



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-3.3-70b-instruct

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

Duration: 1m 38s

Actual run cost: \$0.002010

Projected execution cost: \$0.4422/mo (10/biz day × 22 days)

Report date: July 14, 2026 - 2:00 AM EST

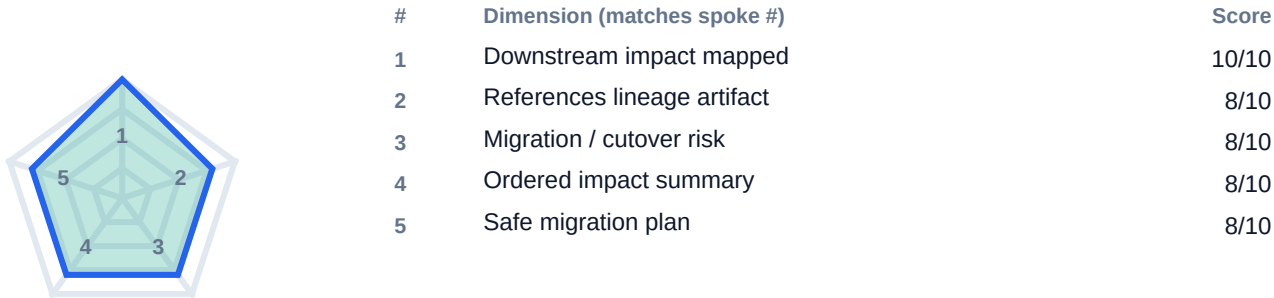
Overall

8.4/10

PASS - trust=high

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

CAPABILITY RUBRIC



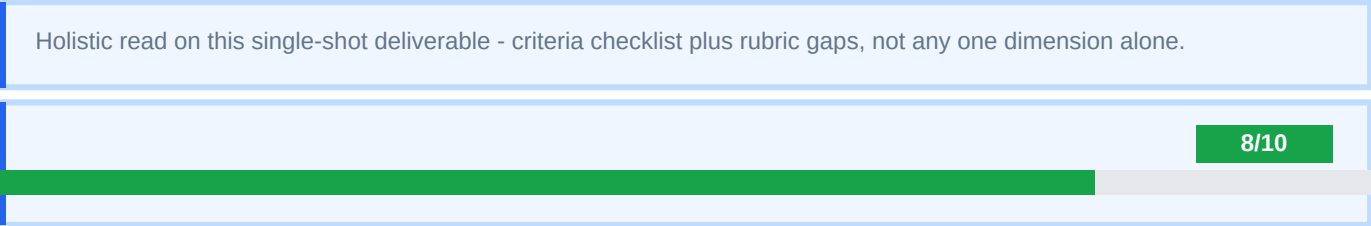
Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

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



Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 6 - Counterpart msgs: 7 - Output: 16824 chars - Latency: 98s - 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 38s

Engineering counterpart

5:58 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

5:58 AM - Responded in 8s

Lineage Impact Review Report

=====

Introduction

This report assesses the downstream impact of dropping the `legacy\_account\_id` column from the `raw.customers` table. The proposed change aims to remove a deprecated column replaced by `crm\_account\_uuid` after the CRM migration.

Affected Marts and Models

The following marts and models will be impacted by the proposed change:

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:

`analytics.dim\_customer` (High Risk): uses `legacy\_account\_id` as a natural key in SCD2 merge

`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

Legacy-only accounts: 1204 accounts (approximately 5.8% of the total) still use the `legacy\_account\_id` and have not been migrated to `crm\_account\_uuid`.

Risk of data loss or corruption: If the `legacy\_account\_id` column is dropped before these accounts are migrated, it may lead to data loss or corruption, particularly in downstream marts that rely on this column.

Recommended Migration Order

To minimize risks and ensure a smooth transition, the following migration order is recommended:

1. Migrate legacy-only accounts: Update the remaining 1204 accounts to use `crm\_account\_uuid` before dropping the `legacy\_account\_id` column.

2. Update intermediate models:

Modify `staging.stg\_customers` to reference `crm\_account\_uuid` instead of `legacy\_account\_id`.

Update `intermediate.int\_customer\_identity` to use `crm\_account\_uuid` as the primary identifier.

3. Update downstream marts:

Modify `analytics.dim\_customer` to use `crm\_account\_uuid` as the natural key in SCD2 merge.

Update `analytics.fct\_subscriptions` to join on `crm\_account\_uuid` instead of `legacy\_account\_id`.

Update `analytics.customer\_360\_snapshot` to use `crm\_account\_uuid` in export to Salesforce.

Update `analytics.churn\_training\_features` to use `crm\_account\_uuid` in feature store entity id.

4. Drop legacy column: Once all dependent models and marts have been updated, the `legacy\_account\_id` column can be safely dropped from the `raw.customers` table.

Rollback Considerations

In case of issues or errors during the migration process, the following rollback strategies should be considered:

Restore the legacy column: If the `legacy\_account\_id` column is dropped prematurely, it can be restored from backups or recreated using historical data.

Revert model changes: If updates to intermediate models or downstream marts cause issues, the changes can be reverted to their previous state.

Use temporary workaround: If some accounts are not yet migrated, a temporary workaround can be implemented to use both `legacy\_account\_id` and `crm\_account\_uuid` until the migration is complete.

By following the recommended migration order and considering the potential risks and rollback strategies, the proposed change can be successfully implemented, and the `legacy\_account\_id` column can be safely dropped from the `raw.customers` table.

Engineering counterpart
5:58 AM - Responded in 0s

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

Agent response

5:58 AM - Responded in 26s

I'll break down my thought process into steps and highlight the assumptions I made.

Step 1: Understand the proposed change

I reviewed the `proposed\_change` section and understood that the `legacy\_account\_id` column in the `raw.customers` table is proposed to be dropped.

Assumption: The `crm\_account\_uuid` column is a suitable replacement for `legacy\_account\_id` and has been properly populated for most accounts.

Step 2: Identify direct dependents

I examined the `direct\_dependents` section and found two models that reference the `legacy\_account\_id` column: `staging.stg\_customers` and `intermediate.int\_customer\_identity`.

Assumption: These models are critical to the data pipeline and must be updated to reference the new `crm\_account\_uuid` column.

Step 3: Assess downstream marts

I analyzed the `downstream\_marts` section and identified four marts that use the `legacy\_account\_id` column: `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features`.

Assumption: The risk levels assigned to each mart (High, Medium) are accurate and reflect the potential impact of dropping the `legacy\_account\_id` column.

Step 4: Evaluate migration risks

I reviewed the `migration\_state` section and noted that 94.2% of accounts have been migrated to use `crm\_account\_uuid`, leaving 1204 accounts still using `legacy\_account\_id`.

Assumption: The remaining 1204 accounts are critical to the business and must be migrated before dropping the `legacy\_account\_id` column.

Step 5: Develop a migration plan

Based on the analysis, I recommended a step-by-step migration plan:

1. Migrate the remaining 1204 accounts to use `crm\_account\_uuid`.
2. Update intermediate models (`staging.stg\_customers` and `intermediate.int\_customer\_identity`) to reference `crm\_account\_uuid`.
3. Update downstream marts to use `crm\_account\_uuid`.
4. Drop the `legacy\_account\_id` column.

Assumption: This order of operations will minimize disruptions to the data pipeline and ensure a smooth transition.

Step 6: Consider rollback strategies

I identified potential rollback strategies in case of issues or errors during the migration process:

- + Restore the `legacy\_account\_id` column from backups or recreate it using historical data.
- + Revert changes to intermediate models or downstream marts.
- + Implement temporary workarounds to use both `legacy\_account\_id` and `crm\_account\_uuid` until the migration is complete.

Assumption: These rollback strategies will be effective in mitigating any potential issues that may arise during the migration process.

Key assumptions made during this analysis include:

The `crm\_account\_uuid` column is a suitable replacement for `legacy\_account\_id`.

The risk levels assigned to each downstream mart are accurate.

The remaining 1204 accounts are critical to the business and must be migrated before dropping the `legacy\_account\_id` column.

The recommended migration plan will minimize disruptions to the data pipeline.

The identified rollback strategies will be effective in mitigating potential issues.

These assumptions should be validated and verified with stakeholders and subject matter experts to ensure the proposed change is successfully implemented.

Engineering counterpart
5:58 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

5:59 AM - Responded in 24s

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

1. Migration of legacy-only accounts

Confirm that the remaining 1204 accounts have been successfully migrated to use `crm\_account\_uuid`. Verify that these accounts are properly updated in the `raw.customers` table and all downstream models.

2. Intermediate model updates

Verify that the `staging.stg\_customers` and `intermediate.int\_customer\_identity` models have been updated to reference `crm\_account\_uuid` instead of `legacy\_account\_id`. Confirm that these updates do not introduce any data inconsistencies or errors.

3. Downstream mart updates

Verify that the `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` marts have been updated to use `crm\_account\_uuid`. Confirm that these updates do not affect the data quality or integrity of these marts.

4. Data validation

Perform data validation checks to ensure that the data in the `raw.customers` table and downstream models is consistent and accurate. Verify that the `crm\_account\_uuid` column is properly populated and matches the expected values.

5. Test cases

Develop and execute comprehensive test cases to ensure that the changes do not introduce any regressions or issues. Test cases should cover various scenarios, including:

- + Data ingestion and processing
- + Data transformations and aggregations
- + Data queries and retrieval

6. Rollback strategy

Verify that a rollback strategy is in place in case of issues or errors during the deployment. Confirm that the rollback strategy includes procedures for restoring the `legacy\_account\_id` column, reverting changes to intermediate models and downstream marts, and implementing temporary workarounds.

7. Monitoring and logging

Ensure that proper monitoring and logging mechanisms are in place to detect any issues or errors during the deployment. Verify that the monitoring and logging mechanisms provide sufficient visibility into the data pipeline and allow for quick identification and resolution of problems.

8. Stakeholder sign-off

Obtain sign-off from relevant stakeholders, including data engineers, data analysts, and business stakeholders, to ensure that the changes meet their requirements and expectations. Confirm that all stakeholders are aware of the changes and their potential impact on the data pipeline.

9. Deployment plan

Develop a detailed deployment plan that outlines the steps required to deploy the changes to production.

Verify that the deployment plan includes procedures for:

- + Deploying the changes to production
- + Validating the deployment
- + Monitoring the deployment
- + Rolling back the changes if necessary

By verifying these items, you can ensure a smooth and successful deployment of the changes to production, minimizing the risk of errors or issues affecting the data pipeline.

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

Agent response

5:59 AM - Responded in 7s

Top Risks:

1. Data Loss or Corruption: Dropping the `legacy\_account\_id` column before all dependent models and marts are updated may result in data loss or corruption, particularly in downstream marts that rely on this column.
2. Disruption to Downstream Processes: Changes to the `raw.customers` table and dependent models may disrupt downstream processes, such as data ingestion, transformations, and aggregations, leading to delays or errors in reporting and analytics.
3. Inconsistent Data: Inconsistent data may be introduced if the updates to the `crm\_account\_uuid` column are not properly synchronized with the updates to the dependent models and marts.
4. Rollback Challenges: Rolling back the changes may be complex and time-consuming, especially if the `legacy\_account\_id` column is dropped and the data is not properly backed up.

Rollback Plan:

Short-Term Rollback (within 24 hours of deployment)

1. Restore `legacy\_account\_id` column: Restore the `legacy\_account\_id` column from backups or recreate it using historical data.
2. Revert model changes: Revert changes to the `staging.stg\_customers` and `intermediate.int\_customer\_identity` models to reference `legacy\_account\_id` instead of `crm\_account\_uuid`.
3. Update downstream marts: Update the downstream marts to use `legacy\_account\_id` instead of `crm\_account\_uuid`.
4. Re-run data ingestion and transformation jobs: Re-run data ingestion and transformation jobs to ensure data consistency and accuracy.

Long-Term Rollback (if short-term rollback is not feasible)

1. Re-create `legacy\_account\_id` column: Re-create the `legacy\_account\_id` column using historical data and update the `raw.customers` table.
2. Re-develop and re-deploy models: Re-develop and re-deploy the `staging.stg\_customers` and `intermediate.int\_customer\_identity` models to reference `legacy\_account\_id` instead of `crm\_account\_uuid`.
3. Re-update downstream marts: Re-update the downstream marts to use `legacy\_account\_id` instead of `crm\_account\_uuid`.
4. Re-run data ingestion and transformation jobs: Re-run data ingestion and transformation jobs to ensure data consistency and accuracy.

Mitigation Strategies:

1. Backup data: Take regular backups of the `raw.customers` table and dependent models to ensure data can be restored in case of issues.
2. Monitor data pipeline: Closely monitor the data pipeline and downstream processes to quickly identify and address any issues.
3. Test thoroughly: Thoroughly test the changes before deploying them to production to minimize the risk of errors or issues.
4. Develop a contingency plan: Develop a contingency plan to address potential issues and have a clear rollback strategy in place.

Engineering counterpart
5:59 AM - Responded in 0s

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

Agent response

5:59 AM - Responded in 21s

Here are the concrete next steps with owners and timing:

Short-Term (Next 2 weeks)

1. Complete migration of legacy-only accounts (Owner: Data Engineering Team, Deadline: 2026-06-25)
Migrate the remaining 1204 accounts to use `crm\_account\_uuid`.
Verify that these accounts are properly updated in the `raw.customers` table and all downstream models.
2. Update intermediate models (Owner: Data Engineering Team, Deadline: 2026-06-28)
Update the `staging.stg\_customers` and `intermediate.int\_customer\_identity` models to reference `crm\_account\_uuid` instead of `legacy\_account\_id`.
Test and validate the updates to ensure data consistency and accuracy.
3. Update downstream marts (Owner: Data Engineering Team, Deadline: 2026-07-01)
Update the `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` marts to use `crm\_account\_uuid`.
Test and validate the updates to ensure data consistency and accuracy.

Pre-Deployment (Next 1 week)

1. Final testing and validation (Owner: Data Engineering Team, Deadline: 2026-07-05)
Perform thorough testing and validation of the updated models and marts to ensure data consistency and accuracy.
Identify and address any issues or errors.
2. Backup data (Owner: Data Engineering Team, Deadline: 2026-07-05)
Take regular backups of the `raw.customers` table and dependent models to ensure data can be restored in case of issues.
3. Develop rollback plan (Owner: Data Engineering Team, Deadline: 2026-07-05)
Develop a detailed rollback plan to address potential issues and have a clear strategy in place.

Deployment (2026-07-08)

1. Deploy updates to production (Owner: Data Engineering Team)
Deploy the updated models and marts to production.
Monitor the data pipeline and downstream processes to quickly identify and address any issues.
2. Drop `legacy\_account\_id` column (Owner: Data Engineering Team)
Drop the `legacy\_account\_id` column from the `raw.customers` table after verifying that all dependent models and marts have been updated.

Post-Deployment (Next 2 weeks)

1. Monitor data pipeline (Owner: Data Engineering Team, Deadline: 2026-07-22)
Closely monitor the data pipeline and downstream processes to quickly identify and address any issues.
Verify that the data is consistent and accurate.
2. Verify rollback plan (Owner: Data Engineering Team, Deadline: 2026-07-22)
Verify that the rollback plan is effective and can be executed in case of issues.

By following these concrete next steps with owners and timing, we can ensure a smooth and successful deployment of the changes to production.

Engineering counterpart

5:59 AM - Responded in 0s

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

Agent response

6:00 AM - Responded in 11s

If I had to choose one thing that I'd want a second pair of eyes on, it would be the validation of the updates to the downstream marts.

Specifically, I'd want another engineer or data analyst to review and verify that the updates to the `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` marts are correct and do not introduce any data inconsistencies or errors.

This is because these marts are critical to the business and are used for reporting and analytics, and any errors or inconsistencies could have significant consequences. Additionally, the updates to these marts involve complex transformations and aggregations, which can be prone to errors if not properly validated.

Having a second pair of eyes on this validation process would help to:

- Catch any potential errors or inconsistencies that may have been missed
- Verify that the updates are correct and align with the business requirements
- Ensure that the data is accurate and reliable

Some specific things that I'd want the second pair of eyes to check include:

- Verification of the data transformations and aggregations
- Checking for any data inconsistencies or errors
- Validation of the data against the business requirements
- Review of the data lineage and dependencies to ensure that the updates are properly propagated

By having a second pair of eyes on this validation process, we can ensure that the updates to the downstream marts are accurate, reliable, and meet the business requirements.

Engineering counterpart

6:00 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.