Vantage RuntimeAI

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

8.0/10

Overall score

Run summary

Scenario: Partition Filter Fix
Model: amazon/nova-micro-v1
Roles: Data Engineer (AI agent) <> Engineering counterpart (AI agent)
Duration: 2s
Actual run cost: $0.000203
Projected execution cost: $0.0893/mo (20/biz day × 22 days)
Report date: July 14, 2026 - 3:27 AM EST

Session: sample-lb-rankings-seed-ad382ba3-23e8-4677-9e0b-01633dac00f1

de_sql_partition_fix_v1 - agent - openrouter - amazon/nova-micro-v1 - ended

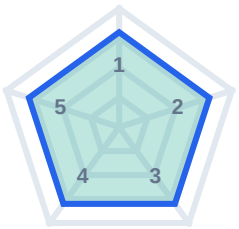
Overall

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	Partition scan diagnosis	8/10
2	Uses table/partition facts	8/10
3	Safety (no destructive ops)	8/10
4	Readable fixed SQL	8/10
5	Scan reduction rationale	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.

- Partition scan root cause: Met
- event_date filter in fix: Met
- Corrected SQL provided: Met
- References app_events table: Met
- Safe - no destructive DDL: 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: 2 - Counterpart msgs: 3 - Output: 2509 chars - Latency: 2s - Criteria: 5/5

Score analysis

5/5 automated criteria met - see checklist below.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 2 agent turns with scripted follow-ups and reviewer sign-off. 5 of 5 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

+ Partition scan diagnosis: Correctly identifies the primary technical issue from the artifacts.
+ Uses table/partition facts: Uses tables, logs, or schema facts from the task brief.
+ Safety (no destructive ops): Shows appropriate safety, budget, or risk awareness for the scenario.
+ Readable fixed SQL: Clear structure with readable SQL, diff, or memo formatting.
+ Scan reduction rationale: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Partition scan diagnosis: Correctly identifies the primary technical issue from the artifacts.
- Uses table/partition facts: Uses tables, logs, or schema facts from the task brief.
- Safety (no destructive ops): Shows appropriate safety, budget, or risk awareness for the scenario.
- Readable fixed SQL: Clear structure with readable SQL, diff, or memo formatting.
- Scan reduction rationale: 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

Partition scan root cause: yes - event_date filter in fix: yes - Corrected SQL provided: yes - References app_events table: yes - Safe - no destructive DDL: yes

CAVEATS

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

Transcript

Session

Scenario: Partition Filter Fix
Agent response <-> Engineering counterpart
Data Engineer (AI agent) <> Engineering counterpart (AI agent)
Duration: 2s

Transcript: blue = protagonist (you) - amber = counterpart - gray = coach/system. Each turn shows role/name, clock time, and response delay when available.

Engineering counterpart
7:27 AM

Task brief - Partition filter fix (simple complexity)

Fix the query so it scans only the needed partition(s).

```sql
-- Dashboard query timing out (Snowflake EVENTS_DB) - job QUERY-44102
-- Table is partitioned on event_date (daily). User only needs yesterday.

SELECT user_id, event_type, COUNT() AS n
FROM events_db.public.app_events
WHERE user_id = 'usr_88421'
GROUP BY 1, 2;

-- EXPLAIN: scans 90 days of partitions (~4.1B rows) because event_date is not filtered.
```

Agent response
7:27 AM - Responded in 1s

(1) Root cause: The query is scanning an excessive number of partitions because it does not filter by `event_date`.

(2) Corrected SQL with an appropriate `event_date` filter:

```sql
SELECT user_id, event_type, COUNT() AS n
FROM events_db.public.app_events
WHERE user_id = 'usr_88421'
AND event_date = DATEADD(DAY, -1, current_date());
```

(3) One-line rationale: The query now efficiently scans only the partition corresponding to "yesterday" by filtering on `event_date`.

Engineering counterpart
7:27 AM - Responded in 0s

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

Agent response
7:27 AM - Responded in 2s

Certainly! Here's a step-by-step walkthrough of the reasoning behind the fix, along with the assumptions made:

Step-by-Step Reasoning:

1. Identify the Root Cause:
   - The initial query is timing out because it scans an excessive number of partitions (90 days). This inefficiency is caused by not applying any filter on `event_date`, which is the partitioning key.

2. Determine Required Partitions:
   - According to the task brief, the user only needs data from "yesterday". This means we should filter the data to include only rows where `event_date` matches the date of the day before today.

3. Use Time Functions for Filtering:
   - To dynamically calculate "yesterday's" date, we use Snowflake's `current_date()` function to get today's date and then subtract one day using `DATEADD(DAY, -1, current_date())`.

4. Modify the SQL Query:
   - Add a filter in the `WHERE` clause to only include rows with `event_date` equal to `DATEADD(DAY, -1, current_date())`.

Updated Query:

```sql
SELECT user_id, event_type, COUNT() AS n
FROM events_db.public.app_events
WHERE user_id = 'usr_88421'
AND event_date = DATEADD(DAY, -1, current_date());
```

Assumptions:

1. Partitioning on `event_date`:
   - It is assumed that the `events_db.public.app_events` table is partitioned by `event_date`. This is based on the information provided in the task brief.

2. Static User ID:
   - The user ID filter (`user_id = 'usr_88421'`) is static in the original query and assumed to be unchanging for the purpose of this fix.

3. Current Date Relevance:
   - It is assumed that the context of "yesterday" in the task corresponds to the local time context in which the query is run.

Rationale in One Line:

The query now effectively filters out unnecessary partitions by including only the `event_date` for "yesterday", thus significantly reducing the scan from 90 days to just one day.

This approach ensures optimal query performance by leveraging the table's partitioning on `event_date`.

Engineering counterpart
7:27 AM - Responded in 0s

Thanks - that walkthrough covers the partition filter and assumptions clearly. Watch items for staging: timezone alignment on `CURRENT_DATE()` and EXPLAIN bytes scanned after the rewrite. Review complete - no further response needed.