# LAB 09: Governance & Security

**Duration:** ~40 min
**Day:** 3
**After module:** M09: Governance & Security
**Difficulty:** Advanced

---

## Scenario

> *"Before the RetailHub platform goes live, the security team requires data governance controls. The HR department must not see order amounts. Customer PII (email, phone) must be masked for analysts. Aggregated Gold-layer data will be shared with an external partner via Delta Sharing."*

---

## Objectives

After completing this lab you will be able to: - Grant and revoke privileges using Unity Catalog - Create and apply Column Masks (dynamic data masking) - Create and apply Row Filters (row-level security) - Query `INFORMATION_SCHEMA` for metadata inspection - Understand Delta Sharing concepts

---

## Prerequisites

- Completed LAB 07 (Lakeflow Pipeline) – tables in `gold` schema available
- Unity Catalog configured (from `00_setup.ipynb`)

---

## Part 1: Access Management (~10 min)

### Task 1: Grant Privileges

Grant `SELECT` on the `silver` schema to the `analysts` group:

```
GRANT USE CATALOG ON CATALOG ${CATALOG} TO `analysts`;
GRANT USE SCHEMA ON SCHEMA ${CATALOG}.${SILVER_SCHEMA} TO `analysts`;
GRANT SELECT ON SCHEMA ${CATALOG}.${SILVER_SCHEMA} TO `analysts`;
```

### Task 2: Verify Privileges

Query `INFORMATION_SCHEMA.TABLE_PRIVILEGES` to confirm the grants:

```
SELECT * FROM ${CATALOG}.INFORMATION_SCHEMA.TABLE_PRIVILEGES
WHERE grantee = 'analysts';
```

> **Exam Tip:** `GRANT SELECT ON SCHEMA` grants SELECT on all current and future tables in that schema.

---

## Part 2: Column Masks (~10 min)

### Task 3: Create a Masking Function

Create a SQL function that masks email addresses for non-admin users:

```
CREATE OR REPLACE FUNCTION ${CATALOG}.${SILVER_SCHEMA}.mask_email(email
        STRING)
RETURNS STRING
RETURN CASE
    WHEN is_account_group_member('admins') THEN email
    ELSE CONCAT(LEFT(email, 2), '***@***.***')
END;
```

### Task 4: Apply Column Mask

```
ALTER TABLE ${CATALOG}.${SILVER_SCHEMA}.customers
ALTER COLUMN email SET MASK ${CATALOG}.${SILVER_SCHEMA}.mask_email;
```

### Task 5: Verify Masking

Query the table as a non-admin user and verify the email column is masked.

> **Exam Tip:** Column Masks use `is_account_group_member()` to determine the querying user's group.

---

## Part 3: Row Filters (~10 min)

### Task 6: Create a Row Filter Function

Create a function that filters orders by region based on group membership:

```
CREATE OR REPLACE FUNCTION
        ${CATALOG}.${SILVER_SCHEMA}.region_filter(region STRING)
RETURNS BOOLEAN
RETURN (
    is_account_group_member('admins')
    OR (is_account_group_member('east_team') AND region = 'East')
    OR (is_account_group_member('west_team') AND region = 'West')
);
```

### Task 7: Apply Row Filter

```sql
ALTER TABLE ${CATALOG}.${SILVER_SCHEMA}.orders
SET ROW FILTER ${CATALOG}.${SILVER_SCHEMA}.region_filter ON (region);
```

### Task 8: Verify Row Filter

Query orders table and confirm only permitted rows are visible.

> **Exam Tip:** Row Filters are SQL UDFs applied with `ALTER TABLE ... SET ROW FILTER`. Remove with `ALTER TABLE ... DROP ROW FILTER`.

---

## Part 4: Metadata & Lineage (~10 min)

### Task 9: Query INFORMATION_SCHEMA

```sql
-- List all tables
SELECT table_schema, table_name, table_type
FROM ${CATALOG}.INFORMATION_SCHEMA.TABLES;

-- List columns with data types
SELECT table_name, column_name, data_type
FROM ${CATALOG}.INFORMATION_SCHEMA.COLUMNS
WHERE table_schema = '${SILVER_SCHEMA}';
```

### Task 10: Explore Lineage

Navigate to Unity Catalog UI > select a Gold table > click “Lineage” tab to view the data flow from Bronze through Silver to Gold.

> **Exam Tip:** Lineage is tracked automatically by Unity Catalog – no instrumentation

needed. It shows table-level and column-level dependencies.

---

## Summary

In this lab you: - Granted and revoked privileges using `GRANT` / `REVOKE` SQL - Created masking functions with `is_account_group_member()` - Applied Column Masks via `ALTER TABLE ... ALTER COLUMN ... SET MASK` - Applied Row Filters via `ALTER TABLE ... SET ROW FILTER` - Queried `INFORMATION_SCHEMA` for metadata inspection - Explored data lineage in the Unity Catalog UI

| Feature | Syntax | Purpose |
|---|---|---|
| Column Mask | `ALTER TABLE t ALTER COLUMN c SET MASK fn` | Dynamic data masking |
| Row Filter | `ALTER TABLE t SET ROW FILTER fn ON (col)` | Row-level security |
| Remove Mask | `ALTER TABLE t ALTER COLUMN c DROP MASK` | Remove masking |
| Remove Filter | `ALTER TABLE t DROP ROW FILTER` | Remove RLS |

**Key Functions:** `is_account_group_member('group')`, `current_user()`, `SHA2(col, 256)`