

What is Power BI?

Power BI is a business analytics service by Microsoft. It provides interactive visualizations and business intelligence capabilities with an interface simple enough for end-users to create reports and dashboards. It allows users to connect to multiple data sources, clean and transform data, create custom calculations, and visualize data through charts, graphs, and tables.

Power BI can be accessed through a web browser, mobile device, or desktop application and integrated with other Microsoft tools like Excel and SharePoint. With its powerful data modeling and analysis capabilities, Power BI enables organizations to make data-driven decisions and gain insights into their business performance.



What is Tableau?

Tableau is a data visualization and business intelligence tool that enables users to connect, visualize and share data in a highly interactive and intuitive way. It allows users to quickly analyze and explore large and complex datasets using a drag-and-drop interface without requiring coding or programming skills. Tableau provides a wide range of chart types and visualization options, such as line charts, bar charts, maps, scatter plots, and many more.

Tableau can connect to various data sources, including databases, spreadsheets, big data platforms, and cloud services. It also allows users to perform data cleaning and transformation tasks, create custom calculations, and generate insights using advanced analytics features.

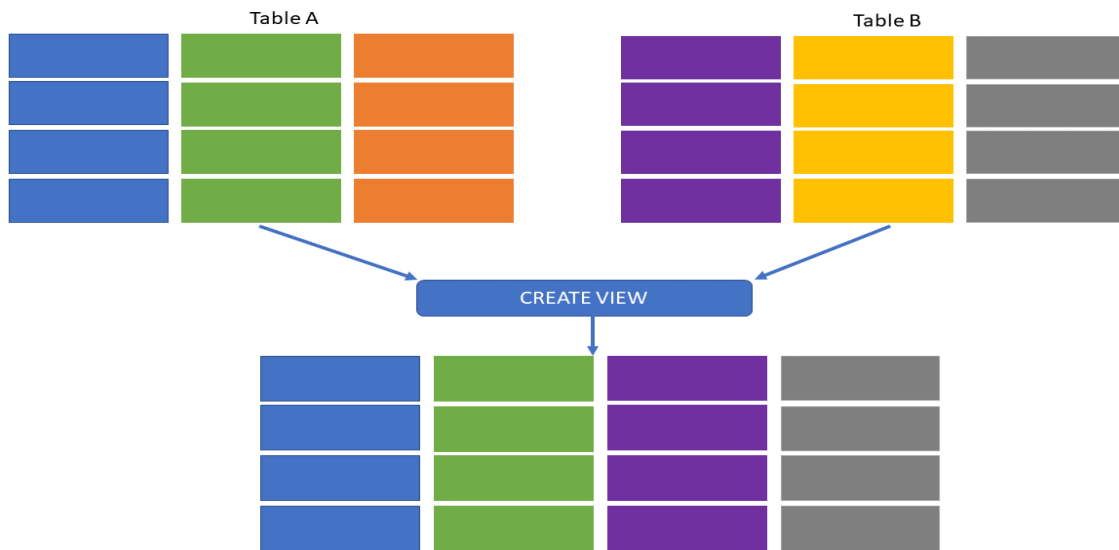
Tableau provides a suite of products, including Tableau Desktop for creating and publishing visualizations, Tableau Server for sharing and collaborating on data and visualizations across an organization, and Tableau Prep for preparing and cleaning data before analysis. With its user-friendly interface and powerful visualization capabilities, Tableau has become popular for businesses and organizations looking to make data-driven decisions.

Comparison Matrix: Tableau vs Power BI

Feature/Aspect	Tableau	Power BI
Ease of Use	Intuitive drag-and-drop interface but steeper learning curve for advanced features.	User-friendly interface with easy integration into the Microsoft ecosystem.
Data Integration	Connects to a wide range of data sources, both on-premises and cloud.	Excellent integration with Microsoft products and a variety of other data sources.
Performance	Strong performance with large datasets, though it can be resource-intensive.	Good performance, optimized for use within the Microsoft Azure cloud infrastructure.

Cost	Higher cost, especially for enterprise-level deployments.	A free version and cost-effective Pro licenses.
Visualization	Extensive and highly customizable visualization options.	Robust visualization capabilities, slightly less flexible than Tableau.
Community & Support	Large, active user community, extensive online resources, and professional support.	Strong community support, with robust Microsoft support and resources.
Embedded Analytics	Strong support for embedding dashboards into other applications.	Excellent embedding capabilities within the Microsoft ecosystem.
Collaboration	Good collaboration features with Tableau Server and Tableau Online.	Collaboration features, especially with integration into Microsoft Teams.
Data Preparation	Tableau Prep for data cleaning and preparation, powerful but separate from the main product.	Integrated data preparation features within Power BI, making it easier for end-to-end workflows.
Artificial Intelligence	Advanced AI features with Tableau AI and integration with R and Python.	Built-in AI capabilities like Power BI Q&A, AI visuals, and integration with Azure ML.
Deployment Options	On-premises, cloud, and hybrid deployments.	Cloud-based with on-premises options via Power BI Report Server.
Mobile Support	Strong mobile support with responsive dashboards and native apps.	Excellent mobile support, with a focus on mobile optimization and native apps.

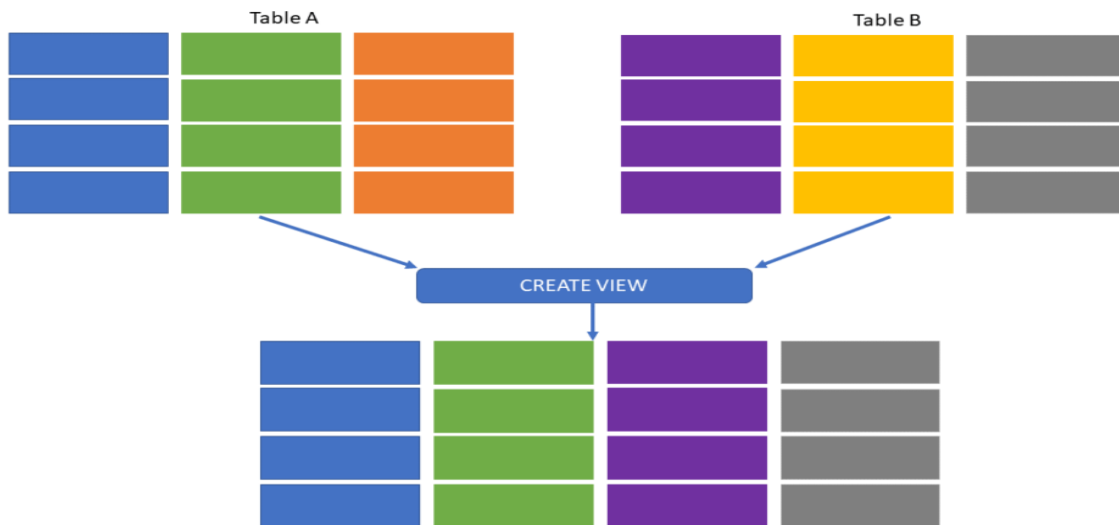
3 reasons to use views instead of tables in Power BI!



In recent months, I was frequently asked if one should use tables or views when *IMPORTING* data in Power BI. I've intentionally marked word *importing*, as the main purpose of this article is to compare usage of views vs tables in Import mode.

What is a view at first place?

I guess you all know what is a table, so I won't spend time explaining it. And, if you're coming from the SQL world, you probably know what is a view:)...However, there are many Power BI content creators with no SQL/databases background, so they deserve at least a basic introduction to views.

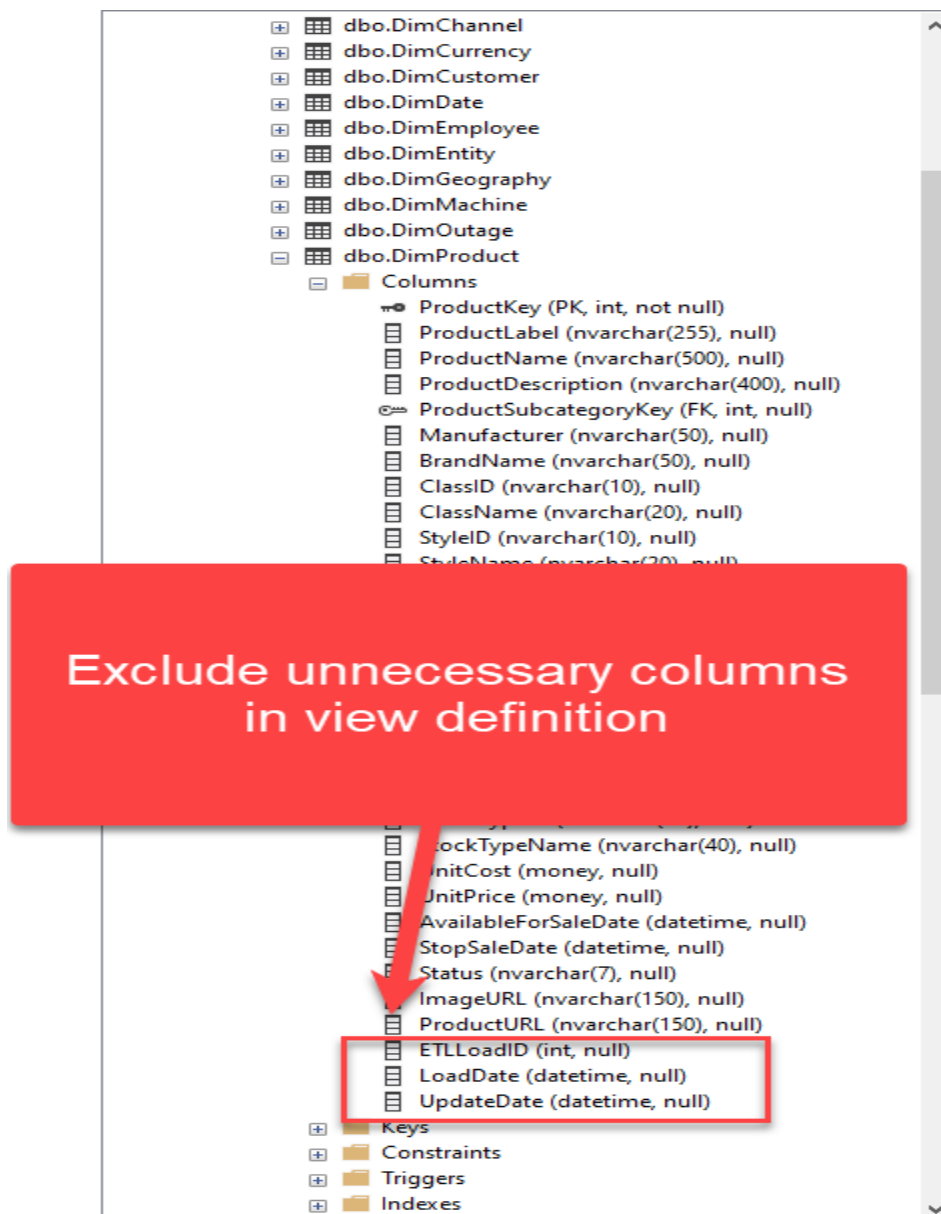


A database view is nothing else but the stored query over data from the database. This stored query can target both tables and other views (you can create a view that queries other views). This stored query (view definition) represents part of the database, but **it doesn't store any physical data**! This is the first important difference to a "regular" table – views don't store the data, which means that every time you need the data from the view, the underlying stored query will be executed against the database. Since views are being run each time you "call" them, they will always pick the relevant data from the underlying tables. That means you don't need to worry if something changed in the underlying table (deleted/updated rows), as you will always get the actual data from the tables.

Advantages of using views

In certain situations, views can provide multiple advantages over regular tables:

- Views can contain (and usually do) joins between multiple tables, producing a single result set, thus reducing the complexity
- Views may contain only a subset of the data (for example, you can create a view that contains only sales with amount greater than X; view containing currently open purchases; view to include only customers that registered in year X, etc.)
- You can use views to pre-aggregate data for your analytic workloads, using aggregate functions, such as SUM, AVG, COUNT, and so on
- Views can be used to control the access to an underlying data: for example, you can create a view that contains sales for the US region only, and then grant access to that view to relevant users. That way, you're not exposing all the data from the table, as you can restrict access to a table, while granting access to a view only
- Views don't consume space in your database (except some trivial amount of memory for storing



- Using view instead of table, you can ensure consistency between all the reports. For example, imagine that you want all your reports to display the value “Brand” instead of “BrandName” – you can define alias for the column in the view definition, thus avoiding additional step in Power Query Editor of each report to rename this column!

CREATE VIEW [dbo].[Product]

AS

SELECT ProductKey

,ProductLabel

,ProductName

,BrandName **AS** Brand

....

FROM dbo.DimProduct

GO

If you import the data from the table, one report creator can rename “BrandName” to “Brand”, the other will call it “Product”, the third will label it “Product Name”, and so on...I think you get the point! By standardizing column names in the view definition, you are ensuring that all reports will be consistent with the naming

- You can easily apply best practices for user-friendly naming of the objects. Instead of exposing users values such as “DimProduct”, “DimDate”, etc, or columns like “UnitOfMeasureName”, “AvailableForSaleDate”...you can (and should) implement user-friendly names, both by removing “Dim” prefix from dimension tables, and by giving your columns meaningful and easy-to-read names. Something like:

So, even though this looks like I’m not transforming my data, in fact, I’m doing exactly that! I’m shaping it and preparing it for convenient and more consistent consumption from Power BI.

Materialized/Indexed Views

I’ve intentionally left this for the end, as indexed/materialized views (the name depends on the relational database management system), represent the best from both worlds! They hold the logic of the view (joins, transformations, calculations), but they keep the result set persistent. Simply said, they physically store the data produced by executing the query. That means that you’ll get performance benefits (you can also create indexes on this type of view), but the trade-off is that materialized views consume space within the database.

The decision on using materialized vs regular views is not a simple one and depends on various factors. In any case, it needs to be carefully evaluated and discussed with your DBA/IT department.

Conclusion

When using a relational database as the data source for Power BI solutions, most report creators choose to take an easier path – import the data directly from tables, without even thinking or being aware of database views! Therefore, in this article, we demystified views and explained why they should be your preferred choice when importing data from the SQL database.