



Music Trends and User Search

Shadi Farzankia, Mina Aminpour, Nahid Taherkhani, Syed muhammad Ali Fatmi, Medhansh Ahuja, Tharun Johny Mekala, Ahsan

Behind the Brainstorm: The Birth of Our Project

- conducted a small NLP analysis on our query texts
- extracted keywords and words most repeated with
- analysed their frequency
- voted at the end

```
→ Top 20 Words and Their Most Frequent Co-occurring Words:  
  
http: f (8001), e (7347), com (7096), www (4309), b (3688)  
new: york (41762), jersey (17619), city (7953), state (4525), mexico (3588)  
free: game (5579), download (4817), online (4688), pic (4127), music (3139)  
school: high (28219), district (7307), county (5441), middle (4506), public (4382)  
county: school (5441), court (4811), orange (3389), florida (3202), record (3019)  
com: www (8322), http (7096), game (1060), free (725), disney (571)  
lyric: song (5890), love (3490), im (2425), dont (1821), like (1522)  
home: sale (7694), funeral (4262), new (3528), mobile (3294), page (2344)  
picture: free (2720), nude (1463), woman (1086), girl (1038), sex (1011)  
florida: county (3202), beach (3069), orlando (2325), south (2038), home (1730)  
city: new (7953), york (6910), kansa (3134), atlantic (2182), school (1825)  
state: united (6360), new (4525), york (3488), park (3312), university (2756)  
sale: home (7694), house (2834), land (2076), used (1909), county (1881)  
www: com (8322), http (4309), myspace (415), org (355), game (305)  
hotel: beach (2623), new (2044), la (1837), city (1539), vega (1399)  
texas: houston (2778), dallas (1962), county (1582), austin (1543), san (1309)  
game: free (5579), online (3857), video (2061), play (1733), cheat (1301)  
center: medical (4250), art (1160), new (1100), health (1006), community (1002)  
york: new (41762), city (6910), state (3488), school (1274), hotel (1190)  
car: used (3304), rental (3290), sale (1624), new (1263), dealer (1112)
```



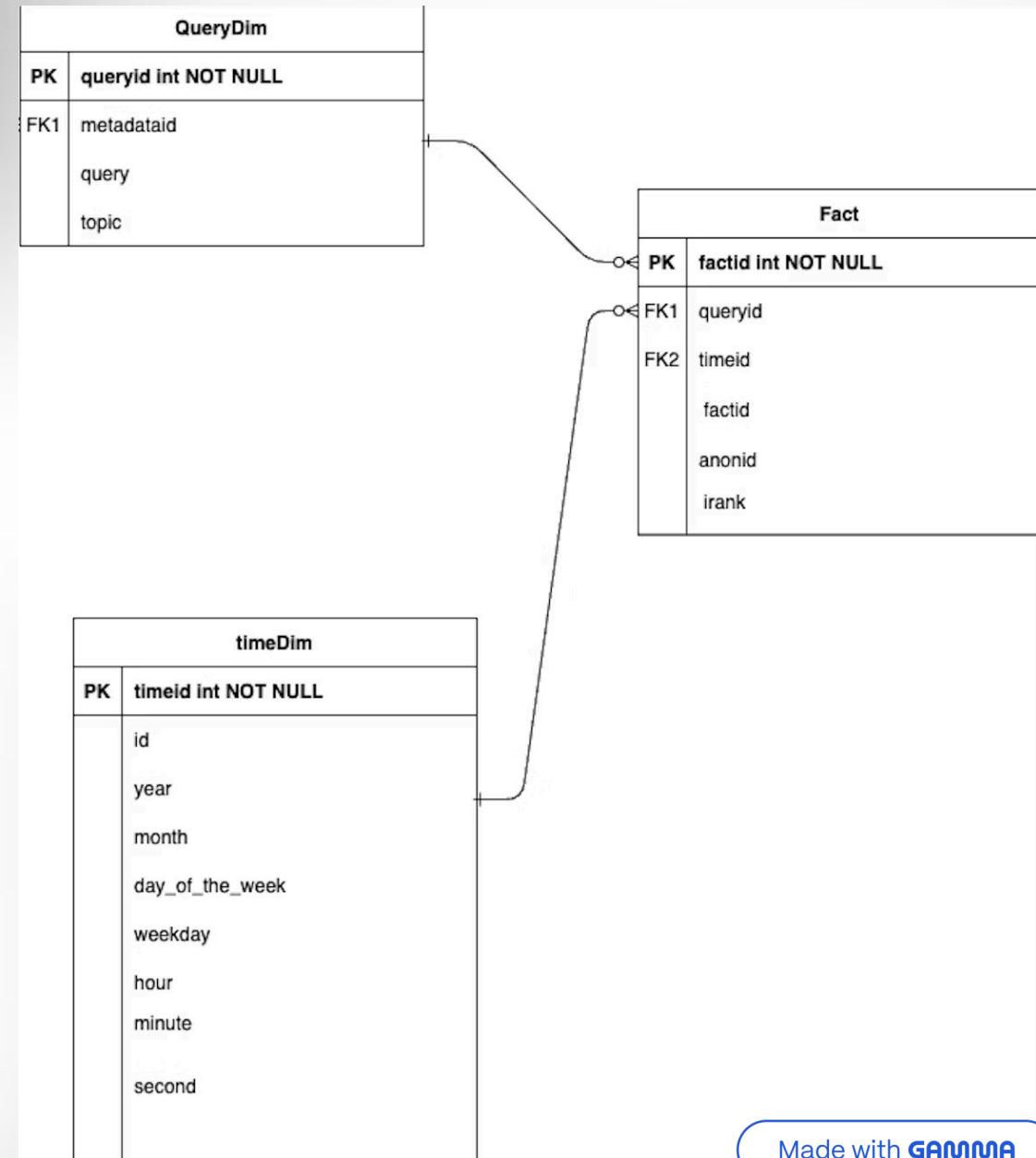
Our Schema Design:

To address our question, we required three tables from the virtual machine data:

- Facts
- Querydim
- Timedim

What We Needed:

- Singers
- Songs
- genres
- popularity of song



External Data:

Our analysis focuses exclusively on music data from the past 10 years.

1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006

The reason behind it:

- Recent hits strongly influence listeners and the industry
- Older data may dilute analysis with outdated trends

We utilized data provided by Dave Tompkins, Associate Professor at the University of Waterloo:

 cs.uwaterloo.ca



Music Database :: Dave Tompkins

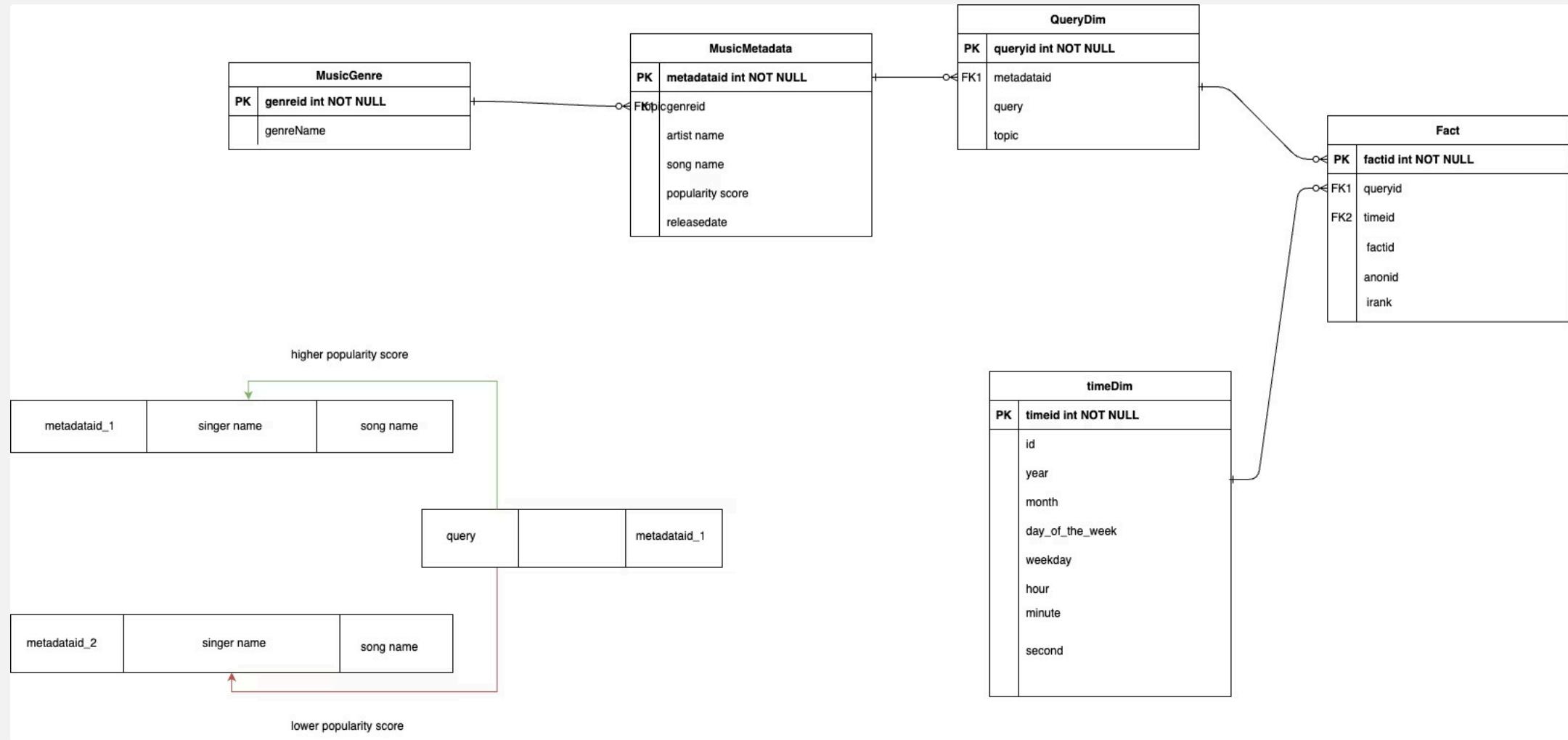
The music collection of Dave Tompkins will help you find and identify music. It is extensive, and can be browsed in a wide variety of ways.

We used web scraping to extract data about:

- | | |
|----------|--------------------|
| - Songs | - Popularity Score |
| - Singer | - Genre |



Data Extraction and Integration: Our Approach



Questions Guiding Our Analysis:

- **Question 1 =>** most popular musical platforms
- **Question 2 =>** What are the most popular days and times for music-related searches?
- **Question 3 =>** What is the correlation between popularity score and number of searches for?
- **Question 4 =>** How do concert search trends vary by month, weekday, and genre?
- **Question 5 =>** How long are the music-related search session on average for each day?



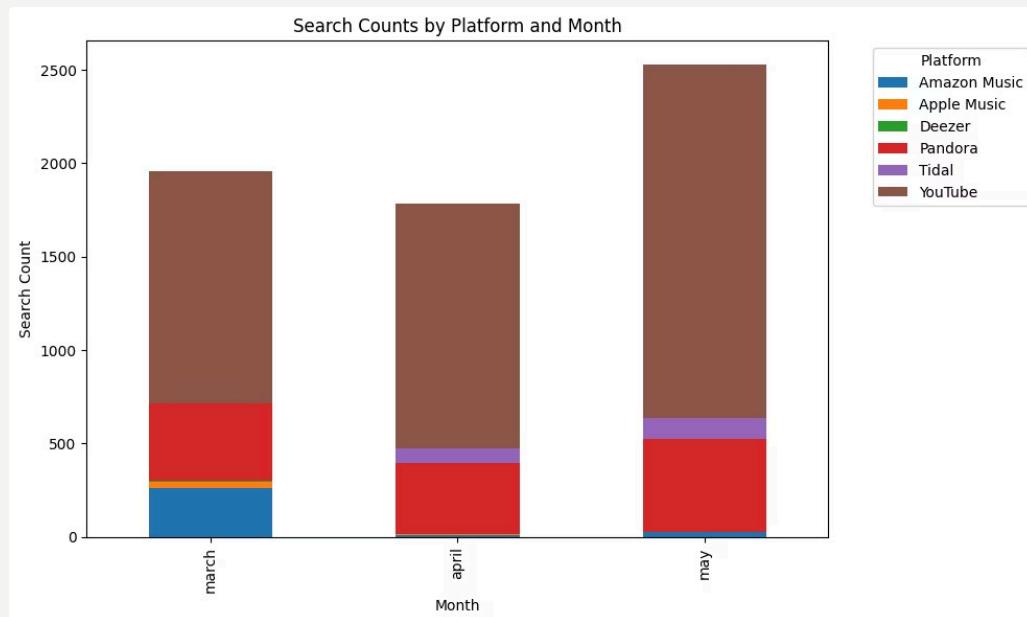
Q1: most popular musical platforms

- what were the most popular musical platforms in each month
- Used Operator: **DENSE_RANK()**
- About **descriptive analysis**
- Tables used to answer the question:
 - facts
 - timedim
 - querydim

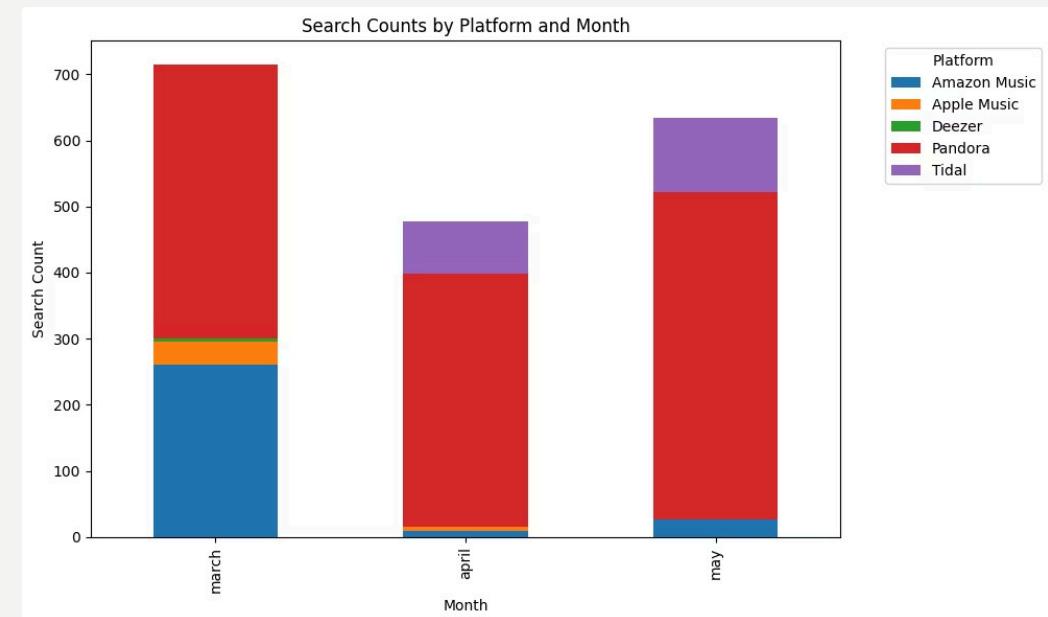
```
SELECT
  t.month AS month_name, -- Month of the search
  CASE
    WHEN q.query ILIKE '%Spotify%' THEN 'Spotify'
    WHEN q.query ILIKE '%YouTube%' THEN 'YouTube'
    WHEN q.query ILIKE '%Apple Music%' THEN 'Apple Music'
    WHEN q.query ILIKE '%Pandora%' THEN 'Pandora'
    WHEN q.query ILIKE '%Tidal%' THEN 'Tidal'
    WHEN q.query ILIKE '%Deezer%' THEN 'Deezer'
    WHEN q.query ILIKE '%SoundCloud%' THEN 'SoundCloud'
    WHEN q.query ILIKE '%Amazon Music%' THEN 'Amazon Music'
    ELSE 'Unknown'
  END AS platform_name,
  COUNT(f.factid) AS search_count,
  DENSE_RANK() OVER ( -- Rank within each month
    PARTITION BY t.month
    ORDER BY COUNT(f.factid) DESC
  ) AS rank
FROM
  aol_schema.querydim q
JOIN
  aol_schema.facts f ON q.id = f.queryid
JOIN
  aol_schema.timedim t ON f.timeid = t.id -- Join with the time dimension to get the month
WHERE
  q.query ILIKE ANY (
    ARRAY[
      '%Spotify%', '%YouTube%', '%Apple Music%',
      '%Pandora%', '%Tidal%',
      '%Deezer%', '%SoundCloud%', '%Amazon Music%'
    ]
  )
GROUP BY
  t.month, platform_name
ORDER BY
  t.month, rank;
```

Q1: Analysis

Considering Youtube:



Not considering Youtube:



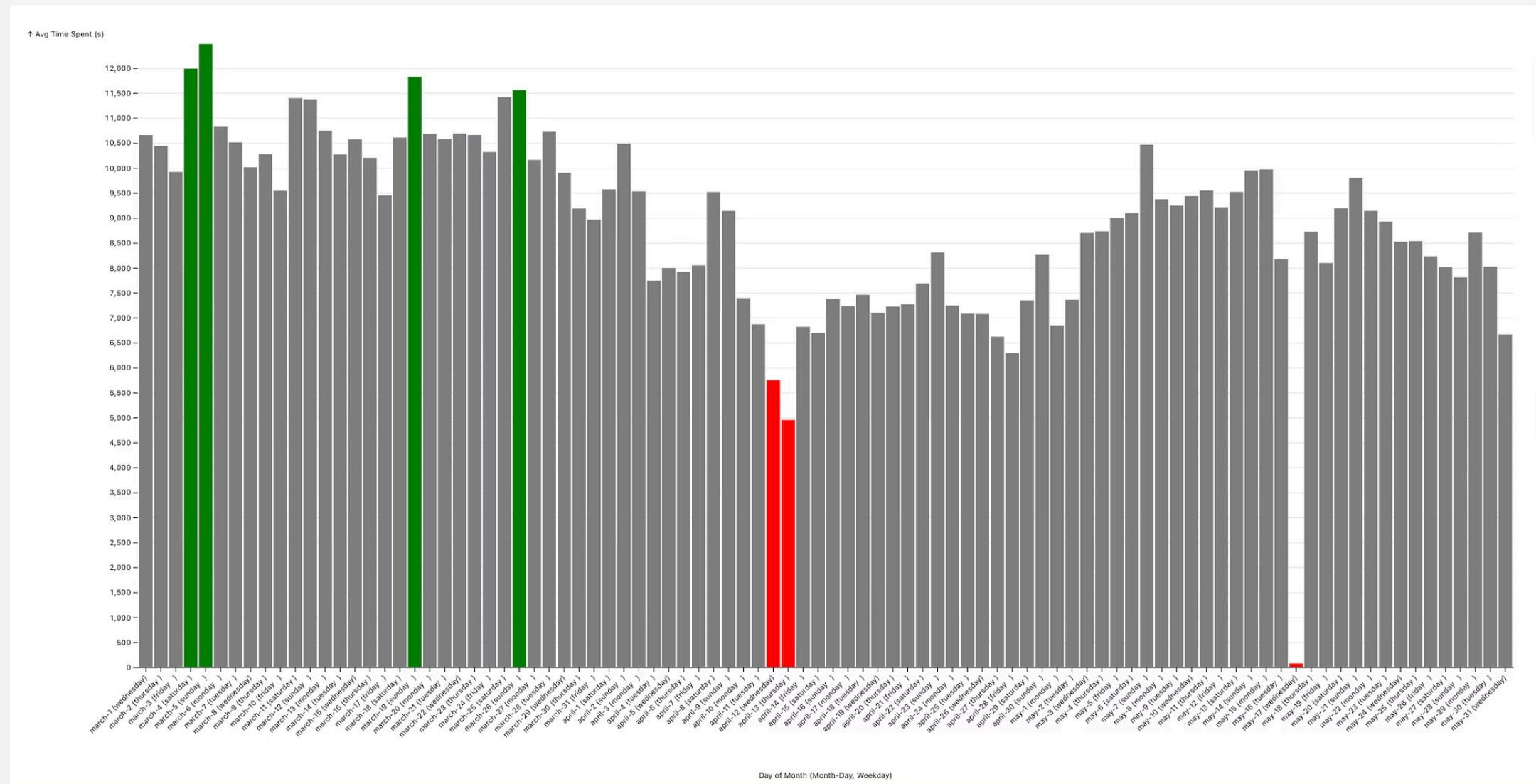
- YouTube consistently dominates search activity across all three months.
- No data for Spotify.
- Pandora was the second leading platform
- **Tidal:** Since Tidal was launched in 2014, it did not exist in 2006. Therefore, the data for Tidal in this context is inaccurate.
- march and then may has the highest search (possibly due to Grammy and Eruvision)

Q2: Peak times for searches

- **What are the most popular days and times for music-related searches?**
- Used Operator: **GROUP BY**
- About **descriptive analysis**
- Tables used to answer the question:
 - fact
 - timedim
 - querydim

```
SELECT
    t.month,
    t.day_of_the_month,
    t.weekday,
    COUNT(*) AS music_search_count
FROM
    aol_schema.facts f
JOIN
    aol_schema.querydim q ON f.queryid = q.id
JOIN
    aol_schema.timedim t ON f.timeid = t.id
WHERE
    (q.topic = 'music' OR q.metadataid IS NOT NULL) -- Filter for music-related queries
GROUP BY
    t.month, t.day_of_the_month, t.weekday
```

Q2: Analysis 1



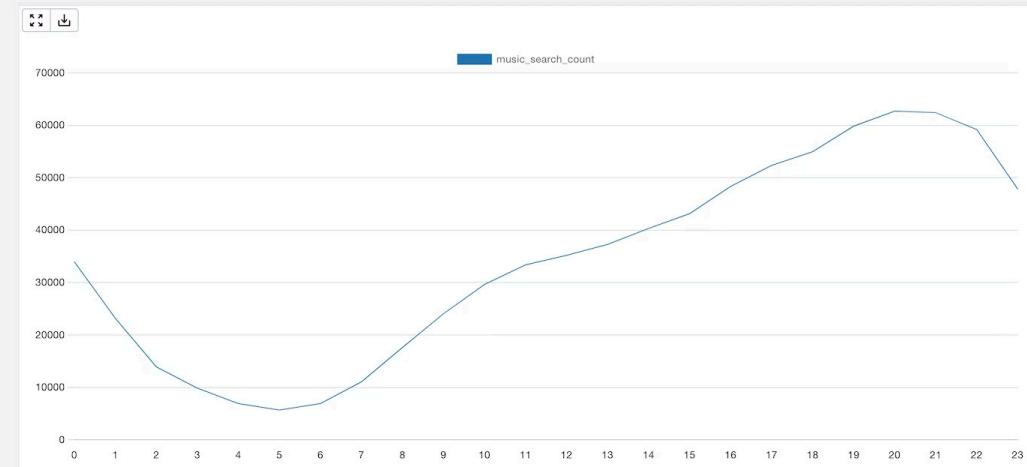
- Consistent Baseline of Music-Related Searches: **9,000 to 12,000** Daily
- Green Bars Indicate Peaks: Search Counts Exceeding **12,000** (maybe due to music release).
- The **red bars** indicate days with unusually low search counts below **6000**:
 - **12–13 April 2006:** powerful tornadoes in Iowa City, Iowa.
 - **17 May 2006:** 14th UEFA Champions League Final, where **Barcelona defeated Arsenal**.
- **March** is overall higher (**Grammy Awards**)
- **May** is also high (**Eurovision**)

Q2: Analysis 2

Days:

	weekday	music_search_count
1	sunday	128920
2	saturday	122138
3	monday	117932
4	wednesday	113234
5	thursday	113105
6	tuesday	113043
7	friday	111251

Hours:



- **Sunday:** highest number of searches (**128,920**), **Saturday (122,138)**.
- **Friday:** lowest search count (**111,251**)
- **Interpretation:**
- Music searches during weekdays appear relatively balanced

- **Between 2 AM and 6 AM:** lowest (typical sleep patterns when most people are inactive online)
- **Between 7 PM and 9 PM:** highest activity(Evening leisure hours when people have free time to explore music-related content)
- **After 9 PM:** search activity gradually declines, dropping sharply after **11 PM** (people preparing for bed or wrapping up their daily activities)

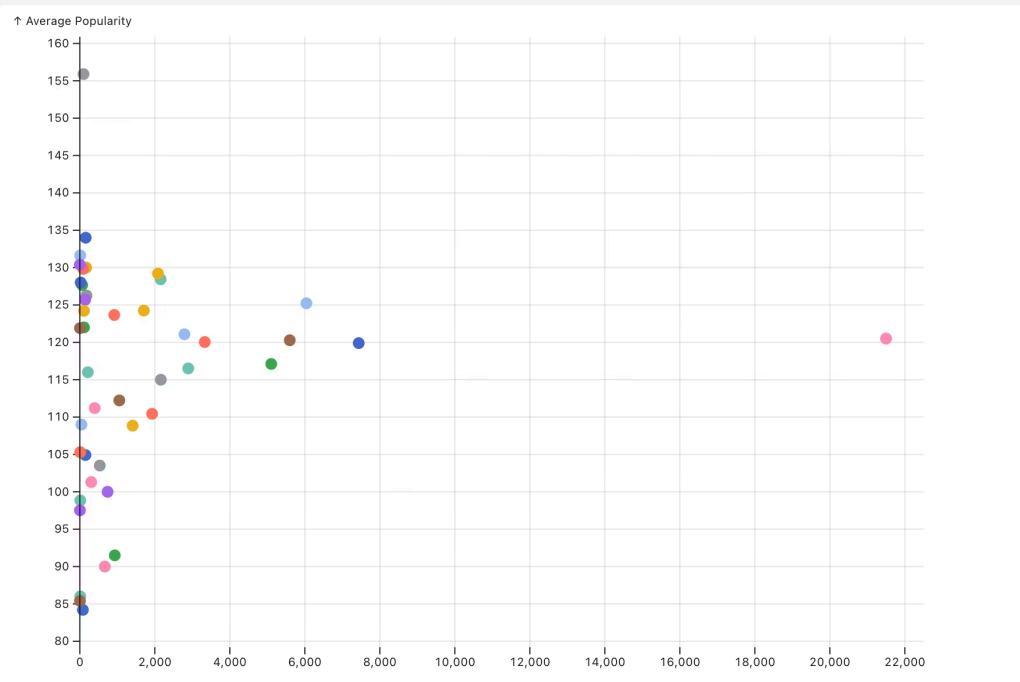
Q3: Popularity-search correlation analysis

- What is the correlation between popularity score and number of searches for genres?
- Used Operator: **GROUP BY, ORDER BY**
- About **inferential analysis**
- Tables used to answer the question:
 - facts
 - querydim
 - music_metadata
 - music_genres

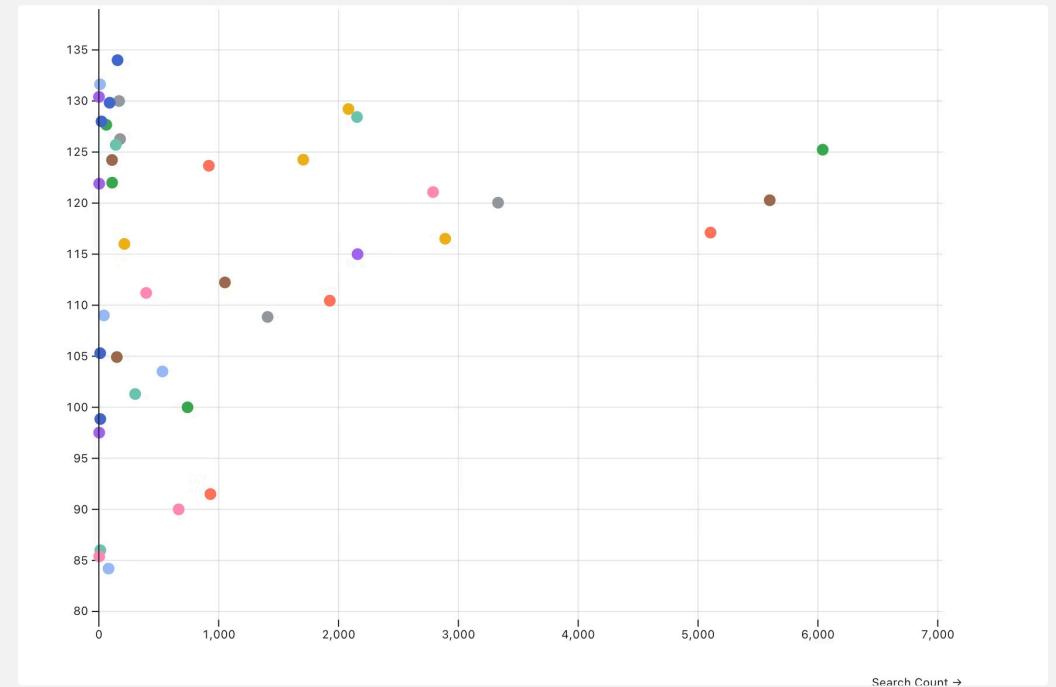
```
WITH GenrePopularityByYear AS (
    -- Calculate average popularity for each genre in 2006
    SELECT
        g.genresname AS genre_name, -- Genre name
        ROUND(CAST(AVG(m.popularityscore) AS NUMERIC), 2) AS avg_popularity -- Average popularity for the genre in that year
    FROM
        aol_schema.music_metadata m
    JOIN
        aol_schema.music_genres g ON m.genresid = g.genresid
    WHERE
        m.releasedate = 2006 -- Consider only the year 2006
    GROUP BY
        g.genresname
),
GenreSearchCounts AS (
    -- Calculate search counts for each genre
    SELECT
        g.genresname AS genre_name, -- Genre name
        COUNT(f.factid) AS search_count -- Count the number of searches for each genre
    FROM
        aol_schema.facts f
    JOIN
        aol_schema.querydim q ON f.queryid = q.id
    JOIN
        aol_schema.music_metadata m ON q.metadataid = m.metadataid
    JOIN
        aol_schema.music_genres g ON m.genresid = g.genresid
    WHERE
        m.releasedate = 2006 -- Consider only the year 2006
    GROUP BY
        g.genresname
)
-- Combine popularity and search counts
SELECT
    gpby.genre_name, -- Genre name
    gpby.avg_popularity, -- Average popularity score for the genre
    gsc.search_count -- Total search count for the genre
FROM
    GenrePopularityByYear gpby
JOIN
    GenreSearchCounts gsc ON gpby.genre_name = gsc.genre_name
ORDER BY
    gsc.search_count DESC; -- Order by average popularity descending
```

Q3: Analysis

Whole Scatterplot:



Without Considering Outliers:



- We have two outliers:
 - **search count around 22,000** and an average popularity (~120)
 - A point near the top of the Y-axis (~155 popularity) with a **low search count (<1,000)**

- The scatterplot suggests a weak positive correlation between popularity and search count
- Overall, **popularity is not solely driven by the number of searches**
- Other factors: critical acclaim, niche appeal, or the cultural impact of the genre.

Q4: Concert query peak days

- How do concert search trends vary by month, weekday, and genre?
- Used Operator: **CUBE**
- About **descriptive analysis**
- Tables used to answer the question:
 - facts
 - timedim
 - querydim

```
SELECT
    t.month,
    t.weekday,
    CASE
        WHEN q.query ILIKE '%rock%' THEN 'Rock'
        WHEN q.query ILIKE '%pop%' THEN 'Pop'
        WHEN q.query ILIKE '%jazz%' THEN 'Jazz'
        WHEN q.query ILIKE '%hip-hop%' OR q.query ILIKE '%hiphop%' THEN 'Hip-Hop'
        WHEN q.query ILIKE '%classical%' THEN 'Classical'
        WHEN q.query ILIKE '%electronic%' OR q.query ILIKE '%edm%' THEN 'Electronic'
        ELSE 'Other'
    END AS genre,
    COUNT(q.query) AS TotalQueries
FROM
    AOL_SCHEMA.FACTS f
JOIN
    AOL_SCHEMA.TIMEDIM t ON f.TIMEID = t.ID
JOIN
    AOL_SCHEMA.QUERYDIM q ON f.QUERYID = q.ID
WHERE
    (q.query ILIKE '%concert%' OR q.query ILIKE '%festival%')
    AND (
        q.query ILIKE '%rock%' OR
        q.query ILIKE '%pop%' OR
        q.query ILIKE '%jazz%' OR
        q.query ILIKE '%hip-hop%' OR
        q.query ILIKE '%classical%' OR
        q.query ILIKE '%electronic%' OR
        q.query ILIKE '%edm%'
    )
GROUP BY
    CUBE (t.month, t.weekday, genre)
ORDER BY
    TotalQueries DESC, t.month, t.weekday, genre;
```

Q4: Analysis

- The row where all dimensions (month, weekday, hour) are NULL represents the **grand total** of all concert related searches: **1761**.
- Month:**
 - March:** 655 searches.
 - May:** 639 searches.
- Weekday:**
 - Friday:** 312 searches.
 - Saturday:** 300 searches.
- Genre:**
 - Jazz:** 1145 searches
 - Rock:** **493**

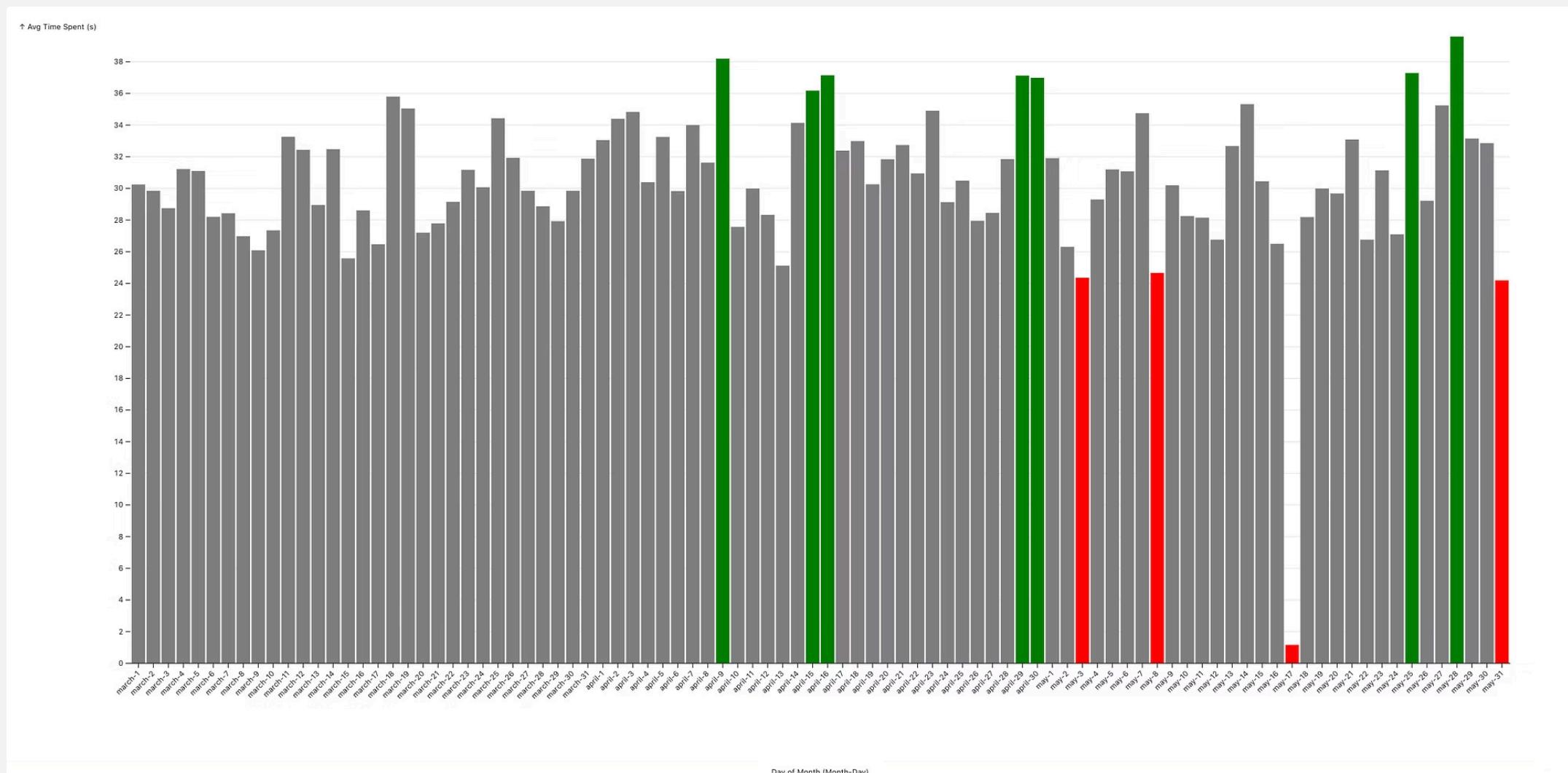
	month character varying (255)	weekday character varying (255)	genre text	totalqueries bigint
1	[null]	[null]	[null]	1761
2	[null]	[null]	Jazz	1145
3	march	[null]	[null]	655
4	may	[null]	[null]	639
5	[null]	[null]	Rock	493
6	april	[null]	[null]	467
7	march	[null]	Jazz	420
8	may	[null]	Jazz	412
9	april	[null]	Jazz	313
10	[null]	friday	[null]	312
11	[null]	saturday	[null]	300
12	[null]	thursday	[null]	266
13	[null]	monday	[null]	256
14	[null]	sunday	[null]	241
15	march	[null]	Rock	207
16	[null]	friday	Jazz	203
17	[null]	wednesday	[null]	197
18	[null]	tuesday	[null]	189
19	[null]	thursday	Jazz	184
20	may	[null]	Rock	182
21	[null]	monday	Jazz	180
22	[null]	saturday	Jazz	169
23	[null]	wednesday	Jazz	148
24	[null]	sunday	Jazz	145
25	march	thursday	[null]	135
26	may	saturday	[null]	131
27	[null]	tuesday	Jazz	116
28	[null]	saturday	Rock	111
29	may	friday	[null]	110
30	march	friday	[null]	107
31	april	[null]	Rock	104
32	march	thursday	Jazz	101
33	may	monday	[null]	99
34	march	monday	[null]	97
35	april	friday	[null]	95
36	[null]	friday	Rock	92
37	march	sunday	[null]	86
38	april	saturday	[null]	85
39	march	saturday	[null]	84
40	[null]	sunday	Rock	83
41	april	sunday	[null]	80
42	may	thursday	[null]	78

Q5: Music-related Session

- How long are the music-related search session on average for each day?
- Used Operator: **GROUP BY, GREATEST**
- about **descriptive analysis**
- Tables used to answer the question:
 - facts
 - timedim
 - querydim

```
WITH music_queries AS (
    -- Select all music-related queries
    SELECT
        f.anonid,
        f.timeid,
        t.day_of_the_year,
        t.month,
        t.day_of_the_month,
        t.hour,
        t.minute,
        t.second
    FROM
        aol_schema.facts f
    JOIN
        aol_schema.querydim q ON f.queryid = q.id
    JOIN
        aol_schema.timedim t ON f.timeid = t.id
    WHERE
        q.topic = 'music' OR q.metadataid IS NOT NULL -- Filter for music-related queries
),
user_sessions AS (
    -- Calculate session start and end time per user per day
    SELECT
        us.anonid,
        us.month,
        us.day_of_the_month,
        MIN(us.hour * 3600 + us.minute * 60 + us.second) AS session_start_seconds,
        MAX(us.hour * 3600 + us.minute * 60 + us.second) AS session_end_seconds
    FROM
        music_queries mq
    GROUP BY
        mq.anonid, mq.month, mq.day_of_the_month
),
session_durations AS (
    -- Calculate the session durations per user per day (in seconds)
    SELECT
        us.anonid,
        us.month,
        us.day_of_the_month,
        GREATEST(us.session_end_seconds - us.session_start_seconds, 0) AS session_duration_seconds
    FROM
        user_sessions us
),
-- Calculate the average session duration per day
SELECT
    sd.month,
    sd.day_of_the_month,
    ROUND(AVG(sd.session_duration_seconds) / 60, 2) AS average_session_duration_minutes
FROM
    session_durations sd
GROUP BY
    sd.month, sd.day_of_the_month
ORDER BY
    sd.month, sd.day_of_the_month;
```

Q5: Analysis



- Stable average around **25–36 seconds**
- **12–13 April 2006: powerful tornadoes in Iowa City, Iowa.**

Improvements for Future Queries:

- Incorporate more granular event information (e.g., album release dates, concert announcements) for better contextual analysis.
- Include location-specific data to analyze how trends vary across regions, providing insights into localized preferences for platforms, genres, or artists.
- Incorporate data from other years to understand long-term trends and changes in user behavior, platform popularity, and genre preferences. This would help track the evolution of music consumption patterns over time.

Thanks for Listening

