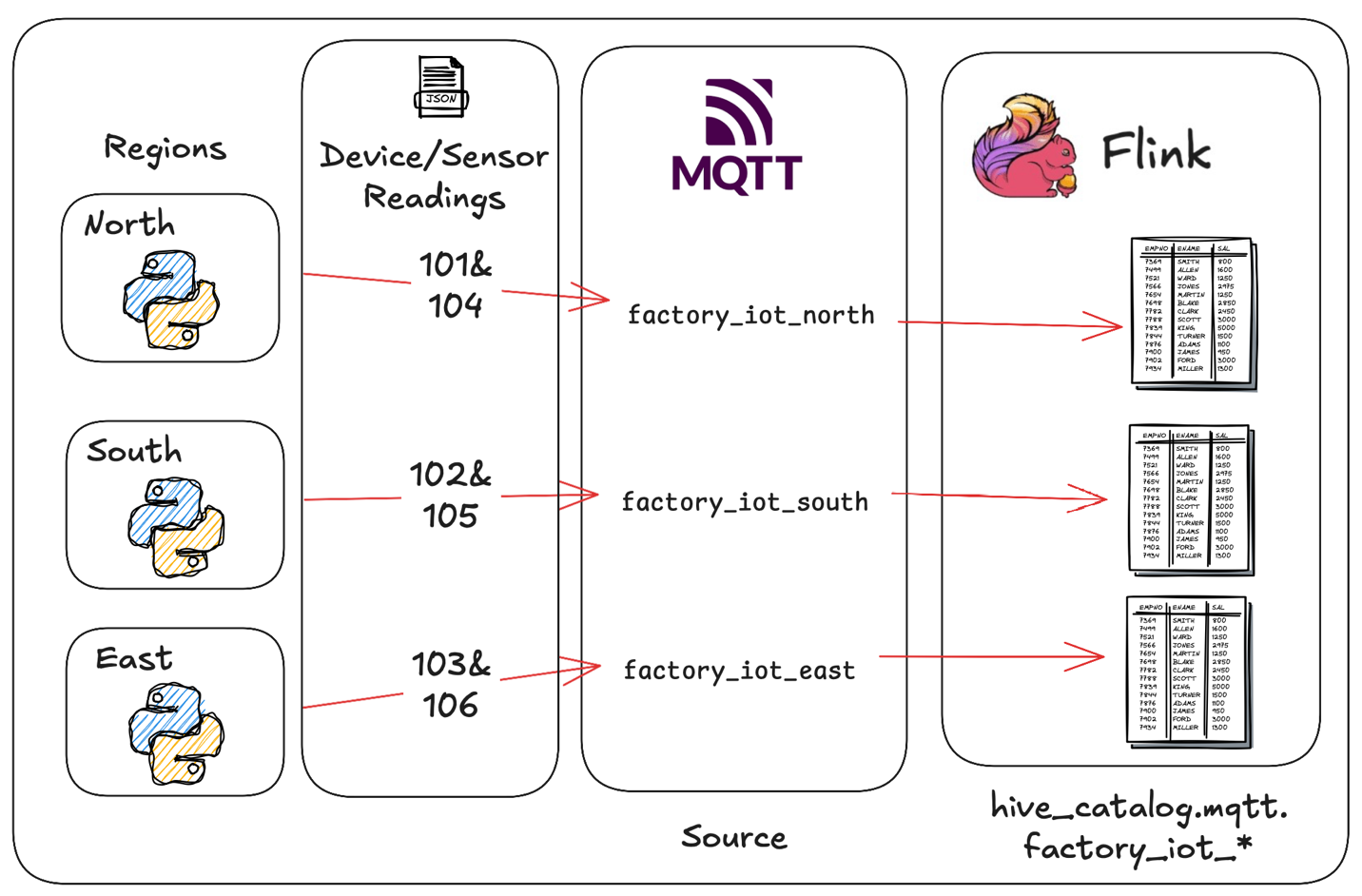
# Apache Flink MQTT Source Connector.

(16 June 2025)

# Overview

This all started with an accompanying blog, to be released shortly. The idea there was to publish IoT based metrics/readings onto MQTT Brokers, source it directly from Apache Flink and then sink into Fluss and Prometheus, to be visualized using Grafana.



Well, I simply could not find anything that I liked, that worked for me, and well, that was it. The hamster brain kicked into gear, and we wrote one.

Needless to say, I would not say it’s an enterprise level production level ready source connector, but it definitely shows what is possible…The code can do with allot more exception handling and error reporting.

My Example is primarily at this time based around my requirement:

* Apache Flink 1.20.1
* Java 17
* MQTT V3

But in the end, it won’t take too much to modify it for a different set of versions. I tried to add some error catching or make that simply log4J log output. All to be configured via your Apache Flink lof4j.properties file.

See: *log4j\_example* for well, an example…

[GIT Repo](https://github.com/georgelza/MQTT-Flink-Source-connector)

High Level Overview of an IoT payload:

{

timestamp,

metadata {

labels1

labels2

labels3

},

measurement

}

Below is my intended Full JSON payload.

{

"ts": 1707882120000,

"metadata": {

"siteId": 103,

"deviceId": 1014,

"sensorId": 10124,

"unit": "Amp",

"ts\_human": "2024-02-14T05:42:00.000000",

"location": {

"latitude": -33.9137,

"longitude": 25.5827

},

"deviceType": "Hoist\_Motor"

},

"measurement": 24

}

Next up, the Apache Flink Table SQL:

CREATE OR REPLACE TABLE hive\_catalog.mqtt.factory\_iot\_101 (

ts BIGINT,

metadata ROW<

siteId INTEGER,

deviceId INTEGER,

sensorId INTEGER,

unit STRING,

ts\_human STRING,

location ROW<

latitude DOUBLE,

longitude DOUBLE>,

deviceType STRING>,

measurement DOUBLE,

ts\_WM AS TO\_TIMESTAMP(FROM\_UNIXTIME(CAST(`ts` AS BIGINT) / 1000)),

WATERMARK FOR ts\_WM AS ts\_WM

) WITH (

'connector' = 'mqtt'

,'broker.host' = 'broker\_north' -- Example: your MQTT broker's hostname or IP

,'broker.port' = '1883' -- Example: your MQTT broker's port (1883 for non-SSL)

,'topic' = 'factory\_iot/north/101'

,'format' = 'json'

,'client.id' = 'flink\_iot\_north\_101\_consumer'

,'qos' = '1' -- QoS, 0, 1 or 2

,'username' = 'mqtt\_dev'

,'password' = 'abfr24'

,'automatic-reconnect' = 'true' -- Valid boolean value ('true' or 'false')

,'clean-session' = 'true' -- Valid boolean value ('true' or 'false')

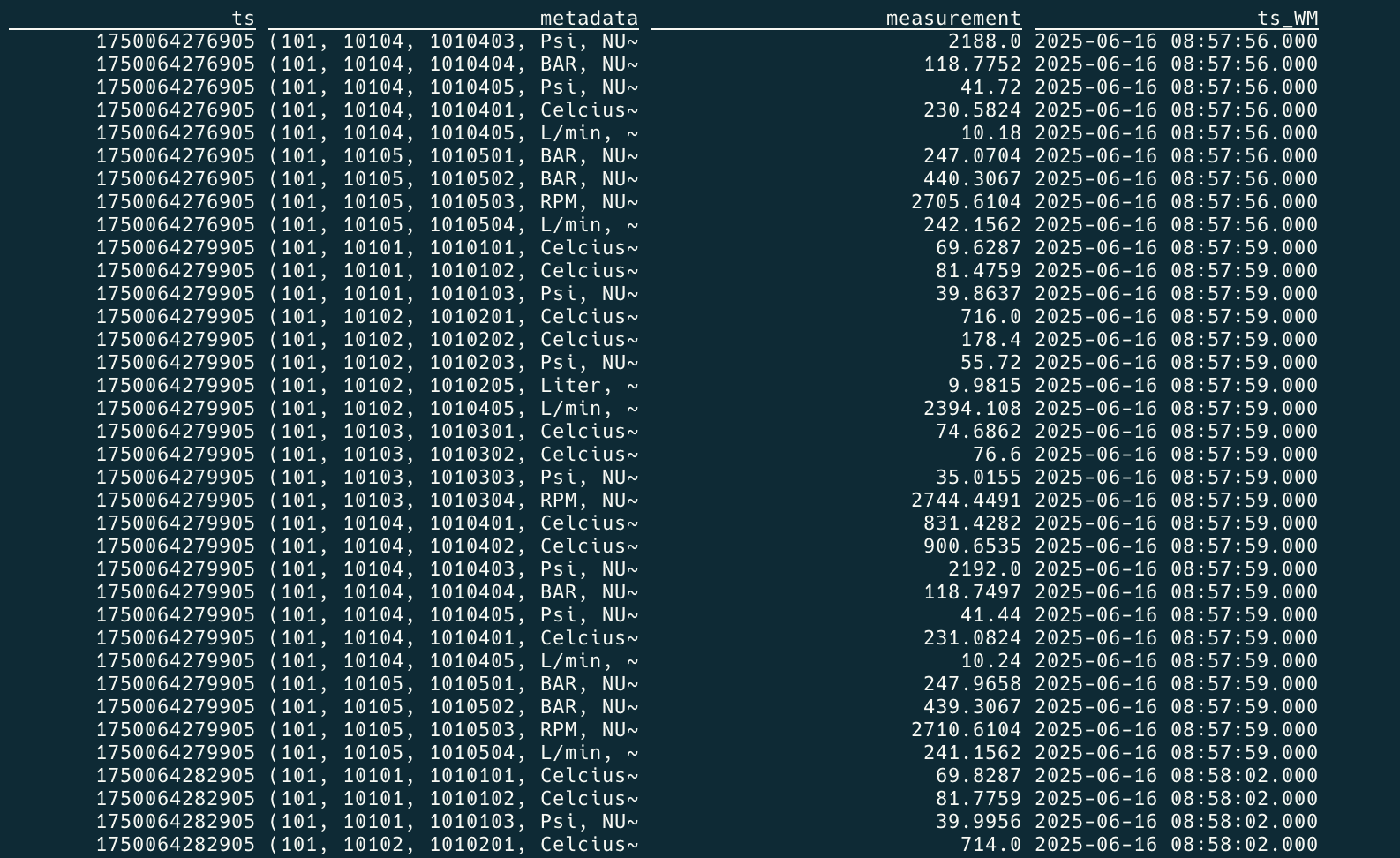
,'ssl' = 'false' -- Set to 'true' if using SSL and configure below

);

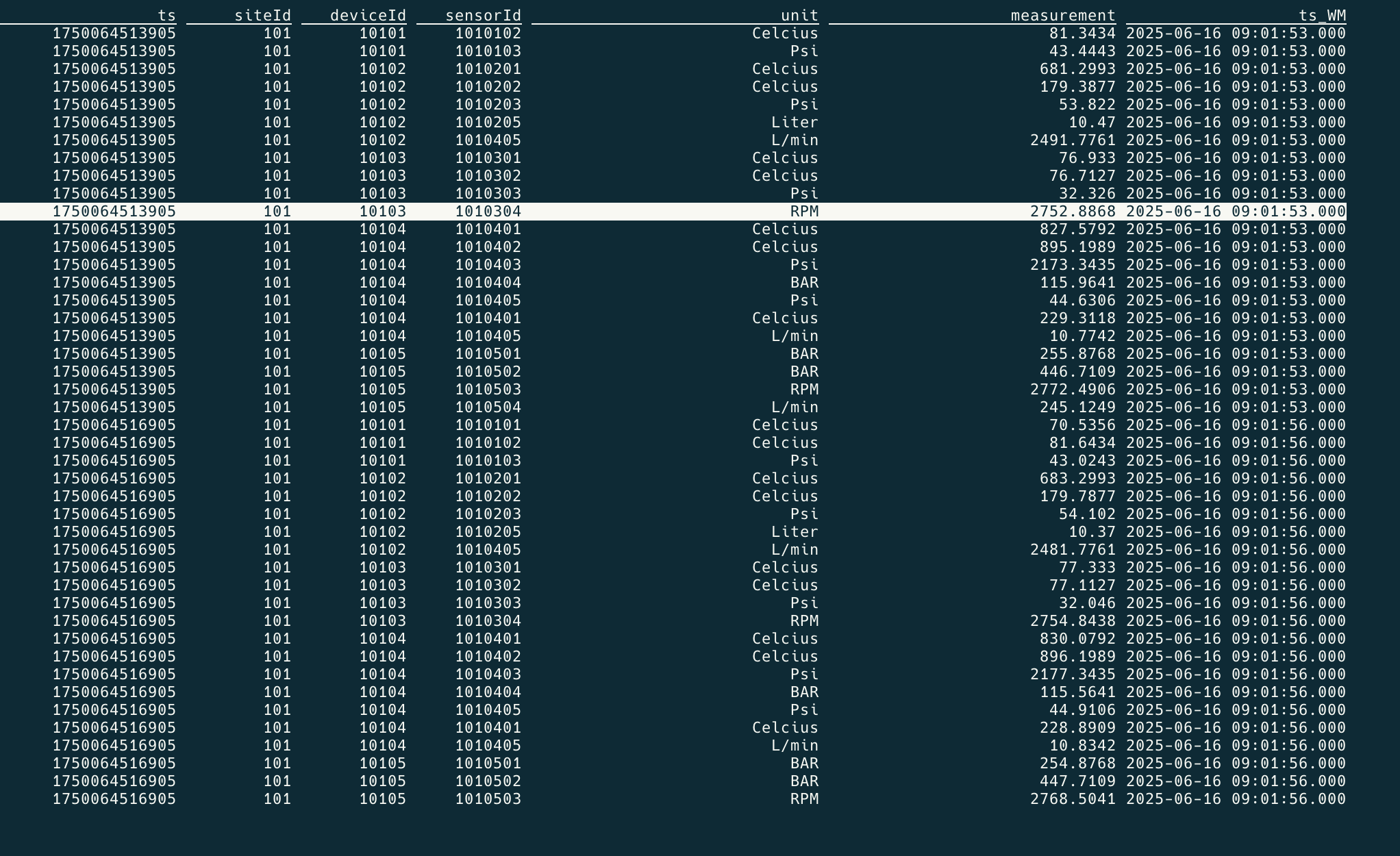
We can now execute:

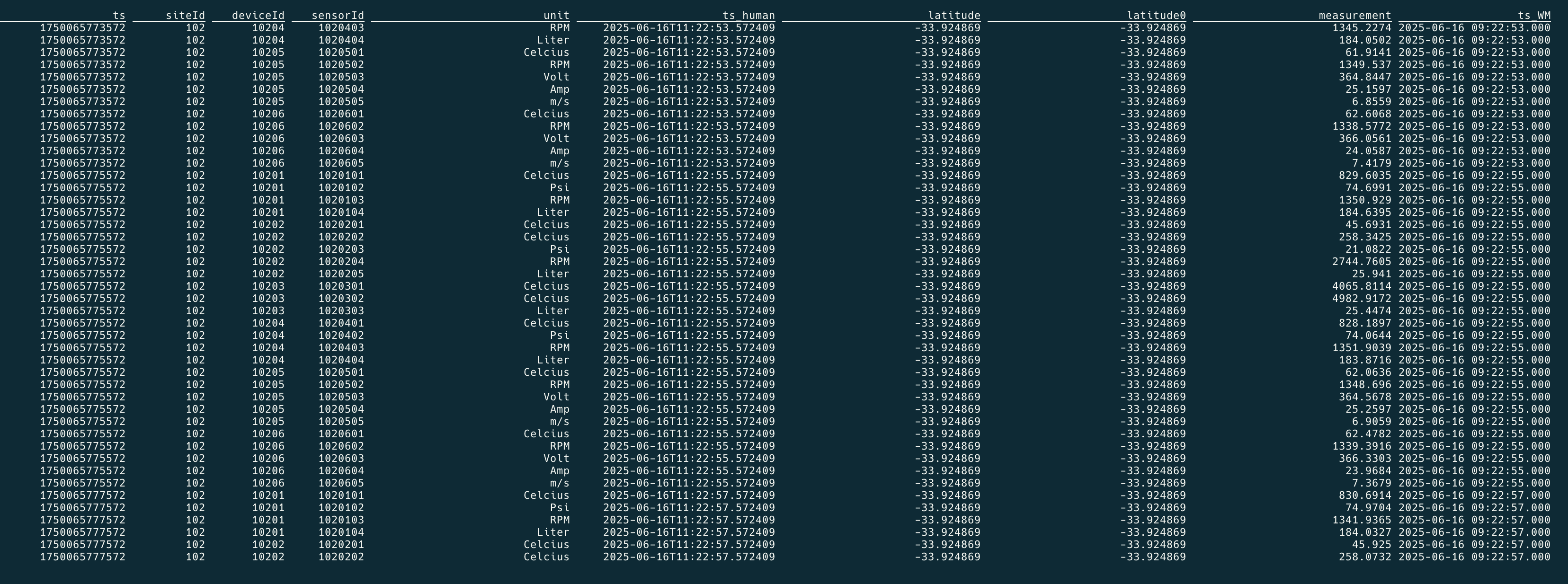
*select \* from factory\_iot\_101*;

This will be executed via the Flink SQL (*bin/sql-client.sh)* client.



Followed by a more definitive query, targeting specific columns in our metadata tag.







# And In Summary.

Now… that was fun… Got to say thank you to ChatGP (not) and definitely Gemini…

But in the end, they gave general direction, with still allow of work required by the biological… ;)

And all of this led to an idea for another Blog ;) Keep watching this space.

As always, I’m predictable, but I really do think this is pretty neat… Hope you enjoyed the exploration.

Good luck, this is all fraught with rabbit holes, as always, so many and you can disappear so easily… but then that’s ½ the fun.



*Note: to execute this blog start with README.md located in the root folder and work from there, it will tell you exactly what to execute in which order to download all the dependencies and build everything. If you have any problems, welcome to reach out to me via one of the below profiles.*

**About Me**

I’m a techie, a technologist, always curious, love data, have for as long as I can remember always worked with data in one form or the other, Database admin, Database product lead, data platforms architect, infrastructure architect hosting databases, backing it up, optimizing performance, accessing it. Data data data… it makes the world go round.

In recent years, pivoted into a more generic Technology Architect role, capable of full stack architecture.

[George Leonard](https://www.linkedin.com/in/george-leonard-945b502/)

[georgelza@gmail.com](mailto:georgelza@gmail.com)

<https://www.linkedin.com/in/george-leonard-945b502/>

<https://medium.com/@georgelza>