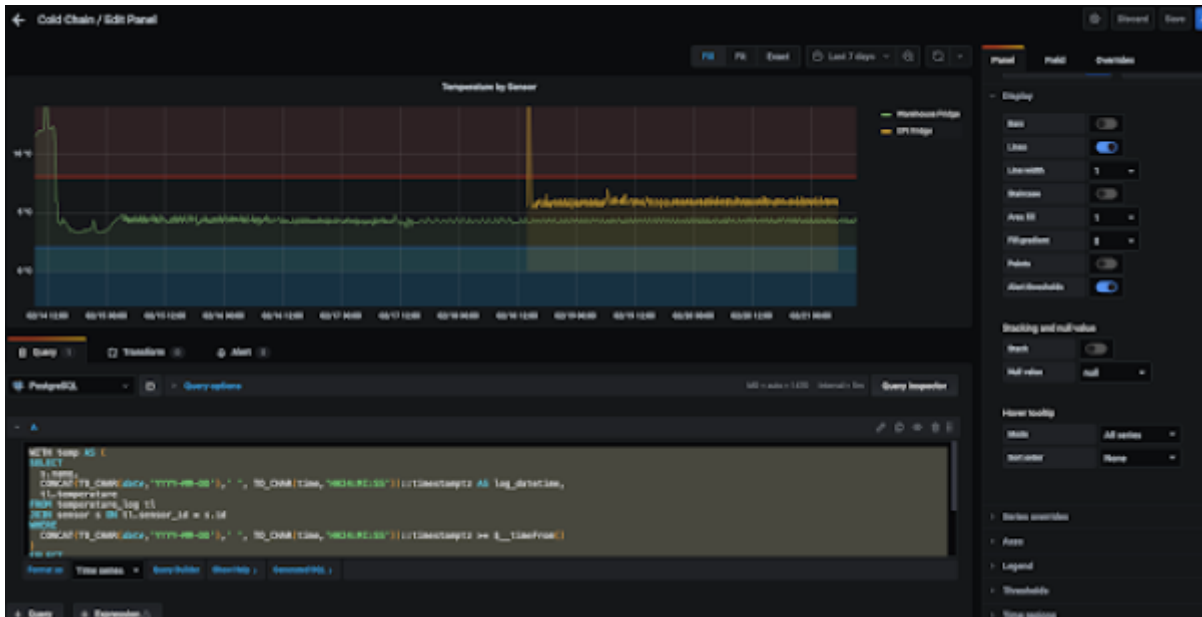


# Dashboard

These queries generally need to be set up by an administrator but have been included here as a reference to demonstrate some of the mSupply cold chain data that can be made available on the [mSupply Dashboard](#).

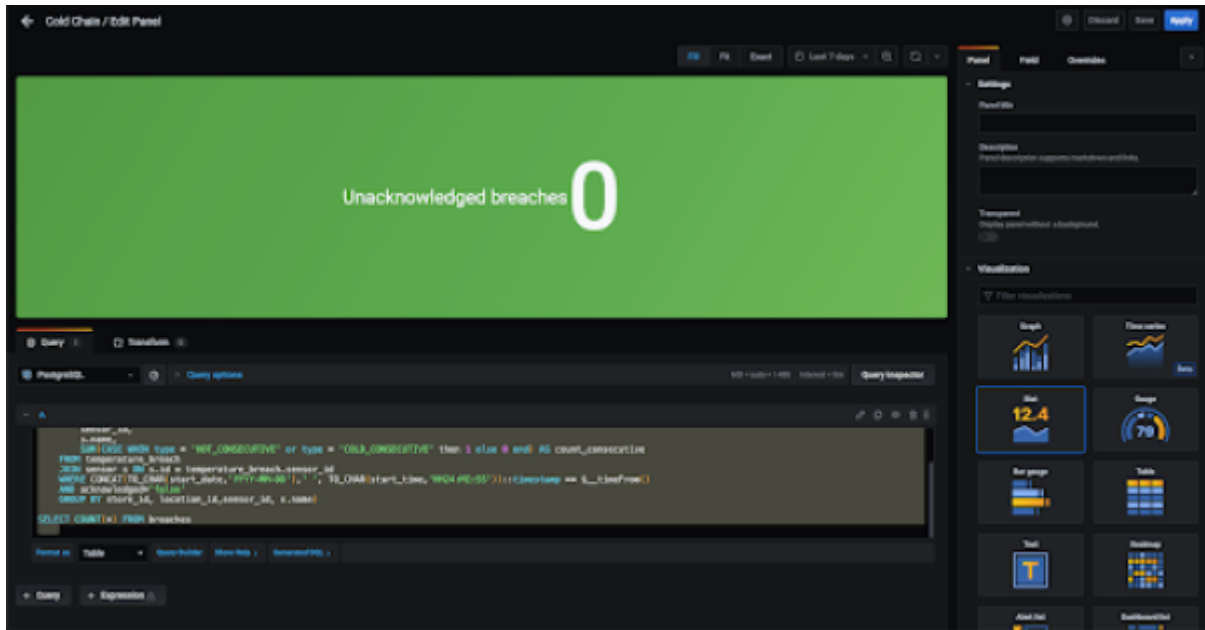
## Sensor Graph



## Query

```
WITH temp AS (
SELECT
  s.name,
  CONCAT(TO_CHAR(date,'YYYY-MM-DD'),' ',
TO_CHAR(time,'HH24:MI:SS'))::timestampz AS log_datetime,
  tl.temperature
FROM temperature_log tl
JOIN sensor s ON tl.sensor_id = s.id
WHERE
  CONCAT(TO_CHAR(date,'YYYY-MM-DD'),' ',
TO_CHAR(time,'HH24:MI:SS'))::timestampz >= $__timeFrom()
)
SELECT
  $__time(log_datetime),
  name,
  temperature
FROM temp
ORDER BY log_datetime
```

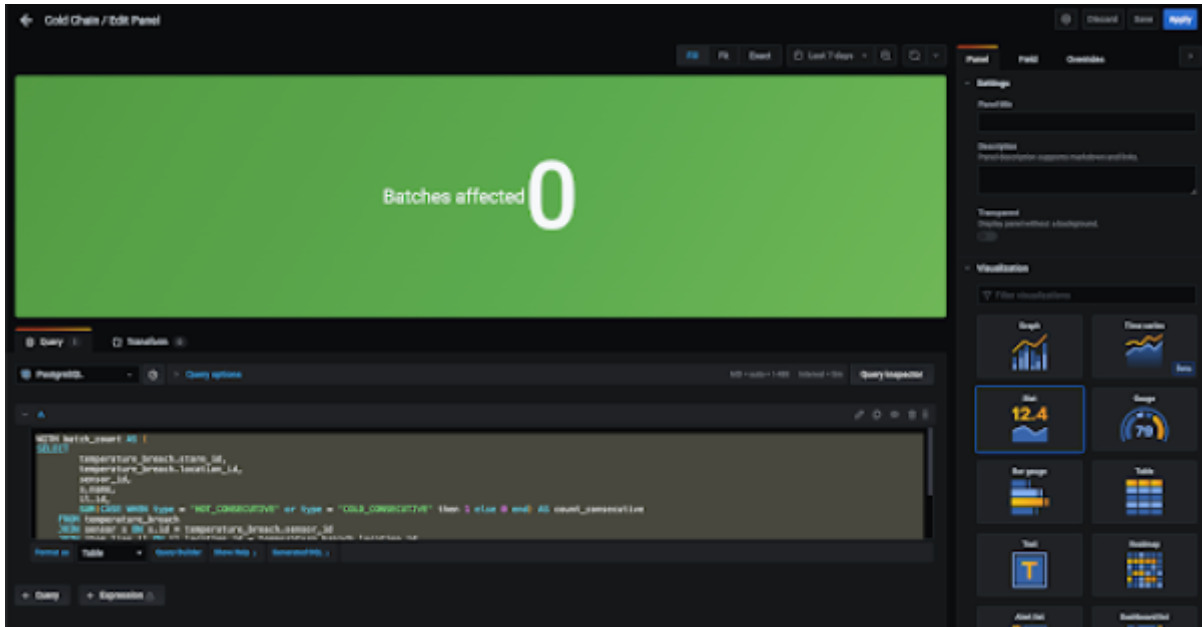
# Unacknowledged Breaches



## Query

```
WITH breaches AS (  
  SELECT  
    store_id,  
    location_id,  
    sensor_id,  
    s.name,  
    SUM(CASE WHEN type = 'HOT_CONSECUTIVE' or type = 'COLD_CONSECUTIVE'  
  then 1 else 0 end) AS count_consecutive  
  FROM temperature_breach  
  JOIN sensor s ON s.id = temperature_breach.sensor_id  
  WHERE CONCAT(TO_CHAR(start_date, 'YYYY-MM-DD'), ' ',  
  TO_CHAR(start_time, 'HH24:MI:SS'))::timestamp >= $__timeFrom()  
  AND acknowledged='false'  
  GROUP BY store_id, location_id, sensor_id, s.name)  
SELECT COUNT(*) FROM breaches
```

## Batches affected



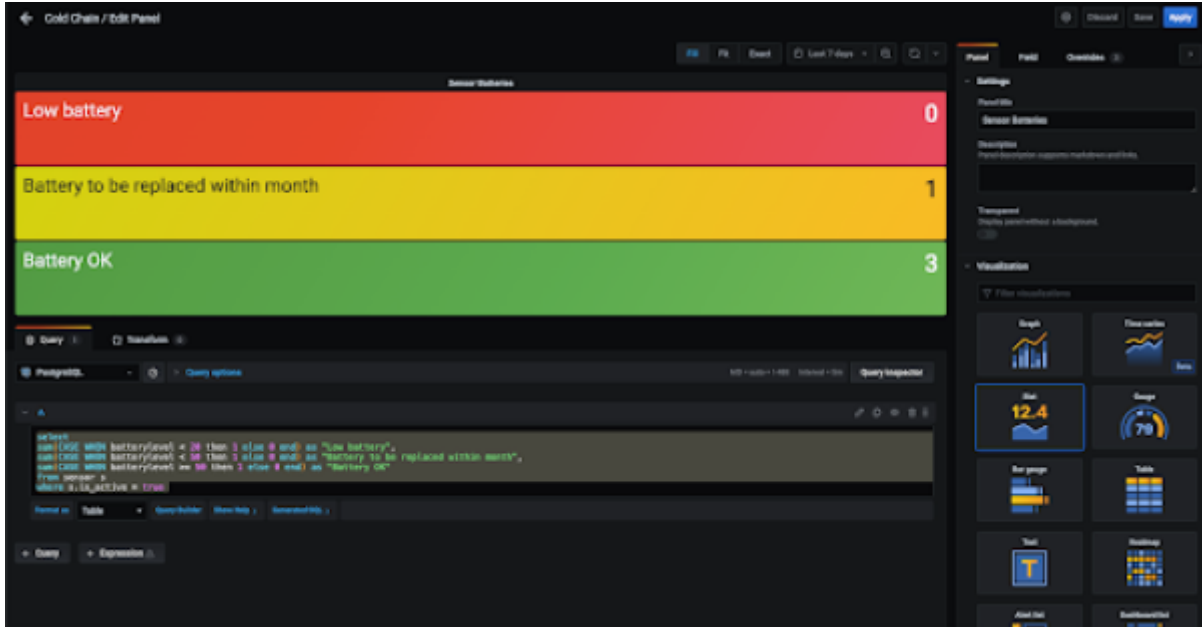
## Query

```

WITH batch_count AS (
SELECT
    temperature_breach.store_id,
    temperature_breach.location_id,
    sensor_id,
    s.name,
    il.id,
    SUM(CASE WHEN type = 'HOT_CONSECUTIVE' or type = 'COLD_CONSECUTIVE'
then 1 else 0 end) AS count_consecutive
FROM temperature_breach
JOIN sensor s ON s.id = temperature_breach.sensor_id
JOIN item_line il ON il.location_id = temperature_breach.location_id
WHERE CONCAT(TO_CHAR(start_date,'YYYY-MM-DD'),' ',
TO_CHAR(start_time,'HH24:MI:SS'))::timestamp >= $__timeFrom()
AND acknowledged='false'
GROUP BY temperature_breach.store_id,
temperature_breach.location_id,sensor_id, s.name, il.id
)
SELECT COUNT(*) FROM batch_count

```

## Sensor battery life



### Query

```
select
sum(CASE WHEN batterylevel < 20 then 1 else 0 end) as "Low battery",
sum(CASE WHEN batterylevel < 50 then 1 else 0 end) as "Battery to be
replaced within month",
sum(CASE WHEN batterylevel >= 50 then 1 else 0 end) as "Battery OK"
from sensor s
where s.is_active = true
```

From: <https://wiki.msupply.foundation/> - mSupply Foundation Documentation

Permanent link: [https://wiki.msupply.foundation/en:cold\\_chain:dashboard?rev=1653621179](https://wiki.msupply.foundation/en:cold_chain:dashboard?rev=1653621179)

Last update: 2022/05/27 03:12

