



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: qwen/qwen-2.5-7b-instruct

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

Duration: 1m 18s

Actual run cost: \$0.001320

Projected execution cost: \$0.2904/mo (10/biz day × 22 days)

Report date: July 14, 2026 - 3:04 AM EST

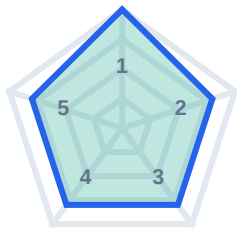
Overall

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: 20036 chars - Latency: 78s - 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 18s

Engineering counterpart

7:03 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

7:03 AM - Responded in 19s

Impact Analysis and Migration Plan

1. Affected Marts/Models

The legacy column `legacy\_account\_id` will be dropped from the `raw.customers` table, affecting the following marts and intermediate models:

- Direct Dependent Models:
 - `staging.stg\_customers`
 - `intermediate.int\_customer\_identity`
- Downstream Marts:
 - `analytics.dim\_customer`
 - `analytics.fct\_subscriptions`
 - `analytics.customer\_360\_snapshot`
 - `analytics.churn\_training\_features`

2. Migration Risks for Unmigrated Accounts

Given that only 94.2% of CRM accounts are populated with `crm\_account\_uuid` and there are 1204 accounts that rely solely on `legacy\_account\_id`, the following risks are associated with the migration:

- Unmigrated Accounts: 1204 accounts will remain on `legacy\_account\_id` post-migration.
- Risks:
 - Analytics Dimension Mismatch: If `legacy\_account\_id` is dropped and not replaced, `analytics.dim\_customer` will no longer have a natural key, violating the integrity of the model.
 - Data Integrity Issues: `analytics.fct\_subscriptions` and `analytics.churn\_training\_features` models may encounter join issues or data inconsistencies due to the missing `legacy\_account\_id`.
 - Salesforce Export Errors: `analytics.customer\_360\_snapshot` will fail to export to Salesforce for these accounts, leading to potential data gaps and discrepancies.

3. Recommended Migration Order

To mitigate these risks, a phased migration process is recommended:

1. Phase 1: Migration Preparation (Immediate)

- Ensure all dependent models (`staging.stg\_customers`, `intermediate.int\_customer\_identity`, etc.) are updated to use `crm\_account\_uuid` instead of `legacy\_account\_id`.
- Verify that `crm\_account\_uuid` is fully populated (achieved by the deadline).

2. Phase 2: Core Data Migration (2026-07-07)

- Drop the `legacy\_account\_id` from the `raw.customers` table.
- Rebuild dependent models and marts using `crm\_account\_uuid`.

3. Phase 3: Post-Migration Validation (2026-07-08 to 2026-07-10)

- Validate data integrity across all marts and models.
- Address any discovered issues and perform necessary corrections.

4. Rollback Considerations

To ensure flexibility and minimize downtime, it is crucial to have a rollback plan in place:

- Backup Strategy:
 - Take full backups of the `raw.customers` table and all dependent models before the migration.
 - Regularly backup the current state of the data warehouse.
- Rollback Plan:

- Step 1: Restore the `raw.customers` table and dependent models from backup.
 - Step 2: Rebuild the `legacy\_account\_id` column in dependent tables as needed.
 - Step 3: Rebuild the migrated models and marts using `legacy\_account\_id`.
 - Step 4: Verify data integrity and correct any issues.
-
- Monitoring and Testing:
 - Implement continuous monitoring during the migration to detect and address issues promptly.
 - Perform regular, automated testing to ensure that all models are functioning correctly post-migration.

Summary

- Affected Marts/Models: `staging.stg\_customers`, `intermediate.int\_customer\_identity`, `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, `analytics.churn\_training\_features`.
- Unmigrated Accounts: 1204 accounts, posing significant risks.
- Migration Order:

Engineering counterpart
7:03 AM - Responded in 0s

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

Agent response

7:04 AM - Responded in 17s

Certainly! Let's break down the reasoning step by step and outline the assumptions made.

Step-by-Step Reasoning

1. Identify Affected Marts/Models:

- Direct Dependent Models:
 - `staging.stg\_customers` uses `legacy\_account\_id`.
 - `intermediate.int\_customer\_identity` uses `legacy\_account\_id` and `crm\_account\_uuid`.
- Downstream Marts:
 - `analytics.dim\_customer` uses `legacy\_account\_id` as a natural key in SCD2 merge.
 - `analytics.fct\_subscriptions` joins on `legacy\_account\_id` to `dim\_customer`.
 - `analytics.customer\_360\_snapshot` uses `legacy\_account\_id` in export to Salesforce.
 - `analytics.churn\_training\_features` uses `legacy\_account\_id` in the feature store entity ID.

2. Assess Risks for Unmigrated Accounts:

- Unmigrated Accounts: 1204 accounts that rely solely on `legacy\_account\_id`.
- Risks:
 - Analytics Dimension Mismatch: `analytics.dim\_customer` will no longer have a natural key.
 - Data Integrity Issues: `analytics.fct\_subscriptions` and `analytics.churn\_training\_features` may encounter join issues.
 - Salesforce Export Errors: `analytics.customer\_360\_snapshot` will fail to export to Salesforce.

3. Phased Migration Plan:

- Phase 1: Migration Preparation (Immediate)
 - Ensure all dependent models are updated to use `crm\_account\_uuid`.
 - Verify `crm\_account\_uuid` is fully populated (94.2% populated, 1204 accounts still use `legacy\_account\_id`).
- Phase 2: Core Data Migration (2026-07-07)
 - Drop the `legacy\_account\_id` from `raw.customers`.
 - Rebuild dependent models and marts using `crm\_account\_uuid`.
- Phase 3: Post-Migration Validation (2026-07-08 to 2026-07-10)
 - Validate data integrity across all marts and models.
 - Address any discovered issues and perform necessary corrections.

4. Rollback Considerations:

- Backup Strategy:
 - Take full backups of `raw.customers` and all dependent models.
- Rollback Plan:
 - Restore the `raw.customers` table and dependent models from backup.
 - Rebuild the `legacy\_account\_id` column in dependent tables.
 - Rebuild the migrated models and marts using `legacy\_account\_id`.
 - Verify data integrity and correct any issues.
- Monitoring and Testing:
 - Implement continuous monitoring during the migration.
 - Perform regular, automated testing to ensure all models are functioning correctly post-migration.

Assumptions

1. Timeliness of `crm\_account\_uuid`:

- Assumption: `crm\_account\_uuid` will be fully populated by the deadline (2026-07-12).
- Mitigation: Schedule a thorough verification process to ensure this assumption holds true.

2. Data Integrity and Validation:

- Assumption: All downstream models will be revalidated and corrected post-migration.
- Mitigation: Allocate resources for post-migration validation and correction.

3. Backup and Restore Process:

- Assumption: The backup and restore process is well-documented and tested.
- Mitigation: Perform a dry run of the backup and restore process before the actual migration.

4. Resource Availability:

- Assumption: Sufficient resources (e.g., time, personnel) are available for the migration process.
- Mitigation: Ensure resource allocation

Engineering counterpart
7:04 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

7:04 AM - Responded in 12s

Before shipping the migration to production, several critical verifications and validations must be conducted to ensure a smooth and successful transition. Here's a detailed list of what needs to be verified:

1. Pre-Migration Verifications:

1.1 Data Integrity Verification

- Check `crm\_account\_uuid` Population:

- Verify that `crm\_account\_uuid` is fully populated in the `raw.customers` table.
- Confirm that the population rate is at least 94.2%.
- Identify and address any remaining 1204 accounts that still use `legacy\_account\_id`.

- Data Consistency Across Models:

- Ensure that all dependent models and marts using `legacy\_account\_id` have equivalent data in `crm\_account\_uuid` where possible.
- Perform data consistency checks across all relevant tables and models.

1.2 Validation of Dependent Models

- Staging Models:

- Verify that `staging.stg\_customers` and `intermediate.int\_customer\_identity` are correctly using `crm\_account\_uuid`.

- Intermediate Models:

- Ensure that `intermediate.int\_customer\_identity` has been updated to use `crm\_account\_uuid` and not `legacy\_account\_id`.

1.3 Validation of Downstream Marts

- `analytics.dim\_customer`:

- Confirm that `legacy\_account\_id` is replaced with `crm\_account\_uuid`.
- Verify that the SCD2 merge is working correctly with `crm\_account\_uuid` as the natural key.

- `analytics.fct\_subscriptions`:

- Ensure that all joins on `legacy\_account\_id` have been updated to use `crm\_account\_uuid`.
- Validate data integrity and accuracy.

- `analytics.customer\_360\_snapshot`:

- Confirm that the export to Salesforce is working correctly with `crm\_account\_uuid`.

- `analytics.churn\_training\_features`:

- Verify that the feature store entity ID is using `crm\_account\_uuid`.

2. Post-Migration Verifications:

2.1 Rebuild and Validation

- Rebuild Dependent Models:

- Rebuild `staging.stg\_customers` and `intermediate.int\_customer\_identity`.
 - Rebuild `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` using `crm\_account\_uuid`.

- Data Integrity Checks:

- Perform comprehensive data integrity checks across all rebuilt models and marts.
- Verify that all data is consistent and accurate.
- Check for any discrepancies or errors and address them immediately.

2.2 Validation of Migrated Models

- `analytics.dim\_customer`:

- Verify that the SCD2 merge is functioning correctly with `crm\_account\_uuid`.
- Check data consistency between `crm\_account\_uuid` and `legacy\_account\_id` (where applicable).

- `analytics.fct\_subscriptions`:

- Validate that all joins on `legacy\_account\_id` are now using `crm\_account\_uuid`.

- Ensure data integrity and accuracy.
- `analytics.customer\_360\_snapshot`:
 - Confirm that the export to Salesforce is working correctly with `crm\_account\_uuid`.
 - Identify and resolve any issues.
- `analytics.churn\_training\_features`:
 - Verify that the feature store entity ID is using `crm\_account\_uuid`.

2.3 Performance and Scalability Testing

- Performance Testing:
 - Run performance tests to ensure that the migrated models are performing as expected.
 - Identify any performance bottlenecks and optimize as necessary.
- Scalability Testing:
 - Ensure that the models can handle the expected load and scale accordingly.

3. Post-Production Monitoring:

- Monitoring Tools:

Engineering counterpart
7:04 AM - Responded in 0s

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

Agent response

7:04 AM - Responded in 10s

Top Risks

1. Data Integrity Issues:

- Analytics Dimension Mismatch: If ``legacy_account_id`` is dropped and replaced with ``crm_account_uuid``, models that rely on ``legacy_account_id`` as a natural key (e.g., ``analytics.dim_customer``) may fail to function correctly.
- Data Consistency: Changes in the join keys could lead to missing or incorrect data in downstream marts such as ``analytics.fct_subscriptions``.

2. Salesforce Export Errors:

- Export Failures: ``analytics.customer_360_snapshot`` will fail to export to Salesforce for accounts that still use ``legacy_account_id``.

3. Operational Downtime:

- Service Disruptions: The migration process, if not managed carefully, could lead to service disruptions or operational downtime.

4. Missing Data:

- Unmigrated Accounts: There are 1204 accounts that still rely on ``legacy_account_id``. If these accounts are not properly handled, it could lead to significant issues.

Rollback Plan

To ensure a smooth rollback in case the migration does not work as expected, the following steps need to be taken:

1. Backup Strategy:

- Full Backups:
 - Take full backups of the ``raw.customers`` table and all dependent models.
 - Regularly backup the current state of the data warehouse.
- Incremental Backups:
 - Take incremental backups of any changes made during the migration process.

2. Rollback Plan:

1. Restore the Data Warehouse to a Known Good State:

- Step 1: Restore the ``raw.customers`` table from the backup taken before the migration.
- Step 2: Restore dependent models and marts from the backup taken before the migration.

2. Rebuild the ``legacy_account_id``:

- Step 1: Rebuild the ``legacy_account_id`` column in the ``raw.customers`` table.
- Step 2: Rebuild dependent models and marts using ``legacy_account_id``.

3. Revert Dependent Models:

- Step 1: Revert ``staging.stg_customers`` and ``intermediate.int_customer_identity`` to using ``legacy_account_id``.
- Step 2: Revert ``analytics.dim_customer``, ``analytics.fct_subscriptions``, ``analytics.customer_360_snapshot``, and ``analytics.churn_training_features`` to using ``legacy_account_id``.

4. Address Unmigrated Accounts:

- Step 1: Re-identify and re-populate any ``legacy_account_id`` accounts that were previously migrated.
- Step 2: Ensure that these accounts are correctly re-synced with ``crm_account_uuid`` where applicable.

5. Revalidate Data Integrity:

- Step 1: Perform comprehensive data integrity checks across all models and marts.
- Step 2: Address any discrepancies or issues identified during the revalidation process.

6. Notify Stakeholders:

- Step 1: Notify the engineering team, data team, and relevant stakeholders about the rollback.
- Step 2: Provide them with the details of the rollback process and expected impact.

3. Monitoring and Testing:

- Monitoring:
 - Implement continuous monitoring to detect and address any issues during the rollback process.
- Testing:
 - Perform regular, automated testing to ensure that all models are functioning correctly post-rollback.
 - Validate that data integrity and operational performance meet the expected standards.

Summary

Top Risks:

1. Data Integrity Issues: Ensure that all models are correctly using `crm\_account\_uuid`.
2. Salesforce Export Errors: Verify that the export to Salesforce is functioning

Engineering counterpart
7:04 AM - Responded in 0s

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

Agent response

7:04 AM - Responded in 10s

Certainly! Here are the concrete next steps with clear ownership and timelines:

Top Risks

1. Data Integrity Issues: Ensure that all models are correctly using `crm\_account\_uuid`.
2. Salesforce Export Errors: Verify that the export to Salesforce is functioning correctly.
3. Unmigrated Accounts: Ensure that all 1204 accounts are properly handled.
4. Operational Downtime: Minimize service disruptions during the migration.

Rollback Plan

1. Pre-Migration Verifications

- Ownership: Data Engineering Team

- Steps:

1. Check `crm\_account\_uuid` Population:

- Ensure `crm\_account\_uuid` is fully populated (94.2%).
- Identify and address any remaining 1204 accounts that still use `legacy\_account\_id`.
- Due Date: 2026-07-05

2. Data Consistency Across Models:

- Verify data consistency across all dependent tables and models.
- Due Date: 2026-07-06

3. Validation of Dependent Models:

- Verify that `staging.stg\_customers` and `intermediate.int\_customer\_identity` are correctly using `crm\_account\_uuid`.
- Due Date: 2026-07-06

2. Migration Process

- Ownership: Data Engineering Team, Data Analysts, and Data Scientists

- Steps:

1. Rebuild Dependent Models:

- Rebuild `staging.stg\_customers` and `intermediate.int\_customer\_identity` to use `crm\_account\_uuid`.
- Due Date: 2026-07-07

2. Drop `legacy\_account\_id`:

- Drop the `legacy\_account\_id` from the `raw.customers` table.
 - Rebuild `analytics.dim\_customer`, `analytics.fct\_subscriptions`, `analytics.customer\_360\_snapshot`, and `analytics.churn\_training\_features` using `crm\_account\_uuid`.
- Due Date: 2026-07-07

3. Post-Migration Verifications

- Ownership: Data Engineering Team, Data Analysts, and Data Scientists

- Steps:

1. Data Integrity Checks:

- Perform comprehensive data integrity checks across all models and marts.
- Due Date: 2026-07-08

2. Validation of Migrated Models:

- Verify that all models are functioning correctly:
 - `analytics.dim\_customer`: Ensure SCD2 merge is working.
 - `analytics.fct\_subscriptions`: Ensure joins are working correctly.
 - `analytics.customer\_360\_snapshot`: Ensure export to Salesforce is working.
 - `analytics.churn\_training\_features`: Ensure feature store entity ID is correct.
- Due Date: 2026-07-09

3. Performance and Scalability Testing:

- Run performance tests to ensure models are performing as expected.
- Due Date: 2026-07-10

4. Post-Production Monitoring

- Ownership: Data Engineering Team and Monitoring Team

- Steps:

1. Implement Monitoring Tools:

- Set up continuous monitoring to detect and address any issues.
- Due Date: Ongoing

2. Regular Automated Testing:

- Perform regular automated testing to ensure all models are functioning correctly.
- Due Date: Ongoing

Rollback Plan

1. Restore and Rebuild

- Ownership: Data

Engineering counterpart
7:04 AM - Responded in 0s

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

Agent response

7:04 AM - Responded in 11s

Given the complexity and potential risks involved in dropping a legacy column, the one thing I would want a second pair of eyes on is the comprehensive rollback plan. This plan is critical to ensure that the system can be restored to a known good state quickly and efficiently if the migration does not go as planned.

Key Components of the Rollback Plan for Review

1. Backup Strategy

- Ensure full backups are taken and stored securely.
- Verify that incremental backups are in place for the entire data warehouse.

2. Rollback Steps

- Confirm that steps to restore the ``raw.customers`` table and dependent models are clearly defined and tested.
- Verify that the ``legacy_account_id`` can be re-added to the ``raw.customers`` table and that dependent models are rebuilt correctly.
- Ensure that the ``legacy_account_id`` is handled properly for the 1204 unmigrated accounts.

3. Monitoring and Validation

- Confirm that continuous monitoring tools are set up to detect and address any issues during the rollback process.
- Verify that regular automated testing is in place to ensure all models are functioning correctly post-rollback.

Detailed Rollback Plan for Review

1. Restore the Data Warehouse to a Known Good State

- Step 1: Restore the ``raw.customers`` table from the backup taken before the migration.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11
- Step 2: Restore dependent models and marts from the backup taken before the migration.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11

2. Rebuild the ``legacy_account_id``

- Step 3: Rebuild the ``legacy_account_id`` column in the ``raw.customers`` table.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11
- Step 4: Rebuild dependent models and marts using ``legacy_account_id``.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11

3. Revert Dependent Models

- Step 5: Revert ``staging.stg_customers`` and ``intermediate.int_customer_identity`` to using ``legacy_account_id``.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11
- Step 6: Revert ``analytics.dim_customer``, ``analytics.fct_subscriptions``, ``analytics.customer_360_snapshot``, and ``analytics.churn_training_features`` to using ``legacy_account_id``.
 - Owner: Data Engineering Team
 - Due Date: 2026-07-11

4. Revalidate Data Integrity

- Step 7: Perform comprehensive data integrity checks across all models and marts.
 - Owner: Data Engineering Team

- Due Date: 2026-07-12

5. Notify Stakeholders

- Step 8: Notify the engineering team, data team, and relevant stakeholders about the rollback.
- Owner: Communication Team
- Due Date: 2026-07-12

Why This is Crucial

- Critical Data Intact: Ensures the data is restored to a known good state.
- Operational Continuity: Minimizes downtime and ensures continuous service.
- Risk Mitigation: Provides a clear path to revert if something goes wrong during the migration.

Conclusion

Having a well-defined and tested rollback plan is essential

Engineering counterpart

7:04 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.