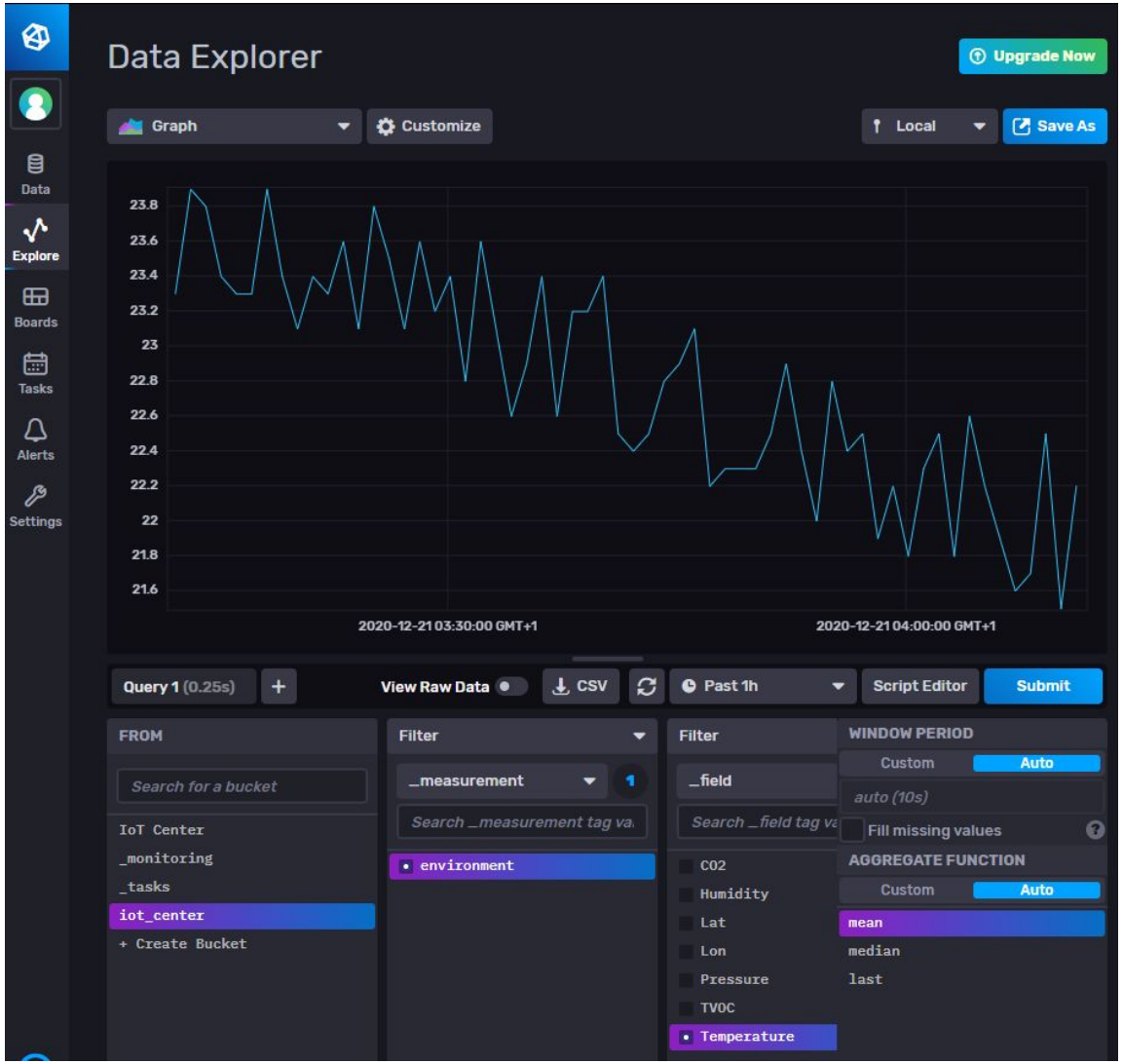
influxdata
Act in Time

InfluxDB Data Explorer

Workshop

Agenda

- InfluxDB Cloud Login
- Introduce Flux Query Language
- Data Explorer
- Exercise: Query GEO Data



Login into Influxdb Cloud 2

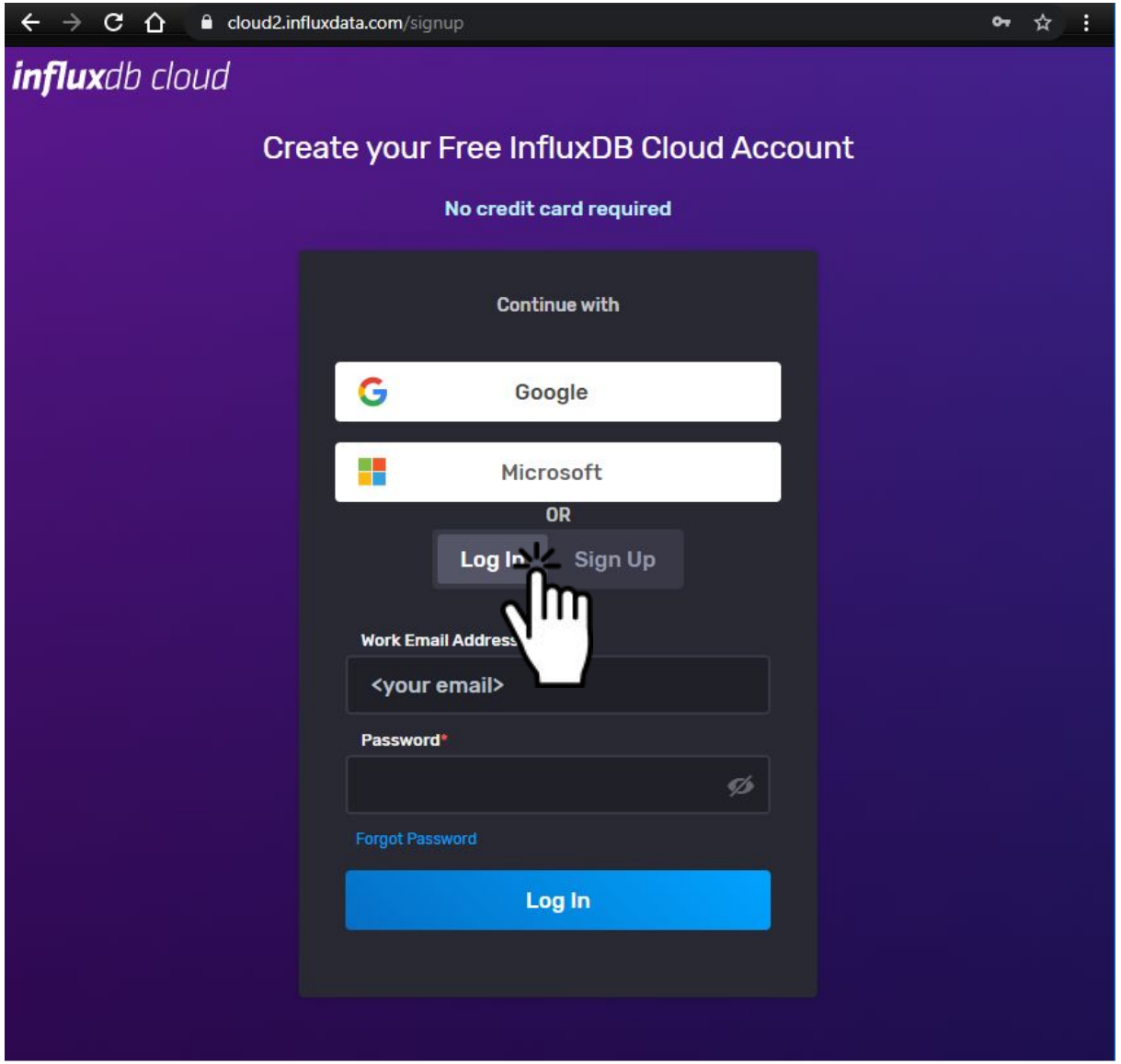
Open Web Browser
https://cloud2.influxdata.com/

Click to **Log In** option

Three options available:

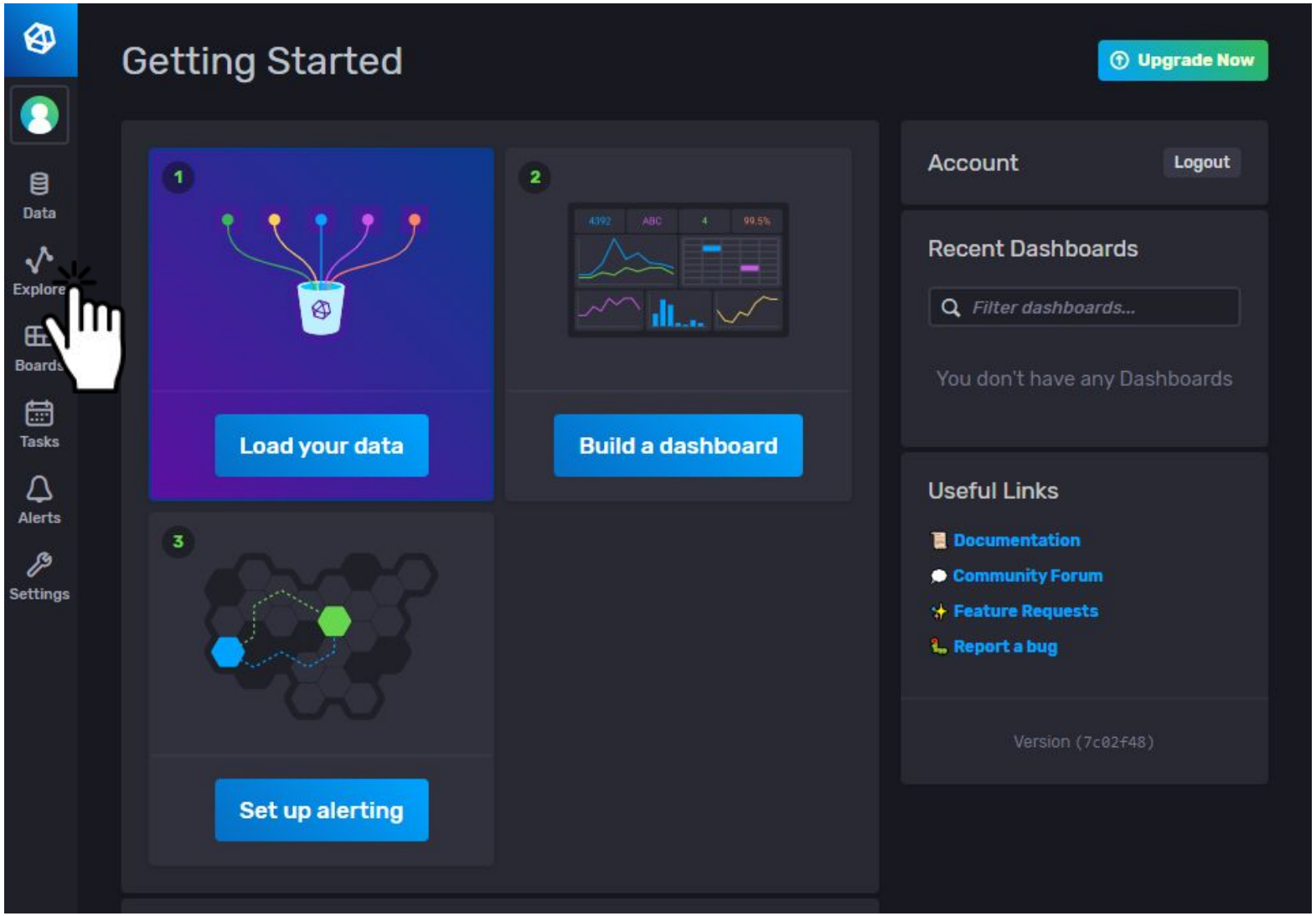
1. Google account
2. Microsoft account
3. Own email

Click to **Log In** option



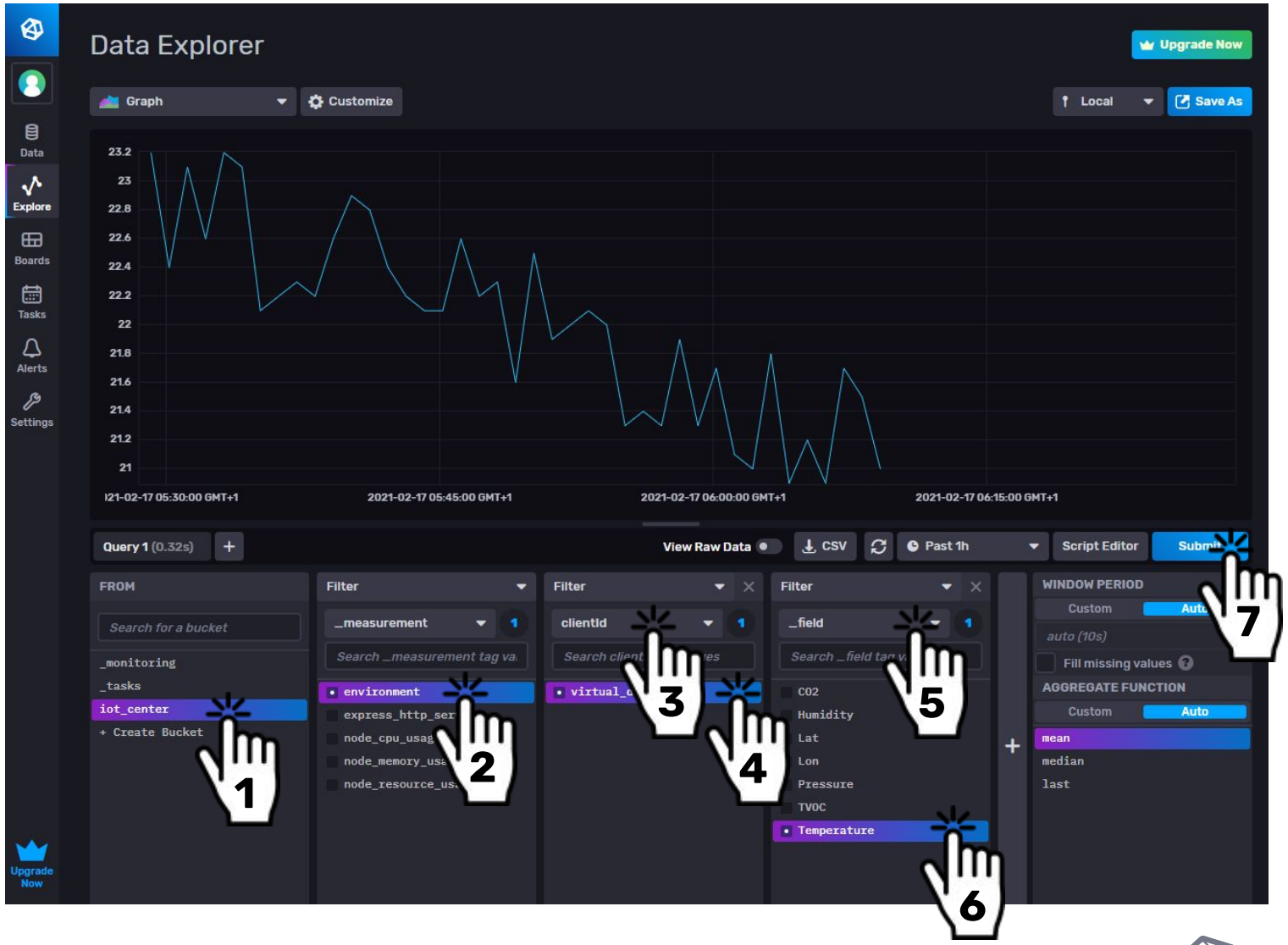
Start Explorer

Click on Explore Icon



Query InfluxDB

- Select bucket
  - **iot_centrum**
- Select measurement
  - **environment**
- Select field
  - **Temperature**
- Keep aggregation
  - **Mean**
- Set Time
  - **Past 1h**

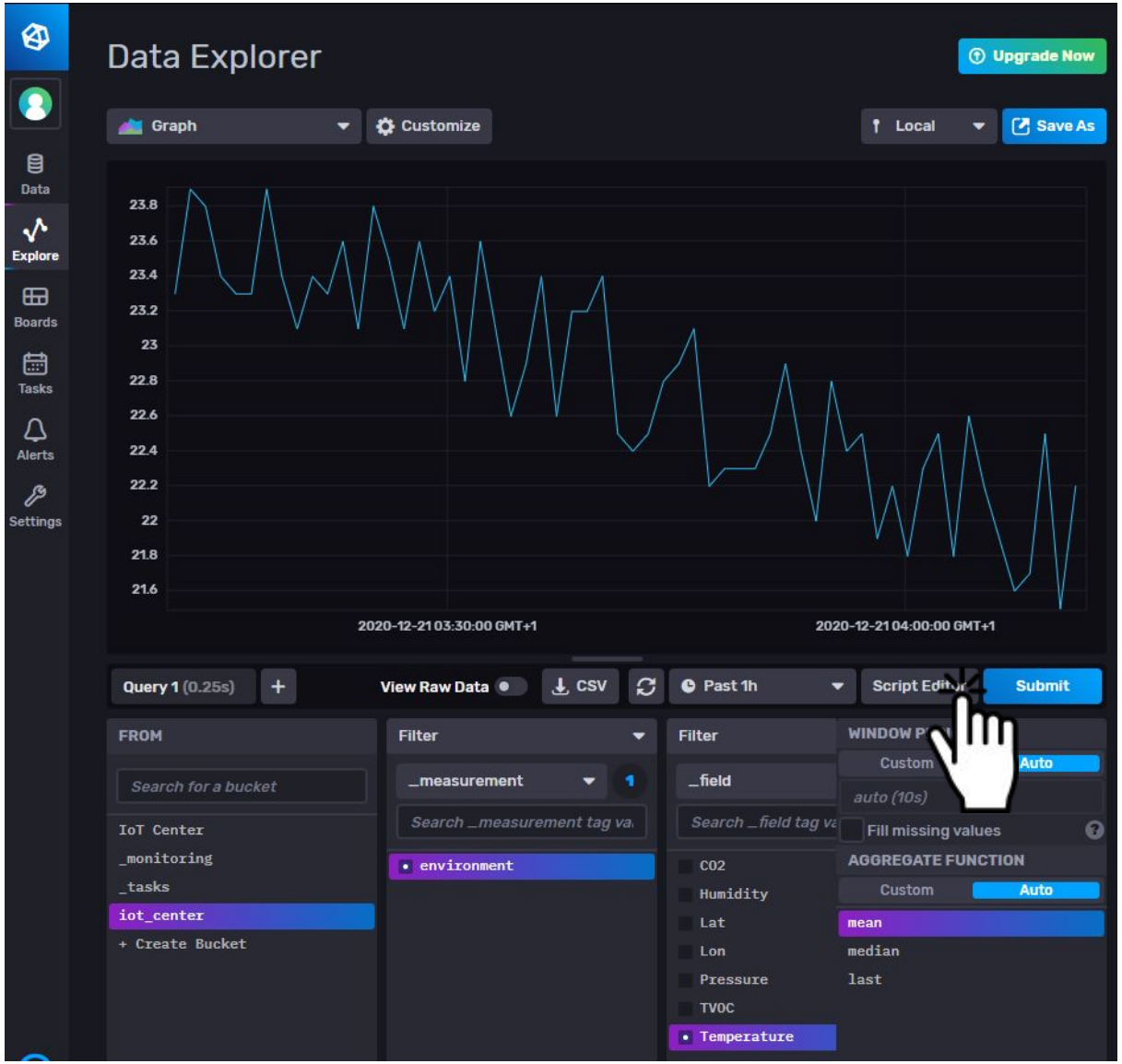


Switch to the Script Editor

Click to Script Editor Button

Show **Flux Query**

```
from(bucket: "iot_center")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "environment")
  |> filter(fn: (r) => r["clientId"] == "virtual_device")
  |> filter(fn: (r) => r["_field"] == "Temperature")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```



Basic Flux Query Structure

bucket
“database with retention policy”

range
time filter

v.<name>
variables generated by UI

_measurement
select specific “database table”

_field
select specific “table column”

tag *clientID*
filter specific “rows in the table”

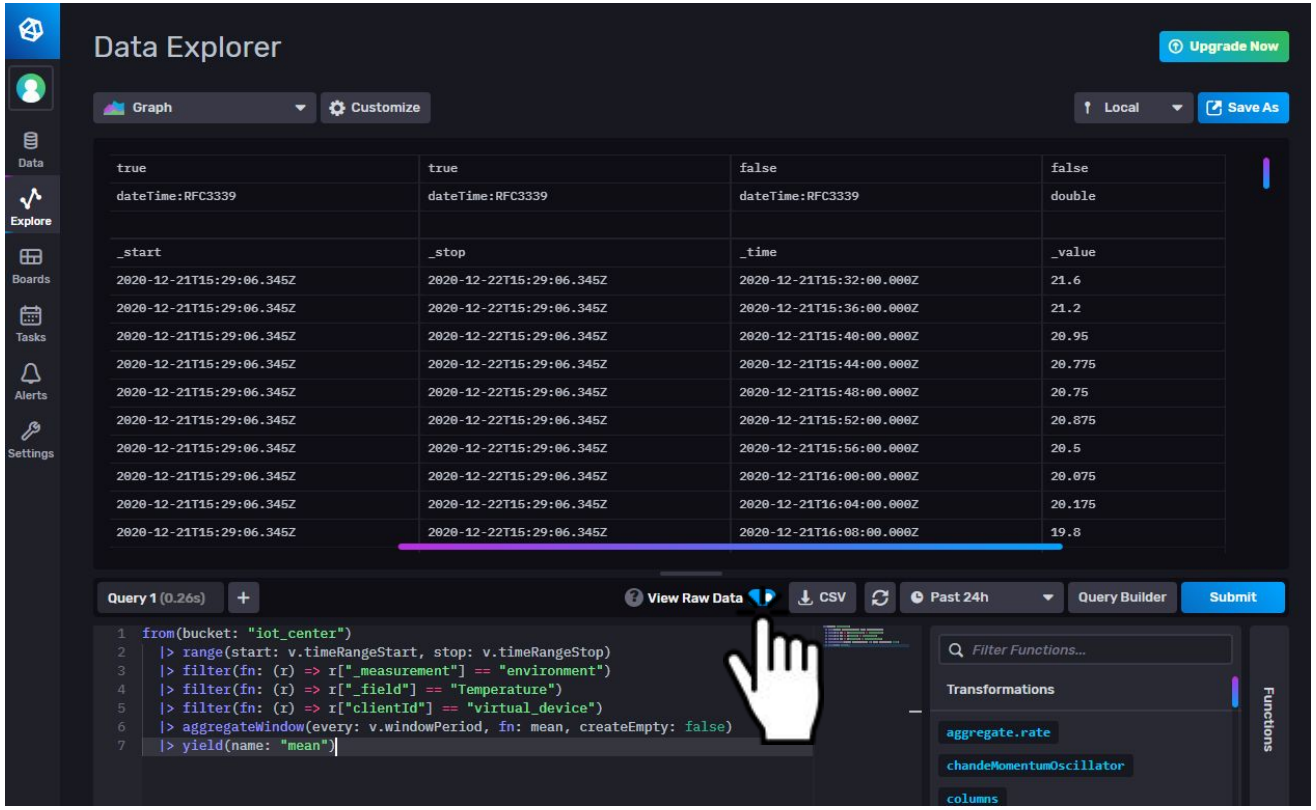
```
from(bucket: "iot_center")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "environment")
  |> filter(fn: (r) => r["_field"] == "Temperature")
  |> filter(fn: (r) => r["clientId"] == "virtual_device")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

aggregateWindow
group by time chunks defined by *v.windowPeriod* variable

Show Raw Query Data

Click to **View Raw Data**

- Table with all the columns



Task: How to Get GPS Coordinates?

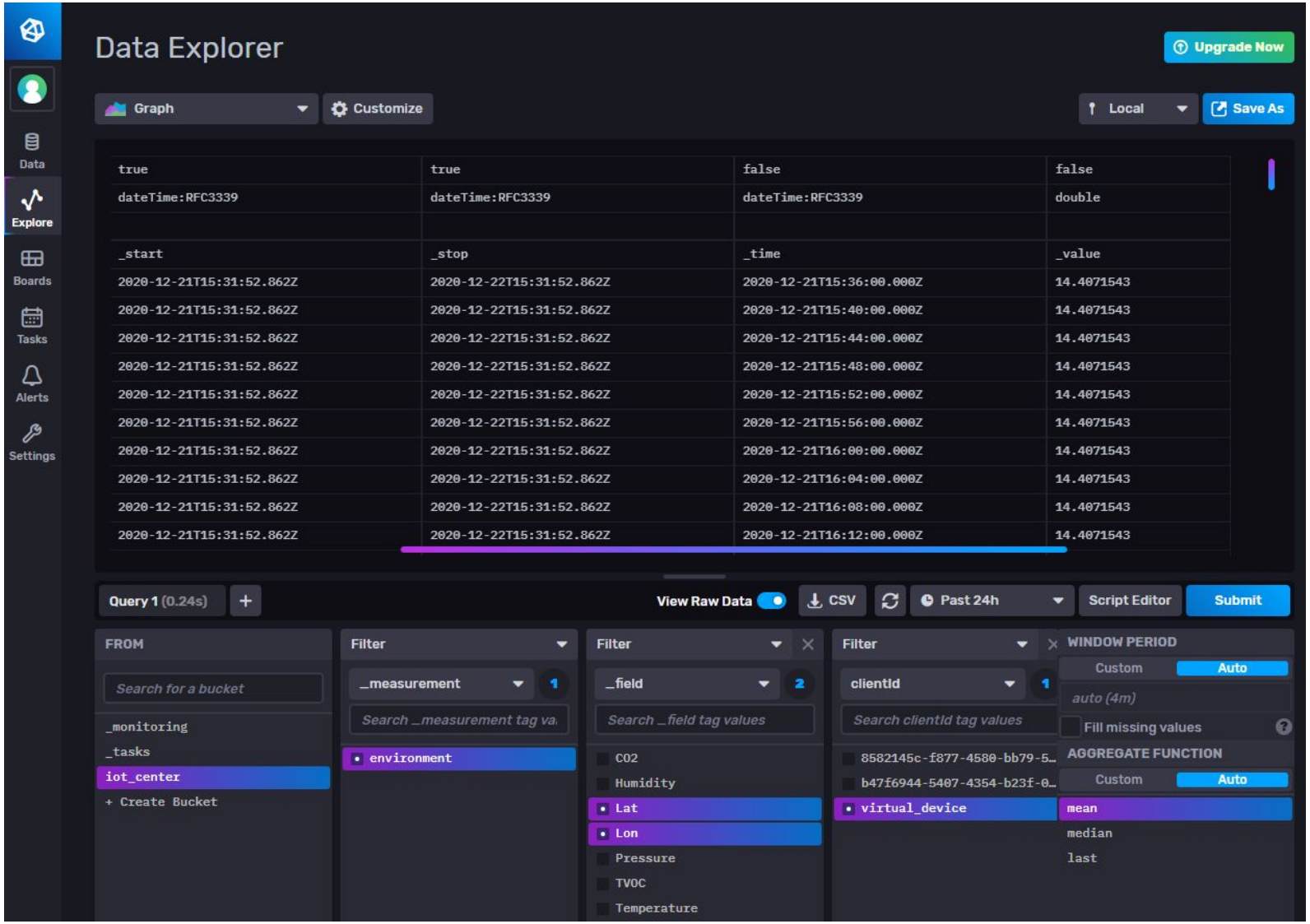
We need this query for the GEO widget

Virtual device provides GPS coordinates

- Fields **Lat** and **Lon**
- How to adjust the query?
- Two options

```
from(bucket: "iot_center")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "environment")
  |> filter(fn: (r) => r["_field"] == "Temperature")
  |> filter(fn: (r) => r["clientId"] == "virtual_device")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

1/2 Using Query Builder



2/2 Using Script Editor

Add Lat, Lon
select
“database columns”

```
from(bucket: "iot_center")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "environment")
  |> filter(fn: (r) => r["_field"] == "Lat" or r["_field"] == "Lon")
  |> filter(fn: (r) => r["clientId"] == "virtual_device")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

_start	_stop	_time	_value
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:36:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:40:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:44:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:48:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:52:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T15:56:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T16:00:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T16:04:00.000Z	14.4071543
2020-12-21T15:31:52.862Z	2020-12-22T15:31:52.862Z	2020-12-21T16:08:00.000Z	14.4071543

Documentation

More information:

https://docs.influxdata.com/influxdb/v2.0/query-data/get-started/