

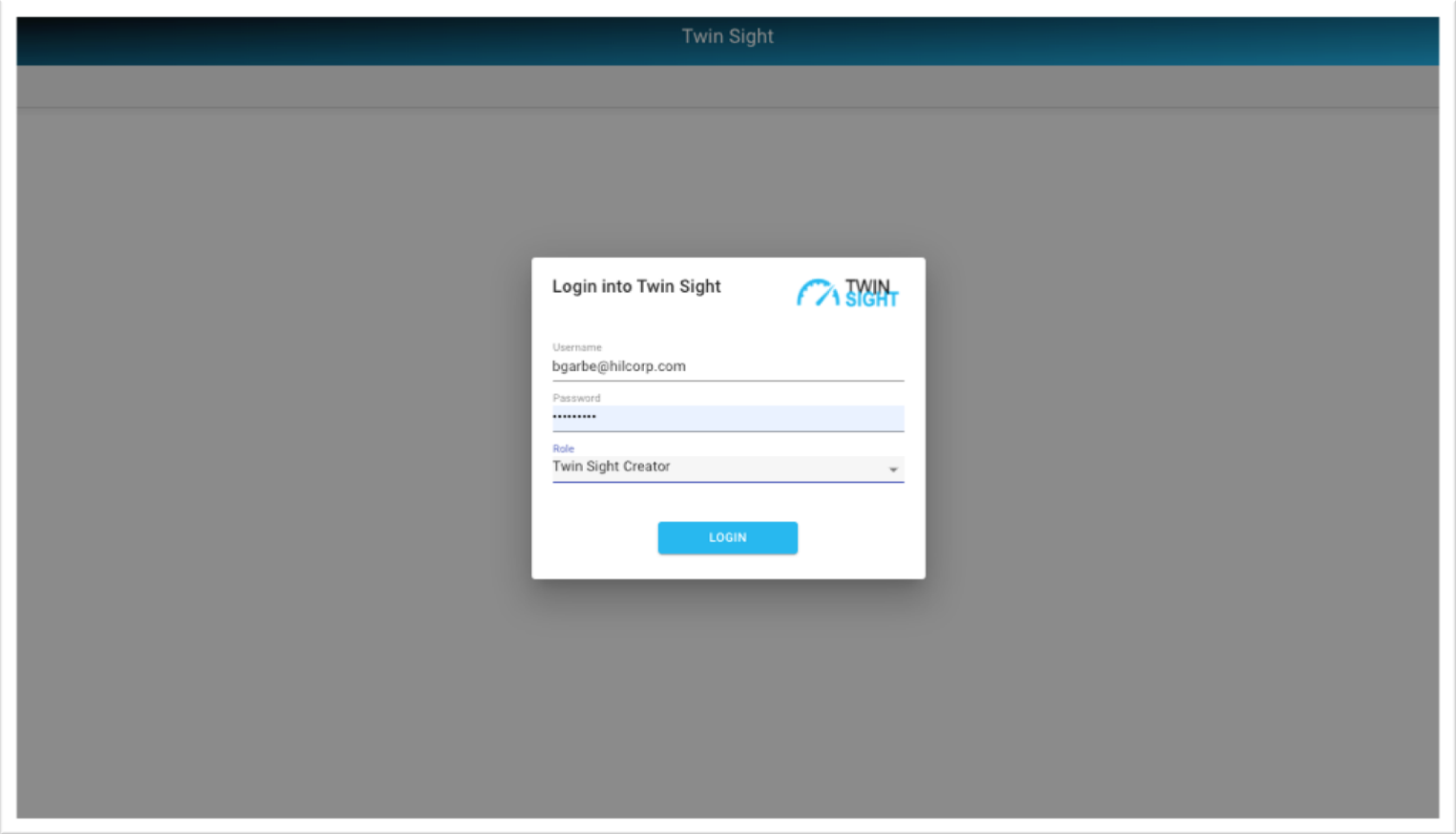


Twin Sight Operational Dashboard Builder

Enable Industrial Corporations to create use-case driven, asset model based dashboards leverage legacy, standard and big data/ml data sources:

- Use multiple data sources to create single sources of truth
- Create use case specific dashboard
- Standardize dashboards across company
- Include significant assets/attributes/trends
- Build, save, & deploy custom dashboards

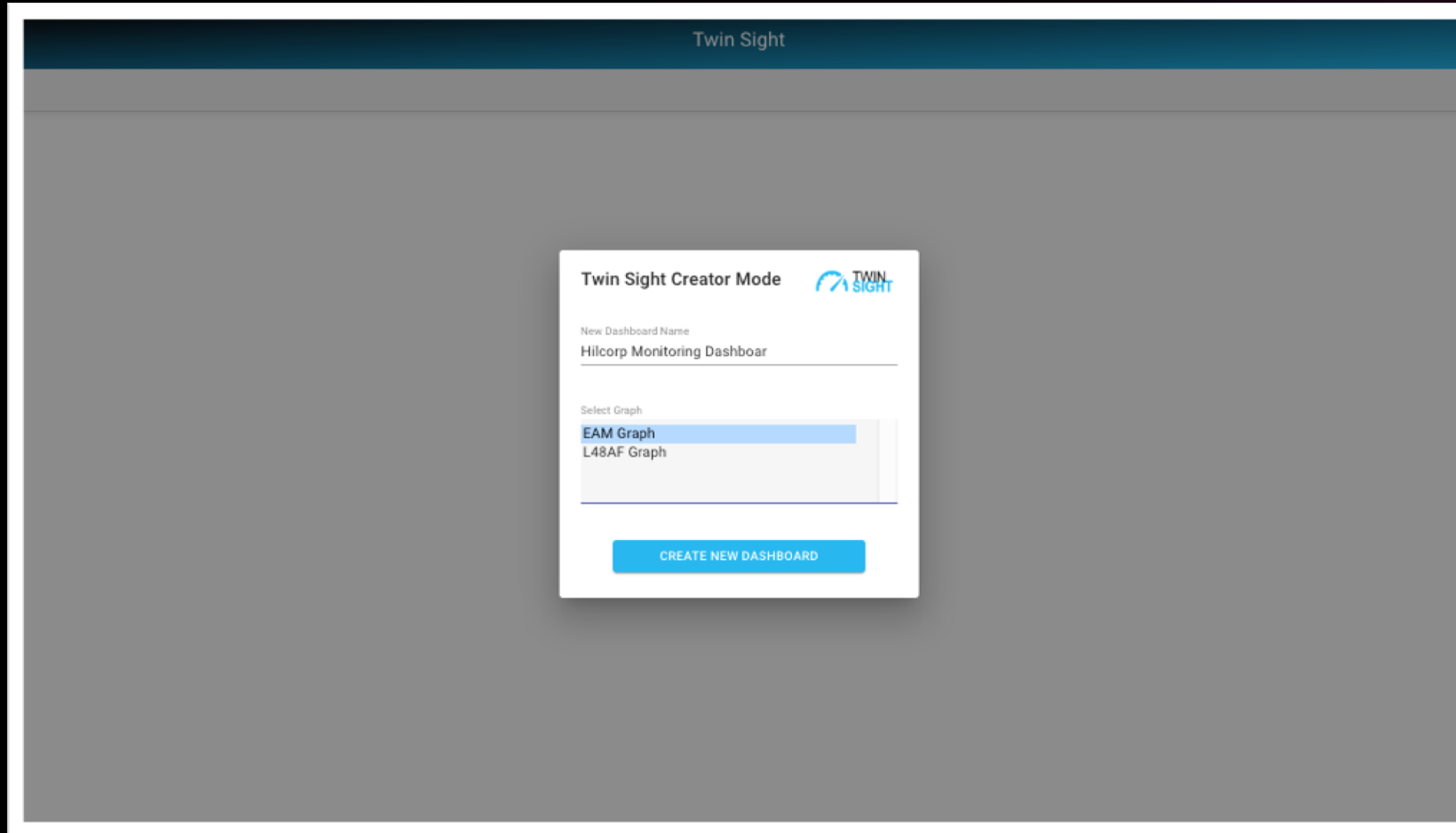
Logging into Twin Sight



Brent	Garbe	bgarbe@hilcorp.com
Matt	Hall	mahall@hilcorp.com
Ostap	Zarivnyy	ozarivnyy@hilcorp.com
Reed	Hanson	rhanson@hilcorp.com

"password": "vip@12345"

Naming Dashboard and Choosing Graph/Tree



The screenshot displays the 'Twin Sight Creator Mode' interface. At the top, a dark teal header bar contains the text 'Twin Sight'. The main content area is a light gray rectangle. Centered within this area is a white modal form. The form has a title 'Twin Sight Creator Mode' and the 'TWIN SIGHT' logo to its right. Below the title, there is a label 'New Dashboard Name' followed by a text input field containing 'Hilcorp Monitoring Dashboar'. Underneath this is a label 'Select Graph' followed by a list box. The list box contains two items: 'EAM Graph' (which is highlighted with a blue background) and 'L48AF Graph'. At the bottom of the form is a blue button with the text 'CREATE NEW DASHBOARD'.

Twin Sight

Twin Sight Creator Mode

New Dashboard Name
Hilcorp Monitoring Dashboar

Select Graph
EAM Graph
L48AF Graph

CREATE NEW DASHBOARD

User Interface Overview

Dashboard Settings

Save Dashboard

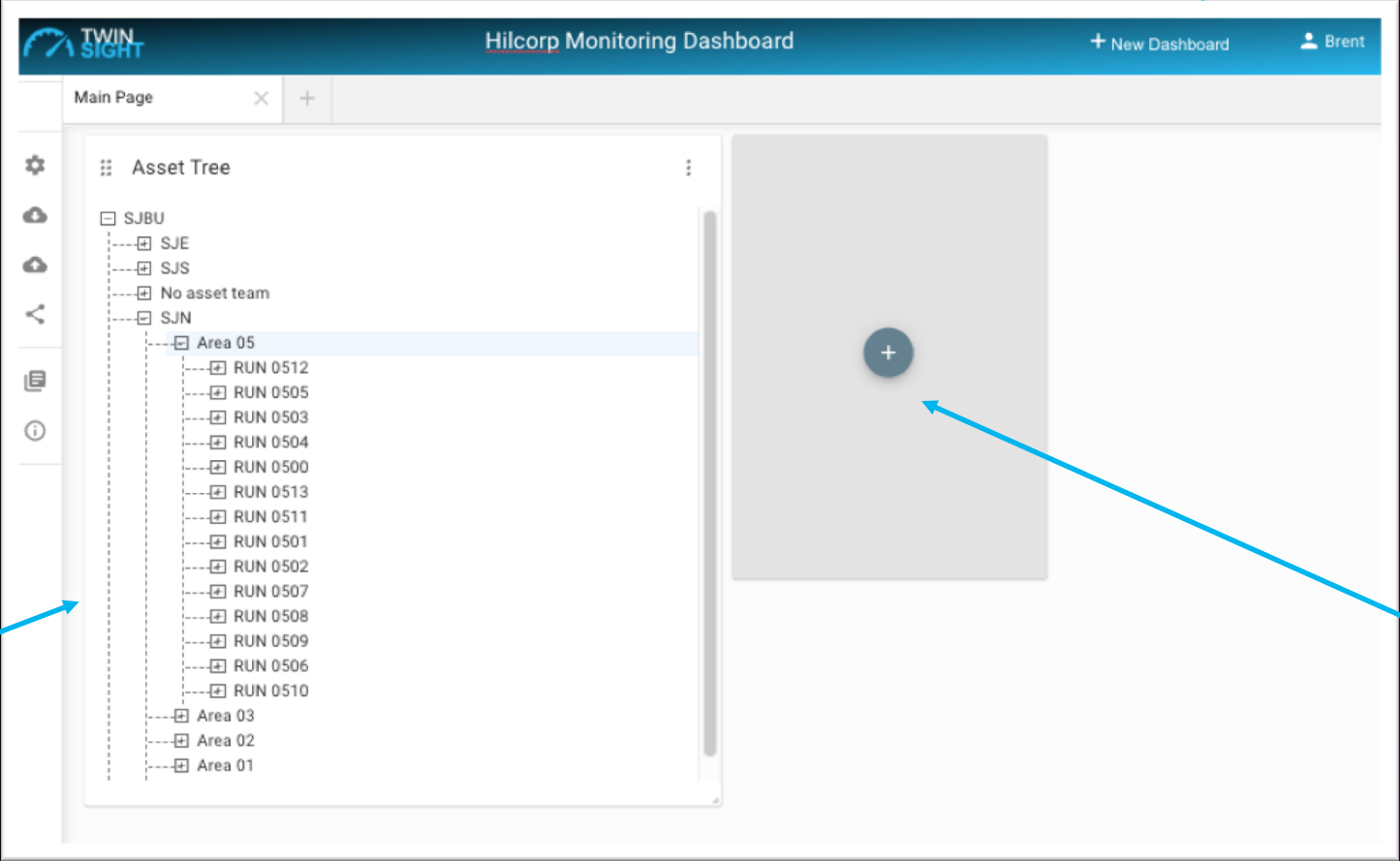
Load Dashboard

Share Dashboard

Future User Guide

Info/Help

Navigation Tree

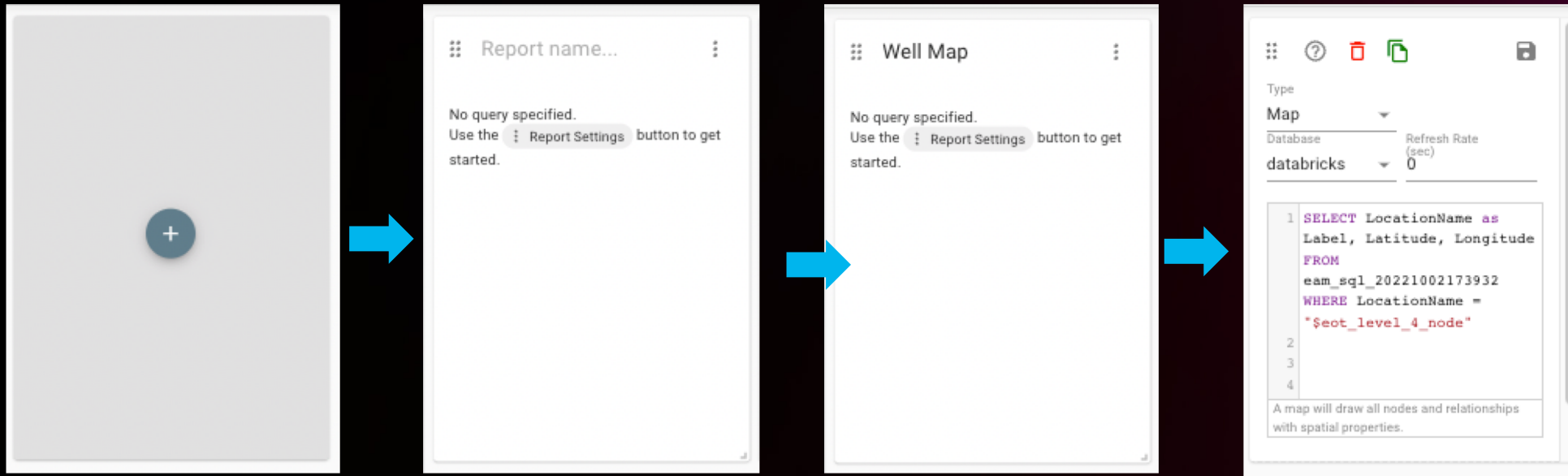


Building new dashboards

User Info

Report Creator

Creating a new report 1 – Name the report



Configure the report

databricks
athena
twinmaker
sitewise
sql

Step 1: Select Datasource

Step 4: Make it context sensitive

Type
Map

Database
databricks

Refresh Rate (sec)
0

```
1 SELECT LocationName as  
   Label, Latitude, Longitude  
FROM  
eam_sql_20221002173932  
WHERE LocationName =  
"$eot_level_4_node"
```

A map will draw all nodes and relationships with spatial properties.

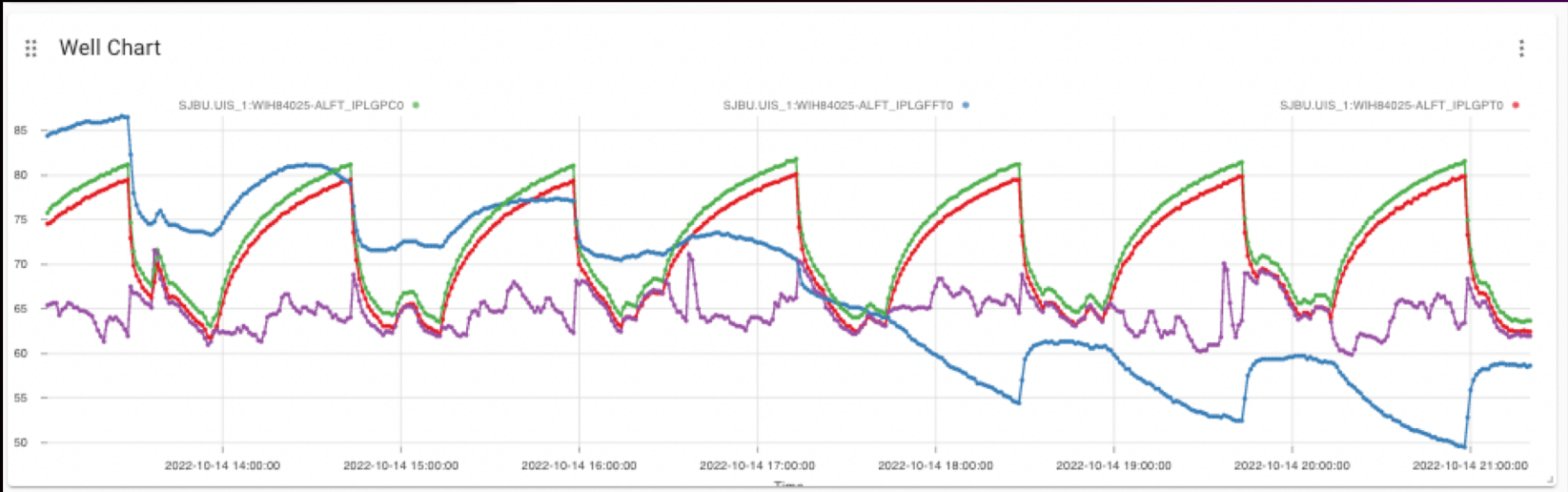
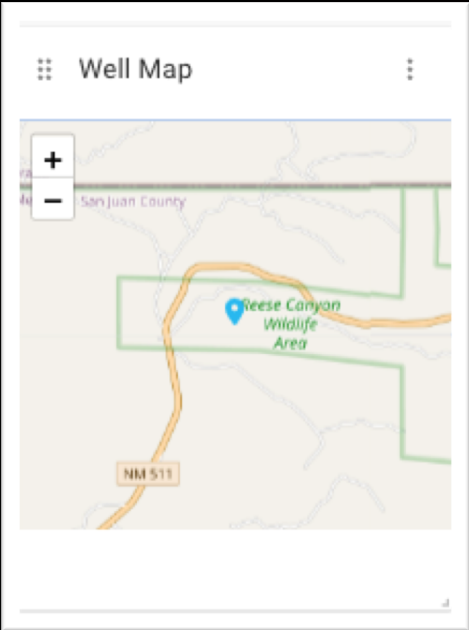
Step 5: Save it

Step 3: Select Chart Type

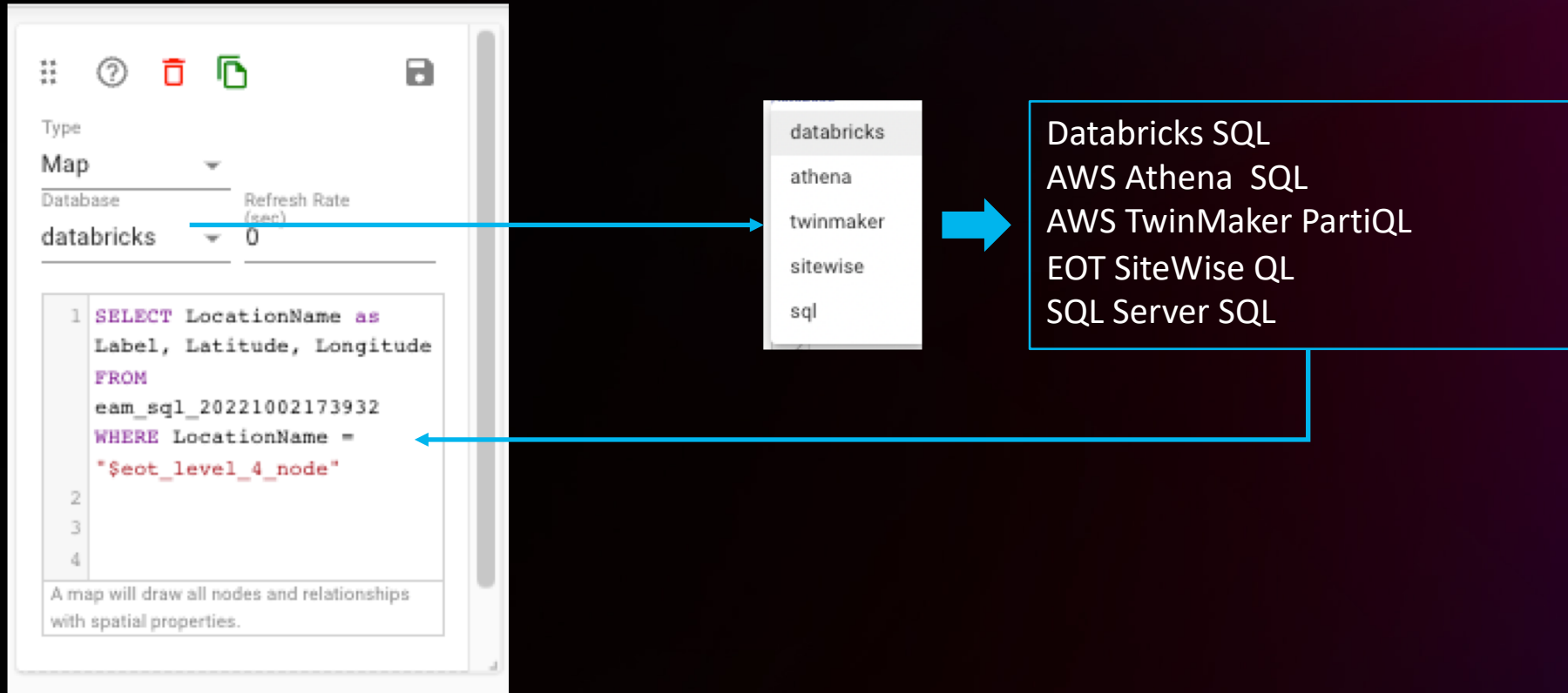
Step 2: Write Query

Tree View
Table
Graph
Bar Chart
Pie Chart
Line Chart
Radar Chart
Map
Single Value
Gauge Chart
Sunburst Chart
Circle Packing
Treemap
Sankey Chart
Choropleth Map
Raw JSON
Parameter Select
iFrame

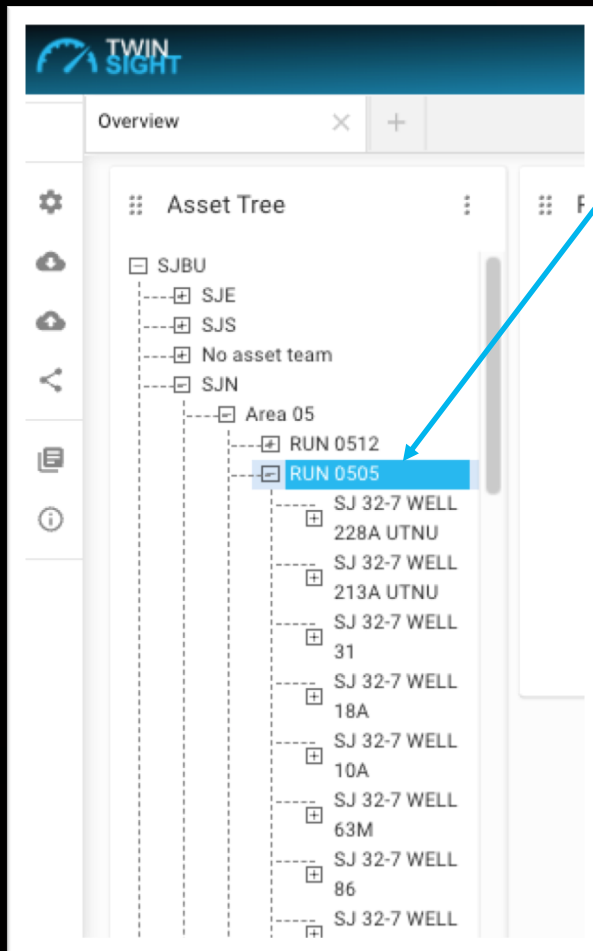
See the results - Examples



Writing the query: Dialect depends on the Datasource

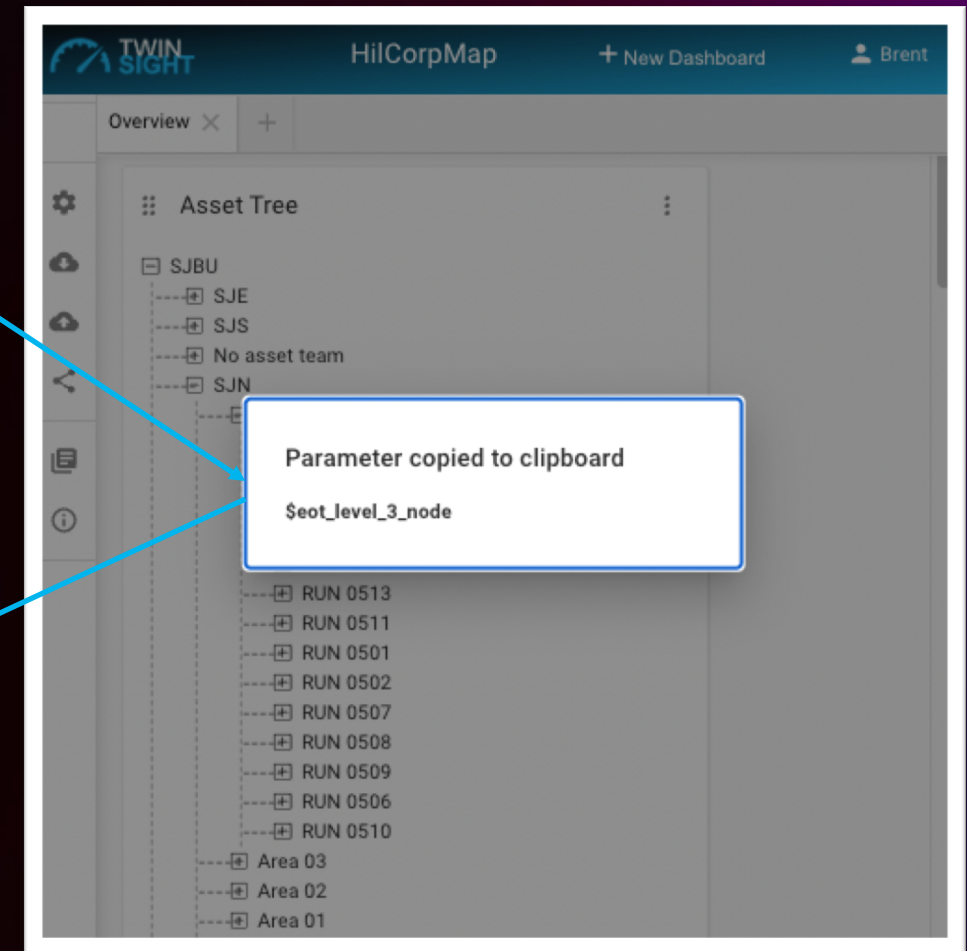
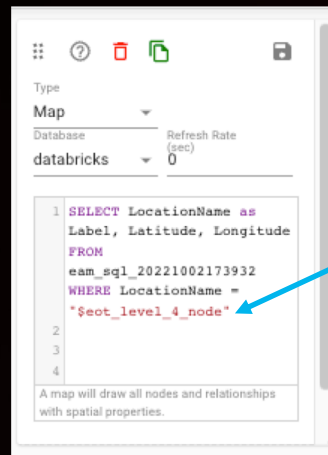


Creating a Contextual Reference



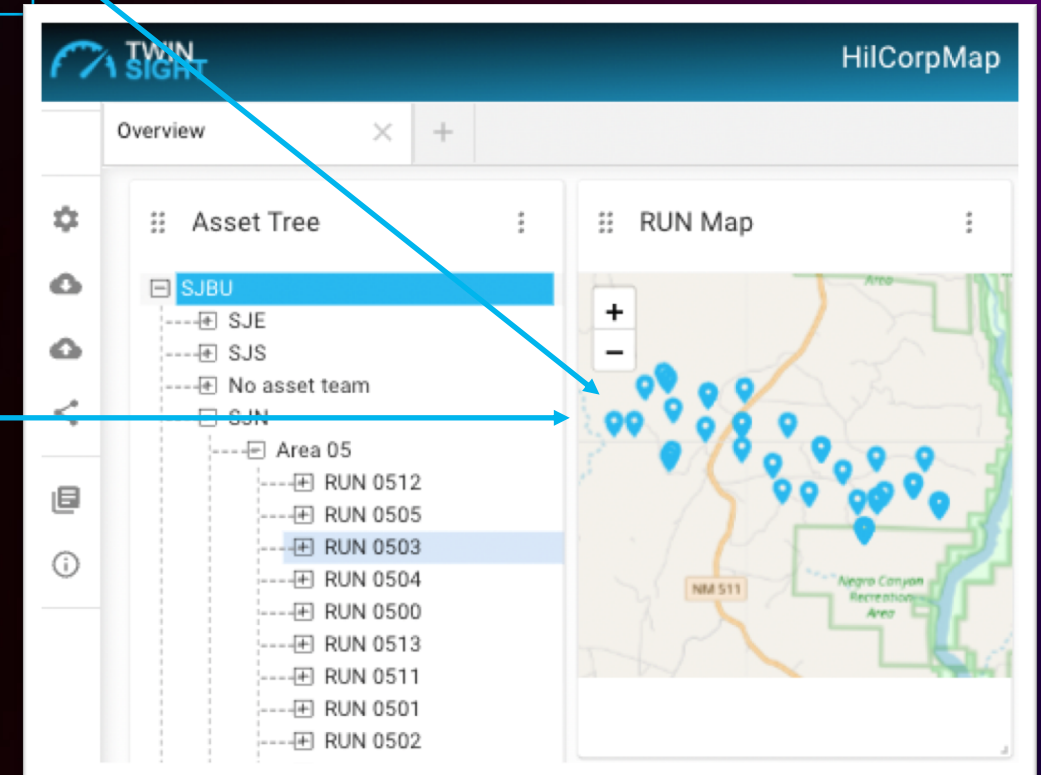
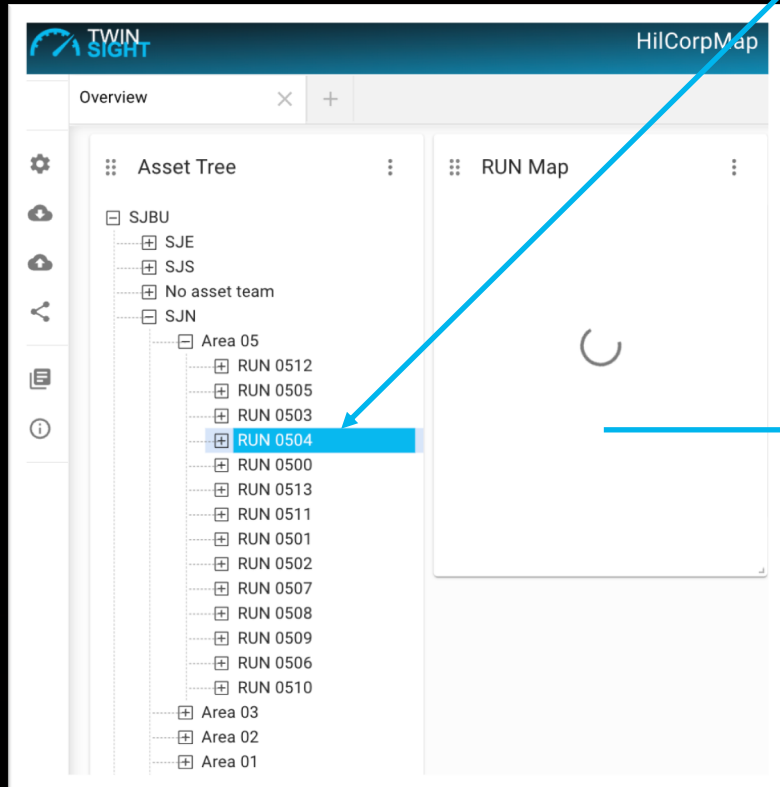
Right-Clicking the Node

- a) Creates a new referenceable variable
- b) Copies its name into the clipboard where it then can be used to paste it into the query



Using a Contextual Reference when Navigating

Left-Clicking the Node will update all reports that use that contextual reference variable



Visualizing Databricks (and other data sources) as Historical Data in Line Chart

Type: Line Chart Database: databricks Refresh Rate (sec): 0

```
1 SELECT measuredate as Date, value, tag as multimeseries
2 FROM historian.measurements
3 WHERE measuredate > to_timestamp('$eot_main_page_time_frame_start') AND measuredate <
  to_timestamp('$eot_main_page_time_frame_end') AND (tag =
  "SJBUSUIS_1:$eot_report_main_page_well_details_meter-ALFT_IPLGFFT0" OR tag =
  "SJBUSUIS_1:$eot_report_main_page_well_details_meter-ALFT_IPLGPT0" OR tag =
  "SJBUSUIS_1:$eot_report_main_page_well_details_meter-ALFT_IPLGPC0" OR tag =
  "SJBUSUIS_1:$eot_report_main_page_well_details_meter-ALFT_IPLGPS0" )
4 ORDER BY measuredate
5 LIMIT 2000;
```

A line chart expects two fields: an x value and a y value. The x value can be a number or a datetime object. Values are automatically selected from your query.

Advanced settings

Contextual variables from Date Time Picker for start and end time

Time Frame

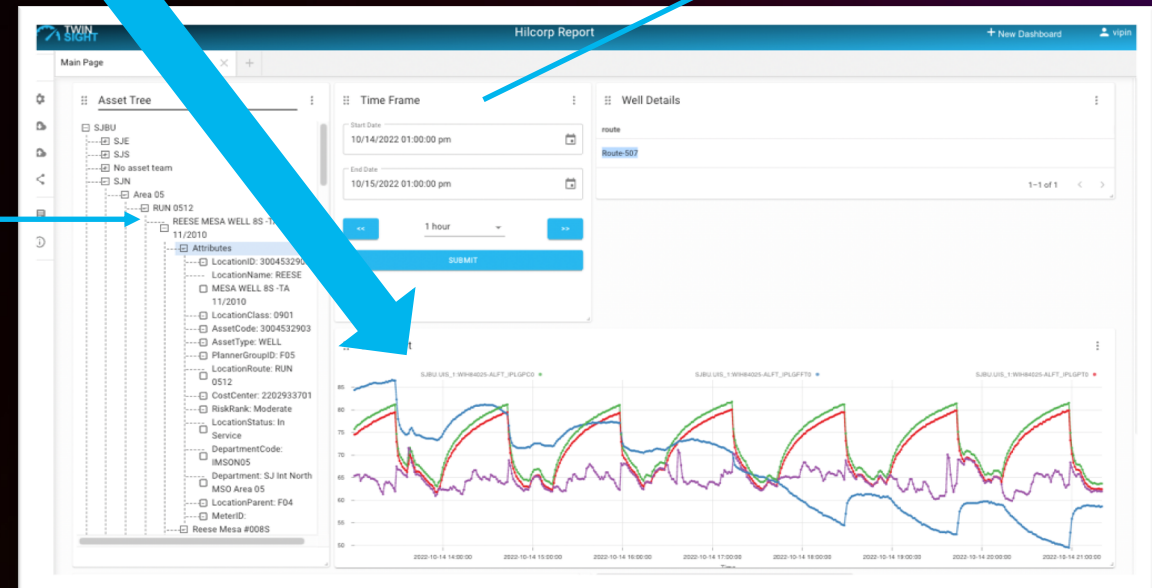
Start Date: 10/14/2022 01:00:00 pm

End Date: 10/15/2022 01:00:00 pm

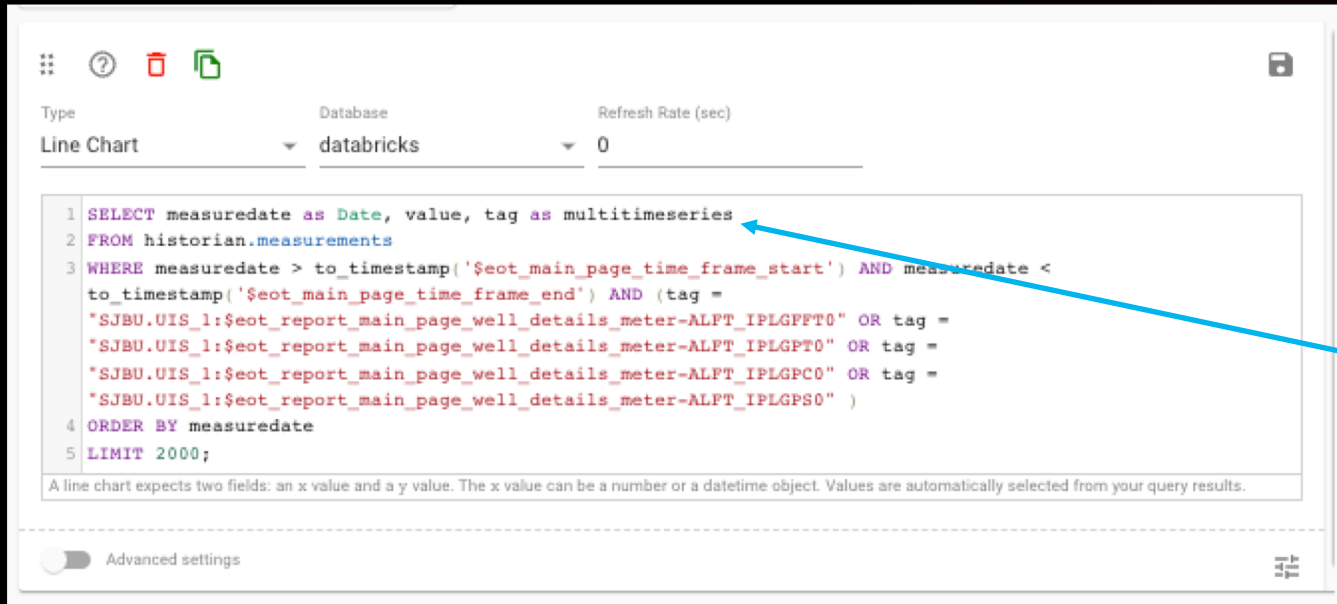
<< 1 hour >>

SUBMIT

Contextual variables for Well from Tree Navigator



Identifying the Lines in the Line Chart for Historical Data Display



The screenshot shows a configuration interface for a line chart. At the top, there are icons for settings, help, delete, and save. Below these are dropdown menus for 'Type' (set to 'Line Chart'), 'Database' (set to 'databricks'), and 'Refresh Rate (sec)' (set to '0'). The main area contains a SQL query editor with the following code:

```
1 SELECT measuredate as Date, value, tag as multimeseries
2 FROM historian.measurements
3 WHERE measuredate > to_timestamp('$eot_main_page_time_frame_start') AND measuredate <
  to_timestamp('$eot_main_page_time_frame_end') AND (tag =
  "SJBUI1:$eot_report_main_page_well_details_meter-ALFT_IPLGFFT0" OR tag =
  "SJBUI1:$eot_report_main_page_well_details_meter-ALFT_IPLGPT0" OR tag =
  "SJBUI1:$eot_report_main_page_well_details_meter-ALFT_IPLGPC0" OR tag =
  "SJBUI1:$eot_report_main_page_well_details_meter-ALFT_IPLGPS0" )
4 ORDER BY measuredate
5 LIMIT 2000;
```

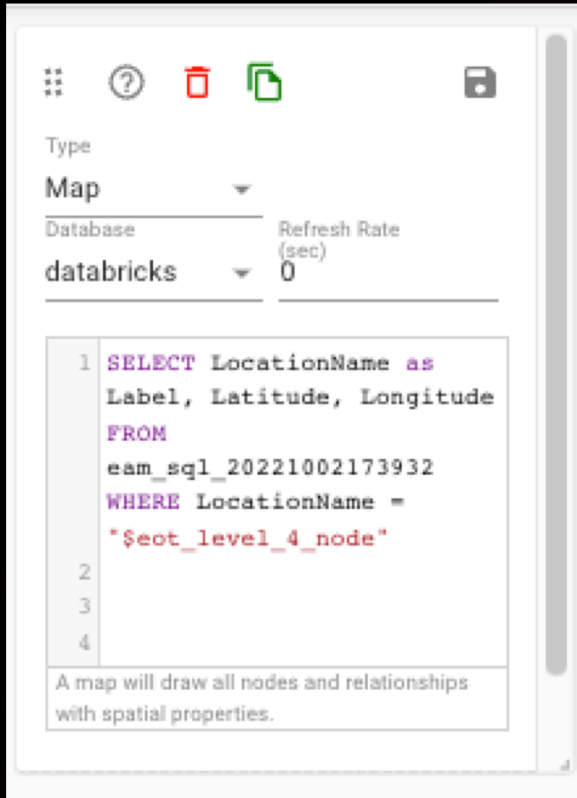
Below the query editor, a note states: "A line chart expects two fields: an x value and a y value. The x value can be a number or a datetime object. Values are automatically selected from your query results." At the bottom, there is an 'Advanced settings' toggle switch.

Historical data consist of:

- Timestamps
- Values
- Names of Sensor/Tags

The field for sensor/tag names has to be identified as "multimeseries" (So it will automatically create a chart with multiple lines using the field as a dictionary)

Visualizing Databricks (and other data sources) as geospatial data in map



The screenshot shows a configuration window for a map. At the top, there are icons for a grid, help, delete, and save. Below these, the 'Type' is set to 'Map'. The 'Database' is set to 'databricks' and the 'Refresh Rate (sec)' is set to '0'. A SQL query is entered in a text area:

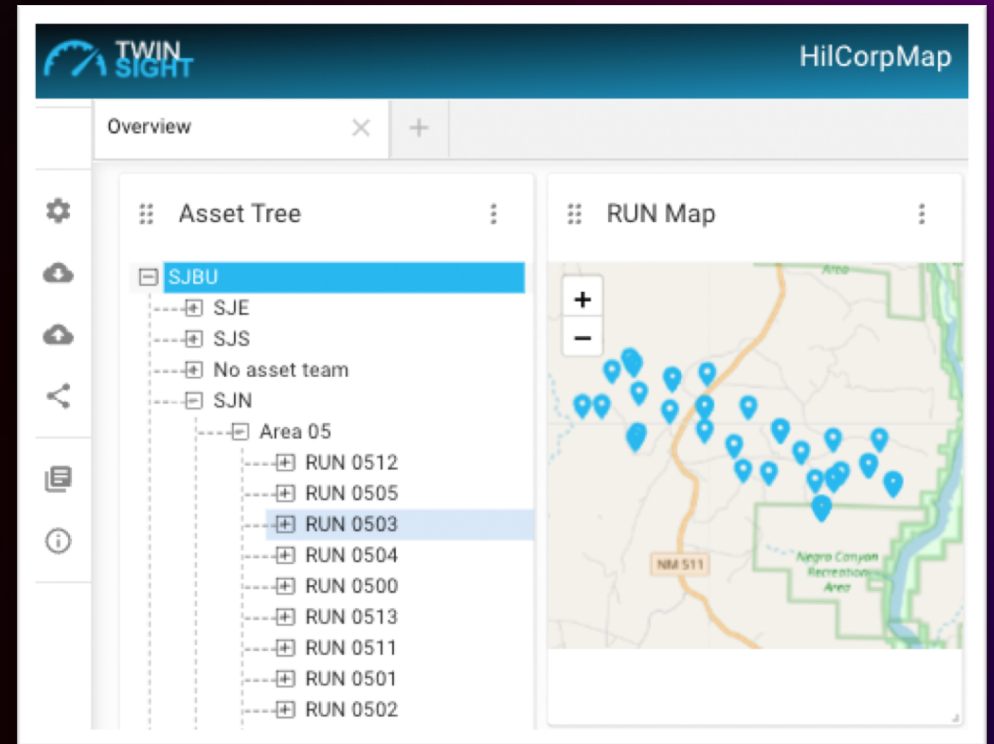
```
1 SELECT LocationName as  
Label, Latitude, Longitude  
FROM  
eam_sql_20221002173932  
WHERE LocationName =  
"$eot_level_4_node"
```

Below the query, a note states: "A map will draw all nodes and relationships with spatial properties."

Queries for geospatial data require 3 fields with the following names:

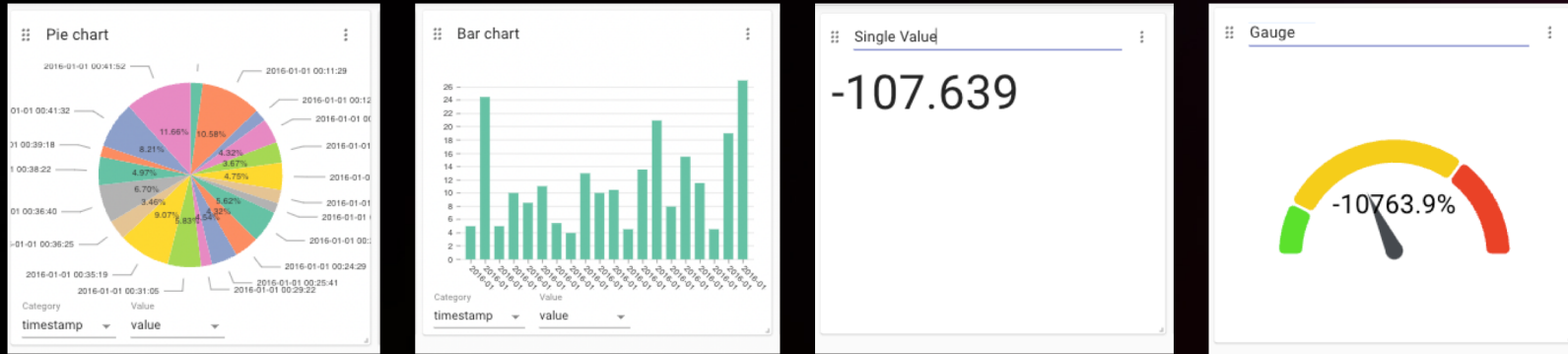
- Latitude
- Longitude
- Label

Use "as" in SQL to create names that match the above.



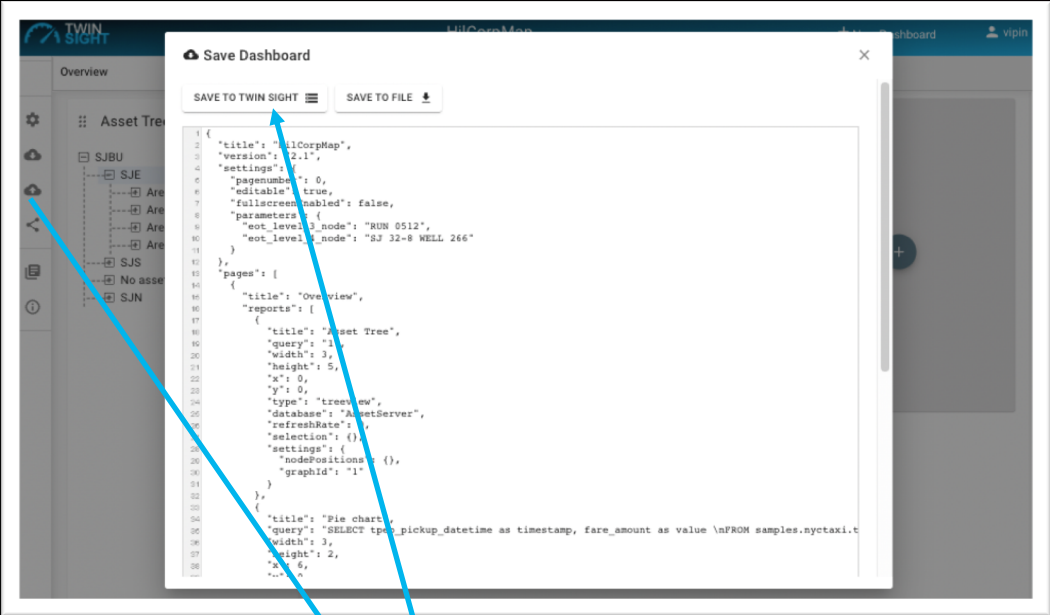
The screenshot shows the TWIN SIGHT HilCorpMap interface. The top bar has the TWIN SIGHT logo and the title 'HilCorpMap'. Below the bar, there are tabs for 'Overview' and '+'. The main area is divided into two panels. The left panel, titled 'Asset Tree', shows a hierarchical list of assets: SJB, SJS, No asset team, and SJN. Under SJN, there is a sub-tree for 'Area 05' containing a list of RUN assets: RUN 0512, RUN 0505, RUN 0503 (highlighted), RUN 0504, RUN 0500, RUN 0513, RUN 0511, RUN 0501, and RUN 0502. The right panel, titled 'RUN Map', shows a map view with blue location pins and a road labeled 'NM 511'. The map also shows 'Angie Canyon Recreation Area'.

Visualizing Databricks (and other data sources) other charts

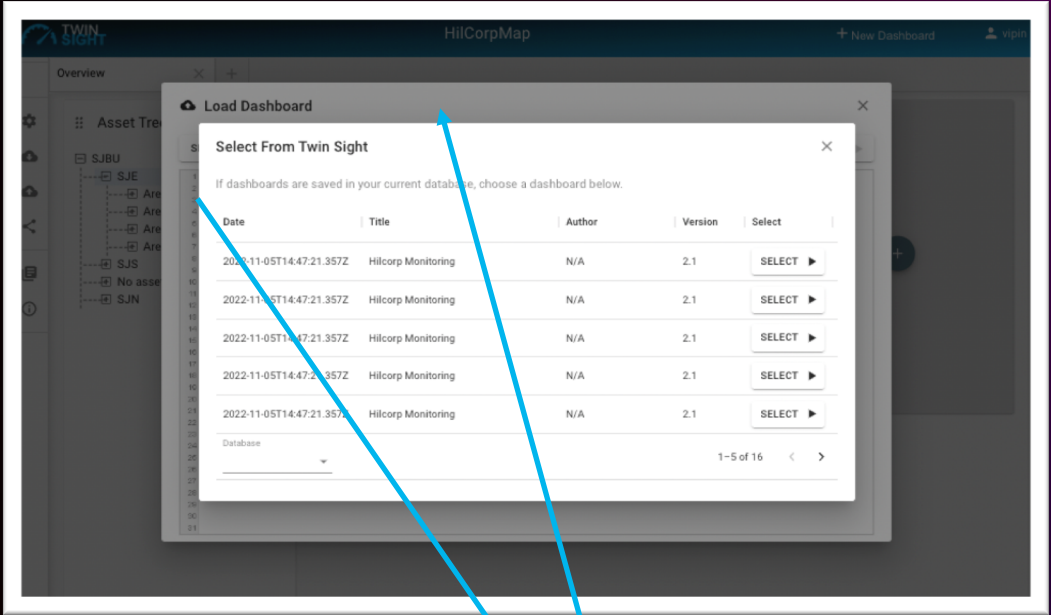


Queries have to match what is required by the chart type (single value for Gauge and Single Value charts, 2 dimensional arrays for Pie and Bar charts, etc)

Save and Loading Dashboards



Dashboards can be saved locally as JSON files or on the Twin Sight Server



Dashboards can be loaded from local JSON files or from the Twin Sight Server

