

BEST PRACTICES

IN POWER BI

Swipe →

INTRODUCTION

Power BI has become an essential tool for data analysis and visualization, enabling businesses to transform raw data into insightful reports and dashboards.

However, to fully leverage its capabilities, adhering to best practices is crucial. This article delves into these best practices, focusing on optimizing data models, enhancing DAX expressions, improving performance, and ensuring a user-friendly layout.

By following these guidelines, users can create efficient, maintainable, and high-performing Power BI reports that meet organizational needs and drive informed decision-making. Whether you're a novice or an experienced user, these practices will help you maximize the potential of Power BI.

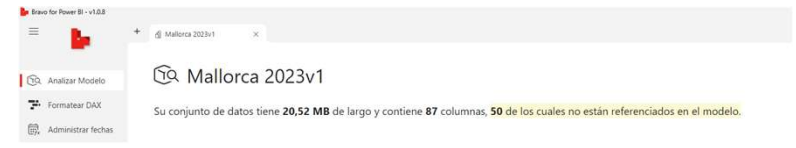
Do you want to know how it works? Keep reading

DATA MODEL OPTIMATIZTION

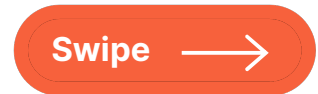
Simplification: Keep the data model as simple as possible, avoiding unnecessary tables and removing unused columns.

It's crucial for both performance and maintenance.

A simplified data model reduces the complexity of your reports, making them easier to manage and faster to execute.



COLUMNA	TABLA	CARDINALID...	TAMAÑO	PESO
IdVenta	fVentas	121.32 K	4.45 MB	22 %
Email	dClientes	18.48 K	888.84 KB	4 %
Contacto	dClientes	19.23 K	754.79 KB	4 %
CIF	dClientes	19.12 K	680.68 KB	3 %
IdDireccion	dVentas	19.12 K	670.18 KB	3 %
Direccion	dClientes	13.59 K	664.85 KB	3 %
Telefono	dClientes	8.97 K	353.10 KB	2 %
Longitud	dDireccion	7.74 K	345.83 KB	2 %
Latitud	dDireccion	7.68 K	342.41 KB	2 %
Latitud	dClientes	7.52 K	339.50 KB	2 %
Longitud	dClientes	7.46 K	336.23 KB	2 %
1.2 TMargenComercial	fVentas	1.61 K	181.56 KB	1 %
1.2 Bruto	fVentas	1.45 K	148.02 KB	1 %
		257.52 K	10.81 MB	53 %

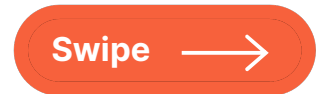


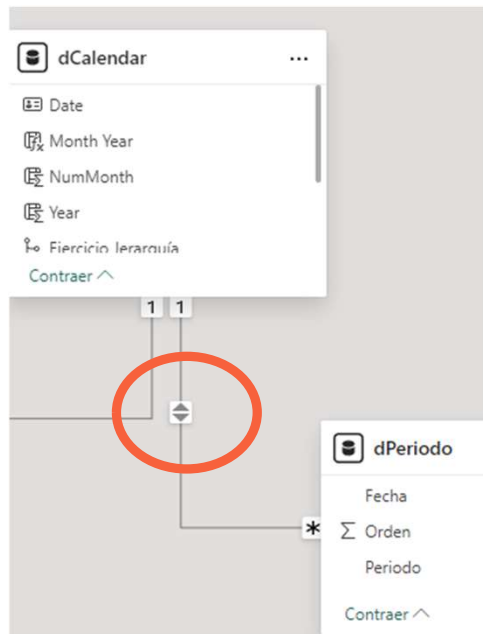
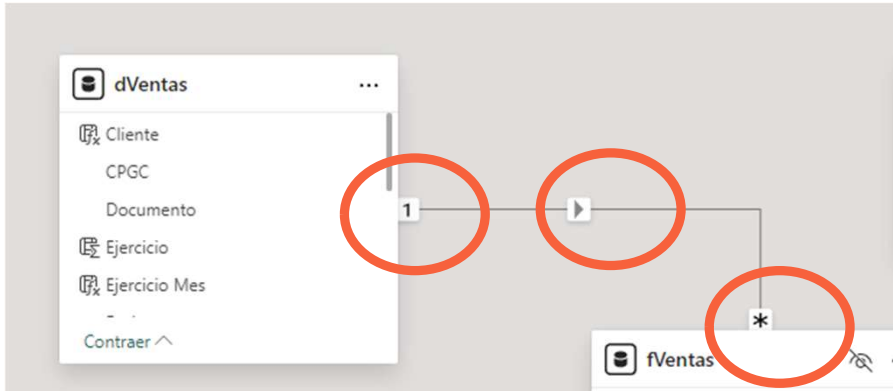
DATA MODEL OPTIMATIZTION



The Bravo tool helps you find tables, columns, and formulas that are not used in your model, so you can go to Power Query and remove them. If you do this, you will always be able to recover them later in Power Query for future analyses where you might need them.

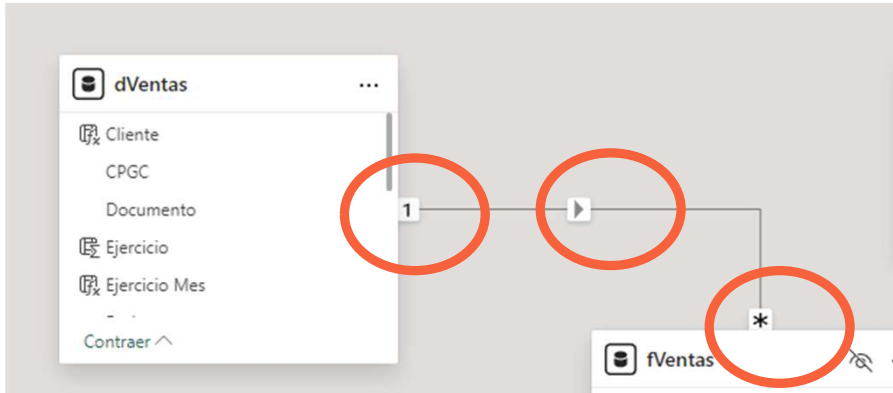
COLUMNA	TABLA	CARDINALID...	TAMAÑO	PESO
IdVenta	fVentas	121.32 K	4.45 MB	22 %
Email	dClientes	18.48 K	888.84 KB	4 %
Contacto	dClientes	19.23 K	754.79 KB	4 %
CIF	dClientes	19.12 K	680.68 KB	3 %
IdDireccion	dVentas	19.12 K	670.18 KB	3 %
Direccion	dClientes	13.59 K	664.85 KB	3 %
Telefono	dClientes	8.97 K	353.10 KB	2 %
Longitud	dDireccion	7.74 K	345.83 KB	2 %
Latitud	dDireccion	7.68 K	342.41 KB	2 %
Latitud	dClientes	7.52 K	339.50 KB	2 %
Longitud	dClientes	7.46 K	336.23 KB	2 %
1.2 TMargenComercial	fVentas	1.61 K	181.56 KB	1 %
1.2 Bruto	fVentas	1.45 K	148.02 KB	1 %
		257.52 K	10.81 MB	53 %



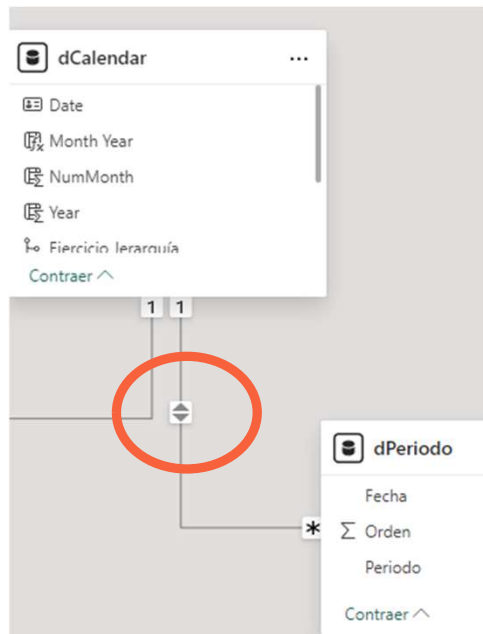


Relationships: Establish the correct relationships between the tables in your model. This is VITAL for accurate data analysis. Much more efficient: Unidirectional from many to one (Fact Table to Dimensions).

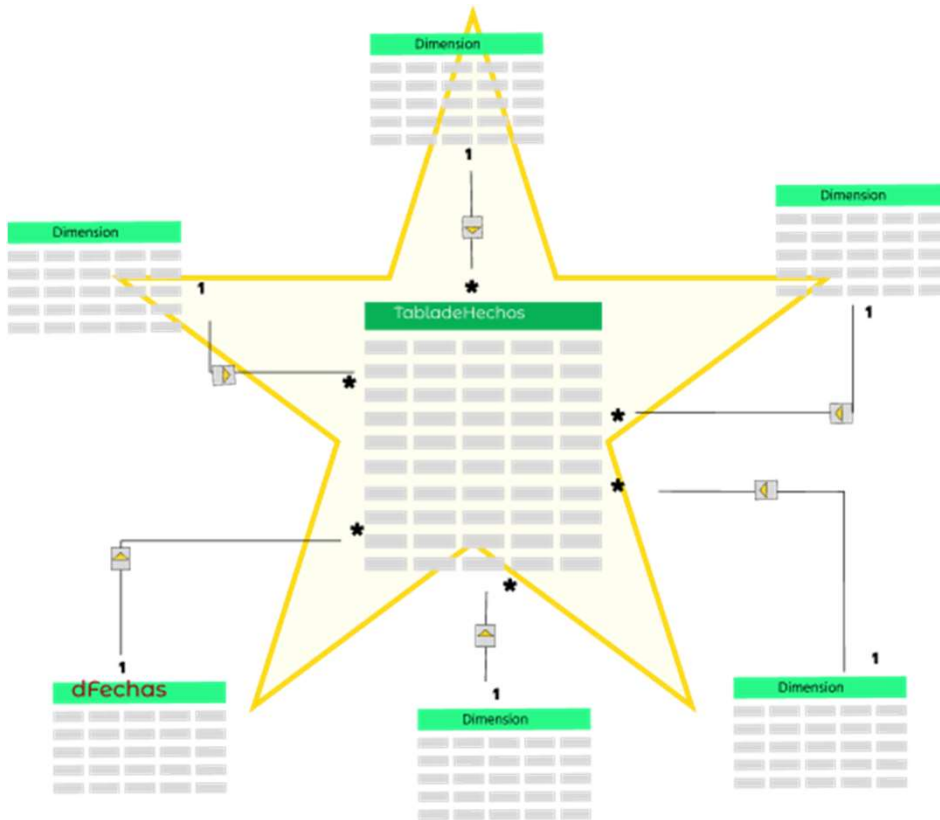
Benefits: Performance improvement: A simplified data model with optimized relationships ensures faster query execution and overall enhanced performance of Power BI reports.



Ease of Maintenance: Simplified models are easier to update and maintain, reducing the effort required for future modifications.



Data Integrity: Correctly defined relationships ensure data consistency and accuracy, which is crucial for obtaining reliable business insights



User Understanding:

A clear and simple data model is easier for end users to understand and interact with, improving their overall experience.

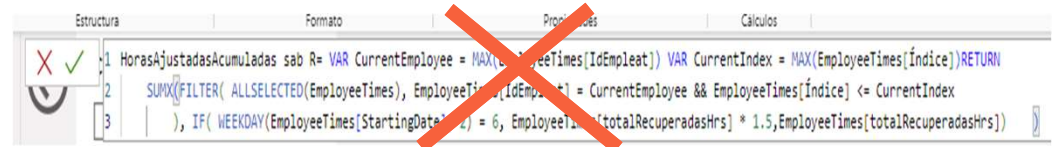
DAX Expressions:

Optimization:

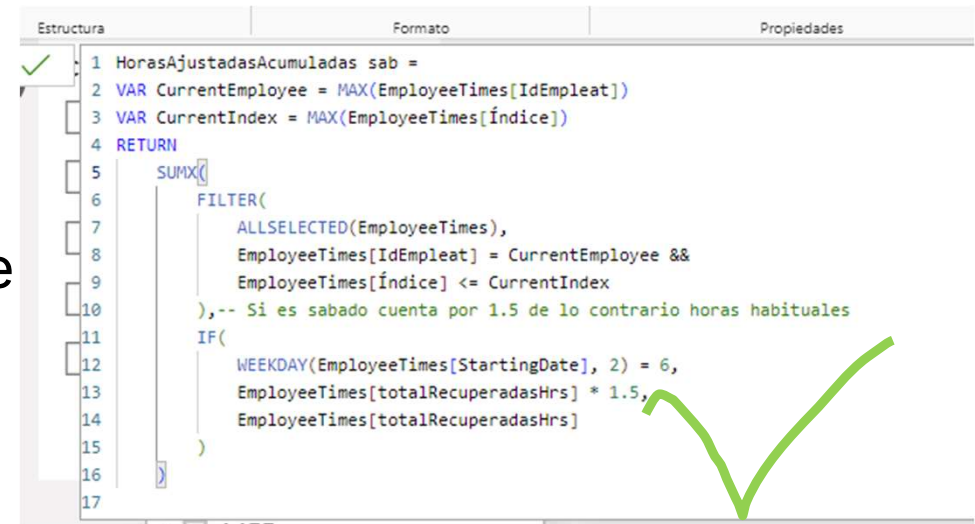
To enhance performance, avoid unnecessary iterations and use appropriate functions.

Readability:

Write clear and commented formulas to facilitate maintenance and understanding.



```
1 HorasAjustadasAcumuladas sab R= VAR CurrentEmployee = MAX(EmployeeTimes[IdEmpleat]) VAR CurrentIndex = MAX(EmployeeTimes[Índice])RETURN
2 SUMX(FILTER( ALLSELECTED(EmployeeTimes), EmployeeTimes[IdEmpleat] = CurrentEmployee && EmployeeTimes[Índice] <= CurrentIndex
3 ), IF( WEEKDAY(EmployeeTimes[StartingDate], 2) = 6, EmployeeTimes[totalRecuperadasHrs] * 1.5, EmployeeTimes[totalRecuperadasHrs])
```



```
1 HorasAjustadasAcumuladas sab =
2 VAR CurrentEmployee = MAX(EmployeeTimes[IdEmpleat])
3 VAR CurrentIndex = MAX(EmployeeTimes[Índice])
4 RETURN
5     SUMX(
6         FILTER(
7             ALLSELECTED(EmployeeTimes),
8             EmployeeTimes[IdEmpleat] = CurrentEmployee &&
9             EmployeeTimes[Índice] <= CurrentIndex
10        ), -- Si es sabado cuenta por 1.5 de lo contrario horas habituales
11        IF(
12            WEEKDAY(EmployeeTimes[StartingDate], 2) = 6,
13            EmployeeTimes[totalRecuperadasHrs] * 1.5,
14            EmployeeTimes[totalRecuperadasHrs]
15        )
16    )
17
```

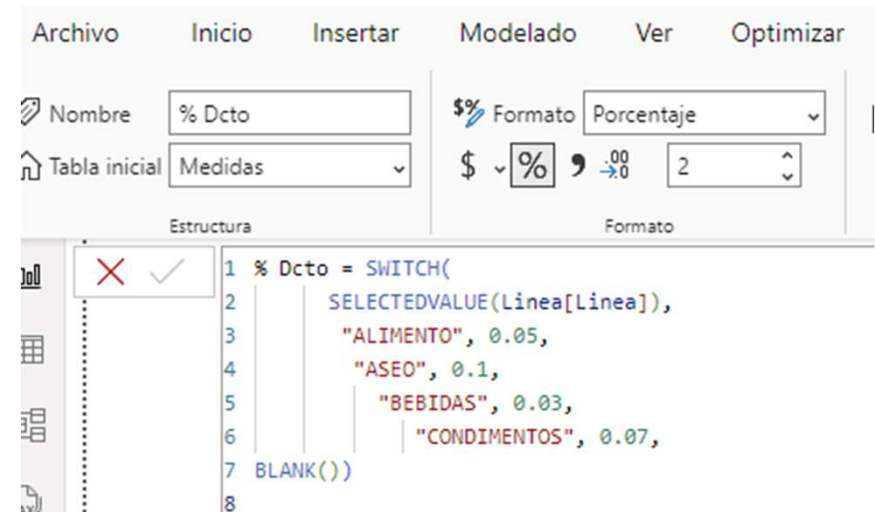

Data Formatting:

Consistent Formatting:

Apply formats to all visible columns and measures to ensure consistency and improve readability.

Data Types:

Avoid using floating-point data types to prevent precision issues



Incremental refresh and real-time data ✕

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). [Learn more](#)

ⓘ These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

1. Select table

Orders ▼

2. Set import and refresh ranges

Incrementally refresh this table

Archive data starting Years ▼ before refresh date

Data imported from 1/1/2017 to 11/17/2022 (inclusive)

Incrementally refresh data starting Days ▼ before refresh date

Data will be incrementally refreshed from 11/18/2022 to 11/20/2022 (inclusive)

3. Choose optional settings

Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

Selected table cannot be folded for DirectQuery.

Only refresh complete days [Learn more](#)

Detect data changes [Learn more](#)

4. Review and apply

5 years before
 Archived

3 days before
 Incremental Refresh

Refresh date

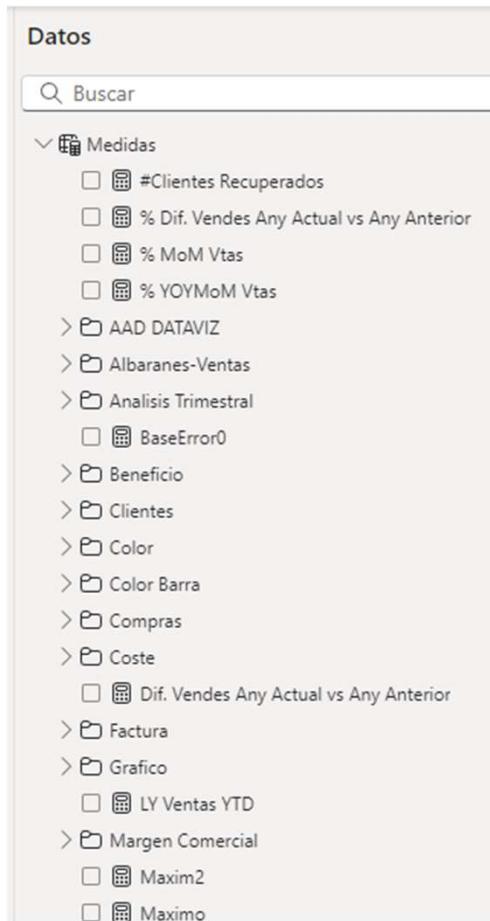
Performance:

Partitioning:

Partition large tables to improve query performance and processing.

Incremental Load:

Implement incremental loads to efficiently manage large volumes of data.



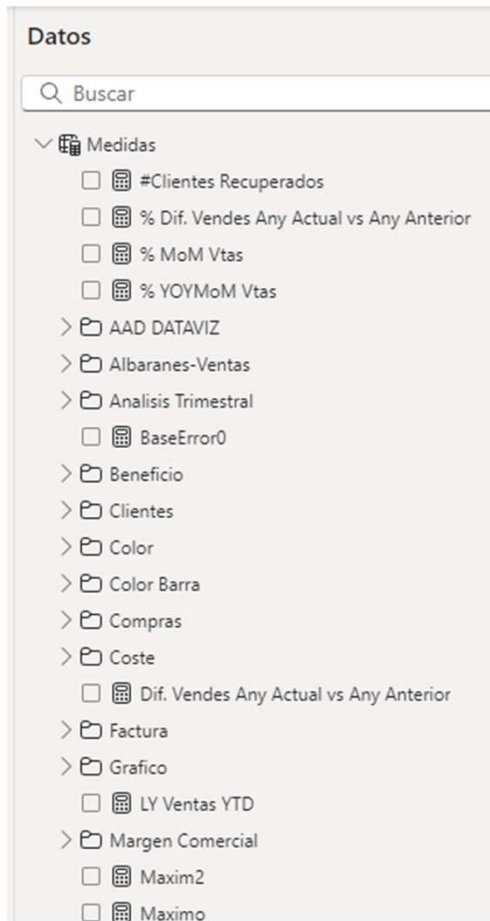
Design and Usability:

Visualization Folders:

Organize columns and measures into visualization folders for better navigation of the model.

Perspectives:

Use perspectives to simplify the model view for different users and use cases.



Metadata and Localization:

Descriptions and Translations:

Provide descriptions and translations for columns, measures, and folders to enhance understanding and usability.



WAS THIS HELPFUL?

**DON'T FORGET TO
SAVE THIS POST**



WWW.FP20ANALYTICS.COM