Dremio Software

Semantic Layer Best Practices

The semantic layer is a data management level that transforms the datasets from your data sources into a format that data consumers can use in their applications.

You will find that Dremio already provides a semantic layer for you to build upon. By default, each user has a home space where they can create a hierarchy of folders and add more spaces. Connected sources and datasets from those sources appear in your data catalog. Icons, too, appear next to each data object to show if it is a table, view, space, or source. However, this is only a starting point.

As more users from your organization use Dremio and create more data objects, it will be much easier to navigate and manage your data if you establish the proper processes and structures. To further categorize and organize these data objects, we have compiled a list of best practices in this guide for building your semantic layer.

Before starting this guide, think about:

- What are the different use cases for Dremio in your organization?
- What types of users will be accessing the semantic layer?
- Have you enabled security roles and assigned them to your users?
- Do you have a list of standard terms used by your organization?

If you have enough information to start building your semantic layer, see the following recommended best practices for making this self-service layer the most effective and efficient for your organization.

Manage Your Data Catalog

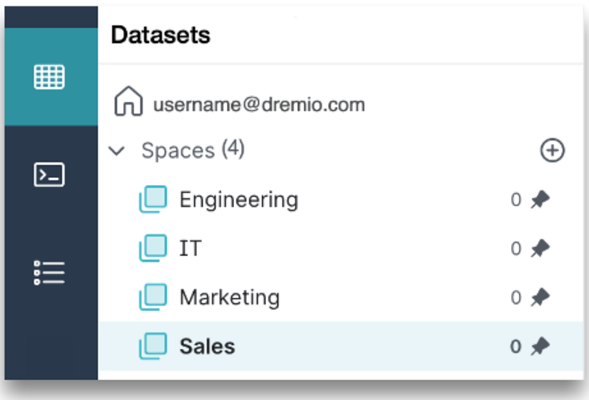
The semantic layer contains an indexed and searchable data catalog consisting of spaces, folders, tables, and views. By using the data catalog, users can query and edit datasets, maintain workspaces, and derive new views.

To help manage this data catalog, the data panel shows the hierarchy of data objects and how they relate to each other. Use this data panel to create, modify, or remove data objects in your catalog. The data catalog is also used when setting the context of data objects in the SQL editor, ensuring that the query references a specific data object from a source, space, or folder.

Organize Environments and Spaces

We recommend having distinct environments for development, quality assurance, and production. If you only have a single environment for building your semantic layer, you can use spaces instead to separate your resources.

Within each environment, use your data panel to create spaces. Spaces let you group datasets by common themes such as a project, purpose, department, or geographic region. See example below:



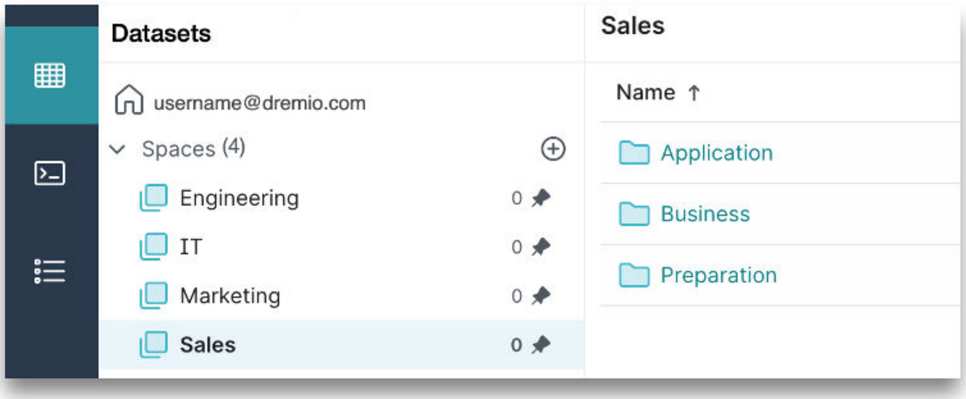
In this example, the production environment contains four spaces: Engineering, IT, Marketing, and Sales. We can then choose whether to create folders in each space as sub-layers, such as dividing Sales by region and creating folders for Sales-US, Sales-EMEA, and Sales-APAC. Within those folders, you can create more sub-layers, and so on.

Layer Your Views

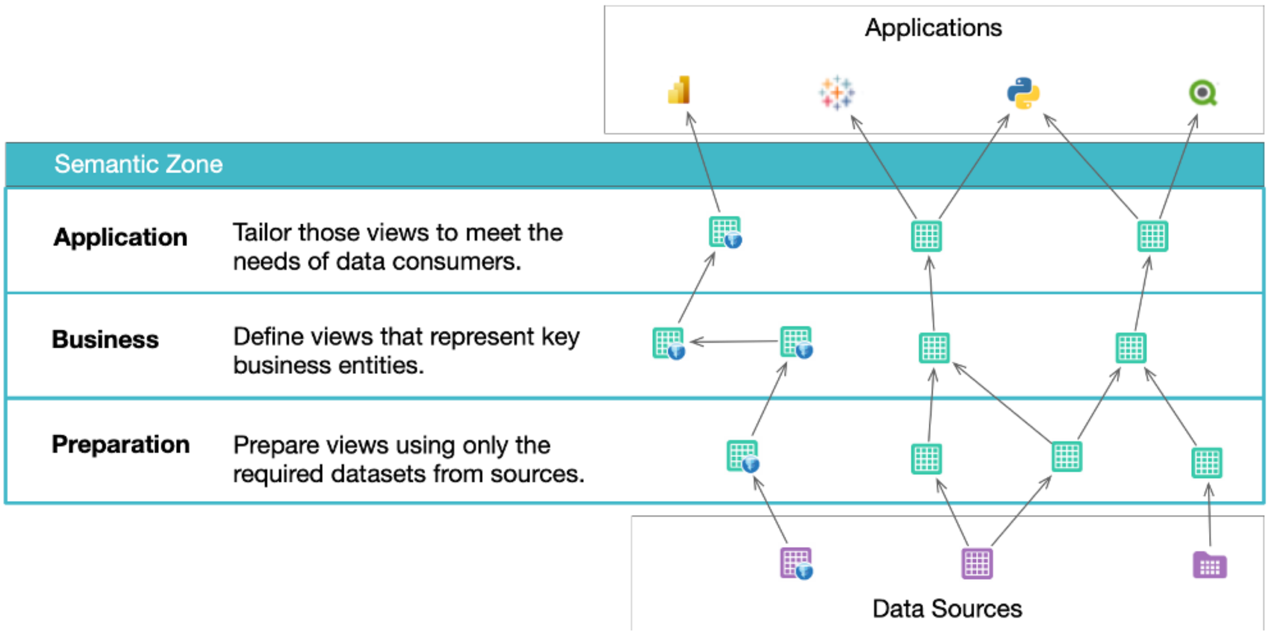
When you create sets of views in the semantic layer, we recommend organizing them into sub-layers so that they can be used many times across multiple projects, providing agility to development teams and users.

You can create sub-layers by organizing your views into three folders: Preparation, Business, and Application. These folders can be created within your designated spaces or folders, depending on how you want to build your semantic layer.

Below, you can see how the layer concepts can be mapped to a coherent space and folder structure inside your project:



Refer to the diagram below for the following descriptions of each sub-layer.

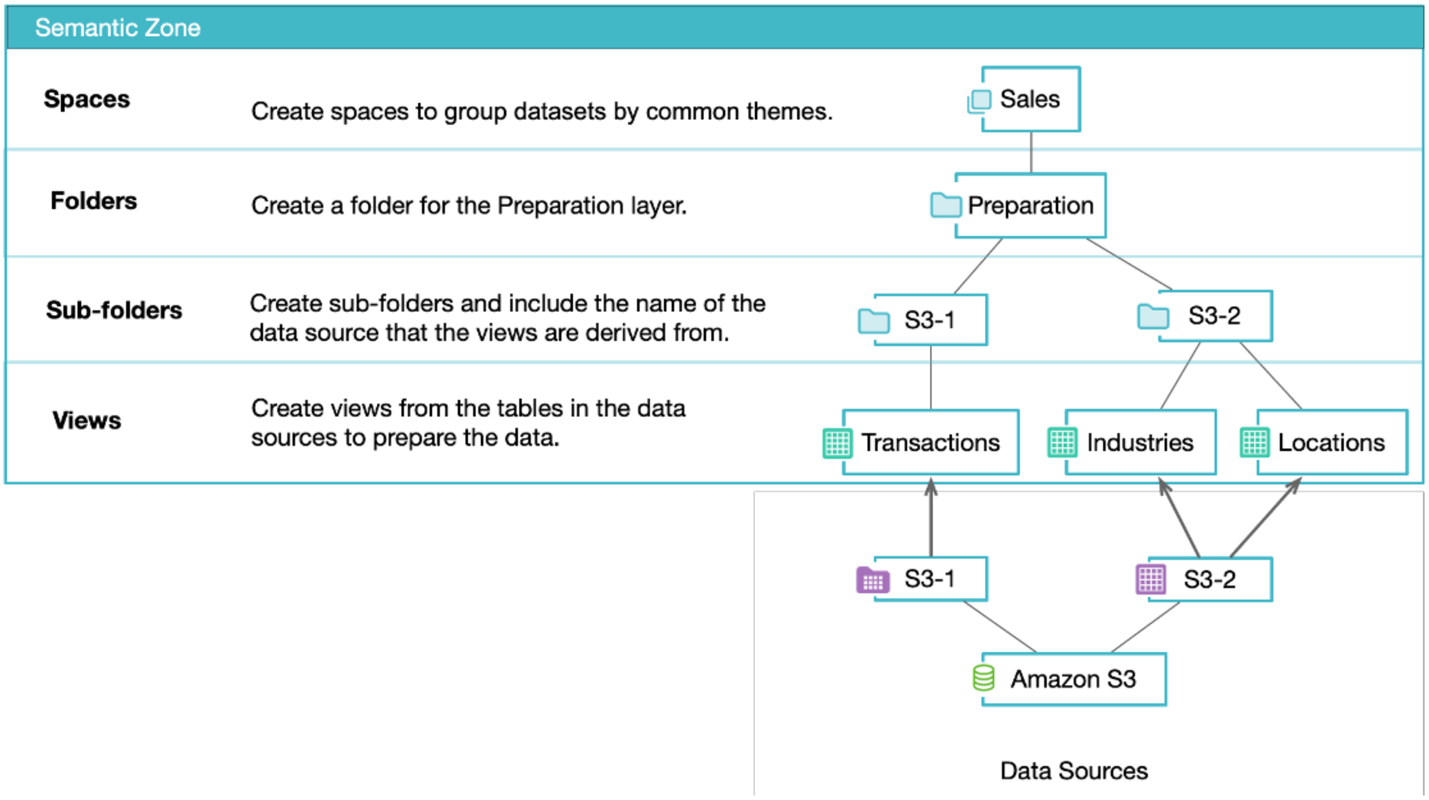


Application	Tailor those views to meet the needs of data consumers.
Business	Define views that represent key business entities.
Preparation	Prepare views using only the required datasets from sources.

Preparation Layer

The Preparation layer is closest to the data sources. This layer organizes and exposes only the required datasets from a source rather than an entire set of datasets in a source. In this layer, each view is mapped to the table that they are derived from in the data source and there are no joins to other views. Typically, a data engineer is responsible for preparing the data in this layer, where they can apply column aliasing, column data type casting, and data cleansing. They can also create some derived columns based on existing columns in the table.

We also recommend dividing this layer into sub-folders, and for each sub-folder, include the name of the data source from which the views are derived. See the diagram below for an example of the Preparation layer:


Semantic Zone
Spaces
Create spaces to group datasets by common themes.
Sales
Folders
Create a folder for the Preparation layer.
Preparation
Sub-folders
Create sub-folders and include the name of the data source that the views are derived from.
S3-1
S3-2
Views
Create views from the tables in the data sources to prepare the data.
Transactions
Industries
Locations
S3-1
S3-2
Amazon S3
Data Sources

Within the Sales space, there is a Preparation folder to contain all views related to sales that have been prepared from the S3-1 and S3-2 tables.

Business Layer

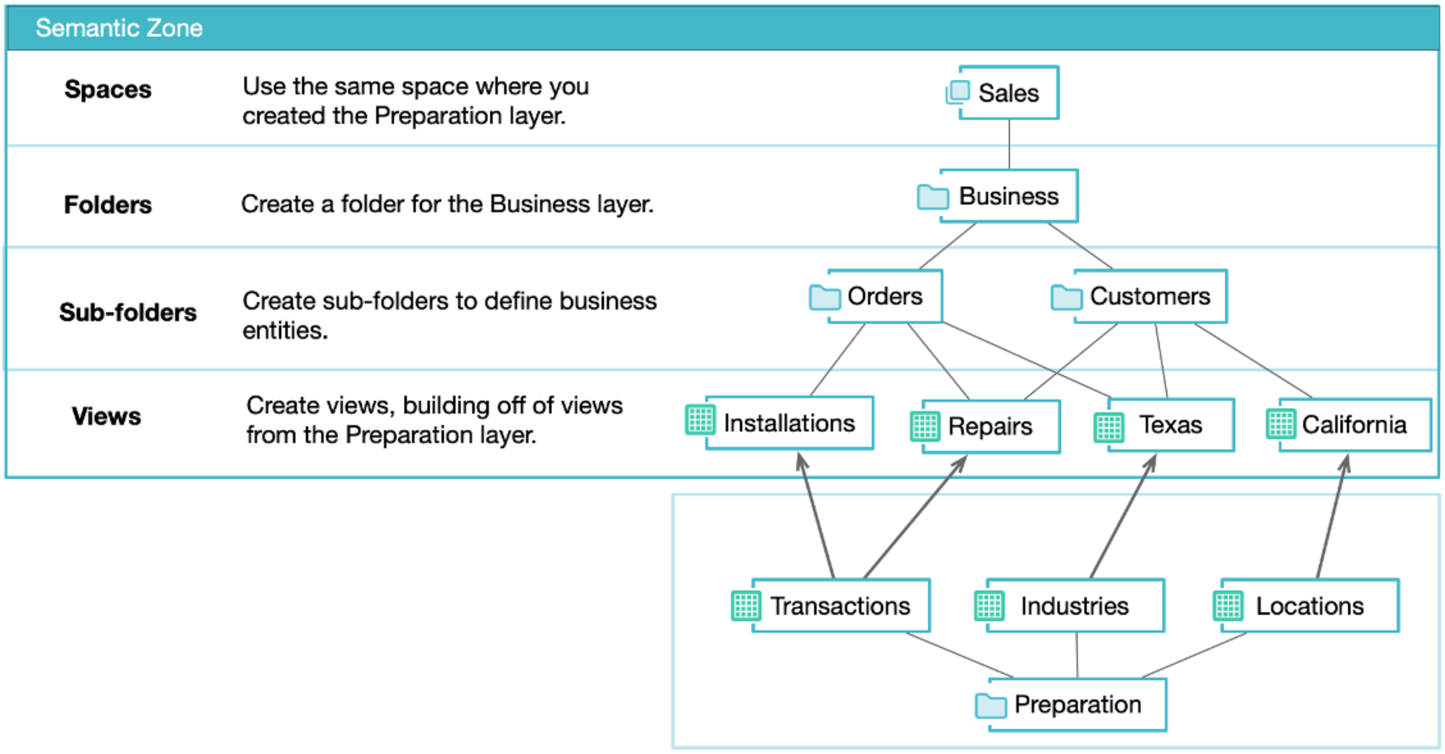
The Business layer provides a holistic view of all the data across your organization. It is the first layer where joins among and between sources should occur, and all views in this layer must be built by either:

a. Querying resources in the Preparation layer

b. Querying other resources in the same Business layer

Use your list of standard terms to describe the key business entities in your organization, such as a customer, a product, or an order. Typically, a data modeler works with business experts and data providers to define the views that represent the business entities.

You can create many sub-layers inside the Business layer, each with views for different subject areas or verticals. These views are reusable components that can and should be shared across business lines. See the diagram below for an example of the Business layer:

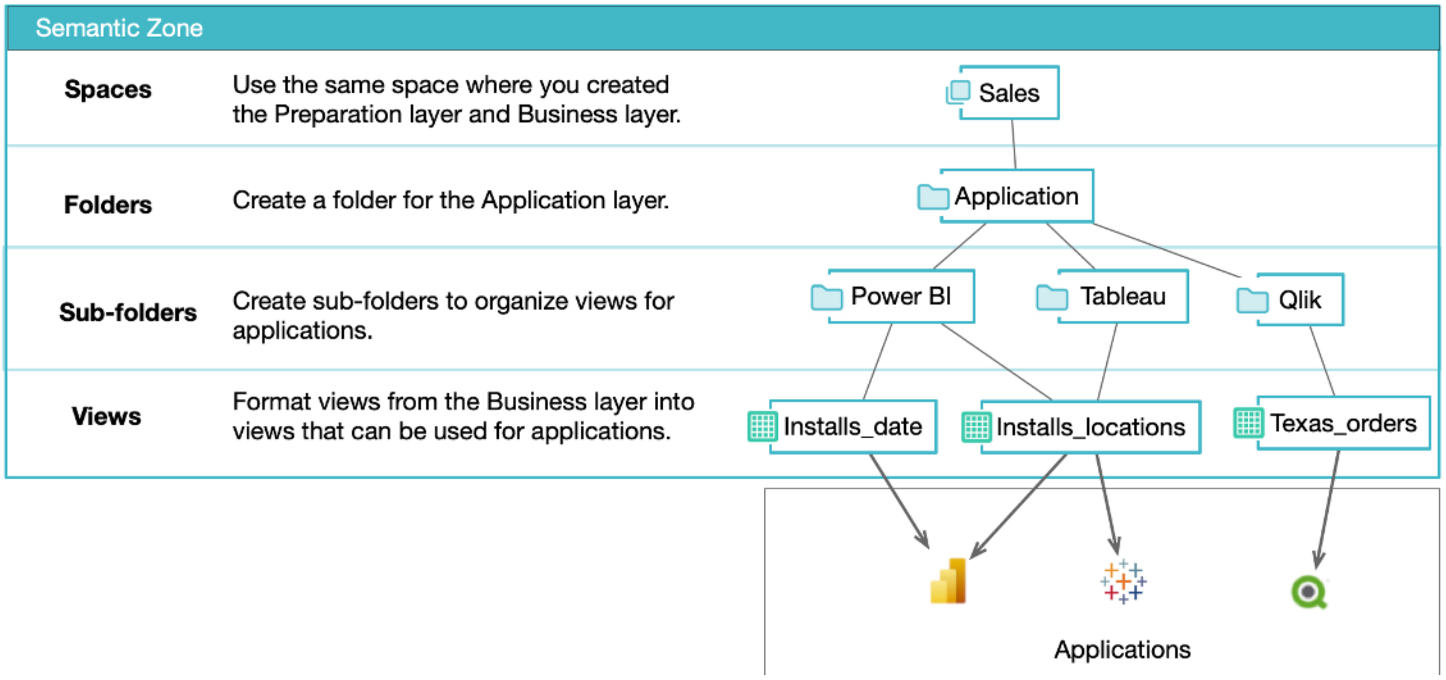

Semantic Zone
Spaces
Use the same space where you created the Preparation layer.
Folders
Create a folder for the Business layer.
Sub-folders
Create sub-folders to define business entities.
Views
Create views, building off of views from the Preparation layer.
Sales
Business
Orders
Customers
Installations
Repairs
Texas
California
Transactions
Industries
Locations
Preparation

Using this layer, you can improve productivity for analytics initiatives and minimize the risk of duplicative efforts in your organization by:

- reducing the cost of service delivery to lines of business
- providing a self-service model for data engineers to provision datasets quickly
- enabling data consumers to put those datasets to use and share with others quickly

Application Layer

After creating views in the Business layer that describe key business entities, you can then format those views in the Application layer for the needs of individual data consumers, organizational departments, etc. Typically, data consumers, such as analysts or data scientists, leverage the views from the Business layer and work directly in the Application layer to create and modify views used in their dashboards. See diagram below for an example of the Application layer:



Semantic Zone

Spaces	Use the same space where you created the Preparation layer and Business layer.
Folders	Create a folder for the Application layer.
Sub-folders	Create sub-folders to organize views for applications.
Views	Format views from the Business layer into views that can be used for applications.

Alternatively, if data consumers do not have direct access to the Dremio console, a designated developer with access to the Application layer needs to create and alter views on behalf of the data consumers via some formal change process. For change management, a developer would work with an Application SME to map the views in this layer.

You can create many sub-layers inside the Application layer to organize the resources being exposed by Dremio into subject areas or verticals. Ultimately, applications only have visibility into Application layer resources; they will not be exposed to lower-level views.

Use Role-Based Access Control

The access for each folder can be managed and regulated by roles. Roles are used to organize privileges at scale rather than manage each user's privileges individually. You can create roles to manage privileges for users with different job functions in your organization, such as *Analyst* and *Security_Admin* roles. Users who are members of a role gain all the privileges granted to the role.

Access control protects the integrity of your data and simplifies the data architecture available to users based on their roles and responsibilities within your organization. Effective controls allow users to access data that is central to their work without regard for the complexities of where and how the data is physically stored and organized.

Development Environments

For a development environment, we recommend granting privileges on each folder according to roles because the folders are at a higher hierarchical level than views. Then, work your way down through the folder to grant privileges to any subfolders and views that differ from the folder privileges.

To grant privileges for each layer folder, you can grant `SELECT` or `ALTER` privileges. `SELECT` allows users to view data on all folders, tables, and views in the layer folder. In contrast, `ALTER` enables users to edit the table or view definitions or settings of all tables and views in the layer folder. See the table below for recommended privileges based on the following roles:

Folder	**Data Engineer/SME**	**Semantic Data Modeler**	**Report Developer/Analyst/ Data Scientist**
Application	`SELECT` or `ALTER`	`SELECT` or `ALTER`	`ALTER`
Business	`SELECT` or `ALTER`	`ALTER`	`SELECT`
Preparation	`ALTER`	`SELECT`	None

In this development environment, the report developer has `SELECT` and `ALTER` privileges for the Application folder and `SELECT` privileges for the Business folder. This enables the report developer to edit only within the Application layer but still see the views and folders in the Business layer.

Note that some roles have multiple possible permissions defined as `SELECT` or `ALTER`. This is because we may want to give nested subfolders different privileges depending on who owns the folder (e.g., the owner can edit the views in a folder, but other roles can only see the views in the folder).

QA/Test and Production Environments

For a QA\test or production environment in Dremio, we recommend that roles only be granted `SELECT` privileges. Use only `SELECT` privileges as needed because changes should only be permitted in a development environment or development space. See the table below for recommended privileges based on the following roles:

Folder	**Data Engineer/SME**	**Semantic Data Modeler**	**Report Developer/Analyst/Data Scientist**
Application	`SELECT`	`SELECT`	`SELECT`
Business	`SELECT`	`SELECT`	`SELECT`
Preparation	`SELECT`	`SELECT`	None

In this production environment, the report developer has only `SELECT` privileges for the Application and Business folders but no privileges for the Preparation folder. Controlling access of the report developer enables access to views and folders central to their work without the complexity of how the data is physically stored and organized.

Build Upon Views

Build upon views in the semantic layer by adding wikis and labels to enhance discoverability and understanding of your data across your organization. You can use the wiki to describe the dataset or content that helps users get started with the data, such as examples, usage notes, or points of contact for questions or issues. Adding labels will help users identify the dataset.

Labeling views and adding wiki content helps make the semantic layer more approachable as a self-service by enabling users to quickly find the resources they are interested in using either as-is or as the basis for a new view. A user will be curious to know what resources are already available to them, so being able to search for keywords or search on table or column names can be very useful. In addition, users will want to understand in more detail the meaning of a set of views in a particular space or folder, their purpose, what data they expose etc., which is where wiki entries are especially useful.

User Experience

According to Dremio's security recommendations above, data consumers such as report developers, analysts, and data scientists do not have visibility into views in the Preparation layer, and the views in the Preparation layer are typically one-to-one mapping with their table equivalents anyway. Therefore, there is little value to data consumers in adding labels and wiki entries to the Preparation layer resources. However, data engineers and semantic data modelers will likely find labels and wiki content useful in the Preparation layer.

By contrast, the Business layer is at the heart of the self-service semantic layer, so labels and wiki entries should be used extensively in this layer. Data consumers will not be responsible for adding labels and wiki content in this layer because the semantic data modelers will do that with assistance from data engineers. Data consumers will spend a reasonable proportion of their time searching for relevant metadata in the Business layer and subsequently using the results of their findings to create new views or alter existing views in the Application layer. Data consumers should add labels and wiki entries to their views and folders in the Application layer.

Below you can see the recommended responsibilities based on folder privileges for the following roles:

Folder	Data Engineer/SME	Semantic Data Modeler	Report Developer/ Analyst/Data Scientist
Application	● Search for views ● Create and alter sub-folders and views on behalf of data consumers (if needed)	● Search for views ● Create sub-folders and views (if needed) ● Alter only sub-folders and views that they own	● Create and alter sub-folders and views for applications ● Add wikis and labels to the views that they own
Business	● Search for views ● Create sub-folders and views (if needed) ● Alter only sub-folders and views that they own	● Create and alter sub-folders and views that represent business entities ● Add wikis and labels to the views that they own	● Search for relevant metadata ● Search for views that they can build from for the Application layer
Preparation	● Create sub-folders and views to prepare data ● Add wikis and labels to the views that they own	● Search for views that they can build from for the Business layer	● None

Use Case Examples

Example 1

One set of data consumers might want to see “customer” views aggregated on field A, whereas a different set of data consumers might want to see the same “customer” views aggregated on field B.

If given the necessary access, data consumers such as analysts or data scientists can work directly in the Application layer in the Dremio console to create and modify views for use in their dashboards. They can leverage the views built inside the Business layer to help them achieve this. Joining multiple views from the Business layer to produce new views in the Application layer is a perfectly valid approach.

Example 2

The Business layer enables an additional layer of abstraction over the raw data sources that facilitate essential use cases, including data migration. Common data migrations are from an on-premise data lake to a cloud data lake and from a non-data lake source, such as a data warehouse, to an on-premise or cloud data lake.

Data consumers can be exposed to views that give them the data they need initially from the original source, while the data is being migrated into the new source in the background. Then, once the data has been migrated successfully, the view is updated to select its data from the new source. The business consumer will continue to receive the data they expect without impact. In this use case, the business consumer will always query the same views throughout the migration process. The only thing that changes in the Dremio console is where that view gets its data; this also applies to any applications that use this data.