

Organize a Fabric lakehouse using medallion architecture design

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/1-introduction>

Introduction

Completed

In this data-driven era, the need to harness vast and diverse data sources is critical for business success. The Fabric lakehouse, blending data lakes and data warehouses, offers an ideal platform to manage and analyze this data. The medallion architecture has become a standard across the industry for lakehouse-based analytics.

The medallion architecture brings structure and efficiency to your lakehouse environment. This module guides you through its bronze, silver, and gold layers, ensuring you know how to organize, refine, and curate data effectively.

Whether you're a seasoned data engineer or an analytics newcomer, this module caters to all levels of experience with practical examples and a hands-on exercise. You should be familiar with the Fabric data lakehouse and have a basic understanding of SQL and Power BI before you begin.

In this module, you'll explore and build a medallion architecture for Fabric lakehouses, query and report on data in your Fabric lakehouse, and describe best practices for security and governance of your Fabric lakehouse.

2. Describe medallion architecture

Describe medallion architecture

Completed

Data lakehouses in Fabric are built on the Delta Lake format, which natively supports ACID (Atomicity, Consistency, Isolation, Durability) transactions. Within this framework, the *medallion architecture* is a recommended data design pattern used to organize data in a lakehouse logically. It aims to improve data quality as it moves through different layers. The architecture typically has three layers – bronze (raw), silver (validated), and gold (enriched), each representing higher data quality levels. Some people also call it a "multi-hop" architecture, meaning that data can move between layers as needed.

This architecture ensures that data is reliable and consistent as it goes through various checks and changes. It also guarantees that the data is safely stored in a way that makes it easier and faster to analyze.

The medallion architecture complements other data organization methods, rather than replacing them. You can think of the medallion architecture as the framework for data cleaning, rather than a data architecture or model. It ensures compatibility and flexibility for businesses to adopt its benefits alongside existing data models, allowing you to customize data solutions and preserve expertise while remaining adaptable in the ever-changing data landscape.

Medallion Architecture



Understand the medallion architecture format

Bronze layer

The bronze or raw layer of the medallion architecture is the first layer of the lakehouse. It's the landing zone for all data, whether it's structured, semi-structured, or unstructured. The data is stored in its original format, and no changes are made to it.

Silver layer

The silver or validated layer is the second layer of the lakehouse. It's where you'll validate and refine your data. Typical activities in the silver layer include combining and merging data and enforcing data validation rules like removing nulls and deduplicating. The silver layer can be thought of as a central repository across an organization or team, where data is stored in a consistent format and can be accessed by multiple teams. In the silver layer you're cleaning your data enough so that everything is in one place and ready to be refined and modeled in the gold layer.

Gold layer

The gold or enriched layer is the third layer of the lakehouse. In the gold layer, data undergoes further refinement to align with specific business and analytics needs. This could involve aggregating data to a particular granularity, such as daily or hourly, or enriching it with external information. Once the data reaches the gold stage, it becomes ready for use by downstream teams, including analytics, data science, or MLOps.

Customize your medallion architecture

Depending on your organization's specific use case, you may have a need for more layers. For example, you might have an additional "raw" layer for landing data in a specific format before it's transformed into the bronze layer. Or you might have a "platinum" layer for data that's been further refined and enriched for a specific use case. Regardless of the names and number of layers, the medallion architecture is flexible and can be tailored to meet your organization's particular requirements.

Move data across layers in Fabric

Moving data across medallion layers refines, organizes, and prepares it for downstream data activities. Within Fabric's lakehouse, there's more than one way to move data between layers, ensuring that you can choose the method that works for your team.

There are a few things to consider when deciding how to move and transform data across layers.

- How much data are you working with?
- How complex are the transformations you need to make?
- How often will you need to move data between layers?
- What tools are you most comfortable with?

Understanding the difference between data transformation and data orchestration helps you select the right tools for the job within Fabric.

Data transformation involves altering the structure or content of data to meet specific requirements. Tools for data transformation in Fabric include Dataflows (Gen2) and notebooks. Dataflows are a great option for smaller semantic models and simple transformations. Notebooks are a better option for larger semantic models and more complex transformations. Notebooks also allow you to save your transformed data as a managed Delta table in the lakehouse, ready for reporting.

Data orchestration refers to the coordination and management of multiple data-related processes, ensuring they work together to achieve a desired outcome. The primary tool for data orchestration in Fabric is *pipelines*. A pipeline is a series of steps that move data from one place to another, in this case, from one layer of the medallion architecture to the next. Pipelines can be automated to run on a schedule or triggered by an event.

3. Implement a medallion architecture in Fabric

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/3-implement-medallion-architecture-fabric>

Implement a medallion architecture in Fabric

Completed

Now that you have a solid understanding of the medallion architecture, let's explore how to put it into action within Fabric.

- **Set up the foundation:** [Create your Fabric lakehouse](#). You can use the same lakehouse for multiple medallion architectures, or alternatively, you can use different lakehouses and even different lakehouses in different workspaces, depending on your use case.
- **Design your architecture:** Create your architecture layout, define your layers, and determine how data will flow between them. The most straightforward implementation is to use Bronze as the raw layer, Silver as the curated layer, and gold as the presentation layer. Your gold layer should be modeled in a star schema and optimized for reporting.

Question	Bronze	Silver	Gold
What happens in that layer?	Ingest raw data	Cleanse and validate data	Additional transformations and modeling
What tool is used?	Pipelines, dataflows, notebooks	Dataflows or notebooks	SQL analytics endpoint or semantic model

- **Ingest data into bronze:** Determine how you'll ingest data into your bronze layer. You can do this using pipelines, dataflows, or notebooks.
- **Transform data and load to silver:** Determine how you'll transform data in your silver layer. You can do this using dataflows or notebooks. Transformations at the silver level should be focused on data quality and consistency, not on data modeling.
- **Generate a gold layer:** Determine how you'll generate your gold layer, what it will contain, and how it will be used.
 - The gold layer is where you'll model your data for reporting using a dimensional model. Here you'll establish relationships, define measures, and incorporate any other elements essential for effective reporting.
 - You can have multiple gold layers for different audiences or domains. For example, you might have a gold layer for your finance team and a separate gold layer for your sales team. You might also have a gold layer for your data scientists that is optimized for machine learning.
 - Depending on your needs, you might also use a Data Warehouse as your gold layer.
 - In Fabric, you can transform your data using dataflows or notebooks, and then load it into a gold Delta table in the lakehouse. You can then connect to the Delta table using a SQL analytics endpoint and use SQL to model your data for reporting. Alternatively, you can use Power BI to connect to the SQL analytics endpoint of the gold layer and model your data for reporting.
- **Enable downstream consumption:** Determine how you'll enable downstream consumption of your data. You can do this using workspace or item permissions, or by connecting to the SQL analytics endpoint.

4. Query and report on data in your Fabric lakehouse

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/4-query-report-data>

Query and report on data in your Fabric lakehouse

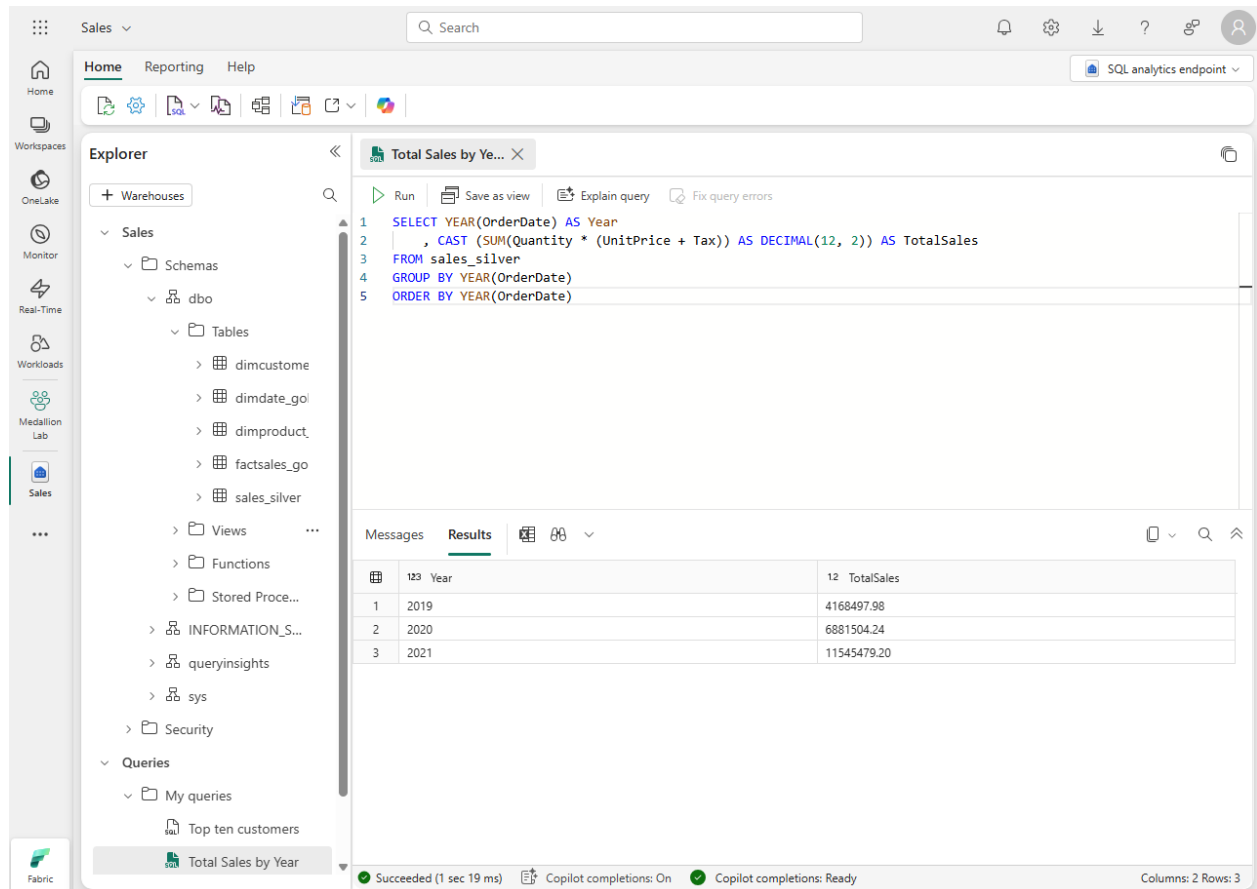
Completed

Now that your medallion architecture is in place, data teams and the business can start using it to query and report on data. Fabric has several tools and technologies that enable you to query and report on data in your lakehouse, including SQL analytics endpoints and Direct Lake mode in Power BI semantic models.

Query data in your lakehouse

Teams can use SQL to explore and query data in the gold layer. You can analyze data in delta tables at any layer of the medallion architecture using the T-SQL language, save functions, generate views, and apply SQL security. You can also use the SQL analytics endpoint to connect to your lakehouse from querying tools and applications.

The SQL analytics endpoint in Fabric enables you to write queries, manage the semantic model, and query data using the visual query experience.



The screenshot displays the Microsoft Fabric SQL analytics endpoint interface. The left sidebar shows the 'Explorer' pane with a tree view of the 'Sales' warehouse, including schemas, tables (like dimcustomer, dimdate_goi, dimproduct, factsales_goi, sales_silver), views, functions, and queries. The main pane shows a SQL query titled 'Total Sales by Year' with the following code:

```
1 SELECT YEAR(OrderDate) AS Year
2     , CAST (SUM(Quantity * (UnitPrice + Tax)) AS DECIMAL(12, 2)) AS TotalSales
3 FROM sales_silver
4 GROUP BY YEAR(OrderDate)
5 ORDER BY YEAR(OrderDate)
```

Below the query editor, the 'Results' tab shows a table with 2 columns and 3 rows:

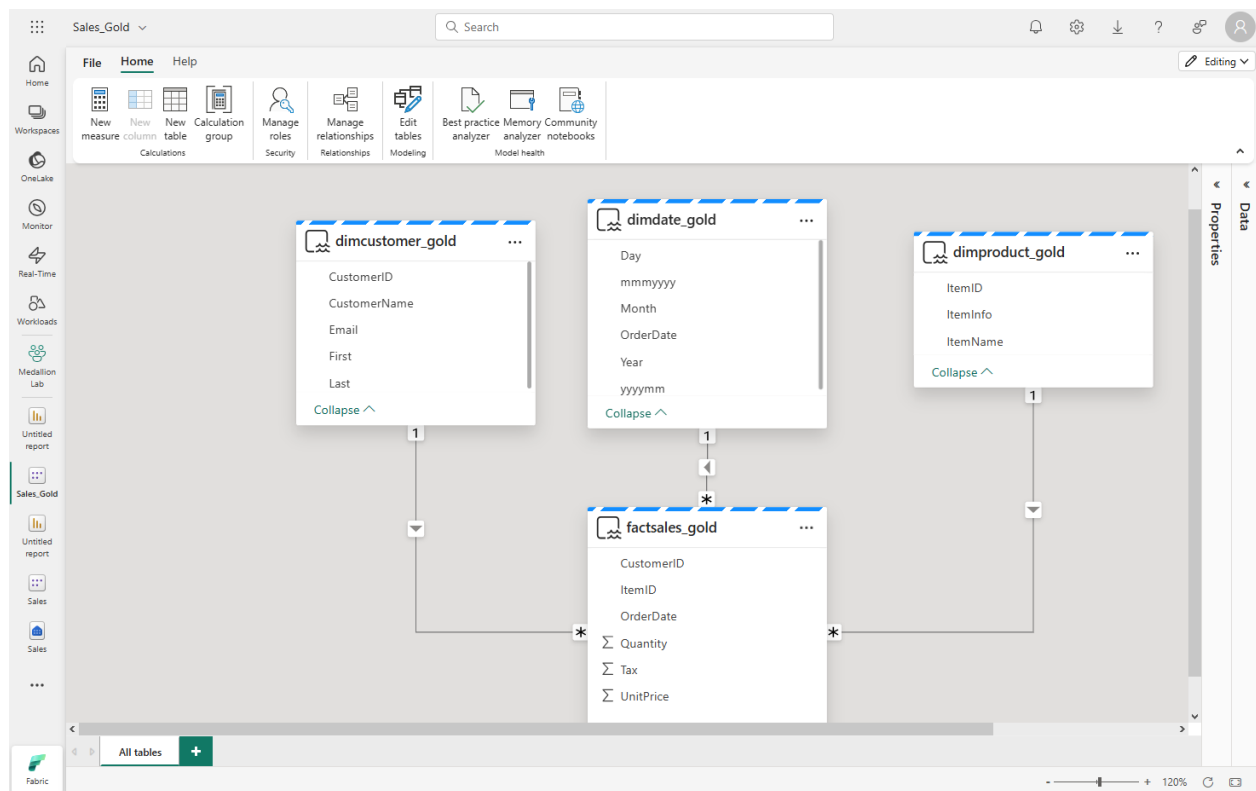
	Year	TotalSales
1	2019	4168497.98
2	2020	6881504.24
3	2021	11545479.20

The bottom status bar indicates 'Succeeded (1 sec 19 ms)' and 'Copilot completions: On'.

Note

The SQL analytics endpoint operates in **read-only** mode over lakehouse delta tables. To modify data in your lakehouse you can use dataflows, notebooks, or pipelines.

In addition to using the SQL analytics endpoint for data exploration, you can use a Power BI semantic model to explore data. A semantic model is a way of viewing data using business friendly terminology. It's typically represented as a star schema with facts that represent a domain of data relevant to a particular business area, and dimensions that allow you to analyze data in the data domain. A default semantic model is created automatically when you create a lakehouse. You can also create a non-default, custom Power BI semantic model.



Data analysts can connect to the semantic model using **Direct Lake** mode, in which the semantic model accesses data directly from Delta tables in a lakehouse.

Tailor your medallion layers for different needs

Tailoring medallion layers to different needs allows you to optimize data processing and access for specific use cases. By customizing these layers, you can ensure that each layer's structure and organization align with the requirements of different user groups, improving performance, ease of use, and data relevance for diverse stakeholders.

Creating multiple Gold layers tailored for diverse audiences or domains highlights the flexibility of the medallion architecture. Finance, sales, data science – each can have its optimized Gold layer, serving specific analytical requirements.

Some applications, third-party tools, or systems require specific data formats. You can utilize your medallion architecture to generate cleansed and properly formatted data.

5. Considerations for managing your lakehouse

Considerations for managing your lakehouse

Completed

There are several considerations when managing your lakehouse including how to secure your lakehouse and how to handle continuous integration and continuous delivery (CI/CD).

Secure your lakehouse

Secure your lakehouse by ensuring that only authorized users can access data. In Fabric, you can do this by setting permissions at the *workspace* or *item* level.

Workspace permissions control access to **all** items within a workspace. Item level permissions control access to **specific** items within a workspace, and could be used when you're collaborating with colleagues who aren't in the same workspace, or they only need access to a single, specific item.

You can strategically store different layers of your lakehouse in separate workspaces for improved security and efficient capacity management. This approach not only enhances security but also optimizes cost-effectiveness.

- **Security and Access Considerations:** Define who needs access at each layer, ensuring only authorized personnel can interact with sensitive data.
- **Gold Layer Access Control:** Restrict access to the Gold layer for read-only purposes, emphasizing the importance of minimal permissions.
- **Silver Layer Utilization:** Decide whether users will be allowed to build upon the Silver layer, balancing flexibility and security.
- **Bronze Layer Access Control:** Restrict access to the Bronze layer for read-only purposes, emphasizing the importance of minimal permissions.

Sharing of Fabric content should be discussed with your organization's security team to ensure that it aligns with your organization's security policies.

Considerations for Continuous Integration and Continuous Delivery (CI/CD)

Designing a Continuous Integration/Continuous Deployment (CI/CD) process for a lakehouse architecture involves several considerations to ensure a smooth and efficient deployment process. Considerations include implementing data quality checks, version control, automated deployments, monitoring, and security measures. Considerations should also include scalability, disaster recovery, collaboration, compliance, and continuous improvement to ensure reliable and efficient data pipeline deployments. While some of these are related to processes and practices, others are related to the

tools and technologies used to implement CI/CD. Fabric natively provides several tools and technologies to support CI/CD processes.

Git integration in Microsoft Fabric enables you to integrate development processes, tools, and best practices straight into the Fabric platform. Fabric's Git integration enables data teams to back up and version work, revert to previous stages as needed, collaborate with others or work alone using Git branches, and leverage the capabilities of familiar source control tools to manage Fabric items.

Note

Learn more about Git integration in Fabric in [Introduction to git integration](#).

CI/CD is crucial at the gold layer of a lakehouse because it ensures that high-quality, validated, and reliable data is available for consumption. Automated processes enable continuous integration of new data, data transformations, and updates, reducing manual errors and providing consistent and up-to-date insights to downstream users and applications. This enhances data accuracy, accelerates decision-making, and supports data-driven initiatives effectively.

6. Exercise - Organize your Fabric lakehouse using a medallion architecture

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/6-exercise>

Exercise - Organize your Fabric lakehouse using a medallion architecture

Completed

Now it's your turn to create a lakehouse in Fabric and move data through the medallion architecture. You'll also learn how to query and report on data in your Fabric lakehouse.

Note

You need a Microsoft Fabric trial license with the Fabric preview enabled in your tenant. See [Getting started with Fabric](#) to enable your Fabric trial license.

Launch the exercise and follow the instructions.

7. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/7-knowledge-check>

Module assessment

Completed

8. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-medallion-architecture/8-summary>

Summary

Completed

You now understand how the *medallion architecture* offers a structured approach to logically organize lakehouse data. Once you integrate this architecture into your Fabric environment, you're on your way to ensuring an organized, refined, and curated data system. You'll eliminate data silos and ensure that downstream teams and apps can access and use the data seamlessly. With this effective setup, you're not just storing data; you're setting it up to be quickly and easily analyzed by whenever it's needed.