

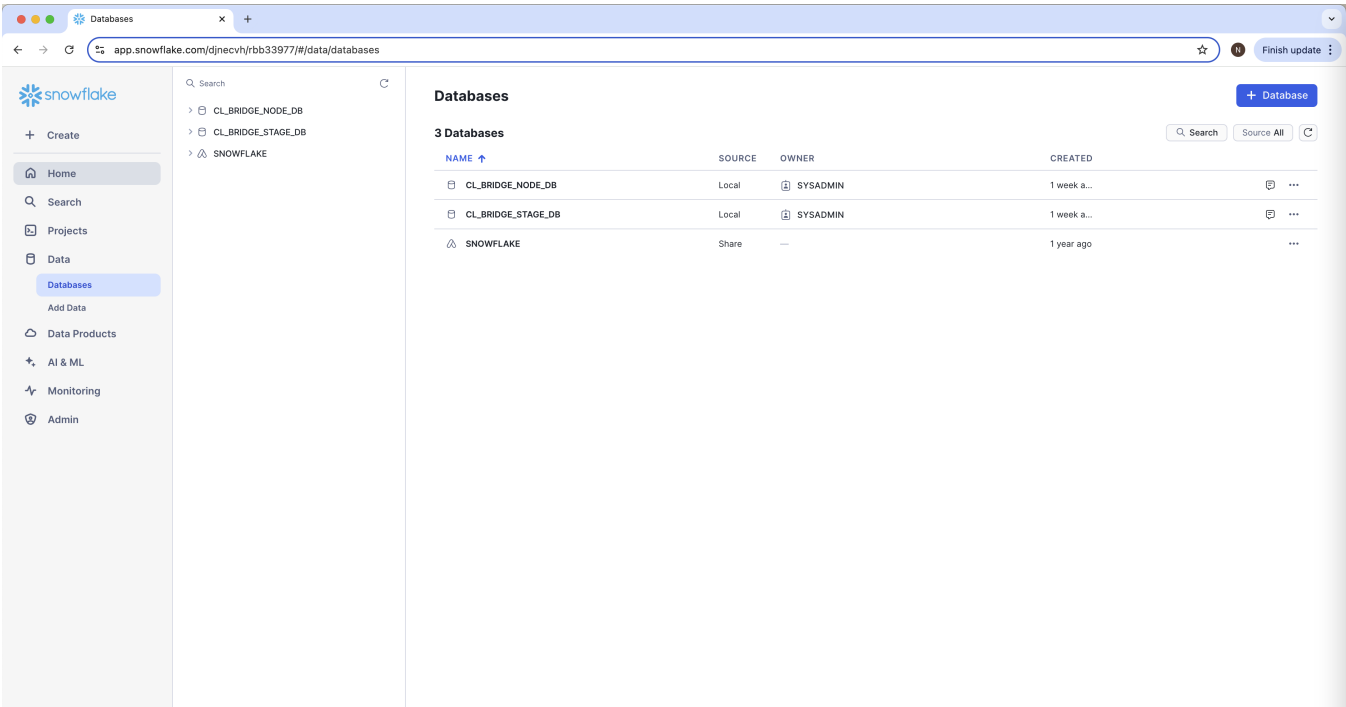
IBSNOW: Viewing the data in Snowflake

**New IoT Bridge for Snowflake Available**

This document covers v2.0.0 and older versions of IoT Bridge for Snowflake. It now ships as part of Chariot. To see the latest IoT Bridge for Snowflake documentation, go to [IoT Bridge for Snowflake in Chariot](#).

Data can be view in the databases created by the [Snowflake Setup Scripts](#) during your initial setup:

- Staging database created by SQL Script 01 (e.g., CL_BRIDGE_STAGE_DB)
- Node database created by SQL Script 02 (e.g., CL_BRIDGE_NODE_DB)



Terminology

As the data moves from the Ignition Edge device through MQTT Sparkplug and into Snowflake different terminology will be used with the equivalence shown below:

Ignition Edge	MQTT Sparkplug	Snowflake	Snowflake tables/views
Ignition UDT Definition	Sparkplug Template	Model	Template_Reference
Ignition UDT Instance	Sparkplug Template Instance	Model Instance	Device_Name

Staging Database

The Staging database contains only a single table, SPARKPLUG_RAW. This table contains information derived from each Sparkplug message, like the Sparkplug IDs: GroupId, EdgenodeId and DeviceId, the message timestamp, the model / model instance, the raw message payload in JSON, etc.

The table is populated by the Snowflake Snowpipe Streaming API as data is streamed in by the IoT bridge.

The screenshot shows the Snowflake SQL interface. On the left, the 'Databases' sidebar is expanded to 'CL_BRIDGE_STAGE_DB', then 'PUBLIC', then 'STAGE_DB', and finally 'Tables', where 'SPARKPLUG_RAW' is selected. The main query editor contains the following SQL:

```
1 select
2   MSG_TOPIC,
3   NAMESPACE,
4   GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
5   MESSAGE_TYPE,
6   MSG,
7   INSERTED_AT
8 from CL_BRIDGE_STAGE_DB.STAGE_DB.SPARKPLUG_RAW
9 where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1'
```

The 'Results' tab shows 2 rows of data:

MSG_TOPIC	NAMESPACE	GROUP_ID	EDGE_NODE_ID	DEVICE_ID	MESSAGE_TYPE	MSG	INSERTED_AT
spBv1.0/G1/DBIRTH/E1/PLC 1	spBv1.0	G1	E1	PLC 1	DBIRTH	{ "metrics": [{ "dataType": "Template", "metaData": { "contentType": "",	1730848515389
spBv1.0/G1/NBIRTH/E1	spBv1.0	G1	E1	null	NBIRTH	{ "metrics": [{ "dataType": "Boolean", "name": "Node Control/Next Server",	1730848511331

On the right, 'Query Details' shows a duration of 139ms and 2 rows. Below the results, there are expandable sections for 'MSG_TOPIC', 'NAMESPACE', and 'GROUP_ID', each showing '100% filled'. An 'Ask Copilot' button is visible at the bottom right.

Node Database

The Node database also contains only a single table, SPARKPLUG_DEVICE_MESSAGES, but also has additional views, user defined functions, store procedures, streams, etc.

The SPARKPLUG_DEVICE_MESSAGES table contains partially processed data where each row in the table is a single message containing data for each model instance (aka. Device Name). This data is sourced from SPARKPLUG_RAW table and is inserted by the synch_device_messages() stored procedure. The synch_device_messages() stored procedure is executed on demand by the IoT Bridge for Snowflake.

The screenshot shows the 'IoT Bridge - Snowflake' application interface. The browser address bar shows 'app.snowflake.com/djnevvh/bb33977/w13NdpK5Gg#query'. The left sidebar shows the 'Databases' sidebar expanded to 'CL_BRIDGE_STAGE_DB', then 'Tables', where 'SPARKPLUG_DEVICE_MESSAGES' is selected. The main query editor contains the following SQL:

```
1 select
2   GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
3   DEVICE_NAME,
4   TEMPLATE_REFERENCE,
5   -- ROOT_MESSAGE_TIMESTAMP,
6   DEVICE_METRIC_TIMESTAMP,
7   DDATA_MSG,
8   MEASURES_INFO,
9   -- MEASURE_TIMESTAMP,
10  -- MEASURE_TS,
11  -- MESSAGE_TYPE,
12  INSERTED_AT
13 from CL_BRIDGE_STAGE_DB.STAGE_DB.SPARKPLUG_DEVICE_MESSAGES
14 where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1'
```

The 'Results' tab shows 13 rows of data:

GROUP	EDGE	DEVICE_ID	DEVICE_NAME	TEMPLATE_REFERENCE	DEVICE_METRIC_TIMESTAMP	DDATA_MSG	MEASURES_INFO
G1	E1	PLC 1	TestMotor	Motor	1730316124378	{ "dataType": "Template", "meta	{ "Amps": 121
G1	E1	PLC 1	Line 1	Assembly Line	1730316159611	{ "dataType": "Template", "meta	{ "State/Current State": 87.56264820205526
G1	E1	PLC 1	Line 1	Assembly Line	1730316700173	{ "dataType": "Template", "meta	{ "Motor 1": { "isDefinition": false, "metrics": 17
G1	E1	PLC 1	Line 1	Assembly Line	1730848510872	{ "dataType": "Template", "meta	{ "Motor 2": { "isDefinition": false, "metrics": 17
G1	E1	PLC 1	Line 1	Assembly Line	1730318408732	{ "dataType": "Template", "meta	{ "Motor 1": { "isDefinition": false, "metrics": 17
G1	E1	PLC 1	Line 1	Assembly Line	1730318054483	{ "dataType": "Template", "meta	{ "State/Last Error": 100232
G1	E1	PLC 1	Line 1	Assembly Line	1730316159611	{ "dataType": "Template", "meta	{ "Motor 1": { "isDefinition": false, "metrics": 17
G1	E1	PLC 1	Line 1	Assembly Line	1730318408732	{ "dataType": "Template", "meta	{ "Speed": 10
G1	E1	PLC 1	Line 1	Assembly Line	1730316700173	{ "dataType": "Template", "meta	{ "State/Last Error": 100232
G1	E1	PLC 1	Line 1	Assembly Line	1730316159611	{ "dataType": "Template", "meta	{ "Speed": 10
G1	E1	PLC 1	Line 1	Assembly Line	1730318054483	{ "dataType": "Template", "meta	{ "Speed": 10
G1	E1	PLC 1	TestMotor	Motor	1730316159611	{ "dataType": "Template", "meta	{ "State/Last Error": 0
G1	E1	PLC 1	Line 1	Assembly Line	1730324917649	{ "dataType": "Template", "meta	{ "Speed": 10

On the right, there is a 'Query Details' section showing 'Query duration: 139ms' and 'Rows: 2'. Below this, there are expandable sections for 'MSG_TOPIC', 'NAMESPACE', and 'GROUP_ID', each showing '100% filled'. An 'Ask Copilot' button is visible at the bottom right.

The screenshot shows the IoT Bridge for Snowflake web application. The interface includes a sidebar with a database tree, a central SQL editor, and a results pane at the bottom.

Database Tree (Left Sidebar):

- CL_BRIDGE_NODE_DB
 - INFORMATION_SCHEMA
 - NB_G1_E1
 - Views
 - PUBLIC
 - STAGE_DB
 - Tables
 - SPARKPLUS_DEVICE_MESSAGES
 - Views
 - Streams
 - Functions
 - Procedures
 - Dynamic Tables

SQL Editor (Center):

```

1 select
2   GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
3   DEVICE_NAME,
4   TEMPLATE_REFERENCE,
5   -- ROOT_MESSAGE_TIMESTAMP,
6   DEVICE_METRIC_TIMESTAMP,
7   DDATA_MSG,
8   MEASURES_INFO,
9   -- MEASURE_TIMESTAMP,
10  -- MEASURE_TS
11  -- MESSAGE_TYPE,
12  INSERTED_AT
13 from CL_BRIDGE_NODE_DB.STAGE_DB.SPARKPLUS_DEVICE_MESSAGES
14 where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1' and TEMPLATE_REFERENCE = 'Motor' and DEVICE_NAME = 'Motor 3'
  
```

Results Pane (Bottom):

	GROUP_ID	EDGE_NODE_ID	DEVICE_ID	DEVICE_NAME	TEMPLATE_REFERENCE	DEVICE_METRIC_TIMESTAMP	DDATA_MSG	MEASURES_INFO	INSERTED_AT
1	G1	E1	PLC 1	Motor 3	Motor	1730848510872	{ "dataType": "Templ...	{ "RPMs": 1202}	1730848515389
2	G1	E1	PLC 1	Motor 3	Motor	1730848510872	{ "dataType": "Templ...	{ "Temperature": 49}	1730848515389
3	G1	E1	PLC 1	Motor 3	Motor	1730848510872	{ "dataType": "Templ...	{ "State/Last Error": 0}	1730848515389
4	G1	E1	PLC 1	Motor 3	Motor	1730848510872	{ "dataType": "Templ...	{ "State/Current State": 0}	1730848515389
5	G1	E1	PLC 1	Motor 3	Motor	1730848510872	{ "dataType": "Templ...	{ "Amps": 12.1}	1730848515389

Query Details (Right):

- Query duration: 110ms
- Rows: 5
- Query ID: 07b42482-0004-2075...
- Show more
- Filters: GROUP_ID (G1), EDGE_NODE_ID (E1), DEVICE_ID (PLC 1)
- Ask Copilot button

The IoT Bridge for Snowflake will automatically create a new schema in its "Node" database for every Sparkplug edge node it finds. The Bridge will then create Views within this schema for each model found in the Sparkplug edge node data ingested via the bridge . There are three views per model:

Base View

This view contains the model instance tag data where each row is a single tag change per model instance. Tag change values are still encoded in JSON in the MEASURES_INFO and DDATA_MSG columns.

- Example: CL_BRIDGE_NODE_DB.NB_GROUP_EDGE.MOTOR

The screenshot shows the IoT Bridge for Snowflake interface. On the left, the 'Databases' tab is active, showing a tree view of the database structure. The 'MOTOR_VW' view is selected. In the center, a SQL query is entered in the editor:

```
select
  GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
  DEVICE_NAME,
  MEASURE_TIMESTAMP,
  MEASURES_INFO,
  INSERTED_AT
from CL_BRIDGE_NODE_DB.NB_G1_E1.MOTOR_VW
where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1' and DEVICE_NAME = 'Motor 3'
```

Below the query editor, the 'Results' tab is active, displaying a table with 5 rows and 7 columns. The columns are: GROUP_ID, EDGE_NODE_ID, DEVICE_ID, DEVICE_NAME, MEASURE_TIMESTAMP, MEASURES_INFO, and INSERTED_AT. The data is as follows:

GROUP_ID	EDGE_NODE_ID	DEVICE_ID	DEVICE_NAME	MEASURE_TIMESTAMP	MEASURES_INFO	INSERTED_AT
G1	E1	PLC 1	Motor 3	1730848316753	{ "State/Last Error": 0 }	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	{ "Temperature": 49 }	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	{ "State/Current State": 0 }	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	{ "Amps": 121 }	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	{ "RPMs": 1202 }	1730848515389

On the right side, the 'Query Details' panel shows the query duration as 70ms and the query ID as 07b42495-0004-2025-...

Intermediate View

This view contains the model instance tag data where each row is a single tag change per model instance. In this view, each model instance tag is separated out into its own column with their values. This view will show each individual model instance tag change and the tags that did not changed will be marked as null.

- Example: CL_BRIDGE_NODE_DB.NB_GROUP_EDGE.MOTOR_VW

The screenshot shows the IoT Bridge for Snowflake interface. On the left, the 'Databases' tab is active, showing a tree view of the database structure. The 'MOTOR_VW' view is selected. In the center, a SQL query is entered in the editor:

```
select
  GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
  DEVICE_NAME,
  MEASURE_TIMESTAMP,
  TEMPERATURE,
  RPMs,
  AMPS,
  INSERTED_AT
from CL_BRIDGE_NODE_DB.NB_G1_E1.MOTOR_VW
where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1' and DEVICE_NAME = 'Motor 3'
```

Below the query editor, the 'Results' tab is active, displaying a table with 5 rows and 9 columns. The columns are: GROUP_ID, EDGE_NODE_ID, DEVICE_ID, DEVICE_NAME, MEASURE_TIMESTAMP, TEMPERATURE, RPMs, AMPS, and INSERTED_AT. The data is as follows:

GROUP_ID	EDGE_NODE_ID	DEVICE_ID	DEVICE_NAME	MEASURE_TIMESTAMP	TEMPERATURE	RPMs	AMPS	INSERTED_AT
G1	E1	PLC 1	Motor 3	1730848316753	null	null	null	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	null	null	null	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	null	null	121	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	49	null	null	1730848515389
G1	E1	PLC 1	Motor 3	1730848316753	null	1202	null	1730848515389

On the right side, the 'Query Details' panel shows the query duration as 40ms and the query ID as 07b42497-0004-2025-...

Asof View

This view is basically the same as the Intermediate view - each model instance tag is separated out into its own column. However, each row in this view has the null values from the Intermediate views removed (unless the tag change really had a null value) and the last known good value for the model instance tag will be replicated in its column. Said another way, each row will have the last known good values for all model instance tags.

- Example: CL_BRIDGE_NODE_DB.NB_GROUP_EDGE.MOTOR_ASOF_VW

The screenshot displays the IoT Bridge for Snowflake web application. The interface includes a sidebar with a tree view of the database schema, a central query editor, and a results pane at the bottom.

Database Schema (Left Sidebar):

- CL_BRIDGE_NODE_DB
 - INFORMATION_SCHEMA
 - NB_G1_E1
 - Views
 - ASSEMBLY_LINE
 - ASSEMBLY_LINE_ASOF_VW
 - ASSEMBLY_LINE_VW
 - MOTOR
 - MOTOR_ASOF_VW** (selected)
 - MOTOR_VW
 - TESTUOT
 - TESTUOT1
 - TESTUOT1_ASOF_VW
 - TESTUOT1_VW

Query Editor (Center):

```
select
  GROUP_ID, EDGE_NODE_ID, DEVICE_ID,
  DEVICE_NAME,
  MEASURE_TIMESTAMP,
  TEMPERATURE,
  RPMS,
  AMPS,
  INSERTED_AT
from CL_BRIDGE_NODE_DB.NB_G1_E1.MOTOR_ASOF_VW
where GROUP_ID = 'G1' and EDGE_NODE_ID = 'E1' and DEVICE_NAME = 'Motor 3'
```

Results (Bottom):

	GROUP_ID	EDGE_NODE_ID	DEVICE_ID	DEVICE_NAME	MEASURE_TIMESTAMP	TEMPERATURE	RPMS	AMPS	INSERTED_AT
1	G1	E1	PLC 1	Motor 3	1730848316753	49	1202	121	1730848515389
2	G1	E1	PLC 1	Motor 3	1730848316753	49		121	1730848515389
3	G1	E1	PLC 1	Motor 3	1730848316753	49			1730848515389
4	G1	E1	PLC 1	Motor 3	1730848316753	49			1730848515389
5	G1	E1	PLC 1	Motor 3	1730848316753	49			1730848515389

Query Details (Right):

- Query duration: 134ms
- Rows: 5
- Query ID: 07b82a8f-0004-20cf-d...

Schema Details (Bottom Right):

- GROUP_ID: 5
- EDGE_NODE_ID: 5
- DEVICE_ID: 5

Metadata (Bottom Left):

Column	DataType
GROUP_ID	VARCHAR(16777216)
EDGE_NODE_ID	VARCHAR(16777216)
DEVICE_ID	VARCHAR(16777216)
DEVICE_NAME	VARCHAR(16777216)
MESSAGE_SEQUENCE	NUMBER(38,0)
ROOT_MESSAGE_TIMESTAMP	NUMBER(38,0)
INSERTED_AT	NUMBER(38,0)
MESSAGE_TYPE	VARCHAR(16777216)
MESSAGE_TYPE_ORDER	NUMBER(38,0)
NBIRTH_OR_NDEATH	VARCHAR(16777216)
NBIRTH_BIDSEQ	NUMBER(38,0)
NDEATH_BIDSEQ	NUMBER(38,0)
NBIRTH_NDEATH_INSERTED_AT	NUMBER(38,0)
IS_LAST_KNOWN_DODD_READING	BOOLEAN
IS_NODE_ALIVE	BOOLEAN
IS_HISTORICAL	BOOLEAN