

DAY 08: Unity Catalog Governance

From Data Swamp to Data Fortress

The Mission: Secure Your Data Platform

Today's objective is to implement a robust governance model using Unity Catalog. We will complete four specific challenges:

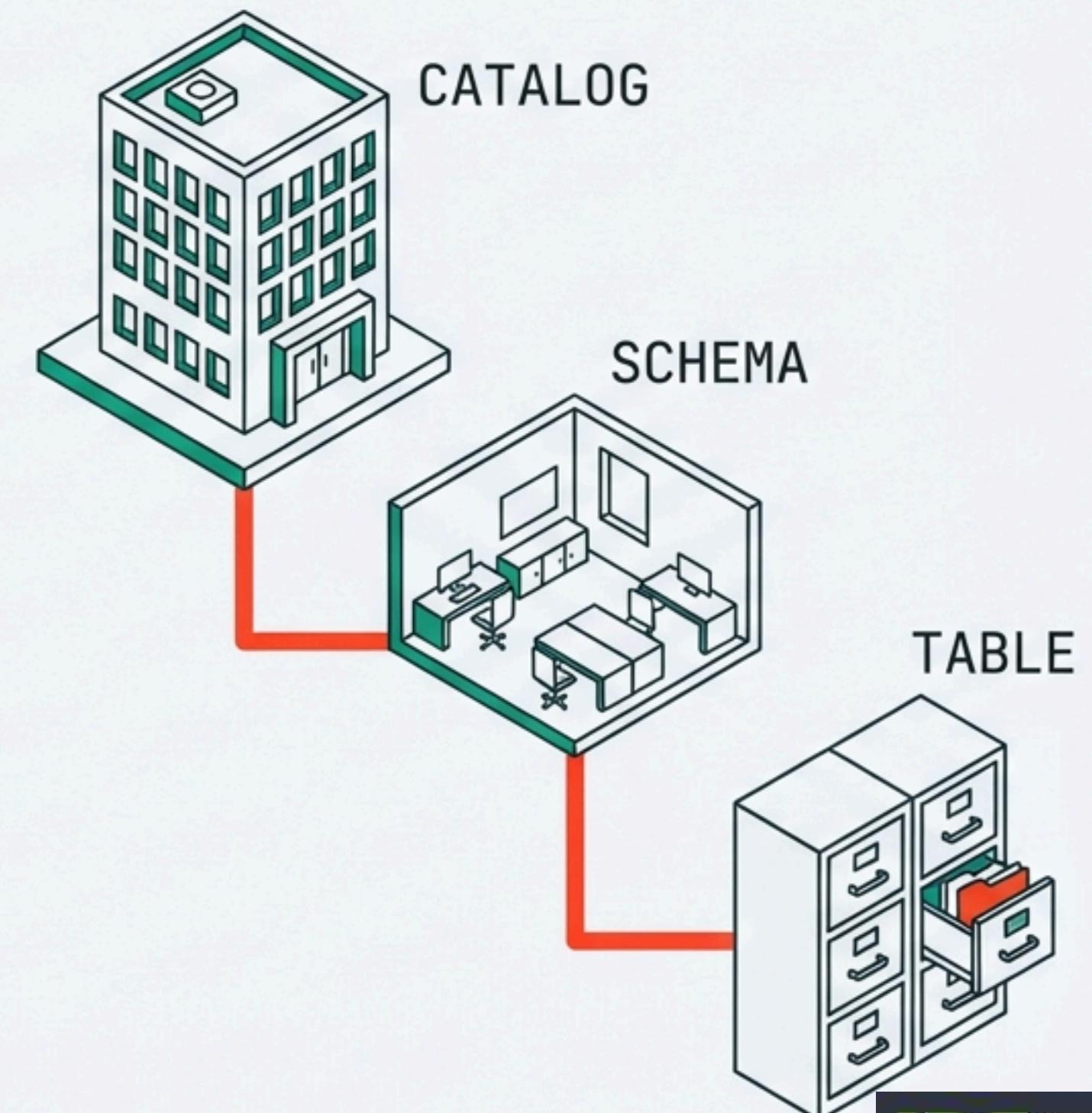
- Create Structure:** Build the Catalog and Schema hierarchy.
- Register Assets:** Map Delta tables to specific storage locations.
- Define Access:** Configure GRANT and REVOKE permissions for different personas.
- Control Visibility:** Specific views to filter sensitive data.



The Mental Model: The Three-Level Namespace

Unity Catalog organizes data using a strict hierarchy to ensure uniqueness and order.

- Level 1: Catalog. The top-level container (Think: The Building).
- Level 2: Schema (Database). Logical groupings (Think: The Rooms).
- Level 3: Table/View. The data assets (Think: The Filing Cabinets).



Addressable via: `catalog.schema.table`

Practice: Building the Foundation

-- Create structure

```
CREATE CATALOG ecommerce;
```

```
USE CATALOG ecommerce;
```

Sets active context

-- Establishing the Medallion Architecture

```
CREATE SCHEMA bronze;
```

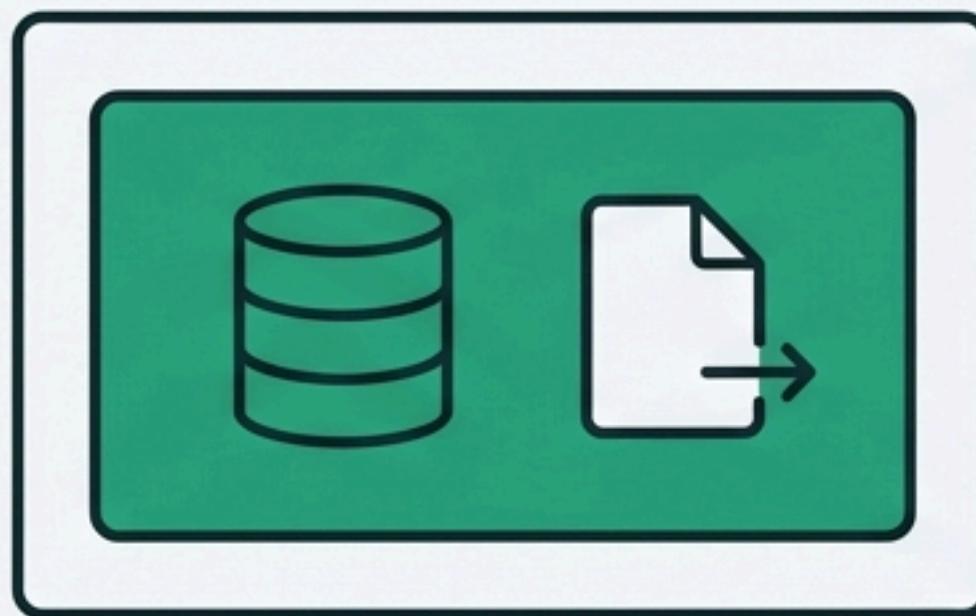
```
CREATE SCHEMA silver;
```

```
CREATE SCHEMA gold;
```

} Medallion
Architecture Setup

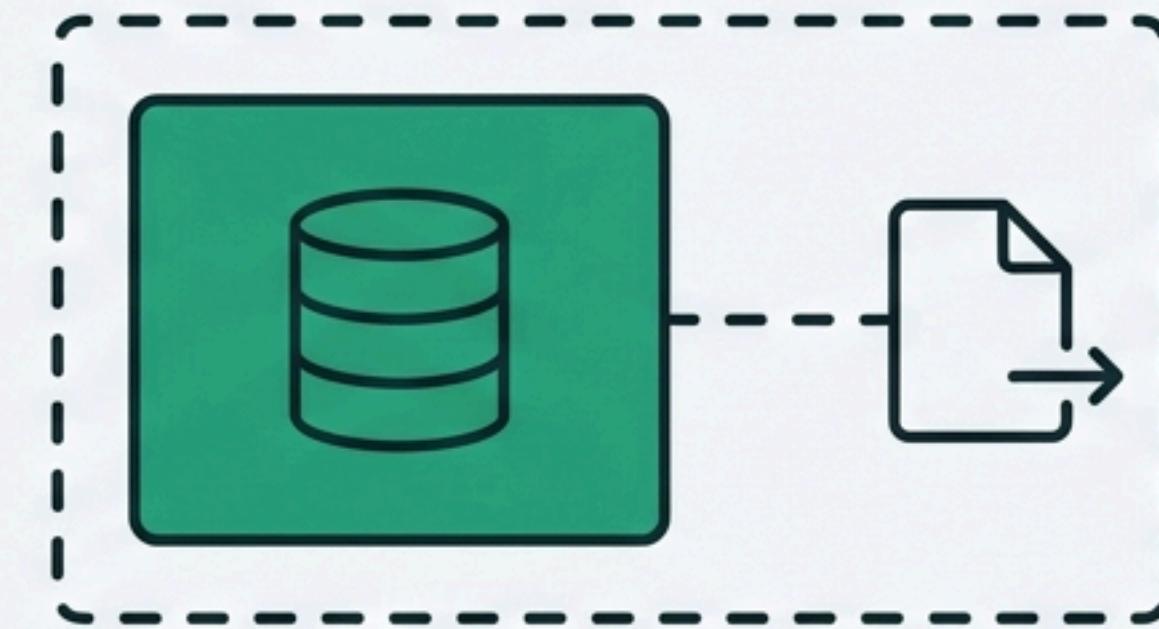
Understanding Table Types

Managed Tables



Unity Catalog manages metadata
AND data lifecycle.
“DROP TABLE” deletes the actual files.

External Tables



Unity Catalog manages metadata only.
You control the storage location.
“DROP TABLE” leaves files in storage.

In this challenge, we define specific locations, effectively using an External approach.

Practice: Registering Delta Tables

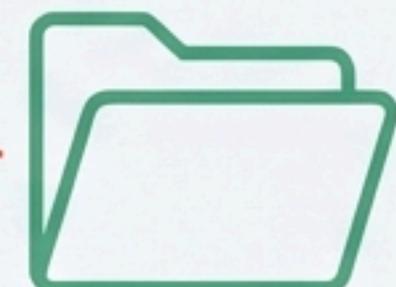
```
-- Register tables  
CREATE TABLE bronze.events USING DELTA  
LOCATION '/delta/bronze/events';  
  
CREATE TABLE silver.events USING DELTA  
LOCATION '/delta/silver/events';  
  
CREATE TABLE gold.products USING DELTA  
LOCATION '/delta/gold/products';
```



/delta/bronze/events



/delta/silver/events



/delta/gold/products

The Gatekeeper: Roles and Permissions

Security via Least Privilege



The Analyst

Needs to read business insights.
No access to raw logs.
Cannot modify structure.

GRANT SELECT



The Engineer

Needs full control to build
pipelines and fix
transformation issues.

GRANT ALL PRIVILEGES

Practice: Setting Up Permissions

-- Permissions

```
GRANT SELECT ON TABLE gold.products TO `analysts@company.com`;
```



Read Only

-- Permissions

```
GRANT ALL PRIVILEGES ON SCHEMA silver TO `engineers@company.com`;
```

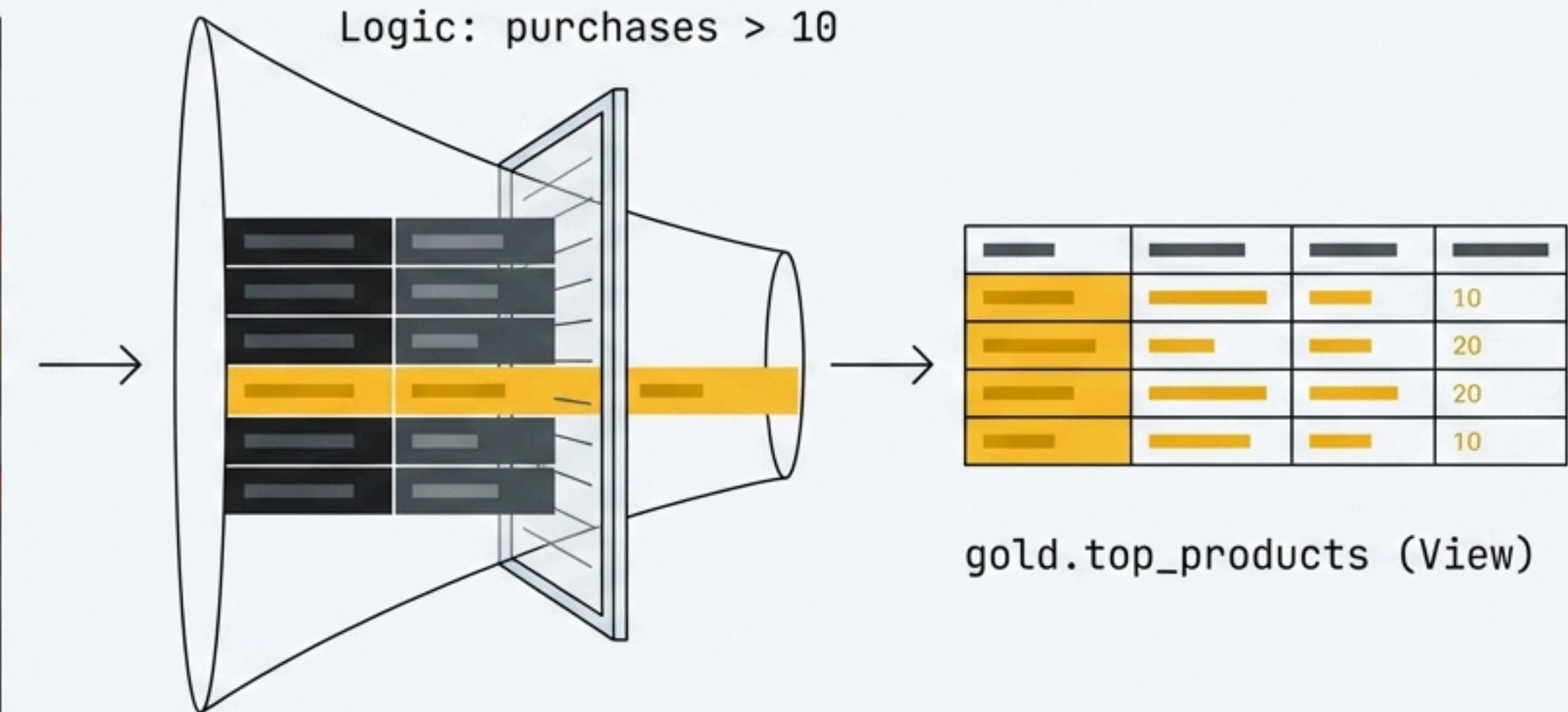


Full Control

Curating the Experience with Views

—	—	—	—	—	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	20	—
—	—	—	—	10	—
—	—	—	—	10	—
—	—	—	—	10	—

gold.products (Table)



A View is a saved query that acts as a virtual table.
It hides complexity and secures sensitive rows.

Practice: Creating Controlled Views

```
-- Controlled view
CREATE VIEW gold.top_products AS
  SELECT product_name, revenue, conversion_rate
    FROM gold.products
   WHERE purchases > 10
   ORDER BY revenue DESC LIMIT 100;
```

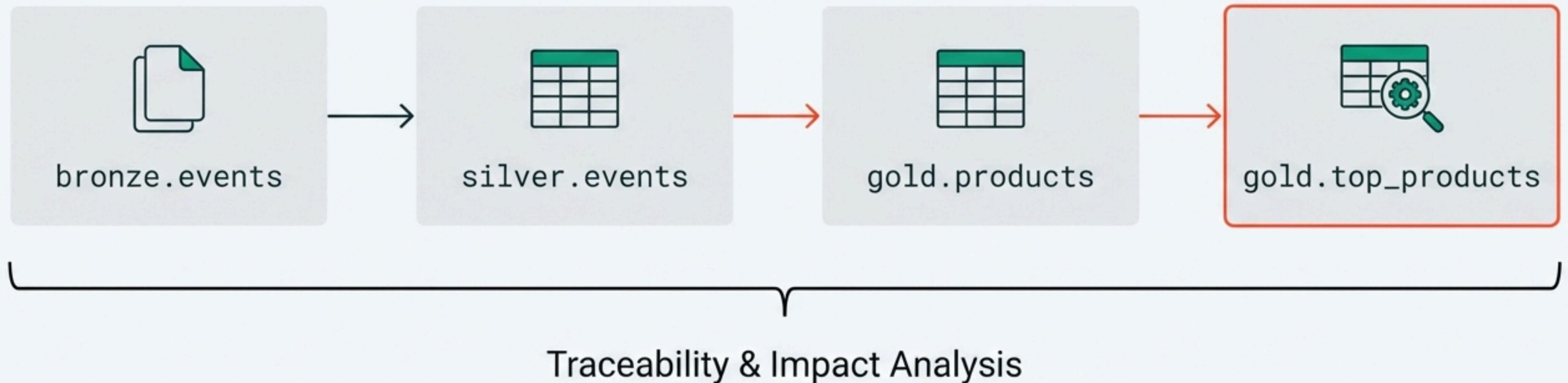
Column Selection

Row-Level Filteringing

Curation Logic

The Big Picture: Data Lineage

Unity Catalog automatically tracks the journey of your data.



Challenge Complete

-  Structure Created: Catalog & Schemas live.
-  Assets Registered: Delta tables mapped.
-  Permissions Set: Analysts/Engineers configured.
-  Access Controlled: Views created.

You have successfully deployed
a governed data architecture.

Feedback & Resources

Share your experience with the Day 08 Challenge.

Fill out the Feedback Form

<https://forms.office.com/r/aZcUUPHUwf>



#DatabricksWithIDC

