Configure Microsoft Fabric workspace settings

1-4. 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

1-4. 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. Settings include:

- 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

1. Configure Spark workspace settings

- To access the Spark workspace settings:
  - In the relevant workspace, click on “Workspace Settings”,
  - In the Data Engineering/Science section, click on Spark settings.
- Initially, you will be using a Starter pool.
  - They are able to be started fairly quickly – typically (say, 9-12) seconds.
  - They also allow for libraries to be installed quickly.
    - However, extra custom Spark properties or libraries from your capacity settings or workspace may take longer.
- You are only billed for the time that they are active/running.
  - You are billed for the time spent when stopping them, but not for when they are idle or deallocating or stopped.
  - The following customizations do not affect currently active Spark sessions.
- Starter pools can be customized to some extent, but only if you have admin access to the workspace. They have the following configuration:
  - Node family: Memory optimized

    - Node Size: medium
        - This has 8 vCores and a memory of 64 GB per node.
    - Autoscale: On.
        - This allows for the scale up/down of compute resources based on actual activity.
    - Number of nodes: by default they are between 1 and 10 for a F64 capacity or above.
        - So they can scale up/down for your requirements.
        - You can adjust both the maximum up to 16 for a F64 capacity.
- VCores are double the number of capacity units.
- The Node size is not configurable – it is always Medium.

SKU name	Default max nodes	Max number of nodes
F2	1	1
F4	1	1
F8	2	2
F16	3	4
F32	8	8
F64 or trial	10	16
F128	10	32
F256	10	64
F512	10	128
F1024	10	200
F2048	10	200

- Starter pool configuration
    - Dynamic Allocation: On.
        - This allows for more executors if needed.
        - The default min and max is 1 to 9 – the number of nodes minus 1.
            - These are the worker nodes. The other required node is the head node.
- You can create a new (not Starter) pool by clicking on the drop-down and selecting "New pool".
    - You can choose a Node size of Small, Medium, Large, X-Large and XX-Large, based on your compute requirements.
    - If you change the "Default pool for workspace", it will then be used for that workspace's Notebooks.

    - Note: it can take 2-3 minutes for custom pools to start.
- To give users the choice of pool (as opposed to just the default):
    - you will need to create an environment (an item in the workspace),
    - assign the pool to that environment, and
    - the end user can use that environment in the Notebook.
- You can:
    - Allow users to "customize compute configurations for items" (in the Pool tab), and
    - In the Environment tab, you can either select the default environment (with the end user can change), or select the Runtime Version.
    - You can also create an Environment object, and then set the Spar compute and acceleration.
        - In the acceleration tab, for Runtime 1.3 (Apache Spark 3.5), you can check “Enable native execution engine”.
        - It allows you to have faster job runs without additional cost, as it uses the native capabilities of the underlying data sources, reducing the overhead needed.
            - You can also enable it:
                - in SQL by using SET spark.native.enabled=TRUE; or
                - in PySpark using spark.conf.set('spark.native.enabled', 'true')
        - The native execution engine does not support:
            - Delta Merge operations, checkpoint scans and deleting vectors,
            - User-defined functions and the array_contains function,
            - JSON, XML and CSV formats, or
            - ANSI mode.
        - Where the native execution engine is not supported, it will generally fall back to the standard PySpark execution engine.
- For custom and starter pools, by default, sessions expire after 20 minutes.
    - This can be changed in the Settings - Jobs tab.
    - You can see the expiration for a current session by going to Home (or Run) – Connect – View session information.
    - You can stop a session in a Notebook by clicking on "Stop Session", or stopping the session in the monitoring hub page.
- Custom pools have a default autopause duration of 2 minutes after the sessions expire, following which they will be unallocated.
- In the Jobs tab, you can also "reserve maximum cores for active Spark jobs".
    - If OFF, it reserves the minimum number of cores for a particular job (which can then be autoscaled), therefore allowing more jobs to be run at the same time.

    - If ON, it reserves the maximum number of cores for a particular job, allowing for higher job reliability.
    - However, if cores are not available, in either the ON or OFF setting, then a job is refused.
- You can use a High Concurrency mode, which allows the same session to be used simultaneously in multiple notebooks.
    - Microsoft says that users get a 36 times faster session start compared to standard Spark session.
    - You can start such a session in a Notebook by going to Home (or Run) – Connect – New high concurrency session.
        - You can also see any currently running high concurrency sessions in that menu.
        - It needs to be switched on first, in the Workspace – Settings – High concurrency. You can switch it on for notebooks, and for pipelines running multiple notebooks.
        - It may make isolating monitoring and debugging for a particular notebook more difficult, and may degrade performance if there is already a lot of compute in that session.
        - All relevant notes will also have the same compute, library packages, and default lakehouse.
- You can see capacity use for spark sessions in the Microsoft Fabric Capacity Metrics app.

2. Configure domain workspace settings

- Domains are used to organize workspaces. It can allow for better discovery, as the items within the workspace are associated with the domain.
    - This can be used to filter information in, for example, the OneLake data hub.
    - You can also use some domain-level settings instead of tenant-level settings, allowing you to manage the settings more easily.
    - You can also create subdomains.
- Often they are organized into structures, such as:
    - Functional structure, for a hierarchy, by roles and business functions
        - Operations, IT, Strategic Development, Procurement
    - Project or product structure, for different teams or product lines
        - For example: Transportation, Retail, Technology, Healthcare
    - By process, if your business is process-oriented and then by products or markets.
    - By region – different continents, countries, or states/regions.
- You may also wish to consider any specific regulations or restrictions in your industry.
- You can assign workspaces by:
    - Workspace name (if they have a suitable prefix),
    - Workspace owner (e.g. all Healthcare in one domain, then all eye Healthcare in a subdomain),

    - By process, if your company uses that as the overriding factor.
- The following roles are involved with domain creation:
    - Fabric admin. They have the following permissions:
        - Creating/edit domains,
        - Assign domain admins and contributors, and
        - Associate workspaces to domains.
        - See, edit and delete all domains.
    - Domain admin. They have the following permissions:
        - See and edit their own domains (for which they are admin), including creating subdomains,
        - Update the domain description,
        - Assign domain contributors,
        - Associate workspaces to domains.
        - Note: they cannot:
            - delete the domain,
            - change its name, or
            - assign/remove other domain admins.
    - Domain contributors. These are workspace admins
        - These need to be Administrator of a least one workspace.
        - They can assign/change such a workspace to a domain.
        - They can do this from the workspace settings.
- Fabric admins can create a domain:
    - In the Admin portal, go to Domains tab.
    - Click on "Create new domain".
    - Enter the name, and optionally add any domain admins.
    - Click create.
- Once a domain has been created, Fabric/Domain admins can associate a workspace to a domain by:
    - Going to the workspace
    - Click on "Workspace settings".
    - In the General tab, go to the Domain section, and assign the workspace to the domain.
- Fabric admins or Domain admins can create subdomains:
    - Go to the relevant domain.
    - Click "New subdomain" and provide a name.
    - Note – there is no such thing as a subdomain admin. The admins for a subdomain are the same admins for the main domain.

- At the timing of creating a new workspace, you can assign that workspace to a domain.
- Fabric admins or Domain admins can assign domains or subdomains to a workspace:
    - Go to the relevant domain.
    - Click on "Assign workspaces".
    - Search for the domains, either by workspace name, workspace admin or capacity.
        - If any workspaces are already associated with another domain, you will get a warning message.
            - You can override this, but only the tenant setting "Allow tenant and domain admins to override workspace assignments" is enabled.
        - If searching for workspace admin: all workspaces the users/security groups are admins will be assigned to the domain apart from "My workspace".
        - If searching for workspace admin or capacity: this is not a continuous process – it only affects domains which are the users/security groups are admins for at that time.
        - If searching for workspace name or capacity: you can select multiple workspaces/capacities at the same time.
- In the Admin portal – Domains tab, you can update the Domain settings, by:
    - clicking on "Domain settings" in the relevant domain, or
    - in the Domain tab, clicking on the … next to the Domain, and selecting Settings.
- The various tabs are:
    - General settings. You can edit the name, and Domain/Fabric admins can edit the description.
    - Image. This is the image/color used in the OneLake data hub when that domain is selected.
    - Admins. Fabric admins can specify Domain admins.
    - Contributors. Fabric/domain admins can assign Domain contributors – either:
        - "The entire organization",
        - "Specific users or security groups", or
        - "Tenant and domain admins only".
    - Default domain. Fabric/domain admins can assign users/security groups to this domain – this includes assigning them the domain contributor role:
        - The workspaces for which these users/security groups are Administrator will be assigned this domain, unless it already has a domain assignment.
        - Any new workspaces that are created by this user or a member of the security group will also be assigned to the default domain.
    - Delegated settings. The following admin settings can be assigned at the domain level, and can there override the tenant level admin settings:
        - Default sensitivity label.
            - This can be assigned by Fabric/domain admins.

        - Go into Information protection, and select the sensitivity label from "Set a default label for this domain".
    - Certification.
        - Allow users/security groups to certify items (apart from Power BI dashboards, which cannot be certified).
        - You can select "All the users in domain" or include or exclude "Specific security groups".
        - You can optionally check "Override tenant admin selection", and optionally provide a URL for relevant documentation.
        - This requires certification to enabled in the Tenant settings.

3. Configure OneLake workspace settings

- To access the Workspace OneLake settings:
    - go to the relevant Workspace,
    - Click on “Workspace Settings".
    - Click on the "OneLake" tab.
- There, you can turn on/off the Cache for Shortcuts. When it is enabled:
    - This is for external Google Cloud Storage (GCS), Amazon S3 and S3 compatible shortcuts (in other words, in a different cloud).
        - So not, for example, for Azure Data Lake Storage (ADLS) Gen2 storage accounts or shortcuts pointing to other Fabric items.
    - When such a shortcut reads files, the files are stored in a cache.
        - Files bigger than 1 Gigabyte are not, however, cached.
    - Future read requests are then read from the cache, rather than be accessed from the other cloud.
        - However, if the external cloud's content has been updated, that is read instead, and the cache updated.
    - Cached files are removed by default 24 hours after their last read.
        - This can be changed to be up to 28 days.
- There are also OneLake settings in the Tenant settings. If you go to Admin portal – Tenant settings, you can select the following:
    - Users can access data stored in OneLake with apps external to Fabric,
        - such as OneLake File Explorer, Databricks, and Azure Data Lake Storage APIs (application programming interface)
    - Use short-lived user-delegated Shared Access Signature (SAS) tokens,
        - allow applications to access OneLake through tokens, which can last up to one hour.
    - Authenticate with OneLake user-delegated SAS tokens, and
        - allows applications to authenticate
    - Users can sync data in OneLake with the OneLake File Explorer app

    - Use the OneLake File Explorer app in Windows File Explorer.

4. Configure data workflow workspace settings

- Apache Airflow jobs were introduced in May 2024.
    - It allows you to create, schedule and monitor complex jobs.
    - Apache Airflow is the name of the open-source platform that is used to power these jobs.
    - It uses directed acyclic graphs (DAGs) to show data pipelines.
    - The creation of these jobs is outside of the DP-700 exam.
- Apache Airflow jobs would need to be enabled first:
    - In Admin Portal, click on the Tenant Settings tab.
    - In the "Microsoft Fabric" section, open the "Users can create and use Apache Airflow Job".
    - You can then enable it for the entire organization or include/exclude specific security groups.
- To configure data workflow workspace settings:
    - Go to your workspace,
    - Click on "Workspace settings",
    - Expand "Data Factory" and click on "Apache Airflow runtime settings" (this used to be called "Data workflow settings").
- In that tab, you can use the Starter Pool or click on "New Pool" to create a Custom pool.
    - Starter Pools shut down after 20 minutes of inactivity. They should be used in Developer environments.
    - Custom Pools are always-on until and unless they are paused manually. They could be used in Production environments.
- For Custom pools, you can select the Computer node size:
    - The "Large" node size is for running complex or production DAGs (Directed Acyclic Graphs). This is used in Starter Pools.
    - The "Small" size is for simpler DAGs.
- In Custom Pools, you can also choose to:
    - Enable autoscale (add nodes up/down as needed), and
    - Allow extra nodes so the DAG can use these nodes concurrently – each node provides up to the three workers.
    - These options are not available in Starter Pools.

Implement lifecycle management in Fabric

5. Configure version control

- 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.

6. Implement database projects

- A database project is automatically implemented when you add a warehouse to a source control.
- It contains SQL code saved in Data Definition Language (DDL) as .sql files for the schema (definition) of items in the database, includes tables, views, functions or stored procedures.
    - It does not include data.
    - It also does not include SQL security features, such as object-, column- or row-level security, or dynamic data masking.
        - They could be exported manually if these features were separately written in code.
    - Git integration also cannot easily handle ALTERations to the TABLE – if you execute an ALTER TABLE, then the table will be dropped and then created again, which may result in data loss.
- You can then manage a warehouse schema in Azure Data Studio and Visual Studio Code using the SQL Database Projects extension.
- This database project can also be downloaded by:
    - Going to the Home tab in the Warehouse, and
    - Clicking on "Download SQL database project".
- You can use this downloaded file to copy the schema to a newly created warehouse.
    - When the warehouse is empty:
        - on the main screen you can click on "SQL database project" in the "Start developing" section, and
        - Upload your SQL database project folder zipped file.
- You can also do this by creating a deployment pipeline, which means that changes in one warehouse's schema can be replicated in another warehouse's schema.

7. Create and configure deployment pipelines

- 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 not 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 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.

Configure security and governance

8. Implement workspace-level access controls

- 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.

9. Implement item-level access controls

- 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.

10. Implement row-level, column-level, object-level, and folder-/file-level access controls

Row-level control

- To implement row-level security in a Data Warehouse, you need to have:
    - A security function. This needs to RETURNS TABLE WITH SCHEMABINDING, and returns SELECT 1 where the appropriate conditions are true. For example:

        CREATE FUNCTION securityfunction(@UserName as VARCHAR(50))

        RETURNS TABLE

        WITH SCHEMABINDING

        AS

        RETURN SELECT 1 AS securityfunction

        WHERE @UserName = LEFT(USER_NAME(), LEN(@UserName)) OR LEFT(USER_NAME(),4) = 'Jane'
    - A security policy, which references the security function.

        CREATE SECURITY POLICY SecurityPolicy

        ADD FILTER PREDICATE dbo.securityfunction(UserName)

        ON dbo.tblActual

        WITH (STATE = ON);

Column-level control

- You can implement column-level control by granting SELECT permissions to only some of the columns in a table or query:
    - GRANT SELECT ON NameOfTable(ColumnNames) TO [UserName]

Object-level control

- There are three main commands to manage permissions:
    - GRANT permission ON object TO user/security group – this permits access to an object, as long as there isn't a DENY

    - DENY permission ON object TO user/security group – this prevents access to an object. It overrules a GRANT.
        - DENY does not apply to owners of objects or members of the sysadmin fixed server role.
    - REVOKE permission ON object TO user/security group – removes a GRANT or DENY
- Examples of objects include Table, View, Function, Stored Procedure.
- Permissions include SELECT, INSERT, UPDATE, DELETE and EXECUTE (run stored procedures)
- You can also add WITH GRANT OPTION to allow that user to GRANT the permission to other user/security groups.
- A list of the principals (user/login/group) can be found by running
    - SELECT * FROM sys.database_principals
    - The type column includes:
        - E – External user from Microsoft Entra ID,
        - S – SQL user
        - X – External group from Microsoft Entra group or applications

File-level control

- To configure access to folders in a Lakehouse, click on “Manage OneLake data access”.
    - You need to have Contributor, Member or Administrator access for the workspace.
    - Once enabled, it cannot be turned off.
- A new role “DefaultReader” will be created, which gives you access to read all files.
    - You can delete this default role by checking it and clicking on “Delete”.
    - You will also be granted all permissions: Read, Write, ReadAll, Reshare, Execute, ViewOutput and ViewLogs.
- You can create a role by clicking on “New Role”.
    - The role name must start with a letter, contain only letters or numbers (up to 128 characters), and be unique.
- You can then select “All folders” or “Selected folders”.
    - Folders also include subfolders within the “Tables” folder. This means that you can control access to the data within tables.
    - Only check a folder if you want the user to have permissions for that folder.
- You can then click “Assign role” to add people, groups, or email addresses to the role.
    - You can also access this by clicking on the … next to the role and click on “Assign”.
    - You can assign users by clicking:
        - inside the “Add people or groups” box, or
        - the “Add users based on Lakehouse permissions” (these are called “virtual members”).

- The Lakehouse permissions are: Read, Write, Reshare, Execute and ReadAll.
- These permissions use an AND – if you select “Read, Execute”, then this role will only apply to users who have both the Read and Execute permissions (not OR).

11. Implement dynamic data masking

- Dynamic data masking limits how data is shown in data warehouses.
  - For example, hiding parts of emails or credit cards to users who do not need this information.
  - Data will not be masked for those with Contributor rights or greater, or with elevated permissions (such as the UNMASK or CONTROL permission) on the Warehouse.
  - The mask can also be bypassed if someone has been shared the warehouse using the ReadAll permission, as they can query the underlying OneLake data.
- The permissions which can be used are:
  - Standard CREATE TABLE and ALTER permissions to incorporate the dynamic data masking when creating a table.
  - The ALTER ANY MASK and ALTER permissions for adding, replacing or removing masks for existing columns.
    - You might want to assign the ALTER ANY MASK for security officers.
    - The ALTER ANY MASK permission is also granted with the CONTROL permission.
  - The SELECT permission is used to view the data.
  - The UNMASK permission allows the SELECT statements to be unmasked.
    - This is also granted with the CONTROL permission.
- Permissions can be granted using:
  - GRANT Name_of_permission ON Table_name TO User_name.
- They can also be REVOKEd.
- To mask, then use the following syntaxes:
  - As part of a table definition: Column_Name Column_Type MASKED WITH (FUNCTION = 'name_of_function') NULL
  - As part of an alter column: ALTER TABLE Name_of_table ALTER COLUMN Column_Name ADD MASKED WITH (FUNCTION = 'name_of_function')
- To unmask, use:
  - ALTER TABLE Name_of_table ALTER COLUMN Column_Name DROP MASKED
- The functions are:
  - default(). This will show:
    - xxxx (or fewer x's) for strings, 0 for numbers, 1 January 1900 for date/time, and ASCII value 0 for binary, varbinary or image data types.

    - email(). This will show the first letter of the email address, followed by XXX@XXX.com.
    - random(first_value, last_value), to show a random number between first_value and last_value.
        - For example, random(1, 10)
    - partial(prefix, [padding], suffix) to show a random string. prefix and suffix contain the maximum number of first/last characters to be shown.
        - For example, partial(1, "XXXXXX", 2)
- Note – this technique only masks the data – it does not stop intelligent guesses.
    - For example, let's say Country was masked with the default(), and I was logged in a user which returns masked data.
    - SELECT * FROM tblTable would return xxxx in the Country column.
    - However, while SELECT * FROM tblTable WHERE Country = 'United States' would still return xxxx in the Country column, it's fairly obvious what the actual Country is.
- Therefore, this should be used in conjunction with other security, such as object-, row-, and column-level security.

12. Apply sensitivity labels to items

- You can apply sensitivity labels to Fabric items, if you have Power BI Pro or Premium Per User license.
    - These labels do not affect access to content from Fabric or the Power BI Service.
        - They can affect access to content in Power BI Desktop.
    - They are also applied to any content which is exported:
        - Export to Excel/PowerPoint/PDF,
        - Analyze in Excel from the Power BI Service,
        - Creating a PivotTable from a semantic model, or
        - Download to .pbix from the Power BI Service.
- This is for items including lakehouses, data warehouses, pipelines, semantic models, notebooks, eventhouse, eventstreams, reports, and dashboards, and Dataflow Gen2.
- There are two ways to do this:
    - It may be at the top of the screen, next to the item name
        - This does not apply to Dataflow Gen2.
    - It may be in the Settings:
        - click … next to the item,
        - go to Settings
        - go to the Sensitivity label section or tab.
    - You may also be able to select “Apply to downstream items”. If so, the label will be applied to items created downstream.

        - For example, a semantic model created from a Lakehouse, or a report created from that semantic model, or a dashboard which has a tile which is pinned from the report.
        - If, for some reason, it cannot be so applied on a downstream item, the new item can still be created.

- The default sensitivity labels are:
    - Highly Confidential
    - Confidential
    - No Sensitivity
    - None (this could be any sensitivity, but hasn't been assessed yet).
- Once set, they will appear in a column in the workspace, and at the top of the screen when the item is opened.
- In Power BI Desktop, you can also add sensitivity labels in Home – Sensitivity.
    - It is then visible in the status bar.
- To enable sensitivity labels to be set, they need to be enabled in the Tenant:
    - Go to the Fabric admin portal, and then Tenant settings.
    - In the Information protection section, enable “Allow users to apply sensitivity labels for content”.
    - To allow for the downstream of updated sensitivity labels, enable “Automatically apply sensitivity labels to downstream content”.
        - You can also “allow workspace admins to override automatically applied sensitivity labels”, and “Restrict content with protected labels from being shared via link with everyone in your organization”.
- This list can be updated in Microsoft Purview Information Protection portal (purview.microsoft.com).
    - You can also create a Policy to apply a default label for new Power BI reports, dashboards and semantic models can be applied in the Microsoft Purview compliance portal.
- If you create semantic models which are connected to data sources which have got sensitivity labels, then the semantic models can inherit those labels. These data sources are:
    - Excel files stored on OneDrive or SharePoint Online (not on premises behind a gateway),
    - Azure Synapse Analytics, and
    - Azure SQL Database.
- It requires additional settings in Microsoft Purview, and in the Fabric tenant:
    - Go to the Fabric admin portal, and then Tenant settings.
    - In the Information protection section, enable “Apply sensitivity labels from data sources to their data in Power BI”.

13. Endorse items

- There are three different types of endorsement: promotion, certification and master data.
    - Promotion.
        - This can be applied to any item apart from Power BI Dashboards.
        - It advertises the item's existence, and should only be used for data which is of sufficient quality to share with others.
        - Promotion can be done by any user.
    - Certification.
        - This can be applied to any item apart from Power BI Dashboards.
        - It is used to say this item is quality-controlled by a user who understands the data, and can be used in the entire tenant.
        - Certification can only be done by authorized reviewers.
    - Master data.
        - This can only be applied to items which have data, such as lakehouses, data warehouses, and semantic models.
        - It is used for "single-source-of-truth" data. It is the central source for that data.
        - Setting "Master data" status can only be done by authorized reviewers.
- To endorse an item:
    - Click on the … next to the item and go to Settings.
    - Change the endorsement.
    - For semantic models, you can also check "Make discoverable". This makes it shown in the OneLake data hub, and users who haven't got access to it can request access.
- Once endorsed, you will see the endorsement:
    - in a column in the Workspace, and
    - in the drop-down at the top of the screen, when the item is opened.
- In the Admin portal – Tenant settings, you can adjust the following:
    - Enable "Certification"
        - You can also "Specify URL for documentation page". This is the page that users will see when clicking "How do I get content certified?"
    - Enable the "Master data" badge.
    - "Make promoted content discoverable" and "Make certified content discoverable".

13a. Implement and use workspace logging

- Workspace monitoring can be used to query the logs stored in a new Eventhouse database, which are to do with security, data collection and access.
- To enable it, you will need:
    - The admin role in a Power BI Premium/Fabric capacity,

  - The tenant "Workspace admins can turn on monitoring for their workspaces (preview)" setting enabled.
  - Log analytics cannot already be enabled.

- To turn it on:
  - Go to a workspace,
  - Click on Workspace settings,
  - Go to the Monitoring tab, and
  - Click on "+ Eventhouse".
- The eventhouse is read-only, and monitoring data is retained for 30 days.
- Once it has been set up, you can query the following:
  - Metrics, stored in the EventhouseMetrics table,
    - Monitor and troubleshoot ingestion performance and trends

```
EventhouseMetrics
| order by MetricSumValue
```

    - Look at materialized views and exports.
  - Command logs, stored in the EventhouseCommandLogs table, and Query logs, stored in the EventQueryLogs table, to:
    - Look at command/query performance and trends,
    - Show commands/queries which use a lot of resources

```
EventhouseCommandLogs
| summarize CPUTimeMs = sum(CpuTimeMs) by
EventhouseCommandType
| order by CPUTimeMs
| limit 10

EventhouseQueryLogs
| order by CpuTimeMs
| limit 10
```

    - Show users and applications which run the most commands/queries.

```
EventhouseCommandLogs OR EventhouseQueryLogs
| summarize CPUTimeMs = sum(CpuTimeMs)
by UPN = tostring(Identity.claims.upn), APPID =
tostring(Identity.claims.appid)
```

  - Data operation logs, stored in the EventhouseDataOperationLogs table,
    - Analyze data operations performance and trends,
    - Look at data operations using a lot of CPU

```
EventhouseDataOperationLogs
```



```
| order by CpuTimeMs
```

        - Identify data operators used on a specific table.
        - Data operations includes:
            - batch ingestion,
            - update policy,
            - materialized view update,
            - Sealing an extent of streaming data.
    - Ingestion results logs
        - Monitor the number of successful ingestions, and
        - Monitor and troubleshoot failed ingestions.

```
EventhouseIngestionResultsLogs
| project OperationName, WorkspaceName, Status,
ResultCode, ShouldRetry
```

- You can also use the “Fabric Workspace Monitoring Dashboard.json”. To use it:
    - Download it from https://github.com/microsoft/fabric-toolbox/tree/main/monitoring/workspace-monitoring-dashboards
    - Create a new Real-Time Dashboard.
    - Go to Manage – Replace from file.
    - Select the downloaded file.
    - Go to Manage – Data sources.
    - Click on “Add+” – “Eventhouse / KQL Database”.
    - Select the “Monitoring KQL database” in the OneLake catalog dialog box.
    - Click Connect, then Add, then Close.
    - Click on the Workspace option (at the top-left hand corner of the Dashboard) and select the “Monitoring KQL database”.
- You can also use the “Fabric Workspace Monitoring.pbit” Power BI template in Power BI Desktop. To use it:
    - Download it from https://github.com/microsoft/fabric-toolbox/tree/main/monitoring/workspace-monitoring-dashboards
    - Open the “Monitoring KQL database”, and copy the “Query URI”
    - Open the template in Power BI Desktop
    - Paste the “Query URI”, change any other settings, and click Load.
    - Save and Publish the report into the Power BI Service.
    - Next to the “Fabric Workspace Monitoring” semantic model, click on the … and go to Settings.
    - In the Data source credentials, click on “Edit credentials” and sign in.
    - Refresh the data and open the report.

Orchestrate processes

14. Choose between a pipeline and a notebook

- For copying high quantities of data with a low-code/no-code, either as a one-off or on a schedule, use a Pipeline Copy Activity.
- You can use these destinations:
    - Various Azure objects,
    - Fabric: Data Warehouse, KQL Database, Lakehouse,
    - Database: MongoDB, Oracle, SQL database and SQL Server,
    - Files: Amazon S3, File system, Google Cloud Storage (GCS),
    - Power Platform: Dataverse,
    - Services and apps: Dynamics CRM, Salesforce, Snowflake, and
    - Generic protocol: ODBC, REST.
- You can also use these sources:
    - Database: Amazon RDS for SQL Server, Amazon Redshift, DB2, Google BigQuery, MariaDB, MySQL, PostgreSQL, SAP HANA, Vertica,
    - Services and apps: Dynamics AX, Microsoft 365, SharePoint Online list,
    - Files: FTP, Oracle Cloud Storage,
    - Generic protocol: HTTP, OData, and
    - ServiceNow.
- If you need to transform data using a no-code or low-code interface, with hundreds of transformation options and sources, then you can use Dataflow Gen 2…
    - but only if the destinations are suitable: Lakehouse, Warehouse, SQL database, Azure SQL database, and Azure Data explorer (Kusto)
- If you need to transform data, then Spark code in notebooks allows you to process large amounts of data in parallel, and then write into Delta tables in lakehouses, where it can be used elsewhere.
- If you need to do a combination, then you can create notebooks, and add them into a pipeline.

	Pipeline copy activity	Dataflow Gen 2	Notebook
Use case, including data ingestion	Data lake and data warehouse migration, lightweight transformation	Data transformation, data profiling, and ...	
		data wrangling	data processing
Primary developer persona	Data engineer, data integrator		Data engineer, scientist, developer
		business analyst	
Primary developer skill set	ETL, SQL, JSON	ETL, M, SQL	Spark (Scala, Python, Spark SQL, R)
Development interface	Graphical - No code/low code		Code using Notebook, Spark job definitions
	Wizard, canvas	Power Query	
Sources	30+ connectors	150+ connectors	100s of Spark libraries
Destinations	18+ connectors	Few	100s of Spark libraries
Transformation complexity	Low: lightweight type conversion, column mapping, merge/split files, flatten hierarchy	Low to high: 300+ transformation functions	Low to high: support for native Spark and open-source libraries

15. Design and implement schedules and event-based triggers

Data pipeline

- To Schedule a 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.

Dataflow

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

Notebook

- To schedule a notebook:
    - In the notebook, go to Run - Schedule.
    - For the scheduling options, see above in data pipelines.

Events from Azure Blob storage

- An event-based trigger, part of the Data Factory, runs when something happens to files or folders in Azure Blob storage.
- To create an event-based trigger in a pipeline:
    - click on Home or Run – Add trigger. This opens the “Set alert” panel.
    - In “Source”, click on “Select events”. This opens the “Connect data source” dialog box.
    - In the “Configure” section, you can select the Storage account:
        - You can connect to an existing, or select a connected, Azure Blob Storage Account.
        - You then select a subscription and an Azure Blob Storage account.
        - Then click “Next”.
    - In the "Configure alert" section, you can select the Event type from Microsoft.Storage:
        - Blob-Created, -Deleted, -Renamed and -TierChanged,
        - DirectoryCreated, -Deleted and –Renamed,
        - BlobInventoryPolicyCompleted,
        - AsyncOperationInitiated, and
        - LifecyclePolicyCompleted.
    - You can also filter the events, so not all of events would run the trigger.
        - The fields are: Source, subject, type, time, id, data and specversion.
        - The operators include in, greater/less than and equals to, contains (and the opposite of these operators).
    - You can click “Next” then “Connect”. It then creates an eventstream.
- You can then create an action of “Run a Fabric item” – either a pipeline or a notebook.
- You can then save this Activator item in a workspace and name the item (by clicking on “Create a new item”).

Events-based trigger using Activator

- Activator is used in Microsoft Fabric for doing an action (such as altering people or starting Power Automate workflows) when needed due to changing data.
- In a workspace, create a new Activator.
- You can "try sample" data":
    - Bicycle rentals,
    - Stock market, and
    - Yellow taxi
- You can "Get data" from:
    - Microsoft sources:
        - Various Azure sources, including Cosmos DB (CDC – Change Data Capture), PostgreSQL db (CDC), Event Hubs, IoT Hub, Service Bus, Azure SQL Database (CDC) and Azure SQL Managed Instance (CDC).
        - SQL Server on a Virtual Machine (CDC).
    - MySQL DB (CDC),
    - Azure Blob Storage events,
    - Fabric events:
        - Workspace Items events: Item creation/deletion/update/read was successful/failed,
        - Job events: Item Job Created, Status Changed, Succeeded and Failed, and
        - OneLake events: file/folder creation/deletion/renamed.
- You can create a new rule by clicking on "+New rule", and then set the attribute (what you are monitoring).
- Then you can set the condition:
    - Numeric change/state,
        - Change:
            - Increases/decrease above/to or above/by
            - Enters/exits range,
            - Changes from/to/by,
        - State: =, <>, >, >=, <, <=, and is within/outside range
    - Text changes from/to, or state (=, <>, [does not] begins, contains, end, and conditions on the length of the text),
    - Logical change (becomes true/false)/state (=. <>),
    - Common change, and
    - Heartbeat – On every value or no presence of data over a certain period of time (e.g. from 10 seconds to 24 hours).
- You can also click "Add filter".

- You may also be able to click “Add summarization” for:
    - Average/Minimum/Maximum over time,
    - Count/Total.
- You can then choose the action to take:
    - Send an email,
    - Add a Teams message,
    - Run a Fabric notebook or pipeline, or
    - New custom action.
- Note: rules are by default Stopped. You will need to “Start” or “Save and start” to activate it.
    - When it Is started, you will see “Running” in the title of the rule card. You will also see a different icon.
    - To halt the rule, click “Stop”.
- If data is late-arriving then:
    - Power BI rules are evaluated every hour, and evaluates all events that arrive a maximum of one hour after the event occurs.
    - For other items such as eventstreams, the “Advanced settings” may have a “Wait time for late-arriving events”.
        - You can set it for 1, 2 or 5 minutes.
        - This configures the balance between waiting for any potential late-arriving events, or processing data more swiftly, though it may be incomplete.
- You can also create alerts from:
    - a Real-Time Dashboard
    - a KQL queryset
        - Run a query that returns a visualization.
        - After the query runs, click “Set Alert”.
            - You can set the time frequency, from 1 minutes to 24 hours. The default is 5 minutes.
            - The Condition is “On each event”.
            - Choose an action from: Send me an email, Message me in Teams, and Run a Fabric item.
            - You can then create the Workspace and Item location.
    - an Eventstream
        - In the “Add destination”, select “Activator”.
        - You can then customize it.
    - a Power BI report.
        - You can also create alerts from Power BI Dashboards, but they do not Activator, and so don’t require a Fabric license.

        - You can click on the … and go to Add alert.
        - The "Set alert" pane will open, with the Monitor – Visual set.
        - You can then assign the condition: a measure, an operator, and a value.
        - You can then set an action: an email or Teams message.

16. Implement orchestration patterns with notebooks and pipelines, including parameters and dynamic expressions

- You can call on notebooks from pipelines with a parameter.
    - This allows you to pass information into the notebook.
    - With a different input, you can get a different output, meaning that you don't have to recreate the notebook to cope with different inputs.
- In the notebook, click on the … in the first cell and select "Toggle parameter cell".
    - The word "Parameters" now appears in the bottom-right hand corner of the cell.
- You can then add a Parameter and assign it a default value.
- Then (for example) write some code to read a table, filter it, and then write to a new table.
- In the Pipeline, add a Notebook activity.
- In the activity Settings:
    - Add the relevant workspace and notebook,
    - In "Base parameters", add the parameter name, type (String, Int, Float or Bool) and the parameter value.
- Then run the Pipeline, and query the new table using the SQL analytics endpoint.
- You can also change the reference to the notebook from a fixed string literal to a variable name.
    - In the Pipeline – Notebook activity – Settings – Notebook, change the name to "Add dynamic content".
    - In the "Add dynamic content" pane, go to the Parameters type, click the + button, and then add a Parameter (Name, Type and Default value).
        - You cannot use a space in the Name – you can only use letters, numbers, and the underscore character.
    - Click on the parameter name to add it to the "Add dynamic content" – it will be in the format of: @pipeline().parameters.ParameterName .
    - Click OK to confirm the change to the "Notebook ID".
- When you run this, the "Pipeline Run" pane will ask for the name of the Notebook, and gives the default name.
- In the dynamic expression, you can also use System Variables.
    - @pipeline().Pipeline – the ID of the pipeline
    - @pipeline().PipelineName
    - @pipeline().TriggerID, .TriggerName and .TriggerTime – the ID and Name of the trigger which started the pipeline, together with the actual time the trigger was run. The trigger could be:

        - a scheduled trigger,
        - an event-based trigger, or
        - a manual trigger (will return “Manual”)
    - @pipeline()?.TriggeredByPipelineRunId and @pipeline()?.TriggeredByPipelineName – the pipeline which triggers this pipeline.
- When the pipeline has a storage event trigger attached, then you can use:
    - @pipeline()?.TriggerEvent?.FileName and .FolderPath – the name of the file/folder for the file whose event triggered the pipeline run.
- When the pipeline has an event trigger attached, then you can use:
    - @pipeline()?.TriggerEvent?.Source, .Subject and ,.Type – the trigger event details.
- You can also add variables in the variable tab.
- You can also use functions in dynamic expressions.
    - Logical functions: if, equals, and/not/or, greater/less/greaterOfEquals/lessOrEquals
    - String functions: concat, endsWith/startsWith, indexOf/lastIndexOf, replace, substring, toLower/toUpper, trim
    - Conversion functions, including: coalesce, float, int, string
    - Mathematical functions, including: add, sub, mul, div, min/max, mod
    - Date functions, including:
        - addDays/Hours/Minutes/Seconds,
        - addToTime, subtractFromTime,
        - dayOfMonth/Week/Year,
        - formatDateTime,
        - startOfMonth/Day/Hour,
        - utcNow, getFutureTime/getPastTime,
        - convertFromUtc/ToUtc, convertTimeZone

Design and implement loading patterns

17. Design and implement full and incremental data loads

- To create an incremental data load from Data Warehouse to Lakehouse, you need:
    - A column (in both databases) which shows when the row was created (a “watermark”).
    - A table in the source database which stores the last watermark that was copied (a “watermark table”).
        - It should have a single row which has a date before any of the watermarked dates.
    - A pipeline which contains:

        - Two lookup activities, one to retrieve the last watermark value from the "watermark table", and another to retrieve the current maximum "watermark" value.
        - A copy activity which copies the data between the two databases for the relevant rows (after the last watermark up to and including the current watermark).
        - A stored procedure activity which updates the watermark table with the current watermark.

- Step 1 – Create tables and stored procedure in a Warehouse (called "IncrementalWarehouse"), and a blank Lakehouse.
- Step 2 – Create a pipeline and add a lookup activity from the "watermark table".
    - In a pipeline, go to:
        - the Home menu and click Lookup, or
        - the Activities menu, click on … and select Lookup.
    - Name this lookup activity "LookuptblWatermark".
    - In the Settings tab:
        - In Connection, select the IncrementalWarehouse.
        - Use the default "Table" for "Use query".
        - In Table, select "dbo.tblWaterMark".
        - Use the default "First row only".
- Step 3 – Add a lookup activity to get the latest watermark
    - Go to Home – Lookup
    - Name this activity "MaximumWatermark".
    - In the Settings tab:
        - In Connection, select the IncrementalWarehouse.
        - Change "Use query" to "Query"
        - Enter: SELECT MAX(OrderDate) AS LatestWatermark FROM FactImport
        - Use the default "First row only".
- Step 4 – Add the copy activity
    - Go to Home – Copy data – Add to canvas.
    - Connect the two lookup activities to the copy table using "On Success".
    - In the Source tab:
        - In Connection, select IncrementalWarehouse.
        - Change "Use query" to "Query".
        - In Query, write the following query:
            - SELECT * FROM FactImport WHERE OrderDate > '@{activity('LookuptblWatermark').output.firstRow.Watermark}' and OrderDate <=

'@{activity('MaximumWatermark').output.firstRow.LatestWatermark}'

- You can click inside the Query and then click on "Add dynamic content"
- Enter SELECT * FROM FactImport WHERE OrderDate >
- Create an empty first row in the query, and click on "LookuptblWatermark first row". This inserts @activity('LookuptblWatermark').output.firstRow
- Move the insertion to after the rest of the query.
- Enter ".Watermark",
- Add curly brackets after the @ sign and at the end, and quote marks before and afterwards.
- then "AND OrderDate <= ".
- Do the same with "MaximumWatermark first row".
- Enter ".LatestWatermark".

  - In the Destination tab:
    - Change the Connection to a Lakehouse.
    - Change the "Root folder" to "Files".
    - Change the File path to Incremental/@concat(pipeline().RunId, '.txt')
    - The File format should be "DelimitedText".

- Step 5 – Add the stored procedure to the pipeline
  - Go to Activities – Stored procedure.
  - Connect it to the Copy data activity using "On Success".
  - In the Stored procedure activity:
    - Go to Settings.
    - Change Connection to IncrementalWarehouse.
    - Next to "Stored procedure name", click on Refresh, then click on "[dbo].[updateWatermark]".
    - Expand "Stored procedure parameters"
    - Click on "+ New".
    - For this new parameter:
      - The Name is Watermark (as per the Stored Procedure).
      - The Type is DateTime (as per the Stored Procedure),
      - The Value is @{activity('MaximumWatermark').output.firstRow.LatestWatermark}
      - Note – there are no quote marks around this value, as it is going to be treated as a DateTime.

- Step 6 – query the end result:
    - In a Lakehouse, create a dataframe which brings the files together.
- You can also do incremental refresh in Dataflow Gen2.
    - The data source must allow for query folding.
    - There needs to be a Date or DateTime column.
    - The data destination must allow incremental refresh, namely:
        - Fabric Warehouse
        - Azure SQL Database
        - Azure Synapse Analytics.
- To implement the incremental refresh:
    - In Dataflow Gen2, right-hand click the query and click on “Incremental refresh”.
    - Select the following from the Incremental refresh dialog box:
        - Check “Enable incremental refresh”.
        - Select the DateTime column to filter the query by.
        - Select the time period to filter (x days/weeks/months/quarter or years).
        - Add a Bucket size. The Bucket holds the data since the last refresh – up to 50 buckets.
            - Fewer buckets mean less data per iteration, but more iterations.
            - More buckets mean more data per iteration, but fewer iterations.
            - For example – 4 years and bucket size of month.
        - Select the column to say that the refresh shouldn’t happen if the maximum value in this column has not changed.
            - While this can be the same as the first DateTime column, it may be different – for example, a Created Date for filtering, and a Modified Date to mark changes.
        - Optionally, you can say to ignore incomplete periods (the period is shown in the Bucket size).
        - Optionally, in the Advanced section, you can check “Require incremental fresh query to fully fold”.
            - If checked, this pushes the query down to the source system, which may improve performance.
- In the Dataflow Gen2, the update method for new data needs to be “Replace”.

18. Prepare data for loading into a dimensional model

- 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.

19. Design and implement a loading pattern for streaming data

- See topic 29.

Ingest and transform batch data

20. Choose an appropriate data store

- 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.
- 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

21. Choose between dataflows, notebooks, KQL, and T-SQL for data transformation

	Dataflow Gen 2	Notebook	Warehouse (T-SQL)
Primary developer skill set	ETL, M, SQL	Spark (Scala, Python, Spark SQL, R)	SQL
Use case, including data ingestion	Data ingestion, transformation, data profiling, and data wrangling	Data ingestion, transformation, data profiling, and data processing	Data storage, transformation, reporting and analytics
Primary developer persona	Data engineer, data integrator, business analyst	Data engineer, scientist, developer	Data warehouse developer, data architect, data engineer, database developer
Development interface	Graphical - No code/low code using Power Query	Code using Notebook, Spark job definitions	Code using SQL scripts
Sources	150+ connectors	100s of Spark libraries	Using New Dataflow Gen2 or data pipeline
Destinations	Few	100s of Spark libraries	Data Warehouse, Power BI Service, other using Pipeline
Transformation complexity	Low to high: 300+ transformation functions	Low to high: support for native Spark and open-source libraries	Low to High: SQL-based transformations for structured data

For KQL, see topic 29.

22. Create and manage shortcuts to data

- 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.

23. Implement mirroring

- You can mirror external databases into Fabric, from:
    - Azure Cosmos DB,
    - Azure SQL Database,
    - Azure SQL Managed Instance,
    - Snowflake.
- This allows for near real-time copying of data.
    - This includes replication of inserts, updates and deletes.
    - It also includes an autogenerated SQL Analytics Endpoint, together with a default semantic model.
- There is also automatic mirroring of Fabric SQL database.
- You can also have metadata mirroring from Azure Databricks.
    - This mirrors names, schemas, and table structures
    - This uses shortcuts to access the data from Fabric.
- To create a mirrored database:
    - go to your workspace,
    - click on “+New Item”,
    - enter “mirror”, and

    - select the appropriate object.
- For Azure SQL Database:
    - To connect to the server, the System Assigned Managed Identity (SAMI) of the Database server must be ON, and the primary identity.
    - You also need a login and mapped database user to connect to the database.
- For the example of an Azure SQL Database:
    - Click on New – Azure SQL Database.
    - Enter the server and database name.
    - Change the Authentication kind and credentials if necessary (if using Entra ID, choose “organizational account”, and sign in with the relevant account).
- For the example of the Azure Cosmos DB:
    - In Networking, Connectivity method should be All networks.
    - In backup, continuous backup needs to be enabled – either 7-day (which is free) or 30-day.
    - You will need to Primary Key of the Cosmos DB from Settings – Keys.
    - To create a sample database, go to Data Explorer – and “Launch quick start”.
    - You can then select what data is to be mirrored.
        - You can also check “Automatically mirror future tables”
    - Unsupported column types will not be mirrored.
    - Click Connect, then Create mirrored database.

24. Ingest data by using pipelines

- 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.

25. Transform data by using PySpark, SQL, and KQL

- See the "DP700CodeUsed" for examples in SQL and KQL.
- You can load raw data into the lakehouse, then refine it and enhance it with other data, and then expose it to business users using a Medallion Architecture.
    - Raw data is loaded in a bronze layer.
    - Refined and enriched data can then be saved in a silver layer.
    - Data can then be exposed to business users in a gold layer.
        - This is usually not the entirety of the data, as business users may not need all of the columns or tables, and may need summarized data.
        - You can have multiple gold layers, exposing different datasets for different user groups.

PySpark

- 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 #.
- 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.

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)

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.

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.

Convert data types

- 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
- 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
-922337203685477580 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

Filter data

- To filter data, you can use 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).

26. Denormalize data

- For KQL and SQL, see DP-600 "20. Denormalize data"
- For PySpark, the 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)
```

27. Group and aggregate data

SQL

- See the "DP700CodeUsed" document for examples.

KQL

- summarize by GroupCol1, Col3 = GroupCol2, …
    - This groups by the GroupCols.
- summarize Aggregation[, Col2 = Aggregation, …] by GroupCol1, Col3 = GroupCol2, …
    - This groups by the GroupCols, using the Aggregations (which could be renamed).
- avg(expression) – Mean Average of expression.
- avgif(expression, condition) - Mean Average of expression when condition is true.
    - Use == to compare in a condition
    - = is for assigning a value
- count – Number of rows
    - | count
    - | summarize CountRows=count() by Category
- countif(condition) – Number of rows when expression is true
- dcount(expression) and count_distinct(expression)
    - Counts the number of distinct values.
    - dcount gives an approprimation, but is quicker than count_distinct.
    - dcount can also use an optional second "accuracy" argument from 0 (less accurate) to 4 (more accurate). The default is 1.
    - Recommend using dcount, as count_distinct is not recognized by Azure Data Explorer.
- dcountif and count_distinctif filters on the second argument.
- max(expression) and min(expression) give the maximum and minimum value.
    - maxif and minif filters on the second argument.
- sum calculates the total.
    - sumif also filters on the second argument.

Aggregation	Description
avg, avgif	Mean average
count, countif	Counting the number of rows
dcount, dcountif	Distinct counting, eliminating duplications
max, maxif	The biggest value of numbers, text, datetime and bool
min, minif	The smallest value of numbers, text, datetime and bool
sum, sumif	The total

PySpark

- For 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
- You can also merge data using a Dataflow Gen2 (in the Power Query environment) by using Home – Group By, just as in Power BI.

28. Handle duplicate, missing, and late-arriving data

28a. Duplicate data

SQL

- To identify duplicate data in SQL, you can GROUP BY the data, then use a HAVING COUNT(*)>1

KQL

- You can identify duplicate data by using a summarise, then a count() > 1. This is equivalent of Having.
    - MyTable
    - | summarize Count = count() by ID, Name, Value
    - | where Count > 1
- To remove duplicate data, then either use:
    - | summarize, or
    - | distinct.

PySpark

- 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()*
- In SQL, you can use SELECT DISTINCT, or the GROUP BY clause.
- 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.

28b. Identify and resolve missing data

KQL

- To identify null values, you can use:
    - | where isnull(column_name).
- To identify missing data, you can join between two (or more table), and use:
    - | where isnull(column_name) for one of the tables, or
    - join use leftsemi or rightsemi to find data in one table which is not in the other table.

PySpark

- 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.

28c. Identify and resolve late-arriving data

- For late-arriving data:

  - In Event Stream – you can use the Offset in the Group By operator – see topic 33.
  - For Activator, see topic 37 (late arrival tolerance)
- If data is late-arriving then:
  - Power BI rules are evaluated every hour, and evaluates all events that arrive a maximum of one hour after the event occurs.
  - For other items such as eventstreams, the "Advanced settings" may have a "Wait time for late-arriving events".
    - You can set it for 1, 2 or 5 minutes.
    - This configures the balance between waiting for any potential late-arriving events, or processing data more swiftly, though it may be incomplete.

Ingest and transform streaming data

29. Choose an appropriate streaming engine

- Eventstreams are used for capturing, transforming and routing real-time events to a destination.
  - The "transforming" part means that the data can be transformed before it is written to the destination, allowing for some analysis, joining/unioning of data, and removal of unnecessary columns and rows (filter).
  - It uses a no-code interface, like a pipeline.
  - "Real-time" doesn't necessarily mean huge quantities of data. It allows you event-driven actions rather than schedule-driven.
- Eventhouses and KQL databases are good for processing data in motion.
  - Ideal for time-based, streaming data.
  - It can use structured, semi-structured, or unstructured data.
  - Data is automatically indexed, and is partitioned based on when it has been ingested.
  - This results in fast analysis, and allows for complex analysis as well.
  - Eventhouse data can be made available into OneLake, so other Fabric objects and use the data.
- KQL querysets allow you to query data, and save the queries for later use and sharing.
  - You can also create Power BI reports for it.

Feature	Eventstream	Eventhouses/KQL Database	Spark Structured Streaming
Purpose	Ingests and **processes** high-velocity data streams, acting as an entry point for real-time data into Fabric.	**Stores** and queries processed streaming data using Kusto Query Language (KQL) for real-time analytics.	Processes real-time data streams using structured queries within the Spark framework.
Data Processing	Captures and optionally **transforms** incoming event data before routing to destinations like **KQL Databases** or Lakehouses.	Performs real-time analytics on ingested data, supporting complex queries and aggregations.	Provides scalable and fault-tolerant stream processing, treating live data streams as unbounded tables for continuous query execution.
Integration	Integrates with various data sources and sinks within Fabric, facilitating seamless data flow.	Serves as a destination for **Eventstream** outputs and integrates with other Fabric components for analytics.	Integrates with Fabric's Lakehouse and other storage solutions for both batch and streaming data.
Use Cases	Real-time data ingestion from IoT devices, logs, or applications requiring immediate processing.	Interactive querying and analysis of streaming data, such as monitoring dashboards and alerting systems.	Complex event processing, real-time analytics, and machine learning on streaming data.
Query Language	Utilizes KQL for defining transformations and routing rules within the stream.	Employs KQL for querying and analyzing stored data.	Uses DataFrame and Dataset APIs with support for SQL-like queries in Python, Scala, and Java.
Scalability	Designed to handle millions of events per second, ensuring low-latency processing.	Optimized for high-concurrency queries on large datasets, providing quick responses.	Scales horizontally to process large volumes of streaming data with low latency.

29a. Choose between native storage, followed storage, or shortcuts in Real-Time Intelligence

- EventhouseDatabase, BikeEventhouse
- Native storage in Real-Time Intelligence is data which has been ingested.
- Database follower shortcuts and shortcuts create pointers to other storage.
    - It means that data is not copied or duplicated, with potential problems when data is updated.
- Database shortcuts (follower)
    - A database shortcut in Real-Time Intelligence is a reference in a KQL database from another KQL Database or from an Azure Data Explorer database.
        - The KQL Database needs to be available in OneLake.
    - This allows you to write KQL code which queries both the current KQL database and the shortcut.
    - It is read-only.

- They must be in the same region, but can be in different tenants.
- Changes in the source are reflected in the shortcut, but it may take a few seconds to a few minutes.
- The shortcut views the data in the same storage account that the source uses, and does not ingest it.
- You can create a shortcut by:
  - In the source:
    - if it is a KQL database, click on "Copy URI" next to "Query URI" (not next to "Ingestion URI") – next to the database, not next to the table.
    - enter the URI of the Azure Data Explorer database.
    - If you are creating a shortcut from an Azure Data Explorer database, you could Share – Invitation token instead.
  - In the destination KQL database:
    - Click on the + next to KQL databases.
    - Enter a new for the shortcut, change the type to "New shortcut database (Follower)", and click Next.
    - Select the Method as Cluster URI, and enter the URI.
      - If you are sharing an invitation token, then select Invitation Token as the Method, and paste the token.
    - Change the Database source, and optionally modify the default cache policy (days).
- You can then query the followed database in KQL by using:

```
database('KQLDatabaseName').table('TableName') OR

database('KQLDatabaseName').TableName
```

- Shortcuts (a single table)
  - You can create a OneLake shortcut in a KQL database which points to:
    - Microsoft OneLake,
    - Amazon S3 or S3 Compatible sources,
    - Azure Data Lake Storage Gen2,
    - Dataverse (from the Power Platform), and
    - Google Cloud Storage.
  - It creates it as an external table. You can only connect to one table or subfolder at a time.
  - To create a shortcut:
    - Go to New – OneLake shortcut,
    - Select the source type,
    - If the source type is external, then enter the Connection settings.

        - Select the Source and click Next,
        - Check the table and click Next.
        - You can turn on/off "Query acceleration" for some data sources (for example, OneLake and Azure Data Lake Storage Gen2).
            - It indexes and caches some data into an Eventhouse.
        - Then Create.
    - To edit the "Query acceleration" policy:
        - click on the … next to the Shortcut, and go to Data policies,
        - You can turn Query acceleration on/off, and change the Caching period from 1 to 36,500 days.
    - You can query the data using:

```
external_table("DimProduct")
```

30. Process data by using eventstreams

- Eventstreams can have the following sources (and you can use multiple data sources):
    - Microsoft Azure sources
        - Including Event Hubs, Service Bus, IoT Hub, Data Explorer.
    - Microsoft Azure Database sources using Change Data Capture (CDC)
        - including Databases for SQL, PostgreSQL, MySQL, Cosmos DB, Managed Instance, and SQL Server on a Virtual Machine.
    - External sources:
        - Google Cloud Pub/Sub (Publisher/Subscriber), a messaging service for events
        - Amazon Kinesis Data Streams,
        - Confluent Cloud Kafka,
        - Apache Kafka, and
        - Amazon Managed Streaming for Apache Kafka.
    - Discrete events:
        - Azure Blob Storage events,
        - Fabric OneLake, Workspace Item and Job events
    - Sample data sources
        - Bicycles,
        - Yellow Taxi, and Stock Market (high data-rate)
    - Custom eventpoint.
- Once a datasource is added to an Eventstream, a new stream is created, which you can view in the Real-Time hub.
- You can see a preview of the data/transformed data in the "Test result" pane by clicking on "Refresh",

- You can also see any "Authoring errors".
    - When transforming data, you may see an "error". This tab may show that you haven't configured it, or you eventually need a Destination.
- You add transformations by clicking on the "Add transformation" box, or going to Home – Transform events.
    - You can edit edges (the notes which connect for example a source to a transform) by clicking on them.
    - You can also have (for example) a stream connect to multiple transform events at the same time by creating the Transformations, and dragging the Edge from the stream to the Transformation.
    - You can change the name of the Transformation by clicking on it and going to the pencil icon (for Edit) and entering a new name, and clicking Save.
- You can transform the data using:
    - Aggregate (calculations)
        - Create SUM, AVG, MIN and MAX functions.
        - You can have multiple aggregations.
        - You can create a single result, or partition by a field.
        - You can also aggregate values recently received, within the last X days, hours, minutes, seconds or milliseconds.
        - If you need more complex options, see Group By.
    - Expand
        - Create a new row for each value in an array.
        - You can choose create/don't create row for missing/empty array.
    - Filter
        - For example, IS NULL or IS NOT NULL
    - Group by
        - You can calculate Average, Count, Maximum, Minimum, Percentile, Standard deviation, Sum and Variance.
        - You can also group the aggregations by a field, and specify how recently the values would have arrived: days, hours, minutes, seconds or milliseconds or microseconds.
        - You can also specify an offset, to ignore the last X seconds. This could be useful to cope with late-arriving data.
        - You can specify what type of time window it is going to be (in the following, I will use "X seconds" to denote a time period, though it could be milliseconds, for example). See topic 33 for details.
    - Join
        - Bring together two streams based on a condition.
        - You can use an inner join or left outer join.

        - You may want to use a "Manage fields" transformation to rename the columns or remove some columns.
    - Union
        - Bring together two streams, combining all rows.
        - Only includes columns with the same names and data type. Other columns are dropped (you could Manage fields before the Union, if you want to change their name).
    - Manage fields. This allows you to select what fields to output from the operation.
        - Any fields you do not add will not be included in its output.
        - You can add existing fields by clicking on "+ Add field" and expanding "Imported Schema".
            - You can also add on "Add all fields".
            - After they have been added, you can also change their Name and Type.
                - All Data Types can be converted to Strings.
                - Int64 and Double can also be converted to the other data type.
                - Strings can be converted to Int64, Double or Datetime, but only it can be converted if it is appropriate (e.g. you cannot convert "one" into an Int64).
        - You can also add additional column, using the following functions:
            - Date/Time functions: SYSTEM.Timestamp() – the current date and time, Year, Month, Day, DateAdd, DateDiff, DateName and DatePart
            - String functions: Upper/Lower, Len, LTrim/RTrim/Trim, Nchar/Unicode, Reverse, CharIndex, Left/Right/Substring, RegexMatch, Replace and Replicate
            - Mathematical functions: Abs, Ceiling, Exp, Floor, Sign, Square, Sqrt, Round and Power
- Once you have added a transform, you can edit it by clicking on the pencil icon.
    - You can also click the "Delete" icon.
- You can insert a node by hovering over a connector and clicking the "Insert a node" icon.
    - You can also click the "Delete" icon.
- Eventstreams can have the following destinations:
    - Eventhouse,
        - This is used for high quantities of data.
        - You can create a KQL database within an eventhouse.
    - Lakehouse,
    - Activator (monitor), and

  - Custom endpoint.
  - If you have added transformations, then you can also add a derived stream, which can be routed to multiple Fabric destinations, including the Real-Time hub.
- To start the Eventstream, click on “Publish”.
  - Enter the details of the Destination, together with the Input data format.
  - After an eventstream has been published, after a minute or so you may also have to click on “Activate” on the destination.
- In the KQL_queryset, use:
- Stream2
- | summarize NoBikes = avg(left_No_Bikes) by left_BikepointID, left_Neighbourhood, right_Window_End_Time
- | order by left_Neighbourhood asc, right_Window_End_Time asc

31. Process data by using Spark structured streaming

- Spark structured streaming allows you to incorporate real-time data streams into a lakehouse using PySpark.
- First, get the schema for the new lakehouse table.
  - This can easily be done by loading a file into the lakehouse, loading it into a dataframe, retrieving the schema, and creating a structure with the query.
  - Then you can delete the table.
- Next, load the streaming data with the specified schema.
- Then write the stream to a Delta Lake table.
  - The checkpointLocation is needed to manage the writing of the data from the stream.
  - .start() is needed to being writing the stream
- Finally, stop the streaming query to finish.
- You can then run the code in your notebook.
  - Once the code has initially run, you can query the table in (for example) the SQL analytics endpoint.
- However, instead of requiring a notebook to be continuously running, you can use a Spark Job Definition instead.
  - Copy the code into a .py text file on your machine.
  - Add the following code to the beginning:
    - from pyspark.sql import SparkSession
    - spark = SparkSession.builder.getOrCreate()
  - This code is implicitly run in a Notebook, but not in a Spark Job Definition.
  - Go to the Workspace, and click on “+New” and Spark Job Definition.
  - In the “Main definition file”, click on “Upload a local file” and Import the .py file from your computer.

- In the "Lakehouse reference", select the appropriate lakehouse.
- Save the Spark Job Definition and then click on Run.

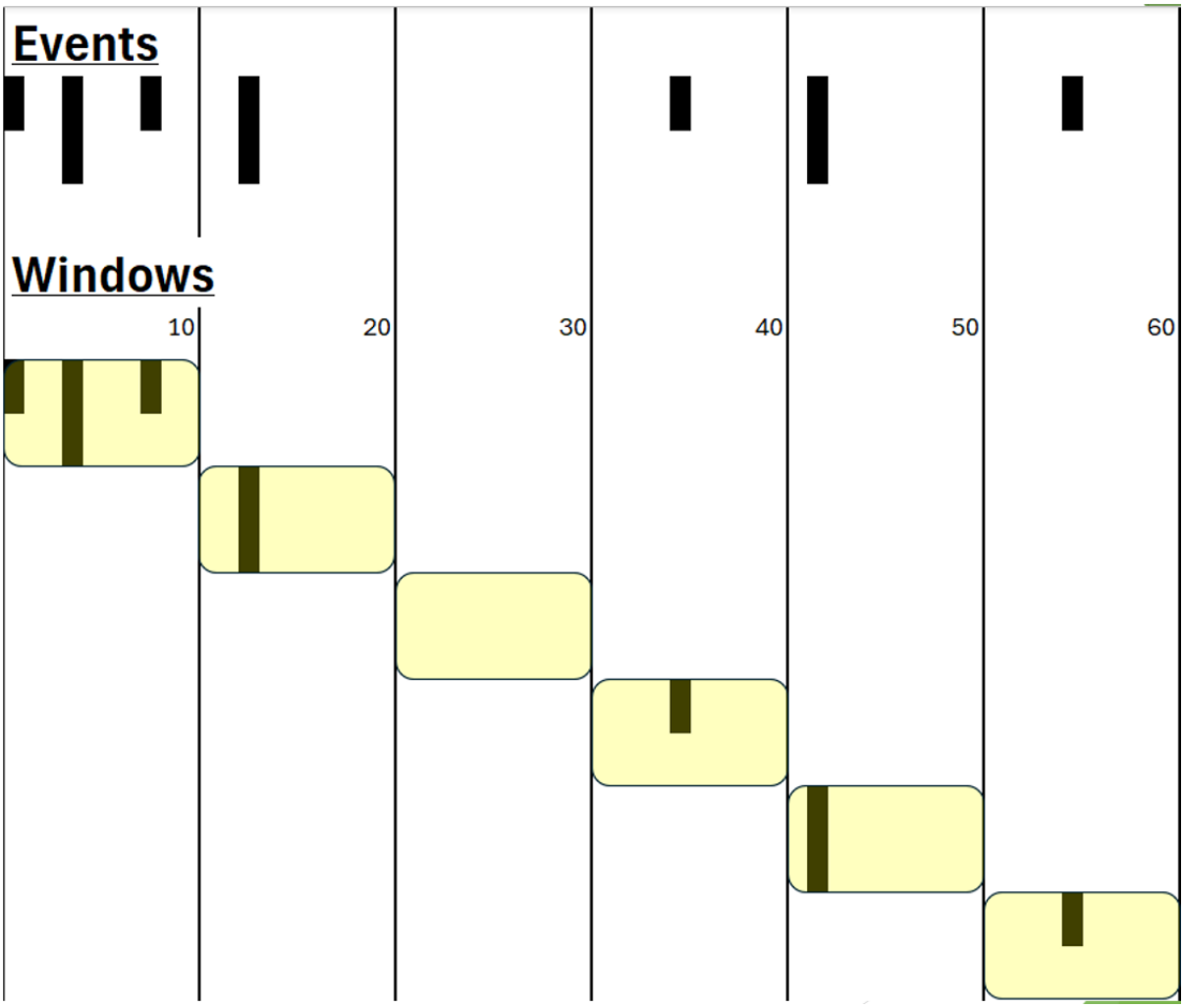
- You can see the job starting in the Message tab, and the runs in the Run tab.
  - You can click on the … next to the Run to stop the job.

32. Process data by using KQL

- See the "DP700CodeUsed" document for examples.

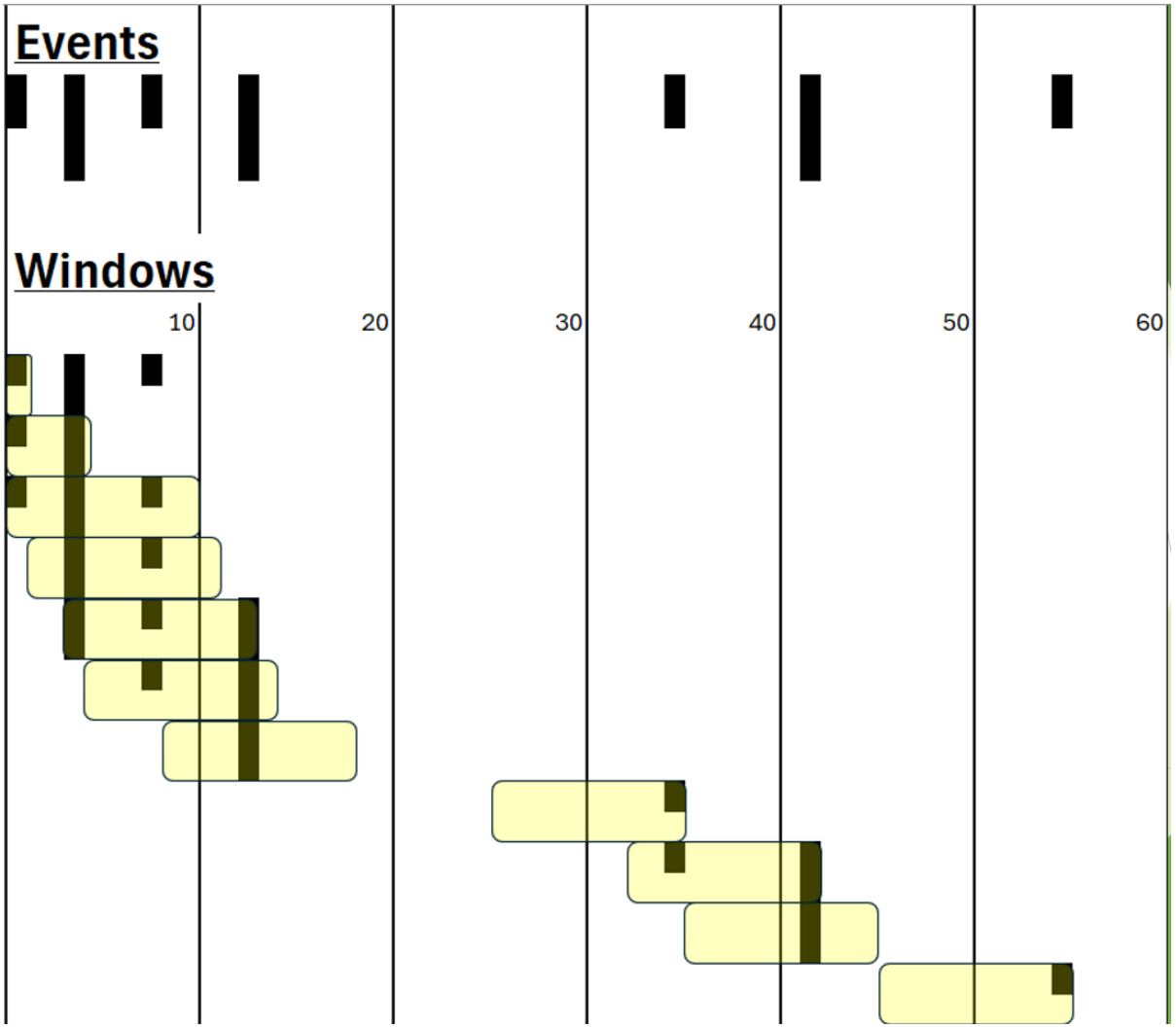
33. Create windowing functions

- Tumbling window
  - Windows are of a fixed length, last the same number of seconds, and one starts when another ends.
  - Windows cannot repeat or overlap, and an event only belongs to a maximum of one tumbling window.

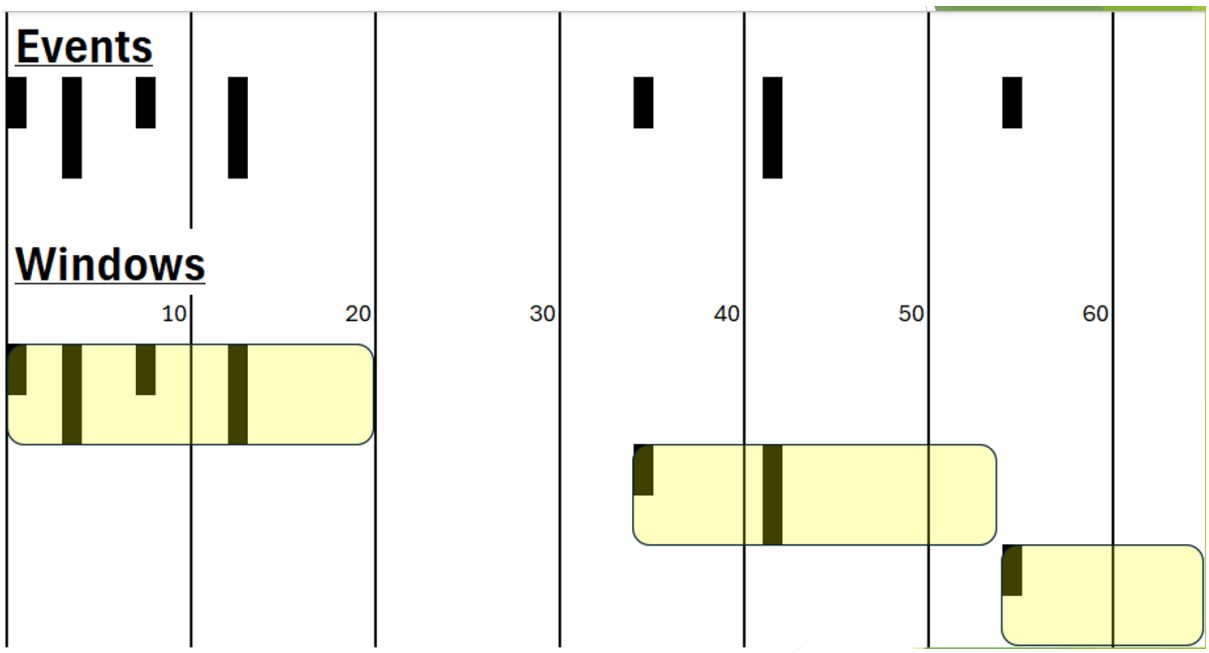


- Hopping window
    - Create a window every timeperiod (say, every X seconds).
    - The window could be bigger than the hop, and so windows can overlap, and events can be in multiple windows.
    - A tumbling window is a Hopping window, where the hop size is equal to the window size.

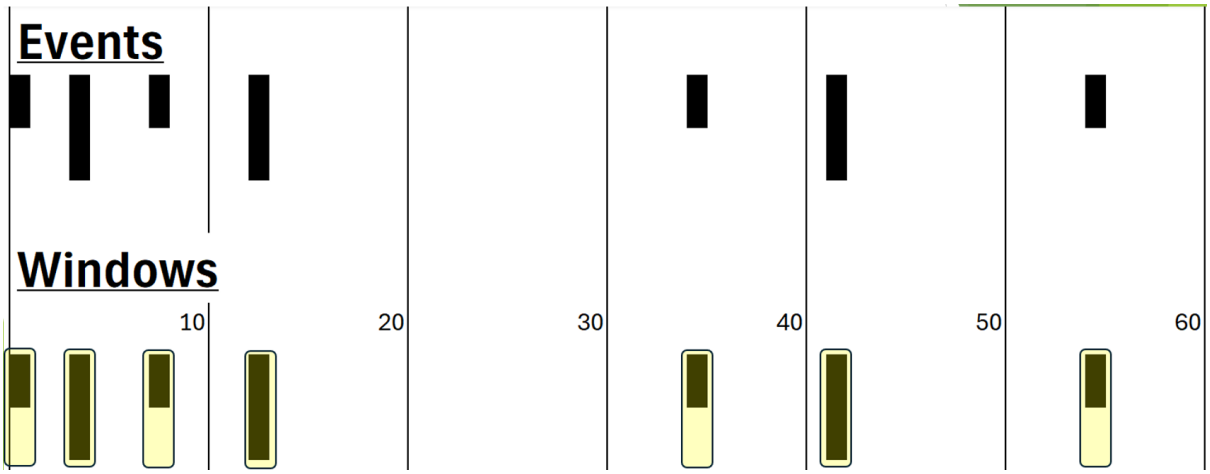
- Sliding window
    - Create a new window when an event enters or leaves a window (as long as there is one event left in the window).
    - Events can belong to more than one window.
    - Every window has at least one event.



- Session window
    - A session window groups events with a similar time.
    - It starts when an event happens.
    - It ends X seconds after the last event has happen, or when the window becomes Y seconds long in total.
    - No window occurs when no events happen.



- Snapshot window
    - Groups events that have the same timestamp.



Monitor Fabric items

34-35. Microsoft Fabric Capacity Metrics app

- 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

34. Monitor data ingestion

Spark Job Definition

- To monitor Spark Job Definition runs, click on the Runs tab inside the item.

KQL databases

- To monitor streams going into KQL databases, open the database to see the ingestion over the last 7 days.
    - If you click on the table view, you can also see:
        - Row count,
        - Original/compressed size,
        - Last ingestion,
        - Caching,
        - Retention,
        - Whether it is available in OneLake, and
        - Created on.

Data pipelines

- 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.

Monitoring hub

- 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.
    - If you click on a specific pipeline run, you can click on view more information, and click on the Input and Output icons.
    - 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".

- A more central location is the Monitor hub – click on Monitor on the left-hand side. It displays activities for:
    - Data pipelines,
    - Dataflow Gen 2,
    - Datamarts,
    - Lakehouses,
    - Notebooks,
    - Semantic models and
    - Spark job definitions.
- Activities are initially sorted by start time, but can be re-sorted.
- You can also filter for a keyword (in the text box), or click the Filter button to filter for:
    - Status (cancelled, succeeded, failed, in progress, not started or unknown)
        - More details can be found by clicking on the "View detail" for that activity.
    - Item type,
    - Start time,
    - Submitted by, and
    - Location (Workspace).
- Once a filter is applied, you can click on "Clear all" to remove the filters.
- The initial columns are the above together with "Activity Name". You can click on "Column Options" to add/remove/rearrange columns.
- You can click on the item to go either to that item or the workspace which contains it.
- You can click on the i symbol next to the activity name open the Details side-panel.
- You can also click on the … next to the item to:
    - Open the item
        - For Spark Job Definition, this allows you to look at Jobs, Resources Used, Logs, Data and Item snapshots.
    - View detail, or
    - See Historical runs.
        - This displays up to 30 days for that item.
        - You can export or refresh that data.

35. Monitor data transformation

- To monitor your Dataflow Gen2, click on the … next to it, and go to "Refresh history". This shows:
    - Start time,
    - Status,
    - Duration and
    - Type.

- Initially, it shows the first 50 refresh histories up to 6 months back.
    - This can be expanded to up to 250 refresh histories up to 6 months back.
- The refresh history can be downloaded to a CSV file by clicking on "Download as CSV".
- You can drill down into a particular refresh by clicking on it. It shows:
    - Status of the dataflow,
    - Type of refresh,
    - Start/End time,
    - Duration,
    - Request ID, Session ID and Dataflow ID.
- There is a section for Tables and Activities. For more information, you can click on a particular table/activity.
- For data pipelines, see topic 34. Click on it and go to "Recent runs".
    - You can then click on "Go to monitoring run" for more details.
    - To make changes to your pipeline, click on "Update pipeline".
    - To see information about a specific activity, you can click on it.
- Also, see topic 34.

36. Monitor semantic model refresh

- To monitor a semantic model refresh:
    - Click on the … next to the semantic model in the Workspace, and
    - Go to Refresh history.
- There you will see tabs for:
    - Scheduled (upcoming refreshes)
    - OneDrive
        - This is if your semantic models/reports are based on a Power BI Desktop file, Excel workbook, or .csv file on OneDrive or SharePoint Online.
        - By default, this checks whether a refresh is needed every hour.
        - You can switch it off by clicking on the … next to the semantic model, going to Settings, then Semantic models, and expanding the "OneDrive refresh" node.
        - Note: it doesn't refresh data in the Power BI Desktop source file – it just refreshes the existing data. If you want to refresh the Power BI Desktop source file as well, you need to do an on-demand refresh.
    - Direct Lake and
    - OneLake integration.
- You can also go to the Refresh summary History:
    - Go to Settings – Admin portal,
    - Then go to Capacity settings – Refresh summary.

- This shows:
    - Schedule
        - This tab is useful for working out if the refresh schedules are separated from each other, thus reducing a spike in workload.
        - It is separated into 30 minute slots.
        - The “Refresh time booked (minutes)” shows the total number of minutes due to happen within that slot (either starting in or going into that slot).
            - If this is over 30 minutes for a 30 minute slot, you may have too many things happening then.
        - The “Refresh time available (minutes)” shows the remaining time.
    - Refresh history
        - You can refresh or export the History to a csv file.
- By clicking on the columns, you can sort ascending/descending, add a text filter, or clear all filters.

37. Configure alerts

- See topic 15.

Identify and resolve errors

38. Identify and resolve pipeline errors

- To identify a pipeline error, you need to monitor it. See topic 34 for details.
    - You can also manually monitor it, by:
        - going to Run – “View run history” inside the Pipeline, or
        - clicking on the … next to the Pipeline and going to “Recent runs”.
        - You can then click on “Go to monitor”, or click on a run for more details.
        - In a particular run, you can click on an Activity name for more details, including the Activity status.
            - You can also filter the table, add/remove columns, and export either the current page or the top 1,000 activities to a .csv file.
    - You should note the status to see whether it succeeded, failed, in progress, cancelled or queued.
        - If it failed, and had multiple activities, then you can see the list of activities to see whether the problem was.
        - If an activity was not completed (failed, timed out, or was cancelled), then you can click on “Rerun from failed activity” to continue it.
        - You can also click on “Update pipeline” to edit it.
- If the problem is Fabric capacity performance, then you can check it by going to the Compute tab in the Microsoft Fabric Capacity Metrics app.
- To resolve a pipeline error, you can add an activity “On Fail”. To do this:
    - Add an activity, such as an Office 365 activity, to send an email on failure, notifying somebody of the failure.

        - You could also add a Teams activity.
    - Connect it to a preceding activity by dragging from the "X – On fail" to that activity.
- You can also add a link between activities:
    - Upon Skip – if the activity did not run,
    - Upon Completion – after the activity ran, either successfully or resulting in failure.
        - This should be used if the success or otherwise of the preceding activity was not critical.
    - Upon Success – the preceding activity was successful.

39. Identify and resolve dataflow errors

- 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,
        - Request, Session and Dataflow IDs.
    - Click on an activity, and you can see:
        - The start and end date/time,
        - The duration, and
        - The Volume processed (bytes/rows read/written).
- 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.
- 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.

40. Identify and resolve notebook errors

- To monitor a notebook run, see topic 34.
- You can view runs which failed from an error.
- You can click on those runs, view the problem, and correct as necessary.
- You can also manually run a notebook, and see and resolve any errors which arise from this manual run.
- You can store your data in partitions (see topic 44).
- 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.

41. Identify and resolve eventhouse errors

- See topic 47.

42. Identify and resolve eventstream errors

- See topics 30 and 47.

43. Identify and resolve T-SQL errors

- General T-SQL knowledge.

Optimize performance

44. Optimize a lakehouse table

- 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.
- 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'
- 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 apply V-order to maximize reading speeds in Fabric (but reduce writing speeds).
            - 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.

45. Optimize a pipeline

- In a data pipeline, you can add a copy activity.
- In the Settings tab, you can select:
- Intelligent throughput optimization.
    - This is a combination of CPU, memory, network resource allocation, and expected cost
    - Choose from Auto (which is dynamic based on the sources and destinations), Standard, Balanced and Maximum.
    - You can also check “Use custom value” specify a value between 4 and 256.
- Degree of copy parallelism
    - This is similar to the maximum number of threads, operating in parallel.
    - You can override the default value, and give a whole number between 1 and 32.
    - At run time, the copy number uses this as the maximum number – it may use less.

- Fault tolerance – what happens if there are errors while copying. You can skip
    - Incompatible rows,
    - Missing files,
    - Forbidden files, and
    - Files with invalid names.
- Enable logging – log copied files and skipped files and rows,
- Enable staging and Staging account connection (advanced).
    - Staging is useful if you want to copy data:
        - to/from Azure Synapse Analytics via PolyBase,
        - to/from Snowflake, or
        - From Amazon Redshift/Hadoop Distributed File System (HDFS).
    - And
        - if, for security reasons, you want to use only ports 80 and 443 (as opposed to port 1433, which is the normal port for Azure SQL Database/Synapse Analytics), or
        - If it takes a while due to a slow network connection.
    - However, as it costs more money, only use staging in a pipeline copy activity when you need to do so.

46. Optimize a data warehouse

- To copy data into a warehouse from an Azure storage account, for the fast throughput, you can use the COPY INTO command.
    - This starts with COPY INTO name_of_table. It could be followed by the columns in brackets/parentheses.
    - It then continues with FROM and the Azure storage account in single quotation marks.
    - It is often then followed by lots of options in a WITH ( ).
    - In Fabric, you can copy Parquet and CSV files from an Azure Data Lake Storage Gen2 account into a Fabric Warehouse.
    - See https://learn.microsoft.com/en-us/sql/t-sql/statements/copy-into-transact-sql for more details.
- Other possibilities are:
    - Data pipelines for no-code/low-code way to import large amount of data, or data on a schedule.
    - Data flows for data which can be transformed before being stored in the data warehouse, and
    - cross-warehouse ingestion, using a second data warehouse in the same Microsoft Fabric workspace.
        - You would then specify the second warehouse as follows:
            - CREATE TABLE FirstWarehouse.Schema.Table AS

        - SELECT * FROM SecondWarehouse.Schema.Table
- It is advisable to avoid INSERT statements which insert a single row if possible.
- A data warehouse caches data using a Solid State Disk (SSD) and memory.
    - Additionally, the internal statistics may be out-of-date, and Fabric would need to update them.
    - For these reasons, the first few executions of a query will be slower than subsequent queries.
    - If the first run's performance is crucial, you can manually create statistics
        - CREATE STATISTICS Stat_Name ON Table_Name (Columns) WITH FULLSCAN;
    - And then update them with:
        - UPDATE STATISTICS ON Table_Name (Columns) WITH FULLSCAN;
    - However, if the first run's performance is not crucial, you should keep the automatic statistics.
- INSERT, UPDATE and DELETE statements run as a single transaction.
    - If they fail, they will be rolled back, which could take a long time.
    - If possible, divide the statements into several statements with smaller amounts of data.
- Use a star schema (instead of a snowflake schema) with fact tables and dimension tables.
- Reduce the query size where possible.
    - The SQL Query editor has a maximum of 10,000 rows.
    - If you need more, use an external program like SQL Server Management Studio (SSMS) or Azure Data Studio.
- Use the smallest data type for your columns. For example:
    - smallint instead of bigint, if the values will fit.
    - VARCHAR instead of CHAR, unless you have a column with a specific length.
    - Numbers instead of strings, if possible.
    - Integers instead of float-point numbers if possible, as SORT, JOIN and GROUP BY work more quickly on integers compared with alternatives.
        - For example, if a column contains the number of kilograms with a decimal place (1.234 Kg), could it instead contain grams without a decimal place (1234 g)?
- You should use Direct Lake mode where available.
- To check how the Warehouse is being used, use the Microsoft Fabric Capacity Metrics app.

47. Optimize eventstreams and eventhouses

- To optimize an eventstream:
    - Either:

        - Click on the … next to the eventstream in the Workspace and go to Settings, or
        - In the Eventstream, click on the wheel icon in the Home tab.
    - In the "Event throughput" tab, you can change the estimated amount of data coming in and out.
        - The eventstream will then optimized for that level.
    - You have these choices:
        - Low (less than 10 Mb per second),
        - Medium (between 10 and 100 Mb per second), and
        - High (over 100 Mb per second).
        - Note: Event throughput can only be increased, not decreased.
    - In the "Retention" tab, you can change the retention period from 1 day (exactly 24 hours) up to 90 days.
- To view a Eventhouse dashboard:
    - Click on the eventhouse in your workspace.
- You can see in a green bubble next to "System overview" the status:
    - "Running" means running optimally.
    - "Maintenance" means temporarily unavailable.
    - "Suspended capacity" (this requires your capacity admin to reverse this),
    - "Unknown" (unavailable for unknown reasons), and
    - "Missing capacity" – because the Fabric compute capacity has been exceeded.
- You can also see the Eventhouse storage:
    - Standard (cold) storage – less used data.
    - Premium (hot) storage
    - Eventhouse size (compressed) – the cold and hot storage combined.
- The default consumption is "On demand". However, you can change that by clicking on the "Minimum consumption", and choose from a range from:
    - "extra extra extra small" to "extra extra large", or
    - custom (which is larger than Extra extra large).
    - These CUs will always be used (and therefore charged for).
    - In addition to an additional minimum CU level, you will also receive a Solid State Drive (SSD) capacity. The free storage level is:
        - 20 Gb for extra extra small,
        - 200 Gb for extra small,
        - 800 Gb for small,
        - Between 3,500 and 4,000 Gb for medium,
        - Between 5,250 and 6,000 Gb for large,

        - Between 7,000 and 8,000 Gb for extra large, and
        - Between 10,500 and 12,000 Gb for Extra extra large.
    - You can compare this with the "Eventhouse size".
    - You can Refresh the view or click on a KQL database to view it.
    - You can also see:
        - Activity in minutes – the compute operations (not identical to CUs) over the last hour/day/week/month,
            - If two queries are running at the same time taking 6 and 8 minutes, the activity would be 14 minutes, but the CUs would be running for 8 minutes.
        - Most queried databases over the last hour/day/week/month,
            - Useful for seeing which databases are taking the most resources.
        - Eventhouse details,
        - Activity in minutes – Top 5 users over the last hour/day/week/month,
        - What's new – recent eventhouse events in the last 7 days:
            - Creating/delete a database/external table,
            - Creating/altering/deleting a table/materialized view/function,
            - Altering a caching/retention/table update policy.

48. Optimize Spark performance

- See topic 44 for partitions, V-Order, Optimize Write, OPTIMIZE and VACUUM in a lakehouse.
- When importing data, define a schema explicitly if possible.
    - This means that Spark doesn't have to work out what the schema could be based on the data.
    - Additionally, you can specify the data types, which may be better than the inferred version. The types are:
        - BinaryType, BooleanType,
        - Numbers without decimal places: ByteType, ShortType, IntegerType, LongType
        - Numbers with decimal places: NumericType, DecimalType, DoubleType, FloatType, FractionalType
        - Strings: StringType
        - Date and Time: DateType, Timestamp Type
- Create a pool where appropriate (see topic 1).
- Use a "filter" to reduce the number of rows earlier in the formula if possible.
- Select only the columns that are needed for the query.
- You can use df.explain(True) for more details about the query execution.
- You can cache frequently used tables using:

    - spark.sql("CACHE TABLE Name_Of_Table")
- You can also enable autotune, which adjusts the Spark configuration by monitoring performance automatically.
    - It is currently in Preview, and is disabled by default.
    - It uses historic workloads to refine its configuration.
    - It will not be used for an unusually large amount of data.
    - To get the current status, use:
        - spark.conf.get('spark.ms.autotune.enabled')
    - To set it, use:
        - %%pyspark
        - spark.conf.set('spark.ms.autotune.enabled', 'true')
    - It enables:
        - spark.sql.shuffle.partitions – the number of partitions for data shuffling during joins/aggregations. The default is 200 partitions.
        - spark.sql.autoBroadcastJoinThreshold – maximum table size (in bytes) for worker nodes when joins are used. The default is said to be 10 Mb (but in my version, it is 25 Mb).
        - spark.sql.files.maxPartitionBytes – the maximum number of bytes for a partition when reading files. The default is 128 Mb

49. Optimize query performance

- You can implement Query folding in Dataflows.
- 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.
- 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.

