Plan a data analytics environment

1. Identify requirements for a solution, including components, features, performance, and capacity stock-keeping units (SKUs)

License Model	Acronym	What you can use	Note
Free			Cannot view shared Power BI objects
Pro		Power BI	
Premium per-user	PPU	Power BI	
Premium per capacity (up to June 2024)	P	Fabric	You can use a free Microsoft Fabric license to viewer shared Power BI objects.
Embedded	A / EM	Power BI	
Fabric capacity	F	Fabric	If F32 or below, you need a Power BI paid license to view shared Power BI objects. If F64 or above, you can use a free Microsoft Fabric license to viewer shared Power BI objects.

SKU	Capacity Units (CU)	Power BI SKU	Power BI v-cores	Cost per hou
F2	2	-	0.25	$0.36
F4	4	-	0.5	$0.72
F8	8	EM/A1	1	$1.44
F16	16	EM2/A2	2	$2.88
F32	32	EM3/A3	4	$5.76
F64	64	P1/A4	8	$11.52
F128	128	P2/A5	16	$23.04
F256	256	P3/A6	32	$46.08
F512	512	P4/A7	64	$92.16
F1024	1024	P5/A8	128	$184.32
F2048	2048	-	256	$368.64

- Reservation (for a year): 41% discount.
- OneLake storage: 2.6 cents/Gb/month, or 4.68 cents if you have enabled Business Continuity and Disaster Recovery.
- You can buy Fabric either pay monthly or with a capacity reservation (reserved for a year in advance).
    - You may be able to cancel a capacity reservation, but there may be a termination fee.
    - It would probably be refunded as a credit.
- You can buy a Microsoft Fabric Capacity reservation by:
    - Going to the Azure portal,
    - Searching for Reservations in the top bar.
    - Click +Add, then Microsoft Fabric

- A Capacity Unit is a pool of compute power, needed to run queries, jobs or tasks in Fabric.
- If you need additional capacity, you can add it on a pay-as-you-go basis.
    - You can scale this additional capacity up or down, or pause it.
- A Power BI Pro or Premium Per User license is needed to view Power BI content, unless you have a Premium capacity or an F64 or larger capacities (the new equivalent).
- You can use other Fabric content without a Power BI Pro license. So you can use pipelines, create data warehouse, use notebooks etc with a Fabric Free license, as long as:
    - There is a Fabric capacity, and
    - You have at least a Viewer role for the workspace.
- To check whether Fabric is enabled, go to the Admin portal – tenant settings, and enable "Users can create Fabric items".

2. Recommend settings in the Fabric admin portal

- To administer the admin portal, you need either:
    - Global administrator,
    - Power Platform administrator, or
    - Fabric administrator.
- If you don't have one of the roles, you will only see "Capacity settings" in the admin portal.
- You can get to the settings by going to Settings – Admin portal – Tenant settings.
    - Microsoft Fabric
        - Users can create Fabric items – can be managed at the tenant and capacity levels.
    - Help and support settings
        - Users can try Microsoft Fabric paid features – a free 60-day trial.
    - Workspace settings
        - Create workspaces (enable),
        - Use semantic models across workspaces – users would still need the Build permission.
        - Block users from reassigning personal workspaces (My Workspace) – to stop users from changing the capacity assignment of My Workspace, as the data might change region, which may be in breach of GDPR or other data-residency rules.
        - Define workspace retention period – by default, workspaces are retained for 7 days before being permanently deleted. This can be changed for up to 90 days.
    - Information protection
        - Allow users to apply sensitivity labels for content.
    - Export and sharing settings
        - Guest users can access Microsoft Fabric – guests would be accessing via Microsoft Entra B2B (Business to Business).

        - Guest users can browse and access Fabric content
        - Guest users can work with shared semantic models in their own tenants

3. Choose a data gateway type

- Covered in PL-300 course.

4. Create a custom Power BI report theme

- To create a custom Power BI report theme:
    - In Power BI Desktop, go to View – Themes – Customize current theme.
- In the Customize theme, you can change:
    - Name and colors – these are:
        - the theme name,
        - the 8 theme colors
            - In the color chooser, the colors shown are 60%, 40%, 20% lighter and 25% and 50% darker from the theme colors.
        - the 3 sentiment colors
            - used in KPI visuals and waterfall charts to show positive, negative or neutral results.
        - 3 divergent colors
            - Used in conditional formatting.
        - Advanced – structural colors
            - First-level elements (also known as foreground), such as table/matrix colors,
            - Second-level elements (Neutral foreground), such as label colors,
            - Third-level elements (background light), such as table/matrix grid colors.
            - Fourth-level elements, such as legend dimmed color.
            - Background elements, such as slicer dropdown items.
            - Secondary background elements, such as Table/matrix grid outline color,
            - Table Accent, is the table/matrix outline color (when an accent color is present).
    - Text
        - Font family, size and color for: General, Title, Cards and KPIs, and Tab headers.
    - Visuals colors
        - Background: Color and Transparency,
        - Border: On/off, color and radius,
        - Header: Background and border color, transparency and icon color,
        - Tooltip: Label, Value and Background color.

- Page
  - Wallpaper: Color and Transparency,
  - Page background: Color and Transparency
- Filter pane
  - Filter pane: Background color, Transparency, Font/icon color, Title and Header font size, and Checkbox/Apply color.
  - Available/applied filter cards: Background color, Transparency, Font/icon color, and Font size.

Implement and manage a data analytics environment

5. Implement workspace and item-level access controls for Fabric items

- Microsoft Fabric uses Power BI roles for Microsoft Fabric capabilities.
- The following is what each role does for Microsoft lakehouses/warehouses and related apps
- Viewer
  - View/read content of data pipelines and notebooks
  - Execute/cancel execution of data pipelines (not notebooks)
  - View execution output of data pipelines and notebooks
  - Connect to and Read data/shortcuts through Lakehouse/Warehouse SQL analytics endpoint
  - Reshare items in a workspace, if you have Reshare permissions
- Contributor (as Viewer, plus)
  - Read Lakehouse/Warehouse data/shortcuts through OneLake APIs and Spark.
  - Read Lakehouse data through Lakehouse Explorer.
  - Write/delete data pipelines and notebooks
  - Execute/cancel execution of notebooks
  - Schedule data-refresh via the on-premises gateway
  - Modify gateway connection settings
- Member
  - Add members, contributor and viewers
  - Allow others to reshare items
- Admin
  - Update and delete the workspace
  - Add/remove admins, members, contributors and viewers.
- To give access to your workspace:
  - In the workspace, click on Manage Access (it may be in the … section)
  - Click “+Add people or groups”.

    - Enter name/email and role, and click Add.
- You can view/modify access later if needed.
- You can also manage permissions for lakehouses by clicking on the … next to the lakehouse (in the Workspace) and going to Manage permissions. You can assign the following permissions:
    - Read all of the lakehouse table data (not files) using the SQL endpoint,
    - Read all of the underlying data files with Apache Spark,
    - Build reports on the default semantic model.
- For data warehouse:
    - They cannot see any of the data unless at least one additional permission is selected.
    - the "Apache Spark" permission is expanded to "Read all of the data warehouse's underlying OneLake files using Apache Spark, Pipelines, or other apps which access the OneLake data directly".
- You can also share notebooks with the following permissions:
    - Share (or Reshare) the notebook with others,
    - Edit (or Write) all notebook cells, and
    - Run (or Execute) all notebook cells.

6. Implement data sharing for workspaces, warehouses, and lakehouses

- To share items via a link:
    - click the Share button.
    - In the “Create and send link” dialog, click “People in your organization can view”.
    - In the “Select permissions” dialog, choose either:
        - People in your organization
            - This allows for read-only access (as a minimum).
            - It does not work for external or guest users.
        - People with existing access
            - This generates a link. It does not change the access.
        - Specific people.
            - This also allows guest users in your organization’s Microsoft Entra ID.
    - You can also allow for re-share and edit permissions.
        - If you re-share, you can only re-share based on the permissions you have.
    - Then click Apply.
- In the “Create and send link” dialog, you can copy the sharing link, generate an email, or share via Teams.
- You can manage item links:

    - Click on “Manage permissions” in the “Create and send link” or click on “Manage permissions” in the … next to the item in the workspace.
    - You can either:
        - Click on the Edit next to “Links that give access”,
        - You can then modify the permissions for the link, or delete the link.
    - Or
        - Click on “Advanced” to:
            - View/manage/create links,
            - View/manage/grant who has direct access,
            - Apply filters or search for links/people.
- Note: The Browse – “Shared with me” does not show Fabric items which are not Power BI objects.

7. Manage sensitivity labels in semantic models and lakehouses

- Covered in PL-300 course.

8. Configure Fabric-enabled workspace settings

- To access workspace settings, go to Workspaces, click on the … next to the relevant workspace, and go to Workspace settings.
    - You can also access it in the workspace by clicking on “Workspace settings”.
- There are the following tabs:
    - About – you can change:
        - The workspace image,
        - Description,
        - Domain (a group of workspaces),
        - Contact list – which users receive notifications about issues in the workspace.
        - Microsoft 365 and OneDrive – configure a Microsoft Group where the SharePoint document library can be made available to workspace users.
            - You will need to give permissions to the Microsoft 365 Group membership.
    - License mode: Pro, Premium per-user, Premium capacity, Embedded, Fabric capacity and Trial.
    - Azure connections: configure dataflow storage to use Azure Data Lake Gen2 storage and Azure Log Analytics.
    - System storage
        - Manage your semantic model storage (workspaces can contain up to 1,000 semantic model per workspace),
        - View storage,
        - Delete semantic models (reports and dashboards based on those models would not longer work),

- Git integration
  - Connect workspace to an Azure Repo (see item 10).
- OneLake
- Other – remove this workspace

9. Manage Fabric capacity

- You can manage Fabric Capacity by installing the Microsoft Fabric Capacity Metrics app.
- To install it:
  - Go to Apps – Get apps,
  - Search for Microsoft Fabric,
  - Click the "Microsoft Fabric Capacity Metrics" app.
  - Click "Get it now".
- To run it for the first time:
  - Go to Apps – click the "Microsoft Fabric Capacity Metrics" app,
  - You will see the message "You have to connect to your own data to view this report". Click Connect.
  - Enter:
    - CapacityID – you will see this in Settings – Admin portal – Capacity settings, and select a capacity. The CapacityID is a series of hexadecimal characters and dashes.
    - UTC_offset – the number of hours before/after UTC (GMT).
    - Timepoint/Timepoint2 – this is an internal value, which you should not fill in.
    - Advanced – whether the app automatically refreshes your data at midnight.
  - Click Next.
  - In the "Connect to Microsoft Fabric Capacity Metrics", fill in:
    - Authentication method – by default, use OAuth2,
    - Privacy level setting for this data source – using Organizational to access all the organization's data sources.
  - Click on "Sign in and connect".
  - Select a capacity from the "Capacity Name" dropdown.
  - It may take a few minutes for the app to get your data.
- The Compute page contains:
  - A ribbon chart containing an hourly view of:
    - Capacity Units (CU) (in seconds),
    - Duration (processing time in seconds),
    - Operations (count),
    - Users (who have performed Operations),

- Capacity utilization
  - It shows:
    - Background % (billable) and non-billable: % of CU consumption used in a 30-second period. These are operations not triggered by users – for example, data refreshes.
    - Interactive % (billable) and non-billable: Resources triggered by users, associated with interactive page loads.
    - Autoscale % – shows timepoints where the capacity is overloaded.
    - CU % Limit – the threshold of the allowed CU %.
- Throttling. This is a limit of your CU. It happens if the CUs for interactive and background operations exceed the allowance in a 30 second timepoint. Either:
  - You have Autoscale enabled. If so, a new CU will be added for the next 24 hours, up to the maximum number of CUs allowed. If it goes above that threshold, throttling will happen.
  - You do not have Autoscale enabled. Then throttling will be applied.
- You can use a linear or logarithmic scale, and use filters.
- It shows:
  - Interactive delay
    - Where capacity went over by between 10 and 60 minutes.
    - User interactive jobs are throttled.
  - Interactive rejection
    - Where capacity went over by between 60 minutes and 24 hours.
    - User interactive jobs are rejected.
  - Background rejection
    - Where capacity went over after 24 hours.
    - User scheduled background jobs are rejected and not executed.
- Overages:
  - Add % - the carry-forward % during the current period,
  - Burndown % - the carry-forward % burned down during that period,
  - Cumulative % - the cumulative %.
- System events:
  - Displays pause/resume capacity events, with
    - Time,
    - State (suspended and active), and
    - State Change Reason
- Matrix by item and operation

        - Shows the “performance delta”, which compares fast operations (under 200 milliseconds to complete) for over the last week, the current value, and the value 7 days ago.
        - This can be used to see if your average performance improved/worsened over the past week.
            - The higher the value, the better the performance.
        - You can sort the matrix by the “performance delta” to find the biggest change in their performance.
        - A high CU utilization means that it is being heavily used or run many operations.
        - A low CU utilization might be volatile.
        - It shows:
            - Workspace,
            - Item kind (type),
            - Item name,
            - CU in seconds over the last 2 weeks,
            - Duration (Processing time) over the last 2 weeks,
            - Users (count), and
            - Billing type (Billable, non-billable, and both).
        - Using the “Select optional column(s)”, you can also add:
            - Rejected/failed/invalid/inProgress/Successful count (number of operations),
            - Virtualized item/workspace,
            - Item Size (Gb),
            - Overloaded minutes (number of 30 second increments where overloading occurred at least once),
            - Performance delta.
- Storage page:
    - You can use the following filters:
        - Capacity Name,
        - Date Range,
        - Experience, and
        - Storage type.
    - You can see in cards:
        - Number of workspaces,
        - Current/billage storage (in Gb)
    - There is a “Top workspace by billable storage %”, which includes:

        - Workspace name/ID,
        - Operation name,
        - Deletion status (whether it is active),
        - Billing type (whether it is billable),
        - Current/billable storage in Gb,
        - Billable storage % (of the capacity)
    - Column charts showing Storage (Gb) and Cumulative Billable Storage (Gb) by date/hour.
    - You can also Export the Data.
- You can also monitor a paused capacity. It shows all the paused capacity events

Manage the analytics development lifecycle

10. Implement version control for a workspace

- Git integration allows you to integrate your development processes into Fabric. It works on a workspace level. Note:
    - this is used through Azure DevOps Git Repos with the same tenant as the Fabric tenant
    - not through GitHub Repos, and not the on-premises version of Azure DevOps.
- It allows you to:
    - Backup and version work,
    - Revert to previous stages if needed,
    - Collaborate with others,
    - Work alone using Git branches,
    - Use Git source control tools.
- You can use it for:
    - Data pipelines,
    - Lakehouse,
    - Notebooks,
    - Paginated reports,
    - Reports (except where the semantic model is in SSAS or Azure Analytics Services, or semantic models hosted in My Workspace),
    - Semantic models (except live connections and models created from the Data warehouse/lakehouse).
- You need:
    - An active Azure account for the same user that uses the Fabric workspace,
    - Access to an existing Azure DevOps repository,
    - Power BI Premium license (for Power BI items only) or Fabric capacity (for all Fabric items),

  - In Settings – Admin Portal, have "Users can create Fabric items" enabled.
- To sign up to Azure Repos:
  - Go to the Azure Portal (portal.azure.com).
  - Search in the top bar for "DevOps", and click on "Azure DevOps organizations".
  - Click on "My Azure DevOps Organizations".
  - Click on "Create new organization", and enter your organization details, including:
    - The organization name,
    - The location for hosting your projects
  - Create a new project.
  - Enter the project details, including:
    - The Project Name, and
    - The visibility.
  - In Repos – Files, click "Initialize" to create an empty branch.
- To connect your workspace to an Azure repo:
  - You will need Admin rights for the Workspace, and Read rights for the Git repo.
  - Go to the relevant workspace.
  - Click on "Workspace settings" (it might be in the … section),
  - Go to Git integration. Select:
    - Organization,
    - Project,
    - Git repository,
    - Branch
      - You can click "+New Branch" to create a new branch.
      - You will need Admin rights for the workspace, and Write and Create branch rights for the Git repo.
    - Folder:
      - Use an existing folder,
      - Enter a name for a new folder, or
      - Leave blank to use the root folder of the branch.
    - You can only connect a workspace to one branch and one folder at a time.
- You can disconnect by going to Git integration and click "Disconnect workspace".
  - You will need Admin rights for the Workspace, but no rights are needed for the Git repo.
- After you connect, if the workspace or Git branch is empty, content will be copied.
  - It doesn't sync data, but only the schema.
- Once connected, the Workspace includes a "Git status" column showing its status.

- To commit changes to Git:
    - You will need at least Contributor rights for the Workspace, relevant permissions for the items and external dependencies, and Read and Contribute rights for the Git repo.
    - In the workspace, click on the “Source control” icon.
    - Go to the Changes tab in the Source control pane.
        - A list shows the changed icons, with icons showing:
            - new (green +),
            - modified (brown non-equal sign),
            - conflict (red x), or
            - deleted (red -).
    - Select all the items you want to “commit” (transfer).
        - To commit all, check the top box.
    - You can add a comment in the Commit Message box.
    - You can then click "Commit".
        - Afterwards, the status of the selected items would change from “Uncommitted” to “Synced”.
        - You can also see the time of the last commit in the footer.
    - If you click "Update", then all changes in the branch will be updated.
- If changes have been made in the connected Git branch:
    - You will see a notification.
    - You can click on the “Source control” icon and go to the Updates tab to see a list of all changed items.
    - You can then click on “Update all”.
    - You will need at least Contributor rights in the Workspace, relevant permissions for the items and external dependencies, as well as Read rights for the Git repo.

11. Create and manage a Power BI Desktop project (.pbip)

- Power BI Project (.pbip) stores your work with report and semantic model item definitions as separate text files in a folder structure.
    - The Semantic Model and Report will be saved in folders called [projectname].SemanticModel and .Report.
    - There is also a [projectname].pbip file, which points to the report folder.
    - There is also a .gitIgnore file, which trackes files that Git should ignore.
- This allows:
    - You can use source control such as Git to track version history, compare revisions, and revert to previous versions.
    - You can edit the definition files in Notepad or Visual Studio Code
        - VS Code can integrate with Git.

    - You can Tabular Model Scripting Language (TMSL) to make changes to your items.
        - This can be used in Tabular Editor.
    - You can use it in a Continuous Integration and Continuous Delivery (CI/CD) system.
- This is currently in preview. To enable it, go to Power BI Desktop – File – Options and settings – Options – Preview features, and check "Power BI Project (.pbip) save option".
    - This stores it in Tabular Model Scripting Language (TMSL).
    - You can also enable "Store semantic model using TMDL format". Tabular Model Definition Language is a more human-friendly format, which is more readable and more easily editable.
- To save a project as a pbip, go to File – Save As, and change the "Save as type" from "Power BI file (*.pbix)" to "Power BI project files (*.pbip)".
- When you have done so, the title bar shows "(Power BI Project)".
    - If you click on the title bar, you will see the report and semantic model links and display names.
- You can then upload it to a Git branch.

12. Plan and implement deployment solutions

- You can create a deployment pipeline of between 2 and 10 stages (workspaces).
    - The workspaces must reside on a Fabric capacity.
- They would generally be in the categories of:
    - Development – create/design new content,
    - Test – release to testers, and
    - Production – share final version.
- To create a pipeline:
    - Go to Workspaces, and click "Deployment pipelines" (near the bottom).
    - Click "Create pipeline".
    - Enter a name and optional description in the "Create a deployment pipeline" dialog box.
    - Enter the pipeline stages.
        - By default, there are 3 stages named Development, Test and Production.
- Pipeline admins who are also Workspace Admins can then assign workspaces.
    - In the pipeline, you should then select the workspaces next to the pipeline stage and click "Assign a workspace".
    - Note: a workspace can only be assigned to one pipeline.
- Pipeline admins who are or are <u>not</u> Workspace Admins can unassign a workspace from a pipeline stage. To do this:
    - Open the pipeline,
    - In the relevant stage, click the ... and select Unassign workspace, then click Unassign.

- You can compare stages by looking at “Compare” next to a stage.
    - The icon compares that stage with the next stage. It shows:
        - Green – metadata for all items in both stages is the same,
        - Orange – either some items have changed/updated, or the number of items are different.
    - Where it is orange, you can click on the Compare link to compare the items. It will show:
        - New – new item in the source stage,
        - Different – exists in both stages, but has been changed in the last deployment. This includes if you have changed folder location.
        - "Not in previous stage" – new item in the target stage.
    - If something has been changed, then there is a “Review changes” button, which allows you to see the changes the item, either side-by-side or inline.
- To deploy content, you can either:
    - Click on “Deploy to X” – this deploys all content to the next stage,
    - Click on “Show more”. You can then select specific items to be deployed.
    - You can then add a note and click “Deploy”.
- To view the deployment history:
    - Go to the pipeline,
    - Click on “Deployment history”. It shows:
        - Deployed to – stage,
        - Date/time – at the end of the deployment,
        - Deployed by – person or service principal,
        - Items – the new, different and unchanged items, and the items which failed to deploy.
        - A note (if it exists)
        - Deployment ID
        - Deployment Status (Successful/Unsuccessful)
- When you deploy items from a previous stage to a later stage:
    - if any content has the same name in both stages, the content will be overwritten in the later stage.
    - Content in the later stage that is not in the earlier stage will remain (will not be deleted).
    - Up to 300 items can be deployed in a single deployment.
    - You can group items together in folders.
- If you are deploying (for example) a report and not the semantic model it relies on, then:
    - If the semantic model exists in the later stage, it will connect to the later stage model.

  - If the semantic model doesn't exist in the later stage, then the deployment will fail.
  - Note: you cannot download a .pbix file after deployment.
  - You cannot deploy semantic models which have real-time data connectivity.
- Any user (free user) can view the list of pipelines.
- To create the pipeline, you would need the "pipeline admin" permission (as a minimum permission) in a Pro, Premium Per User, or Premium Capacity.
- To give a user the "pipeline admin" permission, go to "Manage Access" and click on "Add people or groups".
- It allows:
  - Create a pipeline
  - View/share/edit/delete the pipeline,
  - Unassign a workspace from a stage,
  - Can see workspaces that are assigned to the pipeline,
  - View deployment history,
  - View the list of items in a stage,
  - Manage pipeline settings,
  - Add/remove a pipeline user
- Pipeline admins who are also Workspace Contributors can also:
  - Compare two stages
  - View or set a rule
- Pipeline admins who are also Workspace Members can also:
  - Deploy items to the next stage (if a workspace member/admin of both workspaces)
- Pipeline admins who are also Workspace Admins can also:
  - Assign a workspace to a stage
- Pipeline admins who are <u>not</u> Workspace Admins can:
  - Unassign a workspace to a stage.
- Plan and implement deployment solutions
- "Pipeline admin" allows you to:
  - Create a pipeline
  - View/share/edit/delete the pipeline,
  - Unassign a workspace from a stage,
  - Can see workspaces that are assigned to the pipeline,
  - View deployment history,
  - View the list of items in a stage,
  - Manage pipeline settings,

    - Add/remove a pipeline user
- Pipeline admins who are also Workspace Contributors can also:
    - Compare two stages
    - View or set a rule
- Pipeline admins who are also Workspace Members can also:
    - Deploy items to the next stage (if a workspace member/admin of both workspaces)
- Pipeline admins who are also Workspace Admins can also:
    - Assign a workspace to a stage
- Pipeline admins who are not Workspace Admins can:
    - Unassign a workspace to a stage.
- You can also configure deployment rules.
- These are used for changing the content but keeping some settings as per the deployment rule. It is used for:
    - Dataflow/semantic model/datamart – to specify the data sources or parameters for the dataflow/semantic model/datamart,
    - Paginated report – to specify the data source, and
    - Notebook – the default lakehouse for a notebook.
- To do this:
    - Next to the pipeline stage, click on the "Deployment rules" button.
        - You can't create it in the first stage – it's for the target stage.
    - Select the items to create the rule for.
    - Click on "+Add rule" next to:
        - "Data source rules" – select from a list, or select Other and manually enter the new data source (of the same type).
        - "Parameter rules" – select the parameter and enter the value.
        - "Default lakehouse rules" – select the lakehouse to connect the notebook to in the target stage.
- You can use the following data source types:
    - SSAS or Azure Analysis Services,
    - Azure Synapse,
    - SQL Server or Azure SQL Server,
    - Odata Feed,
    - Oracle,
    - SapHana (using import mode, not direct query)
    - SharePoint, and
    - Teradata.

    - But not dataflows.
- Note:
    - if you delete an item, its rules are deleted as well, and cannot be restored.
    - if you unassign and reassign a workspace, its rules are lost.

13. Perform impact analysis of downstream dependencies from lakehouses, data warehouses, dataflows, and semantic models

- To create an impact analysis, either:
    - Open the item and click on Lineage – Impact Analysis, or
    - Go to the workspace, click on … and select "View lineage".
- You can see:
    - "Child items": Direct children of the item (things which are directly dependent on it), or
    - "All downstream items": All affected dependent items downstream.
- You can browse by item type or by workspace.
- If you are making changes, you can click on "Notify contacts" to notify the contact lists of any relevant workspaces.

14. Deploy and manage semantic models by using the XMLA endpoint

- You can use Tabular Editor 2, SQL Server Management Studio (SSMS) and other external tools to manage semantic models with the XMLA endpoint.
- See topic 57.

15a. Create and update reusable assets, including Power BI template (.pbit) files

- Templates contain:
    - Report pages and visuals,
    - The data model definition, which includes:
        - Schema,
        - Relationships, and
        - Measures.
    - Any query definitions, which includes:
        - Queries,
        - Query Parameters.
- Templates do not contain data.
- To create a template, go to File – Export – Power BI template.
    - You can enter a description for your template.
    - Select your file location and name for your .PBIT file.
- To use a template, either:
    - Double-click on a .PBIT file in Windows Explorer, or
    - Go to File – Import – Power BI template.

    - Go to File – Open, "Browse this device".
- You can then:
    - Enter values for any parameters,
    - Enter the file location for any data sources if necessary.
        - You can then connect the data based on your credentials.

15b. Create and update reusable assets, including Power BI data source (.pbids) files

- .pbids files contain a JSON structure which point towards a single Power BI data source.
- To create a PBIDS File in Power BI Desktop:
    - go to File – Options and settings – Data source settings.
    - click on your Data source settings, and click on “Export PBIDS”.
    - Give the file a name and location, and click Save.
    - Note: if columns are encrypted in the data source, then it will generate an error.
- To open the .pbids file, double-click on it in Windows Explorer.
- When you open the .pbids file, you will be asked for any necessary credentials to open.
- You would then select any tables from that data source, and possibly the database and connection model if it isn’t part of the .pbids file.

15c. Create and update reusable assets, including shared semantic models

- To share a semantic model, either:
    - go to the semantic model and click Share (at the top) or “Share semantic model” (in the main section), or
    - go to the OneLake data hub, click on the … next to the item, and click Share.
- Enter the names or email address that you want to share the semantic model with. You can select:
    - Allow recipient to modify/share this semantic model,
    - Allow recipient to build content with the data associated with this semantic model,
    - Send an email notification.
    - Add an optional message, and then click “Grant access”.
- You manage permissions by:
    - going to the Workspace, click on … next to the semantic model and go to “Manage permissions”.
    - going to the semantic model and clicking on File – “Manage permissions”,
    - going to the semantic model, clicking on Share, then the … and Manage permissions.
    - going to the OneLake data hub, click on “Manage permissions”.
- To use the semantic model:
    - in Power BI Desktop:

        - go to Home – OneLake data hub – Power BI Semantic Model, or go to Home – Get data – Power BI semantic models
        - select the semantic model, and click Connect.
    - in Power BI Service:
        - go to OneLake data hub, and click on the semantic model.
        - You can then create a report.

Create objects in a lakehouse or warehouse

16a. Ingest data by using a data pipeline

- Once you have created a data flow, you can incorporate it into a data pipeline.
- In the workspace, go to New – Data pipeline.
- Provide a name for the data pipeline.
- Click on Dataflow activity.
- In the settings tab, select the Dataflow.
- Optionally, you can add an Office 365 Outlook activity to send an email notification.
    - Go to the Activities menu, click on the "Office 365 Outlook" icon, and connect to Office 365 Outlook.
    - Enter an email address, subject and body.
        - Additional properties can be set in the Advanced area.
    - You can then connect this new activity to the previous activity.
- To run the data pipeline:
    - Go to Home - Run or Run – Run in the data pipeline.
    - You can see the output in the Output tab.
- To schedule the data pipeline, see topic 24.

16b. Ingest data by using a data dataflow

- To ingest data by using a data dataflow:
    - Go to Data Factory.
    - Go to your Fabric-enabled workspace.
    - Go to New – Dataflow Gen2.
    - Go to Home – Get data, and select a data source.
    - You can transform the data, using the Power Query interface.
    - Go to Home – Add data destination, and select either:
        - Azure SQL Database,
        - Lakehouse
        - Azure Data Explorer (Kusto) or
        - Warehouse.
    - Then select:

        - your data destination,
        - the table name,
        - The update method
            - Whether data is being appended, or replaced.
- You can optionally export this dataflow as a template:
    - Go to Share – Export template
    - Add a name and an optional description, and click OK.
    - To use it again, create a Dataflow Gen2
- To schedule the refresh:
    - Go to the workspace and click on the … next to the Dataflow.
    - Go to settings.
    - Expand Refresh.
    - Change "Configure a refresh schedule" to On.
    - Select the refresh frequency - either Daily or Weekly.
        - If Weekly, select which day(s) of the week.
    - Click "Add another time" and select a time
        - This can be on the hour or on the half hour.
    - Add additional times if required.
    - Under "Send refresh failure notifications to", check or uncheck:
        - Dataflow owner, and
        - These contacts (and add contacts)
    - Click "Apply".

16c. Ingest data by using a notebook

- To load data from a file into a dataframe using a notebook:
    - Right-hand click the file (or left-click on the … next to the table) and select:
        - Load data – Spark or Pandas
            - This would generate the data in your notebook.
        - Copy relative path for Spark
            - This should be used if you are using this data source in this lakehouse.
            - Example: Files/MtoMActual.csv
        - Copy ABFS (Azure Blob Filesystem) path for Spark
            - This should be used if you are using this data source in another lakehouse.
            - Example: abfss://06e284ed-4b3f-4882-b35c-e96f16f2479f@onelake.dfs.fabric.microsoft.com/655ad85b-0764-4156-a93c-06f825e09a8d/Files/MtoMActual.csv

- If the last two options, create a PySpark cell and then enter:
- *df = spark.read.parquet("[Location]")*
- To load CSV, JSON, ORC or Parquet files, use:
    - *df = spark.read.csv("[Path]")* or .json or .orc or .parquet

      or
    - *df = spark.read.load("[Path]", format='csv', header=True)*
- To load data from a table into a dataframe, right-hand click the file (or left-click on the … next to the table) and select Load data – Spark.
- To read it using an explicit data structure, then you can add (for example, after *.read*):
    - *.schema(schemaTarget)*
- The schemaTarget is defined using StructType and StructField. You can generate this for an existing table by using:
    - *df.schema*
- An example of the schemaTarget is:
    - *StructType([StructField('Country', StringType(), True),*

      StructField('Location', StringType(), True),

      StructField('Actual', StringType(), True)])
- To save a dataframe as CSV or Parquet files to Files section of the default Lakehouse
    - *df.write.mode("overwrite").format("csv or parquet or json").save("Files/[FileName]")*
        - You can also use *(f"Files/{FileName}")*
        - You can use mode first, or format first – it doesn't matter what the order is.
- To read a table in Spark, use
    - df = spark.read.table("TableName")
- To read the parquet file with Pandas from the default lakehouse mount point and use:
    - *import pandas as pd*
    - *df = pd.read_parquet("/lakehouse/default/Files/[NameOfFile].parquet")*
- To use load the data using Pandas API:
    - right-hand click on the file, and select “Copy File API path”.
    - Example: /lakehouse/default/Files/MtoMActual.csv
    - Use the following code:
        - *import pandas as pd*
        - *df = pd.read_parquet(“[APIPath]/[LakehouseName]/Files/[NameOfFile].parquet")*
- Continued in topic 22c.

17. Create and manage shortcuts

- Shortcuts are OneLake objects which point to other storage locations.
    - They can be other OneLake storage locations, or external to OneLake.
    - The shortcut location is called the target path.
    - They appear as folders in OneLake.
- To create a shortcut:
    - Right-hand click on a folder (table or file) in the Explorer pane of the lakehouse, and select "New shortcut".
    - Select the source:
        - Internal sources – Microsoft OneLake,
        - External sources – Azure Data Lake Storage Gen2, Amazon S3 (or compatible), Dataverse or Google Cloud Storage.
    - Select the datasource (and authentication, if using an external sources),
    - Expand File/Tables, select the subfolder(s) (up to 50 subfolders), and click Next.
    - You will see the selected shortcut locations.
        - You can edit to change the default shortcut name, or delete any selection.
    - Click Create.
    - You need to have Contributor, Member or Admin role for the workspace,
    - You need write permissions for the shortcut location, and read permission in the target location.
- To use it:
    - The calling user must have read permissions for the target location.
- In a Lakehouse, shortcuts are shown at the top level of the Tables folder, or anywhere in the Files folder.
    - Shortcuts are not available in Tables folder subdirectories.
    - Ideally, don't use tables with spaces in the file name.
        - They will not be discovered as a Delta table in the lakehouse.
- To access shortcuts in the Table folder, you can use:
    - *df = spark.read.format("delta").load("Tables/MyShortcut")*
    - *df = spark.sql("SELECT * FROM MyLakehouse.MyShortcut LIMIT 1000")*
    - (In SSMS) *SELECT TOP (100) * FROM [MyLakehouse].[dbo].[MyShortcut]*
- Note:
    - If you delete the shortcut, the target is not affected.
    - If the target path moves, is renamed or is deleted, the shortcut can break.

18. Implement file partitioning for analytics workloads in a lakehouse

- Loading data using partitions allows you to separate data.

    - Each subset is called a partition or shard.
    - These partitions can be processed separately.
- When copying data in a data pipeline, go to the Destination tab.
- In the Advanced section:
    - Check "Enable partition".
    - In "Partition columns", select the relevant partition column(s).
        - The column(s) should be of string, integer, Boolean or datetime type.
        - If you are using multiple columns, then it is partitioned by the first columns, then by the second.
            - To reorder it, you can drag the columns.
- When you run the pipeline, the files will be stored as [ColumnName]=[ColumnValue]
    - You can view the files by right-hand clicking on the Lakehouse table (or left-hand clicking on the table and going to ...) and select "View files".
- To read multiple files into a dataframe, you can use:
    - spark.read.option("recursiveFileLookup", "true").parquet("*.parquet")
    - Using recursiveFileLookup = True will search through subfolders.

19a. Create views

- A view is a statement which has been saved, and can be retrieved.
- To create a view:
    - Go to the SQL analytics endpoint,
    - Write a query.
        - For example, SELECT * FROM NameOfTable
    - And then either:
        - Select the query statement, and click on "Save as view", or
        - Prefix the statement with
    - *CREATE VIEW Schema.TableName AS*
        - You can define the column names at the end of the Schema.TableName in brackets
    - You can use a WITH in the query.
- To use the view:
- *SELECT * FROM Schema.TableName*

19b. Create functions

- In Fabric, a CREATE FUNCTION can return a table, which you can use in the FROM clause.
    - In Fabric, you cannot return a single value (scalar function).
- The syntax is:
- *CREATE FUNCTION Schema.FunctionName*
- *(@Parameter AS ParameterDataType...)*

- *RETURNS TABLE*
- *[AS]*
- *RETURN Select_Statement*
- Parameters use a @ at the beginning, and contain a single value.
    - You cannot pass tables as a parameter.
    - "= default" is a default value for the parameter.
- You can use DECLARE statements creating a local data variable.
- Example function:
- *CREATE FUNCTION dbo.func_AddressData (@Country AS varchar(20))*
- *RETURNS TABLE*
- *AS*
- *RETURN*
- *SELECT **
- *FROM AddressData*
- *WHERE CountryRegion = @Country*
- Calling the function:
- *SELECT AddressID, City*
- *FROM func_AddressData('Canada')*

19c. Create procedures

- A procedure is a sequence of code that you can run outside of a SELECT statement.
    - You can have a single or multiple SELECT statements in a procedure.
    - It allows for input parameters.
    - You can run other procedures from the procedure.
- The syntax is
- CREATE [OR ALTER]
- [PROC or PROCEDURE]
- SchemaName.ProcedureName
- [@parameter datatype - optional]
- AS
- [BEGIN – optional]
- SQL Statement(s)
- [END - optional]
- To run the procedure, use
    - *ProcedureName*
    - optional followed by the parameters

    - preceded optionally by *EXEC* or *EXECUTE*.
- Example procedure:
- *CREATE PROC dataproc @passengerCount INT*
- *AS*
- *SELECT **
- *FROM [LakehouseTrial].[dbo].[datatable]*
- *WHERE passengerCount = @passengerCount*
- Example running of the procedure:
- *exec dataproc 2*
- Spark syntax - dataframes
- SELECT - df.select("columnName", "columnName2"])
    - selects only some columns.
    - you can end with an unnecessary comma: df.select("columnName", "columnName2"],)
- col("string") or column("string") refers to a column called "string".
- df.show()
    - shows dataframe (not list) in a text table.
- display(df)
    - shows dataframe (not list) in a graphical table.
    - From this display, you can:
        - Show the information in a table or a chart.
        - In the table view, you can:
            - sort ascending or descending, or copy the column name.
            - download the information to a CSV, JSON or XML file.
            - click Inspect to show the individual cell, or if you haven't selected an individual cell, the following for the table:
                - Missing, Unique and (for non-strings) Invalid Value,
                - A histogram.
            - "Search" to filter the table, either on all columns (the default), or on an individual column.
        - In the chart view, you can click on "Customize chart" to:
            - Change the chart type to:
                - Line chart,
                - Bar, Column or Area chart,
                - Pie chart,
                - Scatter chart,

    - Box plot,
    - Histogram chart,
    - Pivot table or
    - Word cloud.
  - You can customise the data used. For bar charts, you can also change the Key (the axis), Values, Series Group, Aggregation (Sum, Avg, Min, Max, Count, First and Last), and whether it is Stacked.
- df.collect()
  - shows dataframe in a list.
- df.schema
  - this shows the structure using StructType (one for the table) and StructField (one per column).
- df.summary
  - shows columns and data types.
- Spark syntax - dataframes
- df.column_name.alias and df.column_name. name
  - df.select(df.age.alias("age2"))
- df.column_name.concat(column1, column2) combines all the columns into a single column.
- To add comments, prefix the comment with a #.
- For WHERE clause, see topic 34.
- For GROUP BY clause, see topic 30.
- ORDER BY
  df.orderby(desc("columnName"), "columnName2")
  df.sort(asc("columnName"))
  df.sort("age", ascending=True)
  df.sort(df.column_name.desc()) or .asc())
  - orders dataframe by column(s). Default is ascending.
  - You can use asc(), asc_nulls_first(), asc_nulls_last() or the desc equivalent.
- df.columns
  - shows all the columns as a list
- df.describe(["columnName", "columnName2"].show()
  - show count, mean, stddev, min and max of the columns.
- df.head(n), df.take(n)
  - returns the top n rows as a list. If df.head() is used, returns top row. Cannot use show().
- TOP - df.limit(n)
  - returns the top n rows.

  - Note: in Spark SQL, TOP(10) or TOP 10 is not used. Instead, you should use LIMIT 10 at the end of the query.

- df.tail(n)
  - returns the last n rows as a list.

20. Enrich data by adding new columns or tables

- You can create new tables in a Dataflow Gen2.
  - In the Workspace, go to New – Dataflow Gen2.
  - Use the Power Query window to transform the data and add new columns.
- To add a new column in a notebook in pySpark, use the withColumn method
  - *df = spark.table("datatable")*
  - *df = df.withColumn("hello", col("puLocationID")*0+1)*
  - or
  - *from pyspark.sql.functions import **
  - *df = df.withColumn("hello", lit(1))*
    - or lit('') or lit('NA") or lit(None) for an empty column.
- To use some of these functions, you will need to execute:
- *from pyspark.sql.functions import **
- Date functions include:
  - dayofmonth(col), dayofweek(col) and dayoryear(col) – day of the month/week/year.
  - weekofyear(col), month(col), quarter(col), year(col)
  - hour(col), minute(col), second(col)
  - add_months(start_date, number_of_months)
  - date_add(start_date, days) and date_sub
  - date_truc(format, timestamp) truncates to the nearest unit in the format.
    - Format can be: ‘year’, ‘yyyy’, ‘yy’, ‘month’, ‘mon’, ‘mm’, ‘day’, ‘dd’, ‘hour’, ‘minute’, ‘second’, ‘week’, ‘quarter’
  - datediff(end, start) – number of days between the dates
  - months_between(date1, date2) – number of months between two dates.
  - last_day(date)
  - next_day(date, dayOfWeek)
    - dayOfWeek can be “Mon”, “Tue”, “Wed”, “Thu”, “Fri”, “Sat”, “Sun”
  - length(col) – number of columns
  - current_date() and current_timestamp() – time now
  - trunc(date, format) truncates the date to the "format" unit, either ‘year’, ‘yyyy’, ‘yy’, ‘month’, ‘mon’, ‘mm’.
- Math functions include:

    - abs – the absolute value,
    - signum(col) – the sign of the number (-1, 0 or 1),
    - trigonomical functions – acos, acosh, asin, asinh, atan, atanh, atan2, cos, cosh, sin, sinh, tan, tanh
    - advanced math functions
        - exp(col) – exponential
        - factorial(col)
        - log10(col)
        - radians(col)
    - Power functions
        - cbrt(col) – cube-root
        - pow(col1, col2) – power
        - sqrt(col) – square root
- Rounding functions include:
    - ceil(col) – the ceiling (rounded up)
    - floor(col) – the floor (rounded down)
    - round(col, scale) – rounds to the nearest "scale" decimal places. "scale" can be negative.
- String functions include:
    - concat(string1, string2...) combines multiple strings together.
        - Make sure you use lit(string) if you want to use literals.
    - instr(column, search_string) looks for strings in a column
        - For example: instr(string_column, 'D')
        - 1 is the first character. If not found, it returns 0.
    - lower(column) and upper(column)
    - lpad(column, len, pad) pads the left of the string to the width "len" with the "pad" string.
    - ltrim(column), rtrim and trim – removes spaces from the left, right or both sides of the column string.
    - repeat(column, number) repeats the "column" string "number" times.
- Enrich data by adding new columns or tables
- To use some of these functions, you will need to execute:
- *from pyspark.sql.functions import **
- Date functions include:
    - dayofmonth(col), dayofweek(col) and dayoryear(col)
    - weekofyear(col), month(col), quarter(col), year(col)
    - hour(col), minute(col), second(col)

    - add_months(start_date, number_of_months)
    - date_add(start_date, days) and date_sub
    - date_truc(format, timestamp)
    - datediff(end, start)
    - months_between(date1, date2)
    - last_day(date)
    - next_day(date, dayOfWeek)
    - length(col) – number of columns
    - current_date() and current_timestamp()
    - trunc(date, format)
- Math functions include:
    - abs
    - signum(col)
    - trigonomical functions – acos, acosh, asin, asinh, atan, atanh, atan2, cos, cosh, sin, sinh, tan, tanh
    - advanced math functions
        - exp(col), factorial(col), log10(col), radians(col)
    - Power functions
        - cbrt(col), pow(col1, col2), sqrt(col)
- Rounding functions include:
    - ceil(col), floor(col), round(col, scale)
- String functions include:
    - concat(string1, string2…)
    - instr(column, search_string)
    - lower(column) and upper(column)
    - lpad(column, len, pad)
    - ltrim(column), rtrim and trim
    - repeat(column, number)

Copy data

21. Choose an appropriate method for copying data from a Fabric data source to a lakehouse or warehouse

- Best ways to copy data:
    - If you are uploading a small file(s) from a local machine
        - Use a Local file upload
            - You can right-hand click on the … next to Files, and go to Upload – Upload files/folder.

        - Table names can contain alphanumeric characters and underscores up to 256 characters. No dashes or spaces are allowed.
        - Column names allow upper/lower cases, characters in other languages like Chinese, and underscores up to 32 characters.
    - You can also use the OneLake file explorer app. It integrates OneLake with Windows File Explorer.
        - You can download it from https://www.microsoft.com/en-us/download/details.aspx?id=105222
    - It adds this location into Windows Explorer, and includes a Sync column, showing the synchronization status, showing:
        - Blue cloud icon – online only,
        - Green tick – downloaded to your computer,
        - Sync pending arrows – in progress.
- If you are uploading a small amount of data, or using a specific connector (from over 200 connectors), or want to use Power Query transformations
    - Use a Dataflow
- If you have a large data source without using any data transformations
    - Use the Copy tool in a pipeline
- If you have got complex data transformations
    - Use Notebook code

22a. Copy data by using a data pipeline

- You can use the Copy data assistant:
    - In the pipeline, click on "Copy data" or go to Home or Activities – Copy data – Use copy assistant
    - Source
        - Select a data source, including sample data
        - Enter your connection settings, either using an "Existing connection" or "Create new connection".
        - Choose the specific data to be transferred (for example, file/folder).
    - Select a data destination source
        - Select a data source
        - Enter your connection settings, either using an "Existing connection" or "Create new connection".
        - Map your data to the destination.
    - Review the details, and click OK to save.
        - It will then be added to your data pipeline canvas.
        - Advanced settings will be available in the tabs.
- You can also a copy activity.

- Go to Home or Activities - Copy activity – Add to canvas
- In the general tab, you can select:
  - Name and Description, and whether it is enabled (in the Activity state),
  - Timeout – how long the activity can run. The default is 12 hours. It shown in the format D.HH:MM:SS.
  - Maximum number of retry events,
  - Number of sections between each retry attempt,
  - "Secure output/input". When this is checked, details of the activity is not logged.
- In the Source tab:
  - select an existing connection, or click on +New to create a new connection.
  - In a dialog box, you can select the data source and connection.
  - Back in the source tab, you can select more details, depending on the connection type – for example, the connection type, user query (table/query/stored procedure) or root folder, and table.
  - There are more settings in the Advanced section.
- In the Destination tab:
  - select the connection, and more details.
  - In the advanced section, you can select more settings, such as:
    - Max rows per file,
    - Table action – Append or Overwite, and
    - Max concurrent connections.
- In the Mapping tab, you can select the mapping from the source table to the destination table.
  - This allows you to map between columns which are differently named in the two sources.
  - In the Type conversion settings, you can select:
    - Allow data truncation (for example, from decimal to integer, or DatetimeOffset to Datetime),
    - Treat Boolean as number (true = 1),
    - Date and DateTime format (for example "yyyy-MM-dd HH:mm:ss.fff").
    - DateTimeOffset format (for example "yyyy-MM-dd HH:mm:ss.fff zzz").
    - TimeSpan format (for example "dd.hh:mm:ss")
    - Culture (for example, "en-us", "fr-fr")
- In the Settings tab, you can select:

        - Intelligent throughput optimization. Choose from Auto (which is dynamic based on the sources and destinations), Standard, Balanced and Maximum.
        - Degree of copy parallelism,
        - Fault tolerance – what happens if there are errors while copying.
        - Enable logging – log copied files and skipped files and rows,
        - Enable staging and Staging account connection (advanced).
- To run the data pipeline, go to Home – Run.
    - You can see the results in the Output tab.
        - You can export the results to CSV.
        - You can filter for a particular "Activity status" (for example, succeeded), hide output columns and show columns for Activity type, Run end, Activity run ID, Source and Destination.
- To schedule it, see topic 24.

22b. Copy data by using a dataflow

- See topic 16b.

22c. Copy data by using a notebook

- Continuing from topic 16c.
- To save the dataframe as a delta lake, parquet table to Tables section of the default Lakehouse
    - *df.write.mode("overwrite* *or append").format("delta").saveAsTable([DeltaTableName])*
        - Use overwrite to save it as new table, or overwrite an existing table.
        - You can also use "ignore" which ignores the write operation if the file already exists.
        - Instead of using (for example) "overwrite", you can also use (without quotation marks) SaveMode.Overwrite.
        - If you don't use anything, the default is "errorifexists" or "error", which returns an error if the file already exists.
        - Use append to add the table to an existing table.
- After saving to a table, you can then create a semantic model by going to "New semantic model" in the Lakehouse (not the notebook).
- In a warehouse, you can create a new semantic model by going to Reporting – New semantic model.
    - New objects are automatically added to its default Power BI semantic model.
    - You may wish to have another model with a more focused list of tables.

23. Add stored procedures, notebooks, and dataflows to a data pipeline

- To add a stored procedure to a data pipeline:
    - Go to Activities – Stored procedure,

        - Then click on the Stored procedure, if it isn't already selected.
    - In the Settings tab:
        - Click on "+New" to create a new connection.
        - Select the data source,
        - Enter the connection and credential details.
        - Select the "Stored procedure name".
        - To add parameters, click on "Import" to import them from the data source, or "+ New" to manually add parameter settings.
- To add a notebook to a data pipeline:
    - Go to Activities – Notebook.
        - Then click on the Notebook, if it isn't already selected.
    - In the Settings tab, select the name of the notebook.
    - Optionally, add any parameters.
    - Alternatively, in the notebook, go to Run – Add to pipeline.
        - You can choose a New or Existing pipeline.
- To add a dataflow to a data pipeline:
    - Go to Activities – Dataflow
        - Then click on the Dataflow, if it isn't already selected.
    - In the Settings tab, select the name of the Workspace and the Dataflow.

24. Schedule data pipelines

- To Schedule the data pipeline, either:
    - Go to Home – Schedule or Run – Schedule in the data pipeline, or
    - Go to the workspace, click on the ... next to the Data Pipeline, and click on Schedule.
- In the Scheduled run tab:
    - Turn "Scheduled run" to On.
    - Choose the Repeat schedule:
        - By the minute, Hourly, Daily or Weekly.
        - If you choose "Daily" or "Weekly", then you can select Times for it to run.
        - If you choose "Weekly", then you can select which day(s) of the week it should read.
            - So "Weekly" can be used to run each weekday.
        - If you choose "Hourly" or "By the minute", you can select the number of hours/minutes between the runs.
        - You can select a start and end date and a timezone.
    - Click Apply.
- To monitor data pipeline runs:

    - Go to Home – "View run history" or Run – "View run history" in the data pipeline, or
    - Go to the workspace, click on the … next to the Data Pipeline, and click on Recent runs.
- You can then see the recent runs.
- You can click on "Go to monitoring hub" to view more details.
    - You can also filter the runs.
    - If you click on a run, you will see more information.
    - To see performance details about an activity, click on it.
        - You can see more information in the Duration breakdown and advanced section.
        - You can view the activities as a Gantt chart, showing the length of bars as the duration of the activity, by clicking on "Gantt".
        - To find more details about the Input and Output in JSON format, click on the icon in those columns.
            - You can copy the details to the clipboard.
    - To run the pipeline again, click on Rerun.
        - You can rerun the entire pipeline, selected activities, or only the failed activity.
    - To make changes to your pipeline, click on "Update pipeline".

25. Schedule dataflows and notebooks

- To schedule a dataflow:
    - In the workspace, go to … next to the pipeline, and go to Schedule.
    - For the scheduling options, see topic 24 (except that you cannot schedule By the Minute or Hourly – just daily or weekly, every half an hour).
- To schedule a notebook:
    - In the notebook, go to Run - Schedule.
    - For the scheduling options, see topic 24.

Transform data

26. Implement a data cleansing process

- The process of data cleansing in Fabric often uses a medallion architecture.
    - The data is stored as Delta Lake storage in the OneLake allows for:
        - ACID transactions (Atomicity, Consistency, Isolation and Durability),
        - Faster read queries,
        - Batch and streaming workloads,
- It typically consists of three layers:
    - Bronze for raw data
        - Maintains the original data.

        - This allows the recreation of data to any state.
    - Can include:
        - files in any format, and
        - tables in Spark (CSV, Parquet or Delta),
        - data from databases, business apps, sensors and IoT
    - Data is in the source format, or delta tables or Parquet format if the data comes from a database.
    - If data comes from OneLake, Azure Data Lake Store Gen2 (ADLS Gen2), Amazon S3, or Google, consider using shortcuts (see topic 17).
    - Data usually gets added to.
    - Can be made of streaming or batch transactions.
  - Silver for validated data
    - Data cleansed, deduplicated and standardized,
    - Structured as tables, with rows and columns,
    - Optionally, integrated with other data.
    - Data is typically stored in Delta tables in Delta Lake format.
        - This is because of its extra capabilities and performance enhancements.
        - It allows for fast reads by Fabric compute, such as Power BI, SQL and Spark.
  - Gold for enriched data
    - To meet specific business and analytics requirements
    - Can be used in Service Level Agreements (SLAs)
    - Data will be aggregated, joined, and filtered as necessary, reducing latency.
    - Data transformed into knowledge, not information.
    - Often stored in a separate storage container to allow for better input and output and reduce deadlocks on data requests.
- The layers can use lakehouses, data warehouses, or a combination.
  - Often, bronze and silver zones are lakehouses, and gold zones are warehouses.
  - Business users therefore just access the gold zone using the data warehouse endpoint or a semantic model.
- You can use notebooks, Spark jobs, and dataflows to prepare and transform the data, going from one layer to another.
- You can use different security requirements with different security groups allowed for each stage.
- You can create semantic models from both warehouses and lakehouses.

27a. Implement a star schema for a lakehouse or warehouse

- Included in the PL-300 exam.

27b. Implement Type 1 and Type 2 slowly changing dimensions

- SCD Type 0
    - attributes do not change, or are "Original" values.
- SCD Type 1
    - attributes always change to the last version.
    - Historical data is not traced.
- SCD Type 2
    - changes to the dimension result:
        - original and latest information between retained in additional rows.
        - Additional Start Date and End Date columns may be added to show when the data is valid.
        - An additional Current column may be added to show the latest data.
    - Both the dimension and fact table may have an additional surrogate key, which links to the correct data in the dimension.
- Ideally, it will be implemented as close to the source as possible.

28a. Implement bridge tables for a lakehouse

- To create a bridge table, create a dataframe/table which includes all the items from both tables.
- Then use this table to join to the other tables.
- For more, see topic 41.

28b. Implement bridge tables for a warehouse

- -- Import Target data
- DROP TABLE IF EXISTS tblTarget;
- CREATE TABLE tblTarget (
- Country VARCHAR(20),
- Type VARCHAR(20),
- Target INT
-);
- INSERT INTO tblTarget (Country, Type, Target) VALUES
- ('England', 'In Store', 10000),
- ('England', 'Internet/Post', 5000),
- ('France', 'In Store', 7500),
- ('France', 'Internet/Post', 3000),
- ('Germany', 'In Store', 8000),

- ('Germany', 'Internet/Post', 4000);
-
- -- Import Actual data
- DROP TABLE IF EXISTS tblActual;
- CREATE TABLE tblActual (
- Country VARCHAR(20),
- Location VARCHAR(20),
- Actual INT
-);
-
- INSERT INTO tblActual (Country, Location, Actual) VALUES
- ('England', 'London', 5000),
- ('England', 'Birmingham', 7000),
- ('England', 'Manchester', 11000),
- ('France', 'Paris', 4000),
- ('Italy', 'Milan', 3000),
- ('Italy', 'Rome', 13000);
-
- -- Combine the three tables and perform aggregation
- WITH c AS
- (SELECT Country FROM tblActual
- UNION
- SELECT Country FROM tblTarget),
- a AS
- (SELECT Country, SUM(Actual) as Actual FROM tblActual GROUP BY Country),
- t AS
- (SELECT Country, SUM(Target) as [Target] FROM tblTarget GROUP BY Country)
- SELECT c.Country, SUM(a.Actual) AS TotalActual, SUM(t.Target) AS TotalTarget
- FROM c
- LEFT JOIN a ON c.Country = a.Country
- LEFT JOIN t ON c.Country = t.Country
- GROUP BY c.Country
- ORDER BY c.Country;
- Normal forms
- 1st Normal Form (1NF)

    - Requirements
        - The Values in each column must be atomic (indivisible),
        - Each value contains only a single value.
    - Actions
        - Eliminate repeating groups in individual tables
        - Create a separate table for each set of related data
        - Identify each set of related data with a primary key
- 2nd Normal Form (2NF)
    - Requirements
        - Is in 1st Normal Form
        - Reduce repeating information
    - Actions
        - Create separate tables for values that apply to multiple records.
        - Relate tables with a foreign key
- 3rd Normal Form (3NF)
    - Requirements
        - Is in 2nd Normal Form
        - Values that are not part of a record's key are to be removed from the table.
    - Action
        - Remove fields that are not dependent on the key

29. Denormalize data

- Third normal form data is efficient for writing data.
- However, for reading data (such as in a data warehouse) first normal data is better.
    - It has fewer joins.
    - It has more columns, but these can be compressed.
    - It helps create a star schema, as opposed to a snowflake schema.
- # Load DimProductCategory
- dfCat = spark.read.format("csv").option("header","true").load("Files/DimProductCategory.csv")
- display(dfCat)
- # Load DimProduct
- dfProd = spark.read.format("csv").option("header","true").load("Files/DimProduct.csv")
- display(dfProd)
- # Load DimProductSubcategory

- dfSubc = spark.read.format("csv").option("header","true").load("Files/DimProductSubcategory.csv")
- display(dfSubc)
- # Join three tables together.
- joined_df = dfProd.join(dfSubc, dfProd.ProductSubcategoryKey == dfSubc.ProductSubcategoryKey, "left")
- joined_df = joined_df.join(dfCat, "ProductCategoryKey", "left")
- display(joined_df)
- # Select final columns.
- selected_columns = ["ProductKey", "EnglishProductName", "EnglishProductSubcategoryName", "EnglishProductCategoryName"]
- final_df = joined_df.select(selected_columns)
- display(final_df)

30. Aggregate or de-aggregate data

- To aggregate data in PySpark, you can use groupBy:
- *df.groupBy("passengerCount").avg("tripDistance").show()*
    - or
- *df.groupBy(df.passengerCount).avg("tripDistance").show()*
    - or
- *df.groupBy(["passengerCount", df.vendorID]).avg("tripDistance").show()*
- *or*
- *df.select("passengerCount", "tripDistance").groupBy("passengerCount").sum("tripDistance").show()*
- You can also use the aggregate functions:
    - avg, count (and countDistinct and approx_count_distinct), first, last, max, mean, min, stddev, sum, sumDistinct and variance
- To aggregate data in SQL, you can use GROUP BY:
- *SELECT passengerCount, SUM(tripDistance) as SumTripDistance*
- *FROM TableName*
- *GROUP BY passengerCount*
- You can also use the following aggregate functions:
    - AVG, COUNT, COUNT_BIG, MAX, MIN, STDEV, SUM and VAR.
- You can also merge data using a Dataflow Gen2 (in the Power Query environment) by using Home – Group By, just as in Power BI.

31a. Merge data

- To merge dataframes in PySpark, you can use

- *df.union(df2)*
- OR
- *df.unionAll(df2)*
    - They both do the same thing. If you want to de-duplicate the rows, you also need to use distinct()
    - The names do not need the same – it merges by position.
- If you want to merge by columns, you should use:
- *df.unionByName(df2, allowMissingColumns = True).show()*
    - allowMissingColumns allows for columns not to be present in one of the dataframes.
- To merge data in SQL, you can use:
- *SELECT **
- *FROM firstTable*
- *UNION [ALL]*
- *SELECT **
- *FROM secondTable*
    - The names do not need the same – it merges by position.
- You can also merge data using a Dataflow Gen2 (in the Power Query environment) by using Home – Append Queries, just as in Power BI.

31b. Join data

- To join data in PySpark, you can use
    - df.join(df2, df.column == df2.column, 'outer') or
    - df.join(df2, 'column', 'outer')
    - The first argument is the second dataframe to join.
    - The second argument is the join column(s).
        - If using multiple columns, then you can use ['column1', 'column2']
    - The third argument is how the join happens:
        - If not included, it is an 'inner'.
        - inner = the same value must be there in both columns
        - left / leftouter / left_outer = all rows from the first dataframe, and all those in the second which matches.
        - right / rightouter / right_outer = all rows from the second dataframe, and all those in the second which matches.
        - Can also use cross; outer; full/fullouter/full_outer; semi/leftsemi/left_semi; anti/leftanti/left_anti
- To join data in SQL, you can use:
- SELECT *

- FROM firstTable
- INNER/LEFT/RIGHT JOIN secondTable
- ON firstTable.column = secondTable.column
- You can also join data using a Dataflow Gen2 (in the Power Query environment) by using Home – Merge Queries, just as in Power BI.

32a. Identify and resolve duplicate data

- To identify duplicate data:
    - in PySpark, you can groupBy the data, count it, then apply a filter where the count is greater than 1.
    - In SQL, you can GROUP BY the data, then use a HAVING COUNT(*)>1
- In PySpark, to remove duplicate data from a dataframe, use:
- *df.distinct().show()*
- Alternatively, you can use dropDuplicates or drop_duplicates, if you only want to consider certain columns:
- *df.dropDuplicates(["vendorID", "passengerCount"]).select("vendorID", "passengerCount").show()*
- You can also join data using a Dataflow Gen2 (in the Power Query environment) by using Home – Remove Rows – Remove Duplicates, just as in Power BI.

32b, c. Identify and resolve missing data and null values

- To look for missing data in PySpark, you can use:
    - column_name.isNull() or column_name.isNotNull()
    - For example: df.where(df.pickupLongitude.isNotNull()).show()
- You can then use replace:
    - Replacing value in all columns or a specific column.
        - df.replace(10, 20) or df.replace('Alice', None, 'Name')
    - If you want to replace nulls, then you can use fillna (or na.fill)
        - df.select("pickupLongitude").fillna({"pickupLongitude": 1}).show()
- If you want to use Spark SQL, then you can use IS NULL or IS NOT NULL in the Where clause.
- If you want to fill in Nulls, then you can use the ifnull or coalesce function.
    - For example: ifnull(field, value)
    - coalesce(column1, column2...) returns the first non-null column.

33. Convert data types by using SQL or PySpark

- In PySpark, you can use cast to change a column to a dataType. For example:
    - df.tripDistance.cast("string")
- You can convert dates to strings by using:
    - date_format(date (such as column_name), format)

        - "format" could be "MM/dd/yyyy". It uses:
            - yy or yyyy (not capitalized) – year
            - Q – quarter of year
            - d – day of month
            - E – day of week ("Tue" or "Tuesday")
            - D – day of year
            - M (capital – otherwise, it would be minute) or L – month of year
                - M is the "standard" form and L the "stand-alone" form, which may be different is some languages (for example, Russian)
                - M or L = 1 or 12
                - MM or LL = 01 or 12
                - MMM or LLL = Jan
                - MMMM or LLLL = January
            - h – hour of day (1 to 12)
            - H – hour of day (0 to 23)
            - K – hour of day (0 to 11)
            - k – hour of day (1 to 24)
            - m – minute of hour
            - s – second of minute
            - S (1 to 9 characters) – fractional second
            - a – am or pm
            - VV – time-zone ID (America/Los_Angeles; Z; -08:30)
            - z (1 to 3) – time-zone name (Pacific Standard Time; PST)
            - O (1, 2 or 4) – offset ("GMT+8" or "GMT+08:00" or "UTC-08:00")
            - X (1 to 5) – zone-offset (Z; -08; -0830; -08:30; -083015; -08:30:15)
            - x (1 to 5) – zone-offset (+0000; -08; -0830; -08:30; -083015; -08:30:15;)
            - Z (1 to 5) – zone-offset (+0000; -0800; -08:00;)
            - ' – escape for text
            - " – string literal
- You can convert numbers to strings by using
    - format_number(number,decimal_places)
        - converts to a string using a number of decimal places from 0 upwards.
- In Spark SQL and the SQL Analytics endpoint, you can use:
    - cast(column_name as Date)

- In the SQL Analytics endpoint only (not in Spark SQL), you can also use:
    - convert(Date, column_name)
- "format" uses:
    - yy or yyyy (not capitalized) – year
    - Q – quarter of year
    - d – day of month
    - E – day of week ("Tue" or "Tuesday")
    - D – day of year
    - M (capital – otherwise, it would be minute) or L – month of year
        - M is the "standard" form and L the "stand-alone" form, which may be different is some languages (for example, Russian)
        - M or L = 1 or 12
        - MM or LL = 01 or 12
        - MMM or LLL = Jan
        - MMMM or LLLL = January
    - h – hour of day (1 to 12)
    - H – hour of day (0 to 23)
    - K – hour of day (0 to 11)
    - k – hour of day (1 to 24)
    - m – minute of hour
    - s – second of minute
    - S (1 to 9 characters) – fractional second
    - a – am or pm
    - VV – time-zone ID (America/Los_Angeles; Z; -08:30)
    - z (1 to 3) – time-zone name (Pacific Standard Time; PST)
    - O (1, 2 or 4) – offset ("GMT+8" or "GMT+08:00" or "UTC-08:00")
    - X (1 to 5) – zone-offset (Z; -08; -0830; -08:30; -083015; -08:30:15)
    - x (1 to 5) – zone-offset (+0000; -08; -0830; -08:30; -083015; -08:30:15;)
    - Z (1 to 5) – zone-offset (+0000; -0800; -08:00;)
    - ' – escape for text
    - " – string literal
- The data types are:
    - tinyint: -128 to 127
    - smallint: -32,768 to +32,767
    - int: -2,147,483,648 to 2,147,483,647
    - bigint: -9223372036854775808, 9223372036854775807

- decimal or decimal(p, s) (or numeric for SQL)
  - p = precision – number of digits
  - s = scale – number of decimal places
  - The default is (10, 0)
- float – single precision floats
- double for PySpark (or float for SQL) – double precision floats
- string (for PySpark), char(n) and varchar(n)
  - strings, with a maximum length of "n".
  - varchar allows for a variable length.
- bool (for PySpark)
- timestamp (for PySpark) and datetime2 (for SQL) – date and time
- date (and time for SQL)

- The data types are:

Description	PySpark	SQL
tinyint	-128 to 127	0 to 255
-32,768 to +32,767	smallint	smallint
-2,147,483,648 to 2,147,483,647	int	int
-9223372036854775808 to 9223372036854775807	bigint	bigint
decimal or decimal(p, s)		or numeric
floating numbers	float and double	float and real
strings	string, char(n) and varchar(n)	char(n) and varchar(n)
boolean	bool	bit
date and time	timestamp	datetime2
date	date	date
time		time

34. Filter data

- WHERE - df.where() and df.filter()
  - reduces the number of rows.
  - df.filter("age = 2") or df.where("age > 2")
  - Conditions are:
    - > and < are greater than and less than.

        - >= and <= include equal to.
        - = or == is equal to
        - != or <> is not equal to
- You can use the following in the where or filter:
    - df.column_name.between(number1, number2)
        - Where the value is between number1 and number2.
    - df.column_name.contains('string'), endswith and startswith
        - Where the column_name contains, ends with or starts with 'string'. It is case sensitive.
    - df.column_name.like('string')
        - Where the column_name is like the string.
        - You can use % for zero, one or many characters.
        - You can use _ for one character only.
    - df.column_name.isin('string', number, 'string')
        - It evaluates true if the column_name is any of the values in the brackets/parentheses.
    - df.column_name.substr(startPos, length)
        - This extracts part of a string – the equivalent to MID in Excel. Don't use substring in PySpark (but you can use it in SQL).

Optimize performance

35a. Identify and resolve data loading performance bottlenecks in dataflows

- When dataflows load data, they can use a staging table.
    - This allows it to be loaded before transformations are done. It may or may not improve overall performance.
    - You can toggle it on and off by right-hand clicking on the query and select "Enable staging".
    - You can also create separate dataflows for loading and then for transformation.
- To check the performance of a dataflow:
    - Go to the Workspace,
    - Next to the dataflow, click on ... and select Refresh History.
    - The runs are shown, together with their duration.
    - Click on a run. This shows the activities, with:
        - The start and end date/time, and
        - The duration.
    - Click on an activity, and you can see:
        - The start and end date/time,
        - The duration, and

        - The Volume processed.
- You can run the dataflow multiple times to compare timings.
- You can check the Fabric capacity performance by going to the Compute tab in the Microsoft Fabric Capacity Metrics app (see topic 9).

35b. Identify and resolve data loading performance bottlenecks in notebooks

- You can store your data in partitions (see topic 18).
- Additionally, to speed up loading in notebooks, you can use a high concurrency mode.
    - You can attach a notebook to an existing Spark session.
    - There is no need for a notebook to start its own Spark session.
    - High Concurrency notebooks are:
        - run by the same user,
        - have the same default lakehouse,
        - have the same Spark compute configurations, and
        - have the same library packages.
    - If you need more dedicated compute, you can use a standard session.
- To allow you to use high concurrency mode in any notebook:
    - going to a workspace,
    - Click on "Workspace settings".
    - Expand the Data Engineering/Science and click on "Spark settings".
    - In the High concurrency tab, you can switch to On the "For notebooks".
- You can do this for an individual notebook by going to Home – Connect – New high concurrency session.

35c. Identify and resolve data loading performance bottlenecks in SQL queries

- Query folding has been covered in the PL-300 exam.
- Avoid SELECT * and get only those columns you need.
- USE LIMIT (or TOP) to return only a few rows.
- Reduce data types.
    - It takes storage to store longer data types that are needed, and network and compute to process them.
    - Don't use CHAR(20) when VARCHAR(20) would work.
        - Both allow for 20 characters.
        - However, CHAR always takes 20 characters. If the majority of the data is less than 20 character, then VARCHAR reduces the amount of size that is needed.
    - Use the smallest number type.
        - Don't use bigint where smallint would store the data.

36a. Implement performance improvements in dataflows

- You can break complicated dataflows into multiple dataflows.
    - It can make it easier to understand, and to reuse.
        - It can also reduce timeout errors.
    - You can have separate dataflows work on different tables, and sequence them using a pipeline.
    - Or you have sequential dataflows work on the same table.
    - You can split the ingesting of data (staging dataflows) from those which transform data.
        - It can reduce the number of read operations from the source, and reduce the requirements for the data gateway.
        - It can also be useful to have a copy of the data that was on the source, in case something changes on the source.
        - It also allows the transformation dataflow to be completely independent on the source.
    - You should separate dataflows which have different refresh schedules.
- If you have more complicated transformations that are used in more than one data source, you may be able to use a "Power Query"-type function.

36b. Implement performance improvements in notebooks

- V-Order
    - The Delta Lake table format can be optimized using V-Order. This enables fast reads for Power BI, SQL and Spark.
    - Microsoft says that read times can be between 10% and 50% faster.
    - It applies sorting, row group distribution, dictionary encoding and compression on Parquet files.
        - This reduces disk space. Therefore, it needs less network and CPU resources to read it.
        - It also decreases write speed. Microsoft says by around 15%.
    - To check the status of V-Order in Apache Spark, or to enable it, use:
        - spark.conf.get('spark.sql.parquet.vorder.enabled')
        - spark.conf.set('spark.sql.parquet.vorder.enabled', 'true')
    - To check the status of V-Order in SQL
        - SET spark.sql.parquet.vorder.enabled
    - To enable it in SQL, use
        - SET spark.sql.parquet.vorder.enabled=TRUE
        - or
        - CREATE TABLE ... USING parquet TBLPROPERTIES("delta.parquet.vorder.enabled" = "true");
- Optimize Write

- This aims to increase individual file size to between 128 Mb and 1Gb, and is enabled by default in Microsoft Fabric.
- To set it in Apache Spark, use:
  - spark.conf.set("spark.microsoft.delta.optimizeWrite.enabled", "true")
- In Spark SQL, use:
  - SET 'spark.microsoft.delta.optimizeWrite.enabled'

36c. Implement performance improvements in SQL queries

- Separate date and times, and any strings that can be separated.
  - This would reduce cardinality – the number of different variations.
  - Lower cardinality allows for better compression and storage, once dictionary and compression has been employed.
- You can ask your database administrator whether your queries can be sped up – for example, with indexes.
  - Indexes should be used where a search is going to be found:
    - In the WHERE clause, for searching,
    - In the JOIN clause, where matches need to be made between tables,
    - In the GROUP BY clause, for aggregating, and
    - In the ORDER BY clause, so SQL has an index of the values for the appropriate fields.
  - Use SARGable conditions (SARG meaning Search ARGument ABLE). For example, use:
    - =, >, <, >=, <=, BETWEEN, LIKE, IS NULL, IS NOT NULL, IN. These are able to make use of indexes.
    - For LIKE, LIKE 'Hello%' can make use of indexes. LIKE '%Hello' cannot.
    - For dates, BETWEEN '2026-01-01' and '2026-12-31 23:59:59' can make use of indexes. Year(myField) cannot.
    - Basically, avoid functions if you can write the expression a different way using SARG and use an index.

37. Identify and resolve issues with Delta table file sizes

- Delta table has higher performance when there are a small number of large files, not a large number of small files.
- For better query performance, data files should be approximately 128 Mb-1 Gb in size.
- To performance optimization, then click on the … next to a table in a lakehouse, and click on Maintenance.
  - Delta Lake identifies tables which should be optimized, and queues them to be optimized.
  - It combines multiple smaller files into larger files.
  - It does not impact on data readers and writers.
  - It can perform:

- OPTIMIZE – it optimizes file size.
- You can also s
- VACUUM – Delta Lake keeps a history of all changes made over time. VACUUM deletes data files not referenced by the Delta table version for several days.
    - By default, it is for the last 7 days.

Design and build semantic models

38. Choose a storage mode, including Direct Lake

- Import and DirectQuery modes were covered in the PL-300 exam.
- Import caches the data, so is often used for smaller amounts of data.
    - It requires time to import, but is swift once imported to generate results.
- DirectQuery retrieves data as and when needed, and will always have the latest data.
    - It requires no time to import, as no importing is required. However, it may take time to retrieve data.
- Composite mode (using Dual mode) is a bridge between Import and DirectQuery modes.
- Direct Lake uses the advantage of data being stored in the OneLake:
    - to give fast performance (similar to import mode),
    - but with the freshest data (like DirectQuery mode).
- It needs a lakehouse or warehouse on a Microsoft Fabric capacity.
    - This lakehouse/warehouse then uses OneLake to store the data.
    - Only tables in the semantic models derived from tables (not views) in the Lakehouse/Warehouse can use Direct Lake mode.
    - You cannot use both Direct Lake tables and other table modes (Import, DirectQuery, Dual).
    - Calculated columns and tables are not supported.
    - It supports write operations using the XMLA endpoint in the latest versions of SSMS, Tabular Editor and DAX Studio.

39. Identify use cases for DAX Studio and Tabular Editor 2

- Tabular Editor - https://tabulareditor.com/
    - Build, maintain, and manage models.
    - Shows your objects, and allows for editing properties for multiple objects at the same time.
    - Include DAX syntax highlighting.
- DAX Studio - https://daxstudio.org/
    - For DAX authoring, diagnosis, performance tuning and analysis.
    - Allows for object browsing, integrated tracing, query execution statistics, DAX syntax highlighting and formatting.
- ALM Toolkit - http://alm-toolkit.com/

  - Used for Application Lifecycle Management (ALM),
  - Perform deployment across environments, while retaining previously imported incremental refresh data.
  - Merge metadata files, branches and repos.
  - Reuse common definitions between semantic models.
- Metadata translator - https://github.com/microsoft/Analysis-Services/tree/master/MetadataTranslator
  - Translate using Azure Cognitive Services, or via .csv files using Excel.
  - Translate captions, descriptions and folder names, for tables, columns, measures and hierarchies.
- When tools are downloaded, they appear in the External Tools ribbon.

40. Implement a star schema for a semantic model

- Covered in the PL-300 exam.
- 41. Implement relationships, such as bridge tables and many-to-many relationships
- Bridge tables can resolve many-to-many relationships, by making them one-to-many relationships.

42a. Write calculations that use DAX variables

- Example:
  - SalesAmountCalculation = (CALCULATE(SUM(FactInternetSales[SalesAmount]),DATESMTD(FactInternetSales[DueDate])) - SUM(FactInternetSales[SalesAmount])) / CALCULATE(SUM(FactInternetSales[SalesAmount]),DATESMTD(FactInternetSales[DueDate]))
  - This formula calculates the MTD SalesAmount, and calculates the percentage for the month already gone excluding the current day over the current month.
- To use a DAX variable, enter, after the name of the measure:
  - VAR NameOfVariable =
  - and then at the end, the answer is
  - RETURN Calculation
- DAX variables allow you to:
  - Improve performance.
    - If you are referring to the same expression twice, it needs to calculate (evaluate) it twice.
    - If you put the expression in a variable, it only evaluates it once.
      - SalesAmountCalculation =
      - VAR MonthToDate = CALCULATE(SUM(FactInternetSales[SalesAmount]),DATESMTD(FactInternetSales[DueDate]))
      - RETURN

- (MonthToDate - SUM(FactInternetSales[SalesAmount])) / MonthToDate

- Improve readability.
  - Variable names can be shorter than the calculation.
- Allow for debugging.
  - You can RETURN the variable instead of the actual answer to debug.
  - You do this by commenting out ('--") the answer, and return the variable.
    - SalesAmountCalculation =
    - VAR MonthToDate = CALCULATE(SUM(FactInternetSales[SalesAmount]),DATESMTD(FactInternetSales[DueDate]))
    - VAR CurrentSales = SUM(FactInternetSales[SalesAmount])
    - RETURN
    - (MonthToDate - CurrentSales) / MonthToDate
- Reduce complexity.
  - Variables are calculated outside of the filter contexts.
  - This may mean that you don't need to use the EARLIER or EARLIEST DAX functions.

42d. Write calculations that use DAX windowing functions

- ROWNUMBER
  - You can use this as a measure. This goes through the entire table.
- RowNumberColumn = ROWNUMBER(ORDERBY(DimProduct[EnglishProductSubcategoryName]))
  - You can order it ASCending or DESCending, and you can also order by multiple columns.
- RowNumberColumn = ROWNUMBER(ORDERBY(DimProduct[EnglishProductSubcategoryName],ASC,DimProduct[EnglishProductCategoryName],0))
  - You can also use it at a particular granularity. The below formula uses only the DimProductSubcategoryName to create the row numbers.
- RowNumberColumn = ROWNUMBER(ALLSELECTED(DimProductSubcategory),ORDERBY(DimProductSubcategory[EnglishProductSubcategoryName], ASC))
  - The blanks are shown at the beginning, because ASC is being used. (This is DEFAULT.) You can change it to the end by using:
    - ASC BLANKS LAST
    - DESC BLANKS FIRST
  - or DESC.
  - You can also use DEFAULT, FIRST or LAST in the blanks argument:

- RowNumberColumn = ROWNUMBER(ALLSELECTED(DimProductSubcategory),ORDERBY(DimProductSubcategory[EnglishProductSubcategoryName], ASC),LAST)
    - To reset every ProductCategoryKey, you can use the partitionBy argument.
        - RowNumberColumn = ROWNUMBER(ALLSELECTED(DimProductSubcategory),ORDERBY(DimProductSubcategory[EnglishProductSubcategoryName], ASC),LAST, PARTITIONBY(DimProductSubcategory[ProductCategoryKey]))
- RANK
    - This needs an additional argument at the front – what to do for ties. Suppose there is a 3-way tie for rows 2-4. The results would then be:
        - ROWNUMBER – 1, 2, 3, 4, 5
        - SKIP – 1, 2, 2, 2, 5
        - DENSE – 1, 2, 2, 2, 3
    - Examples:
- RankNumberColumn = RANK(SKIP,
- ALLSELECTED(DimProduct[EnglishProductSubcategoryName],DimProduct[EnglishProductCategoryName]),
- ORDERBY(DimProduct[EnglishProductCategoryName],ASC))
- 42d. Write calculations that use DAX windowing functions
- For the examples:
    - Create table: SummaryTable = SUMMARIZECOLUMNS(DimDate[CalendarYear], DimProduct[EnglishProductCategoryName], "SalesAmount", SUM(FactInternetSales[SalesAmount]))
- INDEX
    - Returns the item in the 1st/2nd etc. position.
    - Examples:
- IndexColumn = INDEX(1, ALL(SummaryTable[CalendarYear]))
- IndexCalculation = INDEX(2, ALL(SummaryTable[CalendarYear]), ORDERBY(SummaryTable[CalendarYear], DESC))
- OFFSET
    - Returns the number of rows before/after the current row.
    - Example - Create measure:
- PreviousYearSales = CALCULATE(SUM(SummaryTable[SalesAmount]), OFFSET(-1, , ORDERBY(SummaryTable[CalendarYear])))
- WINDOW
    - Creates a series of rows which is based on an absolute or relative position.
    - Example:
- RunningSum = SUMX(

- WINDOW (-1, REL, 0, REL,
    - ALLSELECTED(SummaryTable[CalendarYear], SummaryTable[EnglishProductCategoryName])),
- SUM([SalesAmount]))
    - REL = relative, ABS = absolute

43a. Implement calculation groups

- Calculation groups allow you to create similar calculations for multiple measures.
    - Often used for time intelligence calculations – MTD, QTD, YTD, PY
- The calculation groups contains calculation items – the individual calculations on a generic measures.
    - In place of a measure, you would use SELECTEDMEASURE() instead.
    - If you need the name of the measure, you can use SELECTEDMEASURENAME().
- If you are using calculation groups, implicit measures are disabled.
    - You can no longer drag "SalesMeasure" onto a visual to automatically get SumOfSalesMeasure – it needs to be separated created.
    - You can't use OLS/RLS on calculation group tables.
- Creating calculation groups in Power BI Desktop is currently in preview. To switch it on:
    - Go to File – Options and Settings – Options.
    - Then go to Preview features and check "Model explorer and Calculation group authoring".
- To add a calculation group:
    - Go to the Model view.
    - In the Home tab, go to the new "Calculation group" button.
        - It will then warn you that implicit measures will be disabled.
    - It will then add into the formula pane:
        - Calculation item = SELECTEDMEASURE()
    - It will create a table called "calculation group" – you can rename it.
    - It will also create a column called "Calculation group column" – you can rename it, and an Ordinal column, which is ineffective in the Power BI Desktop.
- 43a. Implement calculation groups
- To view the calculations within a calculation group, in the Data pane of the Model view, click on the new Model tab.
    - You can then expand the relevant calculation groups and see the calculation items.
- To add a new calculation group, click on the "Calculation items" in the Data pane of the Model view, then click on "+ New calculation item" in the Properties pane.
    - Or right-hand click on the calculation group and select "+ New calculation item".

- You can reorder the calculation groups items by clicking on the calculation group, and then reordering them in the Properties pane.
    - There is also an Ordinal column in the Calculation Group table.
- You can also change Precedence of a calculation group.
    - This determine which calculation group is to calculation first, if you have multiple groups.
        - Lower precedence calculation groups are applied first.
    - However, this is very advanced.
- You can create a Matrix with:
    - Dates (Year, Quarter and Month) in the Rows,
    - A Measure in the Values, and
    - The calculation group column in the columns.
- You can also add a slicer with the calculation group column, to allow the user the choice of calculations.
- Examples of calculation items:
- MTD = CALCULATE(SELECTEDMEASURE(), DATESMTD(DimDate[Date]))
- QTD = CALCULATE(SELECTEDMEASURE(), DATESQTD(DimDate[Date]))
- YTD = CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
- PY = CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR(DimDate[Date]))
- Example of using a Time Calculation function in another measure:
- SalesAmountPYMTD = CALCULATE(SUM(FactInternetSales[SalesAmount]), SAMEPERIODLASTYEAR(DimDate[Date]),'Time Intelligence'[Time Calculation] = "MTD")
- Other examples of calculation items:
- PY MTD = CALCULATE(SELECTEDMEASURE(),SAMEPERIODLASTYEAR(DimDate[Date]),'Time Intelligence'[Time Calculation] = "MTD")
- PY QTD = CALCULATE(SELECTEDMEASURE(),SAMEPERIODLASTYEAR(DimDate[Date]),'Time Intelligence'[Time Calculation] = “QTD")
- PY YTD = CALCULATE(SELECTEDMEASURE(),SAMEPERIODLASTYEAR(DimDate[Date]),'Time Intelligence'[Time Calculation] = “YTD")
- YOY = SELECTEDMEASURE() - CALCULATE(SELECTEDMEASURE(), 'Time Intelligence'[Time Calculation] = "PY")
- This last example uses “Sideways recursion”.
- YOY% = DIVIDE(CALCULATE(SELECTEDMEASURE(),'Time Intelligence'[Time Calculation]="YOY"),
    - CALCULATE(SELECTEDMEASURE(),'Time Intelligence'[Time Calculation]="PY"))

- Other functions:
    - SELECTEDMEASURENAME
        - Returns the name of the selected measure
    - ISSELECTEDMEASURE(Measure1, Measure2…)
        - Whether the selected measure is in the list in brackets/parentheses.
        - Used if a measure needs to be differently calculated for different measures.
    - SELECTEDMEASUREFORMATSTRING
        - The format of the SELECTEDMEASURE

43b. Implement dynamic strings

- Dynamic strings in Calculation items
    - The YOY% calculation is showing as 0.04, not 4%.
    - If you click on the YOY% calculation in the Model view, in the Properties pane – Formatting:
        - you can switch the “Dynamic format string” to Yes.
        - Click on Edit.
        - Enter: “0.00%”
- Dynamic strings for measures
    - You can change the formatting for explicit measures, such as Sum(Transactions), based on other fields.
        - You cannot use it for fields themselves (implicit measures).
    - This is a preview feature. To switch it on:
        - Go to File – Options and Settings – Options.
        - Then go to Preview features and check “Model explorer and Calculation group authoring”.
    - Setup:
        - MtoMPeople and MtoMTransactions
        - Create new measure: SumOfTransaction = SUM(MtoMTransactions[Transaction])
        - Create a Table with Owner, BankAccount, SumOfTransactions and Currency.
    - Click on the measure, and go to the Measure Tools menu.
    - Change the Format to the new “Dynamic” option.
    - There is a new drop-down next to the formula bar, containing Measure and Format.
    - Enter a formula for the Format – for example:
- if(MIN(MtoMTransactions[Currency]) <> MAX(MtoMTransactions[Currency]),"Multiple",
- if(MIN(MtoMTransactions[Currency])="USD","$#,##0",

- if(MIN(MtoMTransactions[Currency])="EUR","€#,##0",
- if(MIN(MtoMTransactions[Currency])="GBP","£#,##0",
- MIN(MtoMTransactions[Currency]) & " #,##0"))))
    - 43c. Implement field parameters
- Field parameters allow the end user to change which measure or dimension is being used in your visuals.
- Field parameters are currently in preview. To switch it on:
    - Go to File – Options and Settings – Options.
    - Then go to Preview features and check "Field parameters".
- To create a field parameter, go to Modeling – New parmeter – Fields.
- Select a name for the Parameter, and add and reorder fields for that parameter.
    - You can also check "Add slicer to this page".
        - There is no option to use no fields. This is equivalent to selecting all the fields.
    - You can include measures or dimensions.
- You can then go to your data, and go to the new table, and drag the new parameter into your visual.
- When added, you can right-hand click on the parameter and check "Show selected field".
- To edit a parameter, you need to edit the DAX formula by clicking on the parameter in the Data pane.
    - The last argument of the NAMEOF function shows the order of the parameters.
- Note:
    - You cannot use AI visuals or Q&A.
    - You cannot use this using DirectQuery unless you are using a composite model, with a local model for field parameters.
    - You cannot use implicit measures – they must be explicit.
    - You cannot use field parameters as the link for a drill-through or tooltip page.

44. Design and build a large format semantic models

- If you have a premium capacity, you can use Large semantic models.
    - This is only for semantic models in the Power BI Service.
    - It does not affect Power BI Desktop, which is limited to 10 Gb.
- You will need to have:
    - a Premium P SKU,
    - Embedded A SKUs, or
    - Premium Per User (PPU).
    - It is not available for US Government Department of Defence customers.
- It allows for:

    - Semantic models to grow beyond 10 Gb,
    - Improves performance for XMLA write operations.
        - The difference can be significant for semantic models over 1 Gb (after compression)
        - You can check the estimated size in SSMS using the XMLS endpoint by right-hand clicking on the database and going to Properties. You will see the "Estimated size".
    - Sets the default segment size to 8 million rows.
        - This creates a good balance for large tables between memory requirements and query performance.
    - On-demand load. If your semantic model has been "evicted" so that other models can use the memory, it:
        - retrieves relevant data pages on-demand, and
        - allows the evicted semantic model to be quickly made available for queries.
- To enable it when creating a workspace:
    - Go to the Advanced section, and click on "Large dataset storage format".
- To enable it for all future semantic models for an existing workspace:
    - Go to the workspace.
    - Go to … - Workspace Settings – License info – Edit.
    - In the Default storage format, change "Small dataset storage format" to "Large dataset storage format".
- To enable it for a particular semantic model:
    - In the workspace, click on the … next to the Semantic Model and go to Settings.
    - In "Large semantic model storage format", switch to "On".
- Please note:
    - You cannot download a large format semantic model to Power BI Desktop.
    - It is not available for Pro workspaces.
        - If you have a Premium workspace with a large format sematic model, and downgrade it to Pro, it will not load.
    - Be careful when refreshing large semantic models if the model size is near half of the capacity size, as it may exceed the capacity memory during refreshes.
        - You may need to use fine grained data refreshes using the XMLA endpoint instead.
    - Large semantic models must stay in the region it was first created it.
        - You can change its workspace, but the new workspace must have the same region as the old workspace.

45. Design and build composite models that include aggregations

- Composite models is a model which has:

    - multiple (two or more) data connections from different source groups, and
    - least one DirectQuery data connection (as opposed to import).
    - This can be useful if you have a huge fact table which should not be imported, but you have smaller dimension tables which can be imported.
- Previously, if you used DirectQuery, you could only have the one data connection.
- If you have both DirectQuery and Import tables, the status bar displays "Mixed" storage mode.
- Using composite models, you can more easily establish many-to-many relationships between tables.
- To use composite models on Power BI Service, you would need to have the following in Settings – Admin portal – Tenant settings:
    - Allow XMLA Endpoints and Analyze in Excel with on-premises semantic models
        - Otherwise you can't use a DirectQuery connection.
    - Users can work with Power BI semantic models in Excel using a live connection
        - Otherwise you can't make live connections.
    - Allow DirectQuery connection to Power BI semantic models
    - For Premium capacities or Premium Per User, you also need "XMLA endpoint" setting enabled, and set to "Read Only" or "Read/Write".
- Note about composite models:
    - relationships between different sources are called "limited" relationships (as opposed to "regular" relationships). For "limited" relationships:
        - You can only use INNER JOIN, not LEFT or RIGHT joins,
        - You cannot use the RELATED DAX function to retrieve the "one" side of the relationship.
        - It is marked on the model relationship with ( ).
    - queries to one data source may contain potentially sensitive information given from the other data source.
    - queries to one data source may include a lot of literal values, which may slow execution.
    - if the query is too large, it may need to be split into multiple queries, which again may slow execution.
- Because composite models include DirectQuery sources, which may be very large, you can add aggregations.
    - These cache data for the aggregations, improving performance.
- You need to set up a separate aggregation table, as it will be hidden.
    - You can use your DirectQuery table, and then create a GroupBy version of that in Power Query.
- To set up the aggregations:

  - right-hand click on the aggregated table (not the field or the fact table) in the Data pane, and select "manage aggregations".
    - You can also click on the … next to the aggregated table.
  - Select what each of the aggregated fields maps to.
  - The GroupBy fields are optional if they are part of a relationship. However, they can be set if you would like.
  - If you want to create an Average, then set up two separate columns for Sum and Count.
- Note:
  - The Detail Table must use DirectQuery storage mode.
  - For best performance, the aggregation table should use the Import storage mode.
  - This means that tables connecting to the two tables on the one-side probably would use the Dual storage mode.
  - The Aggregation and Detail Columns must have the same datatype, unless you are using Count or "Count table rows".
  - You cannot use multiple aggregations which use the same Summarization function, Detail Table and Detail Column.
  - It is not used if you only have read-only access. That's because you might be using RLS. Instead, the detail table is used instead.
  - It needs "regular" relationships, not "limited" relationships.

46. Implement dynamic row-level security and object-level security

- Dynamic RLS is included in the PL-300 exam.
- You can create object-level security (OLS) using Tabular Editor.
  - The "object" can be a table or column.
- First of all, create a role in the same way as using RLS.
  - Go to Modeling – Manage roles.
  - Click Create.
  - Define the roles as per RLS, but you don't need to define any RLS rules.
- Then, go to Tabular Editor.
  - If you are securing tables:
    - In the model, go to Roles and click on the relevant role.
    - In Permissions, expand Security – Table Permissions.
    - Change the "Table Permissions" from Default to None.
  - If you are securing columns:
    - Expand Tables – [Name of Table] – [Name of column].
    - In Permissions, expand Object Level Security – [Name of role].
    - Change to "None".
  - Then click Save.

    - Go back to Power BI Desktop, and publish to the Power BI Service.
    - In the Power BI Service, go to the semantic model, …, Security, and assign rules.
- Note:
    - OLS does not apply to Admin, Member or Contributors.
    - You cannot use OLS and any of the following visuals: Q&A, Quick Insights or Smart Narrative.
    - You cannot secure a table which is in the middle of a relationship.
        - For example, DimProductSubcategory links to DimProductCategory and DimProduct,
        - However, you could add an additional relationship between the two affected tables.
    - You can use OLS for columns in a table which is part of the relationship, as long as the key column is not secured.
    - You cannot use RLS using one role and OLS using another role.
        - It could lead to unexpected access to secure data.
    - Calculations (such as measures) do not work if they refer to a secure table or column.

47. Validate row-level security and object-level security

- See topic 46.

Optimize enterprise-scale semantic models

48. Implement performance improvements in queries and report visuals

ALM Toolkit

- ALM Toolkit is used for Application Lifecycle Management (ALM).
    - You can download it from http://alm-toolkit.com/
- You can compare a source and target model and find changes.
    - You can launch it from Power BI Desktop and go to External tools – ALM Toolkit.
- To view the differences, go to Home – Select Actions – Hide Skip Objects.
    - These are objects which don't need any updating.
- You can also select multiple actions, and use the context menu to:
    - Skip any updates to the selected objects,
    - Create any objects which are missing in the Target,
    - Delete any objects which are missing in the Source, and
    - Update any objects which have different definitions.
- You can also do that for all objects in the Home – Select Actions dropdown.
- Once you have selected all of your actions, you should then click on Home - Validate selection, and check any warnings.
- You can select comparison options in the Home - Options dialog.

    - One of the options is “Retain only refresh-policy based partitions”. This only keeps partitions generated by incremental-refresh policies.
- You can choose from the following process modes:
    - Process Recalc (for a semantic model only):
        - Updates/recalculates hierarchies, relationships, and calculated columns.
    - Process Default
        - Performs the processing which is necessary to get it to a fully processed data.
        - For example:
            - data for empty tables/partitions are loaded,
            - Hierarchies, calculated columns, and relationships are recalculated/build/re-built.
    - Do Not Process
    - Process Full
        - Drops all existing data, and then processes the object.
        - This is required when a structural change has been made.
        - It requires the most resources.
- You can also check “Process only affected tables”, to restrict the amount of processing required.
- You can create an Excel spreadsheet containing the comparison by clicking on Home – Report Differences.
- You can then click on Home – Update, which will do the actions.

Metadata Translator

- Metadata Translator
    - allows you to translate captions, descriptions and display folder names of tables, columns, measures and hierarchies.
    - allows you to translate in over 300 cultures in all languages that you can use in Power BI.
    - You can either translate it manually or you can use Azure Cognitive Services.
    - You can also export and import translations using .csv files.
- You can also translate using Tabular Editor.
    - 49. Improve DAX performance by using DAX Studio
- To install DAX Studio, go to https://daxstudio.org/
- You can launch it from Power BI Desktop by going to External tools – DAX Studio.
    - You can also connect to it using a Power BI Premium XMLA Endpoint by going to File – New, and entering the connection next to “Tabular Server”.
- You can enter DAX queries:

  - You can enter "EVALUATE TableName" to query all rows in the TableName.
  - To reduce to a single column, you can use: EVALUATE VALUES(TableName[ColumnName])
- You can click on "Cache: Clear on Run" to remove the previously-loaded cache before running the query.
- You can copy a DAX query from Performance Analyzer in Power BI Desktop to investigate it.
- You can enable the following traces.
  - "All Queries" trace captures all query events, showing the number of milliseconds the query takes.
  - "Query Plan" displays the plan in its raw form.
    - Use only if necessary!
  - "Server Timings" display the timings:
    - Total – this is the total time, taken from the Query End profiler event. It is made up of Storage Engine (SE) time and Formula Engine (FE) time.
      - Storage Engine retrieves data. It is multi-threaded, and so should be used over FE.
      - Formula Engine computes data. It is single-threaded. This time should be minimized.
    - SE CPU – the approximate of CPU time spent on Storage Engine queries.
    - FE – the time spent in the Formula Engine.
    - SE – the amount of time spent in the Storage Engine.
    - SE Queries – a count of the number of Storage Engine queries that were performed.
    - SE Cache – the number of times the SE cache was used.

49. Improve DAX performance by using DAX Studio

  - It also shows:
    - The estimated number of rows to be used in the query. This is useful to understand its cardinality.
    - KB – the size of the SE query. Also known as "data cache".
- If something is running slowly, try a different DAX statement and run it having cleared the cache, and see the difference the results.
- The following might generate multiple SE queries:
  - DISTINCTCOUNT,
  - Complex filters.
- IF statements are hard for the engine to optimize.
  - This includes SWITCH Statements, which are essentially nested IF statements.
- To get statistics, you can use the VertiPaq Analyzer, which is part of DAX Studio. To access it, go to Advanced – View Metrics.

    - The cardinality column contains the number of distinct values (after removing all duplicates).
    - The "Total Size" is the combined size of Data, Dictionary and Hierarchies Size.
    - The "% Table" column shows the percentage of the Total Size over the entire table.
    - The "Data Type" shows DateTime, Int64, String, Double etc.
- Best practices:
    - Reduce number of columns where possible
    - Reduce column cardinality (number of distinct values).
        - Especially do this on the 1-side of a relationship,
        - Don't use date/time unless you need to – split it into date and time to improve cardinality.
        - If you have floating point values, apply a specific precision unless you need it. For example, don't use 12.345 – use 12.3 if that is good enough.
- Traditional database storage
- VertiPaq database storage
- VertiPaq database storage
- VertiPaq database storage

50. Optimize a semantic model by using Tabular Editor 2

- Installing Tabular Editor 2
    - TE2 is the free version. TE3 has more features, but requires a subscription.
        - See https://docs.tabulareditor.com/?tabs=TE3 for more details.
        - You can create calculated columns, measures, hierarchies, perspectives, translations, display folders etc.
    - Both features use the Tabular Object Model (TOM) to show objects and properties, and loads/saves metadata to/from Model.bim files and existing databases.
    - To download it, go to https://github.com/TabularEditor/TabularEditor/releases/latest
- You can open it from Power BI Desktop by going to External Tools – Tabular Editor.
- The left-hand side shows all of the tables. Within the tables, there are columns, measures and hierarchies.
- Right-hand clicking the tree will show a context menu, including adding new measures, hiding/duplicating/deleting objects.
    - You can rename objects by clicking on F2.
    - You can also select multiple objects and rename them all using the context menu.
- In the top-right hand corner, you can use the DAX Editor to edit measures or calculated columns.
    - You can click the "Dax Formatter" button to format the code.
- In the bottom-right hand corner there is a property grid.

    - You can get/set properties such as Format String and Description.
    - You can do this for multiple objects – for example, change the format string for multiple dates.
- The Best Practice Analyzer (BPA) within Tabular Editor 2 gives you recommendations for improving your model.
- To use it:
    - Go to Tools – Manage BPA Rules.
    - Next to "Rule collections", click Add.
    - Click "Include Rule File from URL".
    - Enter the following URL:
- https://raw.githubusercontent.com/microsoft/Analysis-Services/master/BestPracticeRules/BPARules.json
    - Note: The "Severity" is only for information in Tabular Editor.
    - For the other rule collections, you can create a new/clone (duplicate)/edit/delete rule.
        - You cannot do this for the BPA, because it is read-only.
        - The other rule collections are:
            - Rules within the current model,
            - Rules for the local user, and
            - Rules on the local machine.
            - Combined, they become the "(effective rules)".
- The BPA gives suggestions for:
    - DAX Expressions,
    - Error Prevention,
    - Formatting,
    - Maintenance,
    - Naming Conventions, and
    - Performance.
- To check your model, either:
    - click the "X BP issues" at the bottom, or
    - go to Tools – Best Practice Analyse, or
    - Press F10.
- You can open an "issue" and double click on an object to go to it (or right-hand click and "Go to object...").
- You can also:
    - Ignore item,

    - Generate fix script – you can then try it in the “C# script” (previously called the “Advanced Scripting”).
    - Apply fix (don’t forget to “Save” the changes to apply them), or
    - Copy to Clipboard.
- There is also an Undo button, just in case!

51. Implement incremental refresh

- As per PL-300 exam.

Perform exploratory analytics

52. Implement descriptive and diagnostic analytics

- From PL-300 course.

53. Integrate prescriptive and predictive analytics into a visual or report

- From PL-300 course.

54. Profile data

- Profiling data in Power BI have been included in the PL-300 course.
- To create graphs, you can use a plotting library like Matplotlib. You can use:
    - import matplotlib.pyplot as plt
- You would then need to convert the dataframe to a Panda Dataframe to use it, as Matplotlib does not support Spark Dataframes.
    - dfPanda = dfSpark.toPandas()
- To create a figure, use
    - plt.figure(figsize=(8,8))
        - figsize shows the dimension in inches.
- To create a bar chart, use:
    - plt.bar(dfPanda.Country, dfPanda.ActualTotal, label="Actual", color='b')
    - The first two arguments are for the x-axis and y-axis.
    - colors are:
        - b, g and r stand for blue, green and red.
        - c, m and y stand for cyan, magenta and yellow (it is a green-ish yellow).
        - k and w stand for black and white.
        - You can also use CSS colors, such as "red", "darksalmon", "yellow". These are available at https://matplotlib.org/stable/gallery/color/named_colors.html
- To display the chart, use:
    - plt.show()
- To add labels, you can use:
    - plt.xlabel("Country")
    - plt.ylabel("Total")

- To add a title, you can use:
  - plt.title("Country and Totals")
- To create a pie chart, then you can use:
  - plt.pie(Pdfjoin.ActualTotal, labels=Pdfjoin.Country, autopct='%.1f%%', startangle=90, colors=plt.cm.Paired.colors)
    - labels provides the strings for each wedge,
    - autpct contains the format string:
      - % indicates that what follows is a format specifier.
      - The ".1f" specifies one decimal place (or ".0f" or ".2f").
      - The "%%" indicates a percentage – the first percentage sign indicates that the next character is to be used as a string.
    - startangle shows where the first wedge of the pie should be shown.
      - 0 starts on the x-axis (the right side of the pie).
      - 90 starts at the top (it goes counter-clockwise).
    - colors shows a sequence of colors for the pie chart.
      - cm stands for colormap.
      - The colormaps include: 'Pastel1', 'Pastel2', 'Paired', 'Accent', 'Dark2', 'Set1', 'Set2', 'Set3', 'tab10', 'tab20', 'tab20b' and 'tab20c'.
      - For more colors, see https://matplotlib.org/stable/users/explain/colors/colormaps.html
- To add a legend, use:
  - plt.legend() or plt.legend(title="Legend", labelcolor = "b", loc="best", fontsize = "medium")
    - loc (location) can be:
      - "upper/lower/center left/right"
      - "lower/upper center"
      - "center"
      - "best" – this produces the minimum overlap among the other locations.
    - fontsize can be:
      - 'xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large', or
      - the font size (integer)
- To create a line chart, then you can use:
  - plt.plot(Pdfjoin.Country, Pdfjoin.ActualTotal, marker='o', label="Actual", color='b', linewidth=2)
  - markers include:
    - . – point

        - , - pixel
        - o = circle
        - v, ^, <, > = triangle down, up, left and right
        - 1, 2, 3, 4 = tri down, up, left and right
        - 8 = octagon
        - s = square
        - p (lower case) = pentagon
        - P (upper case) = filled plus
        - * = star
        - h or H = hexagon1 and 2
        - + = plus
        - x (lower case) = x
        - X (upper case) = filled x
        - D = diamond
        - d = thin_diamond
        - | = vertical line
        - _ = horizontal line
        - See https://matplotlib.org/stable/api/markers_api.html for details.
    - linewidth is the line width in points.
        - Lakehouse vs Warehouse

Query data by using SQL

55. Query a lakehouse in Fabric by using SQL queries or the visual query editor

- To create SQL queries for a lakehouse, you can use the SQL Analytics Endpoint.
    - This is read-only only – no DML commands (UPDATE, INSERT, DELETE or MERGE).
    - It only allows you to read delta Tables, and not Files.
- You can access the SQL analytics endpoint:
    - as an object in your Workspace.
    - If you are in the Lakehouse, then you can change the connection in the top-right hand corner, and change it from Lakehouse to SQL analysis endpoint.
- To create a new SQL query, in the lakehouse:
    - click on "New SQL query", and enter your SELECT statement,
    - You can drag objects (for example) tables into the SQL query window.
    - You can click on "Run" to run the query.
- In the Data Preview, you can also:
    - See messages/results
    - Open in Excel,

- Explore this data (but you need to highlight a query first),
- Expand/Collapse the Data Preview

- To use the visual editor, click on "New visual query".
  - You can drag objects (for example) tables into the SQL query window.
  - You can use the transformations in this cut-down Power Query window, either in the menu or clicking the + in a data source.
    - Manage columns
      - Choose columns
      - Go to column
      - Remove column
      - Remove other columns
    - Reduce rows
      - Keep top/bottom/range of rows
      - Keep duplicates
      - Keep errors
      - Filter rows
    - Sort ascending/descending
    - Transform
      - Group by
      - Replace values
    - Combine
      - Merge queries/Merge queries as new
      - Append queries/Append queries as new
  - You can also add the following additional transformations by clicking on the + near the data source:
    - Transform any column
      - Change type
      - Rename
      - Move columns after/to end
    - Add column/Transform text column – Format
      - lowercase/UPPERCASE
      - Trim
      - Add prefix/suffix (not Add column)
    - Add column/Transform text column – Extract
      - Length
      - First/Last characters

        - Range
    - Add column/Transform number column – Standard
        - Add/Multiply/Subtract/Divide
        - Modulo
        - Percentage
        - Percent of
    - Add column/Transform number column – Scientific
        - Absolute value
        - Square/Cube/Power/Square root
        - Base-10 logarithm
    - Add column/Transform number column – Trigonometry
        - Sine/Cosine/Tangent
        - Arcsine/Arccosine/Arctangent
    - Add column/Transform number column – Rounding
        - Round up/Round down/Round
    - Add column/Transform number column – Information
        - Sign
    - Add column
        - Add column from examples for all columns/selection
        - Add conditional column
        - Duplicate column
- For a full Power Query window, click on Expand (top-right hand corner).
- In the menu, you can:
    - Save as view,
    - View SQL, or
    - Refresh.
- In the bottom to the view, you can:
    - Reset,
    - Fit to view,
    - Full screen,
    - Show mini-map,
    - change the zoom, and
    - collapse/expand all queries.
- In the Data preview, you can:
    - Download Excel file,

        - Visualize results, and
        - Change the size of the Data preview (including expanding/collapsing it).
    - For the individual columns, you can:
        - Sort ascending/descending,
        - Remove empty,
        - Filter

56. Query a warehouse in Fabric by using SQL queries or the visual query editor

- See topic 56.

57. Connect to and query semantic models by using the XMLA endpoint

- XMLA (XML for Analysis) allow you to connect from Power BI Service.
- It is available for Power BI Premium, Premium Per User, and Power BI Embedded workspaces.
- To enable it:
    - Go to Settings – Admin portal – Tenant settings,
    - Enable Integration settings – Allow XMLA endpoints and Analyze in Excel with on-premises semantic models.
    - If you want to "Analyze in Excel", so you also Enable "Users can work with semantic models in Excel using a live connection".
- By default, it is read-only.
    - So you can query semantic model data, metadata, events and schema.
- You can change it to read-write.
    - So you can perform management, governance, debugging, monitoring, and advanced semantic modeling.
    - To enable this in a Premium capacity:
        - Go to Settings – Admin portal,
        - Go to Capacity settings – Power BI Premium – [capacity name],
        - In Workloads, change the XMLA Endpoint setting to "Read Write".
    - To enable this in a Premium Per User:
        - Go to Settings – Admin portal,
        - Go to Premium Per User,
        - In Semantic model workload settings, change the XMLA Endpoint setting to "Read Write".
- To get the workspace connection URL.
    - In the workspace, go to … - Workspace Settings.
    - In the General tab, click on "Copy" under "Workspace Connection".
- It is the format:
    - powerbi://api.powerbi.com/v1.0/[tenant]/[workspace]

- If you need to specify an “Initial Catalog” (for example, in SQL Server Profile), use the semantic model name.
- The client applications for read-only include:
    - Microsoft Excel,
    - Power BI Report Builder,
    - DAX Studio
- The client applications which need read-write include:
    - Visual Studio with Analysis Service projects,
    - SQL Server Profiler version 18.9+,
    - Analysis Services Deployment Wizard, and
    - PowerShell cmdlets.
    - Please note: if you write to a semantic model from Power BI Desktop, you will no longer be able to download it back as a PBIX file.
- The client applications which can use either read-only (for query operations) or read-write (for scripting metadata) include:
    - SSMS version 18.9+,
    - Tabular Editor,
    - ALM Toolkit.
- To connect in Power BI Desktop:
    - The recommended way is:
        - Go to Home – Power BI semantic model.
        - Select the semantic model (you don’t need the workspace URL).
    - Alternatively:
        - Go to Get Data – Analysis Services,
        - Enter the workspace name,
        - Use “DirectQuery” (not “Import”),
        - In the Navigator, select the semantic model.
- To connect in SSMS:
    - Go to Connect – Connect to Server,
    - Change the “Server type” to “Analysis Services”.
    - In “Server name”, enter the workspace URL.
    - In “Authentication”, choose “Microsoft Entra MFA” (Multi-factor authentication).
    - In “User Name”, enter your user ID.
- In SSMS, you can process individual databases (semantic models), tables, or partitions by right-hand clicking on the database/table/partition.
- You can choose from the following process modes:
    - Process Default

        - Performs the processing which his necessary to get it to a fully processed data.
        - For example:
            - data for empty tables/partitions are loaded,
            - Hierarchies, calculated columns, and relationships are recalculated/build/re-built.
    - Process Full
        - Drops all existing data, and then processes the object.
        - This is required when a structural change has been made.
        - It requires the most resources.
    - Process Data (for a table or partition only)
        - Loads data only.
        - Does not rebuild hierarchies, relationships.
        - Does not recalculate calculated columns or measures.
    - Process Clear
        - Removes all data, either from the database, the partition, or from the table and table partitions.
    - Process Recalc (for a semantic model only):
        - Updates/recalculates hierarchies, relationships, and calculated columns.
    - Process Defrag (for a table only):
        - Defragments table indexes.
    - Process Add (for a partition only):
        - Incrementally updates partition with new data

