



Vantage RuntimeAI

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

8.0/10

Overall score

Run summary

Scenario: Null Rate Alert

Model: qwen/qwen3-235b-a22b-2507

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

Duration: 14s

Actual run cost: \$0.000867

Projected execution cost: \$1.43/mo (75/biz day × 22 days)

Report date: July 14, 2026 - 3:08 AM EST

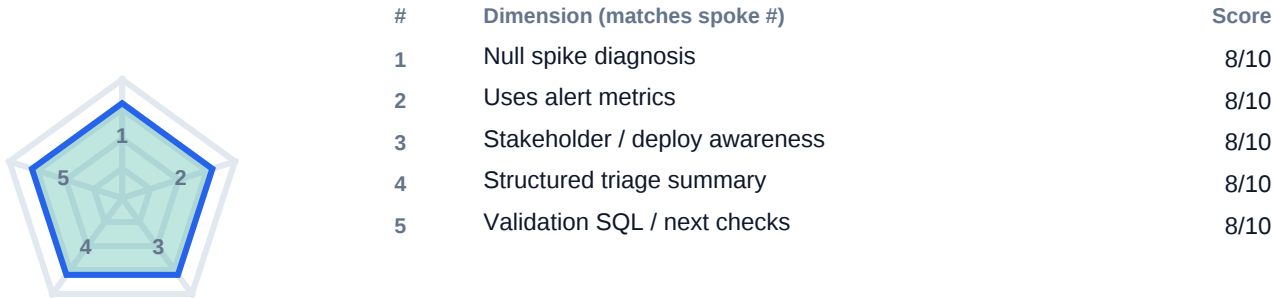
Overall

8.0/10

PASS - trust=high

Pass - score 8.0/10 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.

- Null spike identified: Met
- Validation SQL proposed: Met
- Recent deploy considered: Met
- References alert column/table: 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: 3959 chars - Latency: 14s - Criteria: 4/4

Score analysis

Identifies null-rate spike on shipping_country_code.

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. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- + Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- + Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- + Validation SQL / next checks: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- Validation SQL / next checks: 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

Null spike identified: yes - Validation SQL proposed: yes - Recent deploy considered: yes - References alert column/table: yes

CAVEATS

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

Transcript

Session

Scenario: Null Rate Alert

Agent response <-> Engineering counterpart

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

Duration: 14s

Engineering counterpart

7:08 AM

Task brief - Null rate alert (simple complexity)

A column null rate spiked overnight. Triage the alert JSON and recommend next steps:

```
```json
{
 "table": "analytics.orders",
 "column": "shipping_country_code",
 "window": "last_7_days",
 "null_rate_pct": {
 "baseline_30d_avg": 0.3,
 "yesterday": 17.8,
 "today_so_far": 19.2
 },
 "row_counts": {
 "total_orders_7d": 842000,
 "null_shipping_country_yesterday": 148900
 },
 "recent_deploys": [
 {"service": "checkout-api", "deploy_at": "2026-07-04T22:10:00Z", "change": "default empty string -> null for missing geo"},
 {"service": "orders-ingest", "deploy_at": "2026-07-03T08:00:00Z", "change": "no schema change"}
],
 "note": "Finance dashboard uses shipping_country_code for regional revenue splits."
}
```
```

Agent response

7:08 AM - Responded in 5s

1) Likely cause: The spike in nulls is likely due to the recent `checkout-api` deploy that changed handling of missing geography data-now defaulting to `null` instead of an empty string, which directly affects `shipping\_country\_code`.

2) Validation SQL:

```
```sql
```

```
SELECT
```

```
 COUNT() as total_rows,
```

```
 COUNT(NULLIF(shipping_country_code, '')) as non_empty_count,
```

```
 COUNT() - COUNT(NULLIF(shipping_country_code, '')) as empty_or_null_count,
```

```
 SUM(CASE WHEN shipping_country_code IS NULL THEN 1 ELSE 0 END) as truly_null_count,
```

```
 DATE(event_timestamp) as event_date
```

```
FROM analytics.orders
```

```
WHERE event_timestamp >= '2026-07-04'
```

```
GROUP BY event_date
```

```
ORDER BY event_date;
```

```
```
```

3) Recent deploy suspect? Yes - the `checkout-api` deploy at `2026-07-04T22:10:00Z` introduced the change to emit `null` instead of empty strings, aligning with the timing and nature of the null rate spike.

4) Stakeholder comms (one line):

"Null spike in shipping_country_code is linked to a recent checkout API change that now sends missing country data as null instead of empty; we're assessing impact on finance reports."

Engineering counterpart

7:08 AM - Responded in 0s

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

Agent response

7:08 AM - Responded in 9s

Absolutely - let's walk through the reasoning step by step, highlighting the key observations, inferences, and assumptions made.

? Step 1: Understand the symptom

We're told:

- Column: `shipping\_country\_code` in `analytics.orders`
- Baseline null rate: 0.3% (very low - almost always populated)
- Yesterday: 17.8% null
- Today so far: 19.2% null
- Total orders in last 7 days: 842,000 -> ~120,000/day
- Nulls yesterday: 148,900 -> ~124,000 new nulls vs baseline (~360 expected nulls at 0.3%)

?? Inference: This is a massive spike - from ~360 nulls/day to ~149k. It's not noise. It's systemic and recent.

?? Step 2: Check timing of deploys

Recent deploys:

1. `orders-ingest` - older (2026-07-03), no schema change -> unlikely culprit
2. `checkout-api` - 2026-07-04T22:10:00Z -> just before the spike (assuming "yesterday" starts after that time)

?? Inference: The timing lines up: the spike likely began on July 5th, right after the deploy on July 4th at 10:10 PM UTC. That's strong circumstantial evidence.

?? Step 3: Evaluate the nature of the code change

The `checkout-api` change:

> "default empty string -> null for missing geo"

That means:

- Before: If country was unknown, it sent `""` (empty string)
- After: It now sends `NULL`

?? Inference: This directly explains the increase in nulls - data that previously appeared as non-null (empty string) is now being recorded as `NULL`.

But wait - why does the analytics table now show them as nulls?

That depends on how the ingestion pipeline treats incoming `null` values.

- If the pipeline preserves nulls, then this change would propagate.
- If it coerces nulls to empty strings, it wouldn't.

But we're seeing a spike in nulls, so:

?? Assumption 1: The ingestion system preserves `null` values from the source API and does not convert them to empty strings or defaults.

Therefore: this behavioral change in `checkout-api` likely passed through unchanged.

? Step 4: Consider data types and definitions

- If `shipping\_country\_code` is defined as `STRING`, both `NULL` and `` are valid.
- But monitoring tools typically count only `NULL` as "null" - empty strings are considered "non-null".

So:

- Before deploy: missing country = `` -> counted as non-null -> low null rate
- After deploy: missing country = `NULL` -> counted as null -> spikes null rate

?? This is the smoking gun: The definition of "null" in the monitoring system excludes empty strings. So the semantic meaning of missing data hasn't changed - only how it's encoded.

?? Assumption 2: The null rate metric does not include empty strings - it only counts actual SQL `NULL`s. This is standard.

? Step 5: Validate volume

-

Engineering counterpart**7:08 AM - Responded in 0s**

Thanks - assumptions and validation path are reasonable. Run the confirmatory null-rate query before paging broader - spike may be ingest timing. Review complete - no further response needed.