



# Dashboard techniques to visualize system and business performance

Phil Weber  
Lead Technical Training Specialist



© 2024 New Relic, Inc. All rights reserved.

## Training account credentials

Log out of other New Relic accounts (or open private browser window)

URL: <https://one.newrelic.com/>

Email: [demotron@newrelicuniversity.com](mailto:demotron@newrelicuniversity.com)

Password: o11y-as-code

© 2024 New Relic, Inc. All rights reserved.



# Agenda

## New query experience

Single 'Query your data' experience, keep context, survives navigation, new features

## NRQL & Dashboards 101

Event data, event types, NRQL syntax in NRDB New Relic Database

## Dashboard design strategies

Pyramids (linked charts & dashboards), patterns, markdown annotation, wallboards

## Trends and comparing data sets

Multi-function/facet, COMPARE WITH, multi-query per chart, filter function

## Using different chart types to visualize data

Funnels, Histogram, Heatmap

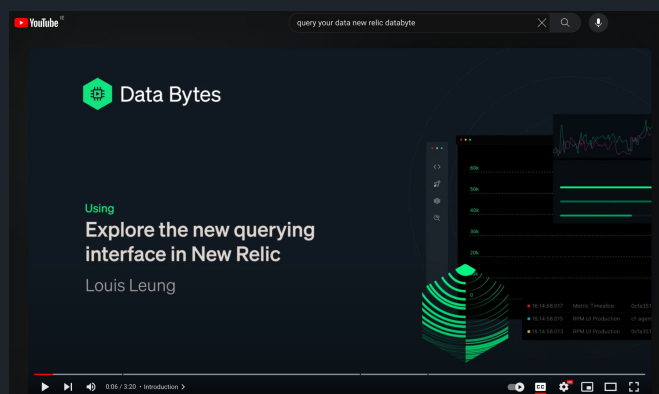
© 2024 New Relic, Inc. All rights reserved.



# New query experience

## What's new?

- New, improved UI
- Replaces existing multiple query builders
- Explore with context
- Always present, survives navigation
- Minimizable & resizeable for large queries
- Cross-account queries
- Multi-query
- Recommended queries button



<https://youtu.be/sjms5NYRMLs>

© 2024 New Relic, Inc. All rights reserved.





# Some of the main event types

Browser  
PageView

APM  
Transaction

Synthetics  
SyntheticCheck

Mobile  
MobileRequest

Infrastructure  
SystemSample



**SELECT** function(attribute) **FROM** EventType

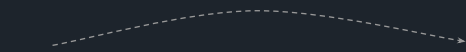


**FROM** EventType **SELECT** function(attribute)

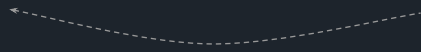
- Build queries around your **chosen event type**
- Use the **auto-complete** to help write your queries
- Functions and attributes will be suggested for each Event type

## NRQL syntax

### SELECT ... FROM



**SELECT** function(attribute) **FROM** EventType **WHERE** condition **FACET** group by



**SELECT** count(\*) **FROM** Transaction **WHERE** appName = 'Frontend' **FACET** name

**FROM** Transaction **SELECT** count(\*) **WHERE** appName = 'Frontend' **FACET** name

5 mins

# Lab: Explore your data

## Try these queries

Select *Query Your Data*

1

```
SHOW EVENT TYPES
```

Which event types are reporting?

Look up default event types and attributes in the [New Relic data dictionary](#)

2

```
SELECT keyset() FROM Transaction
```

Explore the attributes. Do you think any of them are custom / business attributes?

3

```
SELECT count(*) FROM Transaction
```

Aggregate the data to retrieve a single value

NOTE: You can't FACET until you aggregate

☒ `SELECT * FROM`


☒ `SELECT count(*)`

4

```
SELECT count(*) FROM Transaction FACET appName
```

Select DISTINCT values (GROUP BY) using FACET

© 2024 New Relic, Inc. All rights reserved.

 new relic 12

5 mins

# Challenge!

Count transactions in Demogorgon Prod account. Are more related to USD or EUR currency attribute?

Write a query in the *Query Your Data* console

1

Hint: Use FROM Transaction

Hint: Use FACET

```
SELECT count(*) FROM Transaction FACET currencyCode
```

Bonus: How do these numbers compare over time, for a week?

2

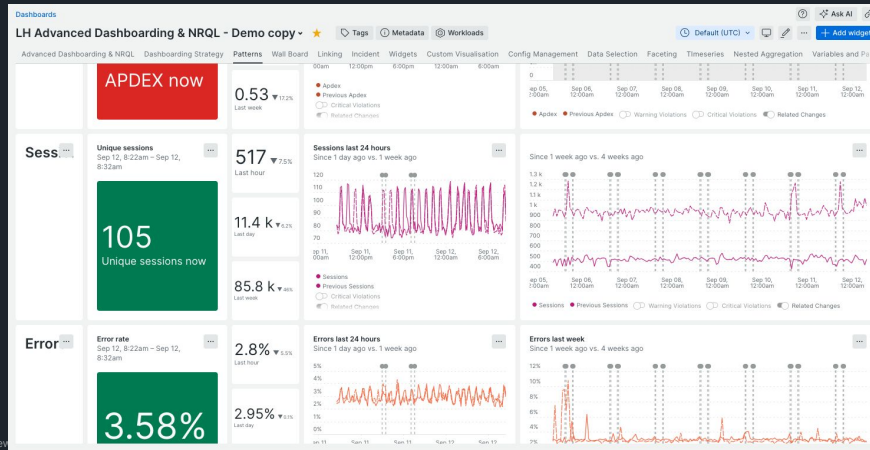
```
SELECT count(*) FROM Transaction WHERE currencyCode IN ('USD','EUR')
SINCE 1 week ago FACET currencyCode TIMESERIES
```

© 2024 New Relic, Inc. All rights reserved.

 new relic 13

# Explore the demo dashboard

- Use the permalink shared by the Instructor  
<https://onenr.io/0kin4mkNXwo>



© 2024 New Relic, Inc. All rights reserved.

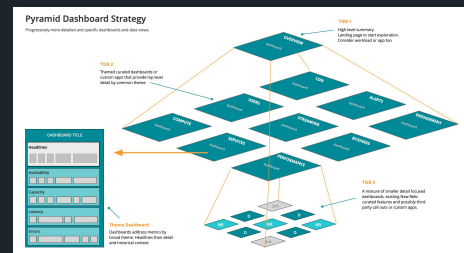
new relic 16

# Dashboard design and layout tips

## Pyramids, patterns, and markdown annotation

Explore the *Dashboarding Strategy* dashboard

- How does it work?
- Have you set up a structure like this?
- What tips would help make your dashboards more user friendly?



© 2024 New Relic, Inc. All rights reserved.

new relic 17

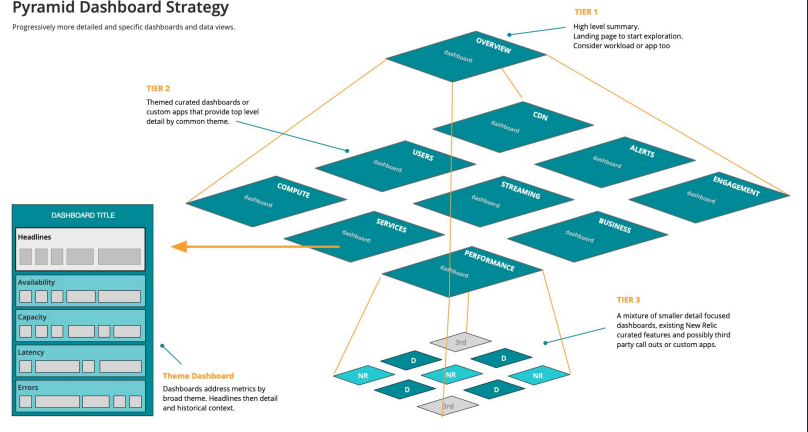
# Dashboard design and layout tips

## Pyramids, patterns, and markdown annotation

- Pyramid  
Tiered structure, interactive
- Patterns  
Now, 24 hrs, 7 days
- Wall boards  
TV mode, clear stats

### Pyramid Dashboard Strategy

Progressively more detailed and specific dashboards and data views.



# Create your own dashboard

- In *Demogorgon Prod* account
- Rename with your initials *XX-Demo-DB*
- Edit access

The screenshot shows the 'Create a dashboard' dialog box with two options:

- Browse pre-built dashboards**: Install a quickstart with dashboards made for what you're monitoring.
- Create a new dashboard**: Build a dashboard from scratch and see what's happening with your data.

The screenshot shows the 'Create a dashboard' form with the following fields:

- Dashboard name**: XX-Demo DB
- Select your account**: Account: 4120837 - Demogorgon Prod
- Permissions**: Edit - everyone in account

Buttons: Back, Create

5 mins

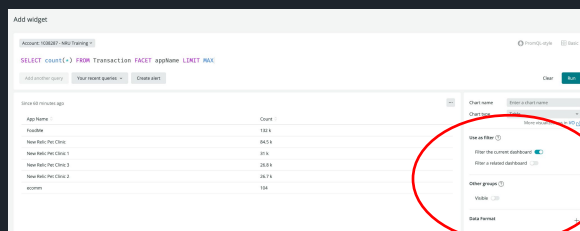
# Create a filter list of applications

Use a FACET and set the Link options in the chart

- Create a chart showing throughput of all your services

```
SELECT count(*) FROM Transaction
FACET appName LIMIT MAX
```

- Use the link options in the chart settings to:
  - Use **each service as an interactive filter** on the dashboard, or
  - Link to a detailed dashboard** of transactions, latency, errors, logs, etc



5 mins

# Patterns

Repeat a similar query: chart Billboard, Timeseries, SINCE 1 hour/day/week ago

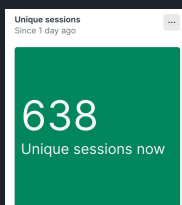
- 1 Add a page to your dashboard

- 2 Create a chart showing visitors to your site:

```
SELECT uniqueCount(session) FROM
PageView WHERE appName = 'Frontend'
SINCE 5 minutes ago
```

Use chart options: Billboard

Add a Threshold value



- 3 Create new charts showing visitors over time (1 day / 1 week ago)

```
SELECT uniqueCount(session) FROM
PageView WHERE appName = 'Frontend'
SINCE 1 day ago TIMESERIES
```

```
SELECT uniqueCount(session) FROM
PageView WHERE appName = 'Frontend' SINCE
1 week ago TIMESERIES
```



# Time ranges and TIMESERIES

## SINCE, SINCE...UNTIL, TIMESERIES

### SINCE

```
SELECT ... SINCE 1 day AGO | SELECT ... SINCE yesterday | SELECT ... SINCE sunday
```

### SINCE ... UNTIL

```
SELECT ... SINCE 3 days ago UNTIL 5 hours ago
```

### TIMESERIES to plot over time (auto-buckets)

```
SELECT ... SINCE 3 days ago SINCE 5 hours agoTIMESERIES
```

### Add a value **after** the TIMESERIES to override the auto-buckets, and set your evaluation window

```
SELECT ... SINCE 1 day AGO TIMESERIES 30 minutes
```

### Add MAX after TIMESERIES to get the max no. of intervals in the time window (auto scales)

(The maximum number of TIMESERIES buckets that will be returned is 366)

```
SELECT ... SINCE 1 day AGO TIMESERIES MAX
```

© 2024 New Relic, Inc. All rights reserved.

 new relic 23

5 mins

## Lab: Try out time spans and TIMESERIES

### Try out these queries

Go to Query Your Data and build the following charts, querying the Demogorgon Prod account.

1

```
SELECT count(*) FROM Transaction WHERE httpResponseCode = '500'
```


- SINCE \_\_\_\_\_
- SINCE \_\_\_\_\_ UNTIL \_\_\_\_\_

2

```
SELECT average(pageRenderingDuration) FROM PageView SINCE 12 hours ago
```

- TIMESERIES
- TIMESERIES 5 minutes
- TIMESERIES MAX

© 2024 New Relic, Inc. All rights reserved.

 new relic 24

# Adding your own values and thresholds to charts

Use *Customize Chart Settings > Thresholds*

or `SELECT` function(attribute), value x AS 'x', value y AS 'y'

```
SELECT average(duration), 0.1 as 'threshold', 0.05 as 'warning' FROM Transaction  
SINCE 1 day ago TIMESERIES
```



5 mins

## Compare performance over time SINCE...COMPARE WITH

**SINCE** or **UNTIL** time period is the fixed time period for the comparison (REQUIRED)

**COMPARE WITH** is the time shift amount relative (before) the time specified by **SINCE** or **UNTIL**

Example: compare yesterday with the day before, or yesterday with same day one week ago

```
SELECT ... SINCE 1 day ago COMPARE WITH 1 day ago
```

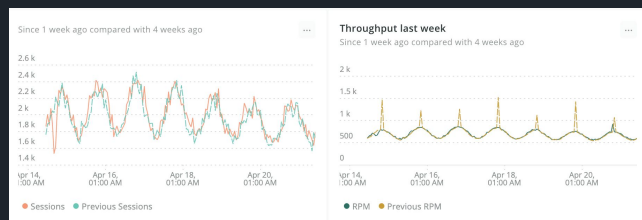
```
SELECT ... SINCE 1 day ago COMPARE WITH 1 week ago
```

Use the ... menu and **View Query** to explore the **COMPARE WITH** queries in the [Patterns dashboard](#):

1

Now try your own in your Test dashboard:

```
SELECT uniqueCount(session) FROM PageView  
WHERE appName = 'Frontend' SINCE 1 week ago  
COMPARE WITH 1 week ago  
TIMESERIES 1 hour
```



# Compare metrics/attributes with multi-function

SELECT more than one function on an event type

Comma-separate functions to create series of values / custom tables

```
SELECT average(duration), max(duration) FROM PageView
```

Add labels ... AS 'my label',

```
SELECT average(duration) AS 'avg pageload' FROM PageView
```

Example: Is backend or page load taking more time to complete?

```
SELECT average(backendDuration), average(duration) FROM PageView
WHERE appName = 'Frontend' TIMESERIES 1 minute
```

Try this query:

```
SELECT average(duration) AS 'page load', average(pageRenderingDuration)
AS 'page render', average(backendDuration) AS 'web app + network'
FROM PageView WHERE appName = 'Frontend'
```

## Compare data sets with multi-FACET and FACET CASES

Multi-FACET

```
SELECT average(duration) from PageView facet pageUrl, countryCode, city
```

FACET CASES (using labels)

```
SELECT count(*) FROM Transaction FACET CASES (WHERE name LIKE '%cart%' AS
'cart', WHERE name LIKE '%checkout%' AS 'checkout', WHERE name LIKE
'%shipping%' AS 'shipping')
```

FACET CASES (defining brackets)

```
SELECT count(*) from PageView FACET CASES (WHERE backendDuration < 0.5
AS '< 0.5', WHERE backendDuration < 1 as '0.5 - 1', WHERE backendDuration >=
1 as '> 1')
```

# Compare data sets with multi-query / multi-account

Add another query to double up on queries per chart!

Compare 2 services

... or compare different accounts to compare performance in staging and production

Account: Demo Prod

```
SELECT average(duration) AS 'Frontend' FROM Transaction WHERE appName = ' Frontend' TIMESERIES
```

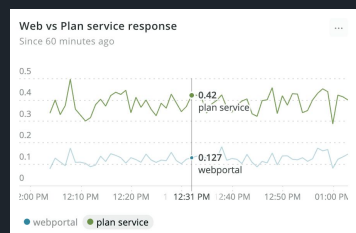
Account: Other

```
SELECT average(duration) AS 'xxxxx' FROM Transaction WHERE appName = ' xxxxx' TIMESERIES
```

1

Try out this feature -

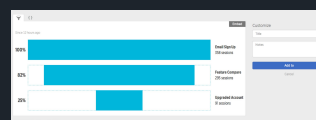
or explore the chart in your current dashboard > Visualize page



## Funnel

SELECT funnel(attribute, Step A, Step B)

Example [dashboard](#):



```
SELECT funnel(session, WHERE pageUrl LIKE '%/' AS 'Homepage',
```

```
WHERE pageUrl LIKE '%/browse/plans' AS 'Click Product',
```

Step A                      Label

```
WHERE pageUrl LIKE '%login.jsp' AS 'Login',
```

Step B                      Label

```
WHERE pageUrl LIKE '%/checkout' AS 'Confirm')
```

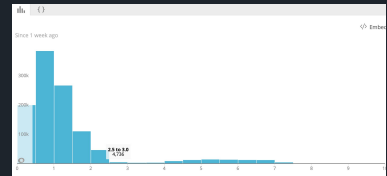
# Histogram

```
SELECT histogram(attribute, ceiling , [number of buckets])
```

Example:

Show me a Histogram of page load times ranging up to 10 seconds over 20 buckets

```
SELECT histogram(duration, 10, 20) FROM PageView  
SINCE 1 day ago
```



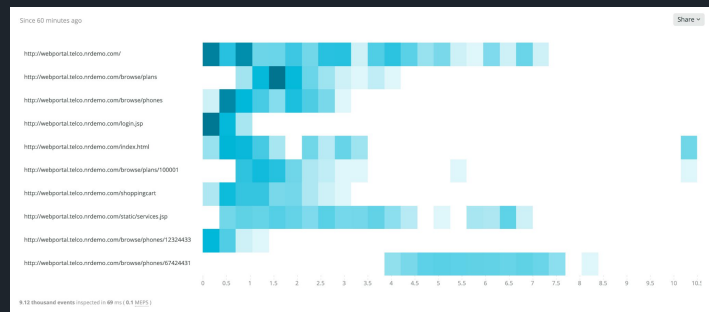
Diagnose slow pages with Histogram & Heatmap New Relic blog: [DataByte video :heatmaps-histograms](#)

# Heatmap

## Add FACET to Histogram

Example: Show me a heatmap of response times ranging up to 10 seconds, 20 buckets, grouped by pageUrl

```
SELECT histogram(duration, 10, 20) from PageView FACET pageUrl
```



# Lab: Create a histogram and heatmap

- 1 Create a chart with a Histogram of pageload times (PageView) ranging up to 10 seconds with 30 buckets
- 2 Turn this into a heatmap of page load times for different urls by adding **FACET** pageUrl
- 3 Explore the **Funnel chart** in your current dashboard > Visualize page

## Customize widgets

Examples: y-axis, number/time formats, ignore timepicker

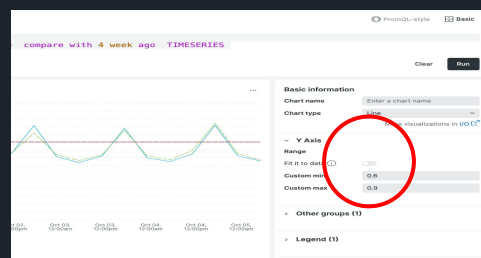
**Data format:** attribute, number, decimals, 2

```
SELECT average(duration) FROM Transaction
WHERE appName = 'Frontend' AND name LIKE
'%browse%' SINCE 1 day ago
```

**Y-axis: (add TIMESERIES) Fit to range OR edit min / max values**

```
SELECT average(duration) FROM Transaction
WHERE appName = 'Frontend' AND name LIKE
'%browse%' SINCE 1 day ago TIMESERIES
```

**Dashboard options:** ignore timepicker (fixed query)



**TIMESERIES:** Edit Y Axis (0.6 to 0.9)

**Bullet:** Edit Data format (+ attribute, number, decimals, 2  
**All:** Dashboard options - ignore timepicker

# Config, import, API

## Sharing and setting up access on dashboards

- Dashboards API: Full C R U D
- Dashboards are entity-based. Use the GUID in API calls
- Learn more here:  
<https://docs.newrelic.com/docs/query-your-data/explore-query-data/dashboards/dashboards-api/>

## More resources

1

NRQL: Joins, Lookup, GeoIP  
[Databyte video](#)

3

Custom visualization: Line and Bar Charts  
[Databyte video](#)

2

Charts feature: Connected data sets  
[Databyte video](#)

4

Dashboard - Template Variables  
[Databyte video part 1](#)  
[Databyte video part 2](#)