**Overview of Home Menu**

**This documentation covers all tabs and their practical usage, including examples, from the Power Query Home tab.**

**Dataset Description:**

This uncleaned employee dataset contains 30 rows and 12 column includes various inconsistencies such as missing values, duplicates, inconsistent formats, and other issues. It is designed to help demonstrate and practice cleaning operations using Power Query.

**Columns and their descriptions:**

1. **EmployeeID**: A unique identifier for each employee. (Inconsistent formatting and missing values)
2. **Name**: The employee's full name. (Contains duplicates and missing entries)
3. **Department**: The department to which the employee is assigned. (Contains null values)
4. **Age**: The age of the employee. (Some ages are missing and inconsistencies in format exist)
5. **Salary**: The employee's salary in numbers. (Missing values and inconsistent formats)
6. **DateOfJoining**: The date the employee joined the company. (Various date formats)
7. **Email**: The employee’s email address. (Inconsistent format and missing values)
8. **PhoneNumber**: The employee's contact number. (Contains inconsistent formats, missing country codes)
9. **Address**: The residential address of the employee. (Some entries are incomplete)
10. **Gender**: Gender of the employee. (Includes incorrect or inconsistent entries)
11. **Manager**: The employee’s manager’s name. (Duplicate and missing entries)
12. **JobLevel**: Job level of the employee. (Contains inconsistencies in format)

**Tabs and operations Available inside Home menu of Power Query UI  
are,**

**1.New Source:**This operation allows you to connect to a new data source from which you can import data into Power Query.

**Options Available:**

a) **File**: Import data from Excel, CSV, XML, or other local files.  
b) **Database**: Connect to a database such as SQL Server, Access, or Oracle.  
c) **Azure**: Import data from Azure services like Azure SQL Database or Azure Blob Storage.  
d) **Online Services**: Connect to data from online services such as SharePoint, Dynamics, or Salesforce.  
e) **Other Sources**: Import data from other sources such as web URLs or OData feeds.

**Steps**:  
Home >> New Source >> Select the desired source (e.g., Excel).

**Example**:  
Connect to an Excel file to import employee data.

**Steps**:

1. Home >> New Source >> File >> Excel.
2. Browse to the Excel file.
3. Select the sheet containing employee data.
4. Click **OK** to load the data into Power Query.

**2. Recent Sources:**

The Recent Sources option lists the last few data sources you connected to, allowing you to quickly reconnect without manually browsing.

**Options Available:**

No additional sub-options. It simply lists your most recent connections.

**Steps**:  
Home >> Recent Sources >> Choose the desired recent source.

**Example**:  
Reconnect to the most recent Excel file used for data analysis.

**Steps**:

1. Home >> Recent Sources >> Select the most recent Excel file.
2. **Enter Data:**The Enter Data operation allows you to manually enter data into Power Query.

**Options Available:**

No additional sub-options. It opens a grid where you can manually input data.

**Steps**:  
Home >> Enter Data >> Enter the required data in the grid.

**Example**:  
Manually input employee names and salaries.

1. **Steps**: Home >> Enter Data.
2. Copy paste above data inside grid
3. Click **OK** to load this manually entered data into Power Query.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EmployeeID** | **Name** | **Department** | **Age** | **Salary** | **DateOfJoining** | **Email** | **PhoneNumber** | **Address** | **Gender** | **Manager** | **JobLevel** |
| E101 | Jane Doe | Sales | 29 | 45000 | 10-04-2018 | jane.doe@email.com | 123-456-7890 | 123 Main St | Female | Michael Scott | 2 |
| E102 | John Smith | IT | 35 | 75000 | 15-11-2016 | john.smith@email.com | 321-654-0987 | 456 Oak St | Male | Dwight Schrute | 3 |
| E103 | Emily White | HR | 41 | 58000 | 01-07-2012 | emily.white@email.com | 111-222-3333 | 789 Pine St | Female | Jim Halpert | 4 |
| E104 | Kevin Brown | Marketing | 30 | 61000 | 25-01-2019 | kevin.brown@email.com | 444-555-6666 | 987 Cedar Ave | Male | Angela Martin | 3 |
| E105 | Sarah Green | Finance | 38 | 72000 | 12-09-2015 | sarah.green@email.com | 555-777-8888 | 654 Willow Rd | Female | Toby Flenderson | 4 |

1. **Data Source Settings:**

This operation manages your data source credentials, permissions, and connections.

**Options Available:**

No additional sub-options.

**Steps**:  
Home >> Data Source Settings >> Choose the data source.

**Example**:  
Change the credentials for the Excel file data source.

**Steps**:

1. Home >> Data Source Settings.
2. Select the Excel file data source.
3. Click **Edit** to update the credentials or connection settings.

**5.Manage Parameters:**  
Create and manage parameters in Power Query to use dynamic inputs or control certain transformations.

**Options Available:**

a) **New Parameter**: Create a new parameter.  
b) **Manage Parameters**: Modify or delete existing parameters.

**Steps**:  
Home >> Manage Parameters >> New Parameter

**Example**:  
Create a parameter to dynamically filter data based on department names.

**Steps**:

1. Home >> Manage Parameters >> New Parameter.
2. Set a name (e.g., "Department").
3. Choose **Text** as the data type.
4. Set default and current values (e.g., "Sales").
5. Use this parameter in your query to filter data.
6. **Refresh Preview:**

This operation refreshes the data preview to reflect any recent changes in the source or transformations.

**Options Available:** No additional sub-options.

**Steps**:  
Home >> Refresh Preview

**Example**:  
Refresh the preview after making changes to the data source.

**Steps**:

1. Home >> Refresh Preview.
2. Wait for the preview to update.
3. **Properties:**

Rename and describe your query for easier identification.

**Options Available:**

No additional sub-options.

**Steps**:  
Home >> Properties >> Modify the name and description.

**Example**:  
Rename the query to "EmployeeData" for clarity.

**Steps**:

1. Home >> Properties.
2. Enter "EmployeeData" as the new query name.
3. Optionally, add a description for future reference.
4. **Advanced Editor:**

The Advanced Editor allows you to write or modify M-code directly for advanced custom transformations.

**Options Available:** No additional sub-options.

**Steps**:  
Home >> Advanced Editor >> Modify M code as needed.

**Example**:  
Edit the query to add a custom column calculation.

**Steps**:

1. Home >> Advanced Editor.
2. Modify the M code to include a new custom column.
3. Click **Done** to apply the changes.
4. **Manage:**

The Manage allows you to choose what you want to do with current dataset/query?

**Options Available:**

1. Delete: To delete existing dataset
2. Duplicate: To duplicate existing dataset
3. Reference : To make reference to other datasets/query

**Steps**:  
Home >> Manage >> Choose desired option

**Example**:

To duplicate Emp Data 2

**Steps**:

1. Select “Emp Data 2” query
2. Home >> Manage.
3. Click **duplicate**
4. **Choose Columns**

**Purpose**:  
This feature allows you to select specific columns to include in your data. Unselected columns will be excluded from the dataset.

**Options Available:**

a) **Choose Columns**: Opens a window to manually select which columns to keep.  
b) **Go To Column**: Quickly navigate to a specific column.

**Steps**:  
Home >> Choose Columns >> Select desired columns.

**Example**:  
Keep only the **Name**, **Department**, and **Salary** columns in the dataset.

**Steps**:

1. Home >> Choose Columns.
2. Check the boxes next to the **Name**, **Department**, and **Salary** columns.
3. Click **OK** to remove all other columns.
4. **Remove Columns**

**Purpose**:  
The Remove Columns option allows you to remove specific columns from your dataset.

**Options Available:**

a) **Remove Columns**: Removes the selected columns from the dataset.  
b) **Remove Other Columns**: Removes all columns except the selected ones.

**Steps**:  
Home >> Remove Columns >> Choose the desired option (e.g., Remove Clumns).

**Example**:  
Remove the **Joining Date** column.

**Steps**:

1. Home >> Remove Columns.
2. Select the **Joining Date** column and click **OK**.
3. **Keep Rows**

**Purpose**:  
The **Keep Rows** operation allows you to retain specific rows based on certain criteria like keeping the top or bottom rows, or keeping rows that fit a pattern, while removing the rest.

**Options Available:**

a) **Keep Top Rows**: Keeps the specified number of rows from the top of the dataset.  
b) **Keep Bottom Rows**: Keeps the specified number of rows from the bottom of the dataset.  
c) **Keep Range of Rows**: Keeps rows from a specific range based on their position (e.g., rows 3 to 8).  
d) **Keep Duplicates**: Keeps only the duplicate rows based on selected columns.

**Example**:  
You can use Keep Top Rows to retain the first 10 entries in your dataset.

**Steps**:

1. Home >> Keep Rows >> Keep Top Rows.
2. Specify the number of rows (e.g., 10) in the prompt.
3. Click **OK**.
4. **Remove Rows**

**Purpose**:  
Remove rows from the dataset based on specific criteria like removing duplicates, alternate rows, or top rows.

**Options Available:**

* 1. **Remove Top Rows**: Removes the specified number of rows from the top of the dataset.
  2. **Remove Bottom Rows**: Removes the specified number of rows from the bottom of the dataset.
  3. **Remove Alternate Rows**: Removes rows in a specified pattern (e.g., every other row).
  4. **Remove Duplicates**: Removes all duplicate rows based on selected columns.

**Steps**:  
Home >> Remove Rows >> Choose the desired option (e.g., Remove Duplicates).

**Example**:  
Remove duplicate rows based on the **Employee Name** column.

**Steps**:

1. Home >> Remove Rows >> Remove Duplicates.
2. Choose the **Employee Name** column.
3. Click **OK** to remove rows where duplicate names exist.
4. **Sort**

**Purpose**:  
The Sort feature allows you to sort the rows in ascending or descending order based on the values in a selected column.

**Options Available:**

* **Sort Ascending**: Sorts from lowest to highest.
* **Sort Descending**: Sorts from highest to lowest.

**Steps**:  
Home >> Sort >> Sort Ascending (or Descending)

**Example**:  
Sort the dataset by the **Salary** column in ascending order.

**Steps**:

1. Home >> Sort >> Sort Ascending.
2. Select the **Salary** column and click **OK**.
3. **Split Column**

**Purpose**:  
Split the contents of a column into multiple columns based on a delimiter, number of characters, or other rules.

**Options Available:**

* **By Delimiter**: Split by a specific character (e.g., comma, space).
* **By Number of Characters**: Split after a set number of characters.

**Steps**:  
Home >> Split Column >> Choose the desired option (e.g., By Delimiter).

**Example**:  
Split the **Name** column into two columns (First Name, Last Name) based on the space delimiter.

**Steps**:

1. Home >> Split Column >> By Delimiter.
2. Choose **Space** as the delimiter.
3. Click **OK** to split the **Name** column into First Name and Last Name.
4. **Group By**

**Purpose**:  
This feature allows you to group rows based on specific column values and perform aggregate functions like sum, count, average, etc.

**Options Available:**

a) **Group By Basic**: Group by one or more columns and summarize the data with an aggregate function.  
b) **Group By Advanced**: Perform multiple aggregation functions on the grouped data.

**Steps**:  
Home >> Group By >> Choose the desired grouping method.

**Example**:  
Group the data by **Department** and calculate the total salary for each department.

**Steps**:

1. Home >> Group By.
2. In the Group By window, select **Department** as the grouping column.
3. Set the aggregation to **Sum** and choose **Salary** as the column to sum.
4. Click **OK**.
5. **Data Type**

**Purpose**:  
Change the data type of a column to ensure proper formatting and compatibility for calculations and analysis.

**Options Available:**

a) **Text**: Set the column as text (for strings).  
b) **Number**: Set the column as a number (for calculations).  
c) **Date**: Set the column as a date (for date operations).  
d) **Time**: Set the column as time.

**Steps**:  
Home >> Data Type >> Choose the desired data type.

**Example**:  
Change the **Salary** column data type to **Number** for calculations.

**Steps**:

1. Home >> Data Type >> Number.
2. Select the **Salary** column to apply the numeric data type.
3. **Use First Row as Headers**

**Purpose**:  
If the first row of your dataset contains column names, this feature promotes that row to become the header.

**Options Available:**

1. **Use First Row as Headers**: This promotes the first row of data to be the column headers.
2. **Use Headers as First Row**: This takes the current headers and converts them into a row in the dataset.

**Steps**:  
Home >> Use First Row as Headers

**Example**:  
Promote the first row as headers if the first row contains column names.

**Steps**:

1. Home >> Use First Row as Headers.
2. **Replace Values**

**Purpose**:  
Replace specific values in a column with new values.

**Options Available:** No additional sub-options.

**Steps**:  
Home >> Replace Values

**Example**:  
Replace the **Bonus** values of "N/A" with zero.

**Steps**:

1. Home >> Replace Values.
2. In the **Bonus** column, replace all "N/A" values with "0".
3. Click **OK** to apply the change.
4. **Merge Queries**

**Purpose**:  
The **Merge Queries** operation allows you to combine columns from two or more tables based on a matching column (or key) that they share. This is particularly useful when you want to bring data from one table into another based on related information.

**Options Available:**

a) **Merge Queries**: Merges the selected query with another table, creating a new column with values from the second table that match based on the chosen key.  
b) **Merge Queries as New**: Similar to **Merge Queries**, but this operation creates a new table with merged data without modifying the existing table.

**Steps**:  
Home >> Merge Queries >> Select the second table to merge with.

**Example**:  
Merge the uncleaned employee dataset with another table (e.g., department data) based on the Department\_ID column.

**Steps**:

1. Home >> Merge Queries.
2. Select the table to merge with (e.g., Department Data).
3. Choose the matching column (Department\_ID in both tables).
4. Select **Join Kind** (e.g., Left Outer Join).
5. Click **OK** to merge the data into the original table, adding a new column with the merged data.
6. **Append Queries**

**Purpose**:  
The **Append Queries** operation allows you to combine rows from two or more tables. It adds the rows from one table to another table to create a longer dataset.

**Options Available:**

a) **Append Queries**: Appends the rows from one table into another, adding them to the bottom of the dataset.  
b) **Append Queries as New**: Similar to **Append Queries**, but this creates a new table with appended data, leaving the original table unchanged.

**Steps**:  
Home >> Append Queries >> Choose the table(s) to append.

**Example**:  
Append a smaller dataset (e.g., an additional 5 rows of employee data) to the uncleaned employee dataset.

**Steps**:

1. Home >> Append Queries.
2. Select the table to append (e.g., the smaller employee dataset).
3. Click **OK** to append the rows from the smaller dataset to the uncleaned employee dataset, extending the total number of rows.
4. **Combine Files**

**Purpose**: Combine data from multiple files (e.g., CSV or Excel) into one query.

**When to Use**: Use this option when you have multiple files with a similar structure (same columns) and want to consolidate them into a single dataset for analysis. This is particularly useful when dealing with periodic reports or data exports that are stored separately.

**Options Available**: No additional sub-options.

**Steps**:

1. **Home** >> **Combine Files**.
2. Select the folder containing the files you want to combine.
3. Click **OK** to combine all files into one dataset.

**Example**: Combine multiple CSV files with employee information (e.g., Employee1.csv, Employee2.csv, etc.) that all have the same structure (columns like EmployeeID, Name, Department, and Salary).

**Type of Table Needed**: Ensure that all files have the same column names and data types. For example, if one file has columns named ID, Full Name, Dept, and Pay, while another has EmployeeID, Name, Department, and Salary, the combining process may fail or result in inconsistencies.

**Example Steps:**

1. **Home** >> **Combine Files**.
2. Select the folder containing the CSV files (e.g., a folder named "EmployeeReports").
3. Ensure all files are structured similarly (e.g., each file contains EmployeeID, Name, Department, and Salary).
4. Click **OK** to combine all files into one consolidated dataset, creating a new table that includes all employee records from the combined files.
5. **Close & Apply:**

In Power BI, the **Close & Apply** dropdown in the Power Query Editor has three options:

1. **Close & Apply**:
   * This option closes the Power Query Editor and applies all changes and transformations made to the data. The modified data will be loaded into the Power BI model and will be available for creating reports and visualizations.
2. **Close**:
   * This option closes the Power Query Editor without applying any changes. If you have made modifications to the data but haven’t applied them, those changes will be discarded, and the original data will remain unchanged in your Power BI model.
3. **Apply**:
   * This option applies the changes made in the Power Query Editor without closing the editor. This is useful when you want to see the effect of your transformations in the Power BI model but continue working in the Power Query Editor.