Here's a comprehensive test plan in a tabular format for the project involving the deletion of pricing policies transactions in the analytical, silver, and gold layers of a datalake. The plan ensures no other records are deleted and follows a systematic approach to validate the deletion process across different environments.

| **Step** | **Description** | **Expected Outcome** | **Actual Outcome** | **Owner** | **Status** | **Date** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | **Identify Deletion Criteria** | Criteria for deletion is clearly defined |  | Data Team | Pending | 2024-06-15 |
| 2 | **Define Table Scope** | List of tables and layers (analytical, silver, gold) is identified |  | Data Team | Pending | 2024-06-16 |
| 3 | **Deletion Strategy** | Strategy for deletion is documented |  | Data Team | Pending | 2024-06-17 |
| 4 | **Backup Environment Deletion** | Execute deletion in backup environment | Deletion executed successfully and only specified records are deleted |  | Data Team | Pending |
| 5 | **Backup Environment Validation** | Validate deletion in backup environment | All specified records are deleted, no other records affected |  | QA Team | Pending |
| 6 | **Backup Environment Post-Job Run Validation** | Validate post-job runs and orchestration impacts | No issues found in post-job runs and orchestration |  | QA Team | Pending |
| 7 | **Backup Environment Report Validation** | Validate reports to ensure correctness | Reports show accurate data post-deletion |  | QA Team | Pending |
| 8 | **Review Backup Deletion with Project Team** | Review outcomes with project team | Project team approves the deletion |  | Project Lead | Pending |
| 9 | **Test Environment Deletion** | Execute deletion in test environment | Deletion executed successfully and only specified records are deleted |  | Data Team | Pending |
| 10 | **Test Environment Validation** | Validate deletion in test environment | All specified records are deleted, no other records affected |  | QA Team | Pending |
| 11 | **Test Environment Post-Job Run Validation** | Validate post-job runs and orchestration impacts | No issues found in post-job runs and orchestration |  | QA Team | Pending |
| 12 | **Test Environment Report Validation** | Validate reports to ensure correctness | Reports show accurate data post-deletion |  | QA Team | Pending |
| 13 | **Review Test Deletion with Project Team** | Review outcomes with project team | Project team approves the deletion |  | Project Lead | Pending |
| 14 | **Pre-prod Environment Deletion** | Execute deletion in pre-prod environment | Deletion executed successfully and only specified records are deleted |  | Data Team | Pending |
| 15 | **Pre-prod Environment Validation** | Validate deletion in pre-prod environment | All specified records are deleted, no other records affected |  | QA Team | Pending |
| 16 | **Pre-prod Environment Post-Job Run Validation** | Validate post-job runs and orchestration impacts | No issues found in post-job runs and orchestration |  | QA Team | Pending |
| 17 | **Pre-prod Environment Report Validation** | Validate reports to ensure correctness | Reports show accurate data post-deletion |  | QA Team | Pending |
| 18 | **Review Pre-prod Deletion with Project Team** | Review outcomes with project team | Project team approves the deletion |  | Project Lead | Pending |
| 19 | **Production Environment Deletion** | Execute deletion in production environment | Deletion executed successfully and only specified records are deleted |  | Data Team | Pending |
| 20 | **Production Environment Validation** | Validate deletion in production environment | All specified records are deleted, no other records affected |  | QA Team | Pending |
| 21 | **Production Environment Post-Job Run Validation** | Validate post-job runs and orchestration impacts | No issues found in post-job runs and orchestration |  | QA Team | Pending |
| 22 | **Production Environment Report Validation** | Validate reports to ensure correctness | Reports show accurate data post-deletion |  | QA Team | Pending |
| 23 | **Review Production Deletion with Project Team** | Review outcomes with project team | Project team approves the deletion |  | Project Lead | Pending |

**Notes**

* Each environment (Backup, Test, Pre-prod, Production) should have a testing gate to ensure correctness before proceeding to the next step.
* Thorough validation at each stage is critical to ensure that only the specified records are deleted and that the process does not affect other records or system functionality.
* Cost analysis is explicitly out of scope for this deletion process.

Here's a comprehensive test plan in a tabular format, tailored for your project, which involves deleting pricing policies transactions across different layers of a data lake while ensuring no other records are deleted.

**Test Plan for Pricing Policies Transaction Deletion**

| **Test Plan Description** | **Expected Outcome** | **Actual Outcome** | **Owner** | **Status** | **Date** |
| --- | --- | --- | --- | --- | --- |
| **1. Deletion Criteria Identification** |  |  |  |  |  |
| Identify primary deletion criteria provided by the architecture team. | Criteria correctly identified and documented. |  | Architecture Team | Pending |  |
| **2. Table Scope Identification** |  |  |  |  |  |
| Identify tables and data scope for deletion with input from development and architecture teams. | All relevant tables and scopes identified. |  | Dev/Arch Team | Pending |  |
| **3. Deletion Strategy Confirmation** |  |  |  |  |  |
| Confirm deletion strategy, including table-level, view-dependent, and file-level deletions. | Clear and approved deletion strategy documented. |  | Development Team | Pending |  |
| **4. Backup Plans and Approval** |  |  |  |  |  |
| Develop backup plans and get approvals from the platform team considering financial and effort impacts. | Backup plans approved and documented, ensuring rollback capabilities. |  | Platform Team | Pending |  |
| **5. Initial Deletion in Backup Environment** |  |  |  |  |  |
| Execute deletion in the backup environment to test the process and ensure criteria are met. | Deletion completed without impacting other records, confirmed by validation. |  | Dev Team | Pending |  |
| **6. Validation of Deletion in Backup Environment** |  |  |  |  |  |
| Validate the deletion process in the backup environment, including dependent units, catalogs, databases, and integrity. | Validation confirms no unintended deletions, referential integrity maintained. |  | QA Team | Pending |  |
| **7. Deletion in Test Environment** |  |  |  |  |  |
| Execute the deletion process in the test environment after successful backup environment validation. | Deletion process executes correctly, no impact on other records or environments. |  | Dev Team | Pending |  |
| **8. Validation in Test Environment** |  |  |  |  |  |
| Validate the deletion process in the test environment, checking dependencies and impact on other developers' work. | Validation confirms deletion correctness, no dependency issues or unintended impacts. |  | QA Team | Pending |  |
| **9. Pre-Production Environment Preparation** |  |  |  |  |  |
| Coordinate with development and testing teams to prepare for deletion in the pre-production environment. | All dependencies checked, no impact on ongoing deliveries, and environment ready for deletion. |  | Dev/QA Teams | Pending |  |
| **10. Deletion in Pre-Production Environment** |  |  |  |  |  |
| Execute the deletion in the pre-production environment following successful test validation. | Deletion completed correctly, no other records affected, validated as per expectations. |  | Dev Team | Pending |  |
| **11. Post-Job Run and Orchestration Impact Validation** |  |  |  |  |  |
| Validate post-job run and orchestration impacts in pre-production, ensuring correct deletion and no unintended effects. | All post-job runs complete successfully, no orchestration issues, deletion confirmed as per plan. |  | QA Team | Pending |  |
| **12. Pre-Production Report Validation** |  |  |  |  |  |
| Validate reports and outputs post-deletion in the pre-production environment. | Reports and outputs validated, deletion confirmed to meet expected outcomes, no errors found. |  | QA Team | Pending |  |
| **13. Project Team Review** |  |  |  |  |  |
| Review the deletion process, validation results, and reports with the project team before proceeding to production. | Project team approves the deletion process based on validation and report outcomes. |  | Project Team | Pending |  |
| **14. Deletion in Production Environment** |  |  |  |  |  |
| Execute the deletion in the production environment following approval and validation from previous steps. | Deletion completed successfully, no other records affected, validated as per expectations. |  | Dev Team | Pending |  |
| **15. Final Validation and Reporting in Production** |  |  |  |  |  |
| Validate the deletion process, post-job runs, and reports in the production environment. | Final validation confirms successful deletion, no unintended impacts, and correct execution. |  | QA Team | Pending |  |
| **16. Cost Analysis and Approval for Rollback (if needed)** |  |  |  |  |  |
| If a rollback is necessary, perform a cost analysis and get approval for the reprocessing and rollback operations. | Rollback plans approved, ensuring minimal financial and effort impact if reprocessing is required. |  | Platform Team | Pending |  |
| **17. Documentation and Sign-Off** |  |  |  |  |  |
| Document the entire deletion process, validations, and outcomes, and get final sign-off from all stakeholders. | Complete documentation and sign-off from all stakeholders, confirming successful completion of the deletion plan. |  | Project Team | Pending |  |

**Notes:**

* Ensure each step is meticulously documented, with clear roles and responsibilities assigned.
* Validate each environment's deletion process thoroughly to avoid any unintended data loss.
* Communication and coordination with all teams (development, QA, architecture, platform) are crucial throughout the process.
* Regularly update the test plan status to keep track of progress and address any issues promptly.

This test plan covers all critical aspects, from criteria identification to final validation and reporting, ensuring a thorough and controlled deletion process across different layers of the data lake.

Here is a comprehensive test plan in a tabular format for deleting pricing policies transactions in the analytical, silver, and gold layers of a datalake, ensuring no other records are deleted. This plan includes steps for deletion criteria identification, table scope identification, deletion strategy, and validation processes in different environments.

**Test Plan for Pricing Policies Transaction Deletion**

| **Test Plan Description** | **Expected Outcome** | **Actual Outcome** | **Owner** | **Status** | **Date** |
| --- | --- | --- | --- | --- | --- |
| **Identify Deletion Criteria** | Criteria identified and documented | TBD | Architecture Team | Pending | YYYY-MM-DD |
| 1. Deletion criteria to be provided by the architecture team | Clear and precise deletion criteria serving as the primary identification |  |  |  |  |
| **Identify Table Scope** | Scope of tables for deletion identified and documented | TBD | Developer Team, Architecture Team | Pending | YYYY-MM-DD |
| 2. Identify all tables in analytical, silver, and gold layers that contain pricing policies transactions | Complete list of tables documented |  |  |  |  |
| **Define Deletion Strategy** | Deletion strategy defined and documented | TBD | Development Team | Pending | YYYY-MM-DD |
| 3. Define the level of deletion: table, related views, dependent unit, catalog, database, referential integrity, file level, deep deletion | Clear strategy for each table/layer identified |  |  |  |  |
| 4. Determine any platform-related changes required for deletion | Platform changes identified and documented |  |  |  |  |
| 5. Establish backup plans and cost for implementing them | Backup plans documented and approved |  |  |  |  |
| **Backup Environment Testing** | Deletion in backup environment validated | TBD | Testing Team | Pending | YYYY-MM-DD |
| 6. Execute deletion in the backup environment | Deletion completed as expected, no other records deleted |  |  |  |  |
| 7. Validate post-job run and orchestration impact | No negative impact on jobs or orchestration |  |  |  |  |
| 8. Validate reports | Reports reflect accurate data post-deletion |  |  |  |  |
| 9. Review outputs with project team | Outputs reviewed and approved |  |  |  |  |
| **Test Environment Deletion** | Deletion in test environment validated | TBD | Testing Team, Developers | Pending | YYYY-MM-DD |
| 10. Execute deletion in the test environment | Deletion completed as expected, no other records deleted |  |  |  |  |
| 11. Check dependencies with other developers and testing teams | No negative impact on other deliveries |  |  |  |  |
| **Preprod Environment Deletion** | Deletion in preprod environment validated | TBD | Testing Team, Developers | Pending | YYYY-MM-DD |
| 12. Execute deletion in preprod environment | Deletion completed as expected, no other records deleted |  |  |  |  |
| 13. Validate post-job run and orchestration impact | No negative impact on jobs or orchestration |  |  |  |  |
| 14. Validate reports | Reports reflect accurate data post-deletion |  |  |  |  |
| **Production Environment Deletion** | Deletion in production environment validated | TBD | Testing Team, Developers | Pending | YYYY-MM-DD |
| 15. Execute deletion in production environment | Deletion completed as expected, no other records deleted |  |  |  |  |
| 16. Validate post-job run and orchestration impact | No negative impact on jobs or orchestration |  |  |  |  |
| 17. Validate reports | Reports reflect accurate data post-deletion |  |  |  |  |
| 18. Get final approval from the project team | Final approval received |  |  |  |  |
| **Post-Deletion Monitoring** | Monitoring of the environment post-deletion | TBD | Monitoring Team | Pending | YYYY-MM-DD |
| 19. Monitor the environment post-deletion for any unexpected issues | No unexpected issues found |  |  |  |  |

**Notes:**

* **Owner**: Assign specific individuals or teams responsible for each step.
* **Status**: Track the current status (Pending, In Progress, Completed) of each step.
* **Date**: Indicate the planned or actual completion date for each step.

This test plan ensures a thorough and structured approach to deleting pricing policies transactions, with safeguards to validate accuracy and prevent unintended deletions.

**Key Notes**

1. **Cost Analysis:** This step is essential but is not included in the non-functional testing coverage. However, for any reversal in production, developers must get approval due to the associated costs.
2. **Dependencies:** Ensure to check dependencies with other developers and testing teams before initiating deletions in pre-prod and prod environments.
3. **Testing Gates:** Each environment should have a testing gate to ensure deletion correctness, including post-job run and orchestration impact validation.

This structured approach ensures that the deletion process is well-planned, validated at each stage, and executed with minimal risk and impact.

# Test Plan for Pricing Policies Transaction Deletion in Datalake

## 1. Test Plan Description

### Objective

The objective of this test plan is to validate the deletion of pricing policies transactions from the Analytical, Silver, and Gold layers of the datalake, ensuring no other records are deleted. This plan includes detailed steps for deletion criteria identification, scope determination, and deletion strategy execution, ensuring integrity and minimal disruption across the environment layers.

### Scope

This test plan covers:

* Identification and validation of deletion criteria.
* Scope definition of tables and records for deletion.
* Validation of deletion strategy.
* Execution and validation of deletions in backup, test, pre-production, and production environments.
* Ensuring data integrity and no impact on other processes or records.

## 2. Test Plan Details

### 2.1 Deletion Criteria Identification

**Owner:** Architecture Team  
**Description:** Identify the criteria for deletion which serves as the primary identification for the records to be deleted. This will ensure only the specified pricing policies transactions are targeted.

### 2.2 Table Scope Identification

**Owner:** Development and Architecture Teams  
**Description:** Determine the tables within the Analytical, Silver, and Gold layers that fall within the scope for deletion. This includes identifying all related views, dependent unit catalogs, and databases to understand the impact of the deletion.

### 2.3 Deletion Strategy Confirmation

**Owner:** Development Team  
**Description:** Define the deletion strategy detailing the level of deletion:

* Table level
* Related views and dependent units
* Referential integrity constraints
* File level deletion or deep deletion
* Platform-related changes if any

Ensure backup plans are in place with cost analysis and approval from the platform team, considering the financial and effort implications of reprocessing data.

## 3. Test Execution Steps

### Step 1: Backup Environment Deletion

**Owner:** Development Team  
**Description:** Perform the deletion in the backup environment first. Validate that the deletion is completed as expected without affecting other records.

### Step 2: Test Environment Deletion

**Owner:** Development Team, Testing Team  
**Description:** After successful validation in the backup environment, perform the deletion in the test environment. Coordinate with other developers and testing teams to ensure there are no dependencies or impacts on their work.

### Step 3: Pre-Production Environment Deletion

**Owner:** Development Team, Testing Team  
**Description:** Following successful validation in the test environment, execute the deletion in the pre-production environment. Conduct thorough testing and validation to ensure no unintended impacts.

### Step 4: Production Environment Deletion

**Owner:** Development Team, Testing Team, Project Team  
**Description:** Finally, upon validation and approval from all stakeholders, execute the deletion in the production environment. Ensure all pre-checks are done, and there is no impact on live operations.

## 4. Testing and Validation

### 4.1 Post-Deletion Validation

**Owner:** Testing Team  
**Description:** After each deletion phase (backup, test, pre-prod, and prod), validate the deletion correctness by running post-job run validations, orchestration impact checks, and report validations.

### 4.2 Review and Sign-off

**Owner:** Project Team  
**Description:** Review the validation outputs with the project team and ensure all stakeholders sign off before proceeding to the next environment.

### 4.3 Dependency and Impact Analysis

**Owner:** Testing Team, Development Team  
**Description:** Continuously check for dependencies and impacts on other processes. Ensure no critical functionalities are disrupted.

## 5. Test Plan Schedule

| **Task** | **Owner** | **Status** | **Start Date** | **End Date** |
| --- | --- | --- | --- | --- |
| Identify Deletion Criteria | Architecture Team | Pending | 2024-06-15 | 2024-06-20 |
| Define Table Scope | Dev & Arch Teams | Pending | 2024-06-21 | 2024-06-25 |
| Confirm Deletion Strategy | Development Team | Pending | 2024-06-26 | 2024-06-30 |
| Execute Deletion in Backup Environment | Development Team | Pending | 2024-07-01 | 2024-07-05 |
| Validate Backup Environment Deletion | Testing Team | Pending | 2024-07-06 | 2024-07-10 |
| Execute Deletion in Test Environment | Dev & Testing Teams | Pending | 2024-07-11 | 2024-07-15 |
| Validate Test Environment Deletion | Testing Team | Pending | 2024-07-16 | 2024-07-20 |
| Execute Deletion in Pre-Prod Environment | Dev & Testing Teams | Pending | 2024-07-21 | 2024-07-25 |
| Validate Pre-Prod Environment Deletion | Testing Team | Pending | 2024-07-26 | 2024-07-30 |
| Execute Deletion in Production | Dev, Testing, Project Teams | Pending | 2024-07-31 | 2024-08-05 |
| Validate Production Deletion | Testing Team | Pending | 2024-08-06 | 2024-08-10 |
| Final Review and Sign-off | Project Team | Pending | 2024-08-11 | 2024-08-15 |

## 6. Risk Management

### Potential Risks

* **Incorrect deletion criteria leading to data loss**
* **Unintended impacts on dependent processes or tables**
* **Failures in the backup or restoration process**
* **Higher than expected costs for backup and restoration**

### Mitigation Strategies

* **Thorough validation of deletion criteria with multiple stakeholders**
* **Extensive testing in non-production environments before production deletion**
* **Detailed backup and restoration plans with cost analysis and approvals**
* **Continuous monitoring and immediate rollback plans in case of issues**

## 7. Conclusion

This test plan aims to ensure a meticulous approach to the deletion of pricing policies transactions across various layers of the datalake, ensuring data integrity and minimal disruption. The involvement of multiple teams and thorough validation steps across different environments will help in achieving the desired outcomes without unintended consequences.

Preprare test plan that covers test plan description, expected outcome , actual outcome , owner, status , date for the project , where pricing policies transaction to be deleted in analytical, silver , gold layer of datalake and ensure no other records to be deleted. steps to include deletion criteria identification this to be provided by archecture team as it serves as primary deletion identification criteria , table scope to be identified by devloper and archect team it serves as scope for deletion and testing, and deletion statergy - this is very important step to be confirmed by development team, it should define what level of deletion to be performed table, relelated views dependent Unit catalogue database, referential intergerity, file level deletion or deep deletion, if any platform related changes, back up plans , cost implementing backup plan and approval from platform team as the cost of reprocessing analytical or any layer is huge in terms of financial as well as effort prespective. Then only actual deletion to be performed in any environment other than backup environment. The deletion to be first done in back up environment and test the deletion is completed as per the expected outcome and followed by deletion to be done in test environment , check dependencies with other developers and testing team before initiate the preprod and prod environment to ensure their delivery is not impacted, each environment should have testing gate to ensure deletion correctness, post job run and orchestration impact validation and report validation and get the output reviewed with project team before doing any deletion in production. Cost Analysis is not scope for non functional testing coverage , however in case of reverting any implementated change in production environment is cost involved it essenstial developer to get approve from platform and cloud council before any deletion in preprod and production