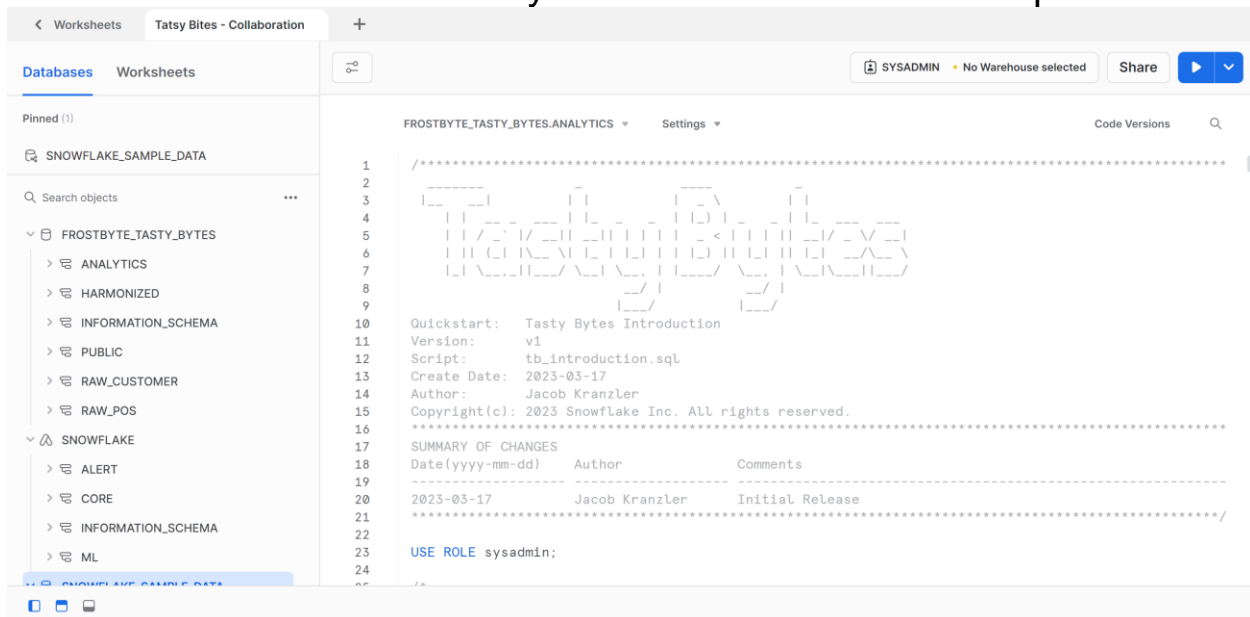


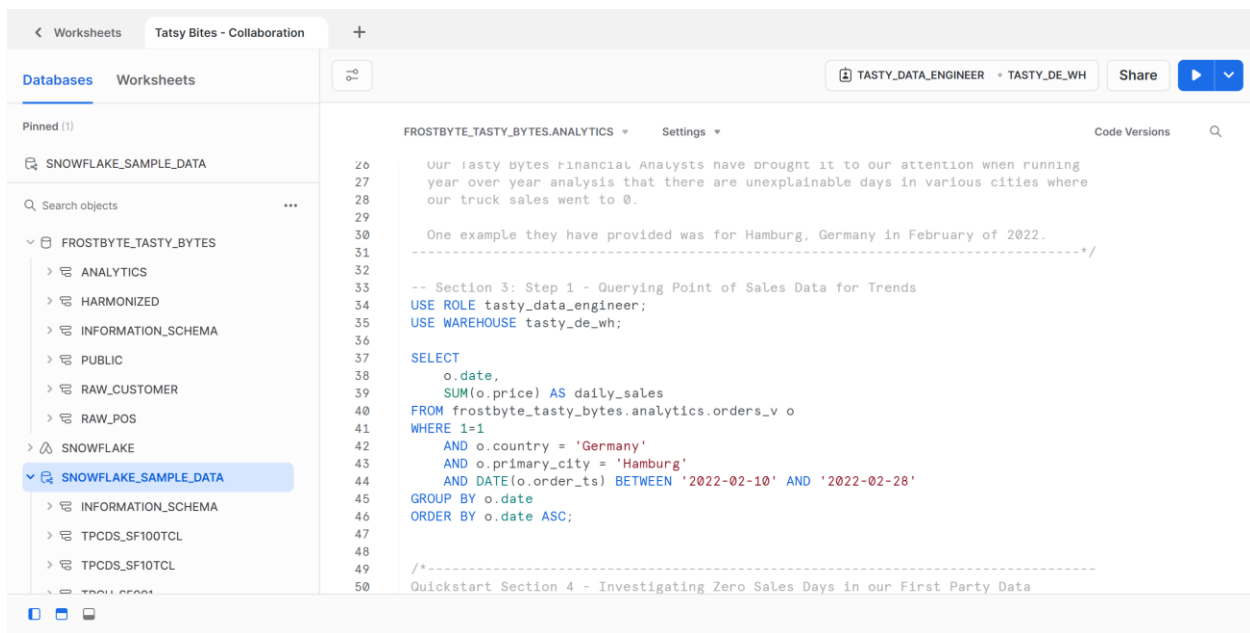
# 1. Direct Access to Third Party Data via Snowflake Marketplace



The screenshot shows the Snowflake web interface. The top navigation bar includes 'Worksheets' and 'Tatsy Bites - Collaboration'. The left sidebar shows a tree view of databases, with 'SNOWFLAKE' expanded. The main panel displays the 'FROSTBYTE\_TASTY\_BYTES.ANALYTICS' database. The script content is as follows:

```
1 /*****
2
3
4
5
6
7
8
9
10 Quickstart: Tasty Bytes Introduction
11 Version: v1
12 Script: tb_introduction.sql
13 Create Date: 2023-03-17
14 Author: Jacob Kranzler
15 Copyright(c): 2023 Snowflake Inc. All rights reserved.
16
17 SUMMARY OF CHANGES
18 Date(yyyy-mm-dd) Author Comments
19
20 2023-03-17 Jacob Kranzler Initial Release
21
22
23 USE ROLE sysadmin;
24
25
```

## 2. Creating a Worksheet and Copying in our SQL



The screenshot shows the Snowflake web interface. The top navigation bar includes 'Worksheets' and 'Tatsy Bites - Collaboration'. The left sidebar shows a tree view of databases, with 'SNOWFLAKE' expanded. The main panel displays a worksheet with the following SQL code:

```
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
```

Our tasty bytes financial analysts have brought it to our attention when running year over year analysis that there are unexplainable days in various cities where our truck sales went to 0.

One example they have provided was for Hamburg, Germany in February of 2022.

```
-- Section 3: Step 1 - Querying Point of Sales Data for Trends
USE ROLE tasty_data_engineer;
USE WAREHOUSE tasty_de_wh;

SELECT
  o.date,
  SUM(o.price) AS daily_sales
FROM frostbyte_tasty_bytes.analytics.orders_v o
WHERE 1=1
  AND o.country = 'Germany'
  AND o.primary_city = 'Hamburg'
  AND DATE(o.order_ts) BETWEEN '2022-02-10' AND '2022-02-28'
GROUP BY o.date
ORDER BY o.date ASC;
```

Quickstart Section 4 - Investigating Zero Sales Days in our First Party Data

## 3. Investigating Zero Sales Days in our First Party Data

### Step 1 - Querying Point of Sales Data for Trends

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing a tree structure with 'SNOWFLAKE' expanded and 'SNOWFLAKE\_SAMPLE\_DATA' selected. The main panel displays a query result for the 'FROSTBYTE\_TASTY\_BYTES' database. The query text is visible at the top, and the results are shown in a table format. The table has two columns: 'DATE' and 'DAILY\_SALES'. The results show data for dates from 2022-02-10 to 2022-02-24. A 'Query Details' panel on the right shows the query duration as 61ms and the number of rows as 13.

	DATE	DAILY_SALES
1	2022-02-10	423019.5000
2	2022-02-11	507394.7500
3	2022-02-12	544517.0000
4	2022-02-13	522072.0000
5	2022-02-14	442119.5000
6	2022-02-15	407481.2500
7	2022-02-22	411560.5000
8	2022-02-23	480826.7500
9	2022-02-24	432811.2500

## 4. Leveraging Weather Source Data from the Snowflake Marketplace

### Step 1 - Acquiring the Weather Source LLC: frostbyte Snowflake Marketplace Listing

The screenshot shows the Snowflake Marketplace listing for 'Weather Source LLC'. The listing is titled 'Get it for Free' and includes a list of features: 'Data updated hourly', 'Data for global at postal code level', 'Data after 01-01-2019 with daily interval', 'Historical Actuals (2019 - present) in daily & hourly format with d...', 'Historical Forecast (2019 - present) in daily & hourly format with d...', and 'Real-time Forecast in hourly & daily format with d...'. The 'Options' section shows the database name 'FROSTBYTE\_WEATHERSOURCE' and the roles 'ACCOUNTADMIN' and 'PUBLIC'. A 'Get' button is visible at the bottom. The right sidebar shows the 'Providers' and 'My Requests' sections, with 'Weather' listed as a provider.

## Step 2 - Harmonizing First and Third Party Data

The screenshot shows the Snowflake SQL Editor interface. On the left, the 'Databases' tab is active, showing a tree view of the 'FROSTBYTE\_TASTY\_BYTES' database. The 'SNOWFLAKE\_SAMPLE\_DATA' database is selected. The main editor area displays a SQL query to create or replace a view named 'frostbyte\_tasty\_bytes.harmonized.daily\_weather\_v'. The query uses a SELECT statement with columns from 'hd' and 'pc' tables, joined with 'frostbyte\_weathersource.onpoint\_id.history\_day' and 'frostbyte\_tasty\_bytes.raw\_pos.country\_c'. The query is executed, and the 'Results' tab shows a single row with the status 'View DAILY\_WEATHER\_V successfully created.'.

```
74 CREATE OR REPLACE VIEW frostbyte_tasty_bytes.harmonized.daily_weather_v
75 AS
76 SELECT
77     hd.*,
78     TO_VARCHAR(hd.date_valid_std, 'YYYY-MM') AS yyyy_mm,
79     pc.city_name AS city,
80     c.country AS country_desc
81 FROM frostbyte_weathersource.onpoint_id.history_day hd
82 JOIN frostbyte_weathersource.onpoint_id.postal_codes pc
83     ON pc.postal_code = hd.postal_code
84     AND pc.country = hd.country
85 JOIN frostbyte_tasty_bytes.raw_pos.country c
86     ON c.iso_country = hd.country
```

status
1 View DAILY_WEATHER_V successfully created.

Query Details: Query duration 257ms, Rows 1, Query ID 01b1c6c9-0000-1dd9-0...

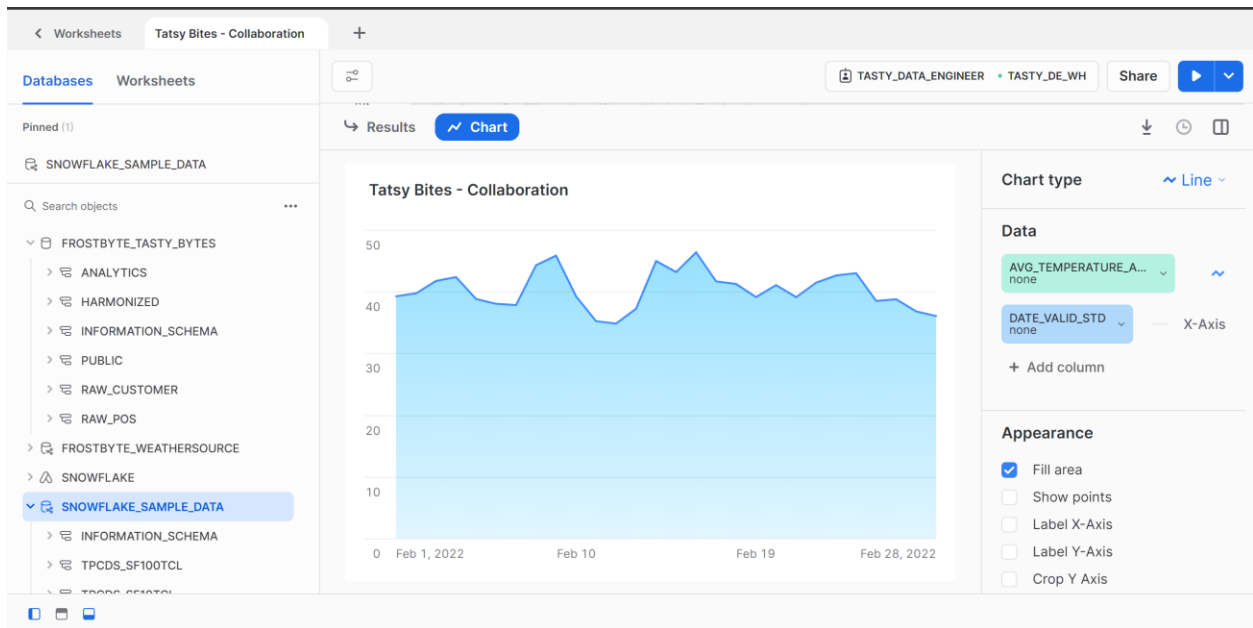
## Step 3 - Visualizing Daily Temperatures

The screenshot shows the Snowflake SQL Editor interface. The main editor area displays a SQL query to select data from the 'frostbyte\_tasty\_bytes.harmonized.daily\_weather\_v' view. The query filters for data from Germany in Hamburg for the year 2022, specifically for the month of February. The results are grouped by country\_desc, city\_name, and date\_valid\_std, and ordered by date\_valid\_std. The 'Results' tab shows a table with 4 rows of data.

```
90 -- Section 4: Step 3 - Visualizing Daily Temperatures
91 SELECT
92     dw.country_desc,
93     dw.city_name,
94     dw.date_valid_std,
95     AVG(dw.avg_temperature_air_2m_f) AS avg_temperature_air_2m_f
96 FROM frostbyte_tasty_bytes.harmonized.daily_weather_v dw
97 WHERE 1=1
98     AND dw.country_desc = 'Germany'
99     AND dw.city_name = 'Hamburg'
100     AND YEAR(date_valid_std) = '2022'
101     AND MONTH(date_valid_std) = '2'
102 GROUP BY dw.country_desc, dw.city_name, dw.date_valid_std
103 ORDER BY dw.date_valid_std DESC
```

	COUNTRY_DESC	CITY_NAME	DATE_VALID_STD	AVG_TEMPERATURF
1	Germany	Hamburg	2022-02-28	
2	Germany	Hamburg	2022-02-27	
3	Germany	Hamburg	2022-02-26	
4	Germany	Hamburg	2022-02-25	

Query Details: Query duration 2.0s, Rows 28, Query ID 01b1c6ca-0000-1dd9-0...



## Step 4 - Bringing in Wind Data

The screenshot shows a Snowflake worksheet interface. The main area displays a SQL query in the editor, which is executed and the results are shown in a table. The query selects data from 'FROSTBYTE\_TASTY\_BYTES.ANALYTICS' for Germany, Hamburg, in February 2022, grouped by date and ordered by date descending. The results table shows three rows of data for Germany, Hamburg, with dates 2022-02-28, 2022-02-27, and 2022-02-26. The right sidebar shows query details including duration (1.7s) and rows (28).

```
SELECT
  dw.country_desc,
  dw.city_name,
  dw.date_valid_std,
  MAX(dw.max_wind_speed_100m_mph) AS max_wind_speed_100m_mph
FROM frostbyte_tasty_bytes.harmonized.daily_weather_v dw
WHERE 1=1
  AND dw.country_desc IN ('Germany')
  AND dw.city_name = 'Hamburg'
  AND YEAR(date_valid_std) = '2022'
  AND MONTH(date_valid_std) = '2'
```

	COUNTRY_DESC	CITY_NAME	DATE_VALID_STD	MAX_WIND_SPEED
1	Germany	Hamburg	2022-02-28	
2	Germany	Hamburg	2022-02-27	
3	Germany	Hamburg	2022-02-26	

Query Details

- Query duration: 1.7s
- Rows: 28
- Query ID: 01b1c6cc-0000-1dd9-0...

## 5. Democratizing Data Insights

### Step 1 - Creating SQL Functions

Worksheets

Tatsy Bites - Collaboration

+

Databases

Worksheets

TASTY\_DATA\_ENGINEER

TASTY\_DE\_WH

Share

Pinned (1)

SNOWFLAKE\_SAMPLE\_DATA

Search objects

FROSTBYTE\_TASTY\_BYTES

ANALYTICS

HARMONIZED

INFORMATION\_SCHEMA

PUBLIC

RAW\_CUSTOMER

RAW\_POS

FROSTBYTE\_WEATHERSOURCE

SNOWFLAKE

SNOWFLAKE\_SAMPLE\_DATA

INFORMATION\_SCHEMA

TPCDS\_SF100TCL

130

131

132

133

134

135

136

137

138

139

140

141

142

143

-----~/

-- Section 5: Step 1 - Creating SQL Functions

--> create the SQL function that translates Fahrenheit to Celsius

CREATE OR REPLACE FUNCTION frostbyte\_tasty\_bytes.analytics.fahrenheit\_to\_celsius(temp\_f NUMBER(35,4))

RETURNS NUMBER(35,4)

AS

\$\$

(temp\_f - 32) \* (5/9)

\$\$;

--> create the SQL function that translates Inches to Millimeter

CREATE OR REPLACE FUNCTION frostbyte\_tasty\_bytes.analytics.inch\_to\_millimeter(inch NUMBER(35,4))

RETURNS NUMBER(35,4)

Results

Chart

status

1

Function FAHRENHEIT\_TO\_CELSIUS successfully created.

Query Details

Query duration

45ms

Rows

1

Query ID

01b1c6cd-0000-1dd9-0...

Worksheets

Tatsy Bites - Collaboration

+

Databases

Worksheets

TASTY\_DATA\_ENGINEER

TASTY\_DE\_WH

Share

Pinned (1)

SNOWFLAKE\_SAMPLE\_DATA

Search objects

FROSTBYTE\_TASTY\_BYTES

ANALYTICS

HARMONIZED

INFORMATION\_SCHEMA

PUBLIC

RAW\_CUSTOMER

RAW\_POS

FROSTBYTE\_WEATHERSOURCE

SNOWFLAKE

SNOWFLAKE\_SAMPLE\_DATA

INFORMATION\_SCHEMA

TPCDS\_SF100TCL

139

140

141

142

143

144

145

146

147

148

149

150

151

152

139

140

141

142

143

144

145

146

147

148

149

150

151

152

-----~/

--> create the SQL function that translates Inches to Millimeter

CREATE OR REPLACE FUNCTION frostbyte\_tasty\_bytes.analytics.inch\_to\_millimeter(inch NUMBER(35,4))

RETURNS NUMBER(35,4)

AS

\$\$

inch \* 25.4

\$\$;

-- Section 5: Step 2 - Creating the SQL for our View

SELECT

fd.data.valid AND AS data

Results

Chart

status

1

Function INCH\_TO\_MILLIMETER successfully created.

Query Details

Query duration

41ms

Rows

1

Query ID

01b1c6ce-0000-1dd9-0...

## Step 2 - Creating the SQL for our View

Worksheets Tasty Bites - Collaboration

Databases Worksheets

Pinned (1)

SNOWFLAKE\_SAMPLE\_DATA

Search objects

FROSTBYTE\_TASTY\_BYTES

- ANALYTICS
- HARMONIZED
- INFORMATION\_SCHEMA
- PUBLIC
- RAW\_CUSTOMER
- RAW\_POS
- FROSTBYTE\_WEATHERSOURCE
- SNOWFLAKE
- SNOWFLAKE\_SAMPLE\_DATA**
- INFORMATION\_SCHEMA
- TPCDS\_SF100TCL
- TPCDS\_SF100TCL

Code Versions

```
-- Section 5: Step 2 - Creating the SQL for our View
SELECT
  fd.date_valid_std AS date,
  fd.city_name,
  fd.country_desc,
  ZEROIFNULL(SUM(odv.price)) AS daily_sales,
  ROUND(AVG(fd.avg_temperature_air_2m_f),2) AS avg_temperature_fahrenheit,
  ROUND(AVG(frostbyte_tasty_bytes.analytics.fahrenheit_to_celsius(fd.avg_temperature_air_2m_f)),2)
AS avg_temperature_celsius,
  ROUND(AVG(fd.tot_precipitation_in),2) AS avg_precipitation_inches,
  ROUND(AVG(frostbyte_tasty_bytes.analytics.inch_to_millimeter(fd.tot_precipitation_in)),2) AS
avg_precipitation_millimeters,
  MAX(fd.max_wind_speed_100m_mph) AS max_wind_speed_100m_mph
```

Results Chart

	DATE	CITY_NAME	COUNTRY_DESC	DAILY_SALES	AVG_
1	2022-02-01	Hamburg	Germany	41807100.0000	
2	2022-02-02	Hamburg	Germany	42598550.0000	
3	2022-02-03	Hamburg	Germany	42015025.0000	

Query Details

Query duration 13s

Rows 28

Query ID 01b1c6cf-0000-1dd9-0...

## Step 3 - Deploying our Analytics View

Worksheets Tasty Bites - Collaboration

Databases Worksheets

Pinned (1)

SNOWFLAKE\_SAMPLE\_DATA

Search objects

FROSTBYTE\_TASTY\_BYTES

- ANALYTICS
- HARMONIZED
- INFORMATION\_SCHEMA
- PUBLIC
- RAW\_CUSTOMER
- RAW\_POS
- FROSTBYTE\_WEATHERSOURCE
- SNOWFLAKE
- SNOWFLAKE\_SAMPLE\_DATA**
- INFORMATION\_SCHEMA
- TPCDS\_SF100TCL
- TPCDS\_SF100TCL

Code Versions

```
-- Section 5: Step 3 - Deploying our Analytics View
CREATE OR REPLACE VIEW frostbyte_tasty_bytes.analytics.daily_city_metrics_v
COMMENT = 'Daily Weather Source Metrics and Orders Data for our Cities'
AS
SELECT
  fd.date_valid_std AS date,
  fd.city_name,
  fd.country_desc,
  ZEROIFNULL(SUM(odv.price)) AS daily_sales,
  ROUND(AVG(fd.avg_temperature_air_2m_f),2) AS avg_temperature_fahrenheit,
  ROUND(AVG(frostbyte_tasty_bytes.analytics.fahrenheit_to_celsius(fd.avg_temperature_air_2m_f)),2)
AS avg_temperature_celsius,
  ROUND(AVG(fd.tot_precipitation_in),2) AS avg_precipitation_inches
```

Results Chart

status
1 View DAILY_CITY_METRICS_V successfully created.

Query Details

Query duration 306ms

Rows 1

Query ID 01b1c6d0-0000-1dd8-...

## 6. Deriving Insights from Sales and Marketplace Weather Data

### Step 1 - Simplifying our Analysis

Worksheets

Tatsy Bites - Collaboration

Databases

Worksheets

Pinned (1)

SNOWFLAKE\_SAMPLE\_DATA

Search objects

FROSTBYTE\_TASTY\_BYTES

ANALYTICS

HARMONIZED

INFORMATION\_SCHEMA

PUBLIC

RAW\_CUSTOMER

RAW\_POS

FROSTBYTE\_WEATHERSOURCE

SNOWFLAKE

SNOWFLAKE\_SAMPLE\_DATA

INFORMATION\_SCHEMA

TPCDS\_SF100TCL

TASTY\_DATA\_ENGINEER

TASTY\_DE\_WH

Share

FROSTBYTE\_TASTY\_BYTES.ANALYTICS

Settings

Code Versions

-- Section 6: Step 1 - Simplifying our Analysis

SELECT

dcm.date,

dcm.city\_name,

dcm.country\_desc,

dcm.daily\_sales,

dcm.avg\_temperature\_fahrenheit,

dcm.avg\_temperature\_celsius,

dcm.avg\_precipitation\_inches,

dcm.avg\_precipitation\_millimeters,

dcm.max\_wind\_speed\_100m\_mph

FROM frostbyte\_tasty\_bytes.analytics.daily\_city\_metrics\_v dcm

WHERE 1=1

Results

Chart

	DATE	CITY_NAME	COUNTRY_DESC	DAILY_SALES	AVG_
1	2022-02-24	Hamburg	Germany	43281125.0000	
2	2022-02-23	Hamburg	Germany	48082675.0000	
3	2022-02-22	Hamburg	Germany	41156050.0000	

Query Details

Query duration6.2s

Rows24

Query ID01b1c6d1-0000-1dd9-0...