Analyzing NPS Data in Snowflake

Shweta Ajay Shinde

Masters in Data Analytics, San Jose State University

Data 226: Data warehousing

Instructor: Keeyong Han

18th Sept 2024

Analyzing NPS Data in Snowflake

1. (+1) Explain what NPS (Net Promoter Score) is and how it is calculated.

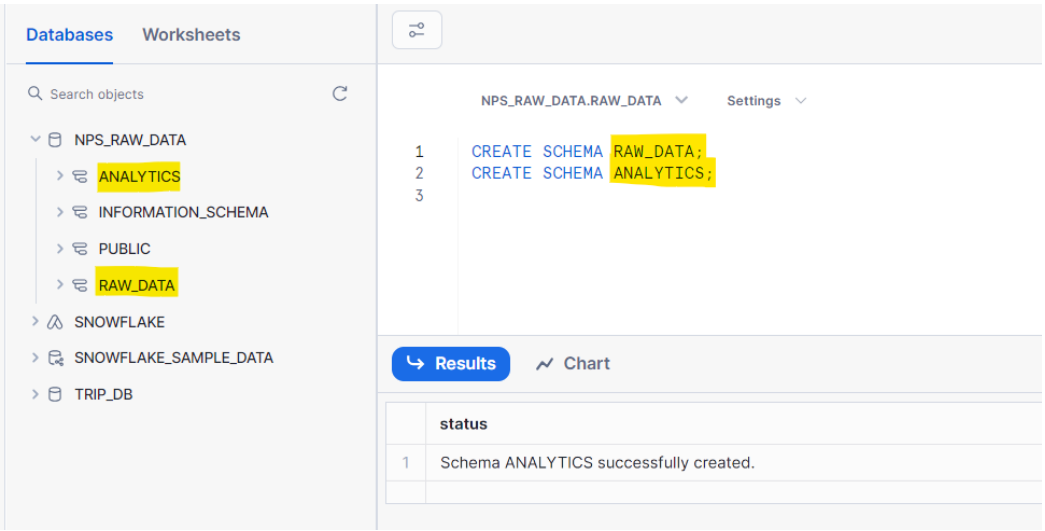
Net Promoter Score (NPS) is a way to measure how happy customers are and how likely they are to tell others about a company's product or service. The score ranges from -100 to 100 and is derived from survey responses to the question: "On a scale from 0 to 10, how likely are you to recommend our company/product/service to a friend or colleague?"
Calculation:

- Categorize Respondents:
  - **Promoters (Score 9-10):** Loyal customers who are likely to spread positive word-of-mouth.
  - **Passives (Score 7-8):** Satisfied but unenthusiastic customers who are vulnerable to competitors.
  - **Detractors (Score 0-6):** Unhappy customers who may spread negative feedback.
- Calculate the Percentage:
  - **% Promoters** = (Number of Promoters / Total Responses) * 100
  - **% Detractors** = (Number of Detractors / Total Responses) * 100
- Compute NPS:
  - NPS = % Promoters - % Detractors OR
  - NPS=(Number of Promoters- Number of Detractors) * 100 / Total Responses

2. (+1) Create raw_data and analytics schemas under a database in your Snowflake.

CREATE SCHEMA RAW_DATA;

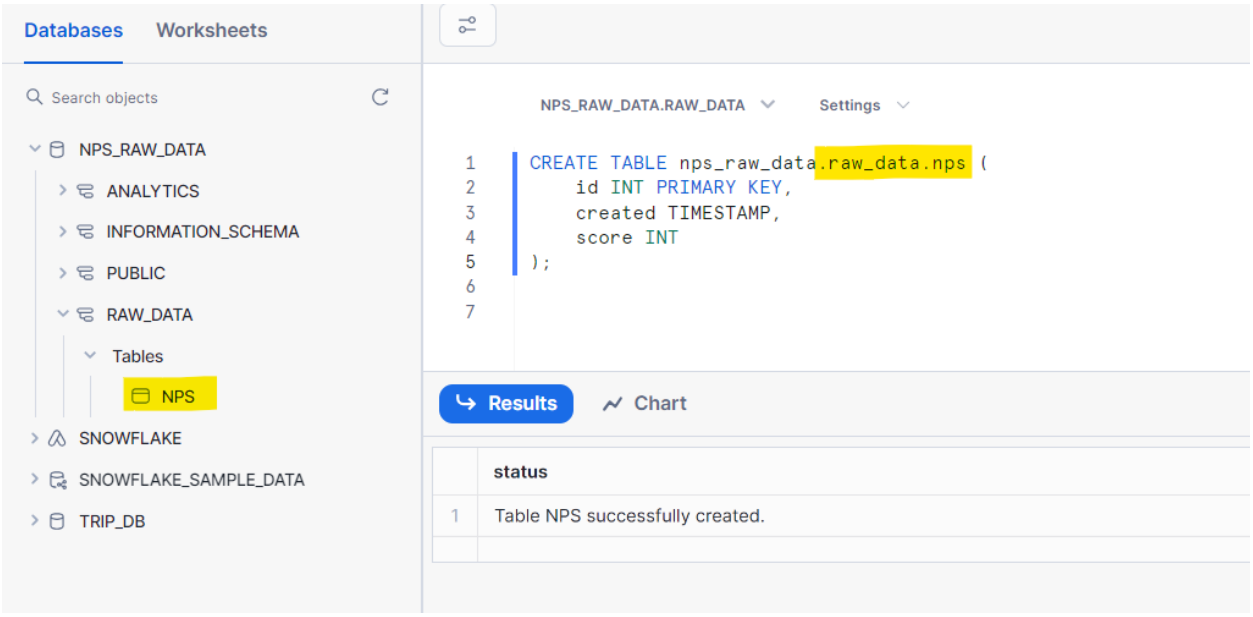
CRATE SCHEMA ANALYTICS;



Analyzing NPS Data in Snowflake

3. (+1) Create a table named nps with primary key attribute under raw_data schema.

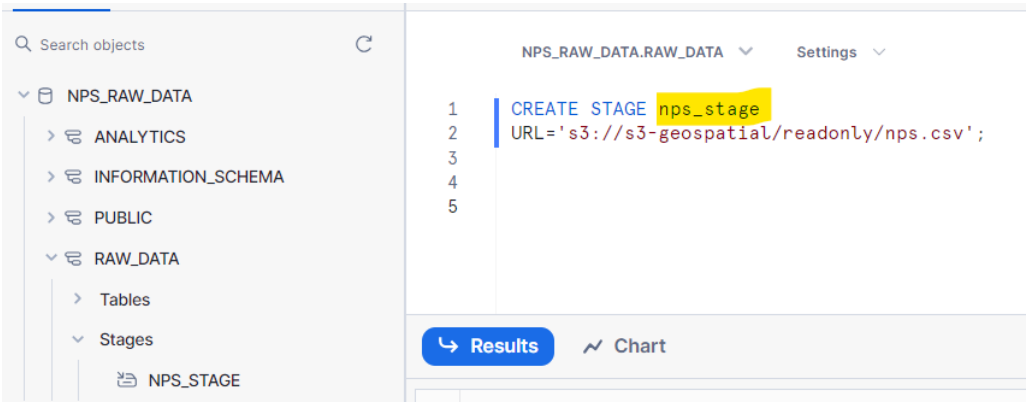
```
CREATE TABLE nps_raw_data.raw_data.nps (
    id INT PRIMARY KEY,
    created TIMESTAMP,
    score INT
);
```



4. (+2) Copy a file (nps.csv) into the nps table using a stage :-s3://s3-geospatial/readonly/nps.csv

First, create a stage:

```
CREATE STAGE nps_stage
URL='s3://s3-geospatial/readonly/nps.csv'
```



Analyzing NPS Data in Snowflake

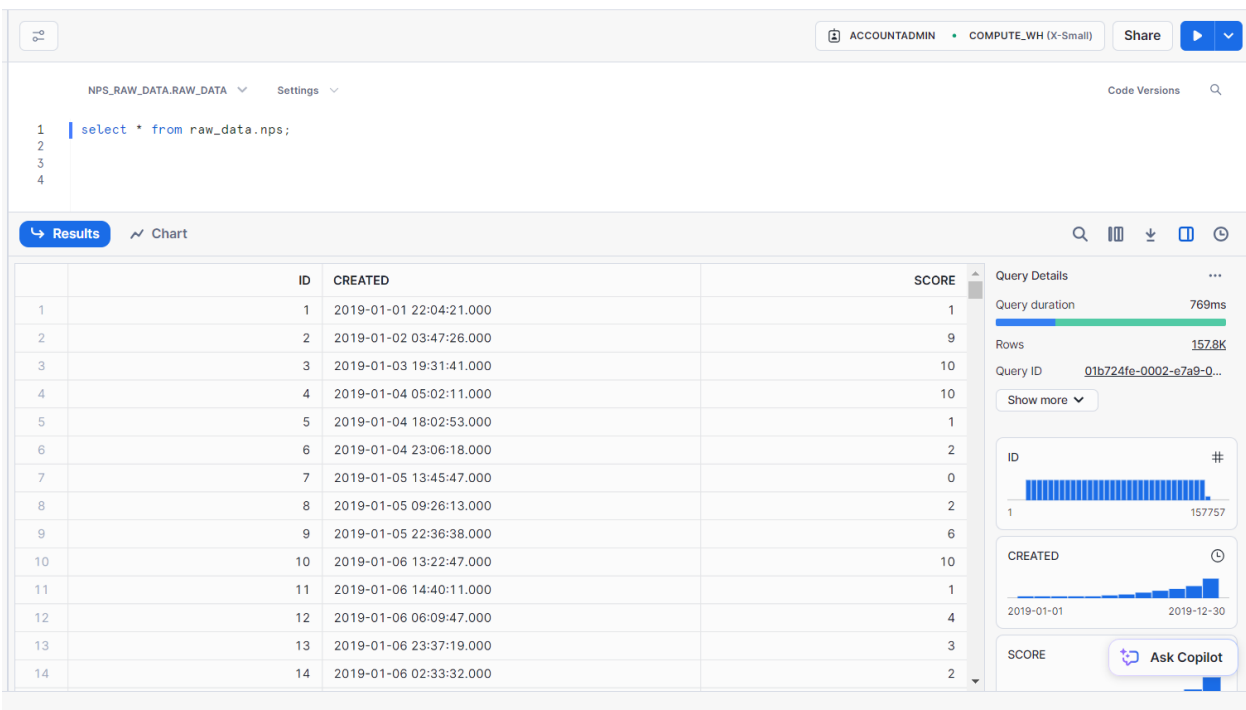
Then, copy the CSV data into the nps table:

COPY INTO raw_data.nps

FROM @nps_stage

FILE_FORMAT = (TYPE = 'CSV' FIELD_OPTIONALLY_ENCLOSED_BY = '"' SKIP_HEADER = 1);

Select * from raw_data.nps



Analyzing NPS Data in Snowflake

5.(+2) What types of data quality validations can we perform against the nps table? Please name at least 3 methods.

Here are three types of data quality validations you can perform against an NPS table using SQL:

1. Check for Null Values
Ensure that there are no missing values in key columns, such as the score and ID identifier.
SQL to check if there are any NULL values.

SELECT COUNT(*) AS NullCount
FROM nps
WHERE score IS NULL OR id IS NULL;

NPS_RAW_DATA.RAW_DATA Settings

```
SELECT COUNT(*) AS NullCount
FROM nps
WHERE score IS NULL OR  id IS NULL;
```

Results Chart

	NULLCOUNT
1	0

2. Validate Score Range
Check that the NPS scores fall within the valid range of 0 to 10.
SQL to check the valid range.

SELECT COUNT(*) AS InvalidScoreCount
FROM nps
WHERE score < 0 OR score > 10;

NPS_RAW_DATA.RAW_DATA Settings

```
SELECT COUNT(*) AS InvalidScoreCount
FROM nps
WHERE score < 0 OR score > 10;
```

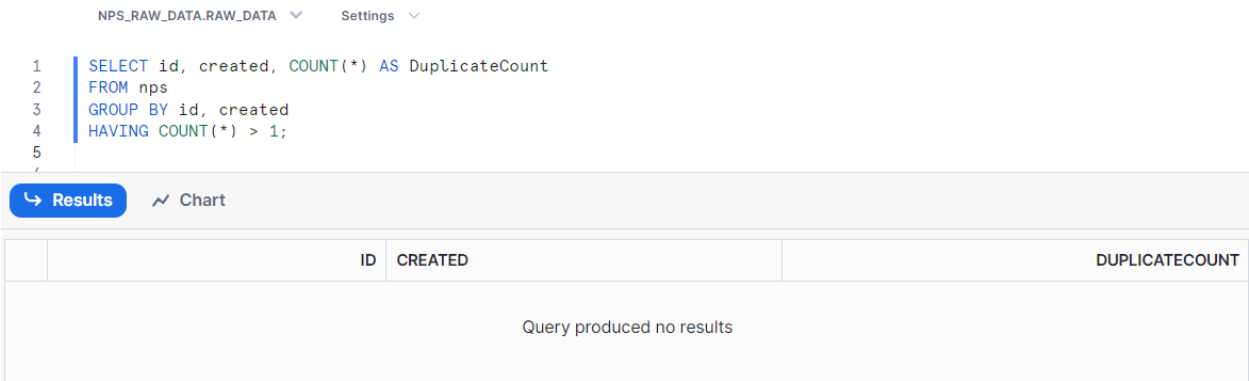
Results Chart

	INVALIDSCORECOUNT
1	0

Analyzing NPS Data in Snowflake

3. Check for Duplicate Entries
Ensure there are no duplicate records for the same ID on the same survey date.
SQL to check if there are duplicates.

```
SELECT id, created, COUNT(*) AS DuplicateCount
FROM nps
GROUP BY id, created
HAVING COUNT(*) > 1;
```



Overall, these tests help maintain accurate and reliable data for better decision-making

6.(+4) Develop a SELECT SQL query to calculate monthly NPS, with results sorted by month in ascending order.

→Here I am using WITH to Improved Readability, Reuse of Logic and Organization of query

```
WITH nps_data AS (
   SELECT
      DATE_TRUNC('MONTH', created) AS month,
      AVG(score) AS average_score,
      COUNT(CASE WHEN score >= 9 THEN 1 END) AS promoters,
      COUNT(CASE WHEN score <= 6 THEN 1 END) AS detractors,
      COUNT(*) AS total_responses
   FROM raw_data.nps
   GROUP BY month
)
SELECT
   TO_CHAR(month, 'YYYY-MM') AS month,
   COALESCE(ROUND(((promoters - detractors) * 100.0 / total_responses), 2 ), 0) AS
     "AverageNPS"
FROM nps_data
ORDER BY month ASC;
```

Analyzing NPS Data in Snowflake

NPS_RAW_DATA.RAW_DATA Settings

```sql
WITH nps_data AS (
    SELECT
        DATE_TRUNC('MONTH', created) AS month,
        AVG(score) AS average_score,
        COUNT(CASE WHEN score >= 9 THEN 1 END) AS promoters,
        COUNT(CASE WHEN score <= 6 THEN 1 END) AS detractors,
        COUNT(*) AS total_responses
    FROM raw_data.nps
    GROUP BY month
)
SELECT
    TO_CHAR(month, 'YYYY-MM') AS month,
    COALESCE(ROUND(
        ((promoters - detractors) * 100.0 / total_responses), 2
    ), 0) AS "Average NPS"
FROM nps_data
ORDER BY month ASC;
```

Results Chart

	MONTH	Average NPS
1	2019-01	2.36
2	2019-02	30.54
3	2019-03	52.91
4	2019-04	53.00
5	2019-05	54.52
6	2019-06	65.02
7	2019-07	64.51
8	2019-08	67.71
9	2019-09	37.95
10	2019-10	53.29
11	2019-11	61.29
12	2019-12	65.99

Or we can use the below SQL without WITH:-

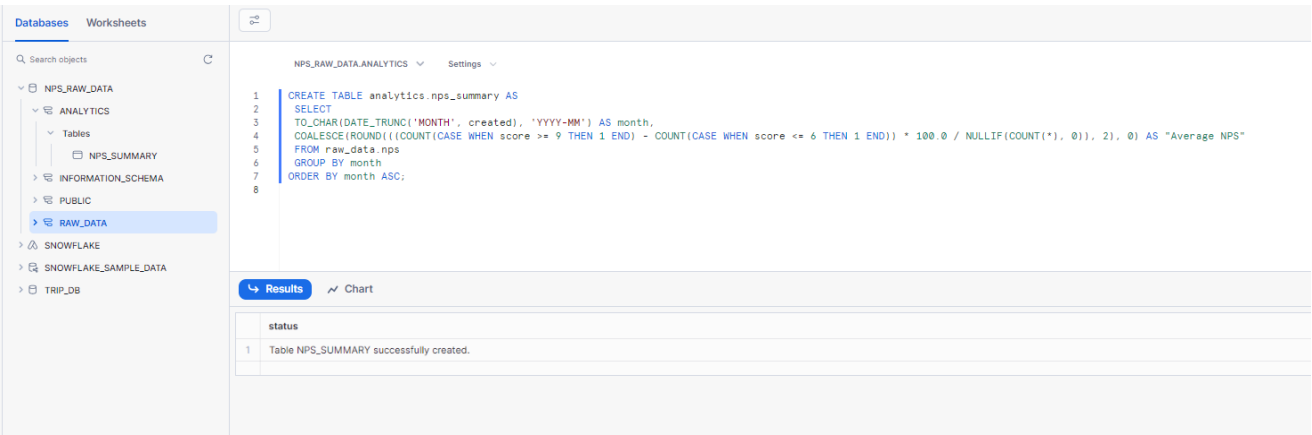
```sql
SELECT
  TO_CHAR(DATE_TRUNC('MONTH', created), 'YYYY-MM') AS month,
  COALESCE(ROUND(
  ((COUNT(CASE WHEN score >= 9 THEN 1 END) –
     COUNT(CASE WHEN score <= 6 THEN 1 END)) * 100.0 /
     NULLIF(COUNT(*), 0)), 2), 0) AS "Average NPS"
FROM raw_data.nps
GROUP BY month
ORDER BY month ASC;
```

MONTH	Average NPS
2019-01	2.36
2019-02	30.54
2019-03	52.91
2019-04	53.00
2019-05	54.52
2019-06	65.02
2019-07	64.51
2019-08	67.71
2019-09	37.95
2019-10	53.29
2019-11	61.29
2019-12	65.99

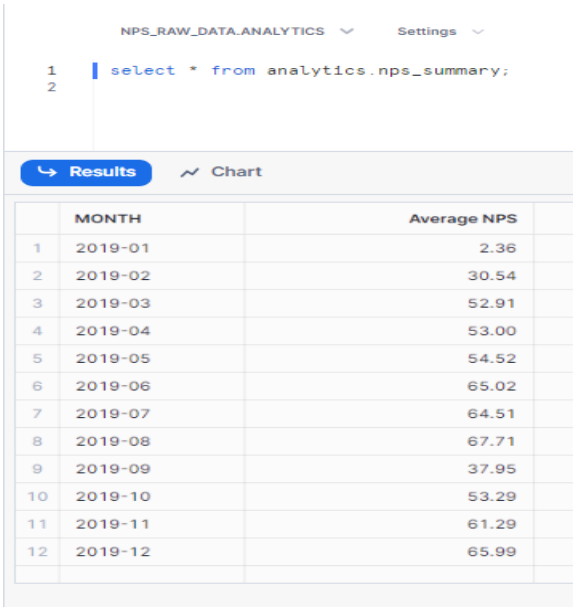
Analyzing NPS Data in Snowflake

7. (+2) Use CTAS (CREATE TABLE AS SELECT) to generate the nps_summary table in the analytics schema, populating it with results from step 6.

CREATE TABLE analytics.nps_summary AS
SELECT
TO_CHAR(DATE_TRUNC('MONTH', created), 'YYYY-MM') AS month,
COALESCE(ROUND(((COUNT(CASE WHEN score >= 9 THEN 1 END) - COUNT(CASE WHEN score <= 6 THEN 1 END)) * 100.0 / NULLIF(COUNT(*), 0)), 2), 0) AS "Average NPS"
FROM raw_data.nps
GROUP BY month
ORDER BY month ASC;



Select * from analytics.nps_summary;



	MONTH	Average NPS
1	2019-01	2.36
2	2019-02	30.54
3	2019-03	52.91
4	2019-04	53.00
5	2019-05	54.52
6	2019-06	65.02
7	2019-07	64.51
8	2019-08	67.71
9	2019-09	37.95
10	2019-10	53.29
11	2019-11	61.29
12	2019-12	65.99