■ **Course:** CSCI E-103: Reproducible Machine Learning

■ **Week:** Lecture 12

■ **Instructors:** Ram Sriharsha (Guest Speaker)

■ **Objective:** Master the fundamentals of Data and AI Governance, including information governance frameworks, Unity Catalog implementation, and attribute-based access control (ABAC)

Contents

1 Introduction: Why Governance Matters

This lecture covers **Data and AI Governance**—the frameworks, policies, and technologies that enable organizations to maximize the value of their data assets while minimizing security and compliance risks.

Lecture Overview

Key Learning Objectives:

- **Understand** the hierarchy: Information Governance $\supset$ Data Governance $\supset$ AI Governance
- **Learn** the four pillars: Policies, Procedures, Standards, and Controls
- **Implement** governance using Databricks Unity Catalog
- **Apply** Attribute-Based Access Control (ABAC) for fine-grained security
- **Monitor** data quality and track data lineage automatically

Key Information

Why Should You Care?

”Data is the new oil”—but unlike oil, data is **renewable and limitless**. Organizations constantly produce, collect, and process data. The challenge is:

- **Maximize Value:** Make data accessible to drive business decisions
- **Minimize Risk:** Protect sensitive data from breaches and regulatory violations

These two goals are in constant tension. Governance is the art of balancing them.

2 Information Governance: The Big Picture

2.1 What Is Information Governance?

Definition:

Information Governance Information Governance is the overarching framework that encompasses **all** organizational information—physical documents, digital files, knowledge assets, and AI models.

Scope: Anything that is created, collected, stored, processed, or shared.

Examples:

- Locking your laptop when leaving your desk
- Shredding printed documents with sensitive information
- Policies about who can access which email folders

2.2 The Governance Hierarchy

Information Governance contains two major subsets:

Type	**Focus Area**
Data Governance	Managing digital data: quality, security, lifecycle, access control
AI Governance	Managing AI models: bias prevention, explainability, ethical use, training data lineage

Warning

You Can't Walk Before You Run:
AI Governance is **built on top of** Data Governance. If your data is messy, inconsistent, or poorly secured, you cannot build fair, transparent, and safe AI systems.
Example: If you don't know the lineage of your training data, how can you verify the model isn't trained on copyrighted or biased content?

2.3 Goals of Information Governance

1. **Maximize Information Value**
   - Make data discoverable and accessible to authorized users
   - Enable self-service analytics and AI development
2. **Mitigate Risk**
   - Prevent data breaches (reputational and financial damage)
   - Ensure regulatory compliance (GDPR, HIPAA, PCI-DSS, CCPA)
3. **Enhance Security**
   - Implement multi-factor authentication
   - Encrypt data at rest and in transit
4. **Optimize Lifecycle Management**
   - Define data retention policies
   - Archive or delete data when no longer needed
5. **Promote Transparency and Accountability**
   - Clear roles: Data Stewards, Governance Officers, Catalog Owners
   - Audit trails for all data access and modifications

Pillar	**Definition**	**Example (GDPR)**
Policies	High-level rules established by leadership or regulators	”Individuals have the right to request deletion of their personal data”
Procedures	Step-by-step instructions for implementing policies	”Process data deletion requests within 45 days”
Standards	Technical specifications and best practices	”Use AES-256 encryption; minimize data storage”
Controls	Mechanisms that enforce policies	”Multi-factor authentication; access logs; role-based permissions”

Table 1: *The Four Pillars of Governance*

3 The Four Pillars: Policies, Procedures, Standards, Controls

3.1 Framework Overview

3.2 How They Work Together

Example:

GDPR Compliance Flow **Policy:** GDPR mandates the ”Right to Be Forgotten”

Procedure:

1. User submits deletion request
2. Request logged in ticketing system
3. Data team identifies all user data across systems
4. Data deleted or anonymized within 30 days
5. Confirmation sent to user

Standards:

- Data must be encrypted at rest
- Access controlled via Role-Based Access Control (RBAC)
- Minimum data retention periods defined

Controls:

- Technical: Encryption, access control lists
- Administrative: Background checks for data handlers
- Physical: Secure data centers

4 Data Governance vs AI Governance

4.1 Data Governance

Definition:

Data Governance Data Governance focuses on managing **digital data assets**—their quality, security, lifecycle, and accessibility.

Key Questions:

- Who can access this dataset?
- Is this data encrypted?
- How long should we retain this data?
- Is the data accurate and up-to-date?

4.2 AI Governance

Definition:

AI Governance AI Governance extends data governance to **AI models and their outputs**.

Key Questions:

- Is the training data free of bias?
- Can the model's decisions be explained?
- Was the training data legally obtained?
- Does the model produce fair outcomes across demographic groups?

Important:

AI Governance is Critical Now With the rise of LLMs trained on internet-scale data:

- **Copyright concerns:** Was copyrighted material used for training?
- **Privacy violations:** Does the model memorize PII from training data?
- **Indemnification:** If you use an LLM to generate content that infringes IP, who is liable?

Organizations are increasingly asking: ”What is our legal exposure when using third-party AI models?”

5 Governance Maturity Model

Organizations progress through maturity levels. **You cannot skip levels**—each stage builds on the previous.

Key Information

The Goal for Most Organizations: Level 3 (Defined/Proactive)

At this level, you have:

- Documented policies and procedures

Level	Stage	Characteristics
1	**Initial/Aware**	No formal governance. Individuals manage data ad-hoc.
2	**Reactive/Managed**	Problems trigger responses. Some documentation exists.
3	**Defined/Proactive**	Enterprise-wide standards established. Most organizations target this.
4	**Quantified**	Governance effectiveness measured with metrics.
5	**Optimized**	Automated detection and remediation. Continuous improvement.

Table 2: *Data Governance Maturity Model*

- Clear roles and responsibilities
- Automated access controls
- Regular audits and compliance checks

6 Governance Operating Models

How governance is implemented depends on organizational culture and structure.

Model	Characteristics	Pros	Cons/Best For
Centralized	Central team controls all governance	High consistency, strong security	Slow, bottlenecks (regulated industries)
Decentralized	Each department self-governs	Fast innovation, flexibility	No standards, duplication (startups)
Federated	Central guidelines + local execution	Balance of control and autonomy	Coordination challenges (enterprises)
Hybrid	Core data centralized, rest decentralized	Protects sensitive data + efficiency	Complex structure

Table 3: *Governance Operating Models*

Example:

Federated Model in Practice **Central Governance Office:**

- Defines global policies (e.g., ”All PII must be encrypted”)
- Maintains the enterprise data catalog
- Conducts compliance audits

Business Unit Data Stewards:

- Implement policies within their domain
- Define local schemas and data quality rules
- Grant access to their datasets

7 Databricks Unity Catalog: Implementation

Unity Catalog (UC) is Databricks' unified governance solution for all data and AI assets.

7.1 The Three-Level Hierarchy

`Metastore → Catalog → Schema → Table / Volume / Model / Function`

Level	Description
Metastore	Top-level container (typically one per cloud region). Stores all metadata.
Catalog	Largest grouping of data assets. Examples: `prod`, `dev`, `hr_data`
Schema	Logical grouping within a catalog (equivalent to a database)
Table/Volume	Actual data. Tables = structured; Volumes = unstructured files

Table 4: *Unity Catalog Hierarchy*

7.2 Managed vs External Tables

Type	Managed Table	External Table
Storage	Databricks manages location	You specify cloud storage path
Lifecycle	DROP TABLE deletes data files	DROP TABLE removes metadata only
Use Case	Recommended for most scenarios	Legacy data, shared storage

Table 5: *Managed vs External Tables*

7.3 How Unity Catalog Security Works

1. **User submits query:** `SELECT * FROM catalog.schema.table`
2. **Access Control check:** Does user have SELECT permission?
3. **If authorized:** UC delegates to cloud IAM role to fetch data
4. **Data returned:** User sees only what they're permitted to see

Key Information

Two-Layer Security:

- **Layer 1 (Cloud IAM):** Controls access to storage buckets (get, put, list)
- **Layer 2 (Unity Catalog):** Fine-grained control (SELECT, INSERT on specific tables/columns)

Advantage: You don't need hundreds of IAM roles for different access patterns. One IAM role with broad access + UC for fine-grained control.

8 ABAC: Attribute-Based Access Control

8.1 RBAC vs ABAC

Approach	RBAC (Role-Based)	ABAC (Attribute-Based)
How it works	Assign permissions to roles; assign roles to users	Define policies based on data attributes (tags)
Example	”Managers can see all HR data”	”Anyone querying PII-tagged columns sees masked data”
Scalability	Role explosion as permissions grow	Scales well with tags
Flexibility	Static; requires role changes	Dynamic; tag changes propagate automatically

Table 6: *RBAC vs ABAC Comparison*

8.2 ABAC in Unity Catalog

The ABAC workflow in Databricks:

1. **Create Governance Tags:** Define tag names and allowed values
2. **Tag Data Assets:** Apply tags to columns/tables (manually or via AI classification)
3. **Create ABAC Policies:** Define rules based on tags
4. **Automatic Enforcement:** Policies apply whenever tagged data is accessed

```sql
-- Step 1: Create a masking function
CREATE FUNCTION ssn_mask(ssn STRING)
RETURNS STRING
RETURN
    CASE
        WHEN is_account_group_member('admin_group') THEN ssn
        WHEN is_account_group_member('analyst_group') THEN CONCAT('***-**-',
            RIGHT(ssn, 4))
        ELSE '***-**-****'
    END;

-- Step 2: Apply mask to a column
ALTER TABLE employees
ALTER COLUMN social_security_number
SET MASK ssn_mask;
```

Listing 1: Creating a Column Masking Function

8.3 ABAC Policy Types

Definition:

Row-Level Filtering Restrict which **rows** a user can see based on a condition.

Example: Sales reps can only see customers in their assigned region.

```
-- Only show rows where region matches user's region
CREATE FUNCTION region_filter()
RETURNS BOOLEAN
RETURN region = current_user_region();
```

Definition:

Column-Level Masking Transform or hide **column values** based on user permissions.

Example: Non-admin users see email as `j***@company.com`

8.4 Automatic Data Classification

Unity Catalog can automatically detect PII using AI models:

- **Email addresses:** Detected and tagged automatically
- **Phone numbers:** Pattern recognition
- **Names, locations:** NER-based detection
- **Credit card numbers:** Regex + Luhn validation

Key Information

Auto-Classification + ABAC = Powerful Automation

When you combine:

1. Auto-classification (AI detects email columns)
2. Governance tags (email columns get ”PII” tag)
3. ABAC policy (PII-tagged columns are masked for analysts)

New tables are automatically protected without manual intervention!

9 Data Quality Monitoring

9.1 Why Monitor Data Quality?

”Garbage in, garbage out” applies doubly to AI. If your data quality degrades, your models and reports become unreliable.

Metric	Description
Freshness	How recently was the data updated?
Completeness	Are all expected records present? (No sudden drops)
Anomaly Detection	Have data patterns changed unexpectedly?
Data Profiling	Statistics: min, max, nulls, distributions

Table 7: *Data Quality Metrics*

9.2 Key Metrics

Example:

Completeness Monitoring **Normal Pattern:** 10,000 rows daily

Day 1: 10,200 rows

Day 2: 9,800 rows

Day 3: 10,100 rows

Day 4: 0 rows ← **ALERT!**

The monitoring system detects the anomaly and triggers an alert before downstream systems are affected.

9.3 Setting Up Monitoring in Databricks

Data quality monitoring is enabled with a single click:

1. Navigate to schema in Unity Catalog
2. Click ”Enable Quality Monitoring”
3. System automatically tracks freshness, completeness, anomalies
4. Alerts can be configured for threshold violations

10 Lineage: Tracing Data Origins

10.1 What Is Data Lineage?

Definition:

Data Lineage A visual representation of data’s journey: where it came from, how it was transformed, and where it goes.

Analogy: A family tree (genealogy) for your data.

10.2 Why Lineage Matters

1. **Debugging:** ”This dashboard number looks wrong. Where did it come from?”
2. **Impact Analysis:** ”If I change this column, what downstream reports break?”
3. **Compliance:** ”Can we prove this data wasn’t derived from restricted sources?”

4. **AI Governance:** ”What data was used to train this model?”

10.3 Lineage in Unity Catalog

Unity Catalog automatically captures lineage:

- **Table-level:** Which tables feed into which tables
- **Column-level:** How specific columns are derived (e.g., substring, join)
- **Custom lineage:** Connect external sources (Salesforce) and targets (PowerBI)

Example:

Lineage Use Case **Scenario:** A BI report shows incorrect revenue figures.

With Lineage:

1. Navigate to the report’s source table
2. Click ”View Lineage”
3. Trace back through transformations
4. Discover: A join condition was changed 3 days ago
5. Fix the join and reprocess

11 Lakehouse Federation: Unified Governance

11.1 The Problem

Organizations have data in multiple systems:

- Snowflake data warehouse
- PostgreSQL operational database
- MySQL legacy systems
- Cloud storage (S3, Azure Blob)

Managing governance separately in each system is impractical.

11.2 The Solution: Federation

Definition:

Lakehouse Federation Connect external databases to Unity Catalog **without copying data**. Query remote data as if it were local, with unified governance.

Analogy: An embassy on foreign soil—your laws (governance rules) apply, even though the data physically resides elsewhere.

11.3 How Federation Works

1. **Create Connection:** Register external database (Snowflake, PostgreSQL, etc.)

2. **Create Foreign Catalog:** Maps external schemas to UC
3. **Query with Pushdown:** Queries are pushed to the source system for efficiency
4. **Unified Governance:** Same grant statements, same ABAC policies

```sql
-- Create connection to external Snowflake
CREATE CONNECTION snowflake_conn
TYPE snowflake
OPTIONS (
    host = 'account.snowflakecomputing.com',
    warehouse = 'COMPUTE_WH'
);

-- Create foreign catalog
CREATE FOREIGN CATALOG snowflake_catalog
USING CONNECTION snowflake_conn;

-- Query as if local!
SELECT * FROM snowflake_catalog.schema.table;
```

Listing 2: Creating a Federated Connection

12 Delta Sharing: Secure Data Exchange

12.1 The Challenge of Data Sharing

Traditional methods are insecure and inefficient:

- Email CSV files (security nightmare)
- FTP transfers (no access control)
- Copy data to partner's system (data duplication)

12.2 Delta Sharing Solution

Definition:

Delta Sharing An open protocol for securely sharing data **without copying**. Recipients don't need Databricks—they can consume via Pandas, Tableau, PowerBI.

What Can Be Shared:

- Tables (Delta format)
- Volumes (files)
- Notebooks
- AI Models
- Even federated tables from external sources!

Key Information

Cross-Cloud Sharing:
You can share a Snowflake table (connected via Federation) with a partner on AWS who uses Pandas. The data never leaves Snowflake, but governance is controlled through Unity Catalog.

13 The Governance Platform Wars

13.1 Who's Competing?

Every major data platform is building governance capabilities:

Platform	**Governance Layer**	**Differentiator**
Databricks	Unity Catalog	ML/AI-first; models, volumes, functions
Snowflake	Polaris + Horizon	SQL warehouse heritage; Iceberg focus
Microsoft	Fabric	Office 365 integration; broad enterprise
AWS	Glue Catalog + Lake Formation	Native AWS integration

Table 8: *Governance Platform Comparison*

Key Information

The Winner's Strategy:
The platform that can govern **everyone's assets**—not just their own—will win. If Databricks can effectively govern Snowflake data and vice versa, the most interoperable platform gains the most "assets under management."

14 Real-World Governance Considerations

14.1 Data Breaches and Reputational Risk

Example:

Case Study: Capital One Breach In 2019, Capital One suffered a major data breach affecting 100+ million customers.

Consequences:

- $80 million in regulatory fines
- Class action lawsuits
- Years of reputational damage
- Increased scrutiny from regulators

Lesson: The cost of poor governance far exceeds the cost of implementing it.

14.2 The Governance-Agility Tradeoff

- **Too Strict:** "I need 5 approvals to access any data"—innovation stalls

- **Too Loose:** ”Everyone has access to everything”—breaches happen

Solution: Risk-Based Governance

- **High-risk data (PII, financial):** Strict controls, approval workflows
- **Low-risk data (public datasets, experiments):** Permissive access

15 Quick Summary: One-Page Review

Key Summary

Key Takeaways from Lecture 12:

1. **Governance Hierarchy:** Information Governance $\supset$ Data Governance $\supset$ AI Governance
2. **Four Pillars:** Policies (what) → Procedures (how) → Standards (specifications) → Controls (enforcement)
3. **Unity Catalog Structure:** Metastore → Catalog → Schema → Table/Volume/Model
4. **ABAC Advantages:**
   - Tag-based policies scale better than per-table permissions
   - Auto-classification discovers PII automatically
   - Policies propagate to new data without manual intervention
5. **Data Quality:**
   - Monitor: Freshness, Completeness, Anomalies
   - Enable with one click in Unity Catalog
6. **Lineage:**
   - Automatically captured for all Databricks operations
   - Custom lineage for external sources/targets
7. **Federation:**
   - Query external databases without copying data
   - Unified governance across heterogeneous systems
8. **Delta Sharing:**
   - Share data without copying
   - Recipients don’t need Databricks

16 Frequently Asked Questions

Q: Why use Unity Catalog if we already have cloud IAM roles?

A: Cloud IAM controls access at the file/folder level. Unity Catalog provides fine-grained control at the table, column, and row level. You can have one IAM role with broad storage access, and use UC for precise governance.

Q: Does governance slow down innovation?

A: Initially, there's overhead in setting up policies. Long-term, governance **accelerates** work by:

- Reducing time spent finding/validating data
- Avoiding security incidents that halt projects
- Enabling self-service access to pre-approved datasets

Q: What's the difference between RBAC and ABAC?

A: RBAC assigns permissions to roles ("Managers can see HR data"). ABAC uses attributes/tags ("Anyone querying PII sees masked data"). ABAC is more flexible and scales better.

Q: How does federation handle performance?

A: Queries are pushed down to the source system. If PostgreSQL is efficient at filtering, that filter runs on PostgreSQL—not in Spark. This minimizes data movement.