**Databricks Compliance Issues with Mitigations**

## Databricks Compliance Issues Comparison Table with Mitigations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Compliance Issue | One-Liner Description | Where It Typically Arises | Compliance Areas Most Affected | Mitigation Strategies |
| 1 | Incomplete Access Controls | Missing or inconsistent permissions lead to unauthorized data access. | Data sharing, workspace provisioning | Unity Catalog, Tables, Storage | Use RBAC with Unity Catalog; regularly review and audit permissions; follow least privilege principle. |
| 2 | Lack of Data Lineage | Teams can’t trace where data came from or how it was transformed, complicating audits. | ETL pipelines, table evolution | Tables, Views, Pipelines | Enable lineage tracking; document transformations; leverage Delta transaction logs. |
| 3 | Missing Encryption Enforcement | Data at rest or in transit is not consistently encrypted, violating security standards. | Storage configuration, network setup | Cloud Storage, Clusters, Jobs | Enable encryption at rest and TLS for all connections; verify compliance with cloud security policies. |
| 4 | Inadequate Audit Logging | User and system activities are not fully logged, reducing traceability and accountability. | Production workloads, sensitive data processing | Jobs, Tables, Clusters | Enable workspace audit logs; export logs to secure storage or SIEM platforms for monitoring. |
| 5 | Unclassified Sensitive Data | Personally identifiable or regulated data is stored without appropriate tagging or classification. | Data ingestion, schema onboarding | Tables, Unity Catalog | Use column-level tags for sensitivity classification; enforce tagging policies on new datasets. |
| 6 | Retention Policy Gaps | Data is retained longer than necessary, increasing legal exposure. | Table lifecycle management | Tables, Backups | Define retention policies; schedule automated purges; document retention periods in governance policies. |
| 7 | Uncontrolled External Sharing | Data is shared externally without proper approval or tracking. | Cross-org collaboration, exports | Tables, Views, Jobs | Implement approval workflows for data sharing; monitor usage of public or external access credentials. |
| 8 | Insufficient Data Masking | Sensitive fields are exposed in queries, reports, or logs without redaction. | BI dashboards, ad hoc analysis | SQL Warehouses, Notebooks | Use dynamic data masking; restrict access to raw sensitive columns; enforce masking in dashboards. |
| 9 | Non-compliance with Regional Regulations | Workloads or storage reside in regions incompatible with data sovereignty requirements. | Multi-region deployments | Storage, Compute | Ensure region alignment with regulatory requirements (e.g., GDPR); document region-specific configurations. |
| 10 | Lack of Incident Response Planning | No documented processes exist to detect and respond to compliance incidents or breaches. | Security and compliance management | All Platform Areas | Develop incident response plans; run periodic tabletop exercises; train teams on breach procedures. |

**Quick Reference**

* **RBAC:** Role-Based Access Control for managing who can access what.
* **Unity Catalog:** Centralized governance for permissions, lineage, and metadata.
* **Audit Logs:** Logs capturing all key activities for traceability.
* **Data Masking:** Techniques to hide sensitive fields from unauthorized users.
* **Retention Policies:** Defined durations to keep or purge data.

**Example Mitigation Actions and Configurations**

**Grant Fine-Grained Permissions:**

sql

CopyEdit

GRANT SELECT ON SCHEMA main.finance TO `finance\_analysts`;

**Enable Workspace Audit Logs:**

* In the workspace admin console, enable audit logging to secure cloud storage.

**Tag Sensitive Columns:**

sql

CopyEdit

ALTER TABLE main.hr.employee\_data

SET TAGS ('sensitivity' = 'high', 'compliance' = 'GDPR');

**Mask Sensitive Data Dynamically:**

* In Unity Catalog (where available), define masking policies:

sql

CopyEdit

CREATE MASKING POLICY ssn\_mask

AS (val STRING) RETURNS STRING ->

CASE

WHEN is\_account\_group\_member('privileged') THEN val

ELSE 'XXX-XX-XXXX'

END;

ALTER TABLE main.hr.employee\_data

ALTER COLUMN ssn

SET MASKING POLICY ssn\_mask;

**Implement Retention Policies:**

sql

CopyEdit

DELETE FROM main.logs

WHERE event\_timestamp < current\_timestamp() - INTERVAL 90 DAYS;

**Verify Encryption at Rest:**

* In Azure Storage account settings:
  + Enable *Secure Transfer Required*.
  + Enforce *Encryption with Customer-Managed Keys* if needed.

**Document Incident Response:**

* Maintain runbooks in a shared knowledge base.
* Assign clear roles for breach detection and escalation.