

Objective



Set of guidelines that address many aspects of new projects.

Recommendations are applicable in most use cases.

Living document with ongoing revisions with the goal to provide a comprehensive, best practices guide as the platform continues to mature and as experts continue to use it.

This Session is:

...about options and choices

...it is:

- NOT a 100-level introduction to Power BI
- NOT a technical “how-to” deep dive but we will discuss some technical topics
- Guidelines for experienced Business Intelligence & Power BI practitioners
- Good knowledge for less-experienced Power BI developers



...our theme:





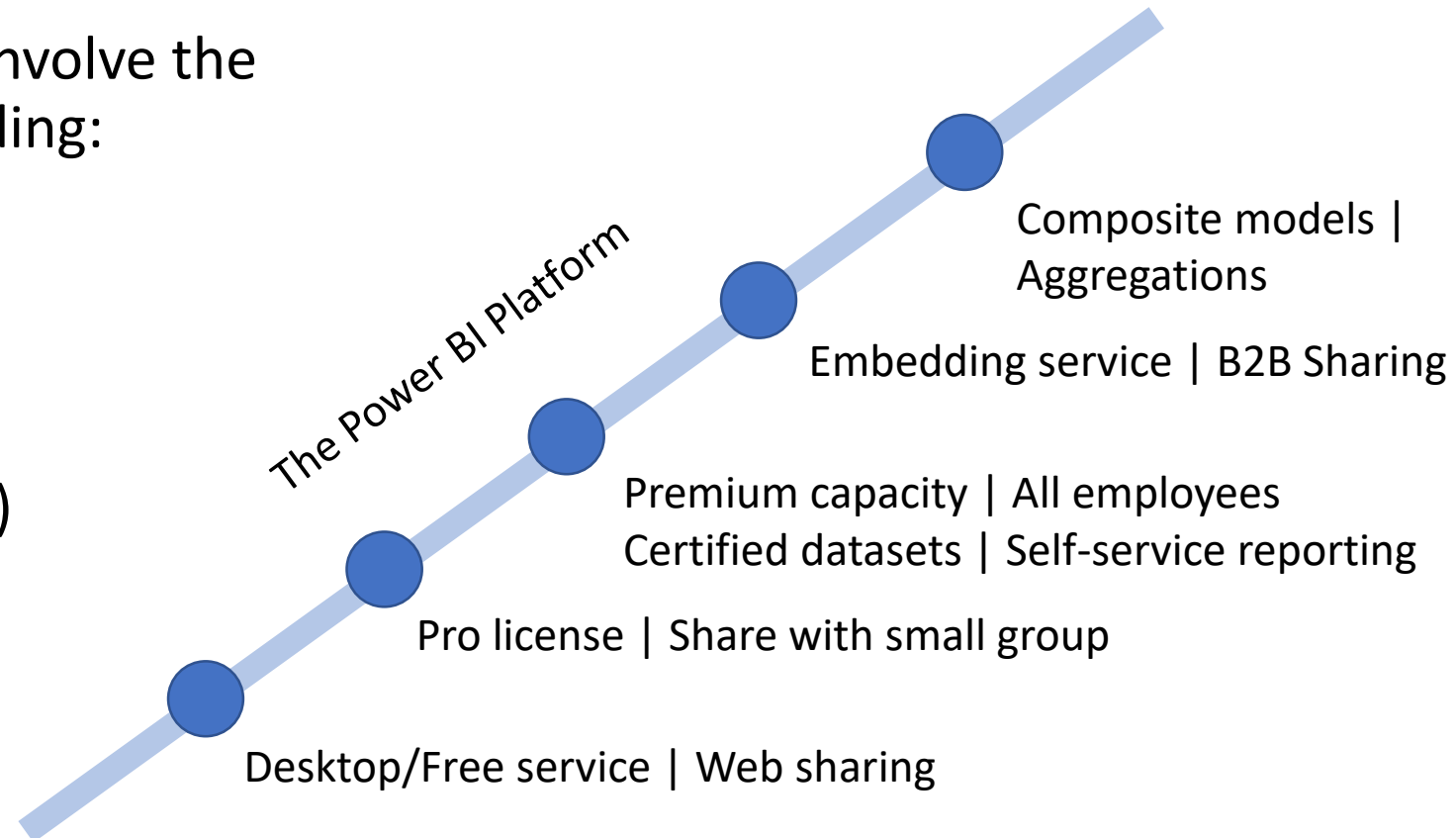
Power BI Offers Many Options

- Trade-offs between “quick” and “right”
- Different project types
- Governed & certified data
- Flexibility vs control
- Data volume & latency
- Self-service data modeling
- Self-service reporting

Solution Architecture

All Business Intelligence projects involve the same essential components including:

- Source Queries
- Data transformation steps
- Semantic data model
- Calculations (typically measures)
- Data visualizations



Checklist: Identify Your Audience

Categorize the solution by identifying the author & user roles related to the project

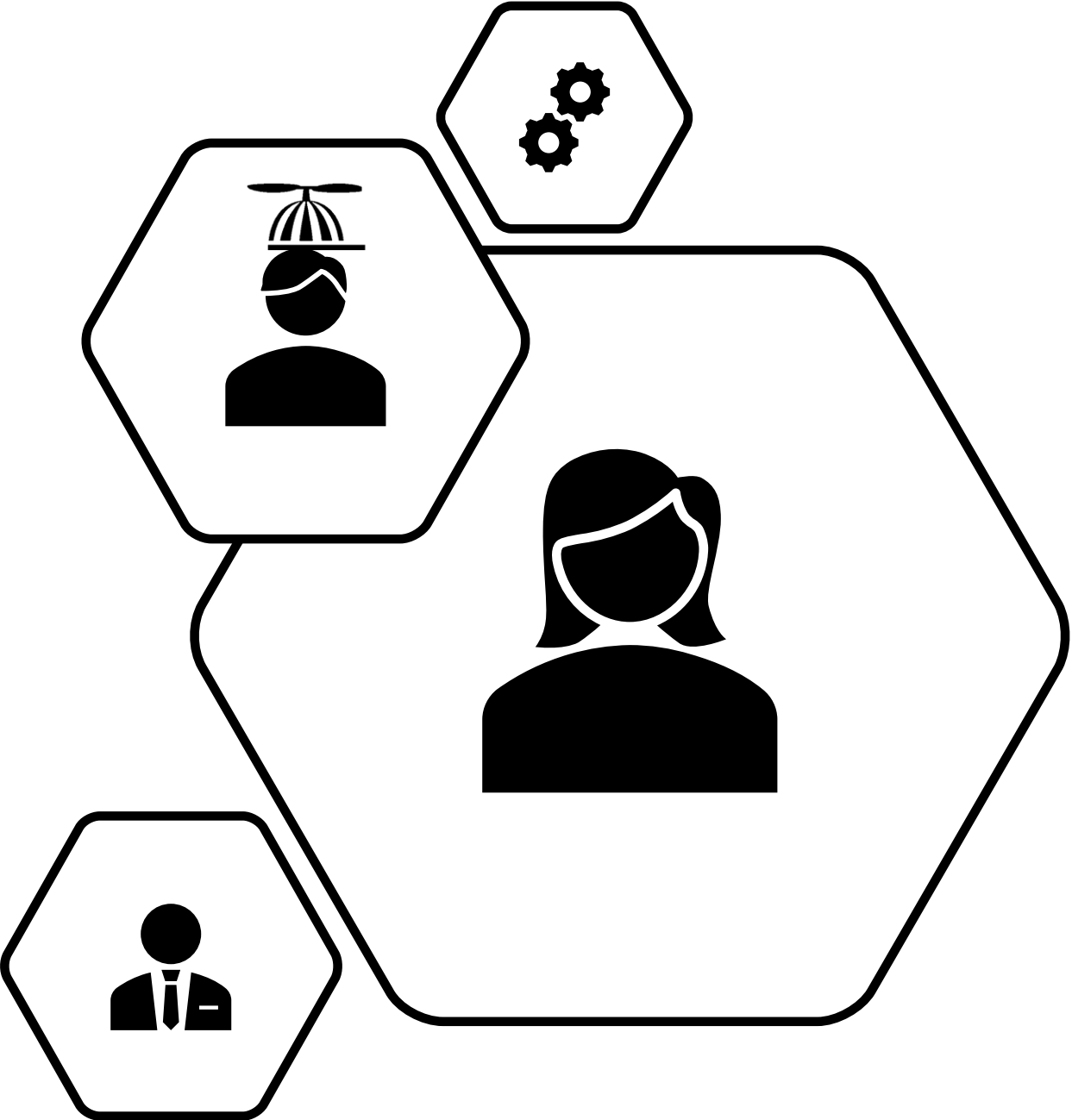
Author roles:

- Author role: Business Data Analyst
- Author role: Skilled Data Modeler, Analyst, Data Scientist
- Author role: IT BI Developer

%

User roles:

- Users' role: Report/Dashboard Consumer
- Users' role: Self-service Report Author
- Users' role: Advanced Data Analyst




Formal vs Informal Project



Formal Project

Decision Criteria:

- “Git ‘er Done” vs “Do It Right”
 - Project longevity/extensibility goals
 - Ownership: Business or IT?
 - Team development?
 - Version control, ALM/DevOps ?
- 

Checklist: Solution Type for the project

Identify the Project Type & related Solution Architecture:

Formal projects

are scoped, funded, staffed and executed with the collaboration of a business champion and stakeholders; and IT Business Intelligence developers and data managers. These projects promote business and IT-governed datasets and certified reports.

Informal projects

are executed by business users and are considered ad hoc in nature. Datasets are generally not IT governed, and reports are typically not certified.

Hybrid projects

can be anything in-between. They might be a user-authored report using published, certified dataset used for self-service reporting. Informal, self-service datasets can be migrated to governed datasets in collaborative IT/business projects.

Combined vs Separate Dataset & Report Files

Decision Criteria:

- Simplicity vs Separation of development
- Self-service reporting vs self-service data modeling
- Certified datasets?
- Certified reports?



Join The Separatist
Movement

Checklist: Dataset & Report Architecture

Choose dataset architecture:

- Single PBIX file**
For small group, departmental project authored by one developer for a limited group of users
- Separate dataset and report PBIX**
Design & deploy a separate dataset PBIX file – from report file(s) – when the dataset should be branded as a Certified dataset.
For formal projects with more than one dataset & report developer, to coordinate work
- SSAS/AAS as a data modeling option**
when those databases exist or where IT operations insist on managing development and maintenance through integrated source control (e.g. Visual Studio Team Services & Azure DevOps)

A red circle containing the text "Vern's Red Dot".

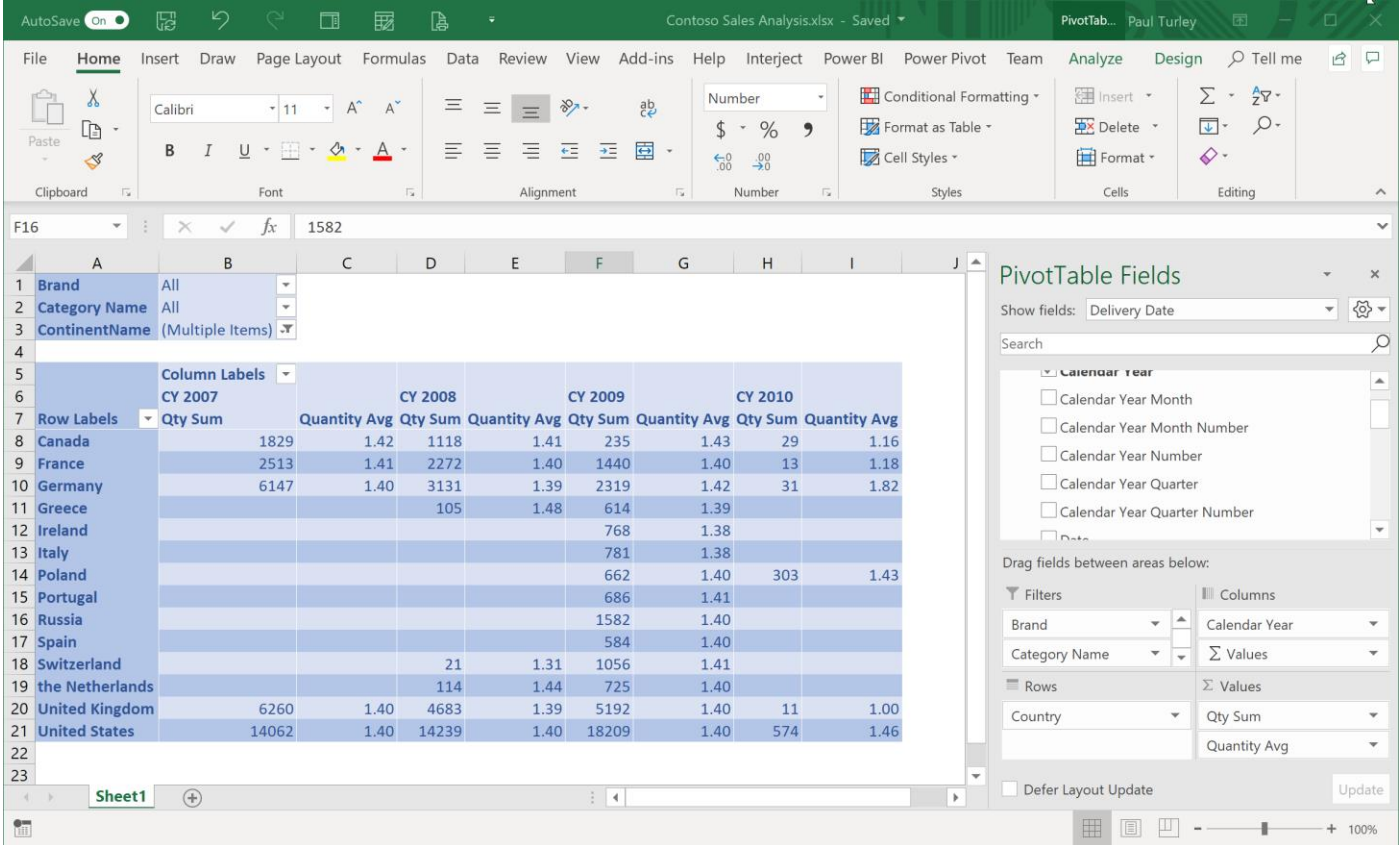
Vern's Red
Dot

Operational & Paginated Reports

- Power BI is not a replacement for paginated, operational reporting
- For static, multi-page, printable reports; use SQL Server Reporting Services (SSRS) aka “Paginated Reports” instead of Power BI
- Paginated Reports/SSRS is integrated into the Power BI service with Premium capacity licensing and can be integrated with interactive Power BI reports and Power BI data datasets
- To a limited degree, some operational reports can be reproduced using Power BI reports and SSRS can be used, some a limited degree, to create interactive reports

If Users Need Excel, Give them Excel

- Teach analyst users how to use Excel with Power BI
- Don't "export", ... "connect"
- **"Analyze In Excel"** allows Excel to connect, live, to a published Power BI dataset
- Now available to Power BI Pro & Free Premium licensed users
- Now available to "free" licensed users in a Premium



The screenshot shows the Microsoft Excel interface with a PivotTable connected to a Power BI dataset. The PivotTable is located in the main workspace and displays sales data by country and year. The PivotTable Fields task pane on the right shows the data source and the fields used in the PivotTable.

Row Labels	Qty Sum	Quantity Avg	Qty Sum	Quantity Avg	Qty Sum	Quantity Avg	Qty Sum	Quantity Avg
Canada	1829	1.42	1118	1.41	235	1.43	29	1.16
France	2513	1.41	2272	1.40	1440	1.40	13	1.18
Germany	6147	1.40	3131	1.39	2319	1.42	31	1.82
Greece			105	1.48	614	1.39		
Ireland					768	1.38		
Italy					781	1.38		
Poland					662	1.40	303	1.43
Portugal					686	1.41		
Russia					1582	1.40		
Spain					584	1.40		
Switzerland			21	1.31	1056	1.41		
the Netherlands			114	1.44	725	1.40		
United Kingdom	6260	1.40	4683	1.39	5192	1.40	11	1.00
United States	14062	1.40	14239	1.40	18209	1.40	574	1.46

Checklist: Report Types

Dashboard & Scorecard style reporting

- Infographics
- KPIs & scorecards
- Segmented comparisons
- Time-series trends

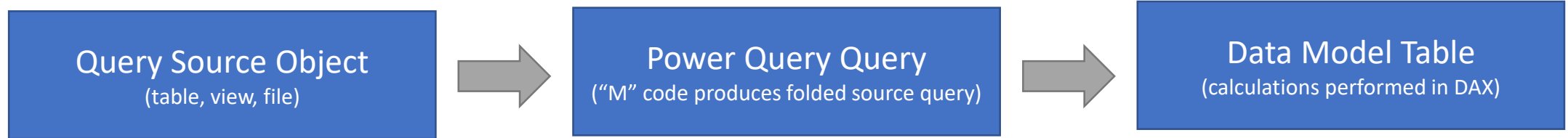
Statistical & Scientific analysis

- Deviations & percentiles
- Forecast trends & predictions
- Scatter plots
- Population analysis

Financial balances & worksheets

- Cost accounting & balance sheets
- General ledger
- Accounts receivable & payable
- Invoices
- Forms & lists

Checklist: Query Optimization

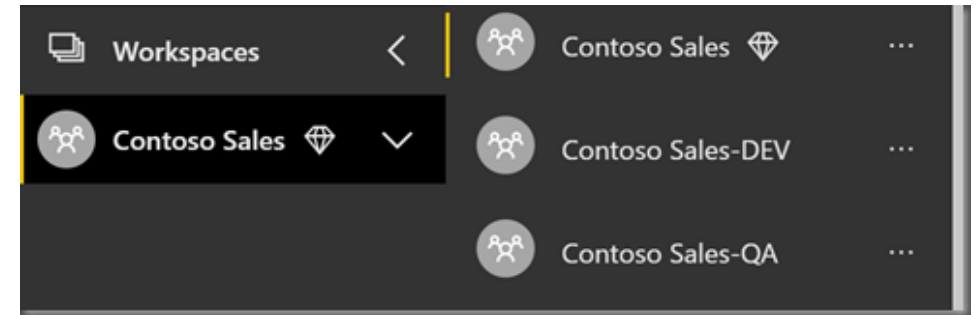


- Decide:** Perform column transformations in ETL, database view, Power Query or DAX?
- Decide:** How is process managed & governed? who maintains the query?
- Avoid using SQL statements in PQ queries. Use database views.
Views and tables support query folding. SQL statements generally do not.
- Remove unnecessary columns & filter rows early in the query
- Consolidate field renaming, removing fields and data type changes
- Add custom columns in Power Query instead of calculated columns in DAX, where possible
- Use friendly field names for all fields that won't be hidden in the data model
- Rename steps and add annotations in M script

Workspace and App Management

For a formal project, create the following workspaces:

- DEV Workspace** - Only development team members need Contributor access to this workspace. This workspace does not need to have Premium capacity; unless, developers need to unit test incremental refresh or other Premium features.
- QA Workspace** - All testers must have View access for testing and Contributor access for report authoring. Should be in Premium capacity to test incremental refresh.
- PROD Workspace** - Omit the “PROD” designation in the name. This workspace will be the name of the published app that users will see in their Apps, Home and Favorite pages so use a name that is simple and sensible. Must have Premium capacity to share the app with non-Pro licensed users.



Vern's Red
Dot

Deployment Options:

- PowerShell script may be used to publish datasets and reports, and to change dataset bindings. It is possible to either publish to a production workspace or to effectively move assets from one workspace to another. This approach is discussed briefly in the [Power BI Enterprise Deployment Guide](#). Other approaches are discussed here: [Power BI release management](#)
- OneDrive folder sync for development workspaces (later slide)

Promote Self-service Reporting

Non-governed Data

- Teach & support analyst users to use Power BI to acquire, mashup & model data
- “make mistakes, get messy”
 - Lilly Tomlin, Miss Frizzle
- Deploy to “user” designated workspaces
- User-authored solutions be used to prototype & pattern governed data models

Governed Data

- Separate datasets from reports
- Publish to a secured & managed workspace
- Promote & Certify datasets
- Use dataflows for standardized common data models
- Enable users to connect to published datasets & create their own reports

Service vs On-premises

- Power BI services offers the most complete feature set

Microsoft's priority & strategic direction is the cloud
To The Cloud
Features are slower to arrive on-prem, if at all

On-prem can support in-house storage compliance

- Cloud aversion prevents some customers from using the Power BI service

Storage Mode: Import vs DirectQuery

- Power BI is 99% Import data into model
- Uses columnar storage
- Highest performance



DirectQuery doesn't copy data into model

- Usually slower & restrictive
- Hybrid/composite models enable both modes... drill to real-time data

Implicit vs Explicit Measures

- “Implicit measure”: Numeric columns can be aggregated using default summarization
- Explicitly-defined measures provides a single, consistent method for all calculations
- Implicit measures are not supported by many client tools such as Excel
- Explicit measures provide more calculation flexibility

**Warning:
Explicit
Measures**



Implicit and Explicit Measure Guidelines

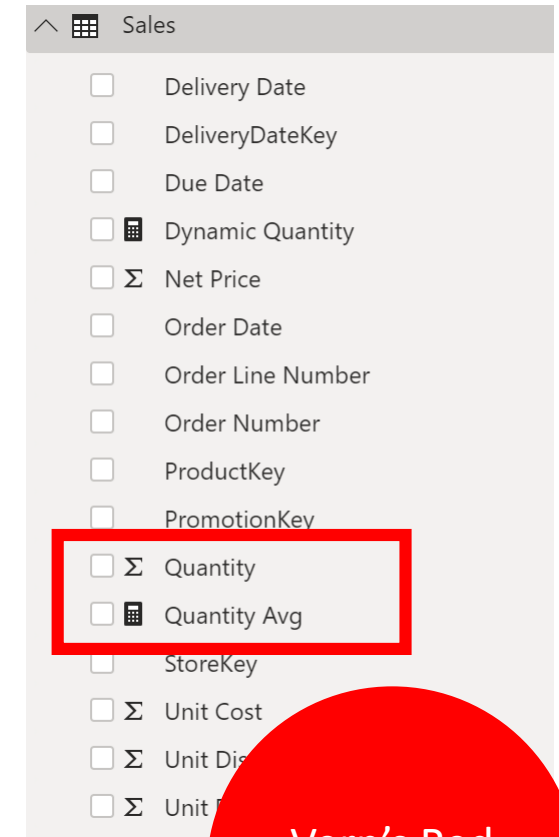
Implicit measure

= numeric field with default summarization

Explicit measure

= Defined using DAX expression

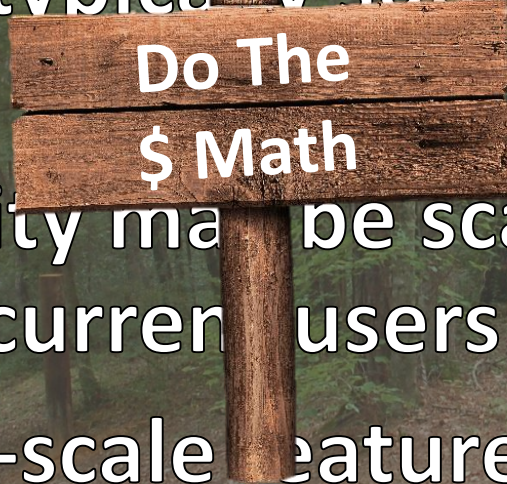
- Implicit measures are typically OK in informal projects
- Measures should be explicitly defined in formal data models
- Implicit measures don't work in some client tools



Vern's Red
Dot

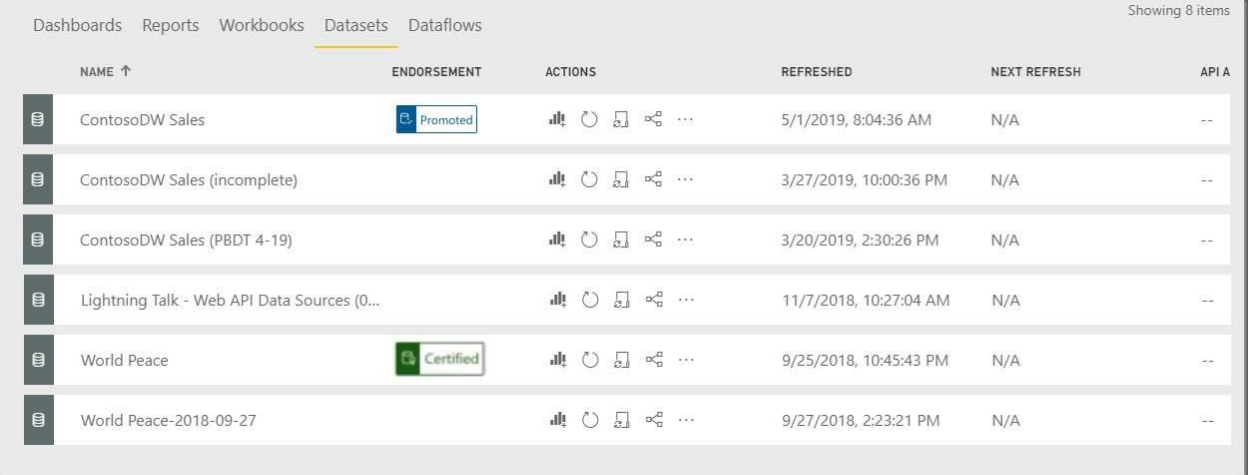
Premium vs Pro Licensing

- Shared capacity typically supports 8-10 concurrent users
- Dedicated capacity may be scaled to support hundreds of concurrent users
- Many enterprise-scale features require Premium licensing
- Monthly license break-even cost: 500 users





























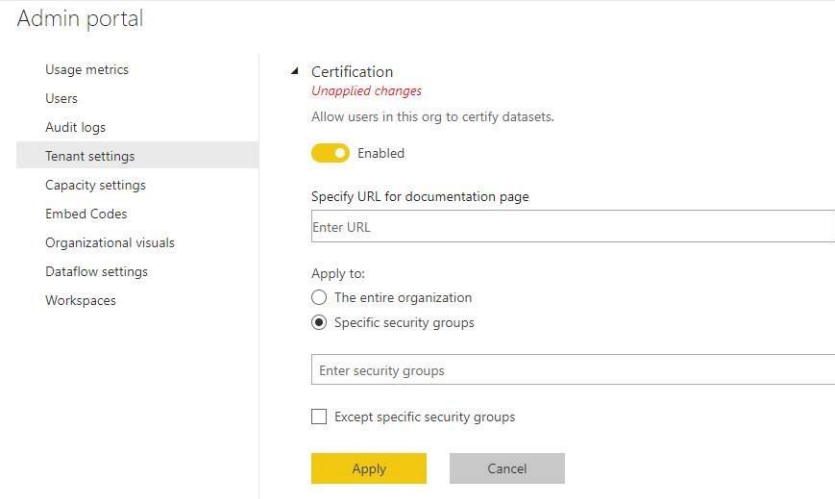
Certified & Shared Datasets

- Use Dataset endorsement & certification in the service
- Certification can be managed by security group
- Access to datasets can be restricted to certified datasets
- Organization defines certification policy & provides documentation



Showing 8 items

NAME ↑	ENDORSEMENT	ACTIONS	REFRESHED	NEXT REFRESH	API A
ContosoDW Sales		   	5/1/2019, 8:04:36 AM	N/A	--
ContosoDW Sales (incomplete)		   	3/27/2019, 10:00:36 PM	N/A	--
ContosoDW Sales (PBDT 4-19)		   	3/20/2019, 2:30:26 PM	N/A	--
Lightning Talk - Web API Data Sources (0...		   	11/7/2018, 10:27:04 AM	N/A	--
World Peace		   	9/25/2018, 10:45:43 PM	N/A	--
World Peace-2018-09-27		   	9/27/2018, 2:23:21 PM	N/A	--



Admin portal

- Usage metrics
- Users
- Audit logs
- Tenant settings
- Capacity settings
- Embed Codes
- Organizational visuals
- Dataflow settings
- Workspaces

Certification

Unapplied changes

Allow users in this org to certify datasets.

Enabled

Specify URL for documentation page

Enter URL

Apply to:

The entire organization

Specific security groups

Enter security groups

Except specific security groups

Apply Cancel

Enterprise Scale Options

In many ways, Power BI has now surpassed the capabilities of SQL Server Analysis Services. Microsoft are investing in the enterprise capabilities of the Power BI platform by enhancing Power BI Premium Capacity, adding Paginated Report and features to support massive scale specialized use cases. Consider the present and planned capabilities of the Power BI platform; before, choosing another data modeling tool such as SSAS.

Resources:

Power BI Licensing Plan Checklist

Capacity and platform:

Shared capacity service:

Assign user licenses

Dedicated capacity:

Are Premium features required?

Is dedicated capacity needed?

Is Premium more cost-effective than licensing all users?

On-premises server:

SQL Server Enterprise + SA, or:

Premium license

Assign user licenses and access:

Assign Pro licenses to all developers, admins and report author users

If Premium, use app deployment & assign Free licenses to all users

Assign membership and access to workspaces

Managing Power BI Desktop Files

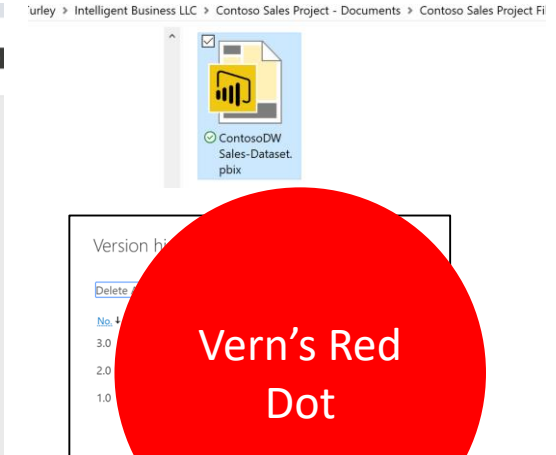
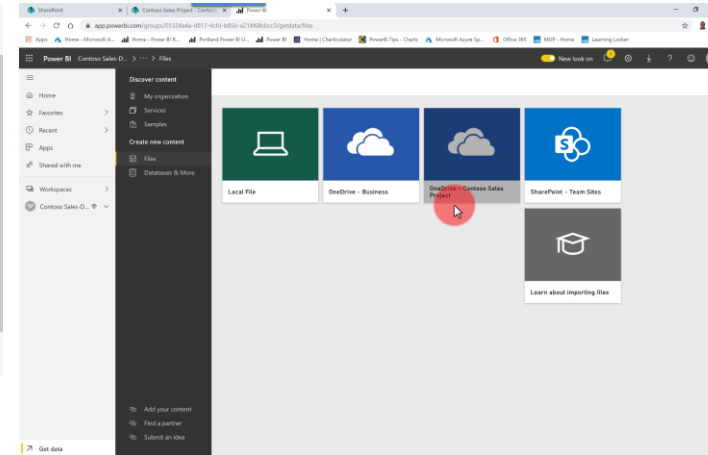
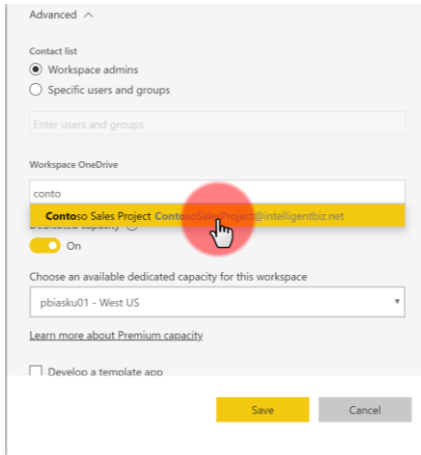
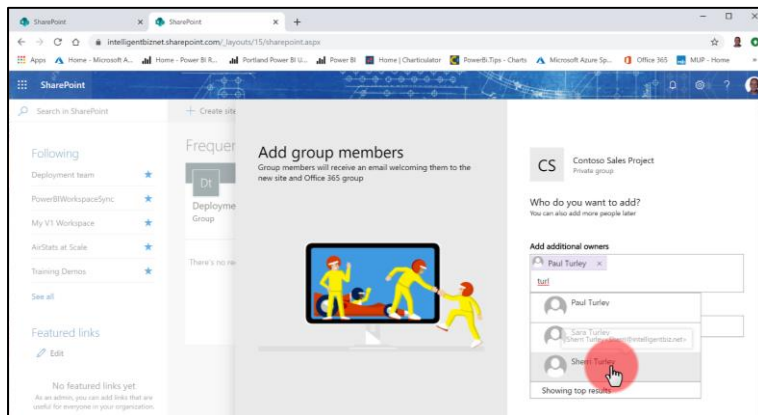
- **Store in a centrally managed network-assessable folder**
The storage folder should support automatic backup and recovery in the case of storage loss.
- **Report and dataset developers must open files from the Windows file system**
Files must either reside in or be synchronized with the Windows file system.
- Files containing imported data typically range in size from 100 to 600 MB. Any shared folder synchronization or disaster recovery system should be designed to effectively handle multiple files of this size.

Options:

- OneDrive For Business (shared by team, with folder synchronization).
- SharePoint or SharePoint Online (with folder synchronization).
- GitHub and/or VSTS with local repository & folder synchronization. If used, Git must be configured for large file storage (LFS) if PBIX files are to be stored in the repository.

Folder & Workspace Synchronization

1. Create team site in Office 365, add developers
2. Create development folder in team site & synchronize with desktop
3. Create workspace(s) & set OneDrive group
4. Add PBIX files to workspace using Get Data from team OneDrive folder
5. Edit & save PBIX files. Deployment is automatic.



File & Workspace Management Checklist

Create storage locations and folder structure for Development file management:

- Development file storage
- Team member collaboration environment & processes
- Folder synchronization
- Define File naming standards
- Decide on dataset and report names

Define the Version Control & Lifecycle Management:

- Postfix files with 3-part version number
- Remove version number from published files in QA and PROD
- Create Version History table in Power Query
- Increment version numbers in data model
- Backup PBIT files for archive
- Create measures: Last Refresh Date/Time
- Create measure: Current Version
- Add data model info page to report

Decide on Workspace and App Management, workspace & app name, etc.:

- Create PROD workspace (omit PRD from name), assign dedicated capacity if available.
- Create QA workspace (post-fix name with QA), assign dedicated capacity
- (*optionally*) Create DEV workspace (postfix name with DEV), dedicated capacity not required (or combine with QA workspace).

Model Design Guidelines

- Dimensional design concepts haven't changed in 20 years & are as true as ever
- Dimensional modeling “rules” should be followed but can be relaxed for Power BI in certain cases, such as:
 - Leaving some dimensional attributes in fact tables
 - Use natural keys rather than generating surrogate keys
- The art of dimensional modeling ranges from simple to complex. Start with the basics.
- Flattened “spreadsheet” models are OK for small, informal projects but have significant limitations
- As models grow in size & complexity, data quality challenges will surface that can be solved by implementing proper governance controls

Model Design Checklist

- Model for the user experience, not for developers**
- Build star schemas**

Wherever possible, reshape data into fact a dimension tables with single key, one-to-many relationships from dimensions to fact.
- Enforce dimension key uniqueness**

Just because a key value “should” be unique, there is no guarantee that it will be unless enforced at the data source. Perform grouping and duplicate reduction in the data source views or Power Query queries to guarantee uniqueness. Duplicate record count checks and other mechanisms can be applied to audit source data for integrity but do not allow the data model to violate these rules.
- Avoid bi-directional filters & unnecessary bridging tables**

These data modelling patterns adversely affect performance.
- Consider using DAX measures** rather than complex & inefficient relationships
- Create custom columns in Power Query**

Rather than DAX calculated columns wherever possible for row-level derived columns. This maintains a consistent design pattern for maintainability.
- Annotate code**

Use in-line comments and annotations in all code including SQL, M and DAX; to explain calculation logic and provide author and revision information.
- Remove all unused fields** – if in doubt, take it out
- Hide all fields not used directly by users**

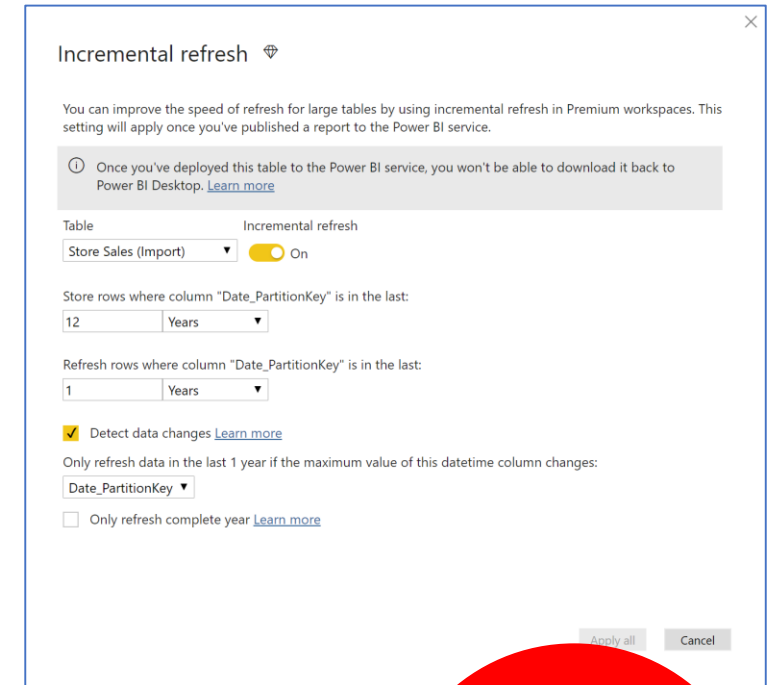
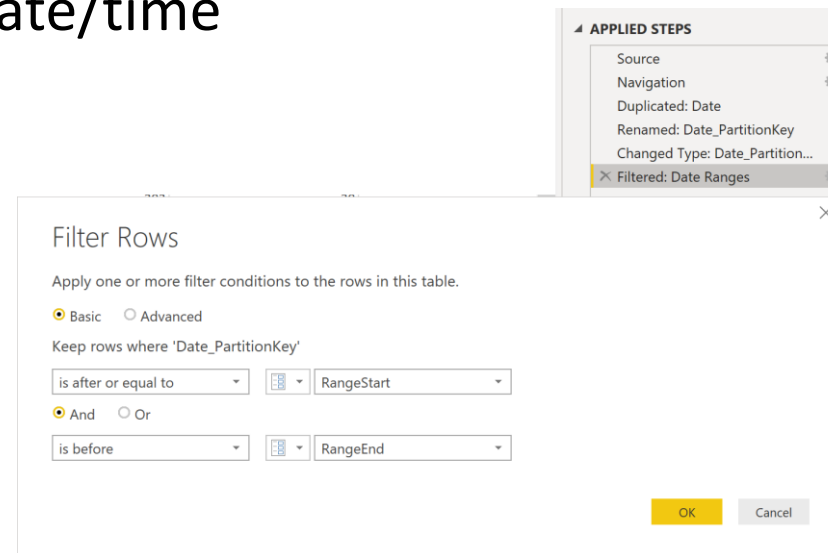
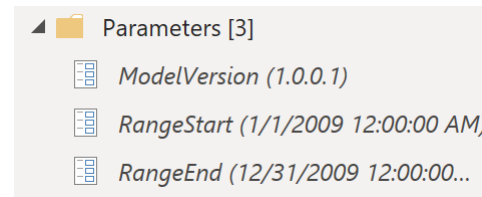
primary and foreign key columns, numeric columns used to create measures, and columns used to specify the sort order of other fields.
- Use friendly field names**

Rename all visible columns (in Power Query) to short but user-friendly names with mixed case and spaces.
- Set to Do Not Summarize**

Any non-hidden numeric columns that are not intended to roll-up or summarize values. Columns set to summarize are indicated with a Sigma icon.

Managing Dataset Size with Parameters

- Use parameters whether implementing incremental refresh or not
- **RangeStart & RangeEnd** parameters must be date/time type
- Apply range filter on date/time column in Power Query



*Incremental Refresh is a Premium feature

Vern's Red Dot

Training & Usability Support Plan Checklist

Training Guidelines:

For general best practice training, don't reinvent the wheel. There are many good books and training programs available that took several years to develop. Best practices continue to evolve quickly.

Promote and teach "your way" within your organization. Don't just turn users loose with the tools and expect them to make good decisions.

Training and Usability Support:

Develop & Document Support & training plan for users:

- Usability training for read-only report/app users
- Self-service reporting for Novice Report Authors & Data Analysts
- Training for advanced analysts & developers

Choose or develop training platform & curriculum:

- Third-party training courses for developer orientation
- Use internal training & support to direct users to your solution
- Teach users to use governed datasets, standard or self-service reports

Master Project Preparation Checklist

Solution Audience:

- Categorize the solution by identifying the author & user roles related to the project:
- Author role: Business Data Analyst
- Author role: Skilled Data Modeler, Analyst, Data Scientist
- Author role: IT BI Developer
- Users' role: Report/Dashboard Consumer
- Users' role: Self-service Report Author
- Users' role: Advanced Data Analyst

Training and Usability Support:

- Develop & Document Support & training plan for users:
 - Usability training for read-only report/app users
 - Self-service reporting for Novice Report Authors & Data Analysts

Solution Type & Architecture:

- Identify the Solution Type for the project. This will guide other project management designs:
- Design single PBIX file for small group, departmental project authored by one developer for a limited group of users
- Design & deploy a separate dataset PBIX file – from report file(s) – when the dataset should be branded as a Certified dataset
- Design separate dataset and report PBIX files for formal projects with more than one dataset & report developer, to coordinate work
- Use SSAS/AAS as a data modeling option when those databases exist or where IT operations

insist on management development and maintenance through integrated source control (e.g. Visual Studio Team Services & Azure DevOps)

Identify the Project Type & related Solution Architecture:

- Project type: Formal project
- Project type: Informal project
- Project type: Hybrid project
- Architectural approach: Single PBIX
- Architectural approach: Separate dataset and report PBIX
- Architectural approach: Report PBIX connected to SSAS or AAS

Understand DirectQuery model trade-offs and special use cases. Avoid if possible.

Define your Release Management, DevOps & Automation strategy (if any – Might be OK to deploy files manually, to automate or not to automate)

File & Workspace Management:

- Create storage locations and folder structure for Development file management:
- Development file storage
- Team member collaboration environment & processes
- Folder synchronization
- Define File naming standards
- Decide on dataset and report names
- Define the Version Control & Lifecycle Management:
 - Postfix files with 3-part version number

- Remove version number from published files in QA and PROD
- Create Version History table in Power Query
- Increment version numbers in data model
- Backup PBIX files for archive
- Create measures: Last Refresh Date/Time
- Create measure: Current Version
- Add data model info page to report

Decide on Workspace and App Management, workspace & app name, etc.:

- Create PROD workspace (omit PRD from name), assign dedicated capacity if available.
- Create QA workspace (post-fix name with QA), assign dedicated capacity
- (optionally) Create DEV workspace (postfix name with DEV), dedicated capacity not required (or combine with QA workspace).

Assign licenses and access:

- Assign Pro licenses to all developers, admins and report author users (QA?)
- Assign Free licenses to all users if Premium/app deployment will be used
- Assign membership and access to workspaces

Query Design:

- Create fact date range filter parameters: RangeStart & RangeEnd to reduce volume in PBIX file under 400 MB.
- Filter large fact tables with range filters, consider incremental refresh policies if slow and/or over 800 MB compressed.
- Design source queries (T-SQL?) to reshape

source data into conformed dimension & fact tables

- Create views in database for each dimension and fact
- Enforce key uniqueness to remove all duplicate keys from all dimension tables
- Query Date dim/lookup table at source if it exists
- If not available, generate Date dim/lookup table in Power Query

Data modeling:

- Build star schemas
- Enforce dimension key uniqueness
- Avoid bi-directional filters & unnecessary bridging tables
- Consider using DAX measures rather than complex & inefficient relationships
- Create custom columns in Power Query
- Annotate code
- Hide all fields not used directly by users
- Use friendly field names
- Set to Do Not Summarize