



Vantage RuntimeAI

Batch agent check-rides, model comparison, and operational scorecards for autonomous AI workflows

8.4/10

Overall score

Run summary

Scenario: Lineage Impact Review

Model: mistralai/mistral-nemo

Roles: Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 1m 11s

Actual run cost: \$0.000234

Projected execution cost: \$0.0515/mo (10/biz day × 22 days)

Report date: July 13, 2026 - 9:11 PM EST

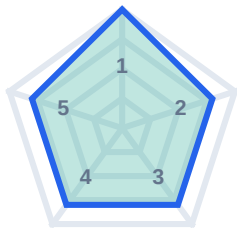
Overall

8.4/10

PASS - trust=high

Pass - score 8.4/10 clears 7.0 with acceptable trust/closure.

CAPABILITY RUBRIC



#	Dimension (matches spoke #)	Score
1	Downstream impact mapped	10/10
2	References lineage artifact	8/10
3	Migration / cutover risk	8/10
4	Ordered impact summary	8/10
5	Safe migration plan	8/10

Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

Technical criteria checklist

Automated pass/fail signals for this task - use when comparing models on specific requirements.

- Legacy column named: Met
- Multiple downstream marts covered: Met
- Migration plan for unmigrated rows: Met
- Rollback / compat considered: Met

TECHNICAL JUDGEMENT

Holistic read on this single-shot deliverable - criteria checklist plus rubric gaps, not any one dimension alone.

8/10

Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 6 - Counterpart msgs: 7 - Output: 17550 chars - Latency: 70s - Criteria: 4/4

Score analysis

Maps impact across multiple downstream marts.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 6 agent turns with scripted follow-ups and reviewer sign-off. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Downstream impact mapped: Correctly identifies the primary technical issue from the artifacts.
- + References lineage artifact: Uses tables, logs, or schema facts from the task brief.
- + Migration / cutover risk: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Ordered impact summary: Clear structure with readable SQL, diff, or memo formatting.
- + Safe migration plan: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Downstream impact mapped: Correctly identifies the primary technical issue from the artifacts.
- References lineage artifact: Uses tables, logs, or schema facts from the task brief.
- Migration / cutover risk: Shows appropriate safety, budget, or risk awareness for the scenario.
- Ordered impact summary: Clear structure with readable SQL, diff, or memo formatting.
- Safe migration plan: Strong rationale and actionable follow-through.
- Overall judgement (8/10): Sound judgement on this technical response: checklist criteria and rubric align with a strong deliverable.

Signals

Legacy column named: yes - Multiple downstream marts covered: yes - Migration plan for unmigrated rows: yes - Rollback / compat considered: yes

CAVEATS

- Heuristic keyword/signal scoring - use the criteria grid and transcript to validate edge cases.

Transcript

Session

Scenario: Lineage Impact Review

Agent response <-> Engineering counterpart

Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 1m 11s

Engineering counterpart

1:10 AM

Task brief - Lineage impact review (complex complexity)

Engineering proposes dropping a legacy column. Assess downstream impact:

```
```json
{
 "proposed_change": {
 "action": "DROP COLUMN",
 "table": "raw.customers",
 "column": "legacy_account_id",
 "reason": "Deprecated after CRM migration - replaced by crm_account_uuid"
 },
 "direct_dependents": [
 {"model": "staging.stg_customers", "column_refs": ["legacy_account_id"]},
 {"model": "intermediate.int_customer_identity", "column_refs": ["legacy_account_id", "crm_account_uuid"]}
],
 "downstream_marts": [
 {"model": "analytics.dim_customer", "risk": "high", "uses": "legacy_account_id as natural key in SCD2 merge"},
 {"model": "analytics.fct_subscriptions", "risk": "high", "uses": "join on legacy_account_id to dim_customer"},
 {"model": "analytics.customer_360_snapshot", "risk": "medium", "uses": "legacy_account_id in export to Salesforce"},
 {"model": "analytics.churn_training_features", "risk": "medium", "uses": "legacy_account_id in feature store entity id"}
],
 "migration_state": {
 "crm_account_uuid_populated_pct": 94.2,
 "legacy_only_accounts": 1204
 },
 "deadline": "2026-07-12 (CRM cutover)"
}
```
```

Agent response

1:10 AM - Responded in 16s

Impact Review: Dropping the "legacy_account_id" Column in "raw.customers"

1. Affected Marts/Models:

- Staging: `staging.stg\_customers` (Direct dependent)
 - Impact: Loss of legacy account ID, may affect historical data and merge logic.
- Intermediate: `intermediate.int\_customer\_identity` (Direct dependent)
 - Impact: Loss of legacy account ID, may affect identity resolution logic and data integrity.
- Analytics:
 - `analytics.dim\_customer` (High risk)
 - Impact: High risk due to use of legacy_account_id as natural key in Slowly Changing Dimension (SCD) Type 2 merge. This will disrupt historical data tracking.
 - `analytics.fct\_subscriptions` (High risk)
 - Impact: High risk due to join dependency on legacy_account_id. This will break the subscription fact table's relationship with the customer dimension.
 - `analytics.customer\_360\_snapshot` (Medium risk)
 - Impact: Medium risk due to use of legacy_account_id in exports to Salesforce. This will cause data discrepancies in the exported snapshots.
 - `analytics.churn\_training\_features` (Medium risk)
 - Impact: Medium risk due to use of legacy_account_id in feature store entity ID. This may cause data inconsistency in churn prediction models.

2. Migration Risks for Unmigrated Accounts (1204 Accounts):

- Data inconsistency: 5.8% of accounts have not been migrated, leading to potential data loss or inaccuracy.
- Manual effort required: These accounts must be migrated manually, which can be error-prone and time-consuming.
- Data availability: Data for these accounts may be required for historical analysis or data recovery purposes.

3. Recommended Migration Order:

- Phase 1 (2026-01-01 to 2026-03-31): Migrate the 1204 unmigrated accounts manually. Validate data quality and consistency.
- Phase 2 (2026-04-01 to 2026-06-30): Update the intermediate and staging models (staging.stg_customers, intermediate.int_customer_identity) to reflect the migration.
- Phase 3 (2026-07-01 to 2026-07-11): Prepare and test the analytics models (analytics.dim_customer, analytics.fct_subscriptions, analytics.churn_training_features) for the column drop. Update data flow and ETL jobs.
- Phase 4 (2026-07-12): Drop the "legacy_account_id" column in the production environment.

4. Rollback Considerations:

- Maintain a backup of the legacy data in a separate table or database for at least one year after the column drop.
- Ensure all ETL jobs, models, and reports are thoroughly tested post-migration.
- Establish a rollback plan in case of data inconsistency or disruption of critical business processes. This may involve re-inserting the legacy column or restoring the backup data.
- Monitor data quality and user feedback post-migration to address any issues promptly.

Engineering counterpart

1:10 AM - Responded in 0s

Thanks - walk me through your reasoning step by step. What assumptions did you make?

Agent response

1:10 AM - Responded in 8s

Step-by-Step Reasoning and Assumptions

1. Identify affected models and dependencies:

- Started by listing all direct and downstream dependent models from the provided JSON data.
- Assumed that each model's behavior and data dependency were as described in the JSON, with no additional hidden dependencies.

2. Categorize impact risk:

- Categorized the impact risk for each downstream model into 'High', 'Medium', or 'Low' based on the following assumptions:
 - 'High' risk: Loss of critical data (legacy_account_id) that disrupts key functionalities like natural key in SCD, joining fact and dimension tables, or data export.
 - 'Medium' risk: Loss of data that may cause data inconsistency or affect business processes but is not critical for core functionalities (e.g., feature store entity ID).
 - 'Low' risk: Minimal impact on data or processes (no explicit low-risk cases in this scenario).

3. Assess migration risks for unmigrated accounts:

- Assumed that the 1204 unmigrated accounts represent a static number, and no further automatic migration would occur before the column drop.
- Estimated the potential data inconsistency and manual effort required to migrate these accounts.

4. Determine the recommended migration order:

- Phase 1: Assumed that manual migration of unmigrated accounts should be the first step to minimize data loss and inconsistency.
- Phase 2: Assumed that updating intermediate and staging models would be necessary to reflect the migration and prevent data discrepancies in the analytics models.
- Phase 3: Assumed that preparing and testing analytics models would require updating data flow and ETL jobs to accommodate the column drop.
- Phase 4: Assumed that the final column drop could occur after thorough testing and preparation in the production environment.

5. Consider rollback strategies:

- Assumed that maintaining a backup of legacy data and establishing a rollback plan would be necessary to address any issues post-migration.
- Assumed that monitoring data quality and user feedback would help identify and resolve any problems promptly.

6. Additional assumptions and considerations:

- Assumed that the 'crm_account_uuid' column could serve as a suitable replacement for the 'legacy_account_id' column, given that the migration was driven by a CRM migration.
- Considered that historical data might be affected, especially for the high-risk models (analytics.dim_customer and analytics.fct_subscriptions), but assumed that addressing this would be part of the migration plan.
- Considered that the deadline of "2026-07-12 (CRM cutover)" might influence the migration timeline and ordering, but assumed that the provided migration phases would align with this deadline.

By following these steps and making the stated assumptions, I arrived at the initial impact review, recommended migration order, and rollback considerations for dropping the "legacy_account_id" column in "raw.customers".

Engineering counterpart

1:10 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

1:10 AM - Responded in 10s

Before shipping the "legacy_account_id" column drop to production, verifying the following aspects is crucial to ensure a successful and risk-free migration:

1. Data Quality and Consistency:

- Migrated Accounts: Verify that the 1204 manually migrated accounts have been successfully updated in the "raw.customers" table and downstream models.
- Data Integrity: Ensure that there are no duplicate or conflicting records introduced during the migration process.
- Null/Empty Values: Check for any null or empty values in the "crm_account_uuid" column that might indicate incomplete or failed migrations.

2. Model and ETL Job Behavior:

- Intermediate and Staging Models: Validate that the "staging.stg_customers" and "intermediate.int_customer_identity" models have been updated correctly to reflect the migration.
- Analytics Models:
 - analytics.dim_customer: Test the SCD Type 2 merge functionality to ensure historical data is preserved correctly.
 - analytics.fct_subscriptions: Validate the join between the fact and dimension tables using the "crm_account_uuid" column.
 - analytics.customer_360_snapshot: Confirm that data exports to Salesforce no longer rely on the "legacy_account_id" column.
 - analytics.churn_training_features: Verify that the feature store entity ID uses the correct column ("crm_account_uuid") after the column drop.
- ETL Jobs: Test all ETL jobs involved in the migration to ensure they function correctly with the updated data flow.

3. Performance:

- Query Performance: Validate that queries and joins using the "crm_account_uuid" column perform as expected, as this column might have different indexing or data distribution compared to the "legacy_account_id" column.
- Data Load Time: Ensure that data loads and migrations are completed within acceptable timeframes, especially for larger datasets.

4. End-to-End Testing:

- Perform end-to-end testing, covering all affected models and ETL jobs, to validate data consistency and accuracy from raw data to the final analytics reports or dashboards.

5. Rollback Plan:

- Ensure that the rollback plan is well-documented, tested, and understood by the team. This includes backup data availability and restoration procedures.

6. Communication and Documentation:

- Clearly communicate the migration plan, timeline, and expected impacts to all relevant stakeholders.
- Document the migration process, assumptions, and any resolved or outstanding issues for future reference.

7. User Acceptance Testing (UAT):

- Provide access to the updated environment to end-users for testing and validation of business processes and reports that rely on the affected models.

By verifying these aspects thoroughly, you can increase confidence in a successful migration and minimize the risk of data loss, inconsistencies, or disruptions in production.

Engineering counterpart

1:10 AM - Responded in 0s

What are the top risks, and what's the rollback plan if this doesn't work?

Agent response

1:10 AM - Responded in 10s

Top Risks and Rollback Plan

Top Risks:

1. Data Loss or Inconsistency:

- Failure to correctly migrate the 1204 unmigrated accounts.
- Incorrect handling of historical data in the `analytics.dim\_customer` model.
- Incomplete or failed updates in downstream models and ETL jobs.

2. Breakage of Business Processes or Reports:

- Disruption of the join between `analytics.fct\_subscriptions` and `analytics.dim\_customer`.
- Inability to export data to Salesforce in `analytics.customer\_360\_snapshot`.
- Incorrect feature store entity IDs in `analytics.churn\_training\_features`.

3. Performance Issues:

- Slower query or join performance due to changes in data distribution or indexing.
- Increased data load times during migrations or ETL jobs.

4. Compliance and Data Governance:

- Incomplete or inaccurate data might lead to non-compliance with regulations or internal data governance policies.

5. User Impact:

- Disrupted or incorrect analytics reports, affecting decision-making processes.
- Temporary unavailability of data or reports during the migration.

Rollback Plan:

1. Immediate Rollback:

If data loss, inconsistency, or critical business processes are impacted during the migration, immediately halt the column drop process.

Revert the affected tables in the production environment to their pre-migration state using a backup or the transactions logs (e.g., using Point-In-Time Recovery (PITR) in databases like PostgreSQL).

2. Temporary Workarounds:

If immediate rollback is not feasible or causes significant downtime, implement temporary workarounds to restore critical functionality:

- + Re-enable the "legacy_account_id" column temporarily in the affected models or ETL jobs.
- + Manually update or export specific records to maintain data consistency.

3. Assess and Address Root Causes:

- Investigate the root cause of the issues encountered during the migration.
- Update the migration plan, data flow, or ETL jobs as necessary to address the identified problems.

4. Retry Migration:

Once the root causes have been addressed, and the updated migration plan has been tested thoroughly, retry the column drop migration following the revised plan.

5. Communicate and Document:

- Clearly communicate the rollback, any temporary workarounds, and the revised migration plan to all relevant stakeholders.
- Document the rollback process, lessons learned, and any updates to the migration plan for future reference.

Additional Considerations:

Ensure that backups of the legacy data and transaction logs are maintained throughout the migration process to facilitate a rollback if necessary.

Establish clear thresholds for data loss, inconsistency, or downtime that would trigger a rollback or temporary workaround.

Regularly review and update the rollback plan based on feedback, lessons learned, or changes in the environment.

Conduct periodic dry runs of the rollback plan to ensure familiarity with the process and validate the availability of required resources.

By being aware of the top risks and having a well-defined rollback plan in place, you can minimize the impact of any issues

that may arise during the "legacy_account_id" column drop migration and ensure a smoother recovery process if required.

Engineering counterpart
1:10 AM - Responded in 0s

Can you tighten this into concrete next steps with owners and timing?

Agent response

1:11 AM - Responded in 19s

Concrete Next Steps with Owners and Timing

1. Data Quality and Consistency Verification (2 weeks)

| Task | Owner | Duration |

|---|---|---|

| Verify migrated accounts | Data Engineer | 1 day |

| Check data integrity and null/empty values | Data Engineer | 2 days |

| Validate intermediate and staging models | ETL Developer | 3 days |

| Test analytics models (dim_customer, fct_subscriptions, customer_360_snapshot, churn_training_features) | Data Engineer, Analytics Engineer, ETL Developer (as needed) | 5 days |

| Total | | 11 days |

2. Performance Testing (1 week)

| Task | Owner | Duration |

|---|---|---|

| Query and join performance testing | Data Engineer | 2 days |

| Data load time testing | ETL Developer | 3 days |

| Total | | 5 days |

3. End-to-End Testing (2 weeks)

| Task | Owner | Duration |

|---|---|---|

| Prepare test cases and data | Data Engineer, QA Tester | 2 days |

| Execute end-to-end tests | QA Tester | 5 days |

| Analyze and address any issues | Data Engineer, ETL Developer, Analytics Engineer (as needed) | 3 days |

| Total | | 10 days |

4. Rollback Plan Validation and Documentation (1 week)

| Task | Owner | Duration |

|---|---|---|

| Review and update rollback plan | Data Engineer, Database Admin, Team Lead | 2 days |

| Conduct dry run of rollback plan | Data Engineer, Database Admin, Team Lead | 3 days |

| Document rollback plan and lessons learned | Data Engineer, Team Lead | 2 days |

| Total | | 7 days |

5. User Acceptance Testing (UAT) and Final Approval (1-2 weeks)

| Task | Owner | Duration |

|---|---|---|

| Prepare UAT environment and documentation | Data Engineer, Team Lead | 2 days |

| Conduct UAT with end-users | End-users, Data Engineer, Team Lead | 5-10 days (depending on user availability and feedback) |

| Address any UAT-related issues | Data Engineer, ETL Developer, Analytics Engineer (as needed) | 3 days |

| Obtain final approval from stakeholders | Team Lead, Data Governance | 2 days |

| Total | | 12 days (max) |

6. Migration to Production (1 day)

| Task | Owner | Duration |

|---|---|---|

| Deploy updated models and ETL jobs to production | ETL Developer, Data Engineer | 4 hours |

| Monitor migration progress and address any issues | Data Engineer, ETL Developer, On-call Team Member | 8 hours |

| Total | | 12 hours |

7. Post-Migration Monitoring and Support (Ongoing)

| Task | Owner | Duration |

|---|---|---|

| Monitor data quality and consistency post-migration | Data Engineer | Ongoing (weekly checks for 1 month, then monthly)

|

| Address any post-migration issues or user feedback | Data Engineer, ETL Developer, Analytics Engineer (as needed) | Ongoing |

| Document lessons learned and updates to migration plan | Data Engineer, Team Lead | Ongoing |

By breaking down the verification and rollback preparation process into concrete next steps with clear owners and timelines, you can effectively manage the "legacy_account_id" column drop migration and ensure a well-orchestrated, low-risk deployment to production. Regularly review and update these steps as needed to adapt to changing circumstances or feedback from stakeholders.

Engineering counterpart**1:11 AM - Responded in 0s**

What's the one thing you'd want a second pair of eyes on?

Agent response**1:11 AM - Responded in 7s**

The one thing I'd want a second pair of eyes on is the verification of data consistency and integrity throughout the migration process, from the raw data to the final analytics models. This is crucial to ensure that the column drop and migration have been successfully implemented without introducing data loss, inconsistencies, or inaccuracies.

Here's why it's essential to involve a second pair of eyes:

1. Catch potential oversights or misunderstandings: A second reviewer can help identify any missed dependencies, incorrect assumptions, or overlooked data issues that might have been missed during the initial assessment.
2. Validate data flow and transformation logic: By reviewing the data flow and transformation logic in ETL jobs, a second pair of eyes can ensure that data is correctly handled at each stage of the migration, from the raw data to the final models.
3. Confirm the accuracy of test results: A second reviewer can independently verify the results of data quality checks, performance testing, and end-to-end testing, ensuring that the migration has been successful and that any identified issues have been addressed appropriately.
4. Cross-check rollback plan and assumptions: Involving a second pair of eyes in reviewing the rollback plan can help validate assumptions, ensure that all necessary steps are covered, and identify any potential gaps or weaknesses in the plan.

To make the most of a second pair of eyes, consider the following:

Choose a reviewer with relevant expertise and domain knowledge.

Clearly communicate the goals, scope, and expectations of the review.

Provide the reviewer with all necessary documentation, access to tools, and any relevant background information.

Encourage the reviewer to ask questions, challenge assumptions, and provide constructive feedback.

Collaborate with the reviewer to address any identified issues or concerns, updating the migration plan and documentation as needed.

By having a second pair of eyes focus on data consistency and integrity, you can increase confidence in the success of the "legacy_account_id" column drop migration and minimize the risk of data-related issues in production.

Engineering counterpart**1:11 AM - Responded in 0s**

Thanks - impact read and migration risks are clear. Migrate unmigrated accounts before dropping the legacy column in prod. Review complete - no further response needed.