|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Business Intelligence  Practical #2 | | |
|  |  |  |  |
| **Name** | **SWEETY DANIEL** | **Roll Number** | **21302D0012** |
| **Class** | TY BSCIT | **Division** | B |
| **Subject/Course:** | Business intelligence | | |
| **Topic** | Perform Data Wrangling (ETL) | | |
|  |  |  |  |
| **What is ETL?** | | | |
| ETL, which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.  ETL provides the foundation for data analytics and machine learning workstreams. Through a series of business rules, ETL cleanses and organizes data in a way which addresses specific business intelligence needs, like monthly reporting, but it can also tackle more advanced analytics, which can improve back-end processes or end user experiences. ETL is often used by an organization to:   * Extract data from legacy systems * Cleanse the data to improve data quality and establish consistency * Load data into a target database | | | |
|  | | | |
| **What is Power Query?** | | | |
| Power Query is a data transformation and data preparation engine. Power Query comes with a graphical interface for getting data from sources and a Power Query Editor for applying transformations. Because the engine is available in many products and services, the destination where the data will be stored depends on where Power Query was used. Using Power Query, you can perform the extract, transform, and load (ETL) processing of data.  Power Query input, transformation, and destination | | | |
|  | | | |
| **What are the components of Power Query Editor?** | | | |
| The components of Power Query Editor include   1. **The ribbon,** which consists of four tabs: Home, Transform, Add Column, View, Tools, and Help. 2. **The Queries pane**, which provides a view of all your available queries. 3. **The Data view**, which is your main working view that displays a preview of the data for your query. 4. **The Query Settings pane**, which provides options for your query and access to different ribbon buttons to complete various tasks. | | | |
| **Write the steps to perform ETL in Power BI?** | | | |
| **Import Excel product data** 1.Select the arrow next to **Get data** in the Power BI Desktop ribbon's **Home** tab, and then select **Excel** from the **Common data sources** menu.  Screenshot that shows the Get data menu. 2.In the **Open** dialog box, navigate to and select the **Products.xlsx** file, and then select **Open**.  3.In the **Navigator**, select the **Products** table and then select **Transform Data**. Screenshot that shows the Navigator screen with the Products table highlighted. Screenshot that shows the Power Query Editor. **Import the OData feed's order data**  1. In Power Query Editor, select **New Source** and then, from the **Most Common** menu, select **OData feed**.   Screenshot that highlights the OData Feed option.   1. In the **OData feed** dialog box, paste the Northwind OData feed URL, <https://services.odata.org/V3/Northwind/Northwind.svc/>. Select **OK**.   Screenshot that highlights the URL field in the OData feed dialog box.   1. In **Navigator**, select the **Orders** table, and then select **OK** to load the data into Power Query Editor.   Screenshot that highlights the Orders table in the OData navigator.  **Expand the order data** 1. Scroll to the right in the **Orders** table until you see the **Order\_Details** column. It contains references to another table and not data.  Screenshot that highlights the Order_Details column.  2.Select the **Expand** icon ( ) in the **Order\_Details** column header.  3.In the dropdown menu:  a. Select **(Select All Columns)** to clear all columns.  b. Select **ProductID**, **UnitPrice**, and **Quantity**, and then select **OK**.  Screenshot that highlights the ProductID, UnitPrice, and Quantity columns.  After you expand the **Order\_Details** table, three new nested table columns replace the **Order\_Details** column. There are new rows in the table for each order's added data.  Screenshot that highlights the expanded columns. **Create a custom calculated column**  1. In the Power Query Editor's **Add Column** ribbon tab, select **Custom Column**.   Screenshot that highlights the Custom Column button.   1. In the **Custom Column** dialog box, type **LineTotal** in the **New column name** field. 2. In the **Custom column formula** field after the **=**, enter **[Order\_Details.UnitPrice]** \* **[Order\_Details.Quantity]**. You can also select the field names from the **Available columns** scroll box and select **<< Insert**, instead of typing them. 3. Select **OK**.   Screenshot that highlights the New column name and Custom column formula fields.  The new **LineTotal** field appears as the last column in the **Orders** table. **Set the new field's data type**  **Clean up the orders columns**  * OrderDate * ShipCity * ShipCountry * Order\_Details.ProductID * Order\_Details.UnitPrice * Order\_Details.Quantity * LineTotal   You can rename the columns prefixed with "**Order\_Details.**" to make them easier to read:   1. Double-click or tap and hold each column header, or right-click the column header, and select **Rename** from the dropdown menu. 2. Delete the **Order\_Details.** prefix from each name.   Finally, to make the **LineTotal** column easier to access, drag and drop it to the left, just to the right of the **ShipCountry** column.  Screenshot that shows the cleaned up columns in the table. | | | |