



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: inclusionai/ling-2.6-flash

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

Duration: 24s

Actual run cost: \$0.000153

Projected execution cost: \$0.0337/mo (10/biz day × 22 days)

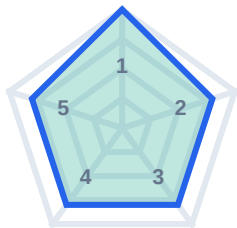
Report date: July 13, 2026 - 10:44 PM EST

Overall

8.4/10

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

CAPABILITY RUBRIC



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

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

Technical criteria checklist

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

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

TECHNICAL JUDGEMENT

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

8/10

Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 6 - Counterpart msgs: 7 - Output: 17049 chars - Latency: 24s - 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: 24s

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 4s

Assumption: The drop is scheduled for after the deadline (2026-07-12) and only after migration is fully complete. If the drop is intended before the deadline, the plan must be executed in reverse (rollback-first) and timeline must be re-sequenced.

1. Affected Marts / Models

| Model | Role in Pipeline | Current Reliance on `legacy_account_id` | Post-Drop Impact if Unmigrated |
|------------------------------------|--------------------------------|--|---|
| staging.stg_customers | Ingestion/standardization | Column present in source, used in pass-through fields and unit tests | Hard failure on any run touching source data; breaks all downstream |
| intermediate.int_customer_identity | Identity resolution (matching) | Uses `legacy_account_id` and `crm_account_uuid` in matching logic and SCD2 | Breaks in entities still in "legacy-only" state; incomplete identity coverage |
| analytics.dim_customer (SCD2) | Conformed dimension | Uses `legacy_account_id` as natural key in SCD2 merge | Merge strategy breaks; surrogate key continuity at risk; type-2 rows stall |
| analytics.fct_subscriptions | Transactional facts | Joins to `dim_customer` via `legacy_account_id` | FK failures or cross-walks to wrong customer; revenue/monetization impact |
| analytics.customer_360_snapshot | Export to Salesforce | Uses `legacy_account_id` for external key mapping | External system sync failures; Salesforce record association lost |
| analytics.churn_training_features | Feature store | Uses `legacy_account_id` as entity ID | Entity misalignment; features detached from CRM identifiers |

Downstream consumers (BI, apps, ML) that rely on any of these models will also be affected (indirect dependents).

2. Migration Risks for Unmigrated Accounts (1,204 legacy-only)

- Identity Resolution Gaps:
 - `legacy\_account\_id` is required to link legacy CRM records to new `crm\_account\_uuid`.
 - If dropped before 100% migration, 1,204 accounts will have no identity anchor, causing:
 - Orphaned transactions in `fct\_subscriptions`
 - Missing customer rows in `dim\_customer`
 - Data loss in Salesforce exports
- SCD2 Merge Failures:
 - Type-2 dimension merges depend on natural key (`legacy\_account\_id`).
 - Dropping the column mid-SCD2 can corrupt slowly changing logic and break conformed dimension integrity.
- Referential Integrity:
 - Fact tables will fail joins if dimension rows are missing or if migration is partial.
 - Surrogate key backfills become inconsistent if identity resolution fails.
- Rollback Complications:
 - Once column is dropped, historical lineage cannot be replayed for pre-drop runs without restoring from backup.
 - Re-adding the column requires schema reversion and backfill of derived values.

3. Recommended Migration Order (Pre-Drop)

1. Complete Data Migration for Legacy Accounts
 - Prioritize the 1,204 legacy-only accounts.

- Ensure `crm\_account\_uuid` is populated and validated (94.2% already done).
- Create explicit mapping table: `legacy\_account\_id -> crm\_account\_uuid`.

2. Update Staging Layer (`staging.stg\_customers`)

- Add derived column `crm\_account\_uuid` using mapping logic.
- Backfill for legacy rows using mapping table

Engineering counterpart
2:43 AM - Responded in 0s

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

Agent response

2:43 AM - Responded in 4s

Below are my step-by-step assumptions and reasoning that led to the prior assessment.

Core assumptions

- The DROP is scheduled after the stated deadline (2026-07-12) and only after migration completes. If the intent is to drop before the deadline, the sequence must invert (rollback-first, then migrate).
- "Legacy accounts" (1,204) are accounts present in the old CRM but not yet mapped to `crm_account_uuid`; until they are migrated, they still depend on `legacy_account_id` as the only stable identifier.
- No explicit backfill or cutover plan was provided, so I assumed a "lift-and-shift" of existing identifications plus a migration window after which `legacy_account_id` is truly retired.
- Downstream models are implemented as dbt, with SCD2 behavior in `dim_customer` and standard joins in `fact_subscriptions`; I assume no runtime feature-flagging or compatibility shim exists.

Step-by-step reasoning

1) Identify direct column references

- Confirmed `column_refs` in `staging.stg_customers` and `intermediate.int_customer_identity` from the "direct_dependents".
- Confirmed usage patterns in downstream marts from the "downstream_marts" list (natural key in SCD2 merge, join keys, export keys, feature entity id).

2) Classify risk by dependency criticality

- High risk: `analytics.dim_customer` (SCD2 natural key) and `analytics.fact_subscriptions` (fact join to dim). Breakage here causes referential integrity failure and transaction misattribution.
- Medium risk: `analytics.customer_360_snapshot` (external key to Salesforce) and `analytics.churn_training_features` (entity ID in feature store). These cause external sync issues and feature drift but do not immediately corrupt conformed dimensions.

3) Assess migration completeness from provided metrics

- `crm_account_uuid_populated_pct` = 94.2% implies ~5.8% remain in legacy-only state (1,204 accounts).
- Until 100% population and until staging/int layer are rewritten to rely solely on `crm_account_uuid`, `legacy_account_id` must remain available.

4) Infer migration sequencing from dependency graph

- Upstream models (staging, intermediate) must be updated before downstream marts because downstream models consume their outputs.
- Therefore: staging -> intermediate -> dim -> facts -> snapshot/feature store, with explicit backfills for legacy accounts before each downstream cutover.

5) Anticipate operational pitfalls

- SCD2 merge on a natural key that disappears mid-process can break type-2 row creation and produce duplicate or orphaned rows.
- Fact joins will fail or silently redirect if surrogate keys are remapped without a compatibility layer.
- Salesforce export and feature store are external systems; misalignment will surface as sync errors hours/days after schema change.

6) Decide rollback implications

- Once DROP is executed, you cannot reconstruct pre-drop state without backups because:
 - The column no longer exists on raw.customers and downstream models may have altered schemas.
 - Re-running historical pipelines will produce different keys or fail, making backfill non-trivial.
- Therefore rollback must be prioritized as a controlled, sequenced migration (reverse order: restore schema, backfill, verify) rather than an emergency revert.

7) Deadline semantics

- With deadline = 2026-07-12 (CRM cutover), I assume cutover is the trigger to begin removal. If cutover happens earlier and migration is incomplete, the plan must pause cutover until migration succeeds.

Minimal checklist I expected before proceeding (not explicitly requested but implicitly required)

- 100% mapping of legacy_account_id -> crm_account_uuid (including edge cases like duplicates or nulls)

Engineering counterpart
2:43 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

2:43 AM - Responded in 4s

Before shipping the DROP to production, verify the following (grouped by priority):

1) Migration completeness and correctness

- 100% of legacy_account_id values have a valid crm_account_uuid mapping (no NULLs or duplicates).
- Mapping table is stable, idempotent, and has referential integrity into raw.customers.
- Backfill for the 1,204 legacy-only accounts is complete and reconciled against source systems.

2) Upstream model readiness

- staging.stg_customers updated: new crm_account_uuid column is populated, tests pass, and no SELECT references remain.
- intermediate.int_customer_identity updated: logic uses crm_account_uuid as primary identity, with fallback/handling for any residual legacy accounts during cutover window.
- Unit tests and schema tests explicitly validate the new columns and no longer expect legacy_account_id.

3) Downstream model readiness

- analytics.dim_customer:
 - SCD2 natural key logic switched to crm_account_uuid; surrogate key continuity verified.
 - Backfill/merge run for legacy accounts to ensure no orphaned rows or duplicate natural keys.
 - Type-2 attribute patterns verified post-migration (no unintended row expiration).
- analytics.fct_subscriptions:
 - Join validation to dim_customer on crm_account_uuid returns identical or expected row counts.
 - Revenue/monetization metrics reconciled (no drop due to broken joins).
- analytics.customer_360_snapshot:
 - Salesforce external key mapping tested end-to-end (successful upserts/lookups).
- analytics.churn_training_features:
 - Entity ID remapped and features backfilled; model training/inference pipelines pass checks.

4) Data quality and reconciliation

- Row counts and key cardinality reconciliations between old and new identifiers.
- Null checks, duplicates checks, and referential integrity checks across dim/fact.
- Snapshot diffs for key metrics pre- and post-migration (revenue, subscriptions, churn features).

5) Operational readiness

- Runbook with rollback procedure (pre-drop backup, restore steps, sequence) reviewed and dry-run.
- Monitoring and alerting in place for join failures, missing customers, Salesforce sync errors, and feature store staleness.
- Deployment window coordinated with CRM cutover; feature flags or compatibility shims considered for gradual rollout.
- Access controls and permissions verified for any new columns/tables introduced.

6) Testing and sign-offs

- End-to-end integration test covering full pipeline from staging to Salesforce export and feature store.
- Business sign-off from CRM/BI/Machine Learning owners confirming no regression in key reports or models.
- Performance testing on large joins and SCD2 cycles to confirm no query plan regressions.

If these checks are satisfied, schedule the DROP after the cutover confirmation and maintain a short observation window with heightened monitoring before closing the change.

Engineering counterpart
2:43 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 5s

Top risks

- Incomplete migration of the 1,204 legacy accounts: if `crm_account_uuid` is missing or duplicated for any account, joins in `fact_subscriptions` and `dim_customer` will break, causing failed queries and lost transactions.
- SCD2 natural-key corruption in `dim_customer`: removing `legacy_account_id` before the SCD2 merge logic is fully switched can orphan type-2 rows, duplicate natural keys, or stall type-2 chaining.
- Referential integrity failure: downstream models and facts that still reference `legacy_account_id` will fail joins, producing missing rows or wrong customers in subscriptions and churn features.
- External sync breakage: Salesforce snapshot and any downstream systems consuming `customer_360_snapshot` will lose key mappings, causing upsert failures and data divergence.
- Partial or failed backfill: if the backfill of legacy accounts is non-idempotent or encounters hidden duplicates/natural-key collisions, reconciliation will be inconsistent and hard to repair.
- Cutover-timing mismatch: if the drop occurs before all consumers are updated or before monitoring/alerts are tuned, incidents may go unnoticed until business impact is felt.

Rollback plan (reverse-cutover, prioritized)

- Pre-requisites (must be in place before DROP):
 - Full logical/physical backup of `raw.customers`, `staging.stg_customers`, `intermediate.int_customer_identity`, `analytics.dim_customer`, `analytics.fact_subscriptions`, `analytics.customer_360_snapshot`, and `analytics.churn_training_features`.
 - Mapping table (`legacy_account_id` -> `crm_account_uuid`) frozen and validated for all 1,204 legacy accounts.
 - Documented runbook with step-by-step restore commands and a rollback owner.
- Rollback sequence (reverse of migration order):
 - 1) Pause any automated jobs/airflow runs that touch affected models.
 - 2) Restore `analytics.churn_training_features` and `analytics.customer_360_snapshot` from backup and reapply any post-backfill changes if necessary (accept potential data loss window).
 - 3) Restore `analytics.fact_subscriptions` and `analytics.dim_customer` to pre-drop schema (re-add `legacy_account_id` as nullable column) and reload slices/partitions using backup.
 - 4) Restore `intermediate.int_customer_identity` and `staging.stg_customers` to include `legacy_account_id`; ensure mapping table is used to re-populate any derived fields.
 - 5) Re-add the column to `raw.customers` only if source schema still expects it; otherwise restore the entire raw layer from backup if needed.
 - 6) Re-run identity resolution and backfills using the mapping table to recover any missed associations.
 - 7) Replay pipelines for affected models from the point of restore to bring them back to current state.
 - 8) Validate row counts, key cardinality, and critical metrics (subscriptions revenue, churn features, Salesforce sync) against pre-drop snapshots.
- Validation after rollback:
 - Confirm referential integrity: no broken joins and no new duplicates.
 - Confirm SCD2 continuity: no unexpected type-2 row explosions or gaps.
 - Confirm external syncs (Salesforce) are healthy and backfill jobs report no errors.
- Forward path once rollback is verified:
 - Re-attempt migration only after root causes are fixed and full mapping completeness is reached; otherwise iterate with staged cutover and feature flags.

If rollback is not feasible (e.g., irreversible DROP with no backup)

- Immediate containment: stop pipeline jobs and revert consumers to use a compatibility view or shim that rederives `legacy_account_id` from `crm_account_uuid` or a surrogate.
- Emergency restore from the latest consistent backup and replay pipelines to recover identifications.
- Incident review and postmortem to define prevention (backups, pre

Engineering counterpart
2:44 AM - Responded in 0s

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

Agent response

2:44 AM - Responded in 5s

Concrete next steps with owners and timing (assumes drop will occur after the deadline 2026-07-12)

1) Final migration validation (7 days)

- Owner: Data Migration Lead

- Actions:

- Confirm 100% mapping: legacy_account_id -> crm_account_uuid for all 1,204 legacy accounts.
- Run end-to-end pipeline dry-run in staging and validate no NULL/duplicates in mapping.
- Reconcile row counts and key cardinality across staging -> intermediate -> dims/facts.

- Deliverable: Validation report and sign-off from Migration Lead + Data Quality Owner.

2) Backfill & backfill rehearsal (3-5 days)

- Owner: Data Engineer (model: intermediate.int_customer_identity)

- Actions:

- Execute deterministic backfill for legacy accounts using mapping table.
- Ensure idempotency and write reconciliation job to detect collisions.

- Deliverable: Backfill job with success metrics and reconciliation diff < 0.01%.

3) Update staging & intermediate (2 days)

- Owner: Staging & Intermediate Owners (staging.stg_customers; intermediate.int_customer_identity)

- Actions:

- Add crm_account_uuid as primary identity; remove legacy_account_id from business logic/tests.
- Add automated tests for new columns and edge cases.

- Deliverable: Merged PRs and CI passing for staging & intermediate models.

4) Downstream model readiness (4-6 days; parallel where possible)

- Owner: Analytics Engineers (per model)

- Actions:

- analytics.dim_customer: switch SCD2 natural key to crm_account_uuid; backfill legacy accounts; verify SCD2 chaining.
- analytics.fct_subscriptions: validate joins to dim on crm_account_uuid; run revenue reconciliation.
- analytics.customer_360_snapshot: test Salesforce key mapping and upsert flow.
- analytics.churn_training_features: remap entity IDs and backfill features; run training/validation checks.

- Deliverable: Per-model readiness checklist and data diffs vs pre-migration baseline.

5) Data quality & reconciliation (2 days)

- Owner: Data Quality Owner

- Actions:

- Key cardinality and uniqueness checks.
- Null checks, duplicate mapping checks, and referential integrity tests.
- Metric snapshot comparison (subscriptions, revenue, churn features, Salesforce syncs).

- Deliverable: Data quality sign-off.

6) Runbook, monitoring & dry-run rollback (3 days)

- Owner: Reliability/DBA Owner

- Actions:

- Finalize and dry-run rollback plan (restore from backups, re-add legacy_account_id if needed, replay pipelines).
- Implement monitoring/alerts for join failures, missing customers, Salesforce errors, feature staleness.

- Deliverable: Runbook, alert rules, and dry-run rollback report.

7) Staged rollout & cutover (1-2 days)

- Owner: Release Manager

- Actions:

- Feature flag or compatibility shim for gradual rollout.
- Pause Airflow/CI jobs, apply DROP to raw.customers after final backup.

- Run post-cutover smoke checks and reconcile metrics within 1 hour.
- Deliverable: Cutover completion and smoke check sign-off.

8) Post-cutover observation (48 hours)

- Owner: On-call Data + Reliability rotation
- Actions:
 - Monitor key dashboards, Salesforce sync, and job logs.
 - Triage any anomalies with hotfix path (restore segment from backup if required).
- Deliverable: Observation report and Go/No-Go

Engineering counterpart

2:44 AM - Responded in 0s

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

Agent response

2:44 AM - Responded in 1s

The one thing I'd want a second pair of eyes on: the 1:1 mapping table (legacy_account_id -> crm_account_uuid) for the 1,204 legacy accounts-specifically its completeness, uniqueness, and joinability back to raw.customers before we freeze legacy_account_id.

Engineering counterpart

2:44 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.