Test Strategy: Metadata Framework Functionality (Microsoft Fabric)

# 1. Objective

To verify that the metadata framework for Microsoft Fabric pipelines correctly interprets, orchestrates, and manages pipelines based solely on the metadata structure — without validating the actual data being loaded.

# 2. Scope

Included:

* - Metadata-driven orchestration logic  
  - Behavior when metadata changes (add/edit/delete)  
  - Pipeline activation, skipping, or rerouting based on metadata  
  - Logging and status updates in metadata tables

Excluded:

* - Data validation  
  - Source/target correctness  
  - Transformation accuracy  
  - Performance or volume testing

# 3. Key Test Scenarios

## A. Metadata Schema Validation

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Outcome |
| TC-01 | Required columns present in metadata table | Framework starts without failure |
| TC-02 | Invalid column name or missing value | Framework skips entry or fails gracefully |
| TC-03 | Add new optional field | Pipeline does not break; ignores or adapts |

## B. Pipeline Invocation via Metadata

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Outcome |
| TC-04 | Insert new row in metadata table | Framework invokes child pipeline accordingly |
| TC-05 | Set pipeline to inactive (enabled=false or status=skip) | Pipeline is skipped as expected |
| TC-06 | Add multiple rows | Framework iterates correctly through ForEach |

## C. Dynamic Routing & Parameters

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Outcome |
| TC-07 | Metadata specifies full load | Framework routes to appropriate branch |
| TC-08 | Metadata specifies incremental load | Correct sub-pipeline is triggered |
| TC-09 | Changing load type triggers new path | Framework adjusts behavior dynamically |

## D. Error Handling & Logging

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Outcome |
| TC-10 | Metadata missing mandatory value | Logged as failed in status/error column |
| TC-11 | Invalid reference (e.g. pipeline not found) | Framework logs error in metadata |
| TC-12 | Unexpected value in metadata (e.g. load\_type=‘partial’) | Default error branch triggered |

## E. Metadata Updates

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Outcome |
| TC-13 | Modify watermark value | Framework uses updated value in logic (if relevant) |
| TC-14 | Update existing row to switch pipeline | Invokes new pipeline correctly on next run |
| TC-15 | Delete metadata row | Pipeline is no longer executed |

# 4. Environment Setup

* - Test Metadata Table: A dedicated metadata table with dummy config values (no real data sources)  
  - Mock Pipelines: Minimal pipelines with logging-only actions  
  - Isolated Environment: Dev/QA Fabric workspace for controlled testing

# 5. Acceptance Criteria

* - All valid metadata rows trigger correct pipeline logic.  
  - No pipelines run when metadata indicates inactive.  
  - Metadata errors are logged clearly in output.  
  - Adding/removing/modifying metadata dynamically changes pipeline behavior without code changes.  
  - Framework never executes data logic (only metadata-driven orchestration).

# 6. Suggested Test Artifacts

* - Test Metadata Table SQL Template (with dummy config)  
  - Orchestrator pipeline (no data tasks, only logs or counters)  
  - Mock pipelines (child pipelines that only log input parameters)  
  - Validation script to compare metadata rows vs logs