Explain the report with Sequelize vs Raw query performance

Here's an expert analysis of your JMeter results comparing **Raw SQL** vs **Sequelize** for the GET Trip API, based on the attached `.jtl` file[1]:

1. What Was Tested?

- **Endpoints:**
  - `/api/raw/trip/{id}` (Raw SQL query)
  - `/api/sequelize/trip/{id}` (Sequelize ORM)
- **Operation:**
  - GET trip details by ID
- **Test Tool:**
  - Apache JMeter

2. Key Observations from the Data

Raw - Get Trip

- **Response times (elapsed):**
  - Range: ~9 ms (fastest) to ~258 ms (slowest)
  - Many requests clustered between 10–70 ms, but some spikes up to 200+ ms
- **All requests succeeded** (`responseCode=200`, `success=true`)
- **Latency and elapsed times are closely matched**

Sequelize - Get Trip

- **Response times (elapsed):**
  - Range: ~95 ms (fastest) to ~261 ms (slowest)
  - Most requests between 110–250 ms, with a few as low as 95 ms
- **All requests succeeded** (`responseCode=200`, `success=true`)
- **Latency and elapsed times are closely matched**

3. Performance Comparison Table

Metric	Raw SQL GET	Sequelize GET
Fastest response (ms)	~9	~95
Slowest response (ms)	~258	~261
Most responses (ms)	10–70	110–250
Overall trend	Faster, lower median	Slower, higher median
Errors	0	0

4. Interpretation

- **Raw SQL is consistently faster** for GET operations in this test. The majority of raw queries completed in under 70 ms, with many as low as 10–20 ms.
- **Sequelize GET requests are slower**: Most responses are above 100 ms, and there are no ultra-fast responses like with raw SQL.
- **Both endpoints are stable**: No errors, all requests returned HTTP 200.

Why is Raw SQL Faster?

- **No ORM overhead**: Raw queries go straight to the database and return results, while Sequelize adds abstraction, parsing, and object mapping.
- **Less JavaScript processing**: Sequelize must convert DB rows to JavaScript objects and manage model logic, which adds time.
- **Direct execution**: Raw queries avoid the extra steps of validation, hooks, or associations that an ORM may perform.

5. When Might Sequelize Be Slower?

- **Simple lookups**: ORM overhead is more apparent when the query is simple (like a primary key lookup).
- **Large datasets or complex models**: Sequelize's model parsing and associations can add significant time.

6. When to Use Each Approach?

Use Case	Recommendation
Maximum performance, minimal overhead	Raw SQL
Maintainability, rapid development	Sequelize (ORM)
Complex queries with many joins	Raw SQL (for speed)
Standard CRUD, business logic in code	Sequelize (for safety)

7. Summary

- **Raw SQL GET is significantly faster** than Sequelize GET in your Node.js API under load.
- **Sequelize GET is stable but slower** due to ORM abstraction and processing.
- **Both are reliable** (no errors), but if performance is critical for GET by ID, raw SQL is the clear winner in this test.

If you want a chart or deeper statistical breakdown (mean, median, percentiles) or have questions about optimizing Sequelize performance, let me know!

⁂

1. results_Get-sequelize-vs-raw.jtl