# Google Search Trends

Joseph Bailey, Sam LeFebre,
Archie Korale

8 December 2025

# Contents

Background of Data

01

## Name, Contents, and Source

- “Google Trends Search Data” from Kaggle
- Contains about 16500 rows and 8 columns of data
- Contains trending search queries over time
- Helps us understand what topics are gaining attention during which periods of time

02

## Key Fields

- Search queries: What the user actually searched in google
- Time period: The start and end date of each search trending, as well as a marker on if the trend was still active
- Volume and Growth: Showed how popular the search was by how many people searched it and the percentage that search has increased
- Category: What topic the search aligned with the most

03

## Sample Data

| Query | NBA scores | Kraken v. Senators | Is the voice on tonight |
|---|---|---|---|
| Start Date | 2025-10-21 | 2025-10-21 | 2025-10-21 |
| End Date | null | null | null |
| Active | TRUE | TRUE | TRUE |
| Volume | 2000 | 5000 | 500 |
| Increase | 300 | 400 | 75 |
| Category | Sports | Sports | Entertainment |
| Breakdown | null | Kraken game, Seattle | null |

# Bronze Layer - Raw Data Storage

```
--Bronze layer setup
--====================
USE schema Bronze;
LIST @bronze_stage
--create stage for file upload
--CREATE OR REPLACE STAGE bronze_stage
--FILE_FORMAT = (
--    TYPE = 'CSV'
--    FIELD_DELIMITER = ','
--    SKIP_HEADER = 1
--    FIELD_OPTIONALLY_ENCLOSED_BY = '"'
--    TRIM_SPACE = TRUE
--    NULL_IF = ('NULL', 'null', '')
--);

--create bronze table (raw data)
CREATE OR REPLACE TABLE google_trends_raw (
query VARCHAR(500),
start_date TIMESTAMP_NTZ,
end_date TIMESTAMP_NTZ,
active BOOLEAN,
search_volume NUMBER(12,0),
increase_percentage FLOAT,
categories VARCHAR(1000),
trend_breakdown VARCHAR(5000)
);
```

## Set Up

- Started by creating a stage for the csv file upload
- We then created the bronze table called 'google_trends_raw' that would house our data
- Lastly, we ran a SELECT COUNT to make sure that the full data set was uploaded

```
--load data from the stage into the table created above
COPY INTO google_trends_raw (
    query,
    start_date,
    end_date,
    active,
    search_volume,
    increase_percentage,
    categories,
    trend_breakdown
)
FROM (
    SELECT
        $1::VARCHAR,
        $2::TIMESTAMP_NTZ,
        $3::TIMESTAMP_NTZ,
        $4::BOOLEAN,
        $5::NUMBER,
        $6::FLOAT,
        $7::VARCHAR,
        $8::VARCHAR
    FROM @bronze_stage
)
ON_ERROR = 'CONTINUE' --if there is an error uploading, continue on
PURGE = FALSE;  --don't delete from stage

-- View first 10 rows
SELECT *
FROM google_trends_raw
LIMIT 10;


-- View sample of data by category
SELECT
    categories,
    COUNT(*) AS query_count,
    AVG(search_volume) AS avg_volume,
    MAX(increase_percentage) AS max_increase
FROM google_trends_raw
GROUP BY categories
ORDER BY query_count DESC
LIMIT 10;
```

## Implementation

- After completing the setup, we loaded the data from the stage into the tables created
- Lastly, we ran a SELECT command to see the first 10 rows of data from our data table to make sure the dataset was imported correctly

# Silver Layer - Dimensional Model

```
--================================
--Create necessary dimension tables
--================================
--Queries
CREATE OR REPLACE TABLE dim_query (
    query_key NUMBER AUTOINCREMENT PRIMARY KEY,  -- Surrogate key (auto-generated ID)
    query VARCHAR(500) NOT NULL,                 -- The actual search term
    query_length NUMBER,                         -- Length of the query string
    word_count NUMBER,                           -- Number of words in the query
    UNIQUE (query)                               -- Prevents duplicate queries
);

--Categories
CREATE OR REPLACE TABLE dim_category (
    category_key NUMBER AUTOINCREMENT PRIMARY KEY,  -- Surrogate key
    category_name VARCHAR(500) NOT NULL,            -- Category name (e.g., "Sports", "Technology")
    UNIQUE (category_name)                          -- Each category appears only once
);

--Start and End Date
CREATE OR REPLACE TABLE dim_date (
    date_key NUMBER PRIMARY KEY,        -- Format: YYYYMMDD (e.g., 20241115)
    full_date DATE NOT NULL,            -- The actual date
    year NUMBER,                        -- Year (2024)
    quarter NUMBER,                     -- Quarter (1-4)
    month NUMBER,                       -- Month number (1-12)
    month_name VARCHAR(20),             -- Month name (January, February, etc.)
    week NUMBER,                        -- Week of the year (1-52)
    day_of_month NUMBER,                -- Day of the month (1-31)
    day_of_week NUMBER,                 -- Day of week (0=Sunday, 6=Saturday)
    day_name VARCHAR(20),               -- Day name (Monday, Tuesday, etc.)
    is_weekend BOOLEAN,                 -- TRUE if Saturday or Sunday
    UNIQUE (full_date)
```

## Set Up - Create Tables

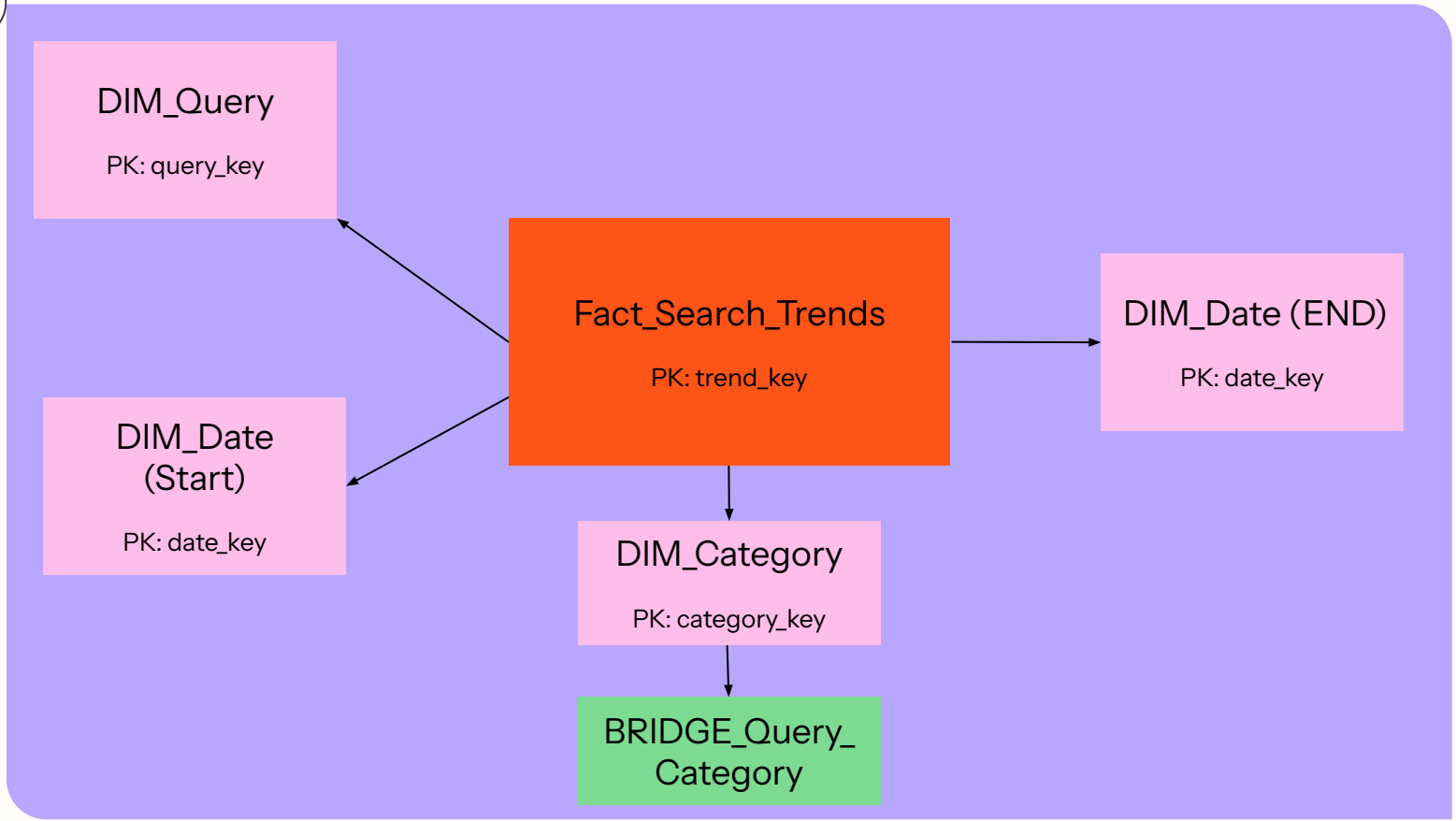
- Set up a Star Schema in order to transform the raw data from the bronze layer into the silver layer for analysis
- Created 3 dimension tables for queries, categories, and dates
- Created the necessary Fact Table to analyze the fact_search_trends
- Created 1 Bridge Table for the many-to-many relationship of queries and categories

```
--Populate DIM_QUERY
INSERT INTO dim_query (
    query,
    query_length,
    word_count
)
SELECT DISTINCT
    query,
    LENGTH(query) AS query_length,  -- Count total characters
    -- Word count formula: Count spaces and add 1
    LENGTH(query) - LENGTH(REPLACE(query, ' ', '')) + 1 AS word_count
FROM Bronze.google_trends_raw
WHERE query IS NOT NULL;

--Populate DIM_CATEGORY
INSERT INTO dim_category (category_name)
WITH split_categories AS (
    SELECT DISTINCT
        TRIM(VALUE::STRING) AS category_name
    FROM Bronze.google_trends_raw,
    LATERAL FLATTEN(INPUT => SPLIT(categories, ','))
    WHERE categories IS NOT NULL
)
SELECT category_name
FROM split_categories
WHERE category_name != ''  -- Exclude empty strings
ORDER BY category_name;
```

## Implementation - Move Data

- Populated the tables with the data from the bronze layer while handling data quality throughout with null dates and split categories
- Split comma-separated categories into individual records, calculated attributes of the dates (weekend, what quarter, etc)

Star Schema


DIM_Query
PK: query_key
Fact_Search_Trends
PK: trend_key
DIM_Date (END)
PK: date_key
DIM_Date (Start)
PK: date_key
DIM_Category
PK: category_key
BRIDGE_Query_Category

# Gold Layer

## Category Popularity (The “Strategic” View)

**Business Question:** "Which broad topics (e.g., Sports, News) are capturing the most user attention?"
**Why it Matters:** Helps executives understand user interests at a high level to guide content strategy.
**Gold Table:** `gold_category_metrics`
**Key Metrics:** Total Search Volume, Number of Unique Trends, Average Growth %.

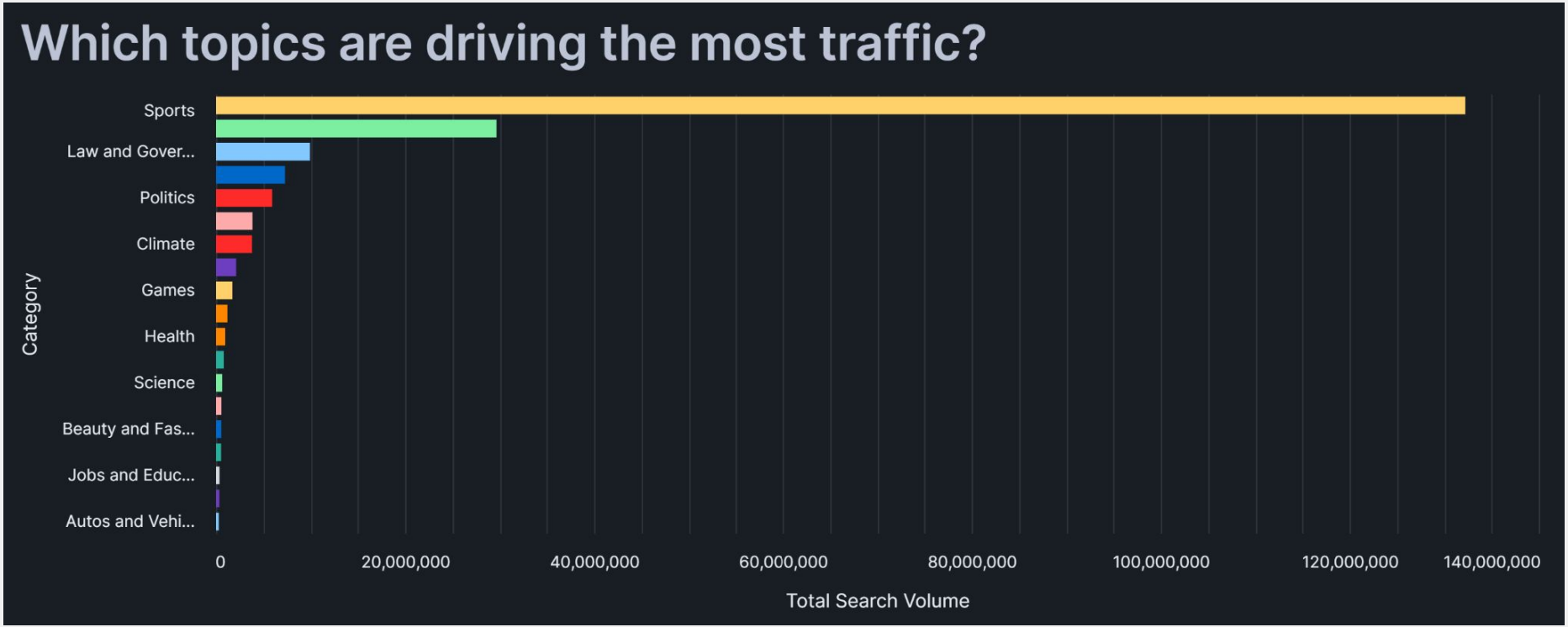
## Daily Traffic Trends (The "Operational" View)

**Business Question:** "When is our peak traffic occurring, and how does volume fluctuate day-to-day?"
**Why it Matters:** Essential for resource planning and identifying specific days with viral events.
**Gold Table:** `gold_daily_traffic`
**Key Metrics:** Total Daily Volume, Active Trends Count, Peak Days.

## Category Leaders (The "Tactical" View)

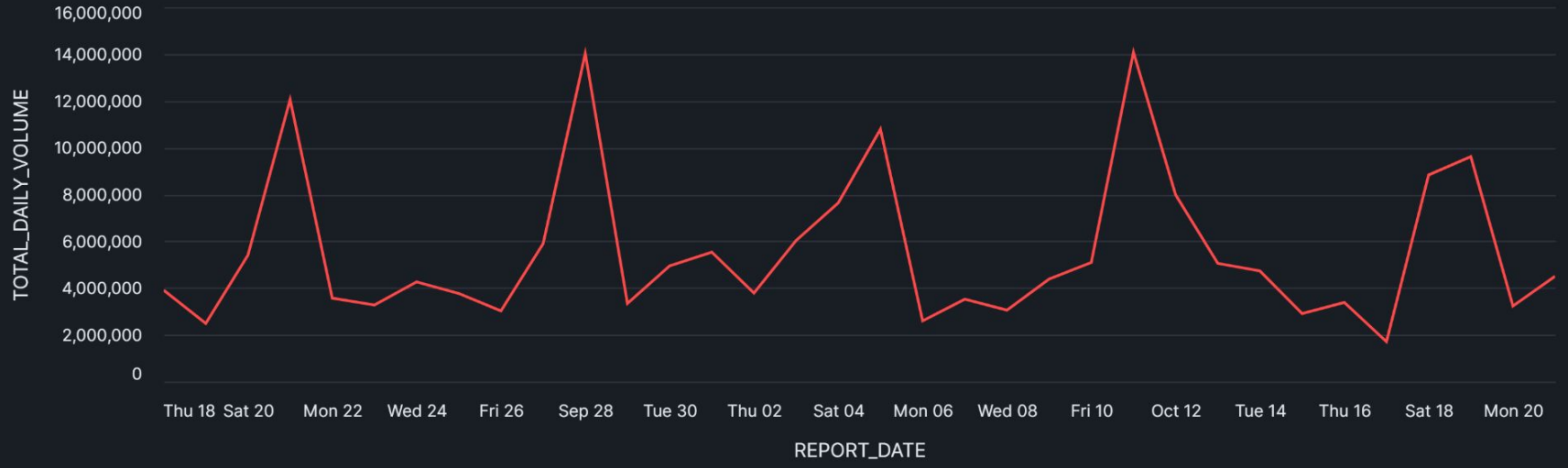
**Business Question:** "What specific search terms are the top drivers within our most popular categories?"
**Why it Matters:** Allows teams to drill down from a broad category (like "Sports") to see the exact events (like "Bills vs Dolphins") driving the numbers.
**Gold Table:** `gold_category_leaders`
**Key Metrics:** Top 10 Ranked Queries per Category, Search Volume.

# Category Popularity



# Daily Traffic Trends

## How does search volume change over time?



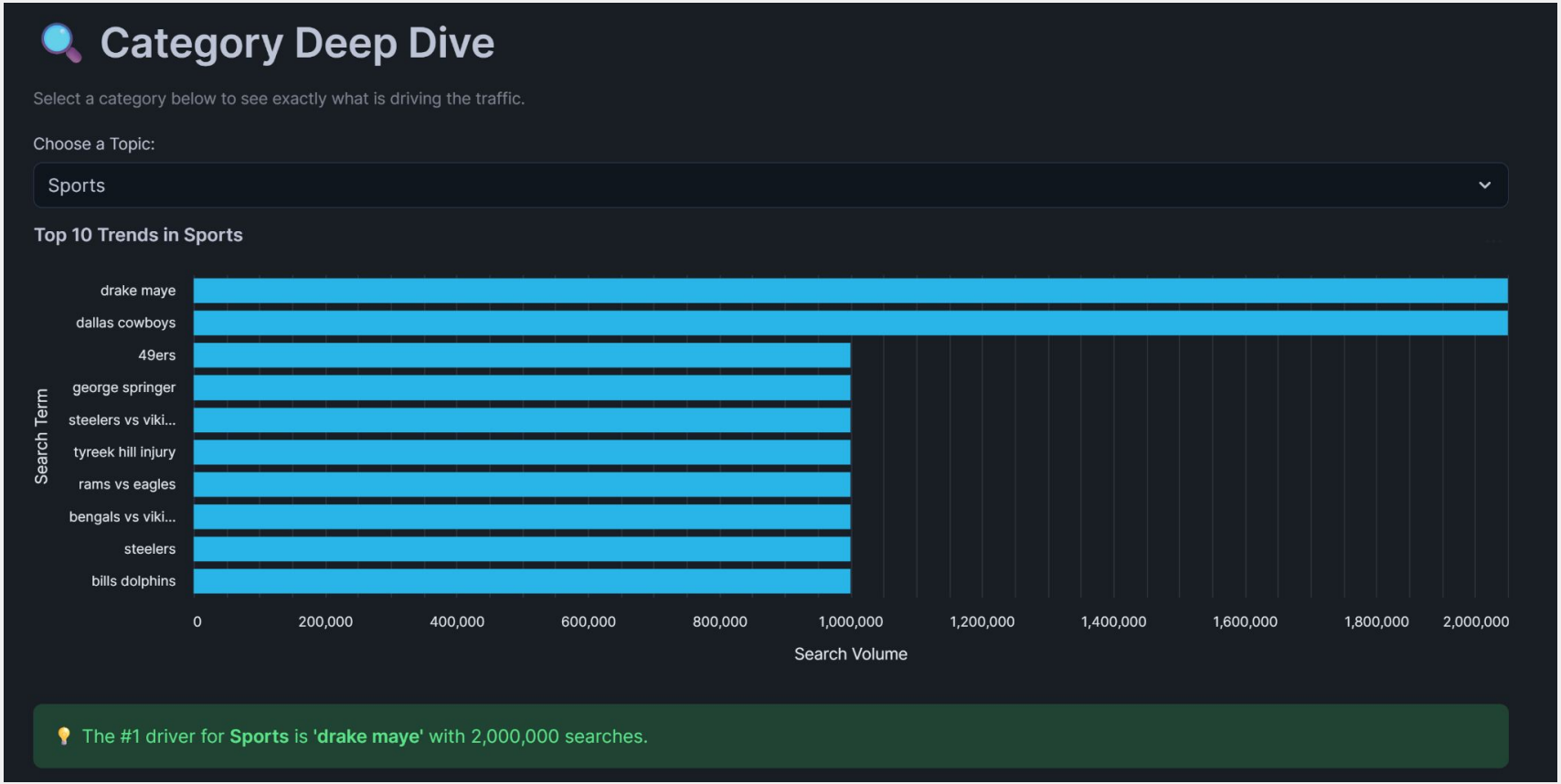

Total Volume Recorded

188,324,000

Peak Traffic Day

2025-10-11

# Category Leaders



# AI-SQL Function Call

Applied Snowflake Cortex AI functions on the Bronze layer which generated:

- Trend sentiment scores
- Automated text summaries

AI applied to:

- TREND_BREAKDOWN (raw trend description)

Output stored in:

- BRONZE.GOOGLE_TRENDS_RAW

QUERY

```
USE ROLE ROLE_Team_JAS;
USE DATABASE DB_Team_JAS;
USE SCHEMA BRONZE;
USE WAREHOUSE Animal_Task_WH;

SELECT
  QUERY,
  TREND_BREAKDOWN,
  TREND_SENTIMENT_SCORE,
  TREND_SUMMARY
FROM BRONZE.GOOGLE_TRENDS_RAW
LIMIT 5;
```

OUTPUT

Table | Chart — 5 rows — 47ms

| | QUERY | TREND_BREAKDOWN | TREND_SENTIMENT_SCORE | TREND_SUMMARY |
|---|---|---|---|---|
| 1 | neil degrasse tyson | null | 0.0859375 | |
| 2 | tracy mcgrady | null | 0.0859375 | |
| 3 | jabari smith jr | jabari smith | 0.125 | jabari smith |
| 4 | is the voice on tonight | null | 0.0859375 | |
| 5 | sharks vs islanders | null | 0.0859375 | |

# Cortex Search & Cortex Analyst

Built Cortex Search Service on AI-enriched Bronze data which enabled:

- Natural language trend discovery
- Semantic ranking using similarity scores

Our search returns:

- Query
- Category
- AI sentiment score
- AI-generated summary

Finally, output feeds into Cortex Analyst for BI exploration

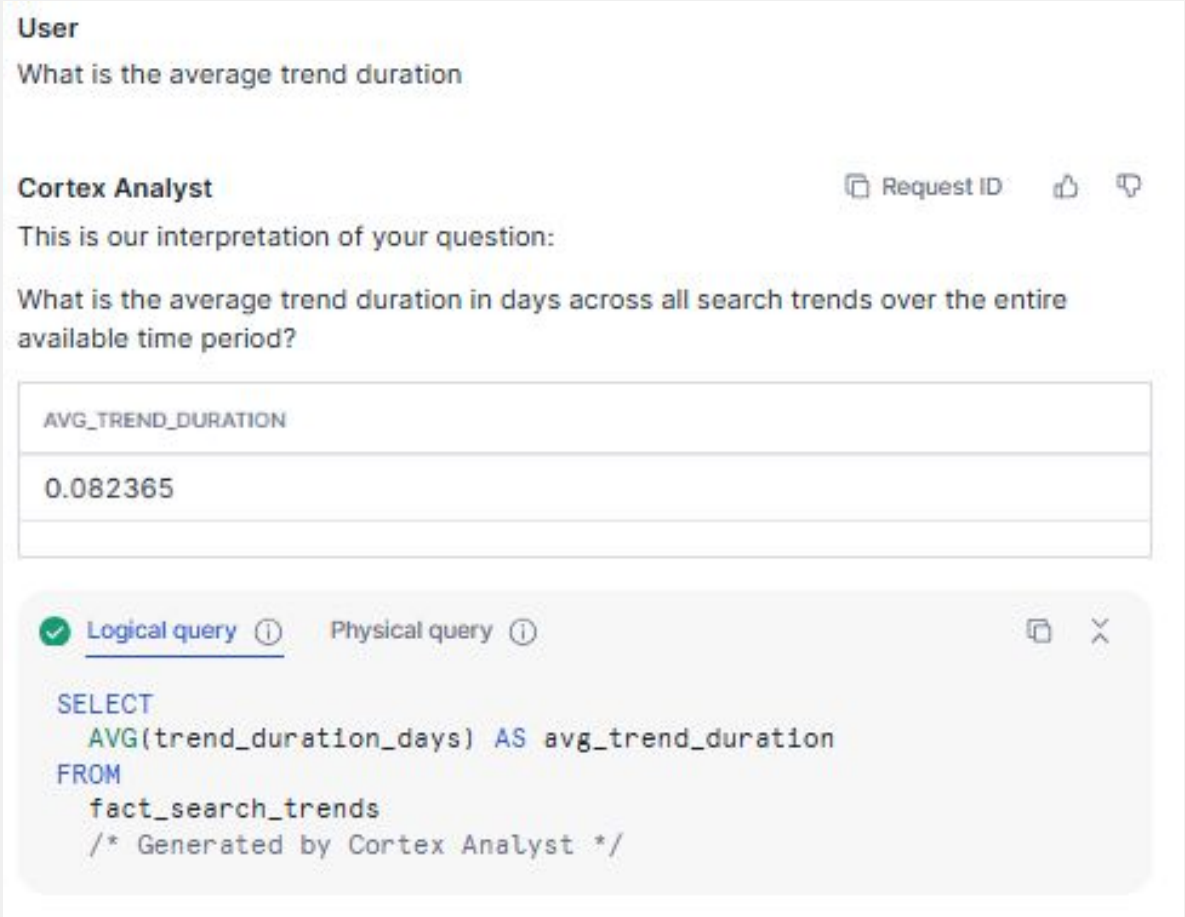
```
USE ROLE ROLE_Team_JAS;
USE DATABASE DB_Team_JAS;
USE SCHEMA BRONZE;
USE WAREHOUSE Animal_Task_WH;

SELECT PARSE_JSON(
  SNOWFLAKE.CORTEX.SEARCH_PREVIEW(
    'DB_Team_JAS.BRONZE.GOOGLE_TRENDS_CSS',
    '{
      "query": "holiday shopping",
      "columns": ["QUERY", "CATEGORIES", "TREND_BREAKDOWN", "TREND_SENTIMENT_SCORE", "TREND_SUMMARY"],
      "limit": 5
    }'
  )
)['results'];
 %I to generate
```

Cortex Search Output

| Rank | Query | Category | Sentiment | Short AI Summary |
|---|---|---|---|---|
| 1 | holiday shopping | Retail, Shopping | -0.18 | Spike in searches due to holiday promotions |
| 2 | are wells fargo banks closing | Business and Finance | 0.18 | october 13 holiday |
| 3 | starbucks holiday menu 2025 hello kitty | Food and Drink | 0.28 | starbucks holiday drinks |
| 4 | is today a federal holiday | Hobbies and Leisure | 0.28 | Questions about whether Oct 13 is a holiday, banks/mail/school status |
| 5 | yogurt shop murders documentary | Law and Government | 0.24 | “shop” |

← Query

# Cortex Search & Cortex Analyst



# Incremental File Upload

New Google Trends files loaded incrementally into Bronze

Only new records inserted into:

- DIM_DATE
- DIM_QUERY
- DIM_CATEGORY
- BRIDGE_QUERY_CATEGORY
- FACT_SEARCH_TRENDS
-

Prevented duplicates using:

- Query + date matching

Gold layer recomputed after each load

Bronze before

| | BRONZE_ROWS_BEFORE |
|---|---|
| 1 | 16555 |

Bronze after

| | BRONZE_ROWS_AFTER |
|---|---|
| 1 | 16557 |

Silver after

| | COUNT(*) |
|---|---|
| 1 | 16555 |

Gold Aggregate after refresh

Results (just now)

Table | Chart | 5 rows | 319ms

| | CATEGORY_NAME | NUMBER_OF_UNIQUE_TRENDS | TOTAL_SEARCH_VOLUME | AVG_GROWTH_PCT |
|---|---|---|---|---|
| 1 | Sports | 5975 | 132228800 | 255.769009217 |
| 2 | Entertainment | 2538 | 29676800 | 161.212976023 |
| 3 | Law and Government | 999 | 9933000 | 188.954270343 |
| 4 | Other | 856 | 7288900 | 175.180505415 |
| 5 | Politics | 596 | 5934800 | 167.593612335 |

# Q & A