Power BI Desktop

Power BI Workflow Overview

- Backend mostly to transform and shape data
  - Power Query - ETL
- Front End
  - Model View – configure data models
  - Data View – any calculated columns, etc.. DAX,
  - Report View
- Publish and Share via Power BI Service

Storage and Connection Modes: Import

- Dataset < 1 GB
- Source does not change frequently
- No restrictions on power query, data modeling, and DAX functions

Storage and Connection Modes: Direct Query

- Dataset too large to be stored in-memory
- Source data changes frequently
- Company policy / restrictions on data usage, importing and access restrictions

Storage and Connection Modes: Composite Model

- Hybrid of import and direct query, based on individual table(s)
- So we combine DirectQuery model with imported data
- Create a single model from 2 or more DirectQuery modes

Storage and Connection Modes: Live Connection

- Create one dataset that serves as a central source of truth
- Different Roles can work with the same dataset, i.e. analyst can create different reports, another team can build models, facilitates multi-developer teams
- More integrated with the Microsoft BI platform

Power Query editor recaps

- Connect to data sources:
  - Files, folders
  - Database
  - Web, etc..
- Transform and reshape data
  - Datatypes, keep/remove rows, add/remove columns
  - Data Profiling and QA, column value distribution, basic stats, etc..
  - Text, Numeric and date data type specific transformation functions
  - *Explore Calendar and Datetime functions in detail*
  - Indexing, conditional / calculated columns

Power Query editor recaps

- Transform and reshape data
  - Grouping and Aggregating
  - Pivot (go from narrow column to wide column table)
  - UnPivot (go from wide column table to narrow)
  - Transpose is pivot without sense, i.e. it changes the shape without aggregating and simply moves all rows to columns kinda thing
  - Merging (Inner, Left and Right joins)
  - Appending (concatenating) – folder data source settings are especially helpful for this operation

Power Query editor recaps

- Transform and reshape data
  - Data Source Settings:
    - Use folders where possible, and make it rich by adding any filters to get data refreshed when folder gets updated
    - Use parameters to reference folders, files, and database connection strings to connect to development, test and production regions and/or different schemas
  - Design considerations to turn on/off report refresh settings
  - Import entire excel models
- Use folders to help manage/maintain category specific queries

Power Query refresher tips and stuff

- If we do not want to load a dataset or a table and data into the power bi model, we can disable enable load option in the power query editor (right click on each data set and uncheck option to load)
- In power query while profiling a column, right clicking and following thru gives different error details to address or use auto clean up/remove – and basically each of those is recorded as a step as well as see the generated M code
- Date time transforms for calendar are rich, and calendars and more advanced date columns can be created. Customizations like day.Monday to change start of week, or first of month to correctly reflect are some useful tools for analysis
- Learn to convert a list to a table – we could use a date series, and convert to a calendar table
- Remember if power query has multiple steps, then each step will need to be gone through whenever data is loaded, so remember to have optimal steps

Calculated Column Best Practices

- Create them where we can most optimally and practically do. As a general rule, the most optimal path begins at:
  - Data Source
    - Power Query
      - Power BI Front end
        - Published Reports
- Remember that more calculations on the front end may not be able to take advantage of the Vertipaq engine

Data Modeling recap/refresh

- It is an extension of ETL but since we need to setup relationships in Power BI, we will recap in this context. Data Model understanding begins at or prior to Power Query ETL phase in the workflow
- Normalization and Denormalization
- Facts (data – quants, metrics..) and Dimensions (lookups – descriptive attributes, categorical, filter/group…)
- PK (dim tables), FK (fact tables), Composite Key, Index, Unique Key…

Data Model view

- View all tables, properties, etc..
- Create and Manage relationships, directionality, active/inactive…
- Star Schema
  - Simple 1 level model, for example 1 facts described by dimension tables, and that ends the traversal of relationship depth
- Snowflake schema
  - Multi-level star schema
  - Dimension can have their own starred dimension / lookup tables
  - Fact to dim table relationship chain is longer, traverses multiple levels/tables
- Only 1 active relationship between 2 table columns, i.e. dim table key can’t map to 2 columns in related fact table, making inactive helps use DAX to still leverage relationship advantages on an as needed when needed and for the duration needed

Data Modeling continued

- Relationship cardinality (material is covered in the other DAX deck also)
- Basically it is the uniqueness of values in a column
- **Many-to-many is very complex can be visited after learning all the other stuff**
- Connecting multiple fact tables (ex: families and adults) requires using shared dimTables

Data Modeling continued (filter context again)

- Filter context and flow:
  - Filter context is in the direction of the arrow
  - It always flow downstream and not upstream
- Bi-directional filters:
  - Updating cross-filter direction from single to both (this is a hack to connect from fact table to another fact table thru dim table
  - We should be able to avoid this when possible, because we need to know our data really well otherwise debugging headache
  - Too many ambiguity issues since multiple filter contexts arise, and they could start conflicting with each other

Data Modeling continued

- Hide fields from report view so users do not have access to fields we do not want them to filter by, so for example we can hide foreign keys and allow them to only filter by primary keys. We can hide entire table if the use case needs it
- We can create additional layout tabs in the model view of Power BI. This is particularly helpful when there are a lot of tables, helps organizing views logically
- Leverage the data format and category options in the model view to enrich the display of table fields on reports
- Similarly leverage hierarchy setups that can help with matrix style reports and drill downs

DAX

- Quick Measures are more NLP style, and are threatening to me ☺
- We can organize explicit measures under a dedicated measures table, and have folders within (using the properties pane on the model view) to categorize the different measures we create
- The filter icon above the visual provides insight into the filters used/applied with the measures on the visual
- Calculated Columns:
  - Row context, usually value in column is dependent on other columns in the same row, though occasionally it could reference other table values, increases file size and table data load times potentially
  - Mostly used as a shortcut for filtering data in reports
- Measures:
  - Filter context mostly, does not add to the size of the file (table) since it is an aggregation function that may not be table-row specific as such
  - Useful primarily in visuals, and is designed to interact with various filter contexts on the visual/report

DAX

- Use a dedicated Measures table as mentioned earlier, and better still organize measures under folders so the groups can be more readable and maintainable
- Measure values whether they appear in a row style report or points on line charts or individual values on a card are evaluated independently, i.e. for example if there are 10 rows and a total field in a table chart, each row's measure value is independently evaluated, and the total is not a summation of the 10 rows but rather a value evaluated in the context of any row-context absence
- From a flow perspective it starts with the detection of the overall filter context, and then proceeds to look at it from the row perspective, and then it flows down to the related table if relevant and finally evaluates against the filtered table
- Refer for more details in the DAX ppt

Visualzzzz

- Start with the 3 fundamental questions at this stage of the workflow:
  - Type of data
    - Geospatial, timeseries, hierarchical, financial, etc..
  - What we want to communicate
    - Compare, composition, relations, distribution, etc..
  - Who is the end user
    - Role(s) of the stakeholders
      - Analysts (tables, combo charts, granular, attention to details)
      - Managers (summaries, actionable insights, some data to support insight)
      - Executives (High Level, crystal clear, not cumbersome – KPIs, cards, simple charts, minimal detail and stuff)
- The 4th element would be to be aware of any compliance, security and accessibility standards
- KEEP IT SIMPLE AND CLEAN

Visualzzzz – Dashboard design framework

- Define the purpose
- Choose the right metrics
- Present the data effectively
- Eliminate clutter and noise
- Use layout to focus attention
- Tell a story

- **Perfection is achieved when there is nothing left to take away**

Visualzzzz – Dashboard design framework

- F pattern and Z pattern for layout dashboard layout design
- Try to sketch it out mentally on a piece of paper the overall layout/navigation stuff, the potential data cards and visual components we could show
- Play around in Power BI by laying out containers images placeholders (without data) to help get a better sense of space and visual. Placing navigation menu item placeholders if planned would also help get a sense of look and feel
- Learn to align/format boxes above using the appropriate alignment menus
- Use the View→Selection pane to name objects/shapes and placeholders, and maybe grouping too since it becomes easier to manage down the line

Visualzzzz – continued

- While designing line charts, explore trends, forecasts, confidence levels, units for forecasting (try to go with points since it auto adjusts the last x number of time dimensions based on the overall chart line period
- Use tooltips and cards wherever possible
- Copy and paste if using same visuals so a lot of formatting can be avoided. You can then just focus on changing the data to be presented in those pre-formatted charts

Visualzzzz – filters, chart types overview

- While using cards, remember to provide contexts and KPI visuals really help to illustrate the comparison and context of numbers. KPI cards are also very useful on dashboards
- Donuts are good but not ideal for comparison. So better to use bar charts and better brings out categorical comparative charts
- Donuts and pies if using limit to 3-4 components and filter out outliers..
- Filters: Visual Level, Page Level, report (all pages) level

Visualzzzz – filters, chart types overview

- For granular and more dense info to be presented tabular/matrix is best – certain industry/domains prefer this, generally this is the closest to what most BI developers/report jobs do – another characteristic is these types of reports are go-to for SQL developer skills sets – very important
- Tabular/Matrix reports are also very conducive to conditional formatting (Format pane under Cell elements)
- Generally as mentioned earlier tabular reports outside BI are done from SQL to excel/csv. Top N / Limit top x number of results is another popular variation since the nature of granular data might be overwhelming to the user. In Power BI, we can do the Top N from the filter pane and set the N value from the column being limited
- Hack: In Power BI the card visual always shows 1 value and if the data is more than 1, it just shows the 1st value. So if we do not use a measure, we can use the filter pane, and do a Top N and set it so we can show the Top N highlight on a card!

Visualzzzz – filters, chart types overview

- Power BI maps works best if we assign categories to geospatial fields, lat/long fields work best
- Sometimes the administrator may have to allow users to enable the maps feature in tenant settings to use it
- Slicers are visual filters and apply to the report page by default. It is a good idea to have the ability to Apply/Clear all filters option for the slicers
- While exploring data over multiple periods generally older data is messed up, and it is particularly seen when we are doing a Top N or related aggregation. Learn to use HASONEVALUE in a IF condition to ensure duplicates or related messiness can be handled
- Gauge charts are good to illustrate performance measurements and how we are doing against targets/benchmarks
- We can create target measures and use it to do a **rule** based conditional formatting

Visualzzzz – filters, drills, chart types overview

- Area charts are similar to line charts but filled beneath like a stack
- To change levels of granularity use drill-up and drill-down. Drill-thru is for all levels
- We can either use the complete default hierarchy by using date fields so all levels date components can be drilled down, or depending on the consumer of the report, we can add the field levels that would be appropriate to the user, for example program directors want to drill down to month level maybe not week or day level
- Reports on a page can be set as a drill through – this is set on the page information pane where we can change from standard to drill through. Setting this up like this, will enable a drill-through option from any other page on the dashboard/reports, so we can select a county from another report, right click and it leads us to the county detail page

Visualzzzz – filters, drills, interactions, bookmarks

- Report interactions to customize how filters can be applied across related visuals using cross-filters, etc are very valuable – whether to highlight or filter or even not interact decisions need to be made. Remember to enable edit interactions menu in the format page before we start playing around
- Bookmarks are essentially that – you can go back to the state you had bookmarked after playing around. There are other use cases too like provide quick actions to specific report states we know the user frequently looks to. So we can enable bookmark when we have report in a state, and then we can add a reset button from the buttons menu, and assign the action to take us back to the bookmark

Visualzzzz – interactions, slicer panels, navigation buttons

- While adding custom navigation buttons, remember to start off configuring them in a blank page because if we try to do it in a page that already has the reports we want to go to, it will not allow us to navigate, so do it on a blank page, and cut and paste
- The custom buttons can be different icon/images that we can place, and we can even configure it to be different icons when certain mouse actions are performed like when we hover or press or what it would be by default state and so on. Once that part is configured, we can assign a page navigation as an action, and mapping it to on-press will take us to that report page
- Adding custom slicer panels is similar to adding custom buttons but you can do it on the same page. Basically you add different shapes there, and add slicers, add data fields to it and configure. We can then add a back button to hide the slicer panel and go back to main navigation menu. We will need to add 2 bookmarks one for hiding slicer panel and another for showing. Once we create the slicer panel, we also want to ensure that all the objects are grouped together as “Slicer Panel” or whatever name we want. The final step would be to associate the bookmark action to the corresponding buttons. It is also a good idea to unselect the data menu option from the hide and show slicer panel bookmarks so that the report state is not altered when the slicer panel is shown or hidden

Visualzzzz – sliders from parameters for forecasting what-if scenarios and reusable charts

- Numeric range parameters are helpful for what if analysis. So we basically specify a range and increment and Power BI auto creates a table for this with the two DAX measures within. We can then create new DAX measures for example increasing work activity hour participation by 1+numerical range parameter percentage for example. The numerical range parameter itself behaves like a slicer, and so when we change it, the corresponding changes are reflected on the line chart for example if we have it setup with 2 lines – 1 for original activity hours, and another data point referencing the new DAX measure
- Fields parameter is another cool deal where we can change the dimension/metric we are seeing on the same chart. So we can add multiple measures or table fields as a list to the parameter list, and change the chart measure to refer to the new parameter we created it just works!

Visualzzzz – Tooltipping and *custom visual*

- Custom tool tips are another cool and value add component. We create a report page as a tooltip, and then add the various measures and stuff we want to see on it. We can then attach this page which is now of type tooltip to the visual where we want to show this custom tooltip page (change the default tooltip to the report page option and select the custom tooltip page)
- To import custom visual, we need to be logged into Power BI service but there maybe a learning curve to use it and there maybe compatibility issues as well

Visualzzzz – closing out from desktop to publication

- We can define roles in desktop but the user assignment happens in Power BI service, and role assignment enables data access restrictions, so for example, 1 county administrator if set the role accordingly, will not be able to view another county data or whatever rule we set up for this role. But we can test the role view without publishing to service
- Mobile layout – pretty straightforward we design in desktop and drag/drop visuals to the mobile layout, not sure if we want to focus yet on this, we will see