**This is just to help during my lightning talk**

**Basically, a dialog.**

**\*\*Notes:  
- Workspace is just an environment of data assets**

**- ETL Pipeline is the Extract, Transform, Load pattern, it is basically an automatic or triggered process that moves and transforms data from one place to another**

**- Lineage refers to the full journey of data from the source to the destinations. It shows the conversions or transformations or processes applied to it.**

SLIDE 1 (The presentation)

My name is Leonardo, but some of you know me as Giorgio. Today I’m going to talk about how to bring structure to your data with databricks

SLIDE 2 (Brief explanation)

The certification I chose is the Data engineer from Databricks. And to be honest this is my first time having contact with data engineering. I chose this certification to learn something new, something I think is useful and interesting.

First, a brief explanation of databricks. Databricks is a unified data analytics platform built on Apache Spark. Basically, it combines data engineering, data science, machine learning and analytics on a single platform powered by Lakehouse architecture and it exists to empower data teams to collaborate, and analyze and innovate at scale.

SLIDE 3 (Example)

Let´s imagine that a client has multiple analysis teams within the different departments, Let’s say they have one team of data analyst in the sales departments, another in marketing, another in finances… and so on and so on. And they *are all trying to access and analyze data*

***What happens without a unified governance layer?***

* *‘Who is accessing this information?’*
  + *Nobody knows exactly who has access or why.*
* *‘Why does everyone have access to each other’s tables?’*
  + *There’s no clear boundary or control across teams.*
* *‘Why is there duplicated data everywhere?’*
  + *Because each team is making their own copies—there’s no single source of truth.*
* *‘Where’s the trusted version?’*
  + *This becomes a nightmare when you’re trying to build reliable dashboards or reports.*
* *And finally: ‘I need to go back to the previous version; I made a mistake.’*
  + *Without proper version control, there’s no easy way to recover.*

SLIDE 4 (definition)

Unity Catalog is a unified governance layer inside Databricks.  
It helps us organize, secure, and audit all our data assets across multiple workspaces

And it introduces a clear structure. So we now think in terms of: catalogs, which contain schemas, which contain tables.  
For example: we might have a catalog named sales\_data, with a schema for north\_America, and inside that schema a table called orders\_2025

* First: it give us centralized permissions
  + Permissions on files, permissions on tables and views, permissions on columns and rows, permissions on ml models, on dashboards, etc. Everything together
* Second: it enables auditing and lineage tracking
  + So we know who accessed what, when, and so we can know how data is flowing.
* Third: it makes data discovery easy
  + Because, now everything is cataloged and searchable across the entire platform and workspaces.
* Finally: data sharing
  + So we can share the data across the multiple workspaces, applications or even platforms

*SLIDE 5 (how to do it)*

*So, there’s a few key principles to keep in mind*

* *We should organize data clearly, using catalogs and schemas that reflect business domains or teams*
* *Permissions must be managed centrally*
* *Naming conventions should be consistent, so that discovery is simple and intuitive*
* *Lineage tracking and audit logs should not just exist, they should be reviewed regularly to ensure everything is in order*
* *And finally, we should treat all of this as the trusted single source of truth for all data assets, avoiding unnecessary data duplication, and confusion.*

SLIDE 6 (the organization)

Here, on the first line, we’re creating a catalog named Sales Data on our unity catalog (this is important).

Then, we’re creating a schema, or a database named North America within the catalog we’ve just created.

And finally, we created a table named Orders 2025 within the schema we’ve just created. This table automatically inherits governance, lineage and auditability through Unity Catalog.

This isn’t just visual structure, as you can see in the catalog explorer – It reflects a clean, governed data structure that Unity Catalog enforces across all workspaces.

SLIDE 7 (accessibility and permissions)

Here’s another way a data engineer works with accessibility and permissions.

For this example, we already created the analyst role and the data engineering role.

On the first line, you can see a simple SQL command that is granting SELECT permissions to the analyst role. So, every member that’s given the analyst role can make use of the information ONLY.

And, on the third line, you can see that we’re giving all privileges to the schema to the data engineering role. So, any data engineer can keep working on the governance and structure and security of data of the database of north America.

-“Explain images with the permissions”

*SLIDE 8 (Problems it solves)*

*On the left, we see that Unity Catalog gives us centralized governances, it standardizes access control, applies security to columns and rows level, audits every action, tracks lineage, and simplifies discovery across workspaces, but there are things it doesn’t solve.*

*As you can see on the right, it doesn’t magically ensure data quality, it doesn’t design the schemas for us, nor does it define the business rules or business logic. It won’t handle data transformation or automate ETL pipelines for us. And it doesn’t enforce by any way naming conventions. That’s still our job as data engineers.*

SLIDE 9 (Wrapping up)

So, Unity Catalog isn’t just about governance and access control, like you saw with the examples. It’s about good engineering purposes.

When we use unity catalog properly, we aren’t just ticking a compliance box, we are building a sold, well-structured platform that helps everyone work better.

The whole framework encourages us to apply good practices like consistency, organization, discoverability and auditability, because when we are doing everything correctly, the whole process from data ingestion to endpoints consumption or dashboard making makes sense and is intuitive.

***SLIDE 10 (Q&A)***

***If you have any questions, comments, or feedback, feel free to speak up.***

***Please, if you’ve got something on your mind, just say it, if not, I thank you for your attention. Have a nice day.***

• • • •