

BI4SME

boosting business
intelligence skills for SME
growth

BI4SME - R2 — Materiales de Entrenamiento Unidad 6: Laboratorios



Co-funded by
the European Union

GRANT AGREEMENT
2021-1-E501-KA220-VET-000033132

Contents

UNIDAD 6: LABORATORIOS	3
6.1. INTRODUCCIÓN	3
6.2. BI4SME LABORATORIOS	4
6.2.1. Paso 1 – Prerrequisitos - configuración de herramientas	4
6.2.2. Paso 2 – Source Data	4
6.2.3. Paso 3 – ETL	6
6.2.4. Paso 4 – Almacén de datos usando SQL	34
6.2.5. Paso 5 – Visualización de datos usando Power BI	41
6.2.6. Paso 6 – Uso de Python para BI	56

Public Licenc e



Este trabajo © 2023 por los Socios del Consorcio BI4SME está licenciado bajo la licencia Attribution-NonCommercial-NoDerivatives 4.0 International. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>

UNIDAD 6: Laboratorios

6.1. Introducción

Como saben, el objetivo de esta unidad de formación es enseñar los **fundamentos de la Inteligencia de Negocios (BI)**. Concretamente, en las unidades anteriores han aprendido sobre una serie de herramientas y técnicas de código abierto y gratuitas relacionadas con BI y analítica de una manera práctica. Han aprendido a trabajar con conjuntos de datos, cómo extraer, transformar, analizar y visualizar datos.

Ahora, con el fin de reforzar los conocimientos recién adquiridos, se sumergirán en laboratorios de práctica. Su objetivo es permitirles **aplicar nuevas habilidades técnicas en un entorno empresarial concreto**, para que puedan ver cómo podrían aplicarse en su negocio diario durante el proceso de aprendizaje. Trabajarán con datos reales, intentando aplicar diferentes técnicas y herramientas de BI que han aprendido en el programa BI4SME desde ETL hasta Python. Intenten manejar los ejercicios por su cuenta. Y no se preocupen, si tienen dudas en cualquier momento, les proporcionaremos explicaciones y soluciones para todos los ejercicios después de cada uno de ellos.

6.2. BI4SME Laboratorios

6.2.1. Paso 1 – Prerrequisitos - configuración de herramientas

Para comenzar con nuestro laboratorio, será necesario instalar varias herramientas en su computadora. Posiblemente ya las tengan configuradas, ya que han estado trabajando con ellas en las unidades anteriores, pero por si acaso estamos adjuntando a esta unidad las guías para instalarlas como materiales descargables.

- MariaDB, including HeidiSQL
- SQL Server Integration Services (SSIS)
- Power BI
- Anaconda and Spider

6.2.2. Paso 2 – Source Data

Otro paso es asegurar los datos con los que trabajaremos. Los datos pueden recopilarse de varias fuentes. Hay sitios que ofrecen datos que se pueden utilizar libremente. Así que, en el caso de nuestros laboratorios, trabajaremos con este tipo de datos, llamados **Datos Abiertos** (para saber más sobre datos abiertos, pueden consultar este artículo: <https://data.europa.eu/en/dataeuropa-academy/what-open-data>).

Concretamente, hemos obtenido algunos archivos csv con datos exportados de una **cafetería** (fuente: <https://www.kaggle.com/datasets/ylchang/coffee-shop-sample-data-1113?select=201904+sales+reciepts.csv>). Estos archivos están adjuntos a esta unidad como materiales descargables y son necesarios para avanzar con los siguientes pasos de este laboratorio.

Estos archivos contienen datos relacionados con:

- Customer
- Inventory
- Product
- Sales
- Sales Target
- Staff
- Store

Es necesario tener en cuenta las **relaciones de datos** entre ellos:

- Cliente a Tienda: Muchos a Uno (Cada cliente tiene una tienda de origen)
- Inventario a Producto: Muchos a Uno (Cada entrada de inventario hace referencia a un producto)
- Inventario a Tienda: Muchos a Uno (Cada entrada de inventario hace referencia a una tienda)
- Ventas a Cliente: Muchos a Uno (Cada venta puede hacer referencia a un cliente)
- Ventas a Producto: Muchos a Uno (Cada venta hace referencia a un producto)
- Ventas a Personal: Muchos a Uno (Cada venta hace referencia a un miembro del personal)
- Ventas a Tienda: Muchos a Uno (Cada venta hace referencia a una tienda)
- Tienda a Personal: Uno a Uno (Cada tienda tiene un gerente)

6.2.3. Paso 3 – ETL

6.2.3.1. [Importación básica de datos](#)

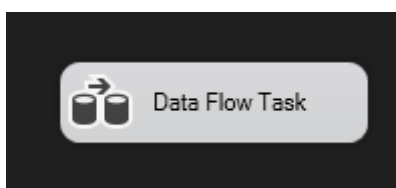
- Cree un paquete SSIS para importar los datos CSV directamente en las respectivas tablas de la base de datos. Use los componentes de Fuente de Archivo Plano (Flat File Source) y Destino OLE DB (OLE DB Destination).

1. Configuración de su proyecto

- Inicie SQL Server Data Tools (SSDT): Inicie la aplicación SSDT en su máquina.
- Inicie un nuevo proyecto SSIS: Dentro de SSDT, navegue para crear un nuevo proyecto de Servicios de Integración.

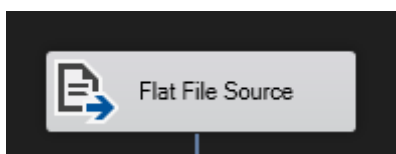
2. Diseño del flujo de datos:

- Integre una Tarea de Flujo de Datos: En el diseñador de Control Flow, arrastre una "Tarea de Flujo de Datos". Esta tarea es fundamental para estructurar el movimiento de datos.

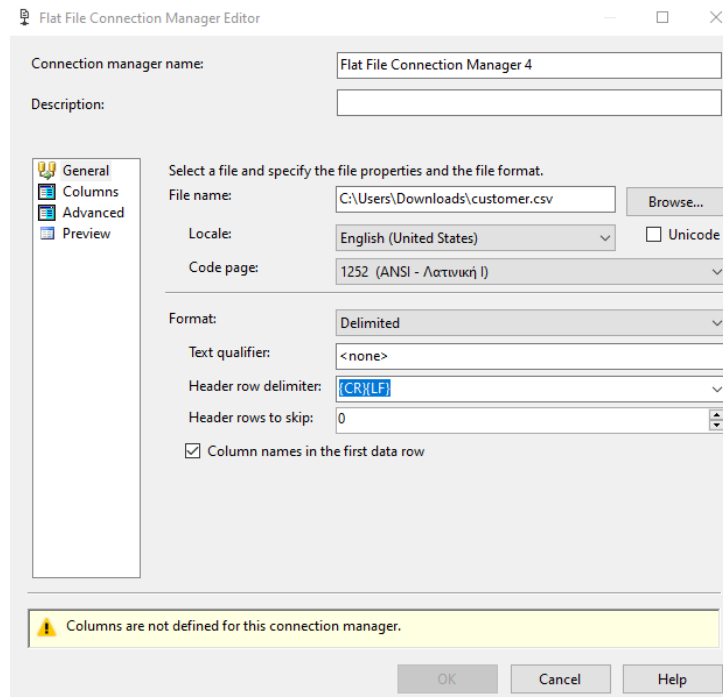


3. Configuración de la fuente de datos:

- Introduzca la Fuente de Archivo Plano: Arrastre el componente "Flat File Source" desde el SSIS Toolbox a su espacio de trabajo de Flujo de Datos.

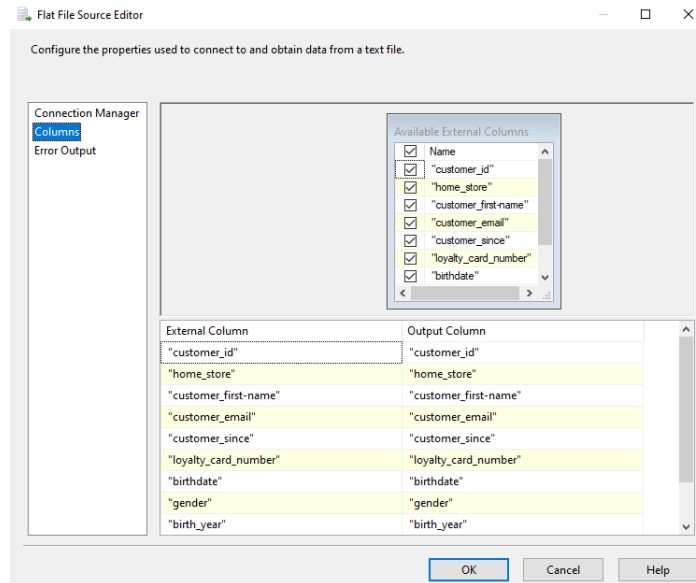


- Configure la Fuente: Haga doble clic en el componente de Fuente de Archivo Plano para configurar sus propiedades. Asegúrese de especificar la ruta, el formato y los detalles relevantes de su archivo CSV.



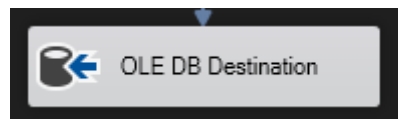
3. Ajuste de columnas de datos:

- Revise las columnas de datos: Después de configurar la conexión del archivo, navegue a la pestaña "Columns" en el editor de Fuente de Archivo Plano. Aquí, puede revisar la estructura de sus datos entrantes.
- Modifique los detalles de las columnas: Ajuste los tipos de datos, longitudes y nombres de las columnas para asegurarse de que coincidan con el esquema de la tabla de destino.

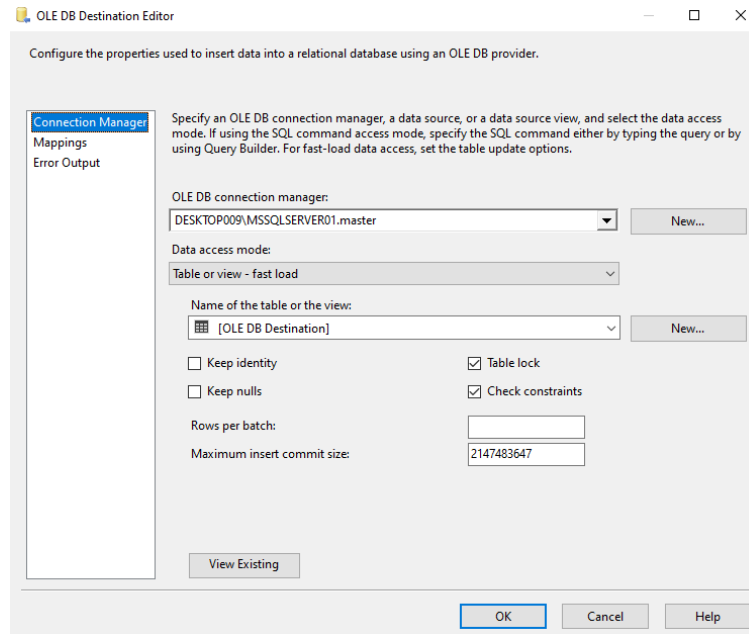


4. Designación del destino de datos:

- Introduzca el Destino OLE DB: Arrastre el componente "OLE DB Destination" desde el SSIS Toolbox a su espacio de trabajo. Este actuará como el punto final de sus datos.



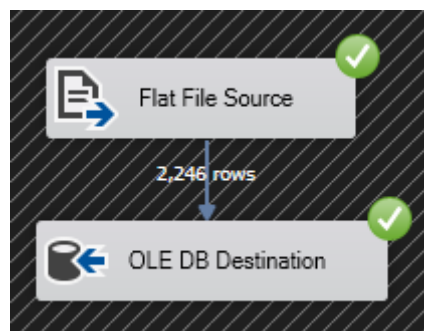
- Configure el destino: Haga doble clic en este componente y configure una conexión a su servidor SQL. Asegúrese de especificar la tabla de destino y de mapear correctamente las columnas entrantes con las columnas de la tabla.



- Nota de atribución: Si hay referencias como "Fuente: 1 2", asegúrese de citarlas adecuadamente.

5. Ejecución del paquete SSIS:

- Ejecute su flujo de datos: Después de configurar la fuente y el destino, inicie el paquete SSIS haciendo clic en el botón "Run". Esto iniciará el proceso de importación de datos.



- Manejo de errores: Si surgen problemas durante la ejecución, especialmente relacionados con la "Página de códigos", navegue a las propiedades del Destino OLE DB. Aquí, establezca la propiedad "AlwaysUseDefaultCodePage" en True para resolver el problema.

En caso de que ocurra un error, seleccione el destino OLE DB y cambie la propiedad **"AlwaysUseDefaultCodePage"** a **True**.

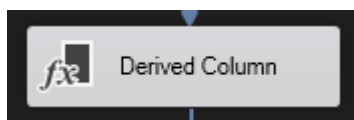
6.2.3.2. Transformación de datos:

- **Importe los datos CSV, pero antes de cargarlos en la base de datos, use la Transformación de Columna Derivada para agregar una nueva columna que calcule el impuesto sobre las ventas basado en el precio del producto.**

1. Configuración de la Transformación de Columna Derivada

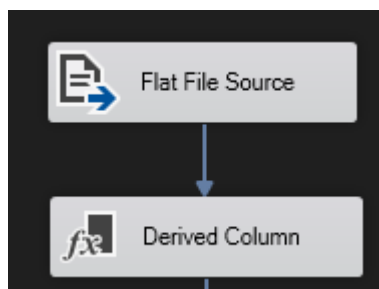
a. Inserte el componente de transformación:

- Arrastre y suelte el componente "Derived Column Transformation" en el espacio de trabajo del diseñador de Flujo de Datos.



b. Conéctese a la Fuente de Archivo Plano:

- Identifique el componente "Flat File Source" en la superficie de diseño.
- Dibuje la flecha azul (ruta de datos) desde el componente "Flat File Source", conectándolo al componente "Derived Column Transformation".



2. Configuración de la Transformación de Columna Derivada:

a. Acceda al editor:

- Haga doble clic en el componente "Derived Column Transformation" para iniciar su editor.

b. Especifique la nueva columna:

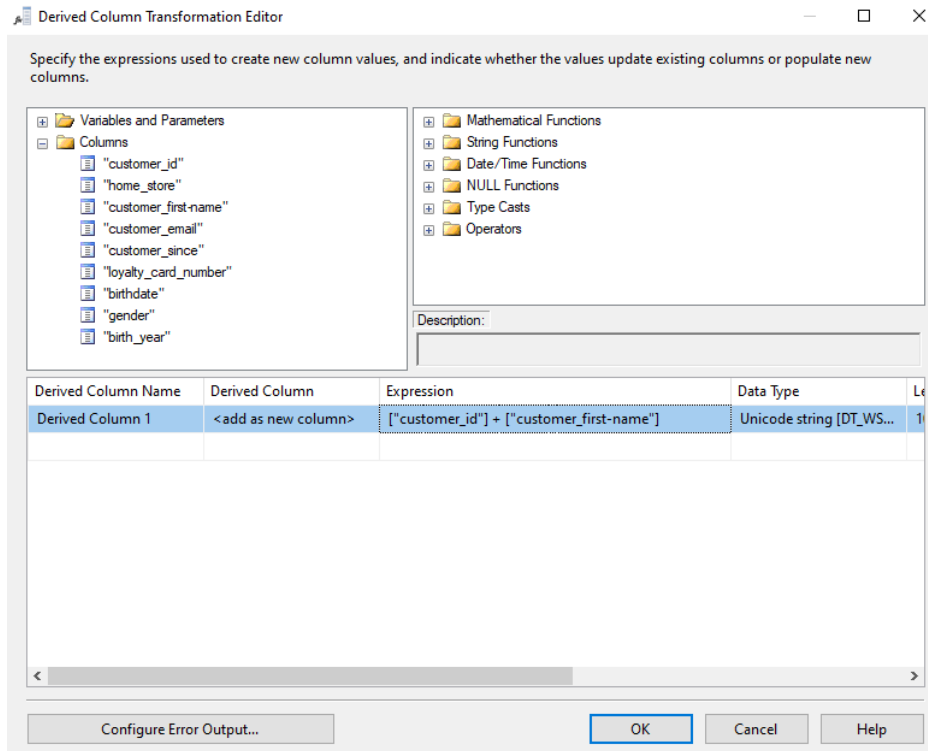
- En la sección etiquetada "Derived Column Name", escriba un nombre único para su nueva columna, como "SalesTax".

c. Añadir como nueva columna:

- Ubique la sección titulada "Derived Column".
- Elija la opción "<add as new column>". Esto asegura que está añadiendo el impuesto sobre las ventas como una columna completamente nueva en lugar de reemplazar una existente.

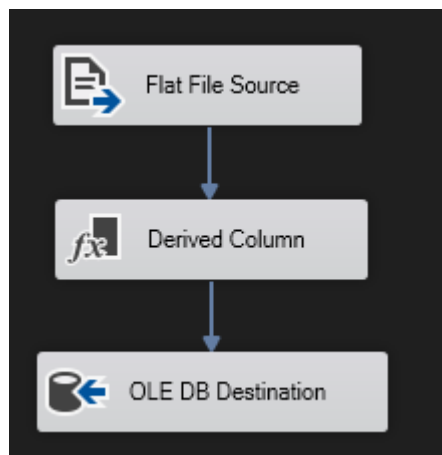
d. Ingrese la expresión de cálculo:

- En el campo "Expression", ingrese la fórmula necesaria para calcular el impuesto sobre las ventas a partir del precio del producto.
- Por ejemplo, si la columna del precio del producto está etiquetada como "Price" y la tasa de impuesto es del 10%, su expresión debería ser: $[Price] * 0.10$.



e. Conéctese al Destino OLE DB:

- Arrastre la flecha azul (ruta de datos) desde el componente "Derived Column Transformation" hasta el componente "OLE DB Destination".

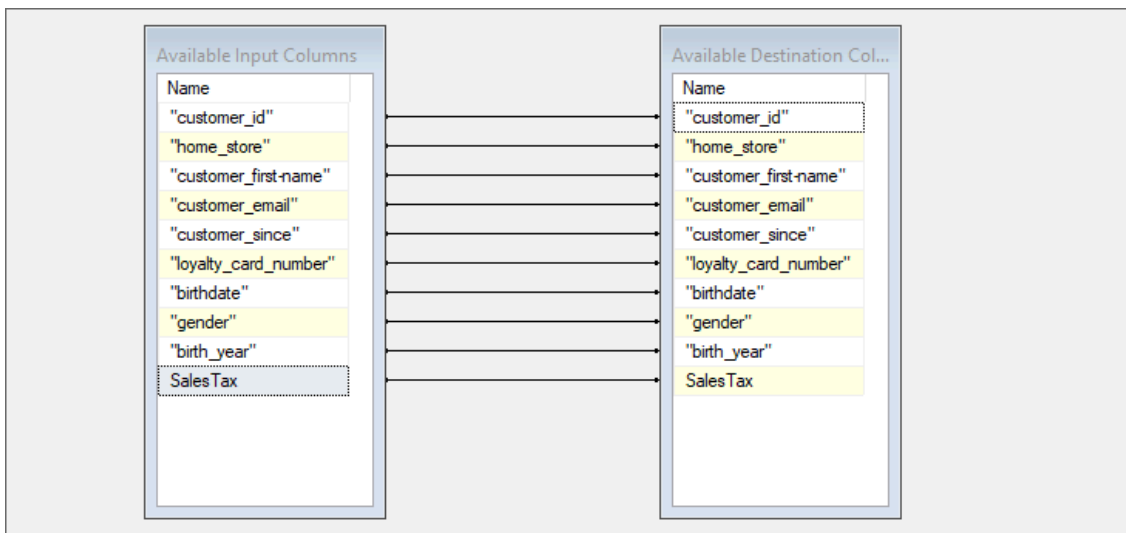


f. Configuración del Destino OLE DB:

- Haga doble clic en el componente "OLE DB Destination" para acceder a su editor.

g. Mapear la nueva columna:

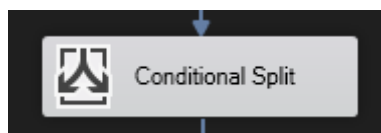
- Confirme que la nueva columna "SalesTax" esté correctamente vinculada a la columna relevante en su tabla de destino. Asegúrese de que las demás columnas también estén adecuadamente mapeadas.



- Use la Transformación de División Condicional para separar registros basados en ciertas condiciones, por ejemplo, ventas por encima de una cierta cantidad.

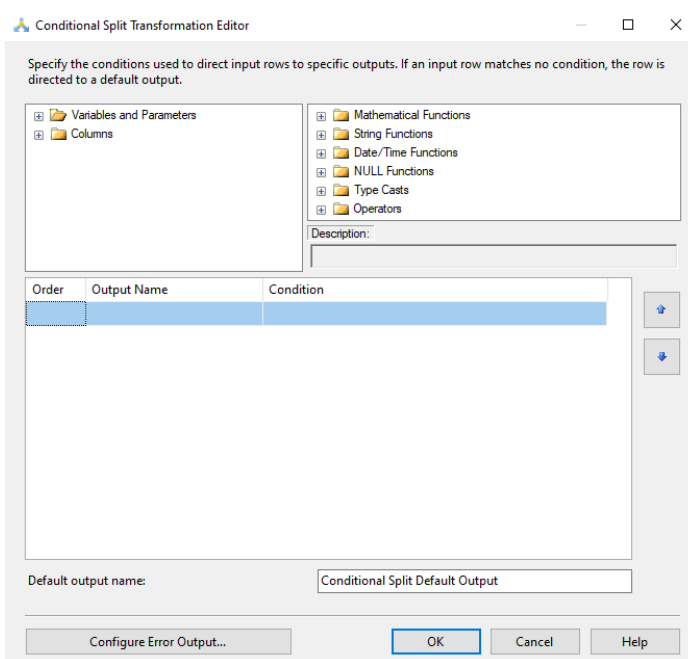
1. Añadir la Transformación de División Condicional

- Desde el SSIS Toolbox, arrastre la Transformación de División Condicional al diseñador de Flujo de Datos.
- Conecte la ruta de datos desde su componente fuente (por ejemplo, Fuente de Archivo Plano o Fuente OLE DB) a la Transformación de División Condicional².



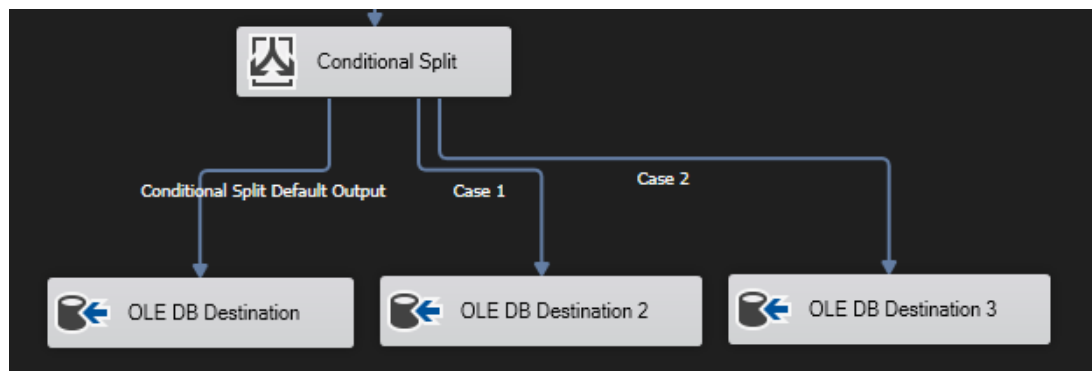
2. Configuración de la Transformación de División Condicional:

- Haga doble clic en la Transformación de División Condicional para abrir su editor.
- Defina las condiciones para dividir los datos. Por ejemplo, para separar los registros de ventas por encima de \$10,000, podría usar una condición como **SalesAmount > 10000**. Nombre la salida con algo descriptivo como **HighValueSales**².



3. Añadiendo componentes de destino:

- Para cada condición que haya definido, arrastre un nuevo Destino OLE DB desde el SSIS Toolbox al diseñador de Flujo de Datos.
- Conecte las respectivas flechas de salida desde la Transformación de División Condicional a cada Destino OLE DB.
- Estos destinos pueden ser tablas separadas en su base de datos o diferentes salidas para procesar aún más los datos².



4. Ejecución del paquete:

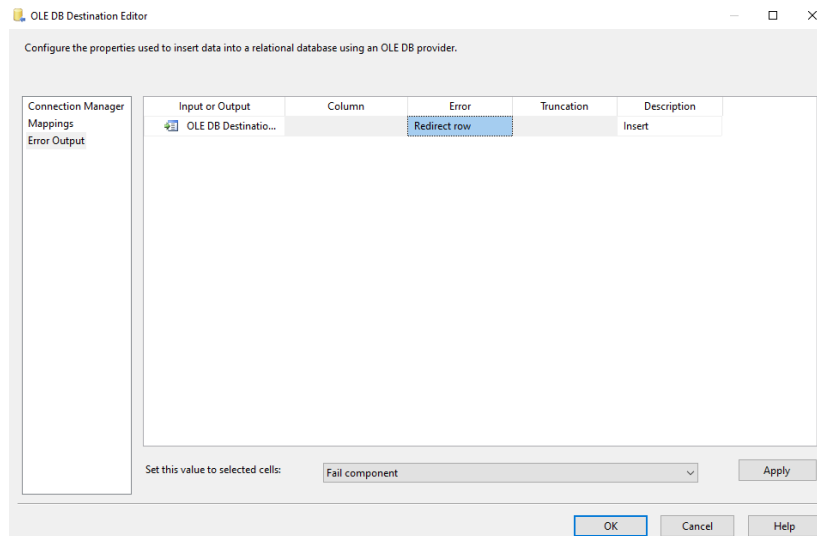
- Después de configurar las transformaciones y los destinos, guarde y ejecute su paquete SSIS para importar los datos CSV a su base de datos, segregando los registros según las condiciones que especificó.
- Valide la importación de datos consultando las tablas de destino en su base de datos y asegurándose de que los datos se hayan segregado y cargado correctamente.

6.2.3.3. Manejo de errores:

- **Mejore el paquete básico de importación de datos para manejar errores. Redirija las filas con errores a una tabla de errores separada en la base de datos o a un archivo CSV de errores.**

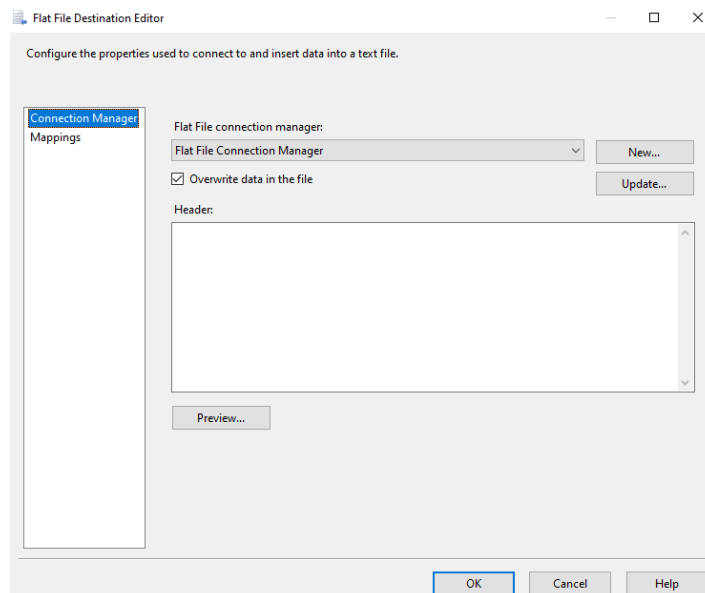
1. Configure la salida de errores

- Para cada fuente o transformación que pueda generar errores, configure la salida de errores:
 - Haga doble clic en el componente para abrir su editor
 - Navegue a la pestaña "Error Output".
 - Para cada tipo de error (por ejemplo, truncamiento, fallo de conversión de datos), seleccione "Redirect row" para enviar las filas con errores a una salida separada¹.



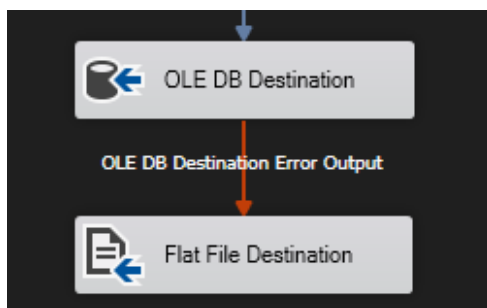
2. Add an Error Destination

- Arrastre un Destino OLE DB o un Destino de Archivo Plano desde el SSIS Toolbox al diseñador de Flujo de Datos:
 - Si elige Destino OLE DB: Conéctelo a su base de datos y cree una nueva tabla de errores para almacenar las filas con errores.
 - Si elige Destino de Archivo Plano: Configúrelo para escribir en un nuevo archivo CSV de errores².



3. Conecte la salida de errores:

- Conecte la flecha roja (ruta de salida de errores) desde la fuente o la transformación al componente de destino de errores. Esto redirigirá cualquier fila con errores a la tabla de errores en la base de datos o al archivo CSV de errores. ¹.



4. Configurar el destino de errores:

- Haga doble clic en el componente de destino de errores para abrir su editor.
- Mapee las columnas de entrada a las columnas de destino.
- Si está utilizando un Destino OLE DB, asegúrese de que la tabla de errores en su base de datos tenga el esquema apropiado para contener los datos de errores, incluyendo información sobre el error (por ejemplo, columna de error, descripción del error).

5. Ejecute SSIS Paquete

- Guarde y ejecute su paquete SSIS.
- Cualquier fila que encuentre errores durante el proceso de importación será redirigida a la tabla de errores en la base de datos o al archivo CSV de errores.

6. Verifique el manejo de errores:

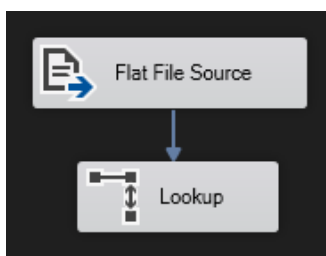
- Examine la tabla de errores en la base de datos o el archivo CSV de errores para asegurarse de que las filas con errores se hayan capturado correctamente.

6.2.3.4. Cargas incrementales:

- **Cree un paquete SSIS que realice cargas incrementales. Solo los registros nuevos o cambiados del archivo CSV deben cargarse en la base de datos. Utilice la Transformación de Búsqueda (Lookup Transformation) para lograr esto.**

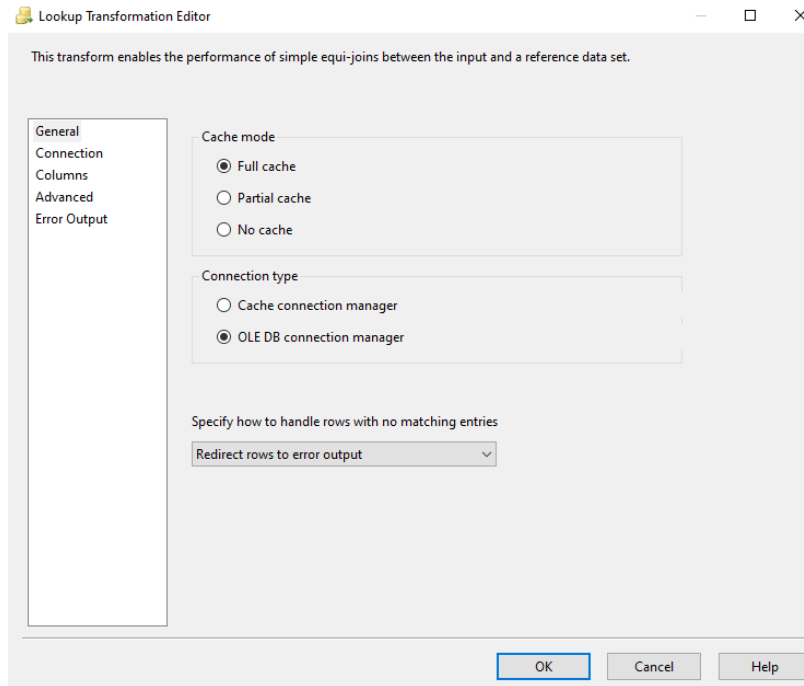
1. Arrastrar y soltar

- En la pestaña de Flujo de Datos, arrastre una Transformación de Búsqueda (Lookup Transformation) desde el SSIS Toolbox y conecte la salida de la transformación de Columna Derivada (Derived Column) a la entrada de la Transformación de Búsqueda (Lookup Transformation).



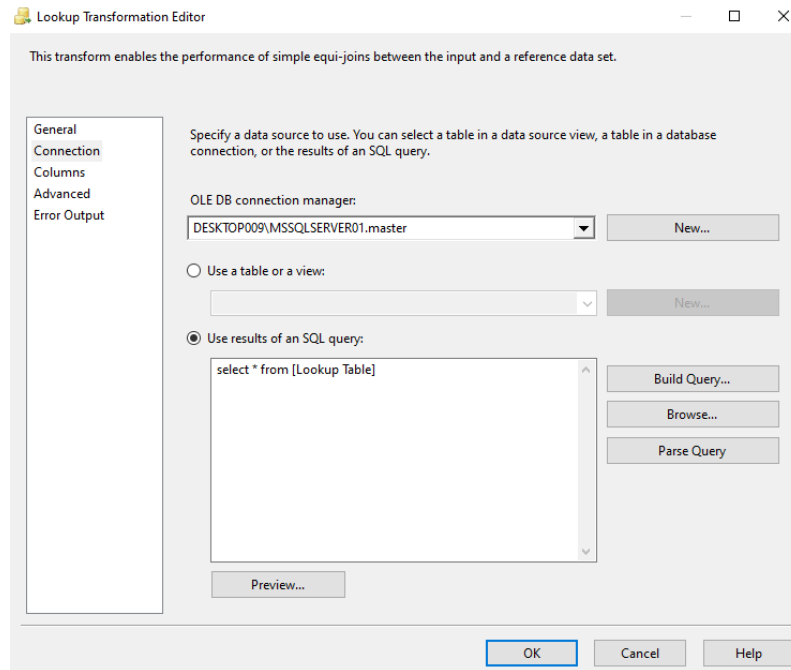
2. Configurar la Lookup Transformation:

- Haga doble clic en la Transformación de Búsqueda para abrir el editor.
- En la pestaña General, configure el comportamiento de No coincidencia (No Match Behavior) a "Redirect rows to no match output". Esto redirigirá las filas que no coincidan a una salida diferente que luego conectará al Destino OLE DB para insertar nuevos registros.



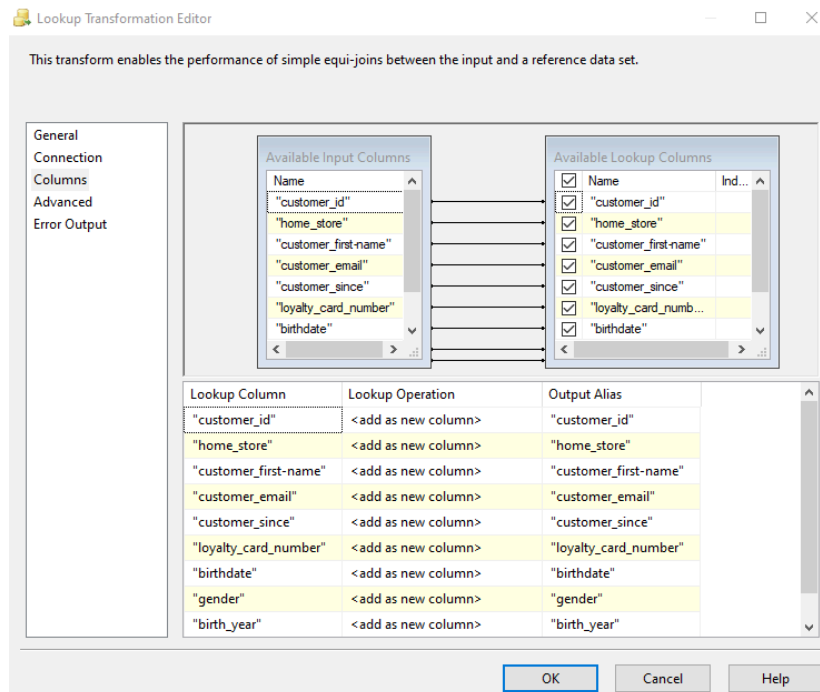
3. Conexión:

- Vaya a la pestaña de Conexión.
- Seleccione su administrador de conexión OLE DB de la lista.
- Especifique la tabla de destino donde desea verificar los registros existentes.



4. Columnas:

- Vaya a la pestaña de Columnas.
- Aquí, mapee las columnas de entrada a las columnas de búsqueda (las columnas en su base de datos). Generalmente, se mapearía una columna clave, como una columna de ID, para verificar coincidencias.



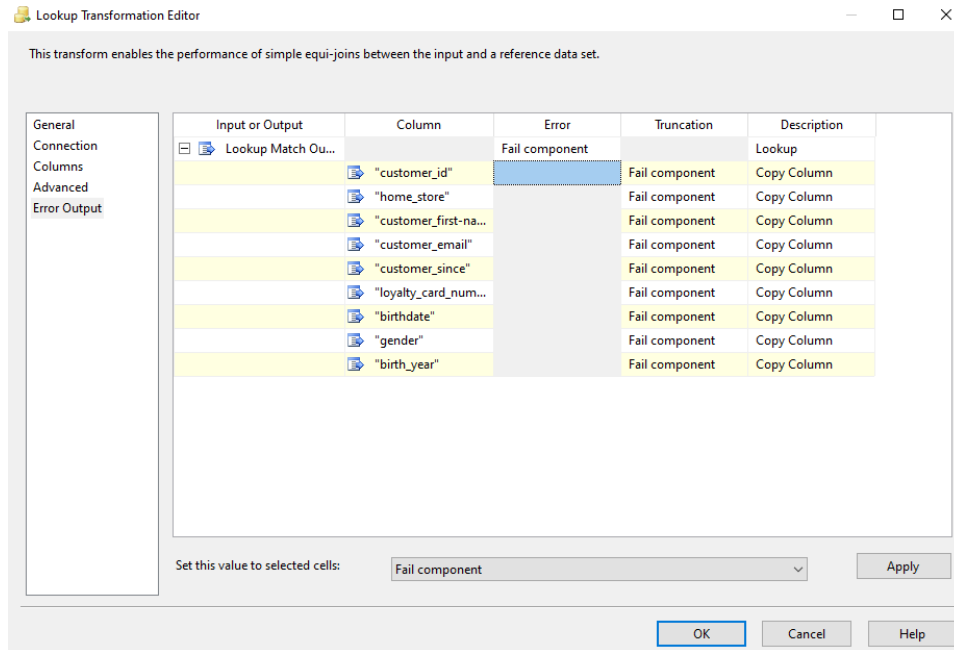
- Asegúrese de que la casilla bajo la columna de Búsqueda (Lookup column) esté marcada para la(s) columna(s) que está utilizando para encontrar coincidencias.

5. Avanzado(Opcional):

- Bajo la pestaña Avanzado, puede especificar opciones de almacenamiento en caché y otras configuraciones para optimizar el rendimiento de la operación de búsqueda. Sin embargo, las configuraciones predeterminadas suelen ser adecuadas para la mayoría de los escenarios.

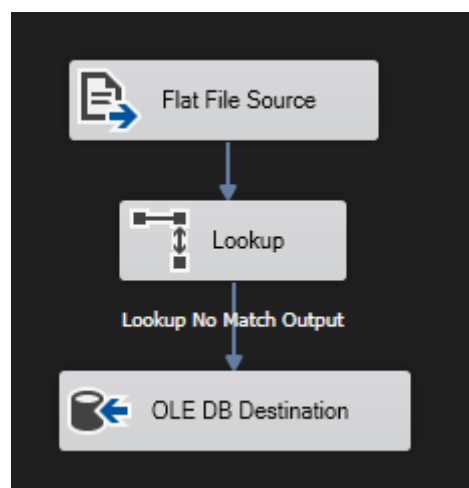
6. Errores Output (Opcional):

- Bajo la pestaña Salida de Errores (Error Output), puede especificar cómo desea manejar las filas con errores. Puede elegir ignorar el fallo, redirigir la fila o hacer que el componente falle



7. Output:

- Finalmente, conecte la 'Salida de No Coincidencia' (No Match Output) de la Transformación de Búsqueda (Lookup Transformation) al Destino OLE DB (OLE DB Destination). Esto enviará los nuevos registros (aquellos que no encontraron una coincidencia en la búsqueda) a su tabla de destino para su inserción.



8. Pruebas:

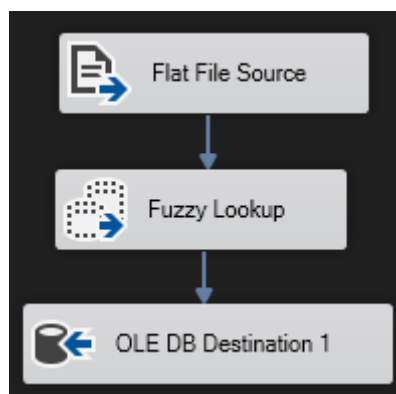
- Una vez configurado, ejecute su paquete SSIS para probar la Transformación de Búsqueda (Lookup Transformation) y asegúrese de que esté funcionando como se espera. Verifique su tabla de destino para ver si los registros nuevos y cambiados se han procesado correctamente.

6.2.3.5. Limpieza de datos:

- **Use los componentes de limpieza de datos en SSIS para limpiar los datos CSV antes de cargarlos. Por ejemplo, utilice la Transformación de Búsqueda Difusa (Fuzzy Lookup Transformation) para corregir nombres de productos que puedan tener errores ortográficos menores.**

1. Configuración del flujo de datos

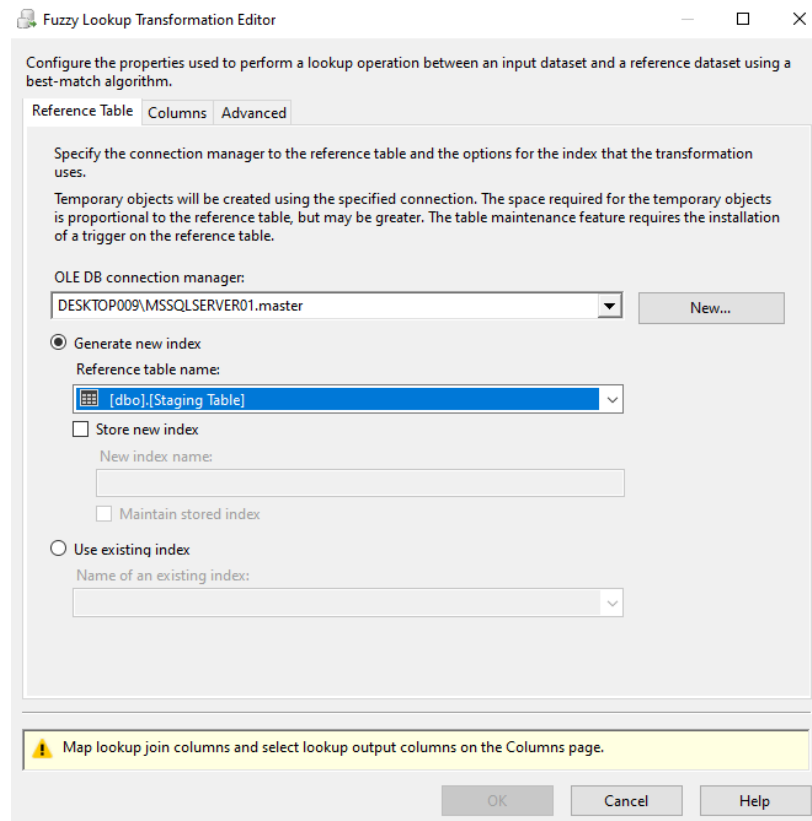
- Haga doble clic en la Tarea de Flujo de Datos (Data Flow Task) para abrir la pestaña de Flujo de Datos.
- Arrastre y suelte la Fuente de Archivo Plano (Flat File Source) desde el Toolbox a la superficie de diseño de Flujo de Datos y configúrela para que apunte a su archivo CSV.
- Arrastre y suelte la Transformación de Búsqueda Difusa (Fuzzy Lookup Transformation) desde el SSIS Toolbox a la superficie de diseño de Flujo de Datos.



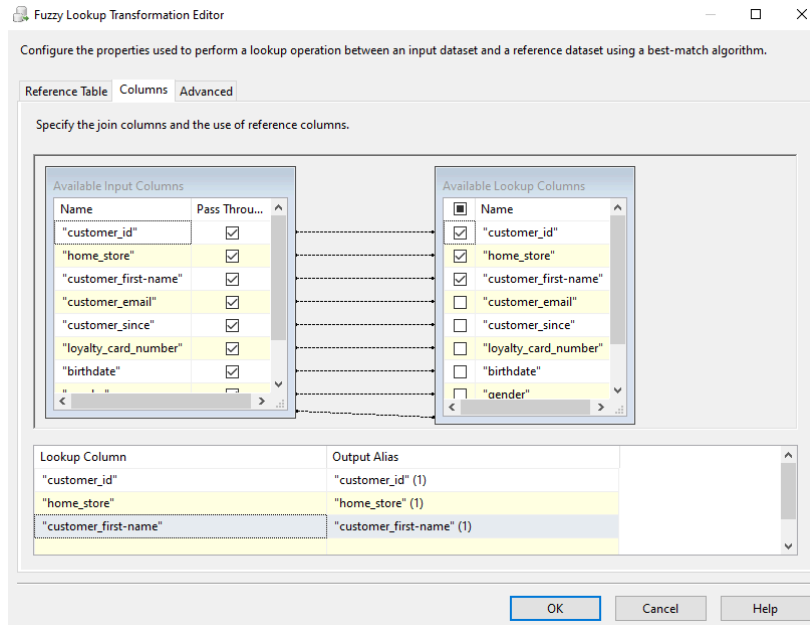
2. Configurar Fuzzy Lookup Transformación

- Haga doble clic en la Transformación de Búsqueda Difusa para abrir el Editor de Búsqueda Difusa.

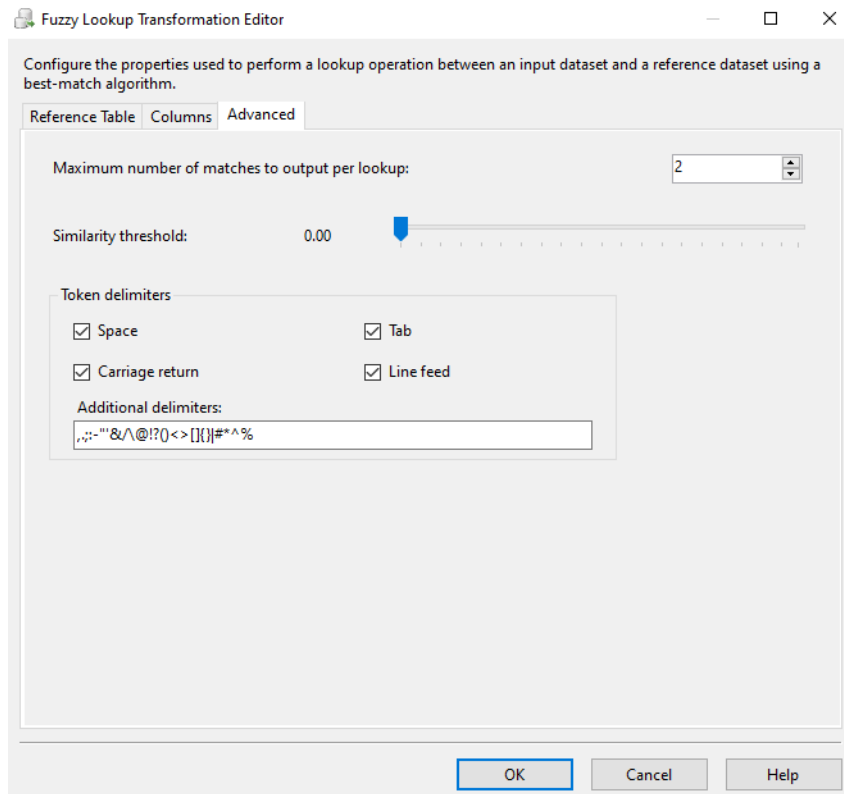
- En el Editor de Búsqueda Difusa:
- Configure el **administrador de conexión OLE DB** para que apunte a su base de datos.
- Especifique la tabla o vista de referencia en la pestaña **"Reference Table"**.



- Configure la pestaña "Columns" para mapear las columnas de entrada a las columnas de la tabla de referencia.



- Defina los parámetros de coincidencia difusa, como el umbral de similitud, en la pestaña **Advanced** tab¹.



3. Ejecute el paquete SSIS:

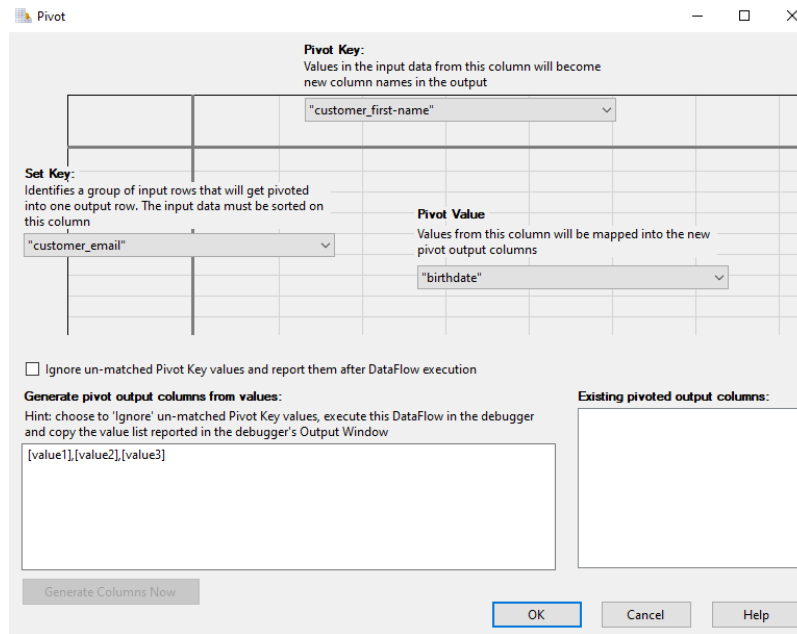
- Una vez configurado, ejecute el paquete SSIS.
- Monitoree los resultados de la ejecución para asegurarse de que los datos se limpien según lo esperado y se carguen en la base de datos.

6.2.3.6. Pivot Data:

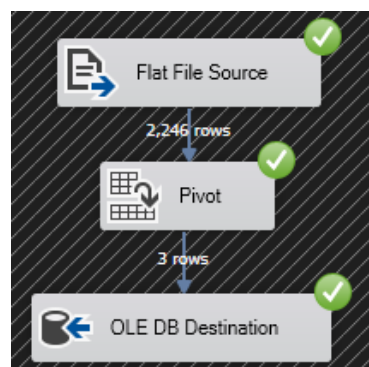
- **Suponga que el CSV contiene datos de ventas en un formato desnormalizado, con los meses como columnas (por ejemplo, Ene, Feb, Mar). Cree un paquete SSIS que pivote los datos a un formato más normalizado utilizando la Transformación de Pivote (Pivot Transformation).**

1. Añadir y configurar la Transformación de Pivote:

- Arrastre y suelte la Transformación de Pivote (Pivot Transformation) desde el toolbox a la región de flujo de datos entre la Fuente OLE DB (OLE DB Source) y el Destino OLE DB (OLE DB Destination).
- Haga doble clic en la Transformación de Pivote para abrir su editor.
- En el Editor de Transformación de Pivote, deberá especificar la Clave de Pivote (Pivot Key) (la columna sobre la cual pivotar, por ejemplo, la columna de mes), la Clave de Conjunto (Set Key) (una clave de agrupamiento) y el Valor de Pivote (Pivot Value) (el valor a agregar).



- Ejecute el paquete de Transformación de Pivote SSIS para verificar si la transformación funciona como se espera.



2. Generar columnas pivotadas:

- Haga doble clic en la Transformación de Pivote en la región de flujo de datos nuevamente.
- Haga clic en el botón "Generate Columns Now" para generar las columnas pivotadas que se agregarán automáticamente a la región de "Existing Pivoted Output Column".
- Haga clic en OK para cerrar la ventana Generate Output Column Results window³.

3. Guardar los datos pivotados en la base de datos:

- Arrastre la flecha de salida de la Transformación de Pivote SSIS al Destino OLE DB.
- Configure el Destino OLE DB para que apunte a la tabla deseada en su base de datos donde desea guardar los datos pivotados³.

6.2.3.7. Unpivot Data:

- **Haga lo inverso a lo anterior. Suponga que el CSV contiene datos de ventas en un formato normalizado. Cree un paquete SSIS que despivote los datos a un formato desnormalizado utilizando la Transformación de Despivote (Unpivot Transformation).**

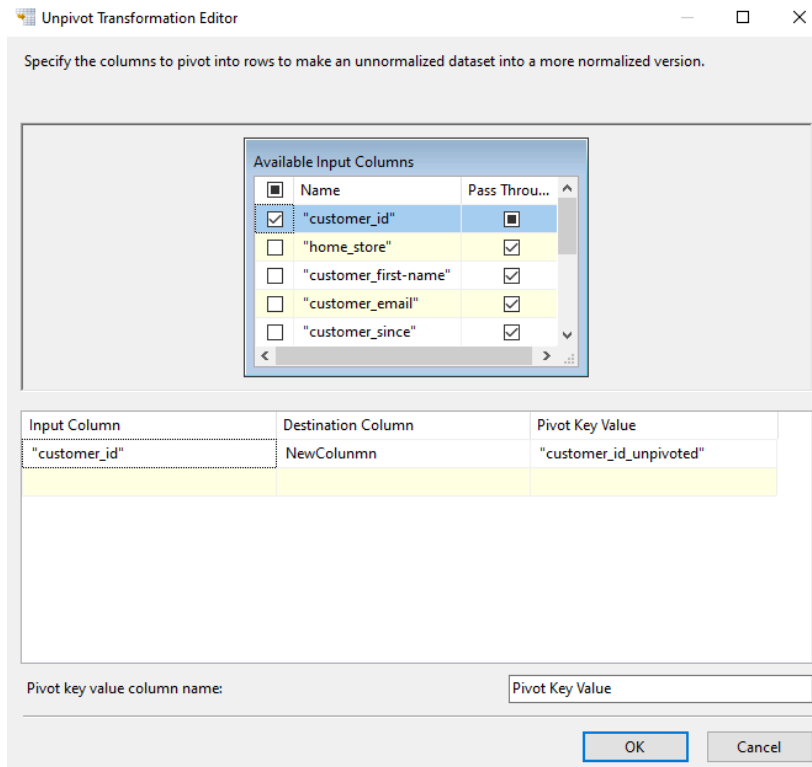
1. Configuración de su fuente::

- En el toolbox, busque la Fuente de Archivo Plano (Flat File Source), arrástrela y suéltela en el diseñador de flujo de datos.
- Haga doble clic en la Fuente de Archivo Plano y configúrela para que apunte a su base de datos y a la tabla o vista que contiene los datos normalizados.

2. Configuración de la Transformación de Despivote:

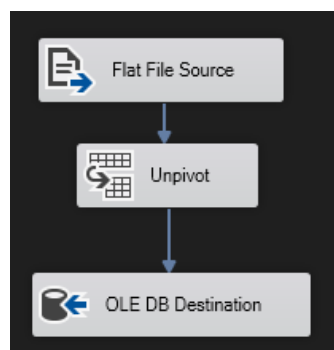
- Encuentre la Transformación de Despivote (Unpivot Transformation) en el toolbox, arrástrela y suéltela en el diseñador de flujo de datos.
- Conecte la Fuente de Archivo Plano a la Transformación de Despivote.
- Haga doble clic en la Transformación de Despivote para configurarla.
- En el Editor de Transformación de Despivote, en la sección "Available Input Columns", seleccione las columnas que desea despivotar. En la sección "Input Column", las columnas seleccionadas se agregarán automáticamente. Especifique el nombre de la "Destination Column" donde desea almacenar los valores y especifique el nombre para el "Pivot Key Value"

que se almacenará como valores de columna dentro de una nueva columna, por ejemplo, "Month". Especifique el nombre que desea ver para la columna de valor de pivote en la sección "Pivot Key Value Column Name"¹.



3. Configurar su destino:

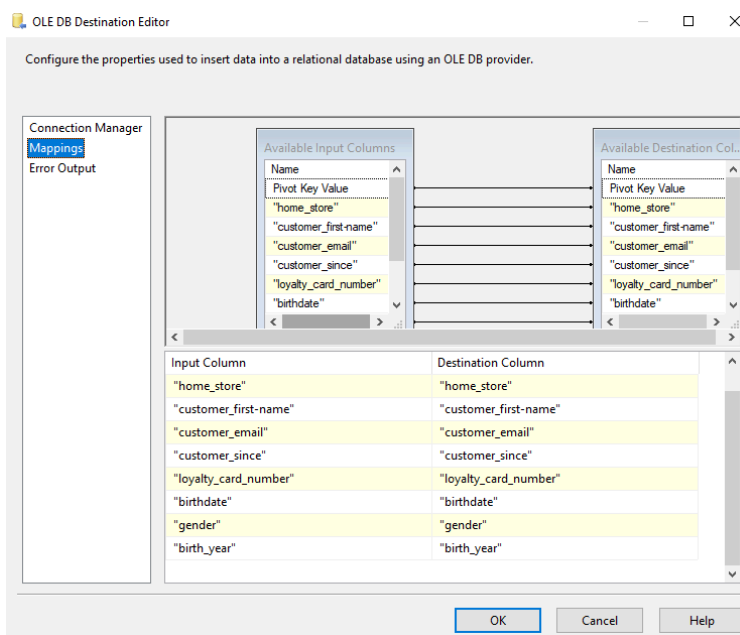
- Busque el Destino OLE DB en el toolbox, arrástrelo y suéltelo en el diseñador de flujo de datos.
- Conecte la Transformación de Despivote al Destino OLE DB.



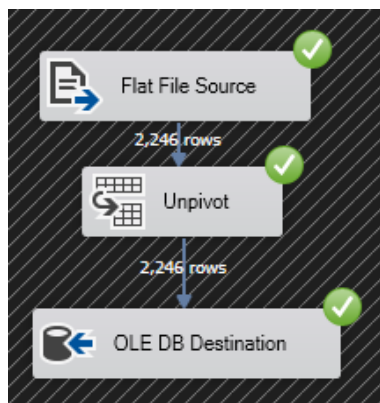
- Haga doble clic en el Destino OLE DB y configúrelo para que apunte a su base de datos y a la tabla donde desea almacenar los datos normalizados.

4. Mapear Columnas:

- Haga clic en la pestaña "Mappings" para verificar si las columnas de origen de la Transformación de Despivotado están mapeadas exactamente a las columnas de destino.



- Haga clic en OK para finalizar el diseño del paquete.
- Ejecute el paquete para ver los resultados de la transformación en su tabla de destino.

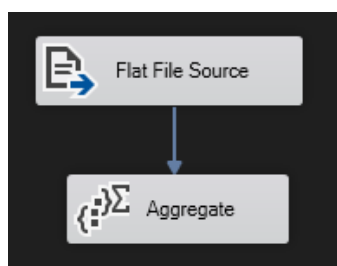


6.2.3.8. Agregación de datos:

- **Utilice la Transformación de Agregado (Aggregate Transformation) para calcular totales o promedios a partir de los datos CSV antes de cargarlos en la base de datos.**

1. Añadir la Fuente y la Transformación de Agregado:

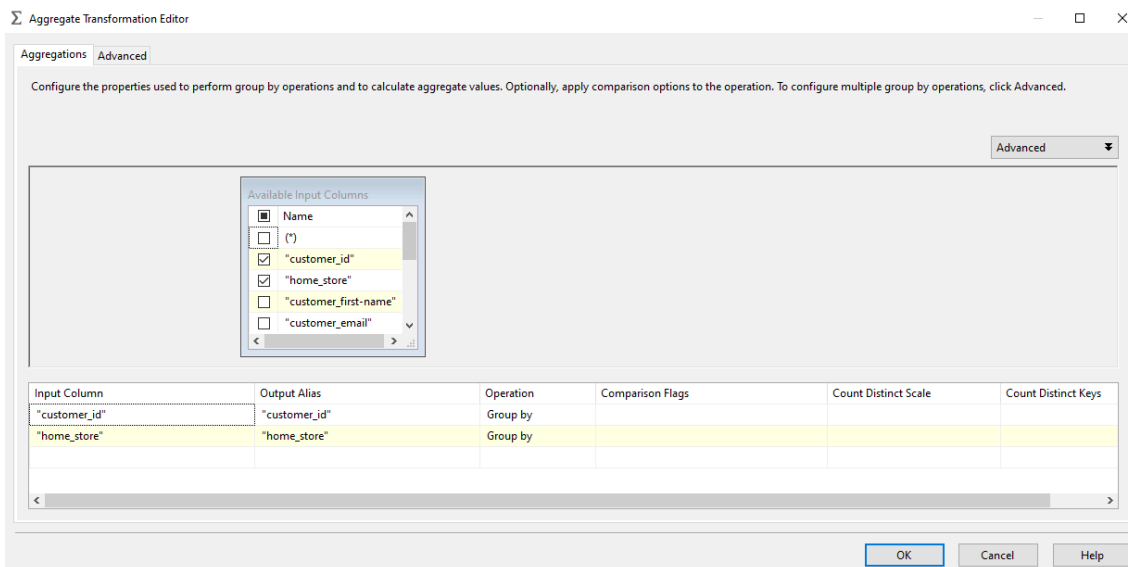
- Drag and drop a Flat File Source (to read the CSV data) and an Aggregate Transformation from the SSIS Toolbox to the Data Flow region.
- Conecte la salida de la Fuente de Flat File Source a la input of the Aggregate Transformation².



2. Configuración de la Transformación de Agregado::

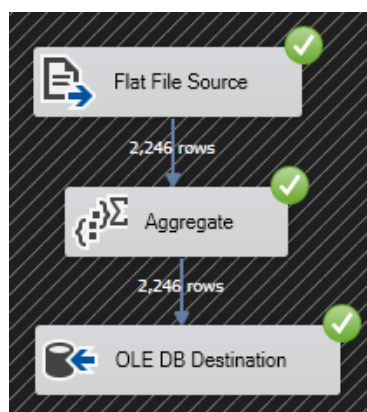
- Haga doble clic en la Transformación de Agregado para abrir el editor.
- En el editor, puede especificar las operaciones como suma, promedio, cuenta, etc., en las columnas necesarias de sus datos CSV.

- También puede usar la cláusula GROUP BY para especificar grupos sobre los cuales realizar la agregación³.



3. Ejecución y prueba:

- Después de configurar la Transformación de Agregado, puede conectar su salida a su destino deseado, como un Destino OLE DB, para cargar los datos agregados en su base de datos.
- Una vez que todo esté configurado, ejecute el paquete SSIS y verifique los datos en el destino para asegurarse de que los procesos de agregación y carga se hayan realizado correctamente⁴.

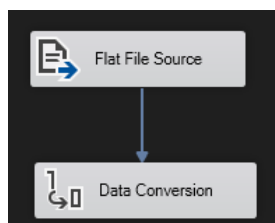


6.2.3.9. Conversión de datos:

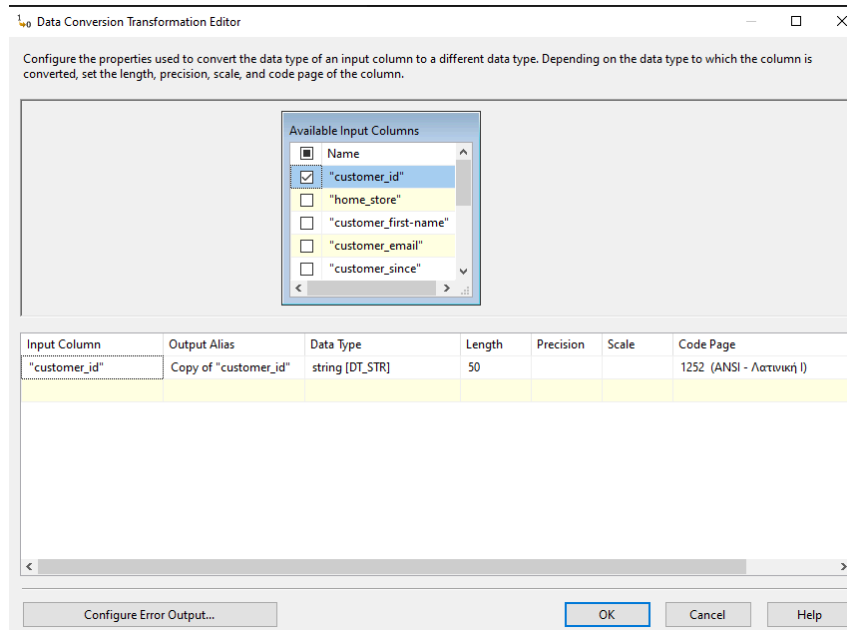
- Suponga que el CSV contiene datos numéricos almacenados como texto. Cree un paquete SSIS que utilice la Transformación de Conversión de Datos (Data Conversion Transformation) para convertir estos campos de texto en tipos de datos numéricos apropiados antes de cargarlos en la base de datos.

1. Añadir y configurar la Transformación de Conversión de Datos:

- Arrastre y suelte la Transformación de Conversión de Datos (Data Conversion) desde el Toolbox al panel de diseño.
- Conecte la flecha azul desde su componente de origen a la Transformación de Conversión de Datos.



- Haga doble clic en la Transformación de Conversión de Datos para abrir el editor.
- En el editor, mapee las columnas de entrada (campos de texto) a las columnas de salida, especificando los tipos de datos numéricos deseados para las columnas de salida.¹



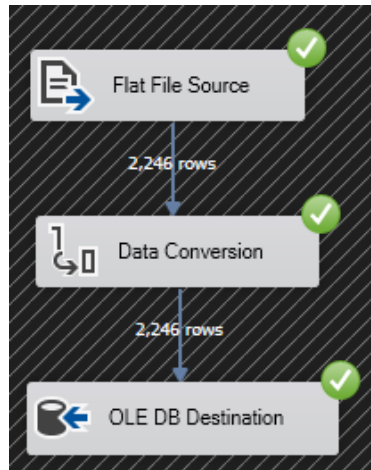
- La Transformación de Conversión de Datos creará nuevas columnas de salida con los tipos de datos especificados, dejando las columnas originales sin cambios.²

Configurar Your Destination:

- Arrastre y suelte su componente de destino (por ejemplo, Destino OLE DB si está cargando datos en una base de datos SQL Server) desde el Toolbox al panel de diseño.
- Conecte la flecha azul desde la Transformación de Conversión de Datos a su componente de destino.
- Haga doble clic en el componente de destino para configurarlo, especificando el administrador de conexión, la tabla de destino y los mapeos entre las columnas de entrada y las columnas de destino.

3. Ejecute su paquete SSIS::

- Una vez que haya configurado todos los componentes, ejecute su paquete SSIS para probar el proceso de conversión y carga de datos.
- Verifique su base de datos de destino para asegurarse de que los datos se hayan convertido y cargado correctamente.



6.2.4. Paso 4 -Almacén de datos usando SQL

Ahora, vamos a continuar con algunos ejercicios para practicar comandos SQL y, de esta manera, crear un almacén de datos.

6.2.4.1. Intente escribir algunas consultas básicas para obtener::

- Listar todos los productos con su precio de venta actual.

```
SELECT name AS product_name,
       current_retail_price
FROM product
```

product_name	current_retail_price
Brazilian - Organic	\$18.00
Our Old Time Diner Blend	\$18.00
Espresso Roast	\$14.75
Primo Espresso Roast	\$20.45
Columbian Medium Roast	\$15.00
Ethiopia	\$21.00
Jamacian Coffee River	\$19.75
Civet Cat	\$45.00
Organic Decaf Blend	\$22.50
Guatemalan Sustainably Grown	\$10.00

- Encontrar las ventas totales para cada tienda..

```
SELECT store_id,
       SUM(line_item_amount) AS total_sales
FROM sales
GROUP BY store_id
```

store_id	total_sales
3	77,210.73
5	76,894.47
8	79,528.25

- Listar todos los clientes que han sido leales desde una fecha específica.

```
SELECT customer.fullname,
       customer.email
FROM customer
WHERE customer_since <= '2017-12-31'
```

fullname	email
Kelly Key	Venus@adipiscing.edu
Clark Schroeder	Nora@fames.gov
Elvis Cardenas	Brianna@tellus.edu
Rafael Estes	Ina@non.gov
Colin Lynn	Dale@Integer.com
Igor Beach	Caleb@morbi.net
Scott Holden	Yen@Integer.edu
Keegan Ayala	Tana@sociis.com
Amir Byers	Madeson@malesuada.us
Magee Malone	Anjolie@sapien.gov

6.2.4.2. Intente unir datos de diferentes tablas:

- Unir la tabla de Ventas con la tabla de Productos para mostrar los detalles del producto para cada venta.

```
SELECT product.category,
       product.name,
       product.type,
       product.description,
       sales.transaction_date,
```

```

sales.transaction_time,
sales.store_id,
sales.order,
sales.line_item_id,
sales.quantity,
sales.unit_price
FROM product
INNER JOIN sales ON product.id = sales.product_id
    
```

Result #1 (49,893r x 11c)

category	name	type	description	transac...	transac...	store_id	order	line_item_id	quantity	unit_price
Tea	Peppermint Rg	Brewed herbal tea	A cool and refreshing cup.	2019-04-03	13:26:51	8	1	1	1	2.5
Bakery	Cranberry Scone	Scone	Like Grandma used to make	2019-04-03	13:26:51	8	1	5	1	3.25
Drinking Cho...	Dark chocolate Rg	Hot chocolate	Slightly bitter, but still ver...	2019-04-03	15:55:36	5	1	1	1	3.5
Tea	Spicy Eye Opener ...	Brewed Chai tea	When you need your eyes...	2019-04-05	17:27:40	3	1	1	1	2.55
Bakery	Cranberry Scone	Scone	Like Grandma used to make	2019-04-05	17:27:40	3	1	5	1	3.25
Coffee	Our Old Time Diner...	Drip coffee	An honest cup a coffee.	2019-04-06	16:05:09	3	1	1	2	2.5
Coffee	Ethiopia Sm	Gourmet brewed coffee	A bold cup when you want...	2019-04-07	07:08:39	5	1	1	2	2.2
Tea	Earl Grey Lg	Brewed Black tea	Tradition in a cup.	2019-04-07	08:46:29	8	1	1	1	3.0
Bakery	Hazelnut Biscotti	Biscotti	Crunch!	2019-04-07	08:46:29	8	1	4	1	3.25
Tea	Peppermint Lg	Brewed herbal tea	A cool and refreshing cup.	2019-04-07	10:16:40	3	1	1	1	3.0

- Unir la tabla de Ventas con la tabla de Personal para averiguar qué miembro del personal realizó más ventas.

```

SELECT staff.first_name,
       staff.last_name,
       COUNT(sales.order) AS total_sales
FROM sales
INNER JOIN staff ON staff.id = sales.staff_id
GROUP BY staff.first_name,
         staff.last_name
ORDER BY total_sales DESC
    
```

staff (25r × 3c)		
first_name	last_name	total_sales
Britanni	Jorden	5,929
Kylie	Candace	4,085
Pandora	Neville	3,648
Joelle	Christen	3,629
Reed	Eve	3,198
Ronan	Magee	3,127
Quail	Octavia	3,059
Damon	Sasha	2,770
Amela	Chadwick	2,430
Tamekah	Maya	2,349

6.2.4.3. Intente realizar algunos cálculos utilizando funciones de agregación

- **Calcular las ventas totales para cada producto.**

```
SELECT product_id,
       SUM(line_item_amount) AS total_sales
FROM sales
GROUP BY product_id
```

sales (80r × 2c)	
product_id	total_sales
1	1,260.0
2	1,170.0
3	781.75
4	961.15
5	750.0
6	1,470.0
7	1,007.25
8	4,275.0
9	1,521.5
10	480.0

Alternative solution with joins:

```

SELECT product.id,
       product.name,
       SUM(sales.line_item_amount) AS total_sales
FROM product
INNER JOIN sales ON product.id = sales.product_id
GROUP BY product.name
    
```

id	name	total_sales
73	Almond Croissant	2,415.94
1	Brazilian - Organic	1,260.0
27	Brazilian Lg	4,991.0
26	Brazilian Rg	4,431.0
25	Brazilian Sm	3,223.0
40	Cappuccino	5,456.25
41	Cappuccino Lg	6,056.25
63	Carmel syrup	667.2
21	Chili Mayan	666.5
76	Chocolate Chip Biscotti	2,219.9

- Encontrar las ventas promedio por día.

```

SELECT transaction_date,
       AVG(line_item_amount) AS total_sales
FROM sales
GROUP BY transaction_date
    
```

transaction_date	total_sales
2019-04-01	4.5045289855
2019-04-02	4.4377899878
2019-04-03	4.4333036245
2019-04-04	4.4540854808
2019-04-05	4.4285495716
2019-04-06	4.4462457338
2019-04-07	4.5444359126
2019-04-08	4.7873997833
2019-04-09	4.7555378486
2019-04-10	4.5664516129

- Contar el número de ventas para cada tienda.


```

SELECT store_id,
       COUNT(transaction_id) total_sales
FROM sales
GROUP BY store_id
    
```

store_id	total_sales
3	16,828
5	15,994
8	17,071

Alternative solution with joins:

```

SELECT store.id,
       COUNT(transaction_id) total_sales,
       store.neighborhood
FROM sales
RIGHT JOIN store ON store.id = sales.store_id
GROUP BY store.id,
         store.neighborhood
    
```

id	total_sales	neighborhood
2	0	Jamaica
3	16,828	Astoria
4	0	Gowanus
5	15,994	Lower Manhattan
6	0	Lower East Side
7	0	Upper East Side
8	17,071	Hell's Kitchen
9	0	Chelsea
10	0	Greenwich Village

6.2.4.4. Ahora, avancemos con una declaración SQL más compleja - Subconsultas (Subqueries):

- **Encontrar productos que nunca han sido vendidos.**

```

SELECT *
FROM product
    
```

```
WHERE id NOT IN (SELECT product_id
                  FROM sales)
```

id	group	category	type	name	description	unit_of_me...	current_...	current_...	tax_ex...	promo...	new_pr...
62	Beverages	Drinking Cho...	Hot chocolate	Snow Day Hot Cho...	Added marshmallow...	8 oz	2.25	\$3.00	Y	N	N
66	Beverages	Coffee	Seasonal drink	Pumpkin Spice Latte	Boo, its that time of ...	1.5 oz	0.99	\$4.95	Y	N	N
67	Beverages	Coffee	Seasonal drink	Pumpkin Spice Latt...	Boo, its that time of ...	3.0 oz	1.19	\$5.95	Y	Y	N
68	Beverages	Drinking Cho...	Seasonal drink	Happy Holidays ho...	Candy cane and hot...	8 oz	2.81	\$3.75	Y	Y	Y
80	Merchandise	Branded	Clothing	I Need My Bean! T...	keep your head bea...	single	7.36	\$23.00	N	N	N
85	Beverages	Coffee	Specialty cof...	Rio Nights	2 shots of Ouro Bra...	6 oz	1.2	\$6.00	Y	N	Y
88	Beverages	Coffee	Barista Esp...	Ouro Brasileiro sho...	Ouro promo	16 oz	0.42	\$2.10	Y	Y	N
89	Food	Bakery	Scone	Ginger Scone promo	Little bit of spice	single	1.72	\$2.65	N	Y	N

Alternative solution with joins:

```
SELECT *
FROM product
LEFT JOIN sales ON product.id = sales.product_id
WHERE sales.product_id IS NULL
```

id	group	category	type	name	description	unit_of_measure	current...	current_f...	tax_...	pro...	new...	trans...	trans...	transa...	store_id	st
62	Beverages	Drinking Cho...	Hot chocolate	Snow Day Hot ...	Added marshallo...	8 oz	2.25	\$3.00	Y	N	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
66	Beverages	Coffee	Seasonal dr...	Pumpkin Spice ...	Boo, its that time ...	1.5 oz	0.99	\$4.95	Y	N	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
67	Beverages	Coffee	Seasonal dr...	Pumpkin Spice ...	Boo, its that time ...	3.0 oz	1.19	\$5.95	Y	Y	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
68	Beverages	Drinking Cho...	Seasonal dr...	Happy Holiday...	Candy cane and h...	8 oz	2.81	\$3.75	Y	Y	Y	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
80	Merchan...	Branded	Clothing	I Need My Bea...	keep your head b...	single	7.36	\$23.00	N	N	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
85	Beverages	Coffee	Specialty c...	Rio Nights	2 shots of Ouro Br...	6 oz	1.2	\$6.00	Y	N	Y	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
88	Beverages	Coffee	Barista Esp...	Ouro Brasileiro...	Ouro promo	16 oz	0.42	\$2.10	Y	Y	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
89	Food	Bakery	Scone	Ginger Scone ...	Little bit of spice	single	1.72	\$2.65	N	Y	N	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

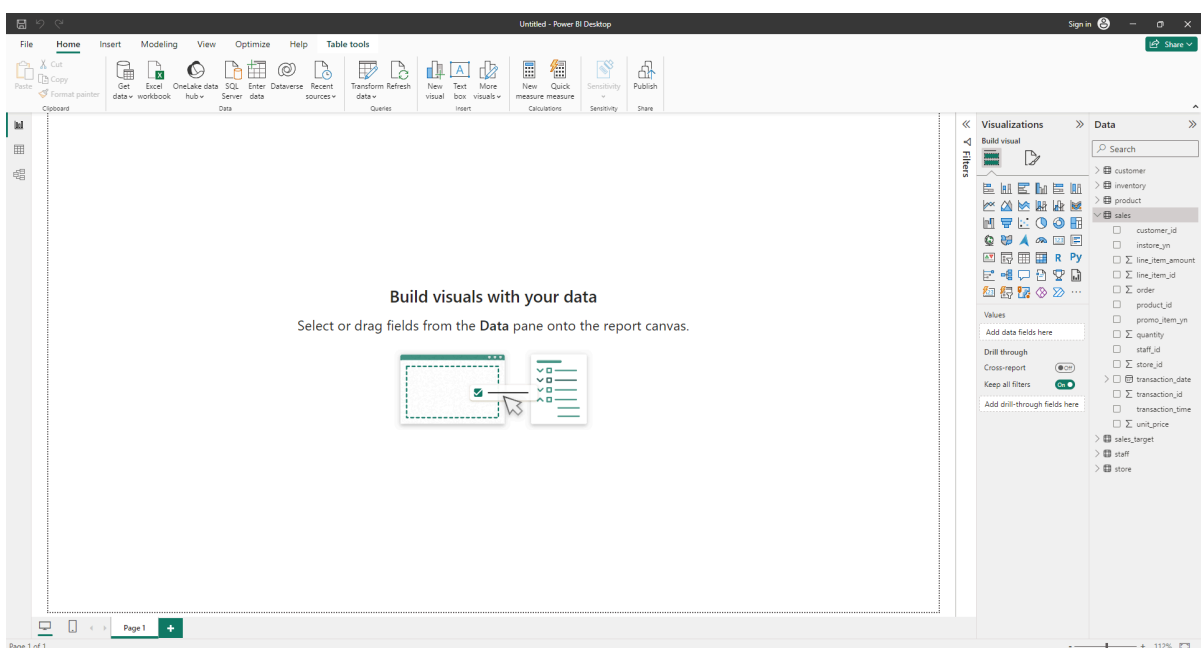
- Listar las tiendas que han superado su objetivo de ventas.

```
SELECT sales_target.store_id,
       NVL(SUM(sales.line_item_amount),0) AS total_sales,
       sales_target.total_goal
FROM sales
RIGHT JOIN sales_target
ON sales.store_id = sales_target.store_id
GROUP BY sales_target.store_id
HAVING SUM(sales.line_item_amount)>sales_target.total_goal
```

store_id	total_sales	total_goal
3	77,210.73	18,000
5	76,894.47	25,000
8	79,528.25	22,500

6.2.5. Paso 5 -Visualización de datos usando Power BI

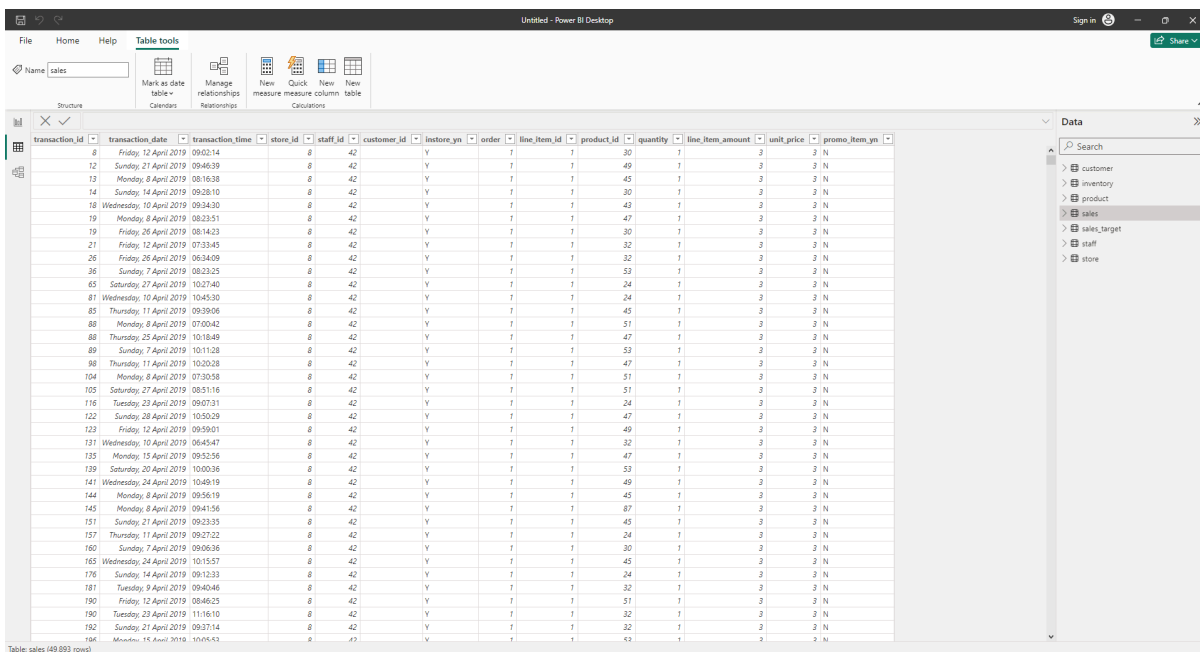
Cuando importe sus datos correctamente desde la base de datos SQL, notará que en la tabla de Ventas en Datos, no tiene una opción de Ventas Totales, pero necesitará una para poder continuar. PowerBI le permite transformar sus datos, y en este caso necesitamos agregar esa columna de Ventas Totales para cumplir con nuestros próximos objetivos.



Para crear esa columna adicional, siga estos pasos:

1. Haga clic en **Vista de tabla** en el panel lateral a la izquierda de su pantalla.
2. En la nueva pantalla, asegúrese de seleccionar **Ventas en Datos**.

Ahora debería tener esta pantalla frente a usted.

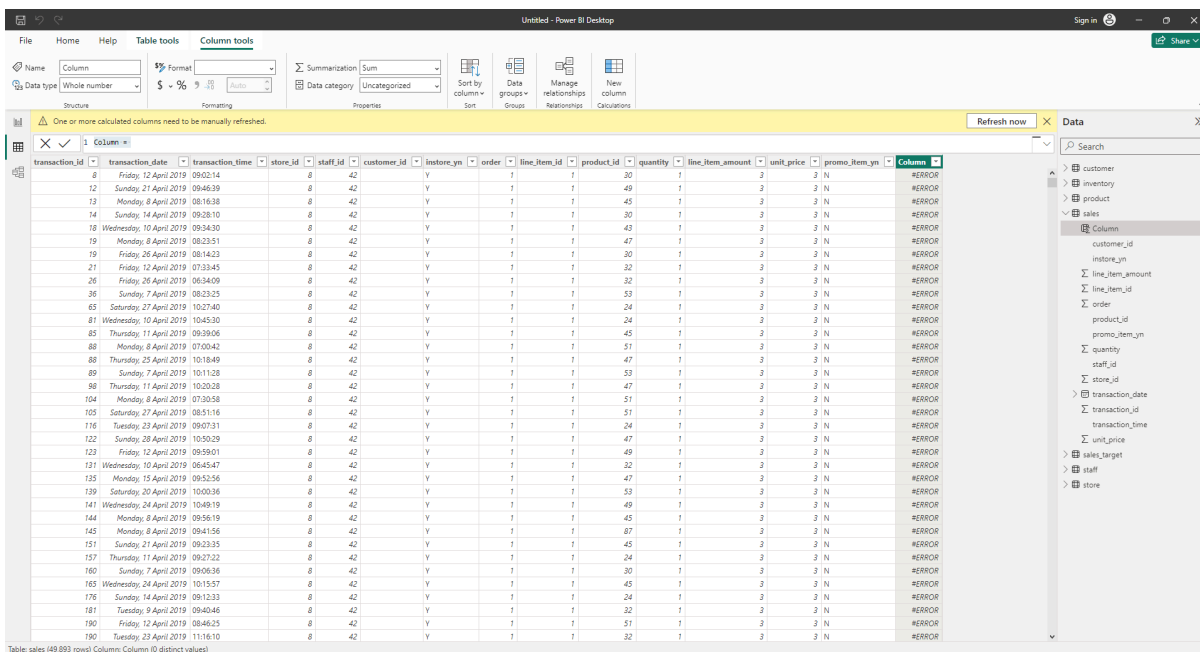


transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	instore_yn	order	line_item_id	product_id	quantity	line_item_amount	unit_price	promo_item_yn
8	Friday, 12 April 2019	09:02:14	8	42	Y	Y	1	1	30	1	3	3	N
12	Sunday, 21 April 2019	09:46:39	8	42	Y	Y	1	1	49	1	3	3	N
13	Monday, 8 April 2019	08:16:38	8	42	Y	Y	1	1	45	1	3	3	N
14	Sunday, 14 April 2019	09:28:10	8	42	Y	Y	1	1	30	1	3	3	N
18	Wednesday, 10 April 2019	09:34:30	8	42	Y	Y	1	1	43	1	3	3	N
19	Monday, 8 April 2019	08:23:51	8	42	Y	Y	1	1	47	1	3	3	N
19	Friday, 26 April 2019	08:14:23	8	42	Y	Y	1	1	30	1	3	3	N
21	Friday, 12 April 2019	07:33:45	8	42	Y	Y	1	1	32	1	3	3	N
26	Friday, 26 April 2019	06:34:09	8	42	Y	Y	1	1	32	1	3	3	N
36	Sunday, 7 April 2019	08:23:25	8	42	Y	Y	1	1	53	1	3	3	N
65	Saturday, 27 April 2019	10:27:40	8	42	Y	Y	1	1	24	1	3	3	N
81	Wednesday, 10 April 2019	10:43:30	8	42	Y	Y	1	1	24	1	3	3	N
85	Thursday, 11 April 2019	09:59:06	8	42	Y	Y	1	1	45	1	3	3	N
88	Monday, 8 April 2019	07:00:42	8	42	Y	Y	1	1	51	1	3	3	N
88	Thursday, 25 April 2019	10:18:49	8	42	Y	Y	1	1	47	1	3	3	N
89	Sunday, 7 April 2019	10:11:28	8	42	Y	Y	1	1	53	1	3	3	N
98	Thursday, 11 April 2019	10:20:28	8	42	Y	Y	1	1	47	1	3	3	N
104	Monday, 8 April 2019	07:30:58	8	42	Y	Y	1	1	51	1	3	3	N
105	Saturday, 27 April 2019	08:51:16	8	42	Y	Y	1	1	51	1	3	3	N
116	Tuesday, 23 April 2019	09:07:31	8	42	Y	Y	1	1	24	1	3	3	N
122	Sunday, 28 April 2019	10:50:29	8	42	Y	Y	1	1	47	1	3	3	N
123	Friday, 12 April 2019	09:59:01	8	42	Y	Y	1	1	49	1	3	3	N
131	Wednesday, 10 April 2019	06:45:47	8	42	Y	Y	1	1	32	1	3	3	N
135	Monday, 15 April 2019	09:52:56	8	42	Y	Y	1	1	47	1	3	3	N
139	Saturday, 20 April 2019	10:00:36	8	42	Y	Y	1	1	53	1	3	3	N
141	Wednesday, 24 April 2019	10:49:19	8	42	Y	Y	1	1	49	1	3	3	N
144	Monday, 8 April 2019	09:56:19	8	42	Y	Y	1	1	45	1	3	3	N
145	Monday, 8 April 2019	09:41:56	8	42	Y	Y	1	1	87	1	3	3	N
151	Sunday, 21 April 2019	09:23:25	8	42	Y	Y	1	1	45	1	3	3	N
157	Thursday, 11 April 2019	09:27:22	8	42	Y	Y	1	1	24	1	3	3	N
160	Sunday, 7 April 2019	09:06:36	8	42	Y	Y	1	1	30	1	3	3	N
165	Wednesday, 24 April 2019	10:15:57	8	42	Y	Y	1	1	45	1	3	3	N
176	Sunday, 14 April 2019	09:12:33	8	42	Y	Y	1	1	24	1	3	3	N
181	Tuesday, 9 April 2019	09:40:46	8	42	Y	Y	1	1	32	1	3	3	N
190	Friday, 12 April 2019	08:40:23	8	42	Y	Y	1	1	51	1	3	3	N
190	Tuesday, 23 April 2019	11:16:10	8	42	Y	Y	1	1	32	1	3	3	N
192	Sunday, 21 April 2019	09:37:14	8	42	Y	Y	1	1	32	1	3	3	N
196	Monday, 15 April 2019	10:05:43	8	42	Y	Y	1	1	43	1	3	3	N

Notará que tiene dos columnas, una titulada "**cantidad**" y otra titulada "**precio_unitario**" y los datos en estas dos columnas cambian de valor a medida que se desplaza por su tabla. Pero no hay una columna de "**Ventas Totales**", que necesitará para producir las tablas necesarias para los siguientes ejercicios.

Puede crear esta columna en PowerBI usando los siguientes pasos:

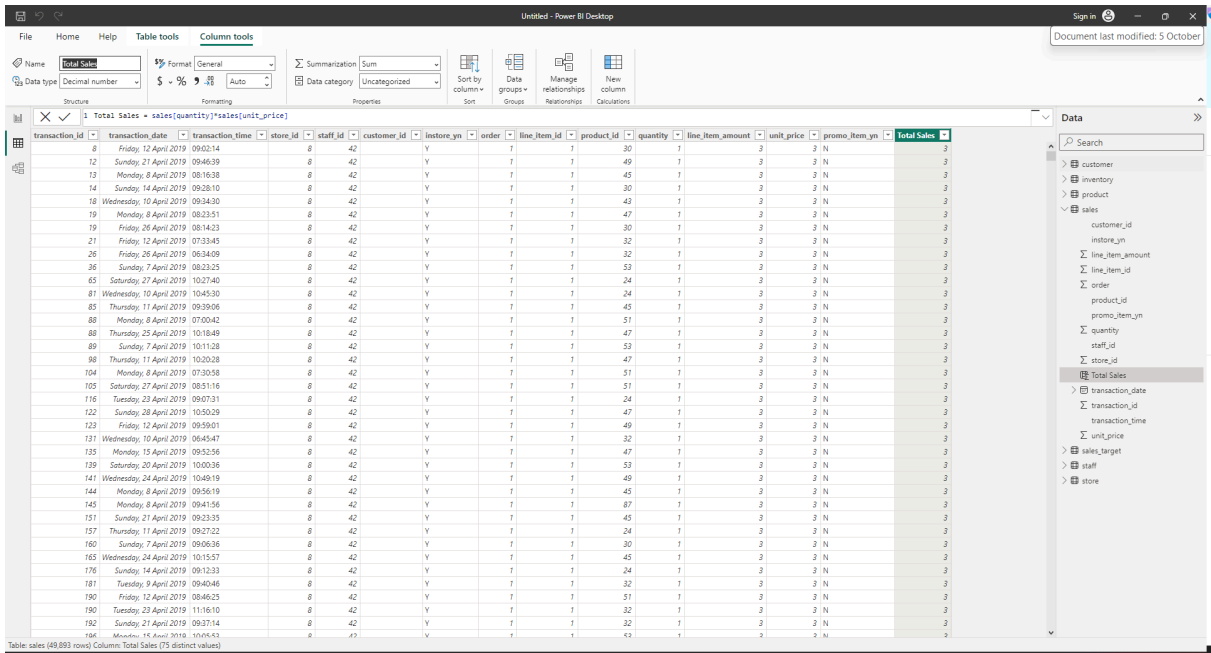
1. Haga clic en la **pestaña Herramientas** de tabla en la parte superior de la pantalla.
2. Haga clic en la opción **Nueva columna** ubicada bajo Herramientas de tabla.
3. PowerBI creará la nueva columna y le pedirá que ingrese la fórmula para esta nueva columna.



transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	instore_yn	order	line_item_id	product_id	quantity	line_item_amount	unit_price	promo_item_yn	Column
8	Friday, 12 April 2019	09:02:14	8	42		Y	1	1	30	1	2	3	N	#ERROR
12	Sunday, 21 April 2019	09:46:39	8	42		Y	1	1	49	1	2	3	N	#ERROR
13	Monday, 8 April 2019	08:16:38	8	42		Y	1	1	45	1	2	3	N	#ERROR
14	Sunday, 14 April 2019	09:28:10	8	42		Y	1	1	30	1	2	3	N	#ERROR
18	Wednesday, 10 April 2019	09:34:30	8	42		Y	1	1	43	1	2	3	N	#ERROR
19	Monday, 8 April 2019	08:23:51	8	42		Y	1	1	47	1	2	3	N	#ERROR
19	Friday, 26 April 2019	08:14:23	8	42		Y	1	1	30	1	2	3	N	#ERROR
21	Friday, 12 April 2019	07:33:45	8	42		Y	1	1	32	1	2	3	N	#ERROR
26	Friday, 26 April 2019	06:54:09	8	42		Y	1	1	32	1	2	3	N	#ERROR
36	Sunday, 7 April 2019	08:23:25	8	42		Y	1	1	53	1	2	3	N	#ERROR
65	Saturday, 27 April 2019	10:27:40	8	42		Y	1	1	24	1	2	3	N	#ERROR
81	Wednesday, 10 April 2019	10:45:30	8	42		Y	1	1	24	1	2	3	N	#ERROR
85	Thursday, 11 April 2019	09:39:06	8	42		Y	1	1	45	1	2	3	N	#ERROR
88	Monday, 8 April 2019	07:02:42	8	42		Y	1	1	51	1	2	3	N	#ERROR
88	Thursday, 25 April 2019	10:18:49	8	42		Y	1	1	47	1	2	3	N	#ERROR
89	Sunday, 7 April 2019	10:11:28	8	42		Y	1	1	53	1	2	3	N	#ERROR
98	Thursday, 11 April 2019	10:20:28	8	42		Y	1	1	47	1	2	3	N	#ERROR
104	Monday, 8 April 2019	07:30:58	8	42		Y	1	1	51	1	2	3	N	#ERROR
105	Saturday, 27 April 2019	08:51:16	8	42		Y	1	1	51	1	2	3	N	#ERROR
116	Tuesday, 23 April 2019	09:07:31	8	42		Y	1	1	24	1	2	3	N	#ERROR
122	Sunday, 28 April 2019	10:50:29	8	42		Y	1	1	47	1	2	3	N	#ERROR
123	Friday, 12 April 2019	09:59:01	8	42		Y	1	1	49	1	2	3	N	#ERROR
131	Wednesday, 10 April 2019	06:45:47	8	42		Y	1	1	32	1	2	3	N	#ERROR
135	Monday, 15 April 2019	09:52:56	8	42		Y	1	1	47	1	2	3	N	#ERROR
139	Saturday, 20 April 2019	10:00:36	8	42		Y	1	1	53	1	2	3	N	#ERROR
141	Wednesday, 24 April 2019	10:40:19	8	42		Y	1	1	49	1	2	3	N	#ERROR
144	Monday, 8 April 2019	09:56:19	8	42		Y	1	1	45	1	2	3	N	#ERROR
145	Monday, 8 April 2019	09:41:56	8	42		Y	1	1	87	1	2	3	N	#ERROR
151	Sunday, 21 April 2019	09:23:35	8	42		Y	1	1	45	1	2	3	N	#ERROR
157	Thursday, 11 April 2019	09:27:22	8	42		Y	1	1	24	1	2	3	N	#ERROR
160	Sunday, 7 April 2019	09:48:36	8	42		Y	1	1	30	1	2	3	N	#ERROR
165	Wednesday, 24 April 2019	10:15:57	8	42		Y	1	1	45	1	2	3	N	#ERROR
176	Sunday, 14 April 2019	09:12:33	8	42		Y	1	1	24	1	2	3	N	#ERROR
181	Tuesday, 9 April 2019	09:40:46	8	42		Y	1	1	32	1	2	3	N	#ERROR
190	Friday, 12 April 2019	08:46:25	8	42		Y	1	1	51	1	2	3	N	#ERROR
190	Tuesday, 23 April 2019	11:16:10	8	42		Y	1	1	32	1	2	3	N	#ERROR

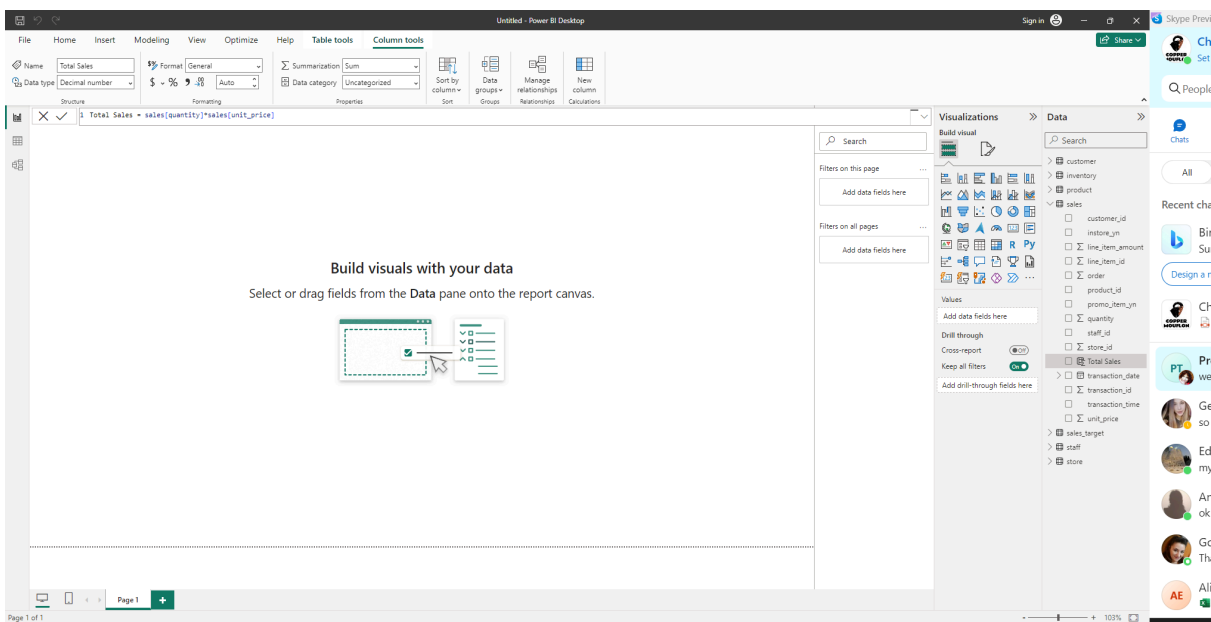
4. Escriba lo siguiente (notará que PowerBI intentará adivinar y autocompletar mientras escribe, por lo que puede seleccionar lo que necesita de los mensajes sugeridos):
 - a. Column = ventas[cantidad]*ventas[precio_unitario]
5. Después de que la columna se haya llenado, puede cambiar el nombre a **"Ventas Totales"** en el cuadro **Nombre** que se encuentra bajo **Herramientas de columna** (debería estar ya en **Herramientas de columna por defecto** cuando hizo clic en **Nueva columna** en el paso 2).

Ahora debería tener una nueva columna titulada Ventas Totales.



transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	instore_yn	order	line_item_id	product_id	quantity	line_item_amount	unit_price	promo_item_yn	Total Sales
8	Friday, 12 April 2019	09:02:14	0	42		Y		1	30	1	3	IN		3
12	Sunday, 21 April 2019	09:46:39	0	42		Y		1	49	1	3	IN		3
13	Monday, 8 April 2019	08:16:38	0	42		Y		1	45	1	3	IN		3
14	Sunday, 14 April 2019	09:28:10	0	42		Y		1	30	1	3	IN		3
18	Wednesday, 10 April 2019	09:34:30	0	42		Y		1	43	1	3	IN		3
19	Monday, 8 April 2019	08:23:51	0	42		Y		1	47	1	3	IN		3
19	Friday, 26 April 2019	08:14:23	0	42		Y		1	30	1	3	IN		3
21	Friday, 12 April 2019	07:33:45	0	42		Y		1	32	1	3	IN		3
26	Friday, 25 April 2019	06:34:09	0	42		Y		1	32	1	3	IN		3
36	Sunday, 7 April 2019	08:23:25	0	42		Y		1	53	1	3	IN		3
65	Saturday, 27 April 2019	10:27:40	0	42		Y		1	24	1	3	IN		3
81	Wednesday, 10 April 2019	10:45:30	0	42		Y		1	24	1	3	IN		3
85	Thursday, 11 April 2019	09:59:06	0	42		Y		1	45	1	3	IN		3
88	Monday, 8 April 2019	07:00:42	0	42		Y		1	51	1	3	IN		3
88	Thursday, 25 April 2019	10:18:49	0	42		Y		1	47	1	3	IN		3
89	Sunday, 7 April 2019	10:11:28	0	42		Y		1	53	1	3	IN		3
98	Thursday, 11 April 2019	10:20:28	0	42		Y		1	47	1	3	IN		3
104	Monday, 8 April 2019	07:20:58	0	42		Y		1	51	1	3	IN		3
105	Saturday, 27 April 2019	08:51:16	0	42		Y		1	51	1	3	IN		3
116	Tuesday, 23 April 2019	09:07:31	0	42		Y		1	24	1	3	IN		3
122	Sunday, 28 April 2019	10:50:29	0	42		Y		1	47	1	3	IN		3
123	Friday, 12 April 2019	09:59:01	0	42		Y		1	49	1	3	IN		3
131	Wednesday, 10 April 2019	06:45:47	0	42		Y		1	32	1	3	IN		3
135	Monday, 15 April 2019	09:52:56	0	42		Y		1	47	1	3	IN		3
139	Saturday, 20 April 2019	10:00:36	0	42		Y		1	53	1	3	IN		3
141	Wednesday, 24 April 2019	10:49:19	0	42		Y		1	49	1	3	IN		3
144	Monday, 8 April 2019	09:56:19	0	42		Y		1	45	1	3	IN		3
145	Monday, 8 April 2019	09:41:56	0	42		Y		1	87	1	3	IN		3
151	Sunday, 21 April 2019	09:23:25	0	42		Y		1	45	1	3	IN		3
157	Thursday, 11 April 2019	09:27:22	0	42		Y		1	24	1	3	IN		3
160	Sunday, 7 April 2019	09:06:36	0	42		Y		1	30	1	3	IN		3
165	Wednesday, 24 April 2019	10:15:57	0	42		Y		1	45	1	3	IN		3
176	Sunday, 14 April 2019	09:12:33	0	42		Y		1	24	1	3	IN		3
181	Tuesday, 9 April 2019	09:40:46	0	42		Y		1	32	1	3	IN		3
190	Friday, 12 April 2019	08:40:25	0	42		Y		1	51	1	3	IN		3
190	Tuesday, 23 April 2019	11:16:10	0	42		Y		1	32	1	3	IN		3
192	Sunday, 21 April 2019	09:37:14	0	42		Y		1	32	1	3	IN		3
196	Monday, 15 April 2019	10:16:43	0	42		Y		1	42	1	3	IN		3

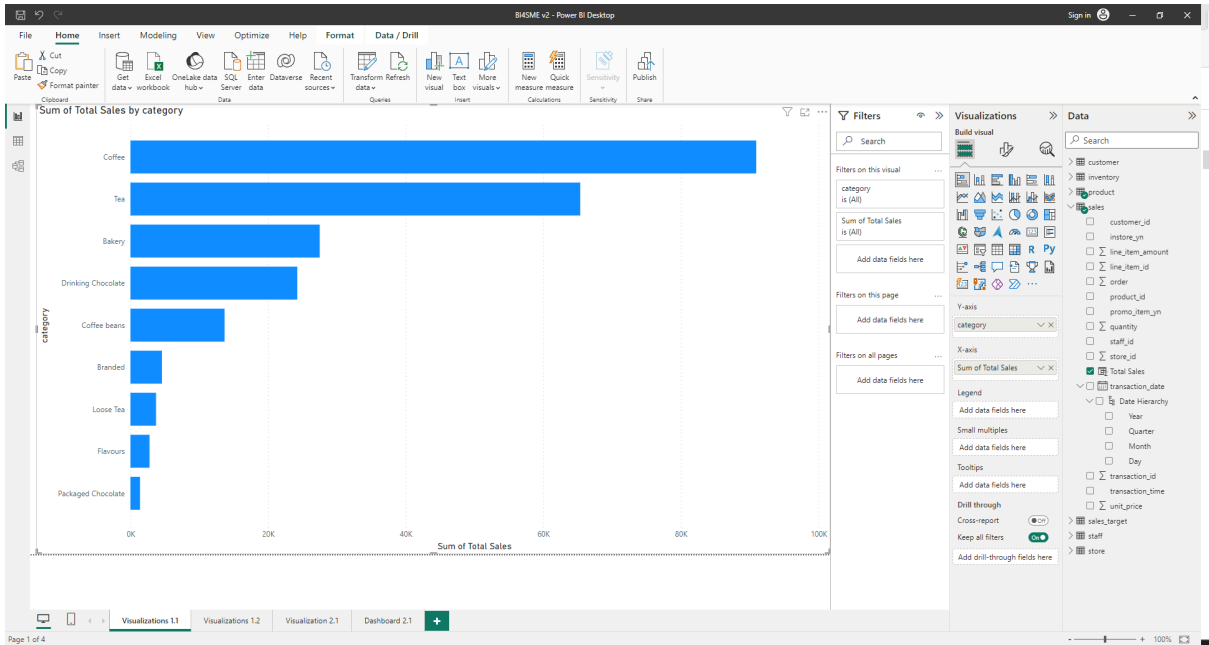
Y cuando regrese a la **Vista de informe** (en el panel lateral a la izquierda de su pantalla) y haga clic en la tabla **Ventas en Datos**, ahora debería ver una nueva opción de **Ventas Totales**.



Ahora, practiquemos la visualización de datos y las funciones relacionadas con la herramienta Power BI.

6.2.5.1. Visualizaciones:

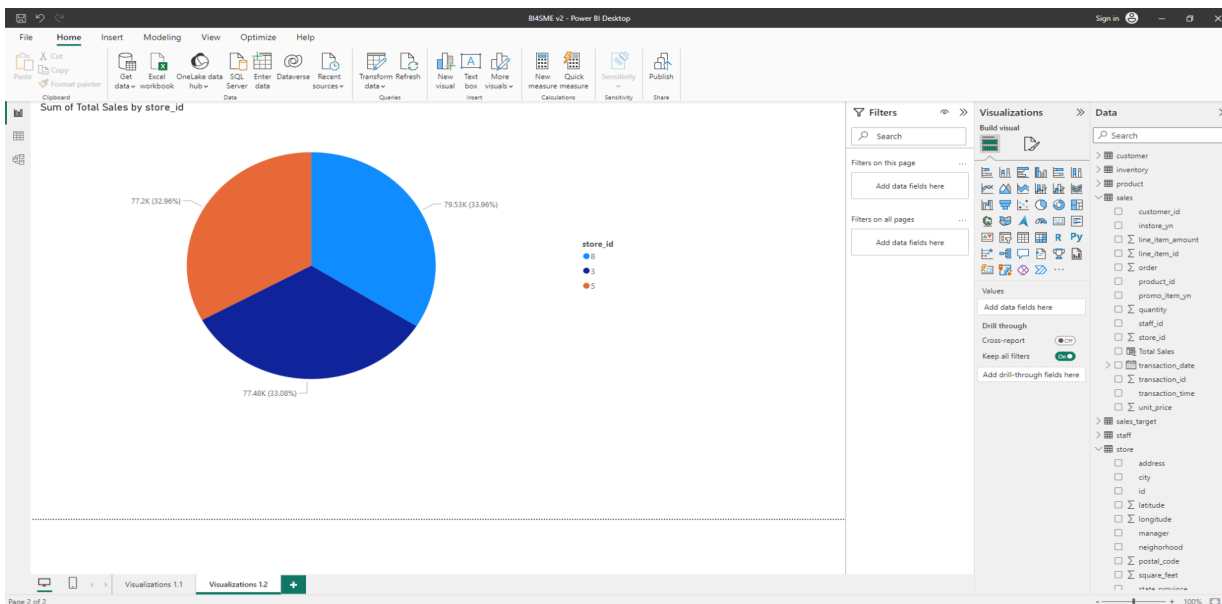
- Crear un gráfico de barras que muestre las ventas por categoría de producto.



Pasos:

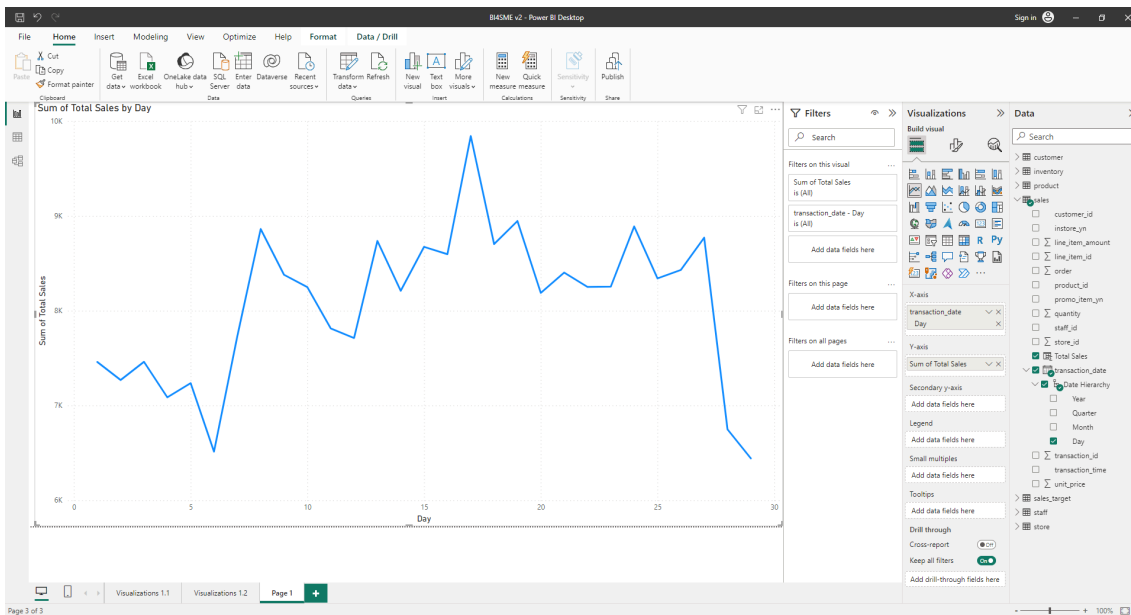
1. Seleccione el gráfico de **barras apiladas** en Visualizaciones.
2. Importe/seleccione **Ventas Totales** bajo la tabla **Ventas** en Datos.
 - a. Asegúrese de que **Suma de Ventas Totales** esté seleccionada en el eje X.
3. Importe/seleccione **Categoría** bajo la tabla **Productos** en Datos.
 - a. Asegúrese de que **Categoría** esté seleccionada en el eje Y.

- Diseñar un gráfico de pastel que muestre la distribución de ventas en diferentes tiendas..



Pasos:

1. Seleccione un **gráfico de pastel** en *Visualizaciones*.
 2. Importe/seleccione **Ventas Totales** bajo la tabla Ventas en Datos.
 - a. Asegúrese de que **Suma de Ventas Totales** esté seleccionada en Valores.
 3. Importe/seleccione **Store_id** bajo la tabla Ventas en Datos.
 - a. Agregue **Store_id** a *Detalles*.
- Utilice un gráfico de líneas para visualizar las tendencias de ventas a lo largo del tiempo.

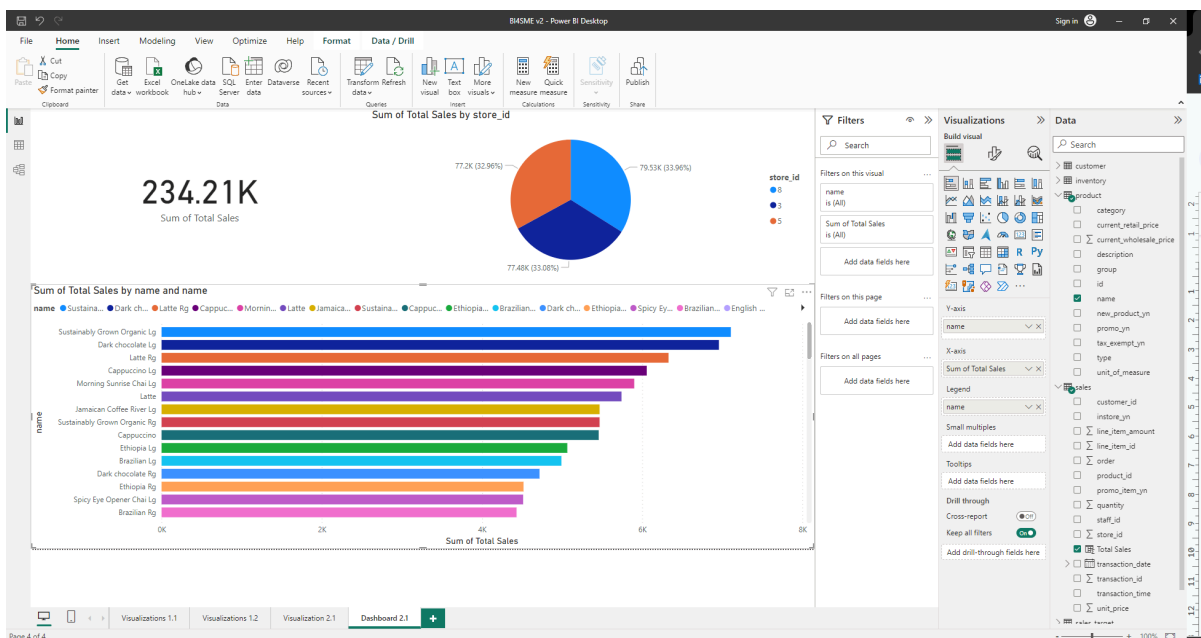


Pasos:

1. Seleccione un **gráfico de líneas** en Visualizaciones.
2. Importe/seleccione **Ventas Totales** bajo la tabla **Ventas** en Datos.
3. En la tabla **Ventas** en Datos, colapse la fecha de transacción, luego colapse la **jerarquía de fechas** y seleccione/importar **día**.
4. Asegúrese de que la fecha de **transacción / día** esté seleccionada en el eje X y que la **suma de Ventas Totales** esté seleccionada en el eje Y.

6.2.5.2. Creación de Informes/Paneles:

- Diseñar un informe de ventas que incluya métricas clave como ventas totales, productos más vendidos por nombre y ventas por ID de tienda..



Para este ejercicio, necesita crear tres visualizaciones diferentes en la misma página del informe. Vamos a desglosarlas.

Pasos para Ventas Totales:

1. Seleccione la **Tarjeta** en Visualizaciones.
2. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

Pasos para Ventas por Store id:

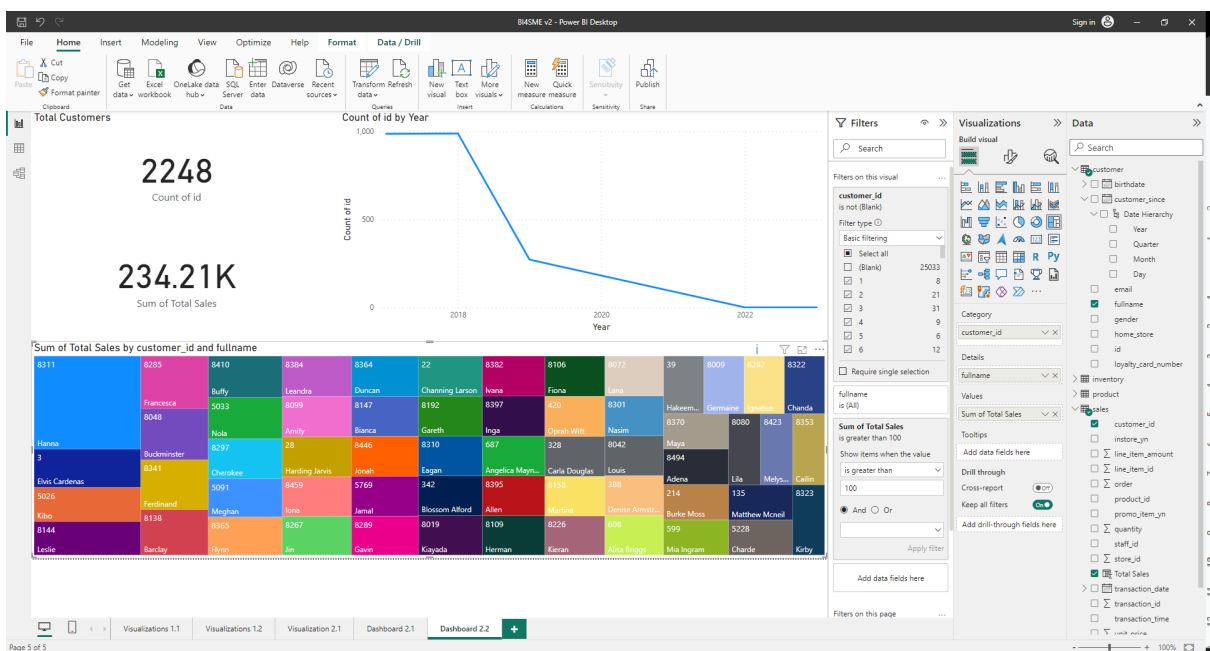
3. Seleccione un **gráfico de pastel** en Visualizaciones.
4. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.
5. Importe/Seleccione **Store_id** bajo la tabla **Ventas** en *Datos*.

Pasos para Productos más vendidos:

6. Seleccione el **gráfico de barras apiladas** en Visualizaciones.
7. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

8. Importe/Selecione **nombre** bajo la tabla Productos en *Datos*. (Consejo: también puede arrastrar y soltar **nombre** en el campo de Leyenda de Visualizaciones para obtener diferentes colores para cada **nombre** de producto).

- **Crear un informe de clientes que muestre métricas como el total de clientes, nuevos clientes adquiridos a lo largo del tiempo (anualmente) y los mejores clientes por ventas..**



Para este ejercicio, necesita crear tres visualizaciones diferentes en la misma página del informe. Vamos a desglosarlas.

Pasos para Total de Clientes:

1. Seleccione la **Tarjeta** en *Visualizaciones*.
2. Importe/Selecione **id** bajo la tabla Clientes en *Datos*.
3. En *Visualizaciones*, mientras la **Tarjeta** está seleccionada, tiene una opción de "Campos" en la parte inferior de esa columna. Asegúrese de que "Campos" esté configurado con "Cuenta de id". Esto le mostrará cuántos ids hay y, por lo tanto, el total de clientes.

Pasos para Nuevos Clientes Adquiridos a lo Largo del Tiempo (Anualmente):

1. Seleccione un **gráfico de líneas** en *Visualizaciones*.
2. Importe/Seleccione **id** bajo la tabla **Clientes** en Datos y arrástrelo al eje Y. Asegúrese de seleccionar **Cuenta de Id** como hizo en la última tabla.
3. En la tabla **Clientes** en Datos, colapse **customer_since**, luego colapse **Jerarquía de fechas** y seleccione/importar **Año**. Asegúrese de que este campo aparezca en el eje X.

Pasos para Cliente Principal por Ventas:

1. Seleccione un **mapa de árbol** en *Visualizaciones*.
2. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.
3. Importe/Seleccione **customer_id** bajo la tabla Ventas en *Datos*.
4. Arrastre y suelte **fullname**, encontrado bajo la tabla **Clientes** en el campo Detalles de *Visualizaciones* para obtener el nombre de cada cliente en cada cuadro.

Tendrá demasiados resultados, lo que significa que no podrá ver ninguna información y necesitarán ser reducidos. Haremos esto aplicando algunos filtros a esta visualización.

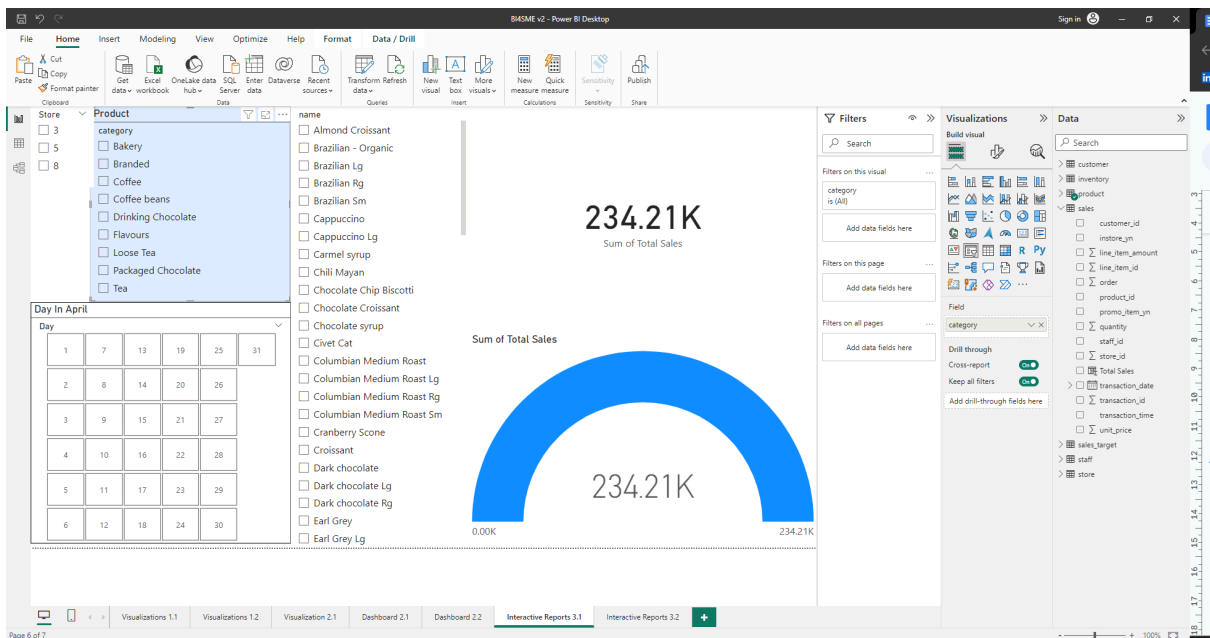
5. Seleccione el **mapa de árbol** que ahora está en su página de informe y vaya a la "Columna de Filtros".
6. Para "customer_id", establezca el **Tipo de Filtro** en *Filtro Básico*, luego seleccione todas las respuestas y deseccione "En blanco". Esto dejará solo a los clientes que tienen ID.
7. A continuación, en "**Suma de Ventas Totales**", vaya al campo **Mostrar elementos cuando el valor**, seleccione es Mayor que y escriba 100 en el campo de abajo.

Como una bonificación adicional, también puede agregar una Visualización de Tarjeta, y de esta manera, cuando seleccione un cliente, también verá la cantidad exacta que ha gastado. Para hacer esto:

8. Seleccione la **Tarjeta** en *Visualizaciones*.
9. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

6.2.5.3. Informes Interactivos:

- **Diseñar un informe donde los usuarios puedan filtrar los datos de ventas por fecha, producto o tienda.**



For this exercise, you need to create different interactable visualisations on the same report page. Let's break them down.

Pasos para seleccionar diferentes tiendas:

1. Seleccione el objeto visual de **Segmentación** de datos en *Visualizaciones*.
2. Importe/Seleccione store_id bajo la tabla **Ventas** en *Datos*.
 - a. Bajo *Visualizaciones*, si selecciona "Formatear su visual", verá una opción para **Configuración de segmentación** de datos, y

allí puede seleccionar el estilo de segmentación que desea. Si no está seleccionado ya, configure el estilo en Lista vertical.

Pasos para seleccionar diferentes fechas:

1. Seleccione el objeto visual de **Segmentación de datos** en *Visualizaciones*.
2. En la tabla **Ventas** en *Datos*, colapse **fecha de transacción**, luego colapse **Jerarquía de fechas**, y seleccione/importar **día**.
 - a. Bajo *Visualizaciones*, si selecciona "Formatear su visual", verá una opción para **Configuración de segmentación** de datos, y allí puede seleccionar el estilo de segmentación que desea. Configure el estilo en Cuadrícula.

Pasos para seleccionar diferentes productos:

1. Seleccione el objeto visual de **Segmentación de datos** en *Visualizaciones*.
2. Importe/Seleccione **Categoría** bajo la tabla **Productos** en *Datos*, y luego haga lo mismo nuevamente pero esta vez importe y seleccione nombre bajo la tabla **Productos** en *Datos*. (Es importante que haga esto en el orden correcto, o reorganice sus elementos en el área de "Campos" en sus visualizaciones para **tener Producto** por encima de **nombre**, de lo contrario no se mostrará correctamente).
 - a. Bajo *Visualizaciones*, si selecciona "Formatear su visual", verá una opción para **Configuración de segmentación** de datos, y allí puede seleccionar el estilo de segmentación que desea. Si no está seleccionado ya, configure el estilo en Lista vertical.

Ahora puede seleccionar una categoría de producto, o colapsar el producto y seleccionar productos individuales por nombre. Ahora lo que falta son las visualizaciones para poder ver las ventas de cada selección que haga.

Pasos para agregar visualizaciones para Ventas Totales:

1. Seleccione la **Tarjeta** en *Visualizaciones*.
2. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

Esto le proporciona un número grande y fácil de leer.

Pasos para agregar una segunda visualización para Ventas Totales:

Esta es puramente para visuales, pero dado que una gran parte de Power BI son las visualizaciones, es bueno practicar las diversas opciones para mostrar datos.

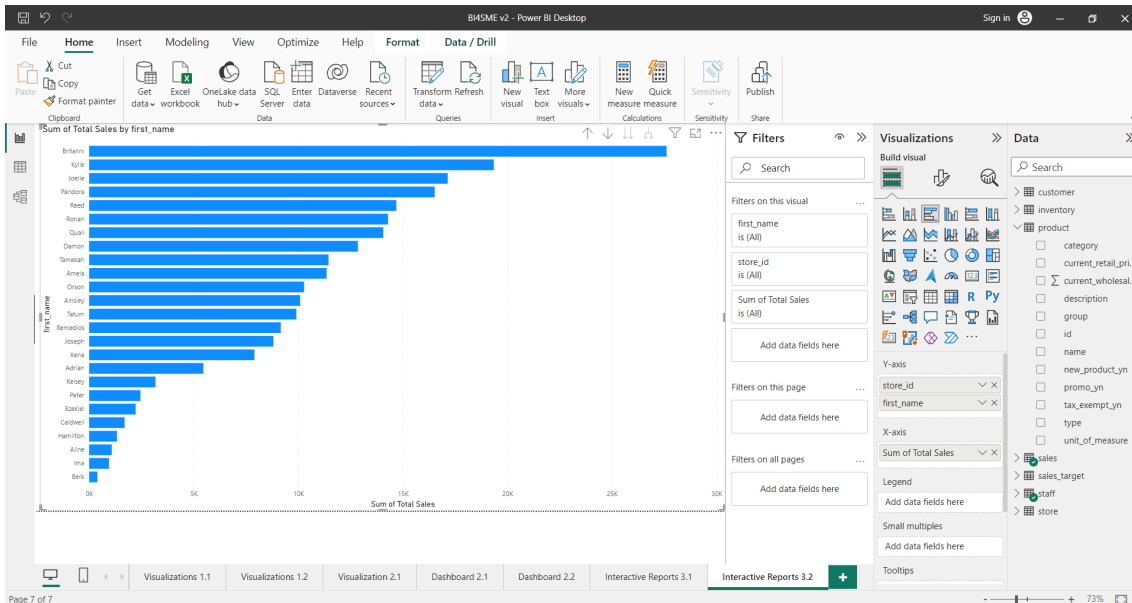
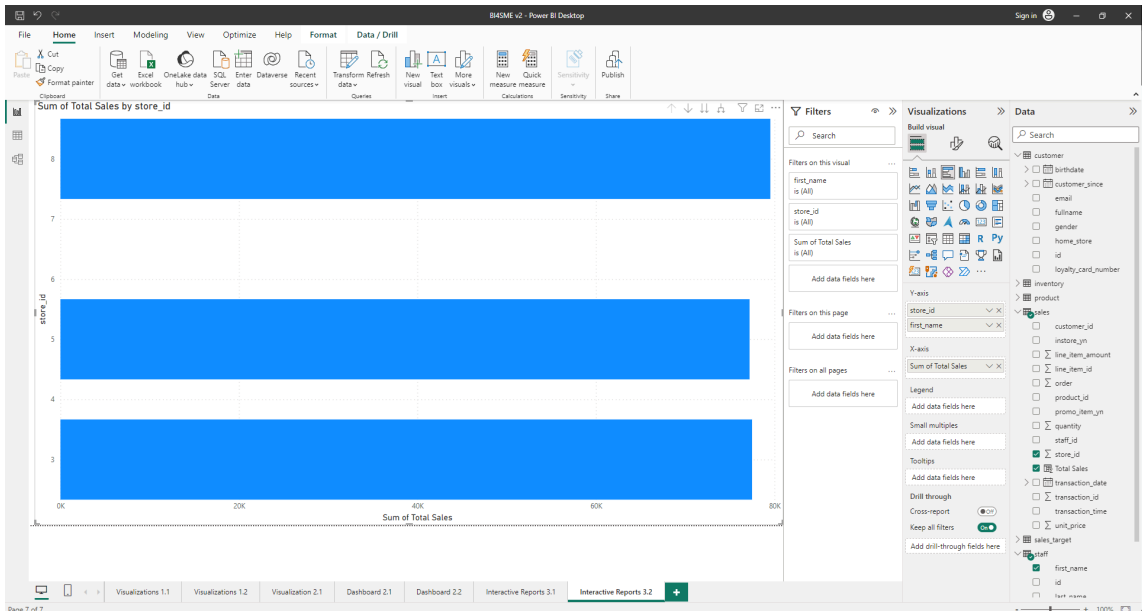
1. Seleccione el **Medidor** en *Visualizaciones*.
2. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

Hay algunos pasos adicionales necesarios para que la visualización del **Medidor** tenga sentido, ya que actualmente no sabe cuál es el valor máximo y solo le da un número aleatorio.

3. Bajo *Visualizaciones*, si selecciona "Formatear su visual", verá una opción para **Eje del Medidor (Gauge Axis)**, y allí puede establecer el valor máximo al número máximo de ventas en el informe ingresando manualmente el número total de ventas.

Si todo se hizo correctamente, ahora las dos visualizaciones deberían cambiar a medida que seleccione diferentes opciones en los tres segmentadores creados anteriormente.

- Crear un informe que permita a los usuarios profundizar desde las ventas totales por tienda hasta las ventas individuales por nombre de cada miembro del personal.



Pasos para crear una profundización desde las ventas totales por tienda hasta las ventas individuales por nombre de cada miembro del personal:

1. Seleccione el **gráfico de barras** apiladas en *Visualizaciones*.
2. Importe/Seleccione **Ventas Totales** bajo la tabla **Ventas** en *Datos*.

3. Importe/Selecione **Store_id** bajo la tabla **Ventas** en *Datos*.
4. Para este paso, no puede simplemente seleccionar la siguiente opción. Necesita arrastrar el campo **first_name** de la tabla de **personal** en *Datos* y soltarlo en el eje Y bajo *Visualizaciones*, debajo del **Store_id** existente.

Si se hace correctamente, al principio no cambiará nada en la visualización, pero aparecerá un nuevo conjunto de opciones en la esquina superior derecha de la visualización en la página del informe.

- El primer icono, cuando se selecciona, permite al usuario "profundizar" y ver las ventas de cada miembro del personal en cada tienda cuando seleccionan la columna de ventas de una tienda específica.
- El segundo icono permite al usuario ir al segundo nivel de visualización y ver una comparación de las ventas de todos los miembros del personal en todas las tiendas.
- El tercer icono llevará al usuario a una tercera visualización donde se muestran juntos tanto el ID de la tienda como el nombre del miembro del personal comparando las ventas totales.

6.2.6. Paso 6 – Uso de Python para BI

En esta parte, practicarás tus conocimientos de Python para fines de BI. Le recomendamos que use las herramientas Anaconda y Spider

6.2.6.1. Creación de Base de Datos

- 1 - Base de Datos de Datos de Ventas::

Cree un código en Python que genere un pequeño conjunto de datos (5 filas) con información básica de ventas utilizando listas y diccionarios.

- Use los datos generados para crear un DataFrame de Pandas.
- Muestre el contenido del DataFrame.

Resultado:

	Customer_ID	Product_ID	Purchase_Date	Quantity_Sold	Total_Sales_Amount
0	1	101	2023-01-01	5	50.0
1	2	102	2023-01-02	3	30.0
2	3	103	2023-01-03	4	40.0
3	4	104	2023-01-04	2	20.0
4	5	105	2023-01-05	6	60.0

Solution:

```

1. import pandas as pd
2.
3. # Create a small sales dataset
4. sales_data = {
5.     'Customer_ID': [1, 2, 3, 4, 5],
6.     'Product_ID': [101, 102, 103, 104, 105],
7.     'Purchase_Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
8.     '2023-01-05'],
9.     'Quantity_Sold': [5, 3, 4, 2, 6],
10.    'Total_Sales_Amount': [50.0, 30.0, 40.0, 20.0, 60.0]
11. }
12. # Create a Pandas DataFrame
13. sales_df = pd.DataFrame(sales_data)
14.
15. # Display the DataFrame
16. print(sales_df)
    
```

- 2 - Base de Datos de Empleados:

Cree un código en Python que genere un pequeño conjunto de datos de empleados (5 empleados) utilizando listas y diccionarios. Incluya atributos como ID de empleado, nombre, fecha de contratación, departamento y salario.

- Use los datos generados para crear un DataFrame de Pandas.
- Muestre el contenido del DataFrame.

Resultado:

	Employee_ID	Name	Hire_Date	Department	Salary
0	1	John	2023-01-01	HR	60000
1	2	Jane	2023-01-02	Finance	55000
2	3	Bob	2023-01-03	IT	70000
3	4	Alice	2023-01-04	Marketing	65000
4	5	David	2023-01-05	IT	75000

Solucion:

```
1. import pandas as pd
2.
3. # Create a small employee dataset
4. employee_data = {
5.     'Employee_ID': [1, 2, 3, 4, 5],
6.     'Name': ['John', 'Jane', 'Bob', 'Alice', 'David'],
7.     'Hire_Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
8.     '2023-01-05'],
9.     'Department': ['HR', 'Finance', 'IT', 'Marketing', 'IT'],
10.    'Salary': [60000, 55000, 70000, 65000, 75000]
11. }
12. # Create a Pandas DataFrame
13. employee_df = pd.DataFrame(employee_data)
14.
15. # Display the DataFrame
16. print(employee_df)
```

-
-
-

- 3 - Inventory Database:

Genere un conjunto de datos básico de inventario con algunos productos (3 productos). Incluya ID de producto, nombre, categoría, cantidad en stock y precio unitario.

- Cree un DataFrame de Pandas utilizando los datos generados.
- Muestre el contenido del DataFrame.

Resultado:

	Product_ID	Product_Name	Category	Quantity_In_Stock	Unit_Price
0	101	Laptop	Electronics	10	800.0
1	102	Phone	Electronics	20	400.0
2	103	Tablet	Electronics	15	300.0

Solucion:

```

1. import pandas as pd
2.
3. # Create a small inventory dataset
4. inventory_data = {
5.     'Product_ID': [101, 102, 103],
6.     'Product_Name': ['Laptop', 'Phone', 'Tablet'],
7.     'Category': ['Electronics', 'Electronics', 'Electronics'],
8.     'Quantity_In_Stock': [10, 20, 15],
9.     'Unit_Price': [800.0, 400.0, 300.0]
10. }
11.
12. # Create a Pandas DataFrame
13. inventory_df = pd.DataFrame(inventory_data)
14.
15. # Display the DataFrame
16. print(inventory_df)

```

6.2.6.2. Filtrado y Selección de Datos

- 1 - Filtrar Data

Dado un DataFrame con información de productos, filtre y seleccione las filas donde la cantidad en stock sea menor a 10 unidades. Muestre los datos filtrados.

```

1.     'Product_ID': [101, 102, 103, 104, 105],
2.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
3.     'Quantity_In_Stock': [8, 15, 3, 12, 5],
4.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]

```



Resultado:

	Product_ID	Product_Name	Quantity_In_Stock	Unit_Price
0	101	Laptop	8	800.0
2	103	Tablet	3	300.0
4	105	Monitor	5	200.0

Solucion:

```

1. import pandas as pd
2.
3. # Sample DataFrame
4. data = {
5.     'Product_ID': [101, 102, 103, 104, 105],
6.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
7.     'Quantity_In_Stock': [8, 15, 3, 12, 5],
8.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
9. }
10.
11. df = pd.DataFrame(data)
12.
13. # Filter and select rows where quantity in stock is less than 10
14. filtered_data = df[df['Quantity_In_Stock'] < 10]
15.
16. # Display the filtered data
17. print(filtered_data)
18.
    
```

- 2 - Selección de Columnas Específicas

Dado un DataFrame con información de empleados, seleccione y muestre solo las columnas "Nombre" y "Departamento" para todos los empleados.

```

1.     'Employee_ID': [1, 2, 3, 4, 5],
2.     'Name': ['John', 'Jane', 'Bob', 'Alice', 'David'],
3.     'Hire_Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
4.     '2023-01-05'],
5.     'Department': ['HR', 'Finance', 'IT', 'Marketing', 'IT'],
6.     'Salary': [60000, 55000, 70000, 65000, 75000]
    
```

Resultado:

	Name	Department
0	John	HR
1	Jane	Finance
2	Bob	IT
3	Alice	Marketing
4	David	IT

Solucion:

```

1. import pandas as pd
2.
3. # Sample DataFrame
4. data = {
5.     'Employee_ID': [1, 2, 3, 4, 5],
6.     'Name': ['John', 'Jane', 'Bob', 'Alice', 'David'],
7.     'Hire_Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
8.     '2023-01-05'],
9.     'Department': ['HR', 'Finance', 'IT', 'Marketing', 'IT'],
10.    'Salary': [60000, 55000, 70000, 65000, 75000]
11. }
12. df = pd.DataFrame(data)
13.
14. # Select and display only the "Name" and "Department" columns
15. selected_columns = df[['Name', 'Department']]
16. print(selected_columns)
17.

```

- 3 - Combinar Filtrado y Selección de Columnas

De un DataFrame que contiene datos de productos, filtre y seleccione las filas donde el precio unitario sea mayor a \$100 y muestre solo las columnas "NombreProducto" y "PrecioUnitario" para esos productos.

```

1.     'Product_ID': [101, 102, 103, 104, 105],
2.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
3.     'Quantity_In_Stock': [8, 15, 3, 12, 5],
4.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]

```

Resultado:

	Product_Name	Unit_Price
0	Laptop	800.0
1	Phone	400.0
2	Tablet	300.0
3	Printer	150.0
4	Monitor	200.0

Solucion:

```

1. import pandas as pd
2.
3. # Sample DataFrame
4. data = {
5.     'Product_ID': [101, 102, 103, 104, 105],
6.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
7.     'Quantity_In_Stock': [8, 15, 3, 12, 5],
8.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]

```

```

9. }
10.
11. df = pd.DataFrame(data)
12.
13. # Filter and select rows where unit price is greater than $100
14. filtered_data = df[df['Unit_Price'] > 100]
15. selected_columns = filtered_data[['Product_Name', 'Unit_Price']]
16.
17. # Display the filtered and selected data
18. print(selected_columns)
19.

```

6.2.6.3. Análisis y Reporte de Datos

- 1 - Calcular Ventas Totales

Dado un DataFrame de Pandas que contiene datos de ventas, calcule las ventas totales para todas las transacciones y muestre el resultado.

```

1. 'Transaction_ID': [1, 2, 3, 4, 5],
2. 'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
3. 'Quantity_Sold': [8, 15, 3, 12, 5],
4. 'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]

```

Resultado:

Total Sales: \$16100.00

Solucion:

```

1. import pandas as pd
2. # Sample DataFrame
3. data = {
4.     'Transaction_ID': [1, 2, 3, 4, 5],
5.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
6.     'Quantity_Sold': [8, 15, 3, 12, 5],
7.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
8. }
9.
10. df = pd.DataFrame(data)
11.
12. # Calculate total sales
13. df['Total_Sales'] = df['Quantity_Sold'] * df['Unit_Price']
14. total_sales = df['Total_Sales'].sum()
15.
16. # Display the total sales
17. print(f'Total Sales: ${total_sales:.2f}')

```

- 2 - Average Order Value

De un DataFrame de Pandas que contiene datos de órdenes, calcule el valor promedio de orden (AOV) para todas las órdenes y muestre el resultado.

```
1. 'Order_ID': [101, 102, 103, 104, 105],
2. 'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
3. 'Quantity_Sold': [8, 15, 3, 12, 5],
4. 'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
```

Resultado:

Average Order Value (AOV): \$3220.00

Solucion:

```
1. import pandas as pd
2. # Sample DataFrame
3. data = {
4.     'Order_ID': [101, 102, 103, 104, 105],
5.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
6.     'Quantity_Sold': [8, 15, 3, 12, 5],
7.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
8. }
9.
10. df = pd.DataFrame(data)
11.
12. # Calculate total sales for each order
13. df['Total_Sales'] = df['Quantity_Sold'] * df['Unit_Price']
14.
15. # Calculate average order value (AOV)
16. aov = df['Total_Sales'].mean()
17.
18. # Display the average order value
19. print(f'Average Order Value (AOV): ${aov:.2f}')
```

- 3 - Encontrar Valores Mínimos y Máximos

De un DataFrame de Pandas que contiene datos de productos, encuentre el producto con el precio unitario más alto y más bajo, y muestre sus detalles.

```
1. 'Product_ID': [101, 102, 103, 104, 105],
2. 'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
3. 'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
```

Resultado:

Product with the Highest Unit Price:

	Product_ID	Product_Name	Unit_Price
0	101	Laptop	800.0

Product with the Lowest Unit Price:

	Product_ID	Product_Name	Unit_Price
--	------------	--------------	------------

3	104	Printer	150.0
---	-----	---------	-------

Solucion:

```

1. import pandas as pd
2. # Sample DataFrame
3. data = {
4.     'Product_ID': [101, 102, 103, 104, 105],
5.     'Product_Name': ['Laptop', 'Phone', 'Tablet', 'Printer', 'Monitor'],
6.     'Unit_Price': [800.0, 400.0, 300.0, 150.0, 200.0]
7. }
8.
9. df = pd.DataFrame(data)
10.
11. # Find the product with the highest unit price
12. max_price_product = df[df['Unit_Price'] == df['Unit_Price'].max()]
13. # Find the product with the lowest unit price
14. min_price_product = df[df['Unit_Price'] == df['Unit_Price'].min()]
15. # Display the product details
16. print(f'Product with the Highest Unit Price:')
17. print(max_price_product)
18. print(f'\nProduct with the Lowest Unit Price:')
19. print(min_price_product)
    
```