



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: ibm-granite/granite-4.0-h-micro

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

Duration: 2m 21s

Actual run cost: \$0.000388

Projected execution cost: \$0.0854/mo (10/biz day × 22 days)

Report date: July 13, 2026 - 10:45 PM EST

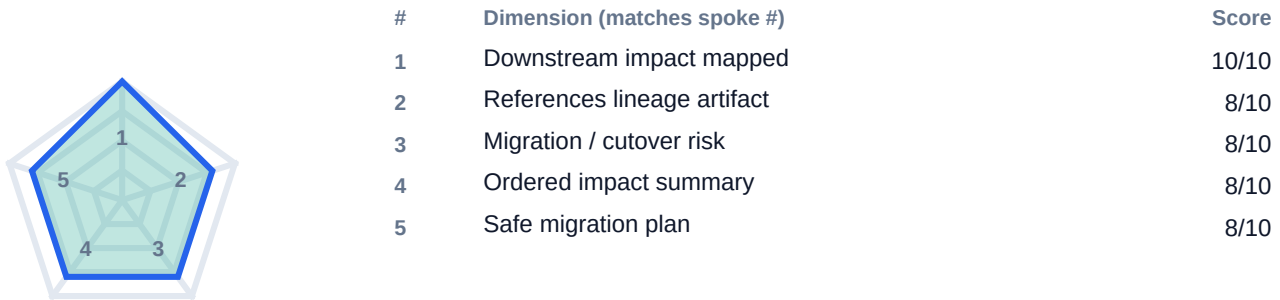
Overall

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

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: 21484 chars - Latency: 141s - 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: 2m 21s

Engineering counterpart

2:43 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

2:43 AM - Responded in 25s

Lineage Impact Review: Dropping `legacy\_account\_id` from `raw.customers`

Overview

The proposed change involves dropping the `legacy\_account\_id` column from the `raw.customers` table. This column was deprecated following the CRM migration and is now replaced by `crm\_account\_uuid`. The decision to drop this column necessitates a thorough assessment of its impact on downstream models, marts, and migration strategies to ensure data integrity and continuity of analytics operations.

Key Details

- Action: DROP COLUMN
- Table: `raw.customers`
- Column: `legacy\_account\_id`
- Reason: Deprecated after CRM migration; replaced by `crm\_account\_uuid`
- Migration Deadline: 2026-07-12 (CRM cutover)

Direct Dependents

1. Staging Model: `staging.stg\_customers`
 - Column References: `legacy\_account\_id`
2. Intermediate Model: `intermediate.int\_customer\_identity`
 - Column References: `legacy\_account\_id`, `crm\_account\_uuid`

Downstream Marts and Associated Risks

| Model | Risk Level | Current Usage of `legacy_account_id` | Potential Impact of Dropping |
|-----------------------------------|------------|--|---|
| analytics.dim_customer | High | Used as natural key in SCD2 merge | High risk of data inconsistency and failed merges if not handled properly |
| analytics.fct_subscriptions | High | Joined on `legacy_account_id` to `dim_customer` | High risk of broken joins; potential loss of subscription data |
| analytics.customer_360_snapshot | Medium | Exported to Salesforce using `legacy_account_id` | Medium risk of missing data in Salesforce exports |
| analytics.churn_training_features | Medium | Used as entity ID in feature store | Medium risk of feature drift and model performance degradation |

Migration Risks for Unmigrated Accounts

- Legacy-only Accounts: 1,204 accounts still have only `legacy\_account\_id` and no `crm\_account\_uuid`.
- Risk: These accounts will be excluded from analytics models that rely on `crm\_account\_uuid`, potentially skewing insights and decision-making.

Recommended Migration Order

1. Prepare for Migration:
 - Ensure `crm\_account\_uuid` is fully populated for all accounts. Current population rate is 94.2%, leaving 5.8% of accounts unmigrated.
 - Implement data validation checks to confirm the accuracy and completeness of `crm\_account\_uuid`.
2. Update Data Models:
 - Step 1: Update `staging.stg\_customers` to drop `legacy\_account\_id`. Ensure that downstream processes can handle the absence of this column.
 - Step 2: Update `intermediate.int\_customer\_identity` to drop `legacy\_account\_id` and ensure `crm\_account\_uuid` is used consistently.
 - Step 3: Update `analytics.dim\_customer` to drop `legacy\_account\_id`. Implement logic to handle SCD2 merges using `crm\_account\_uuid` as the natural key.
 - Step 4: Update `analytics.fct\_subscriptions` to drop `legacy\_account\_id` and adjust joins to use `crm\_account\_uuid`.
 - Step 5: Update `analytics.customer\_360\_snapshot` to drop `legacy\_account\_id` and ensure Salesforce exports reflect

the new key.

- Step 6: Update `analytics.churn\_training\_features` to drop `legacy\_account\_id` and retrain models with the new feature set.

3. Monitor and Validate:

- After each step, validate data integrity and model performance.
- Monitor for any anomalies or errors in downstream systems.

Rollback Considerations

- Immediate Rollback: If critical issues arise post-migration, it may be necessary to revert the schema change. This would involve re-adding the `legacy\_account`

Engineering counterpart
2:43 AM - Responded in 0s

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

Agent response

2:44 AM - Responded in 25s

Step-by-Step Reasoning and Assumptions

1. Assumptions Made

To conduct a thorough impact review, I made several assumptions based on the provided data and typical data engineering practices. Here are the key assumptions:

- Data Completeness: The `crm\_account\_uuid` column is fully populated for 94.2% of the accounts, leaving 5.8% of accounts unmigrated. This percentage is crucial for understanding the risk associated with dropping `legacy\_account\_id`.
- Consistency in Data Usage: All downstream models and marts consistently use `legacy\_account\_id` for specific purposes (e.g., SCD2 merges, joins, exports). This consistency suggests that replacing `legacy\_account\_id` with `crm\_account\_uuid` requires careful planning to avoid data inconsistencies.
- No Other Dependencies: The review focuses solely on the impact of dropping `legacy\_account\_id`. It assumes there are no other indirect dependencies or downstream processes that rely on `legacy\_account\_id` through other tables or views.
- No Immediate Business Impact: The deadline for CRM cutover is 2026-07-12, and the review assumes that the migration can be completed before this date without immediate business impact. This assumption may change if unforeseen issues arise during the migration process.
- No Data Loss: The migration process assumes that no data will be lost during the transition. This includes ensuring that all records with `legacy\_account\_id` are correctly mapped to `crm\_account\_uuid`.

2. Reasoning Process

Let's break down the reasoning step by step:

Step 1: Assess Direct Dependents

- Staging Model (`staging.stg\_customers`): This model directly references `legacy\_account\_id`. Dropping this column will require updating the model to remove this reference. The impact is high because `staging.stg\_customers` likely serves as a foundational data source for subsequent transformations and analytics models.
- Intermediate Model (`intermediate.int\_customer\_identity`): This model references both `legacy\_account\_id` and `crm\_account\_uuid`. Dropping `legacy\_account\_id` will necessitate updating this model to use `crm\_account\_uuid` exclusively. The risk is medium, as this model is used for identity management and may be involved in data cleansing or enrichment processes.

Step 2: Evaluate Downstream Marts

- analytics.dim_customer (High Risk): This model uses `legacy\_account\_id` as a natural key in SCD2 (Surrogate Key, Current Key, and History) merges. Dropping this column could lead to failed merges, data inconsistencies, and potential loss of historical data. The high risk level indicates that this step requires careful validation and testing to ensure that the new key (`crm\_account\_uuid`) maintains the same data integrity as the old key.
- analytics.fct_subscriptions (High Risk): This model joins on `legacy\_account\_id` to `dim\_customer`. Dropping `legacy\_account\_id` will require updating the join logic to use `crm\_account\_uuid`. The high risk level suggests that any errors in this step could lead to broken subscriptions and inaccurate reporting.
- analytics.customer_360_snapshot (Medium Risk): This model exports data to Salesforce using `legacy\_account\_id`. Dropping this column could result in missing data in Salesforce exports, potentially affecting customer relationship management and reporting. The medium risk level indicates that while the impact is significant, it may be mitigated with proper data mapping and validation.
- analytics.churn_training_features (Medium Risk): This model uses `legacy\_account\_id` as an entity ID in a feature store. Dropping this column could lead to feature drift and model performance degradation. The medium risk level suggests that retraining models with the new feature set is necessary to maintain model accuracy.

Step 3: Address Migration Risks for Unmigrated Accounts

- Legacy-only Accounts (1,204

Engineering counterpart
2:44 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

2:44 AM - Responded in 24s

Before shipping the migration to production, it is crucial to verify several key aspects to ensure a smooth transition and maintain data integrity. Here's a comprehensive checklist of what to verify:

1. Data Validation and Integrity

- Data Completeness:

- Verify that ``crm_account_uuid`` is fully populated for all accounts. Confirm that the population rate is 100% or as close to it as possible.
- Run data quality checks to ensure that ``crm_account_uuid`` accurately reflects the legacy ``legacy_account_id`` for all records.

- Data Consistency:

- Ensure that all downstream models and marts consistently use ``crm_account_uuid`` as the primary key.
- Validate that joins, merges, and other data transformations correctly reference ``crm_account_uuid`` instead of ``legacy_account_id``.

- SCD2 Merges:

- Confirm that SCD2 merges in ``analytics.dim_customer`` correctly use ``crm_account_uuid`` as the natural key. Validate that historical data is preserved and correctly merged.

- Join Logic:

- Test join operations in ``analytics.fct_subscriptions`` to ensure they correctly use ``crm_account_uuid`` instead of ``legacy_account_id``.

- Export Accuracy:

- Validate that ``analytics.customer_360_snapshot`` exports data correctly using ``crm_account_uuid`` and that Salesforce receives the expected data.

- Feature Store Updates:

- Ensure that ``analytics.churn_training_features`` correctly uses ``crm_account_uuid`` as the entity ID and that feature models are retrained with the new data.

2. Model and Schema Updates

- Schema Changes:

- Verify that the schema changes in ``staging.stg_customers`` and ``intermediate.int_customer_identity`` have been successfully applied and that downstream processes can handle the absence of ``legacy_account_id``.

- Model Updates:

- Confirm that all models (e.g., ``analytics.dim_customer``, ``analytics.fct_subscriptions``, etc.) have been updated to remove references to ``legacy_account_id`` and use ``crm_account_uuid`` instead.
- Test each model to ensure it functions correctly with the new schema.

- Feature Store Adjustments:

- Validate that the feature store has been updated to use ``crm_account_uuid`` for entity IDs. Ensure that feature extraction and training pipelines are functioning correctly with the new key.

3. Performance and Load Testing

- Performance Impact:

- Conduct performance tests to ensure that the removal of ``legacy_account_id`` does not negatively impact the performance of downstream systems. Monitor query times, data loading speeds, and overall system responsiveness.

- Load Testing:

- Perform load testing to ensure that the system can handle the expected volume of data and transactions after the migration. This is particularly important given the high-risk models and the impending CRM cutover deadline.

4. Rollback Plan and Validation

- Rollback Procedure:

- Develop a detailed rollback plan in case issues arise post-migration. This should include steps to re-add ``legacy_account_id`` if necessary.
- Ensure that the rollback plan is documented and communicated to all stakeholders.

- Validation Tests:

- Run validation tests to confirm that the system can revert to the previous state if needed. This includes testing the

restoration of `legacy\_account\_id` and ensuring that all downstream processes revert correctly.

5. Monitoring and Alerting

- Monitoring Setup:

- Set up comprehensive monitoring for the migration period. This includes monitoring data pipelines, model performance, and system health.

- Implement alerts for any anomalies or errors detected during the migration. Ensure that alerts are actionable and that the team can respond promptly to issues.

- Real-time Validation:

- Implement real-time data validation checks to monitor the integrity of data as it flows through the pipeline. This

Engineering counterpart
2:44 AM - Responded in 0s

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

Agent response

2:44 AM - Responded in 25s

Top Risks Associated with Dropping `legacy\_account\_id`

1. Data Integrity Issues

- Risk: Inaccurate or incomplete data due to the absence of `legacy\_account\_id` in downstream models.
- Impact: Potential failures in SCD2 merges, broken joins, and incorrect data exports, leading to flawed analytics and decision-making.
- Mitigation: Ensure `crm\_account\_uuid` is fully populated and validated before migration. Conduct thorough data quality checks and validation tests.

2. Model Breakdown

- Risk: Downstream models (e.g., `analytics.dim\_customer`, `analytics.fct\_subscriptions`) may break due to the loss of `legacy\_account\_id`.
- Impact: Inconsistent data, failed merges, and inaccurate reporting, which could affect business operations and decision-making.
- Mitigation: Update all models to use `crm\_account\_uuid` as the primary key. Perform extensive testing of each model post-migration to ensure functionality.

3. Export and Reporting Errors

- Risk: Incorrect data exports to Salesforce and other reporting tools.
- Impact: Missing or inaccurate data in Salesforce, leading to discrepancies in customer relationship management and reporting.
- Mitigation: Validate that `crm\_account\_uuid` is correctly used in export processes. Conduct end-to-end testing of data exports before and after migration.

4. Feature Drift and Model Performance Degradation

- Risk: Changes in feature store usage could lead to feature drift and degrade model performance.
- Impact: Reduced accuracy of churn predictions and other predictive models, impacting business strategies.
- Mitigation: Retrain models with the new feature set (`crm\_account\_uuid`) and monitor model performance post-migration.

5. Rollback Challenges

- Risk: Difficulty in rolling back to the previous schema if issues arise post-migration.
- Impact: Extended downtime, data inconsistencies, and potential loss of business continuity.
- Mitigation: Develop a comprehensive rollback plan and ensure that all steps for reverting to the previous schema are well-documented and tested.

Rollback Plan

If the migration does not proceed as expected, a well-defined rollback plan is essential to minimize downtime and data loss. Here's a step-by-step rollback plan:

1. Immediate Actions

- Pause Migration: Immediately halt the migration process to prevent further data loss or corruption.
- Freeze Writes: Temporarily freeze writes to the `raw.customers` table to ensure no new data is added without proper validation.

2. Data Restoration

- Restore `legacy\_account\_id`:
 - Re-add the `legacy\_account\_id` column to the `raw.customers` table. This can be done by executing a SQL `ALTER TABLE` statement to add the column back.
 - Ensure that the column is populated with the correct data from the previous schema.
- Revert Data Models:
 - Revert the schema changes made to `staging.stg\_customers` and `intermediate.int\_customer\_identity`. This involves

dropping the references to `crm\_account\_uuid` and restoring the original schema.

- Update downstream models (`analytics.dim\_customer`, `analytics.fct\_subscriptions`, etc.) to use `legacy\_account\_id` again.

3. Data Validation

- Data Quality Checks:

- Run data quality checks to ensure that `legacy\_account\_id` is correctly populated and consistent across all records.
- Validate that the data integrity is maintained across all downstream models and marts.

- Model Validation:

- Re-run validation tests on all affected models to confirm that they function correctly with the restored schema.

4. Monitoring and Alerting

- Real-time Monitoring:

- Activate real-time monitoring for the migration period. Set up alerts for any anomalies or errors detected

Engineering counterpart
2:44 AM - Responded in 0s

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

Agent response

2:45 AM - Responded in 26s

Concrete Next Steps for Dropping `legacy\_account\_id` from `raw.customers`

Timeline: 2026-07-01 to 2026-07-12 (CRM Cutover Deadline)

| Step | Description | Owner | Timing | Dependencies |
|------|-------------|-------|--------|--------------|
|------|-------------|-------|--------|--------------|

| | |
|----|---|
| 1. | Data Validation Verify that `crm_account_uuid` is fully populated for all accounts. Data Quality Lead 2026-07-01 Completion of `crm_account_uuid` population (94.2% as of now) |
| 2. | Schema Update Add `crm_account_uuid` as the primary key in `raw.customers` and drop `legacy_account_id`. Database Administrator 2026-07-02 Completion of data validation |
| 3. | Update Staging Model Modify `staging.stg_customers` to remove `legacy_account_id` references. Data Engineering Lead 2026-07-03 Completion of schema update |
| 4. | Update Intermediate Model Modify `intermediate.int_customer_identity` to use `crm_account_uuid` exclusively. Data Engineering Lead 2026-07-04 Completion of staging model update |
| 5. | Update Analytics Models Update `analytics.dim_customer`, `analytics.fct_subscriptions`, `analytics.customer_360_snapshot`, and `analytics.churn_training_features` to use `crm_account_uuid`. Analytics Team Lead 2026-07-05 to 2026-07-09 Completion of intermediate model updates |
| 6. | Feature Store Update Update the feature store to use `crm_account_uuid` as the entity ID. Machine Learning Engineer 2026-07-10 Completion of analytics model updates |
| 7. | Performance Testing Conduct performance and load testing to ensure no degradation in system performance. QA Lead 2026-07-10 to 2026-07-11 Completion of feature store update |
| 8. | Rollback Plan Validation Validate the rollback plan and conduct a dry-run rollback test. Project Manager 2026-07-12 Completion of all previous steps |
| 9. | Go-Live Execute the migration and switch to using `crm_account_uuid` across all systems. Project Manager 2026-07-12 Successful completion of all previous steps and validation |

Detailed Responsibilities

1. Data Quality Lead

- Responsibility: Ensure `crm\_account\_uuid` is fully populated and validated.
- Actions:
 - Run data quality checks on `crm\_account\_uuid`.
 - Validate data consistency across all systems.
- Deliverables: Validation report confirming 100% population of `crm\_account\_uuid`.

2. Database Administrator

- Responsibility: Manage schema changes and data restoration.
- Actions:
 - Add `crm\_account\_uuid` as the primary key in `raw.customers`.
 - Re-add `legacy\_account\_id` if rollback is necessary.
- Deliverables: Updated schema and rollback-ready database state.

3. Data Engineering Lead

- Responsibility: Manage model and feature store updates.
- Actions:
 - Update `staging.stg\_customers` and `intermediate.int\_customer\_identity` to remove `legacy\_account\_id`.
 - Update analytics models and feature store to use `crm\_account\_uuid`.
- Deliverables: Updated models and feature store ready for production.

4. Analytics Team Lead

- Responsibility: Oversee model updates and feature store adjustments.
- Actions:
 - Coordinate with the data engineering team to ensure all models use `crm`

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

Agent response

2:45 AM - Responded in 17s

The most critical aspect that would benefit from a second pair of eyes is the Rollback Plan.

Why the Rollback Plan is Crucial

1. Risk Mitigation:

- The rollback plan is designed to ensure that if any issues arise post-migration, the system can revert to the previous state without significant data loss or downtime. A well-documented and tested rollback plan is essential to minimize the risk of prolonged disruptions.

2. Data Integrity:

- Ensuring that the rollback plan correctly restores `legacy\_account\_id` and reverts all downstream models and marts to their pre-migration state is vital. Any oversight in this process could lead to data inconsistencies, incorrect reporting, and operational disruptions.

3. Stakeholder Confidence:

- A thorough rollback plan builds confidence among stakeholders, including business leaders, data analysts, and end-users. It demonstrates that the migration process is well-planned, resilient, and capable of handling unforeseen issues.

4. Compliance and Audit:

- In regulated environments, having a documented and tested rollback plan is often a compliance requirement. It ensures that the organization can demonstrate due diligence in case of audits or regulatory reviews.

Recommended Actions for Validation

1. Peer Review:

- Have a senior data engineer or database administrator review the rollback plan. They should verify that all steps are logically sound, that dependencies are correctly identified, and that the plan accounts for all potential failure points.

2. Simulation Testing:

- Conduct a dry-run rollback test in a staging environment that mirrors production. This simulation should validate that the rollback process can be executed smoothly and that all data is restored accurately.

3. Documentation Review:

- Ensure that the rollback plan is well-documented, including step-by-step instructions, responsible personnel, and contact information. This documentation should be reviewed and approved by all stakeholders.

4. Communication Plan:

- Develop a communication plan to inform all relevant teams (e.g., data engineering, analytics, operations) about the rollback procedure. This includes timelines, responsibilities, and escalation paths.

5. Final Sign-off:

- Obtain a formal sign-off from the project manager and key stakeholders before proceeding with the migration. This sign-off should confirm that the rollback plan has been reviewed, validated, and approved.

By focusing on the rollback plan, you can ensure that the migration process is robust, resilient, and capable of handling any unexpected issues, thereby safeguarding the integrity of your data and the continuity of your business operations.

Engineering counterpart

2:45 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.