Optimize Model Performance

Learning Objectives

By the end of this lesson, you will be able to:

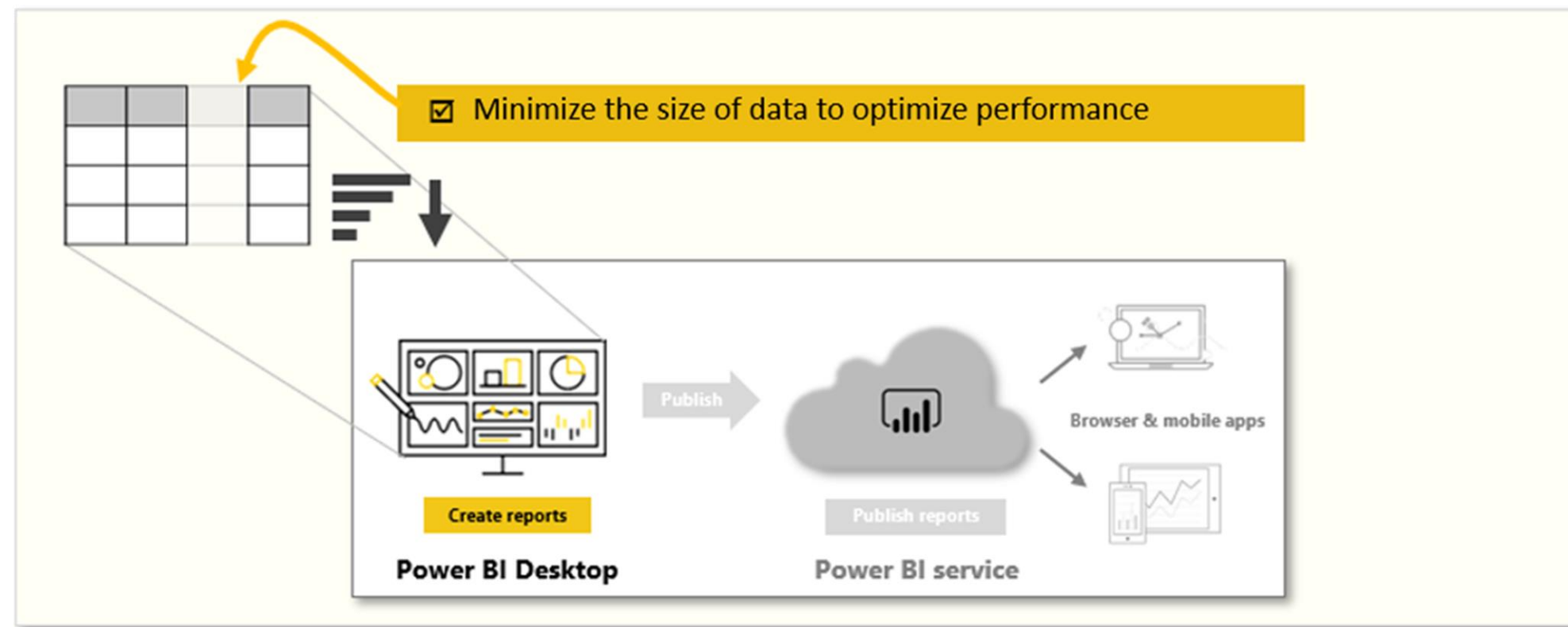- List the tasks involved in performance optimization

- Review performance results

- Identify and reduce cardinality levels

- Create and manage aggregations

Data Model Performance Optimization

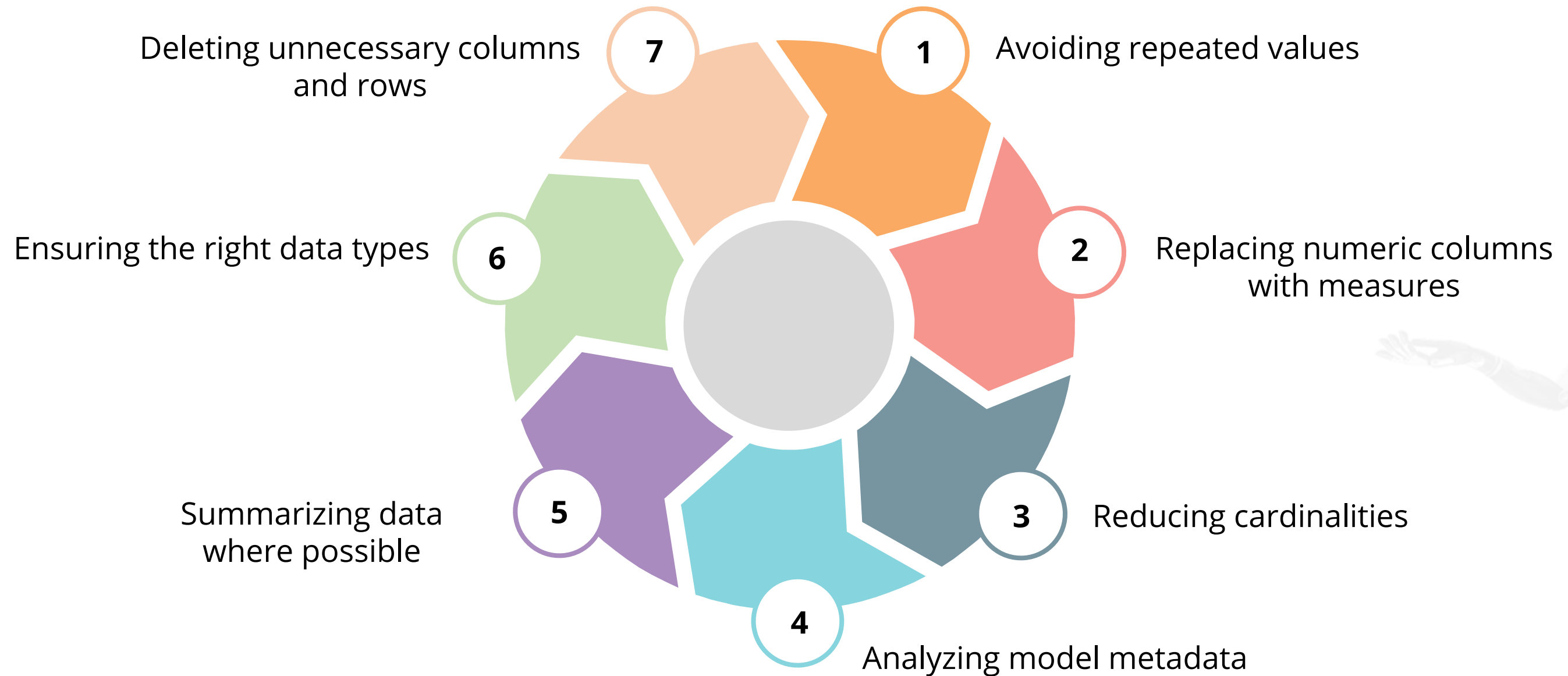# Introduction to Performance Optimization

Performance optimization involves making changes to the current state of the data model to run more efficiently.

An optimized data model performs better during execution.

Introduction to Performance Optimization

It involves minimizing the size of the data model that includes:

Deleting unnecessary columns and rows — **7**

Avoiding repeated values — **1**

Replacing numeric columns with measures — **2**

Ensuring the right data types — **6**

Reducing cardinalities — **3**

Summarizing data where possible — **5**

Analyzing model metadata — **4**
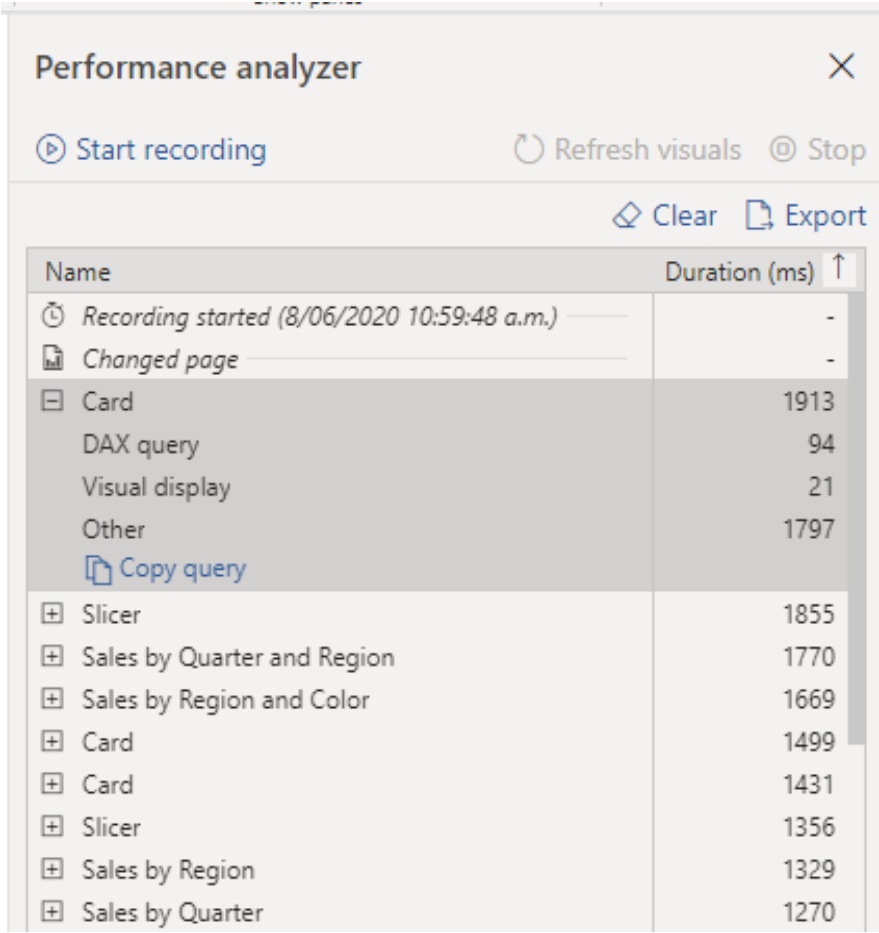
simplilearn

Performance Analyzer

Performance analyzer identifies measures, relationships, and visuals whose performance is poor. These are later optimized for improving the performance.

- Identify the performance of each report element

- Measure report elements during user interaction

- Detect which aspects are least or most resource intensive

Review Performance Results

The performance test results are reviewed in the performance analyzer pane.

To review the tasks in the order of duration, right-click the **Sort** icon next to the **Duration (ms)** column header, and then select **Total time** in descending order.

Review Performance Results

The log information shows the different durations to complete the following tasks:

DAX Query
The total time taken by the visual to send a query and the analysis services to return the results
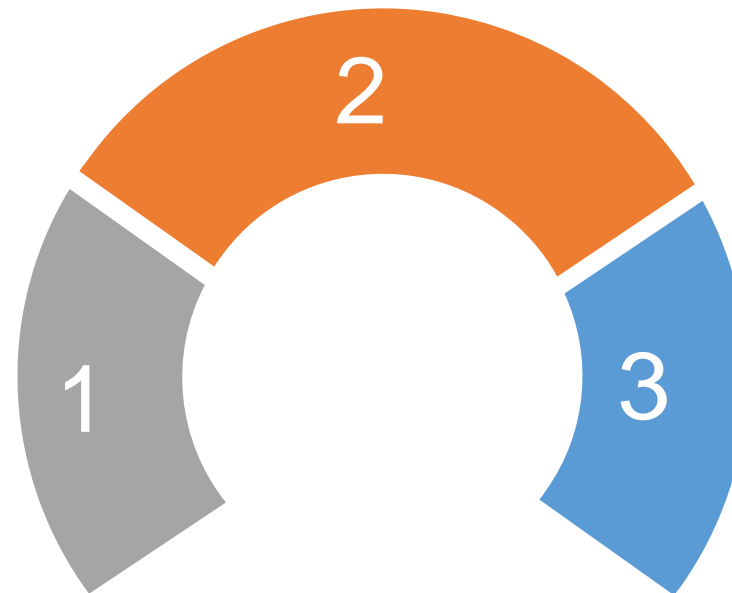
Other
The time taken for the visual to prepare queries, wait for other visuals to complete, or perform other background processing tasks
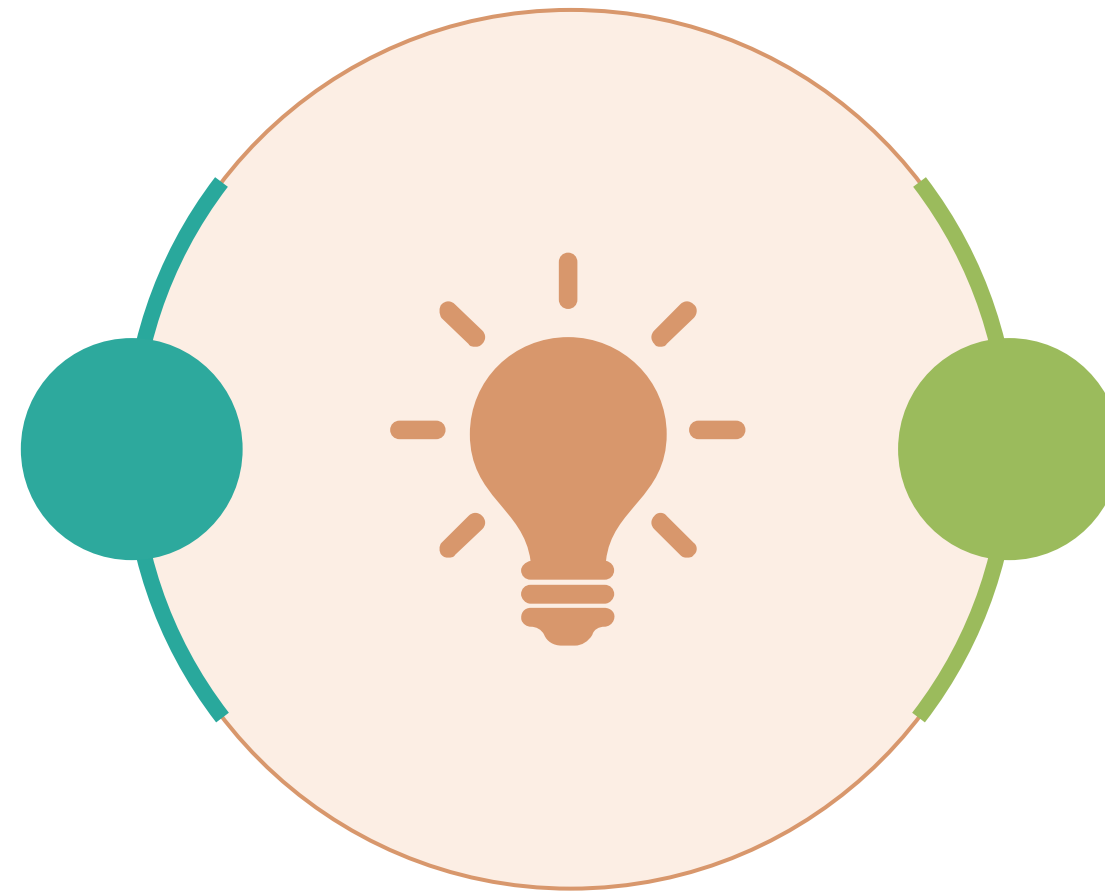
Visual Display
The time taken for the visual to render on screen

Removal of Unnecessary Rows and Columns

Power Query includes the below options:

Unnecessary Columns
Evaluates the need for each column

Unnecessary Rows
Checks the first few rows for null values or any unnecessary data
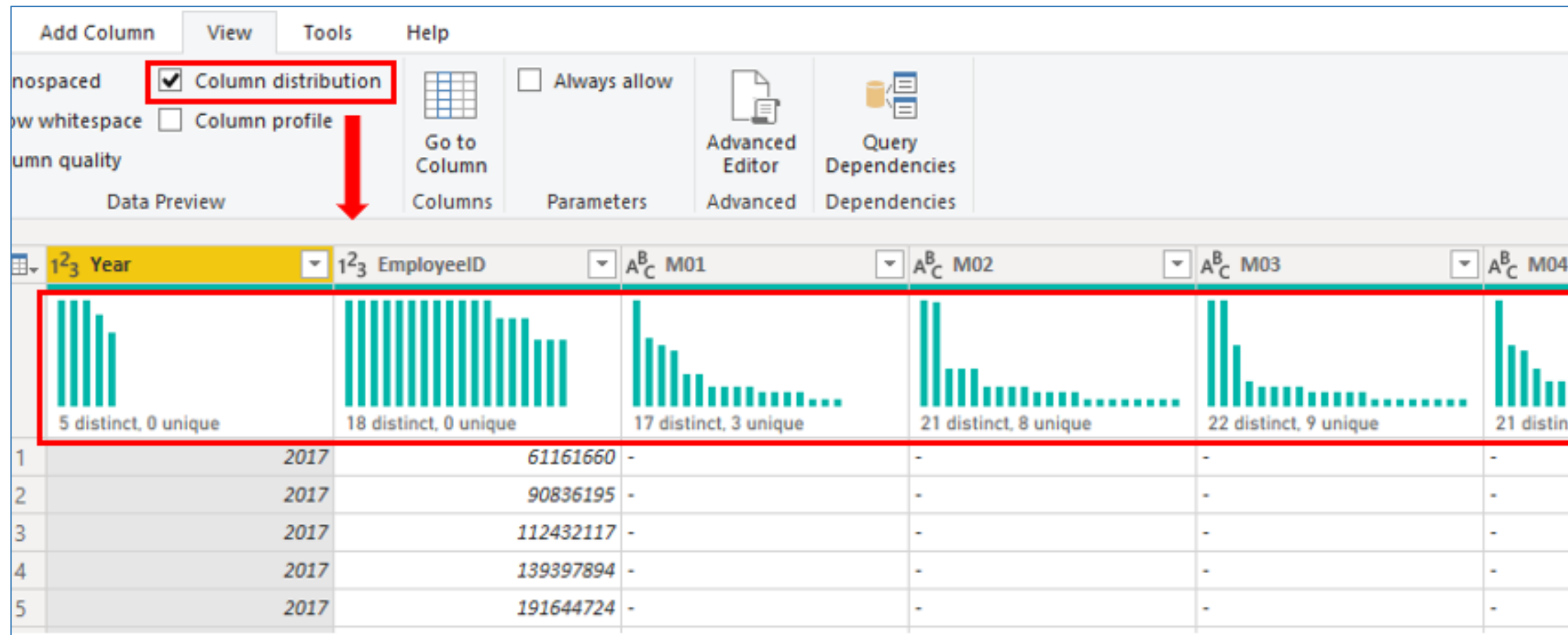
Cardinality Reduction

Cardinality

Cardinality describes the uniqueness of the values in a column.

It is also used in the context of the relationships between two tables to describe its direction.

Cardinality Levels

The column distribution on the **View** tab displays statistics on unique and distinct items present in each column of the data.

Cardinality Levels

Values distributed in a column are of two types:

Distinct Values Count

- Number of different values in a column

Unique Values Count

- Number of values that only appear once in a column

A column that has a lot of repeated values in this range has low level of cardinality.

A column that has a lot of unique values in this range has high level of cardinality.
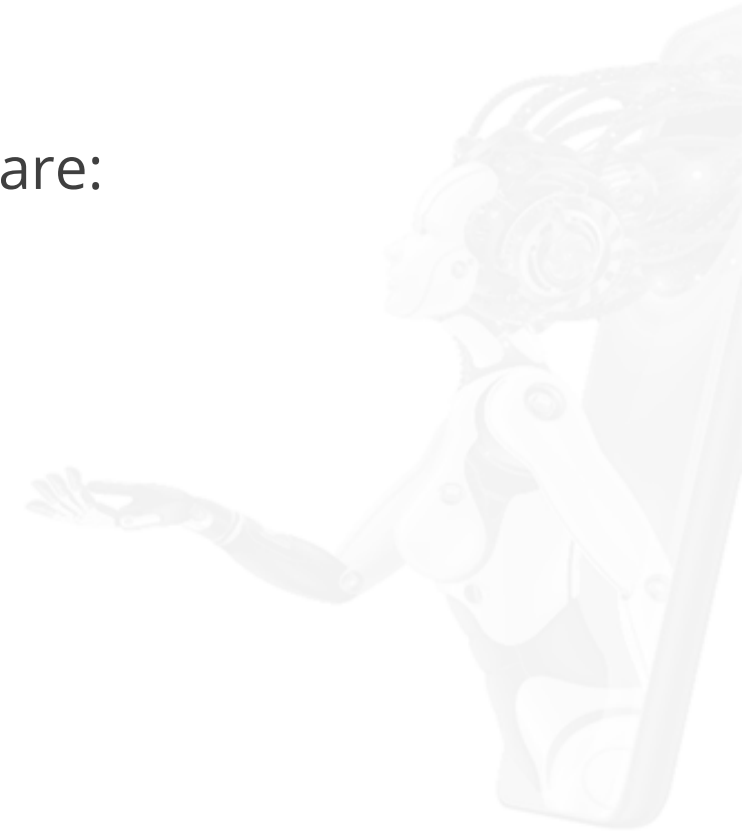
Cardinality Relationship

Relationships between the imported tables are necessary to calculate accurate results and depict the correct information.

The different cardinality options are:

- Many-to-one (*:1)

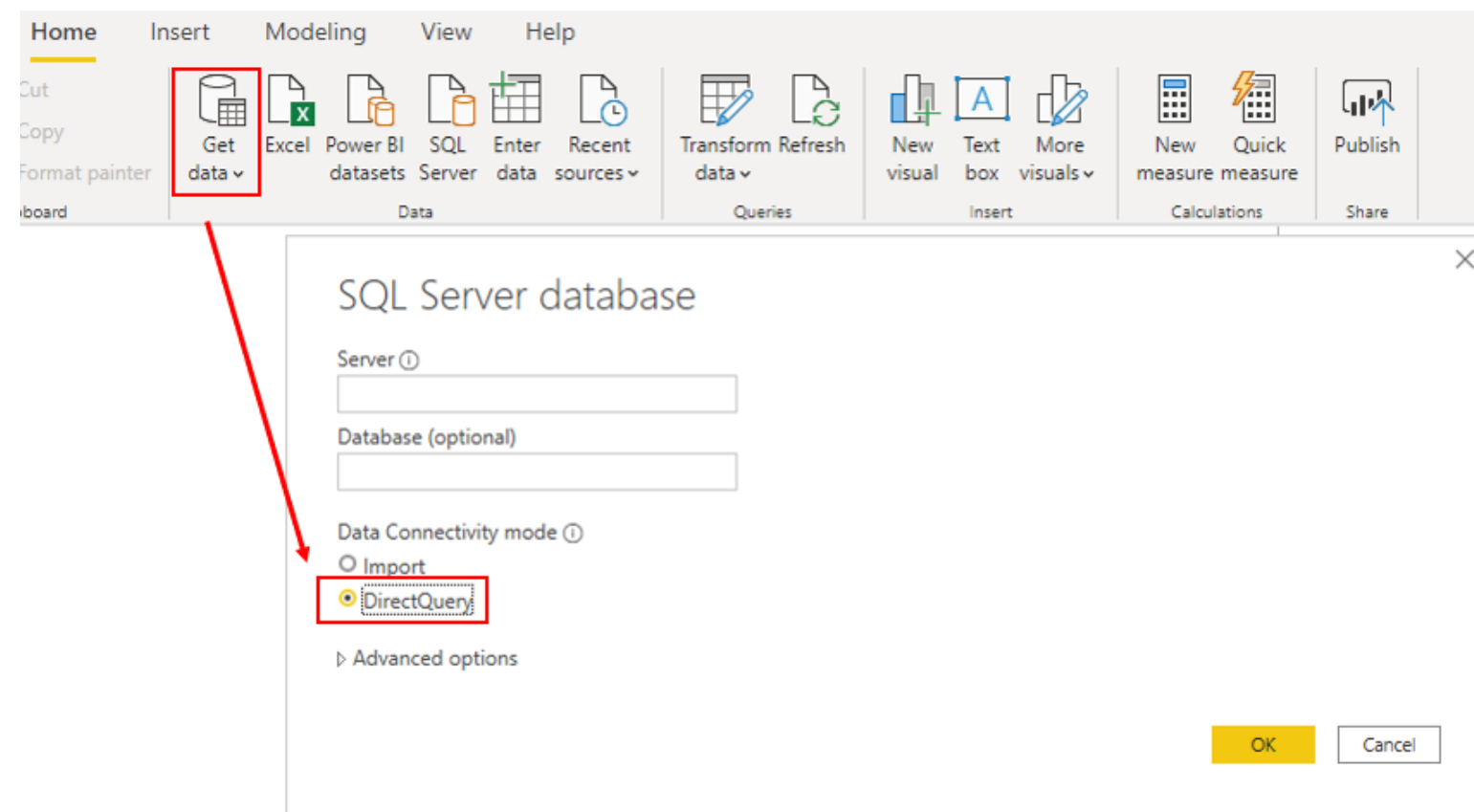- One-to-one (1:1)

- One-to-many (1:*)

- Many-to-many (*:*)

Optimize DirectQuery Model

Introduction to DirectQuery

DirectQuery is a method of connecting data directly to its source repository from the Power BI Desktop.

It is an alternative to importing data into Power BI Desktop.
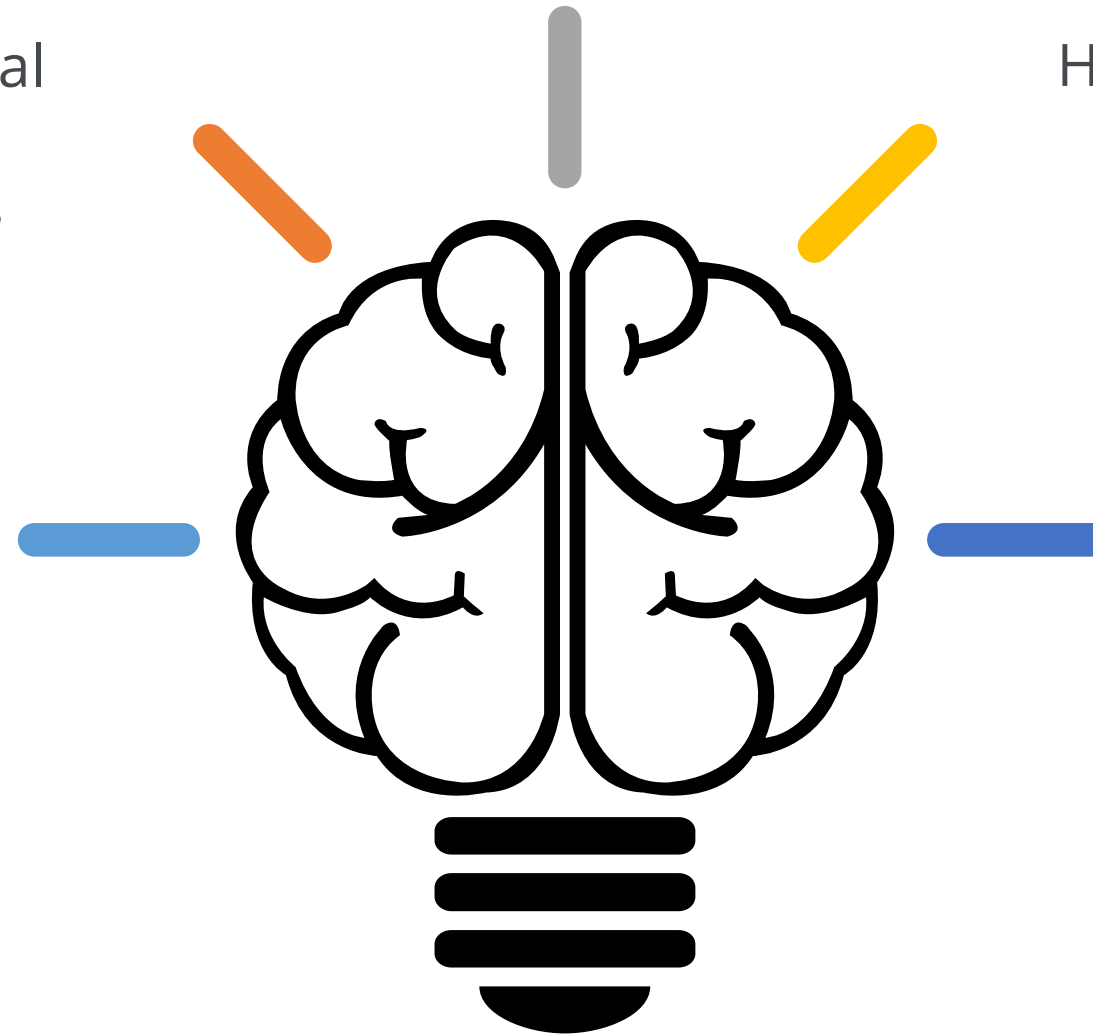
Benefits of DirectQuery

Applies data sovereignty restrictions to comply with legal requirements

Is used with multi-dimensional data sources that contain measures like SAP business

Handles a large amount of data without the need of pre-aggregate

Allows to create visualizations over large datasets

Is suitable in situations where data changes frequently and real-time reporting is required

simplilearn

Limitations of DirectQuery

Transformation
There are limitations to data transformation techniques.

Performance
There is data source dependency for the performance of DirectQuery.

Modeling
There are limitations to modeling capabilities or are not supported.

Security
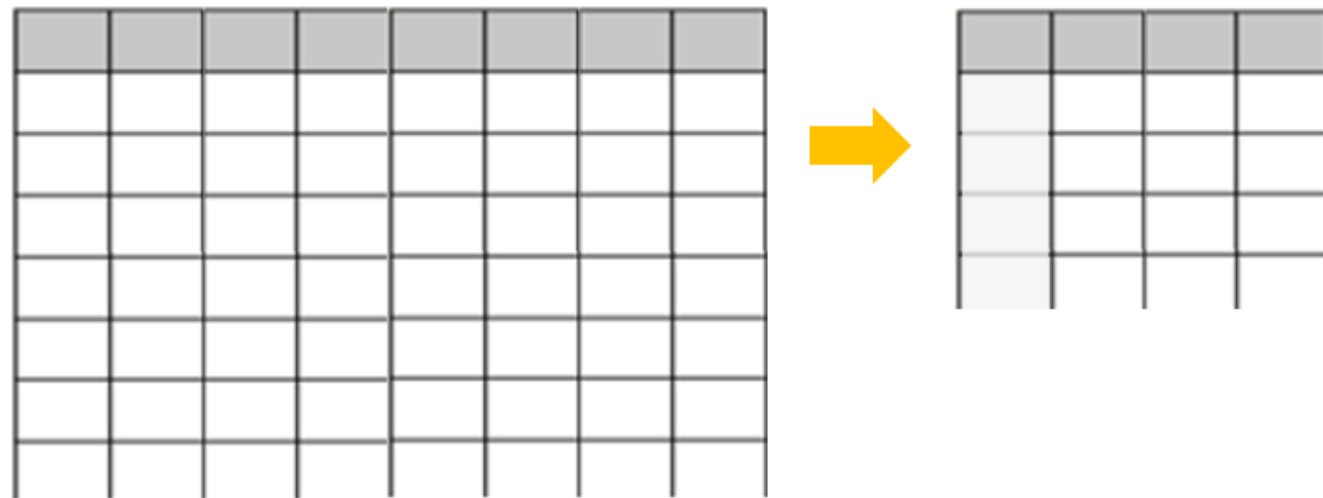It is difficult to understand how data travels from source to destination.
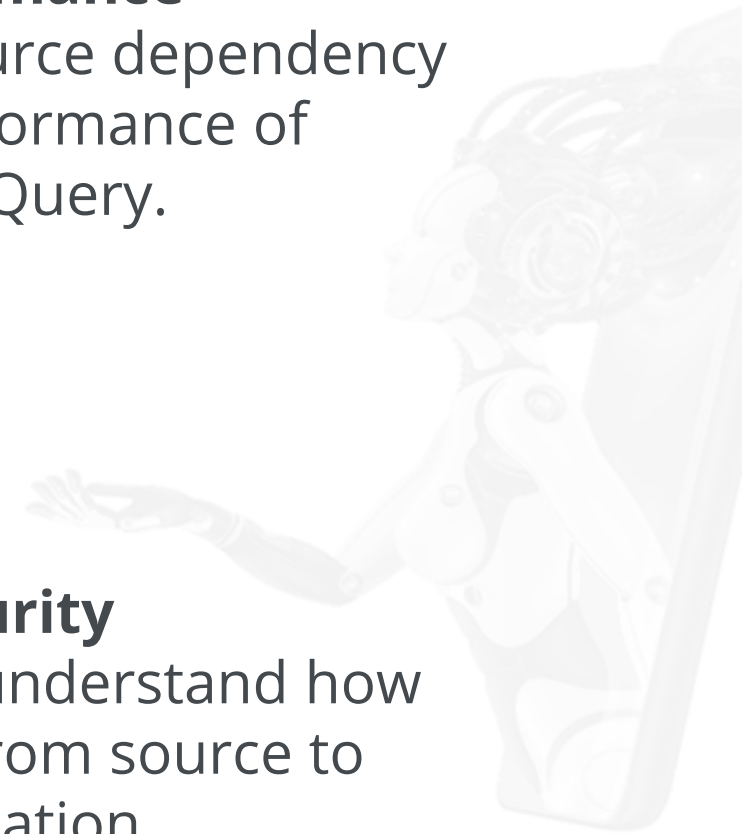
Create and Manage Aggregations

Introduction to Aggregations

Aggregation refers to summarizing data and presenting it at a higher grain or level.

- Aggregation reduces the table sizes in the data model and helps improve query performance.

Aggregation Use Cases

While dealing with large amounts of data, aggregation provides better query performance and helps derive insights.

When the size of the data model increases, aggregation helps to reduce the issues of performance, refresh issues, and overall query problems.
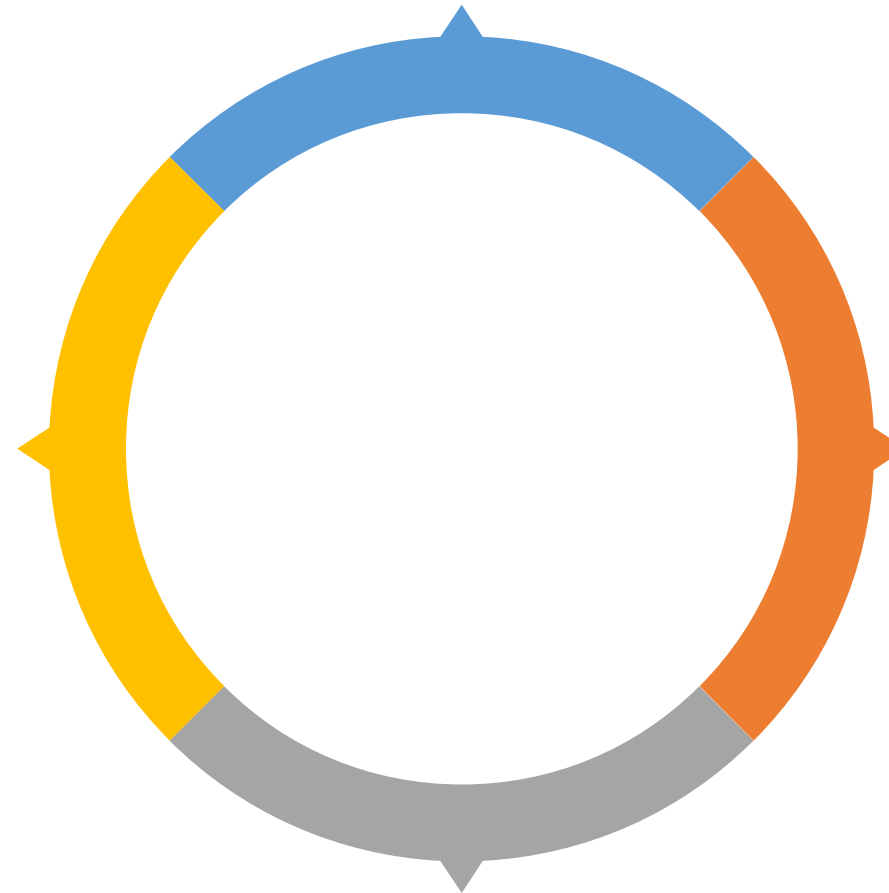
During slow refreshes, aggregation helps speed up the refresh process.

While using large data models, aggregation helps reduce and maintain the size of the model.

Creating Aggregations

Power Query Editor creates aggregations in Power BI Desktop.

Managing Aggregations

In the **Fields** pane, right-click the table and then select **Manage aggregations**.

Query Diagnostics

Analyze Query Plans

In the **Fields** pane, right-click the table and then select **Manage aggregations**.

⊟ Sales by Year	270
DAX query	2754
Visual display	57
Other	160
📋 Copy query	

- When you examine the results in the performance analyzer pane, it shows the duration taken by the Power BI Desktop engine to evaluate each query.

DAX Query

Performance analyzer highlights potential issues, but it does not provide a solution to improve.

- DAX Studio is used to investigate queries in detail.
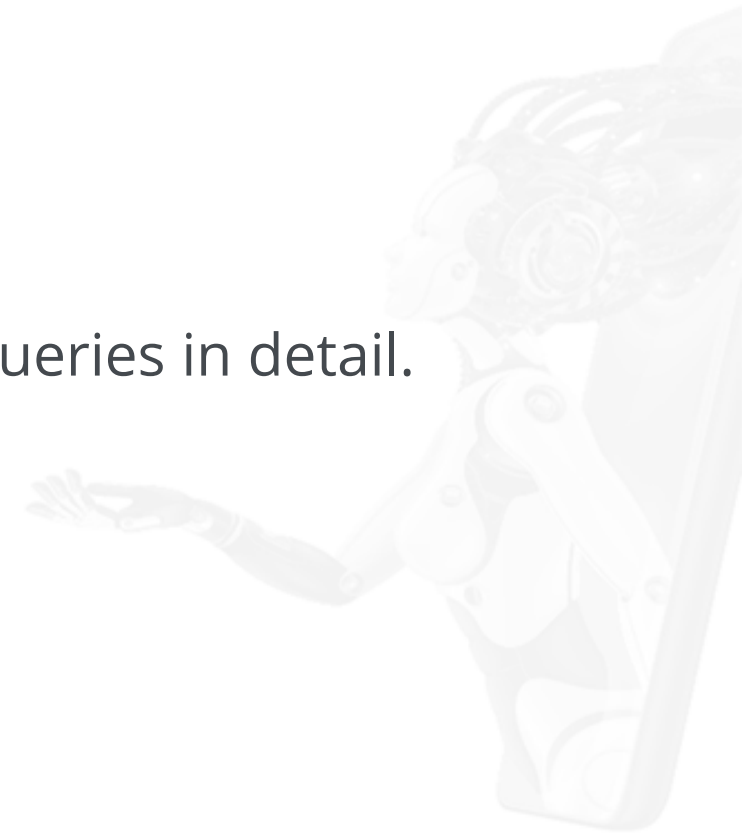
DAX Query: Example

Sales by Year	270
DAX query	**2754**
Visual display	57
Other	160
📋 Copy query	

```
Count Customers =
CALCULATE ( DISTINCTCOUNT (
Order[ProductID] ), FILTER ( Order,
Order[OrderQty] >= 5 ) )
```

Sales by Year	270
DAX query	**54**
Visual display	57
Other	160
📋 Copy query	

```
Count Customers =
CALCULATE ( DISTINCTCOUNT (
Order[ProductID] ), KEEPFILTERS
 (Order[OrderQty] >= 5 ) )
```

Knowledge Check

When does an optimized model perform better?

A.   During execution

B.   After execution

C.   Before execution

D.   During initiation

Knowledge Check

1

When does an optimized model perform better?

A. During execution

B. After execution

C. Before execution

D. During initiation

The correct answer is **A**

An optimized model performs better during execution.

What is the term used to refer to the uniqueness of the values in a column?

A.   DX Query

B.   Cardinality

C.   Aggregation

D.   DirectQuery

Knowledge Check

2

What is the term used to refer to the uniqueness of the values in a column?

A.   DX Query

B.   Cardinality

C.   Aggregation

D.   DirectQuery

The correct answer is   **B**

Cardinality is the term used to refer to the uniqueness of the values in a column.

Which of the following refers to summarizing data and presenting it at a higher level?

A.   DX Query

B.   Cardinality

C.   Aggregation

D.   DirectQuery

Which of the following refers to summarizing data and presenting it at a higher level?

A.   DX Query

B.   Cardinality

C.   Aggregation

D.   DirectQuery

The correct answer is   **C**

Aggregation refers to summarizing data and presenting it at a higher level.

Key Takeaways

◉ Performance optimization involves making changes to the current state of the data model to run more efficiently.

◉ A column that has a lot of repeated values in this range has a low level of cardinality.

◉ The different cardinality relationships are many-to-one, one-to-one, one-to-many, and many-to-many.

◉ DirectQuery is a method of connecting data directly to its source repository from the Power BI Desktop.

simplilearn