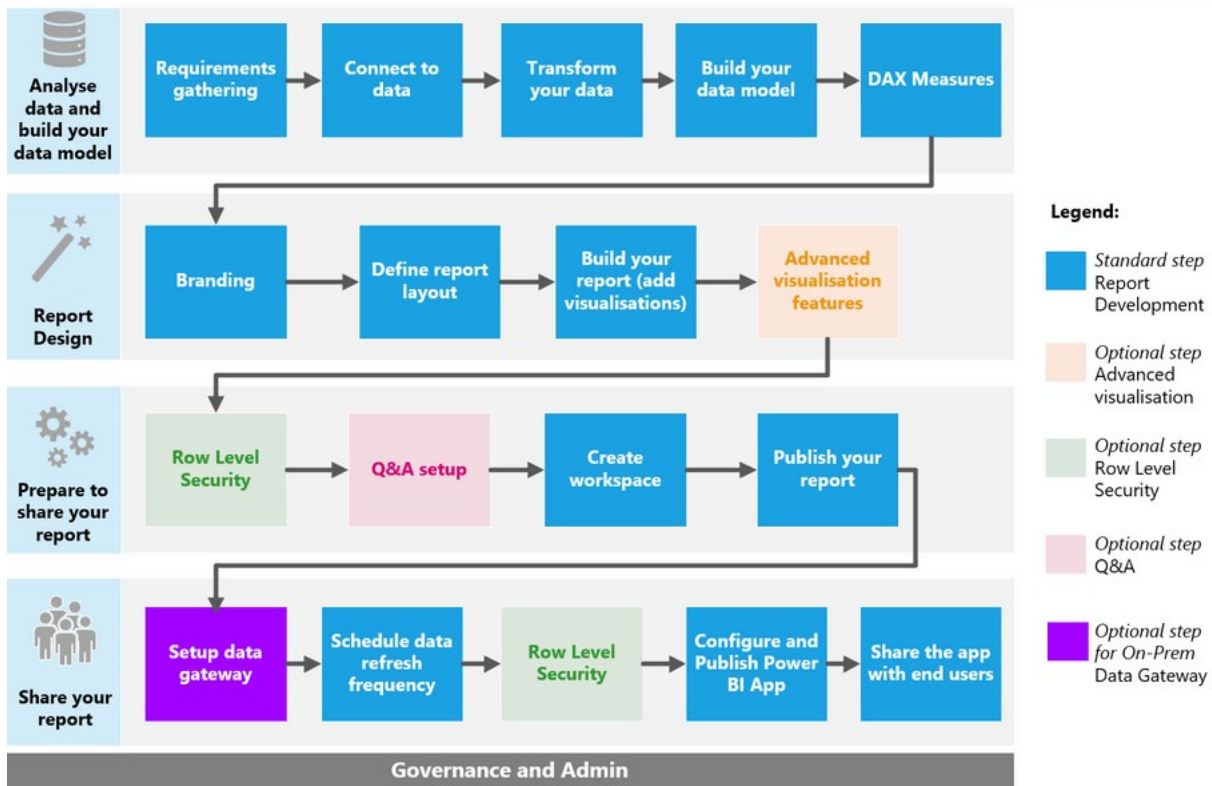


Power BI Report Development Steps

Power BI Report Development 101

> For Cloud & On Premises data sources



Source: [Link](#)

STEP 1: ANALYZE DATA AND BUILD YOUR DATA MODEL

1. REQUIREMENTS GATHERING

- ☐ Understand the problem - Know where you're going
- ☐ Know your audience – Audience is key
- ☐ Know your data - What data do I need/have?
- ☐ Define success criteria - How do stakeholders and end users measure success?

2. CONNECT TO DATA

- ☐ Choosing the data source connection method is the first step in establishing a Power BI data model.
- ☐ Investing enough time here so as to optimize the data model performance
- ☐ Here is a simple matrix for various data connection method

Import	Direct Query	Live Connection
Dataset is less than 1GB (compressed)	Dataset is large (>1GB)	Analysis Services is already in use
Data only needs to be refreshed a few times a day	Data to be updated in near real time	Analysis Services version is 2016+ Tabular instances (you might have performance issues with older AS instances and Multidimensional instances)
Data model is complex (lots of data transformations, complex measures)	Data is stored in a source that supports Direct Query	Analysis Services instance is on the cloud (on prem can cause latency issues)
Best performance is expected	No plan to do complex data transformations (combining tables etc.)	No plan to do transformations over the data or complex measures
	No plan to build measures with complex logic	No plan to use Q&A

2.a IMPORT – WHAT TO DO WHEN YOUR DATA MODEL IS TOO LARGE (>1GB)

- ☐ Consider using dataflows + shared datasets
- ☐ Consider using composite models
- ☐ Use parameters to filter large tables and reduce the amount of data loaded
- ☐ Consider using incremental refresh
- ☐ For very large datasets, consider creating a subset of the model for the most common reporting scenarios (consider using composite models too)
- ☐ Import only necessary fields and tables
- ☐ Sync the power BI refresh with the data source refreshing frequency

2.b HOW CAN I ENHANCE PERFORMANCE IN DATA CONNECTIVITY?

- ☐ Import needed fields and tables
- ☐ Avoid departmental/personal data sources as possible
- ☐ Minimize Excel, csv, and text files wherever possible, try to utilize data from a database
- ☐ Relational databases are quicker, cleaner, and simpler to use
- ☐ Connect to natively queryable and filterable data sources (e.g., SQL Server)
- ☐ Delegate as much processing to the data source as possible
- ☐ Disable privacy settings or set both sources to Organizational
- ☐ During development, frequently test Power BI data refresh.

3.a TRANSFORM DATA (GENERAL)

- ☐ Leverage query folding - Place filters steps before row-holding steps - operations pushed down to source are often much faster
- ☐ Filter out unnecessary columns and rows - Start your report with the bare minimum of data. This involves deleting superfluous columns and columns with computed values.
- ☐ Reduce usage of long-length columns with high precision and cardinality - Long text, decimal columns, Date/Time are examples. Unique column values reduce compression efficiency. Reduce decimal places, separate date and time.
- ☐ Turn off Auto Date Time - Auto Date Time produces several internal date tables for smaller models.
- ☐ Handle dirty data, incorrect data and errors
- ☐ Avoid transformations that scan whole tables like joins etc. - If not folded, the complete table must be loaded into memory before proceeding; use DAX measures instead.
- ☐ Group by and summarize - Load pre-summarized data
- ☐ During development, frequently test Power BI data refresh.

3.a TRANSFORM DATA (DIRECT QUERY MODE)

- ☐ Avoid complex Power Query queries - Eliminating Power Query transforms improve model design. Each query corresponds to a single database table or view.
- ☐ Examine the use of calculated columns and data type changes - Direct Query models allow data type conversions and computations. Materializing transformation results in the relational database source may improve performance.
- ☐ Do not use Power Query relative date filtering -
- ☐ Limit parallel queries - Direct Query's maximum connections per data source can be configured. It controls concurrent data source requests.

4.a BUILD YOUR DATA MODEL (RELATIONSHIP TUNING)

- ☐ Ensure tables have relationships
- ☐ Validate and Use Inactive Relationships Purposefully
- ☐ Avoid bi-directional relationships against high-cardinality columns
- ☐ Avoid excessive bi-directional or many-to-many relationships

- ☐ Many-to-many relationships should be single direction
- ☐ Aim for star schemas, avoid snowflake schemas
- ☐ Ensure tables have relationships
- ☐ Validate and Use Inactive Relationships Purposefully

4.b BUILD YOUR DATA MODEL (MODELLING TUNING)

- ☐ Hide all fields not used directly by users
- ☐ Model should have a date table
- ☐ Reduce number of calculated columns
- ☐ Reduce usage of calculated tables
- ☐ Optimize column data types and precision
- ☐ Turn off column hierarchies (IsAvailableInMDX column property)

4.c BUILD YOUR DATA MODEL (DIRECT QUERY MODE)

- ☐ Avoid relationships on calculated columns - Calculation expressions are integrated in source queries. It is inefficient and inhibits indexing
- ☐ Set relationships to enforce integrity - Assume Referential Integrity influences whether Power BI generates source queries with an inner join
- ☐ Examine the use of calculated columns and data type changes - Materializing transformation results in the relational database source improves performance

5.a DAX MEASURES (PERFORMANCE TUNING)

- ☐ Use DAX variables if possible
- ☐ Try to avoid DAX iterator functions (e.g. sumx, averagex...)
- ☐ Consider using the divide() function
- ☐ Use calculated measures rather than calculated columns when possible
- ☐ Use DAX variables if possible
- ☐ Try to avoid DAX iterator functions (e.g. sumx, averagex...)

5.b DAX MEASURES (USABILITY TUNING)

- ☐ Store all your measures in a separate table

- ☐ Name your measures in a meaningful way - avoid ambiguity in names of columns and measures
- ☐ Format all currency & decimal measures to defined standard (e.g. 2 decimal, thousand separator)
- ☐ Use Explicit Measures, not Implicit Measures. Simply put, implicit measures are measures that are automatically assigned an aggregation such as a Sum or a Count by Power BI

STEP 2: REPORT DESIGN

1. DESIGN TUNING

- ☐ Use templates (.PBIT files) to speed up and standardize report development instead of starting with an empty .PBIX
- ☐ Have a focus on usability of the report for end users
- ☐ Address accessibility early in the design phase
- ☐ Carefully review the data displayed to assess if it can be easily misinterpreted

2. HOW TO PRESENT INFORMATION

- ☐ Use careful placement and a clear hierarchy for displaying information on the page
- ☐ Have a specific purpose for each page, and for each chart on each page
- ☐ Be consistent with placement of items which appear on multiple pages
- ☐ Use a layout that focuses on telling the story you want to tell
- ☐ Avoid clutter on the page to allow report consumers to focus on what is important
- ☐ Use the right visualization for the data (Pie charts, donut charts, gauges, and other circular chart types aren't a data visualization best practice)
- ☐ Use a color palette that follows visualization best practices and the company branding guidelines
- ☐ Use clear titles, labels, and descriptions
- ☐ Use bookmarks to create 'help' information to guide users

3. PERFORMANCE TUNING

- ☐ Avoid dense report pages - Reduce the data and images on a page
- ☐ Avoid "data dump"-style reports - Avoid reports with hundreds of columns and thousands of rows. If the user wants to export data to Excel, construct the report directly in Excel with a live connection to the dataset
- ☐ Use slicers and filters in your reports - To let users choose which data to show. Set slicers or filters at report publishing to ensure acceptable first rendering

- ☐ Consider adding an "Apply filters" button to your report so graphics don't refresh immediately when you change a filter (very handy for Direct Query!)
- ☐ Bookmarks, drill-through pages, and tooltips may minimize amount of data in each page
- ☐ Choose visuals carefully - Focus on its ability to convey information, not just the aesthetic appeal
- ☐ Only use custom visuals from sources that you trust – preferably certified. Test your report's custom visualizations for quick loading
- ☐ Change the interactions between visuals - Unnecessary visual filtering might slow down your report (especially for Direct Query!)

STEP 3: PREPARE TO SHARE REPORT

1. ROW LEVEL SECURITY

- ☐ Enable Row Level Security - With RLS, Power BI only imports data the user is authorized to view
- ☐ RLS filters every DAX query, affecting query performance. So, efficient RLS comes down to good model design
- ☐ Dimension-type tables are frequently better for RLS filters than fact-type tables
- ☐ Ensure RLS filters propagate to other model tables with well-designed relationships
- ☐ Don't use LOOKUPVALUE DAX when model relationships will do
- ☐ Avoid complex RLS rules
- ☐ If RLS filters are imposed on Direct Query tables that are related, optimize the source database

2. Q&A

- ☐ The performance of the queries done through Q&A will depend directly on how well you've optimized your data model earlier
- ☐ Tell Q&A which visual to use
- ☐ Refer here for more best practices related to [Q&A](#)

STEP 4: SHARE THE REPORT

1. SETUP DATA GATEWAY

- ☐ Use Enterprise data gateway instead of Personal Gateway - Personal Gateway takes data and imports it into Power BI.
- ☐ When dealing with huge databases, Enterprise Gateway (on-premises data gateway) imports nothing

- ☐ Use different Power BI gateways for Scheduled Refresh and Direct Query - When Scheduled Data Refresh and Direct Query utilize the same gateway, Direct Query speed slows

2. SCHEDULE DATA REFRESH

- ☐ If possible, enable incremental refresh - Incremental refresh simply refreshes modified data. Fewer refreshes minimize memory and other resource use
- ☐ Carefully analyze the frequency of your data refresh - Users may desire a dataset updated 48x a day, but if your data source only refreshes once a day, it won't help your solution
- ☐ Think about the refresh time for your dataset - Schedule your refresh at a "quieter" period to avoid concurrent report refreshes

3. OTHER BEST PRACTICES

- ☐ Add users to Office or Azure Active Directory groups and then use groups to share report content
- ☐ Use groups to manage role membership in role-based or row-level security
- ☐ Consider promote/certify your dataset so that it can be reused by others in the organization. This can potentially avoid a lot of dataset duplication
- ☐ Ensure the Power BI report and data source are in the same region - Tenant and data source in same location reduces network latency. Data transport and query execution are quicker