



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: meta-llama/llama-4-maverick

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

Duration: 1m 51s

Actual run cost: \$0.003480

Projected execution cost: \$0.7656/mo (10/biz day × 22 days)

Report date: July 14, 2026 - 3:01 AM EST

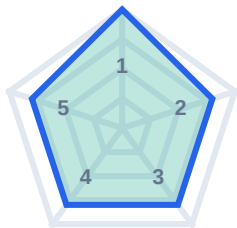
Overall

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: 15612 chars - Latency: 111s - 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 51s

Engineering counterpart

6:59 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

7:00 AM - Responded in 14s

Impact Review: Dropping `legacy\_account\_id` Column

Affected Marts/Models

The proposed change to drop the `legacy\_account\_id` column from `raw.customers` will directly and indirectly affect the following models:

1. Direct Dependents:

- `staging.stg\_customers`: References `legacy\_account\_id`.
- `intermediate.int\_customer\_identity`: References both `legacy\_account\_id` and `crm\_account\_uuid`.

2. Downstream Marts/Models:

- `analytics.dim\_customer`: High risk. Uses `legacy\_account\_id` as a natural key in SCD2 merge logic.
- `analytics.fct\_subscriptions`: High risk. Joins on `legacy\_account\_id` to `dim\_customer`.
- `analytics.customer\_360\_snapshot`: Medium risk. Uses `legacy\_account\_id` in export to Salesforce.
- `analytics.churn\_training\_features`: Medium risk. Uses `legacy\_account\_id` in feature store entity id.

Migration Risks for Unmigrated Accounts

As of the current state, `crm\_account\_uuid` is populated for 94.2% of the accounts, leaving 5.8% (specifically, 1204 accounts) that still rely on `legacy\_account\_id`. The risks associated with unmigrated accounts include:

- Data Inconsistencies: Until all accounts are migrated, there will be a mix of `legacy\_account\_id` and `crm\_account\_uuid` being used across different models and processes.
- SCD2 Merge Issues: `analytics.dim\_customer` uses `legacy\_account\_id` for SCD2 merge. Unmigrated accounts might cause inconsistencies or errors if `legacy\_account\_id` is dropped before these accounts are fully migrated to use `crm\_account\_uuid`.
- Downstream Process Failures: Models like `analytics.fct\_subscriptions` that join on `legacy\_account\_id` might fail or produce incorrect results for unmigrated accounts if `legacy\_account\_id` is no longer available.

Recommended Migration Order

To minimize disruption and risk, the following migration order is recommended:

1. Migrate `analytics.dim\_customer`: This is a high-risk model that uses `legacy\_account\_id` as a natural key. Migrating it to use `crm\_account\_uuid` should be prioritized.
2. Update `analytics.fct\_subscriptions`: Since it joins on `legacy\_account\_id` to `dim\_customer`, its migration should follow or be done in tandem with `dim\_customer` to ensure data consistency.
3. Address `analytics.customer\_360\_snapshot` and `analytics.churn\_training\_features`: These models are categorized as medium risk. Their migration should be planned and executed, focusing on updating the logic to use `crm\_account\_uuid` instead of `legacy\_account\_id`.
4. Complete CRM Migration for Remaining Accounts: Ensure that the remaining 1204 accounts are migrated to use `crm\_account\_uuid`. This is crucial before dropping `legacy\_account\_id`.

Rollback Considerations

In case issues arise after dropping `legacy\_account\_id`, the following rollback considerations should be in place:

1. Restore `legacy\_account\_id`: If technically feasible, be prepared to restore the column temporarily until the issues are resolved.
2. Data Reconciliation: Have a plan to reconcile data that might have been processed or generated during the period when `legacy\_account\_id` was dropped, to ensure data integrity and consistency.
3. Rollback of Dependent Models: Be prepared to rollback changes made to dependent models if they are not compatible

with the absence of `legacy\_account\_id`.

4. Monitoring and Quick Response: Closely monitor downstream processes and models post-change. Have a quick response plan in place to address any issues that arise.

Deadline and Next Steps

Given the deadline is `2026-07-12 (CRM cutover)`, it's essential to:

1. Accelerate Migration: Prior

Engineering counterpart
7:00 AM - Responded in 0s

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

Agent response

7:00 AM - Responded in 42s

Let's break down the reasoning step by step.

Step 1: Understand the Proposed Change

1. Action: DROP COLUMN
2. Table: `raw.customers`
3. Column: `legacy\_account\_id`
4. Reason: Deprecated after CRM migration - replaced by `crm\_account\_uuid`

The task is to assess the impact of dropping a legacy column that has been largely replaced by a new column (`crm\_account\_uuid`) due to a CRM migration.

Step 2: Analyze Direct Dependents

The direct dependents are models that directly reference the `legacy\_account\_id` column.

1. `staging.stg\_customers`: References `legacy\_account\_id`.
 - Assumption: This model likely uses `legacy\_account\_id` for some transformation or validation. Dropping `legacy\_account\_id` will require updating this model.
2. `intermediate.int\_customer\_identity`: References both `legacy\_account\_id` and `crm\_account\_uuid`.
 - Assumption: This model is likely in the process of transitioning from `legacy\_account\_id` to `crm\_account\_uuid`. It might be using both columns for validation or to handle accounts that have not been fully migrated.

Step 3: Assess Downstream Marts/Models

The downstream marts/models are those that are indirectly affected by the change.

1. `analytics.dim\_customer`: High risk. Uses `legacy\_account\_id` as a natural key in SCD2 merge logic.
 - Assumption: SCD2 (Slowly Changing Dimension Type 2) merge logic is sensitive to the natural key. Changing or removing it could cause data inconsistencies or errors. This model is critical and requires careful migration.
2. `analytics.fct\_subscriptions`: High risk. Joins on `legacy\_account\_id` to `dim\_customer`.
 - Assumption: This model relies on `dim\_customer` and uses `legacy\_account\_id` for joining. Its data integrity is dependent on the correctness of `dim\_customer`, making it high risk.
3. `analytics.customer\_360\_snapshot` and `analytics.churn\_training\_features`: Medium risk.
 - Assumption: These models use `legacy\_account\_id` for specific purposes (export to Salesforce and feature store entity id, respectively). While important, they are not as critical to the core data integrity as `dim\_customer` and `fct\_subscriptions`.

Step 4: Evaluate Migration State and Risks

1. `crm\_account\_uuid\_populated\_pct`: 94.2%
2. `legacy\_only\_accounts`: 1204

- Assumption: The remaining 5.8% (1204 accounts) are still using `legacy\_account\_id` and have not been migrated to `crm\_account\_uuid`. There is a risk associated with these unmigrated accounts, especially if `legacy\_account\_id` is dropped before they are fully migrated.

Step 5: Determine Recommended Migration Order

1. Prioritize High-Risk Models: `analytics.dim\_customer` and `analytics.fct\_subscriptions` are high risk due to their critical

roles and dependency on `legacy\_account\_id`.

- Assumption: Migrating these models first will mitigate the most significant risks associated with dropping `legacy\_account\_id`.

2. Address Medium-Risk Models: `analytics.customer\_360\_snapshot` and `analytics.churn\_training\_features`.

- Assumption: These models, while important, are less critical than `dim\_customer` and `fct\_subscriptions`. Their migration can follow the high-risk models.

3. Complete CRM Migration: Ensure all remaining accounts are migrated to `crm\_account\_uuid`.

- Assumption: This is a prerequisite for safely dropping `legacy\_account\_id`.

Step 6: Consider Rollback

- Assumption: In case of issues post-drop, being able to restore `legacy\_account\_id` or rollback changes to dependent models is crucial. This requires having a clear understanding of

Engineering counterpart
7:00 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

7:01 AM - Responded in 19s

Before shipping the changes to production, I would verify the following:

1. Migration Completeness

- Verify that all dependent models have been updated: Ensure that `staging.stg\_customers`, `intermediate.int\_customer\_identity`, `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` have been updated to no longer rely on `legacy\_account\_id`.
- Confirm that all accounts have been migrated: Check that the remaining 1204 accounts (or any other unmigrated accounts) have been successfully migrated to use `crm\_account\_uuid`.

2. Data Integrity and Consistency

- Run data validation tests: Execute tests to verify that data in the affected models is consistent and accurate after the migration.
- Compare data before and after migration: Perform a comparison of data in the affected models before and after the migration to ensure that there are no discrepancies.

3. Downstream Process Validation

- Test downstream processes: Validate that downstream processes, such as exports to Salesforce and feature store entity id generation, continue to work correctly after the migration.
- Monitor data pipelines: Keep a close eye on data pipelines and workflows that rely on the affected models to ensure they are functioning as expected.

4. Rollback Plan

- Verify rollback procedures: Ensure that rollback procedures are in place and have been tested, in case issues arise after dropping `legacy\_account\_id`.
- Confirm data recovery: Validate that data can be recovered in case of a rollback.

5. Production Readiness

- Code review: Perform a thorough code review to ensure that all changes are correct and follow best practices.
- Testing in a production-like environment: Test the changes in a production-like environment to identify any potential issues that might not be apparent in lower environments.
- Monitoring and alerting: Set up monitoring and alerting to quickly detect any issues that may arise after the change is deployed to production.

6. Stakeholder Communication

- Inform stakeholders: Notify relevant stakeholders, including data consumers and business teams, about the upcoming change and its potential impact.
- Coordinate with dependent teams: Coordinate with teams that rely on the affected models to ensure they are aware of the changes and have taken necessary steps to adapt.

By verifying these aspects, you can ensure a smooth transition and minimize the risk of disruptions or data inconsistencies in production.

Engineering counterpart

7:01 AM - Responded in 0s

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

Agent response

7:01 AM - Responded in 15s

Top Risks

Based on the analysis, the top risks associated with dropping the `legacy\_account\_id` column are:

1. Data Inconsistencies: Unmigrated accounts (1204) might cause data inconsistencies or errors if `legacy\_account\_id` is dropped before they are fully migrated to use `crm\_account\_uuid`.
2. SCD2 Merge Issues: `analytics.dim\_customer` uses `legacy\_account\_id` as a natural key. Dropping it before completing the migration might cause SCD2 merge issues.
3. Downstream Process Failures: Models like `analytics.fct\_subscriptions` that join on `legacy\_account\_id` might fail or produce incorrect results for unmigrated accounts.
4. Data Loss: If the migration is not properly validated, there is a risk of data loss or corruption.

Rollback Plan

In case the changes do not work as expected, the rollback plan would involve:

1. Restore `legacy\_account\_id` Column: Restore the `legacy\_account\_id` column to its original state, if technically feasible.
2. Revert Changes to Dependent Models: Revert changes made to dependent models, such as `staging.stg\_customers`, `intermediate.int\_customer\_identity`, `analytics.dim\_customer`, and others, to their previous state.
3. Re-sync Data: Re-sync data that might have been processed or generated during the period when `legacy\_account\_id` was dropped, to ensure data integrity and consistency.
4. Re-validate Downstream Processes: Re-validate downstream processes and models to ensure they are functioning correctly with the restored `legacy\_account\_id` column.

Rollback Considerations

To facilitate a smooth rollback, consider the following:

1. Backup Data: Take a backup of the data before making any changes, to ensure that it can be restored if needed.
2. Document Changes: Document all changes made to the dependent models and data pipelines, to facilitate a quick rollback.
3. Test Rollback: Test the rollback process in a non-production environment to ensure it works as expected.
4. Communicate with Stakeholders: Inform stakeholders about the potential risks and the rollback plan, to manage expectations and ensure a coordinated response in case of issues.

By having a well-planned rollback strategy in place, you can minimize the risk of data inconsistencies and downstream process failures, and ensure a quick recovery in case something goes wrong.

Engineering counterpart

7:01 AM - Responded in 0s

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

Agent response

7:01 AM - Responded in 14s

Concrete Next Steps

To ensure a successful migration and minimize risks, the following concrete next steps are recommended:

1. Migrate High-Risk Models:

- Owner: Data Engineering Team
- Task: Update ``analytics.dim_customer`` and ``analytics.fct_subscriptions`` to use ``crm_account_uuid`` instead of ``legacy_account_id``.
- Timing: Complete within the next 2 weeks (by 2026-06-26).

2. Complete CRM Migration for Remaining Accounts:

- Owner: CRM Team
- Task: Migrate the remaining 1204 accounts to use ``crm_account_uuid``.
- Timing: Complete within the next 4 weeks (by 2026-07-10).

3. Update Medium-Risk Models:

- Owner: Data Engineering Team
- Task: Update ``analytics.customer_360_snapshot`` and ``analytics.churn_training_features`` to use ``crm_account_uuid`` instead of ``legacy_account_id``.
- Timing: Complete within the next 6 weeks (by 2026-07-24), after high-risk models are migrated.

4. Validate Data Integrity and Downstream Processes:

- Owner: Data Quality Team
- Task: Validate data integrity and downstream processes after the migration.
- Timing: Complete within 1 week after the CRM migration is complete (by 2026-07-17).

5. Drop ``legacy_account_id`` Column:

- Owner: Data Engineering Team
- Task: Drop the ``legacy_account_id`` column from ``raw.customers``.
- Timing: After validation is complete and CRM migration is finished (by 2026-07-19).

6. Monitor and Review:

- Owner: Data Engineering and Data Quality Teams
- Task: Monitor data pipelines and review data quality after the change.
- Timing: Ongoing, starting from 2026-07-19.

Rollback Plan and Timing

In case issues arise after dropping ``legacy_account_id``, the rollback plan will be executed within 3 business days.

1. Restore ``legacy_account_id`` Column: Restore the column to its original state.
2. Revert Changes to Dependent Models: Revert changes made to dependent models.
3. Re-sync Data: Re-sync data that might have been processed or generated during the period when ``legacy_account_id`` was dropped.

By following these concrete next steps, we can ensure a successful migration and minimize the risk of data inconsistencies and downstream process failures.

Engineering counterpart

7:01 AM - Responded in 0s

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

Agent response

7:01 AM - Responded in 8s

The one thing I'd want a second pair of eyes on is the validation of data integrity and downstream processes after the migration.

Specifically, I'd want to review the test cases and validation scripts used to ensure that:

1. Data in `analytics.dim\_customer` and `analytics.fct\_subscriptions` is accurate and consistent after the migration.
2. Downstream processes, such as exports to Salesforce and feature store entity id generation, are working correctly.

Having a second pair of eyes on this critical step can help:

1. Catch any potential data inconsistencies or issues that might have been missed.
2. Ensure that the validation is comprehensive and covers all critical aspects of the data and downstream processes.
3. Provide an independent review of the test cases and validation scripts to ensure they are robust and effective.

This review can be done by another experienced data engineer or a data quality specialist, and can help provide additional confidence in the migration process.

Engineering counterpart

7:01 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.