Best Practices for Power BI Desktop, Report Builder and Service

Guidelines for an Efficient and Effective Data Analysis Experience

Contents

[Power BI Desktop 2](#_Toc200471876)

[Data Preparation 2](#_Toc200471877)

[Report Design 2](#_Toc200471878)

[Performance Optimization 2](#_Toc200471879)

[Power BI Service 2](#_Toc200471880)

[Workspace Management 2](#_Toc200471881)

[Report Distribution 3](#_Toc200471882)

[Security and Governance 3](#_Toc200471883)

[Power BI Report Builder 3](#_Toc200471884)

[Data Source Optimization 3](#_Toc200471885)

[Efficient Dataset Design 3](#_Toc200471886)

[Smart Report Layout and Design 4](#_Toc200471887)

[Performance Tuning 4](#_Toc200471888)

# Power BI Desktop

## Data Preparation

* Clean and Transform Data: Utilize the Power Query Editor to clean and transform your data. Ensure data types are correct and remove any duplicates or irrelevant data.
* Model Relationships: Define relationships between tables to ensure accurate data representation and analysis.
* Optimize Data Size: Aggregate large datasets to improve performance and reduce load times.

## Report Design

* Consistent Formatting: Use consistent colors, fonts, and styles to create a cohesive and professional look.
* Intuitive Navigation: Design reports with a clear and intuitive layout to enhance user experience.
* Interactive Elements: Incorporate slicers, filters, and drill-through features to allow users to interact with data.

## Performance Optimization

* Simplify complex calculations and queries to improve performance.
* Create measures instead of calculated columns to optimize data models.
* Avoid distinct counts on fields with high cardinality – i.e. millions of distinct values.
* Take steps to avoid fields with unnecessary precision and high cardinality. For example, you could split highly unique datetime values into separate columns – e.g. month, year, date, etc. Or, where possible, use rounding on high-precision fields to decrease cardinality – (e.g. 13.29889 -> 13.3).
* Be wary of DAX functions, which need to test every row in a table – e.g. RANKX – in the worst case, these functions can exponentially increase run-time and memory requirements given linear increases in table size.
* Push calculated columns and measures to the source where possible – the closer they are to the source, the higher the likelihood of performance.

# Power BI Service

## Workspace Management

* Organize Workspaces: Create and organize workspaces based on teams, projects, or departments to streamline access and management.
* Access Control: Define roles and permissions to ensure data security and appropriate access levels.
* Collaboration: Use shared datasets and reports to foster collaboration and consistency across teams.

## Report Distribution

* Subscriptions: Create report subscriptions to automate the distribution of insights to stakeholders.
* Provide Clear Naming Conventions: Use consistent and descriptive names for your workspaces, reports, dashboards, and datasets to make them easily discoverable.
* Document Your Work: Add descriptions to your workspaces, reports, and datasets. Maintain a data dictionary to help users understand the data sources, measures, and calculations used.

## Security and Governance

* Implement Row-Level Security (RLS): Use RLS to restrict data access for different users based on their roles. This ensures that users only see the data they are authorized to view.
* Utilize Data Classification and Sensitivity Labels: Classify your data based on its sensitivity (e.g., Public, Internal, Confidential). This helps in applying appropriate security measures and preventing data leakage.

# Power BI Report Builder

## Data Source Optimization

* Choose the Right Data Source: Whenever possible, use Power BI datasets as your source. They offer better performance and governance. For other sources, prefer cloud-based options like Azure SQL Database to minimize latency.
* Minimize Data Source Connections: Consolidate your data into fewer sources. Each additional connection adds overhead.
* Use Pre-Aggregated Data: Perform complex calculations and aggregations at the data source level (e.g., in your SQL queries or Power BI data model) rather than in the report itself.

## Efficient Dataset Design

* Write Efficient Queries: Optimize your SQL or DAX queries to retrieve only the necessary data. Use WHERE clauses to filter data early. Avoid SELECT \*.
* Limit the Data Returned: Only bring in the columns and rows you absolutely need for the report. More data means slower processing.
* Use Parameters for Filtering: Implement parameters to allow users to filter the data. This reduces the initial data load and makes reports more interactive. Avoid overly complex or cascading parameters that can slow down the report.
* Shared Datasets for Consistency: For commonly used data, create and certify shared datasets in the Power BI service. This promotes a single source of truth and saves development time.

## Smart Report Layout and Design

* Plan Your Layout: Sketch your report layout before you start building. A well-structured report is easier to understand and maintain.
* Use a Grid System: Align elements on a grid to create a professional and organized look.
* Keep It Simple: Avoid clutter. Each report should have a clear purpose. Use white space effectively to improve readability.
* Consistent Branding: Use report themes and templates to maintain consistent branding with logos, fonts, and colors across all your reports.
* Optimize for Export: Be mindful of how the report will look when exported to different formats like PDF or Excel. Use appropriate page size settings and test your exports.

## Performance Tuning

* Minimize Visuals and Complexity: The more elements on a page (tables, charts, images), the longer it takes to render. Use visuals judiciously.
* Avoid Overlapping Items: Overlapping report items can cause rendering issues and slow down performance.
* Efficient Image Use: Use compressed image formats (like JPG or PNG) and avoid embedding large, high-resolution images.