

# Advanced Data Preparation and Transformations with Power Query (M)

As an intermediate user, you should already have a good foundation in the basics of Power Query, including importing and shaping data, and using basic M functions such as filtering and transformation steps. You should also be familiar with concepts such as parameters, the advanced editor, and query folding. In this course, we will take your skills to the next level by diving deeper into these topics and introducing you to more advanced functions and practices.

Here are some of the key topics we will cover:

**Performance and Optimization:** As you work with larger and more complex data sets in Power Query, it's important to pay attention to performance. We will discuss best practices for optimizing Power Query queries, including minimizing the use of expensive operations and using query folding when possible. We will also introduce you to the Power Query diagnostics tool, which can help you identify and troubleshoot performance issues in your queries.

**Query Diagnostics Tool:** This tool is used to identify and fix common issues that can impact performance, such as long-running queries, large data volumes, and inefficient data transformations. You will also learn how to use the tool to monitor and optimize the memory usage of your queries, and how to use the tool to profile and optimize the performance of individual steps in your data transformation scripts.

**Incremental Refresh:** This feature allows you to refresh only the data that has changed since the last refresh, rather than the entire data set. This can significantly reduce the time and resources required to refresh large data models, and can make it practical to refresh data more frequently. We will discuss how to set up incremental refresh in Power Query and the considerations you need to consider when doing so.

**Query Folding & Lazy Evaluation:** Query folding is a process that allows Power Query to execute certain transformations on the data source, rather than in the client application, which can significantly improve the performance of your data models. Lazy evaluation is a technique that allows Power Query to postpone the evaluation of certain expressions until they are needed, which can also improve performance and reduce memory usage.

**Working with Complex Data Structures:** Complex data structures, such as nested lists and records, can be challenging to work with and can impact the performance of your data models. You will learn how to use Power Query (M) functions and features to handle and transform complex data structures, including how to flatten nested data, how to extract and pivot data from complex structures, and how to use the M language to manipulate data at a low level.

**Advanced M functions:** Power Query's M language offers a wide range of functions for manipulating and transforming data. We will cover more advanced M functions for working with lists and records (such as List.Generate), for creating intelligent sorting and deduplication (such as Table.Buffer), and functions for working with text and data types (such as Text.Combine),

**Advanced Editor Techniques:** The Power Query Editor is a powerful tool that allows you to build, test, and debug data transformation scripts using the M language. In this topic, you will learn about advanced features and techniques for working with the Power Query Editor, including how to use the editor to create custom functions and scripts, how to use the editor to troubleshoot and debug data transformation scripts, and how to use the editor to optimize and improve the performance of your data models.