# Observability Template NRQL Queries

**2024 RX Presentation Takeaways**

## Description

The purpose of this document is to provide a series of NRQL statements in a format that allows you to copy-and-paste easily into NewRelic, in order to gain insight about your Observability-Enabled Rock Environment.

The section titles in this document match the slide titles in our 2024 presentation where we originally published them.

### Important notes:

1. In any query where we include `entity.name = 'rock-rms'`, you may be using a different name in your Rock environment. If you are, you will need to replace that value with your own.

2. In any query where we include `SINCE 2 WEEKS AGO`, you can specify a more limited time frame by replacing that line with something like:

   ```
   SINCE '2024-01-01 15:00:00 EDT'
   UNTIL '2024-01-01 19:00:00 EDT'
   ```

3. Several of the queries here include samples of what kind of visualization you can expect in NewRelic. They are only provided as examples of the visualizations. They are NOT intended to represent ideals, goals, or results you should try to match. One factor alone is that your environment may have more spans or fewer spans, based on your church size and the features you use.

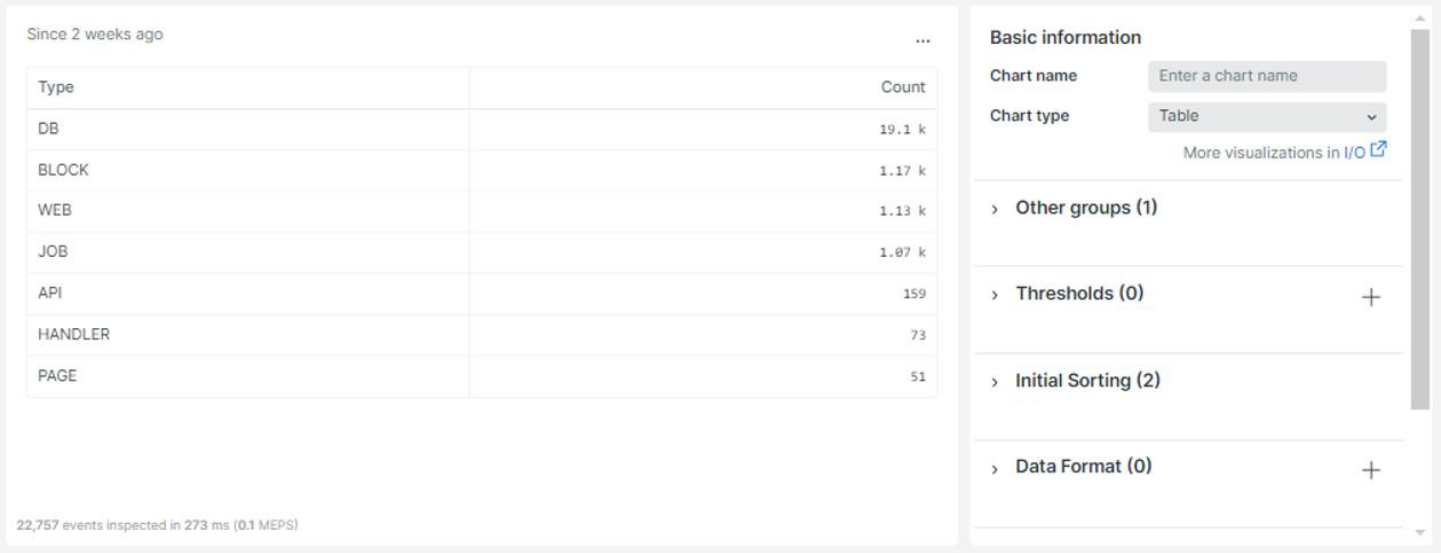
# Query #1: Count of Spans by Type

This is the “baseline” query that we recommend starting with, in order to get a feel for what is happening in your church’s Rock environment.

```
SELECT count(*)
FROM Span
WITH capture(name, r'(?P<Type>[^ :]*).*') AS Type
WHERE instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%'
    AND entity.name = 'rock-rms'
FACET Type
SINCE 2 WEEKS ago
LIMIT MAX
```

**Recommended Visualization:** Table or Bar Chart

**Sample Result:**



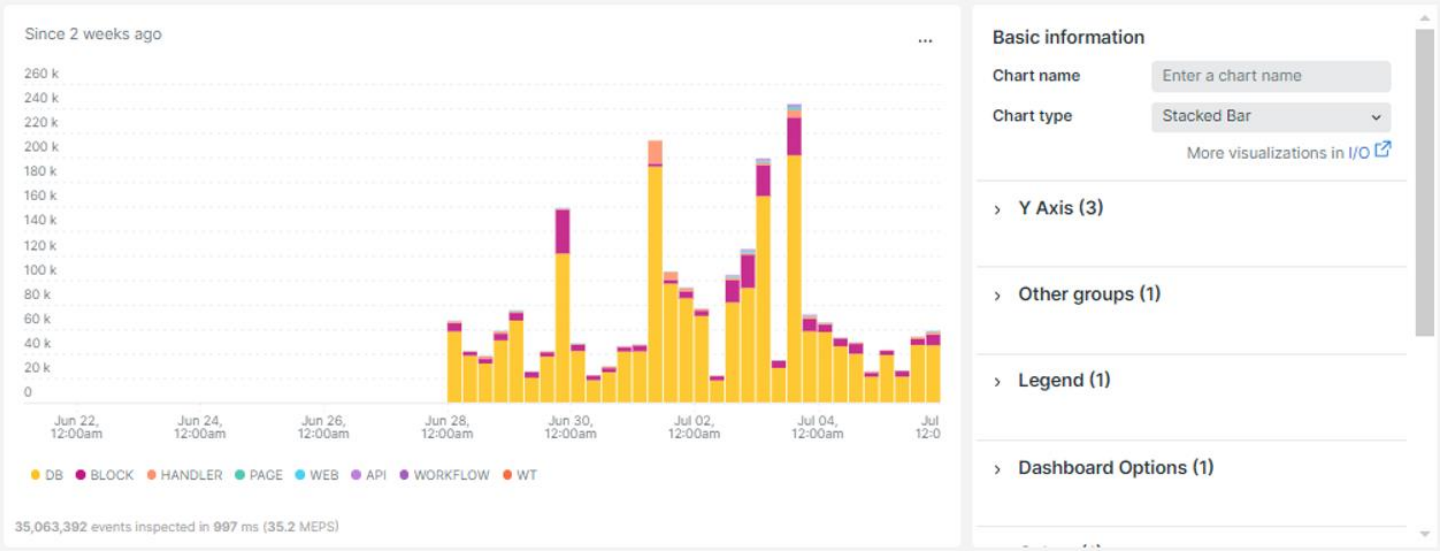
# Query #2: Count of Spans by Type Over Time

Here we've added the "Timeseries" clause to Query #1, so you can see how the counts change throughout the day and from day to day.

```
SELECT count(*)
FROM Span
WITH capture(name, r'(?P<Type>[^ :]*).*') AS Type
WHERE instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%'
    AND entity.name = 'rock-rms'
FACET Type
SINCE 2 WEEKS ago
TIMESERIES 6 HOURS
LIMIT MAX
```

**Recommended Visualization:** Stacked Bar Chart

**Sample Result:**

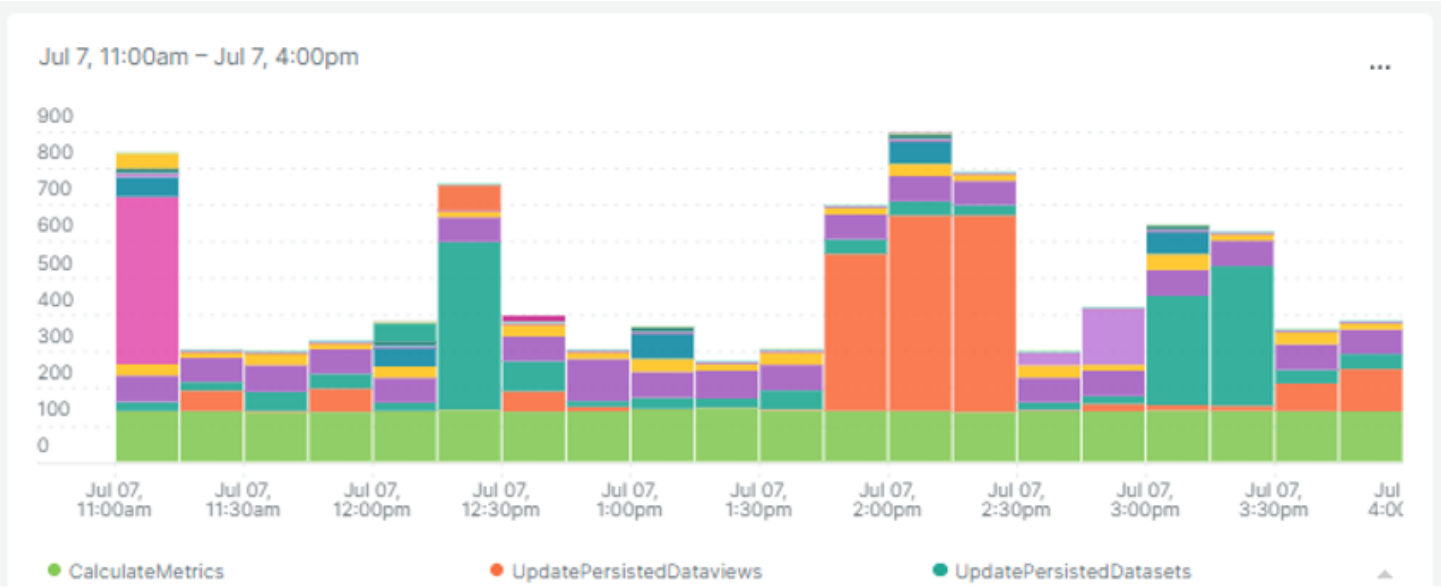


# Query #3: Time for Jobs to Run, by Job Type

```
SELECT sum('rock.job.duration')
FROM Span
WHERE instrumentation.provider = 'opentelemetry'
    AND http.user.agent NOT LIKE '%Bot%'
    AND entity.name = 'rock-rms'
FACET 'rock.job.type'
SINCE 2 WEEKS AGO
TIMESERIES 6 HOURS
LIMIT MAX
```

**Recommended Visualization:** Stacked Bar Chart

**Sample Result:**



## Query #4: Slowest Pages

```
SELECT
    average(duration.ms)
    , count(*)
    , average(rock.db.query_count)
    , max(rock.db.query_count)
    , min(rock.db.query_count)
FROM
    Span
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
    AND entity.name = 'rock-rms' -- This is the "Observability Service Name" defined in
System Settings
    AND name LIKE 'Page%' -- Select only PAGE spans
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table

## Query #5: Page Time Outliers

```
SELECT
    stddev(duration.ms)
FROM
    Span
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
    AND entity.name = 'rock-rms' -- This is the "Observability Service Name" defined in
System Settings
    AND name LIKE 'Page%' -- Select only PAGE spans
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table or Bar Chart

## Query #6: Most Recommended Pages:

```
SELECT
    count(*)
FROM
    Span
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
    AND entity.name = 'rock-rms' -- This is the "Observability Service Name" defined in
System Settings
    AND name LIKE 'Page%' -- Select only PAGE spans
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table or Bar Chart

## Query #7: Pages with Most Queries

```
SELECT
    average(rock.db.query_count)
    , max(rock.db.query_count)
    , min(rock.db.query_count)
FROM
    Span
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
    AND entity.name = 'rock-rms' -- This is the "Observability Service Name" defined in
System Settings
    AND name LIKE 'Page%' -- Select only PAGE spans
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table or Bar Chart

## Query #8: Slowest Blocks by Page

```
SELECT
    average(duration.ms)
    , max(duration.ms)
    , min(duration.ms)
FROM
    Span
WHERE parent.id IN (
        SELECT id
        FROM Span
        WHERE instrumentation.provider = 'opentelemetry'
            AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
            AND entity.name = 'rock-rms'
            AND name = 'PAGE: GET person/{PersonId}/history' -- Enter the PAGE Span name here
        SINCE 2 WEEKS AGO
        LIMIT MAX
    )
    AND name LIKE 'BLOCK%' -- Select only BLOCK spans.
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

## Query #9: Block Time Outliers by Page

```
SELECT
    stddev(duration.ms)
FROM
    Span
WHERE parent.id IN (
        SELECT id
        FROM Span
        WHERE instrumentation.provider = 'opentelemetry'
            AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
            AND entity.name = 'rock-rms'
            AND name = 'PAGE: GET person/{PersonId}/history' -- Enter the PAGE Span name here
        SINCE 2 WEEKS AGO
        LIMIT MAX
    )
    AND name LIKE 'BLOCK%' -- Select only BLOCK spans.
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table or Bar Chart

## Query #10: Blocks with Highest Query Count

```
SELECT
    average(db.span.count)
    , max(db.span.count)
FROM
    Span
    JOIN (
        SELECT count(*) AS 'db.span.count'
        FROM Span
        WHERE name LIKE 'DB%'
        FACET parent.id AS 'block.span.id'
        SINCE 2 WEEKS AGO
        LIMIT MAX
    ) ON id = block.span.id
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%' -- Exclude bots
    AND entity.name = 'rock-rms' -- This is the "Observability Service Name" defined in
System Settings
    AND name LIKE 'BLOCK%' -- Select only BLOCK spans.
FACET name
SINCE 2 WEEKS AGO
LIMIT 100
```

**Recommended Visualization:** Table

## Query #11: Finding Available Span Types

```
SELECT
    uniques(Type)
FROM
    Span
WITH
    capture(name, r'(?P<Type>[^ :]*).*') AS Type
WHERE
    instrumentation.provider = 'opentelemetry'
    AND http.user_agent NOT LIKE '%Bot%'
    AND entity.name = 'rock-rms'
SINCE 2 WEEKS AGO
LIMIT MAX
```

**Recommended Visualization:** Table