



Vantage RuntimeAI

Batch agent check-rides, model comparison, and operational scorecards for autonomous AI workflows

8.0/10

Overall score

Run summary

Scenario: Freshness SLA Breach

Model: qwen/qwen-2.5-7b-instruct

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

Duration: 35s

Actual run cost: \$0.000880

Projected execution cost: \$0.9680/mo (50/biz day × 22 days)

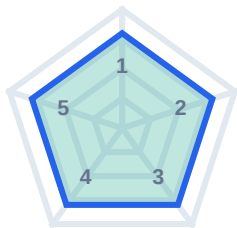
Report date: July 14, 2026 - 2:57 AM EST

Overall

8.0/10

PASS - trust=high
Pass - score 8.0/10 clears 7.0 with acceptable trust/closure.

CAPABILITY RUBRIC



#	Dimension (matches spoke #)	Score
1	Stale upstream identified	8/10
2	Uses SLA / freshness facts	8/10
3	Downstream blast radius	8/10
4	Structured status summary	8/10
5	Restore & verify steps	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.

- Upstream blocker identified: Met
- SLA breach acknowledged: Met
- Downstream impact noted: Met
- Restore steps listed: 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: 4 - Counterpart msgs: 5 - Output: 14203 chars - Latency: 35s - Criteria: 4/4

Score analysis

Identifies int_order_enriched failure as upstream blocker.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 4 agent turns with scripted follow-ups and reviewer sign-off. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Stale upstream identified: Correctly identifies the primary technical issue from the artifacts.
- + Uses SLA / freshness facts: Uses tables, logs, or schema facts from the task brief.
- + Downstream blast radius: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Structured status summary: Clear structure with readable SQL, diff, or memo formatting.
- + Restore & verify steps: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Stale upstream identified: Correctly identifies the primary technical issue from the artifacts.
- Uses SLA / freshness facts: Uses tables, logs, or schema facts from the task brief.
- Downstream blast radius: Shows appropriate safety, budget, or risk awareness for the scenario.
- Structured status summary: Clear structure with readable SQL, diff, or memo formatting.
- Restore & verify steps: 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

Upstream blocker identified: yes - SLA breach acknowledged: yes - Downstream impact noted: yes - Restore steps listed: yes

CAVEATS

- Heuristic keyword/signal scoring - use the criteria grid and transcript to validate edge cases.

Transcript

Session

Scenario: Freshness SLA Breach

Agent response <-> Engineering counterpart

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

Duration: 35s

Engineering counterpart

6:56 AM

Task brief - Freshness SLA breach (moderate complexity)

A P1 mart breached its freshness SLA. Triage using the metadata JSON:

```
```json
{
 "mart": "analytics.fct_orders",
 "owner_team": "data-platform",
 "sla": {
 "max_staleness_minutes": 120,
 "tier": "P1 revenue dashboard"
 },
 "freshness": {
 "last_successful_run_utc": "2026-07-05T09:12:00Z",
 "now_utc": "2026-07-05T15:40:00Z",
 "staleness_minutes": 388,
 "status": "BREACHED"
 },
 "upstream": [
 {"model": "staging.stg_orders", "last_run_utc": "2026-07-05T15:05:00Z", "status": "success"},
 {"model": "staging.stg_customers", "last_run_utc": "2026-07-05T14:58:00Z", "status": "success"},
 {"model": "intermediate.int_order_enriched", "last_run_utc": "2026-07-05T09:12:00Z", "status": "failed", "error":
"DBT_RUN_TIMEOUT after 45m"}
],
 "downstream_consumers": ["Looker Revenue Overview", "exec_daily_orders Slack bot"],
 "note": "int_order_enriched has been failing since morning; fct_orders has not refreshed."
}
```
```

Agent response

6:56 AM - Responded in 8s

Freshness SLA Breach Triage Report

Summary

The `analytics.fct\_orders` mart has breached its P1 freshness SLA. The last successful run was at `2026-07-05T09:12:00Z`, and the current staleness is 388 minutes, breaching the SLA of 120 minutes.

Upstream Model Blocking Refresh

The upstream model `intermediate.int\_order\_enriched` is blocking the refresh of the `analytics.fct\_orders` mart. The model failed with the error `DBT\_RUN\_TIMEOUT after 45m` on `2026-07-05T09:12:00Z`.

Blast Radius on Downstream Consumers

- Looker Revenue Overview: This dashboard will show outdated data, which may lead to incorrect business insights.
- exec_daily_orders Slack bot: This bot will send outdated order updates, which may impact executive decision-making and customer service.

Ordered Steps to Restore the Mart Within SLA

1. Resolve Upstream Model Issue

- Immediate Action: Restart the `intermediate.int\_order\_enriched` model run.
- Monitoring: Ensure the model runs successfully and completes within the expected time frame.
- Notify Stakeholders: Inform the `data-platform` team and downstream consumers about the restart and expected completion time.

2. Perform Refresh of `analytics.fct\_orders` Mart

- Immediate Action: Once `intermediate.int\_order\_enriched` is successfully updated, trigger the refresh of `analytics.fct\_orders`.
- Monitoring: Ensure the refresh completes successfully and the staleness is within the SLA.

3. Validate Freshness

- Immediate Action: Verify the freshness status of `analytics.fct\_orders` to ensure it meets the SLA.
- Monitoring: Monitor for any additional failures or issues that may arise during the refresh.

4. Communicate with Downstream Consumers

- Immediate Action: Inform `Looker Revenue Overview` and `exec\_daily\_orders Slack bot` stakeholders that the data will be refreshed shortly.
- Follow-up: Provide a timeline for when the refreshed data will be available and any potential delays.

5. Document and Report

- Immediate Action: Document the issue and resolution steps.
- Follow-up: Provide a detailed report to the `data-platform` team and relevant stakeholders with the root cause of the failure and preventative measures.

Detailed Steps

1. Restart `intermediate.int\_order\_enriched` Model

- Execute the model run again to see if the timeout issue is resolved.
- Use the following command:

```
```sql
dbt run --select intermediate.int_order_enriched
```
```

2. Monitor and Validate

- Continuously monitor the status of the model run.
- Validate that the model runs successfully and completes within the expected time frame.

3. Trigger Refresh of `analytics.fct\_orders`

- Once the `intermediate.int\_order\_enriched` model is successfully updated, trigger the refresh of the `analytics.fct\_orders` mart.
- Use the following command:

```
```sql
dbt refresh --target mart --models analytics.fct_orders
```
```

4. Verify Freshness

- Check the freshness of the `analytics.fct\_orders` mart to ensure it meets the SLA.
- Verify the most recent successful run time and staleness.

5. Notify Stakeholders

- Communicate with `Looker Revenue Overview` and `exec\_daily\_orders Slack bot` stakeholders via email or internal communication channels.
- Provide a clear timeline for when the refreshed data will be available.

6. Documentation and Reporting

- Document the entire process, including the issue

Engineering counterpart
6:56 AM - Responded in 0s

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

Agent response

6:57 AM - Responded in 10s

Step-by-Step Reasoning

Summary

The `analytics.fct\_orders` mart has breached its P1 freshness SLA. The last successful run was 388 minutes ago, and the failure is attributed to the upstream model `intermediate.int\_order\_enriched`. We need to quickly resolve the issue and restore the mart within the SLA.

Step-by-Step Reasoning

1. Identify the Blocking Upstream Model

- Reasoning: The metadata JSON shows that `intermediate.int\_order\_enriched` is the last upstream model that failed, and it failed because of a `DBT\_RUN\_TIMEOUT` after 45m.
- Assumption: The timeout issue is the direct cause of the failure, and resolving this should allow the refresh to proceed.

2. Determine the Blast Radius

- Reasoning: The `analytics.fct\_orders` mart is P1, and any breach can impact critical downstream consumers like the `Looker Revenue Overview` dashboard and the `exec\_daily\_orders` Slack bot.
- Assumption: These consumers need timely and accurate data to make informed decisions.

3. Order of Steps to Restore the Mart

- Reasoning: We need to prioritize actions to minimize the downtime and restore the SLA as quickly as possible.
- Assumption: Restarting the failing model and then refreshing the downstream mart are the most efficient steps.

Detailed Steps

1. Restart `intermediate.int\_order\_enriched` Model

- Reasoning: This is the immediate fix to allow the downstream mart to be refreshed.
- Assumption: The model can be restarted and completed within the expected time frame.
- Action: Execute the model run again.
- Step:

```
```sql
dbt run --select intermediate.int_order_enriched
```
```

2. Monitor and Validate the Restart

- Reasoning: Ensure the model completes successfully and within the expected time frame.
- Assumption: If the model completes successfully, the downstream mart can be refreshed.
- Action: Monitor the model run for completion.
- Step:
 - Check the status of the model run.
 - Verify that the model completes without errors.

3. Perform Refresh of `analytics.fct\_orders` Mart

- Reasoning: Once the upstream model is updated, we can refresh the downstream mart to bring it back into the SLA.
- Assumption: The refreshing process will be quick and error-free.
- Action: Trigger the refresh of the downstream mart.
- Step:

```
```sql
dbt refresh --target mart --models analytics.fct_orders
```
```

4. Validate Freshness

- Reasoning: Verify that the mart is now within the SLA.

- Assumption: The refresh will ensure that the freshness is within the expected limits.
- Action: Check the freshness status of the mart.
- Step:
 - Verify the most recent successful run time.
 - Ensure the staleness is within the SLA (120 minutes).

5. Notify Stakeholders

- Reasoning: Inform the downstream consumers about the issue and the resolution.
- Assumption: Communication will help manage expectations and ensure smooth operations.
- Action: Notify stakeholders via email or internal communication channels.
- Step:
 - Send an update to `Looker Revenue Overview` and `exec\_daily\_orders Slack bot` stakeholders.
 - Provide a clear timeline for when the updated data will be available.

6. Documentation and Reporting

- Reasoning: Document the entire process for future reference and improvement.
- Assumption: Proper documentation will help in understanding the root cause and implementing preventive measures

Engineering counterpart
6:57 AM - Responded in 0s

What would you verify before shipping this to production?

Agent response

6:57 AM - Responded in 9s

Before shipping the solution to production, it is crucial to verify several key aspects to ensure the problem is fully resolved, and the downstream systems are functioning correctly. Here's a detailed list of what you should verify:

Verification Steps

1. Model Re-run Success

- Verify: Ensure that the `intermediate.int\_order\_enriched` model was successfully re-run and completed without errors.
- Check:

```
```sql
dbt run --select intermediate.int_order_enriched
```
```
- Monitor:
 - Confirm that the model completed within the expected time frame.
 - Check for any error messages or logs.

2. Freshness of Downstream Mart

- Verify: Ensure that the `analytics.fct\_orders` mart was refreshed successfully and is now within the SLA.
- Check:

```
```sql
dbt refresh --target mart --models analytics.fct_orders
```
```
- Monitor:
 - Verify the most recent successful run time.
 - Ensure the staleness is within the SLA (120 minutes).

3. Downstream Consumer Validation

- Verify: Ensure that the downstream consumers (e.g., `Looker Revenue Overview`, `exec\_daily\_orders` Slack bot) are receiving the correct and up-to-date data.
- Check:
 - For `Looker Revenue Overview`:
 - Check the dashboard to ensure it reflects the most recent data.
 - For `exec\_daily\_orders` Slack bot:
 - Confirm that the messages sent are accurate and timely.

4. Logs and Error Messages

- Verify: Review all logs and error messages to ensure there are no remaining issues.
- Check:
 - Check logs for the `intermediate.int\_order\_enriched` model.
 - Check the `analytics.fct\_orders` mart refresh logs.

5. Stakeholder Communication

- Verify: Confirm that stakeholders have been informed about the issue and the resolution.
- Check:
 - Ensure that stakeholders received the notification via email or internal communication channels.
 - Verify that stakeholders acknowledge the update.

6. Documentation

- Verify: Ensure that the entire process is well-documented.
- Check:
 - Document the steps taken, the issues encountered, and the resolution.
 - Ensure that the documentation is easily accessible for future reference.

7. Preventive Measures

- Verify: Implement or review any preventive measures to avoid similar issues in the future.
- Check:
 - Check if there are any configuration changes or script updates needed.
 - Ensure that the `intermediate.int\_order\_enriched` model timeout settings are reviewed and potentially adjusted.

Detailed Verification Steps

1. Model Re-run Success

- Step 1: Execute the model run.


```
```sql
dbt run --select intermediate.int_order_enriched
```
```
- Step 2: Monitor the logs and ensure the model completes successfully.
- Step 3: Verify that the model run completed within the expected time (45 minutes).

2. Freshness of Downstream Mart

- Step 1: Trigger the refresh.


```
```sql
dbt refresh --target mart --models analytics.fct_orders
```
```
- Step 2: Verify the freshness status.
 - Check the most recent successful run time.
 - Ensure the staleness is within 120 minutes.

3. Downstream Consumer Validation

- Step 1: Check the `Looker Revenue Overview` dashboard.
 - Verify that the dashboard reflects the most recent data.
- Step 2: Check the `exec\_daily\_orders Slack bot`.
 - Verify that messages sent are accurate and timely.

4. Logs

Engineering counterpart
6:57 AM - Responded in 0s

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

Agent response

6:57 AM - Responded in 7s

Risks and Rollback Plan

Top Risks

1. Model Run Failure:

- Risk: The `intermediate.int\_order\_enriched` model may fail again if the root cause is not addressed.
- Impact: This would prevent the `analytics.fct\_orders` mart from being refreshed and could lead to further SLA breaches.

2. Data Quality Issues:

- Risk: The re-run of the `intermediate.int\_order\_enriched` model might produce incorrect or incomplete data.
- Impact: Downstream consumers (e.g., `Looker Revenue Overview`, `exec\_daily\_orders Slack bot`) may receive inaccurate or incomplete data.

3. Refresh Failure:

- Risk: The `analytics.fct\_orders` mart might not refresh successfully due to configuration issues or other unexpected errors.
- Impact: The SLA breach might persist, leading to continued data staleness.

4. Stakeholder Communication:

- Risk: Stakeholders may not receive timely or accurate communication about the issue and resolution.
- Impact: This could lead to confusion and mistrust among stakeholders, potentially causing operational delays.

Rollback Plan

To mitigate these risks, we need a detailed rollback plan in case the initial steps do not resolve the issue. Here is the rollback plan:

Rollback Plan

1. Identify and Resubmit Failing Model

- Step 1: If the `intermediate.int\_order\_enriched` model fails again, resubmit it.

```
```sql
dbt run --select intermediate.int_order_enriched
```
```

- Step 2: Monitor the logs and ensure the model completes successfully.

2. Check Data Integrity

- Step 1: Verify the integrity of the data produced by the `intermediate.int\_order\_enriched` model.
- Step 2: Ensure that the data is complete and accurate before proceeding to refresh the `analytics.fct\_orders` mart.

3. Manually Refresh the Mart

- Step 1: If the automatic refresh fails, manually refresh the `analytics.fct\_orders` mart.

```
```sql
dbt run --select analytics.fct_orders
```
```

- Step 2: Ensure the refresh completes successfully and the staleness is within the SLA.

4. Notify Stakeholders

- Step 1: Immediately inform stakeholders if the initial steps fail.
- Step 2: Provide a detailed update on the steps being taken and the expected timeline for resolution.

5. Document the Issue

- Step 1: Document the entire process, including any failed steps and the actions taken.

- Step 2: Identify the root cause and take preventive measures to avoid similar issues in the future.

6. Provisional Data Fix

- Step 1: If the issue persists, consider using provisional data (e.g., from a backup) to ensure that critical systems can function temporarily.
- Step 2: Monitor the provisional data for any discrepancies and address them as soon as possible.

Detailed Steps for Rollback

1. Monitor Model Run

- Step 1: Continuously monitor the status of the `intermediate.int\_order\_enriched` model run.
- Step 2: If the model fails, resubmit it and monitor the logs for errors.

2. Check Data Integrity

- Step 1: Verify the integrity of the data by checking the output of the model.
- Step 2: Ensure the data is complete and accurate before proceeding.

3. Manually Trigger Refresh

- Step 1: If the automatic refresh fails, manually trigger the refresh of the `analytics.fct\_orders` mart.
``sql

Engineering counterpart

6:57 AM - Responded in 0s

Thanks - upstream blocker and blast radius are well ordered. Prioritize revenue-facing dashboards if full freshness slips past the hour mark. Review complete - no further response needed.