programación de la ejecución del **paquete SSIS requiere un trabajo del Agente de SQL** Server.

Expande el **Agente de SQL Server** y haz clic derecho en **Trabajos.** Selecciona **Nuevo Trabajo.**

En el cuadro de diálogo **Nuevo Trabajo**, ingresa el nombre deseado en el campo **Nombre.** Haz clic en **Pasos** para agregar el paso del trabajo.

☐ Haz clic en **Nuevo.**

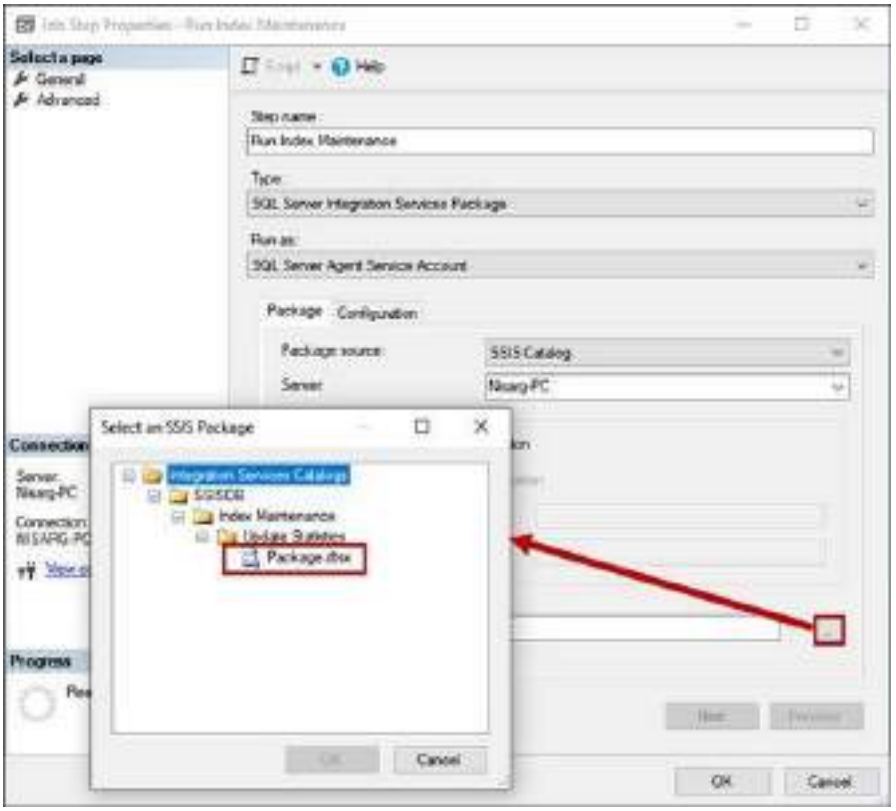


Llegarás a la ventana de propiedades del **paso del trabajo.**

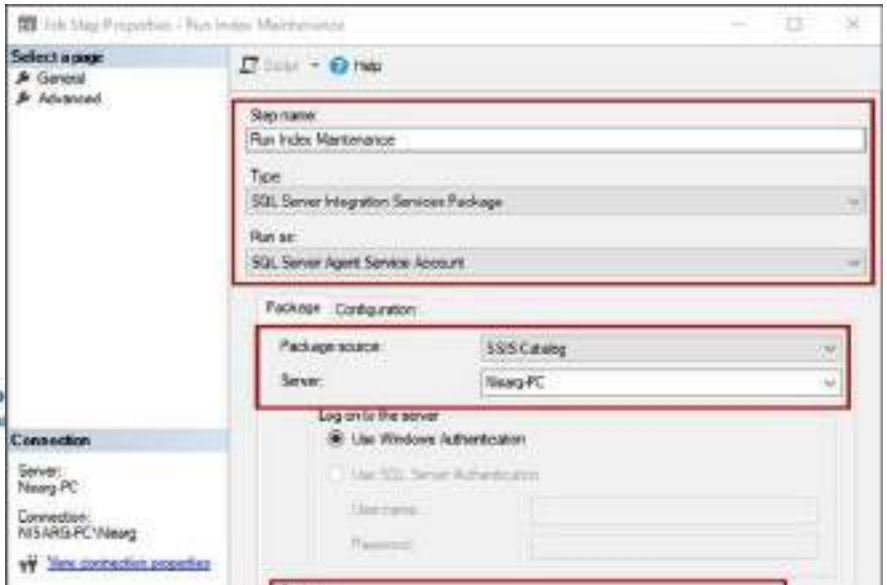
- Proporciona un nombre adecuado para el paso del trabajo y selecciona **Paquete de Integration Services de SQL Server** en el cuadro desplegable Tipo.

- Selecciona **Cuenta de servicio del Agente de SQL Server** en el menú desplegable **Ejecutar como.** Seleccionamos esto porque el paquete SSIS debe ejecutarse a través de la cuenta de servicio del Agente de SQL Server.
- Selecciona **Catálogo SSIS** en el menú desplegable **Fuente del paquete**, ya que hemos desplegado el paquete en el catálogo SSIS.
- Ingresa el **nombre del host del** servidor en el que hemos instalado el servicio de integración.

Ten en cuenta que debemos ingresar la ubicación del paquete en el catálogo SSISDB. Para hacerlo, haz clic en el botón [...] (ver la captura de pantalla a continuación). Se abrirá un cuadro de diálogo **Seleccionar un paquete SSIS.** Allí, selecciona un paquete SSIS adecuado y haz clic en **OK.**



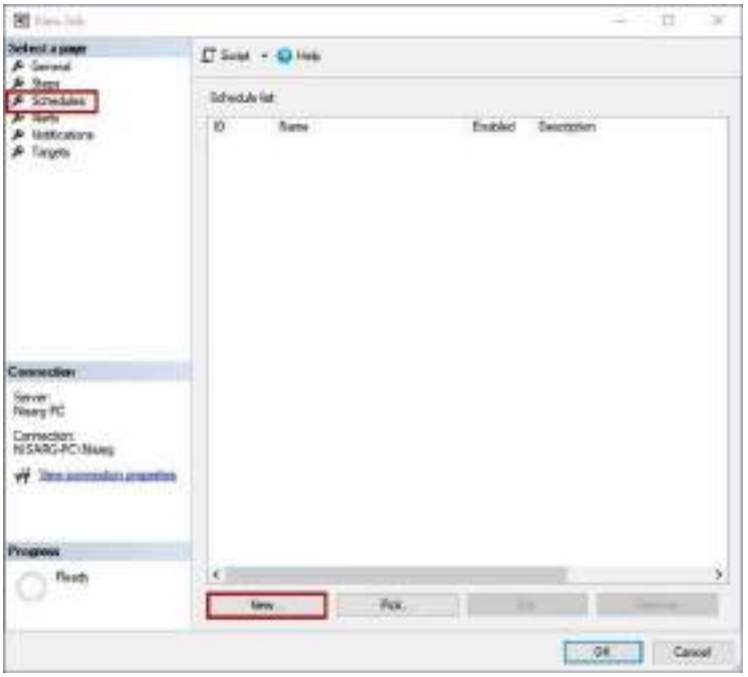
Cuando la configuración esté completa, haz clic en **OK** para guardar el paso de trabajo.



Co-funded b
the Europea

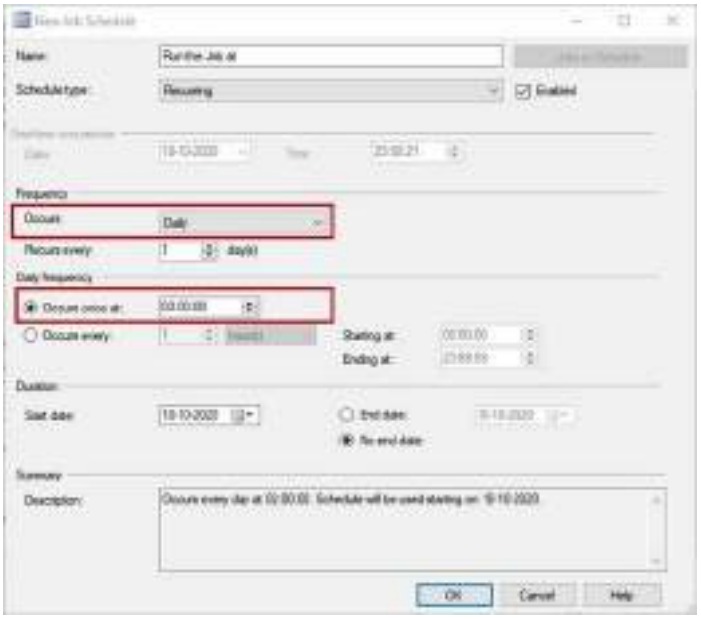
El trabajo debe ejecutarse diariamente a la 1:00 AM.

Para configurar el horario, haz clic en **Horarios > Nuevo**



En el cuadro de diálogo **Nuevo Programa de Trabajo** que aparece, ingresa el nombre del horario deseado en el cuadro de texto **Nombre.**

Selecciona **Diario** en el menú desplegable **Ocurrido** y escribe **02:00:00** en el cuadro de texto **Ocurrirá una vez a:**

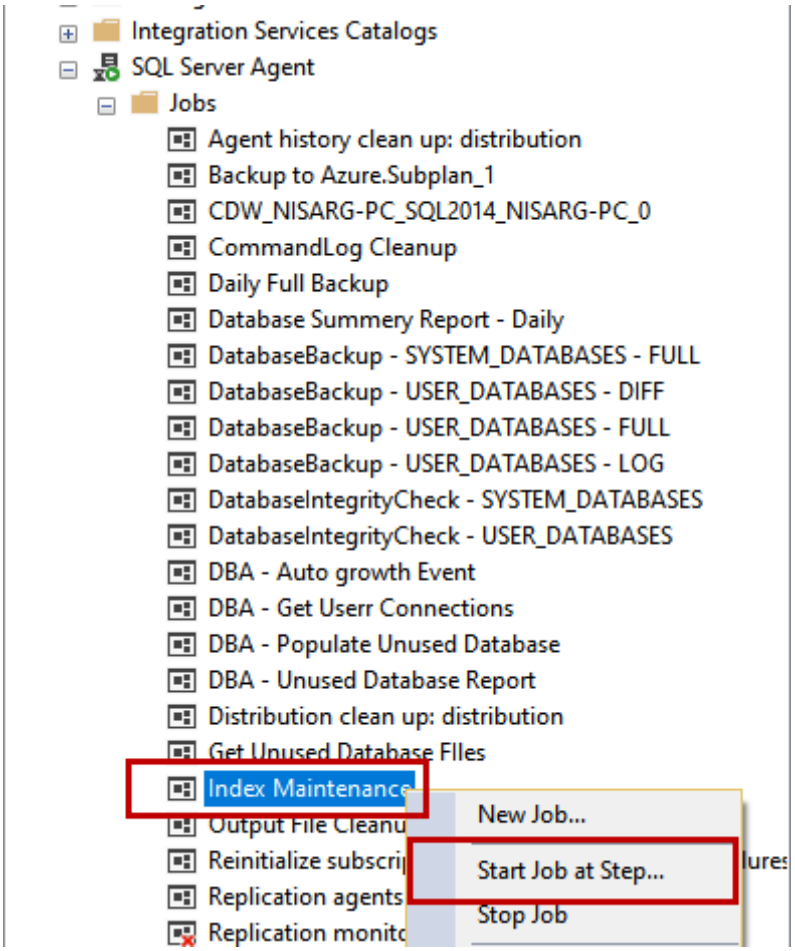


Haz clic en OK para guardar el horario. Luego, haz clic en OK en el cuadro

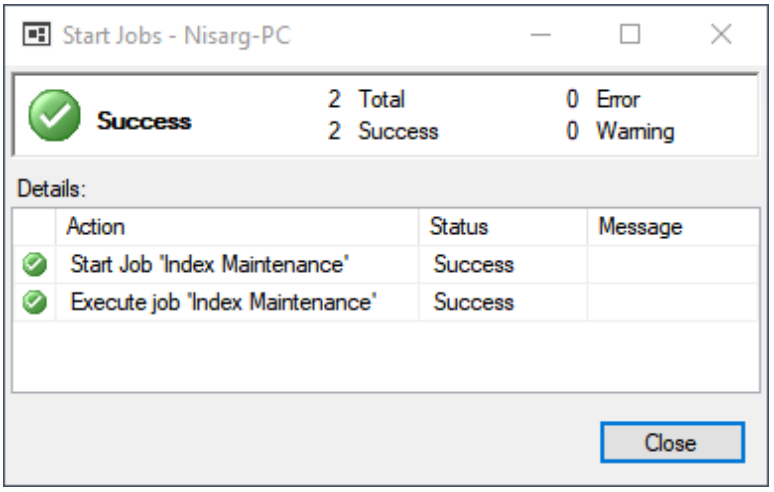
de diálogo **Nuevo Trabajo** para guardar el trabajo del Agente.

Test el SQL Job

Probamos el trabajo ejecutándolo manualmente. Haz clic derecho en el trabajo y selecciona Iniciar trabajo en el paso.



Si el trabajo se completa correctamente, verás el cuadro de diálogo de éxito como se muestra en la captura de pantalla a continuación.



En resumen, las transformaciones ETL son una etapa esencial en la gestión y transformación de datos. Puedes asegurarte de que tus procedimientos ETL sean efectivos, precisos y confiables al conocer el procedimiento y las mejores prácticas involucradas. Comprender los datos, definir los requisitos, extraer los datos, realizar transformaciones, cargar los datos, probar y validar los datos, y monitorear y mantener los datos son todos aspectos cruciales. Puedes aplicar estas recomendaciones para lograr el éxito en las transformaciones ETL de tus proyectos.

3.5.2. Recuperación de datos desde un archivo plano

En el proceso ETL de recuperación de datos desde un archivo plano, existen siete pasos clave: ubicación del archivo, acceso a su contenido, análisis de los datos, extracción de segmentos específicos, carga de los datos en un sistema objetivo, validación del procedimiento y documentación de todo el proceso. Vamos a adentrarnos en cada uno de estos pasos en detalle.

En ETL, la recuperación de datos desde un archivo plano comprende una serie de pasos metodológicos. En primer lugar, es necesario determinar la ubicación exacta del archivo, identificando el directorio o la ruta del archivo plano. Una vez localizado, el contenido del archivo plano se accede utilizando un lenguaje de programación adecuado o una herramienta ETL.

La siguiente fase es el análisis, un paso crucial que transforma los datos crudos en un formato estructurado como tablas o dataframes. Por ejemplo, si el archivo plano está en formato CSV, los datos suelen estar delimitados por comas. Mediante un analizador CSV, se puede segmentar de manera metódica estos datos en un formato donde cada fila representa un registro distintivo y cada columna corresponde a un campo específico. Este proceso es bastante similar para archivos TSV, pero aquí, las pestañas actúan como separadores. Es importante destacar que algunas herramientas ETL tienen la capacidad de reconocer intuitivamente los formatos de archivo, implementando automáticamente el enfoque correcto de análisis.

Después del análisis viene la fase de extracción. Aquí, el enfoque se estrecha seleccionando filas o columnas específicas del archivo analizado. Este subconjunto de datos se traslada a un área de almacenamiento transicional como una base de datos local o un área de preparación, preparándolo para el procesamiento subsiguiente. A modo ilustrativo, al tratar con un archivo plano extenso donde solo ciertas secciones son pertinentes para una tarea o análisis específico, la extracción se vuelve fundamental para aislar las secciones relevantes.

El paso siguiente es la carga, donde los datos previamente extraídos se transfieren desde su ubicación temporal al sistema objetivo designado. Esto puede variar desde bases de datos hasta almacenes de datos u otros sistemas de almacenamiento. Para tener una perspectiva más tangible, considera una situación donde el objetivo es una base de datos relacional: aquí entran en juego los comandos SQL INSERT, colocando cada fila de datos en su tabla correspondiente.

Para asegurar la integridad y precisión de todo este proceso, la validación es esencial. Esto implica cruzar los datos tanto del archivo plano como de su área de retención temporal, examinando diligentemente posibles discrepancias o errores que puedan haber surgido durante el procedimiento. Por último, es de suma importancia documentar minuciosamente cada paso realizado y las reglas empleadas durante la extracción. Esto no solo mejora la comprensión para la persona involucrada, sino que también resulta invaluable para otros que puedan emplear las mismas técnicas de extracción en proyectos futuros. Sin embargo, es fundamental recordar que estos pasos y herramientas pueden variar según el formato del archivo plano y las características inherentes de los datos.

Aquí tienes un ejemplo de archivo que puedes utilizar:
https://people.sc.fsu.edu/~jburkardt/data/csv/csv.html

Para más información, revisa los siguientes enlaces:

Database management

Create, Deploy, and Execute the SSIS package using SQL Server Agent SQL Server Integration Services

3.6. Completando la transformación ETL

3.6.1. Visión general de los procesos comunes de ETL

La carga de datos en un almacén de datos es el último paso una vez que se completa la transformación de datos y el punto identificativo para la finalización del proceso ETL. Es sencillo acceder y utilizar los datos una vez que se han cargado en grandes cantidades en un almacén. El proceso ETL genera una única colección limpia de datos utilizables, sin importar cuántos otros tipos de datos estuvieron involucrados.

Después de haber pasado por la transformación, los datos se colocan en una base de datos objetivo o un almacén de datos donde pueden ser almacenados y recuperados para análisis. Para maximizar el rendimiento y garantizar la integridad de los datos, este procedimiento de carga implica ingresar los datos en tablas, generar índices y llevar a cabo actividades adicionales en la base de datos.

En resumen, el proceso ETL, que se utiliza frecuentemente en el procesamiento y gestión de datos, es crucial para asegurar que los datos sean correctos, consistentes y aptos para el análisis. Estos pasos pueden ayudar a las organizaciones a obtener datos esclarecedores.

Co-funded by
the European Union

Para resumir, los procesos comunes de ETL incluyen:

- **Validación de Datos**: Este proceso implica verificar los datos contra reglas y restricciones definidas para confirmar su fiabilidad antes de utilizarlos en procesos de toma de decisiones. A medida que el volumen y la variedad de datos manejados por las organizaciones continúan creciendo, la validación efectiva de datos dentro del proceso ETL se vuelve cada vez más crucial.
- **Integración de Datos:** Esto se refiere al proceso de combinar datos de diferentes fuentes en una vista unificada que puede ser utilizada para análisis, reportes o toma de decisiones.
- **Calidad de Datos:** Un aspecto crucial de la ingeniería de datos, la calidad de datos asegura que los datos extraídos, transformados y cargados en un sistema objetivo sean precisos, consistentes y confiables.
- Data Governance: Esto se refiere a las reglas, políticas y procedimientos de una organización que aseguran el uso seguro y correcto, así como el almacenamiento adecuado de la información. Una política de gobierno de datos codifica las reglas y requisitos relacionados con los datos que una organización seguirá, además de clarificar los estándares internos de seguridad de datos de la organización.
- Seguridad de Datos: Este proceso implica proteger la información digital de usuarios no autorizados durante todo su ciclo de vida.
- **Respaldo y Recuperación de Datos**: La prueba de recuperación de respaldo ETL se utiliza para asegurar que el sistema de almacén de datos se recupere correctamente ante fallas de hardware, software o de red sin perder datos. Es crucial preparar un plan de respaldo adecuado para garantizar la máxima disponibilidad del sistema
- **Programación de Datos:** Este es el proceso de definir y ejecutar la frecuencia, dependencias y disparadores de tus trabajos ETL.

Para obtener más información, consulta los siguientes enlaces:

What Is ETL (Extract, Transform, Load)? Meaning, Process, and Tools

A Beginner's Guide to ETL Processes: ETL Stages and Benefits Explained

Co-funded by
the European Union

3.7. Ejecutando la Transformación

3.7.1. Cuales son los pasos necesarios para una ETL productiva

La complejidad y la importancia de cada paso de ETL no puede subestimarse. Proporcionan una estructura para manejar los datos de manera integral, asegurando que el resultado final en el sistema objetivo no solo sea preciso, sino también significativo y ejecutable. La ejecución adecuada de estos pasos garantiza que las empresas puedan aprovechar sus datos de manera efectiva, obteniendo ideas que impulsan la toma de decisiones e innovación. Los pasos típicos se enumeran a continuación:

- **Identificar los requisitos para la transformación de datos:** Los criterios particulares de la transformación deben definirse antes de comenzar a implementarla. Para hacerlo, puede ser necesario determinar las fuentes de datos, la base de datos de destino o el almacén de datos, las reglas de transformación de datos y las necesidades de validación y manejo de errores.
- **Desarrollar la lógica para la transformación:** Una vez establecidos los criterios, puede comenzar a crear la lógica de la transformación en sí. Esto implica escribir los scripts o código informático necesarios para transformar los datos del sistema fuente a un formato que el sistema objetivo pueda utilizar. La limpieza de datos, el mapeo de datos, la agregación de datos, el filtrado de datos y la validación de datos son algunos ejemplos de la lógica de transformación.
- **Probar la transformación:** Una vez creada la lógica de transformación, debe probarla adecuadamente para asegurarse de que funcione según lo previsto. Para comprobar que los datos se transforman como se espera, se ejecutan conjuntos de datos de muestra contra la lógica de transformación de esta manera. Para

asegurarse de que la lógica de transformación sea confiable y esté libre de errores, debe probarla contra situaciones extremas y condiciones límite.

- **Realizar la transformación:** Después de probar y validar la lógica de transformación, puede comenzar la transformación. En este caso, los datos transformados deben insertarse en el sistema de destino después de aplicar la lógica de transformación a los datos reales del origen. Debe vigilar el rendimiento de la transformación durante la fase de ejecución para asegurarse de que cumple con sus criterios de rendimiento y escalabilidad.
- **Monitorear y mantener la transformación**: Después de ejecutar la transformación, debe monitorear cómo está funcionando y mantenerla a lo largo del tiempo. Como parte de esto, se debe monitorear la calidad de los datos, encontrar y corregir errores y problemas, y actualizar la lógica de transformación según sea necesario para tener en cuenta modificaciones en los sistemas fuente o objetivo.

Ejecutar la transformación en un proyecto ETL implica identificar los requisitos, desarrollar la lógica de transformación, probarla a fondo, ponerla en práctica, monitorearla a lo largo del tiempo y mantenerla. Siguiendo estos procedimientos, puede asegurarse de que sus datos se conviertan de manera precisa, efectiva y consistente.

3.7.2. Introducción en ETL testing

Si estás integrando y migrando datos a un nuevo sistema utilizando un proceso de Extracción, Transformación y Carga (ETL), es importante asegurarse de que la calidad de tus datos sea alta. Una de las mejores maneras de hacerlo es mediante las pruebas ETL, las cuales evalúan si tus datos son completos, precisos y confiables, y si han sido cargados correctamente en tu nuevo sistema o almacén de datos.

Sin pruebas ETL, las empresas corren el riesgo de tomar decisiones utilizando datos inexactos o incompletos. Esto puede tener impactos negativos en los ingresos, la estrategia y la experiencia del cliente.

Cuándo debes usar las pruebas ETL?

- Es importante utilizar las pruebas ETL en las siguientes situaciones:
- Después de cargar datos por primera vez en un nuevo almacén de datos
- Después de agregar una nueva fuente de datos a un almacén de datos existente.
- Después de una migración de datos.
- Durante el movimiento de datos.
- Cada vez que existan preocupaciones con la calidad de los datos o el rendimiento del proceso ETL.

Cada vez que estés moviendo o integrando datos, debes asegurarte de que la calidad de tus datos sea alta antes de utilizarlos para análisis, inteligencia empresarial o toma de decisiones. Si te han asignado las pruebas ETL, se te pedirá que asumas algunas responsabilidades importantes.

3.7.3. El proceso de ETL testing: etapas y mejores prácticas

Las pruebas ETL efectivas detectan problemas con los datos fuente desde

el principio, antes de que sean cargados en el repositorio de datos, así como también encuentran inconsistencias o ambigüedades en las reglas de negocio destinadas a guiar la transformación e integración de datos. El proceso puede dividirse generalmente en ocho etapas:

1. Identificar los requisitos del negocio — Diseñar el modelo de datos, definir el flujo de negocio y evaluar las necesidades de informes basadas en las expectativas del cliente. Es importante comenzar aquí para que el alcance del proyecto esté claramente definido, documentado y completamente entendido por los probadores.
2. Validar las fuentes de datos — Realizar una verificación de conteo de datos y verificar que los tipos de datos de las tablas y columnas cumplan con las especificaciones del modelo de datos. Asegurarse de que las claves de verificación estén en su lugar y eliminar los datos duplicados. Si no se hace correctamente, el informe agregado podría ser inexacto o engañoso
3. Diseñar casos de prueba — Diseñar escenarios de mapeo ETL, crear scripts SQL y definir reglas de transformación. Es importante validar también el documento de mapeo para asegurarse de que contenga toda la información necesaria.
4. Extraer datos de los sistemas fuente — Ejecutar pruebas ETL según los requisitos del negocio. Identificar tipos de errores o defectos encontrados durante las pruebas y realizar un informe. Es importante detectar y reproducir cualquier defecto, reportarlo, corregir el error y cerrar el informe de errores antes de continuar con la Etapa 5.
5. Aplicar la lógica de transformación — Asegurarse de que los datos se transformen para que coincidan con el esquema del almacén de datos objetivo. Verificar el umbral de los datos y la alineación y validar el flujo de datos. Esto asegura que el tipo de datos coincida con el documento de mapeo para cada columna y tabla.
6. Cargar datos en el almacén de datos objetivo — Realizar una verificación de conteo de registros antes y después de que los datos se muevan desde el área de preparación hasta el almacén de datos. Confirmar que los datos inválidos sean rechazados y que se acepten los valores por defecto.
7. Informe resumido — Verificar el diseño, opciones, filtros y la funcionalidad de exportación del informe resumido. Este informe permite a los

tomadores de decisiones y otras partes interesadas conocer los detalles y resultados del proceso de pruebas. Si alguna etapa no se completó, el informe les informa el motivo.

8. Cierre de pruebas — Archivar el cierre de pruebas. Ahora puedes avanzar con ETL sabiendo que la calidad de tus datos es sólida.

3.8. Otros aspectos de las transformaciones

3.8.1. Ejemplo de conversión de campos

Existen varias formas de cambiar el formato de los campos, dependiendo de la herramienta ETL específica o del lenguaje de programación utilizado. Algunos métodos comunes incluyen el uso de funciones SQL como **CAST o CONVERT.**

Veamos cómo funcionan y cuáles son sus diferencias.

Aquí está la sintaxis general para usar la función CAST:

```
CAST(expression AS data_type)
```

Donde "expression" es el campo o valor que deseas convertir, y "data_type" es el tipo de datos al que deseas convertirlo.

Por ejemplo, para convertir un campo llamado "price" de tipo de datos decimal a tipo de datos entero, usarías la siguiente declaración SQL:

```
SELECT CAST(price AS INT) FROM products;
```

De manera similar, la función CONVERT funciona de la siguiente manera:

```
CONVERT(data_type, expression [, style])
```

Aquí, "expression" es el campo o valor que deseas convertir, "data_type" es el tipo de datos al que deseas convertirlo, y "style" es un parámetro opcional que especifica el formato de ciertos tipos de datos, como fecha y hora.

Por ejemplo, para convertir un campo llamado "created_at" de tipo de datos datetime a tipo de datos date, usarías la siguiente declaración SQL:

```
SELECT CONVERT(DATE, created_at) FROM orders;
```

3.8.2.Joints, ejemplo de cruzamiento de datos

Unir datos y cruzar referencias son tareas esenciales en los procesos de integración de datos y ETL.

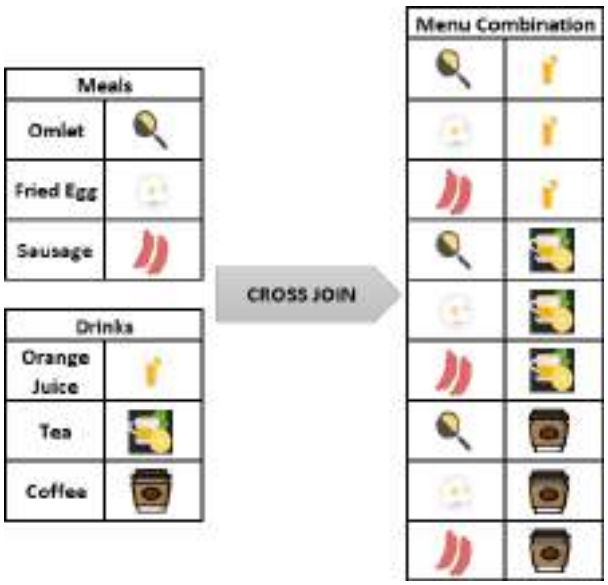
Veamos un ejemplo cotidiano. Imaginemos que estamos en una cafetería y queremos ordenar el desayuno

3.8.2.1. Input data sets

Revisamos el menú para ver qué combinación de comida y bebida nos parece más atractiva. Cuando recibimos la señal, nuestro cerebro comienza a generar todas las posibles combinaciones de comida y bebida.

La imagen a continuación representa todas las posibles combinaciones de menú generadas por nuestro cerebro. De manera similar, el mecanismo SQL CROSS JOIN genera todas las combinaciones emparejadas de las filas de las tablas que se unirán. La idea principal de CROSS JOIN es devolver el producto cartesiano de las tablas unidas. En teoría de conjuntos, el Producto Cartesiano es una operación de multiplicación que genera todos los pares ordenados de los conjuntos dados.

Co-funded by
the European Union



La sintaxis del CROSS JOIN en SQL se verá como la siguiente:

```
SELECT ColumnName_1, ColumnName_2,
ColumnName_N FROM [Table_1]
CROSS JOIN [Table_2]
```

Alternativamente, podemos utilizar la siguiente sintaxis en lugar de la anterior. Esta sintaxis no incluye la palabra clave CROSS JOIN; en su lugar, las tablas que se van a unir se colocarán después de la cláusula FROM y separadas por una coma.

```
SELECT ColumnName_1, ColumnName_2,
ColumnName_N FROM [Table_1],[Table_2]
```

3.8.2.2. Crear tablas

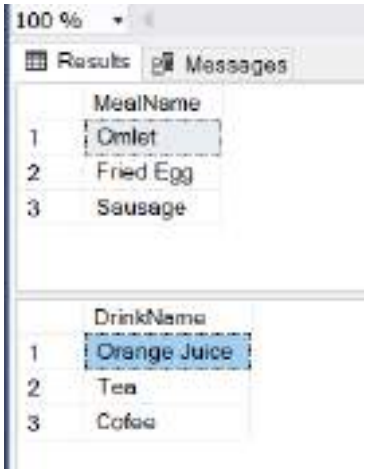
En este ejemplo, revisaremos el ejemplo del menú de desayuno de la sección anterior del artículo. Primero, crearemos dos tablas de muestra con los nombres de las bebidas y las comidas. Luego las llenaremos con algunos datos de ejemplo.

A continuación, se muestra cómo realizaremos estos dos pasos mediante

una consulta:

```
CREATE TABLE Meals(MealName VARCHAR(100))
CREATE TABLE Drinks(DrinkName VARCHAR(100))
INSERT INTO Drinks VALUES('Orange Juice'), ('Tea'),
('Cofee') INSERT INTO Meals VALUES('Omlet'), ('Fried
Egg'), ('Sausage')

SELECT * FROM Meals;
SELECT * FROM Drinks
```



3.8.2.3. *Cross join*

El siguiente query utilizará la palabra clave CROSS JOIN para unir las tablas Meals (Comidas) y Drinks (Bebidas), devolviendo todas las combinaciones emparejadas de nombres de comida y bebida..

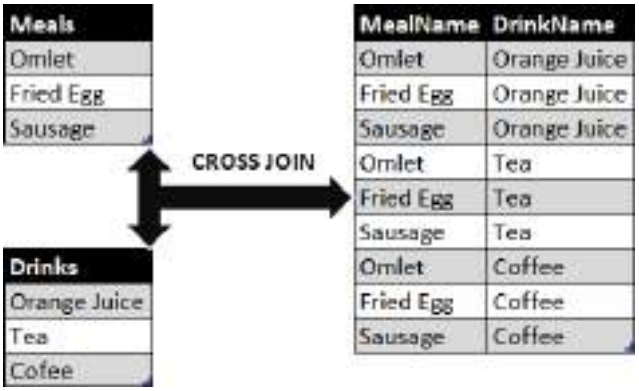
```
SELECT * FROM Meals
CROSS JOIN Drinks
```

100 %

Results | Messages

	MealName	DrinkName
1	Omlet	Orange Juice
2	Fried Egg	Orange Juice
3	Sausage	Orange Juice
4	Omlet	Tea
5	Fried Egg	Tea
6	Sausage	Tea
7	Omlet	Cofee
8	Fried Egg	Cofee
9	Sausage	Cofee

La siguiente imagen ilustra el principio de funcionamiento del CROSS JOIN.



Al mismo tiempo, podemos utilizar la siguiente consulta para obtener el mismo conjunto de resultados con una sintaxis diferente que no incluye un CROSS JOIN.

```
SELECT * FROM Meals ,Drinks
```

100 %

Results | Messages

	MealName	DrinkName
1	Omlet	Orange Juice
2	Fried Egg	Orange Juice
3	Sausage	Orange Juice
4	Omlet	Tea
5	Fried Egg	Tea
6	Sausage	Tea
7	Omlet	Cofee
8	Fried Egg	Cofee
9	Sausage	Cofee

3.8.2.4. Combinar columnas

Consejo: El número de filas del conjunto de resultados será igual a la multiplicación de las filas de las tablas que se van a unir. Para el ejemplo del menú de desayuno, la cantidad de filas de la tabla Meals es 3 y la cantidad de filas de la tabla Drinks es 3, por lo que se puede encontrar el número de filas del conjunto de resultados con el siguiente cálculo.

3 (Meals table row count) x 3 (Drinks table row count) = 9 (Resultset row count)

La función CONCAT_WS ayuda a concatenar las expresiones de columnas. De esta manera, podemos crear un conjunto de resultados del menú de desayuno más significativo.

```
SELECT CONCAT_WS('-',MealName,DrinkName) AS MenuList FROM Meals CROSS JOIN Drinks
```

Results | Messages

	MenuList
1	Omlet-Orange Juice
2	Fried Egg-Orange Juice
3	Sausage-Orange Juice
4	Omlet-Tea
5	Fried Egg-Tea
6	Sausage-Tea
7	Omlet-Cofee
8	Fried Egg-Cofee
9	Sausage-Cofee

Busca más información sobre SQL CROSS JOIN with examples.

Co-funded by
the European Union

Referencias

Yaseen, A. (2017, October 3). *SQL Server Data Type Conversion Methods and performance comparison*. SQLShack https://www.sqlshack.com/sql-server-data-type-conversion-methods-performance-comparison/

Stitchdata (n.d.). *ETL Database Your central database for all things ETL: advice, suggestions, and best practices*. https://www.stitchdata.com/etldatabase/

IBM ETL (n.d.). *ETL (Extract, Transform, Load)*. https://www.ibm.com/topics/etl

Google Cloud (2022, April). *BigQuery for data warehouse professionals*. https://cloud.google.com/architecture/bigquery-data-warehouse?hl=es-419#managing_data

Hughes, A. (n.d.). *The Key Steps in the ETL Data Integration Process*. Cleo

https://www.cleo.com/blog/knowledge-base-etl-integration#:~:text=The%205%20steps%20of%20the,the%20most%20important%20process%20steps.&text=Clean%3A%20Cleans%20data%20extracted%20from,the%20data%20prior%20to%20transformation

Nguyen, R. (2022, September 26). *ETL in Data Warehouse: The Definitive Guide*. GEM https://gemvietnam.com/big-data/etl-in-data-warehouse-the-definitive-guide/

Microsoft (2023, March 01). *Load data into SQL Server or Azure SQL Database with SQL Server Integration Services (SSIS)*. https://learn.microsoft.com/en-us/sql/integration-services/load-data-to-sql-database-with-ssis?view=sql-server-ver16

Talend (n.d.). *ETL testing: A comprehensive guide to ensuring data quality and integration*. https://www.talend.com/resources/etl-testing/

Software Testing Help, (2023, June 24). *What Is ETL (Extract, Transform, Load) Process In Data Warehouse?* https://www.softwaretestinghelp.com/etl-process-in-data-warehouse/#Data_Extraction

Free Time Learn (n.d.). *ETL Testing - Interview Questions*.

https://www.freetimelearning.com/software-interview-questions-and-answers.php?What-are-the-differences-between-ETL-and-BI-

Co-funded by
the European Union

tools?&id=7896#:~:text=The%20ETL%20tools%20are%20used,quarterly%2C%20and%20annual%20board%20meetings

Google Cloud (n.d.). *What is ETL?* https://cloud.google.com/learn/what-is-etl?hl=es-419#section-4

Poddar A. (2022, December 29). *Understanding ETL BI: 6 Comprehensive Aspects*. Hevo https://hevodata.com/learn/etl-bi/#i2

Cloudmoyo (2019, December 3). *Implementing an Effective Extract, Transform, Load Process for Your Data Warehouse*. https://www.cloudmoyo.com/blogs/how-to-implement-the-etl-steps-for-your-data-warehouse/

Microsoft (2023, March 01). *SSIS How to Create an ETL Package*. https://learn.microsoft.com/en-us/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver16

Microsoft (2023, March 01). *Data Conversion Transformation*. https://learn.microsoft.com/en-us/sql/integration-services/data-flow/transformations/data-conversion-transformation?view=sql-server-ver16

Microsoft (2023, August 18). *SQL Server Integration Services*. https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16

Peterson, R. (2023, August 22). *How to Download and Install SQL Server for Windows (FREE)*. https://www.guru99.com/download-install-sql-server.html

Microsoft (2023, March 01). *SSIS How to Create an ETL Package*. https://learn.microsoft.com/en-us/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver16

Dearmer, A. (2023, July 17). *Top 14 ETL Tools for 2023*. Integrate.io https://www.integrate.io/blog/top-7-etl-tools/

Datacamp (2023, July). *A List of The 18 Best ETL Tools And Why To Choose Them*. https://www.datacamp.com/blog/a-list-of-the-16-best-etl-tools-and-why-to-choose-them

javaTpoint (n.d.). *SSIS Tutorial*. https://www.javatpoint.com/ssis

Quora (n.d.). *What is the difference between Pentaho and Microsoft SQL Server Integration Services?* https://www.quora.com/What-is-the-difference-between-Pentaho-and-Microsoft-SQL-Server-Integration-Services

Taylor, D. (2023, July 4). Pentaho Data Integration Tutorial: What is, Pentaho ETL Tool. GURU99 https://www.guru99.com/pentaho-tutorial.html

Tutorialspoint (n.d.). *Pentaho - Overview*. https://www.tutorialspoint.com/pentaho/pentaho_overview.htm

Techtarget (2023, August 24). *Database management*. https://www.techtarget.com/searchdatamanagement/resources/Database-management

Upadhyay, N. (2020, November 06). *Create, Deploy, and Execute the SSIS package using SQL Server Agent*. CodingSight https://codingsight.com/create-deploy-and-execute-ssis-package-using-sql-server-agent/

Ashtari, H. (2022, September 8). *What Is ETL (Extract, Transform, Load)? Meaning, Process, and Tools*. Spiceworks https://www.spiceworks.com/tech/devops/articles/extract-transform-load-etl/

Shykolovych, O. (2020, December 31). *A Beginner's Guide to ETL Processes: ETL Stages and Benefits Explained*. Improvado https://improvado.io/blog/etl-process-extract-transform-load

Erkec, E. (2020, February 24). *SQL CROSS JOIN with examples*. SQLShack https://www.sqlshack.com/sql-cross-join-with-examples/