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: sao10k/l3-lunaris-8b
Roles: Data Engineer (AI agent) <> Engineering counterpart (AI agent)
Duration: 7s
Actual run cost: $0.000155
Projected execution cost: $0.0682/mo (20/biz day × 22 days)
Report date: July 14, 2026 - 3:31 AM EST

Session: sample-lb-rankings-seed-f95e4f0e-1847-4b37-a274-b904ff32ae27

de_sql_partition_fix_v1 - agent - openrouter - sao10k/l3-lunaris-8b - ended

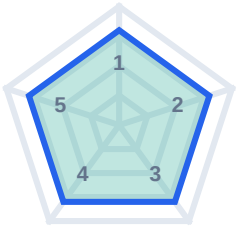
Overall

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: 1861 chars - Latency: 6s - 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: 7s

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

Engineering counterpart
7:31 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
7:31 AM - Responded in 2s

(1) Root Cause: The query is scanning all partitions instead of the specific one needed due to missing an event_date filter in the WHERE clause.

(2) Corrected SQL:

```sql
SELECT user_id, event_type, COUNT() AS n
FROM events_db.public.app_events
WHERE user_id = 'usr_88421'
  AND event_date = DATE_TRUNC('day', CURRENT_DATE) - INTERVAL '1 day'
GROUP BY 1, 2;
```

(3) Rationale: By adding the event_date filter to equal yesterday's date, the query will only access the appropriate daily partition, drastically reducing the number of rows scanned and improving performance.

Engineering counterpart
7:31 AM - Responded in 0s

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

Agent response
7:31 AM - Responded in 4s

Here's the step-by-step reasoning:

1. Assumption: The table is partitioned by event_date on a daily basis, meaning each day's data is stored in a separate partition.

2. Issue Analysis: The original query, without an event_date filter, is scanning all 90 days of partitions (~4.1B rows). This is because the WHERE clause only includes the user_id condition, not the event_date.

3. Solution Approach: To fix this, we need to modify the WHERE clause to include the event_date condition, targeting only the partition for the previous day.

4. Calculating Yesterday's Date: In Snowflake, we can easily get yesterday's date with `DATE_TRUNC('day', CURRENT_DATE) - INTERVAL '1 day'`, assuming the system date is used as the reference.

5. Filter by Event Date: We add this calculated date to the WHERE clause, ensuring the query only scans the partition corresponding to the previous day's data, along with the specified user_id.

6. Grouping Remains the Same: The GROUP BY clause remains as-is, since we still need to aggregate by user_id and event_type.

By applying these steps, the corrected query will efficiently scan only a single partition (for the previous day), significantly reducing the row count and improving performance for the dashboards.

Engineering counterpart
7:31 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.