2006 Fifa World Cup in AOL Dataset

27th November 2023

Eric Jonas, MD. Mostofa Kamal, Tania Sultana, Ahmed Dider Rahat

Agenda

1. External data
2. Extract-Transform-Load
3. Schema
4. Queries
5. Challenges
6. Questions

External data

Dataset	Used columns	Source
Fifa World Cup 2006 Country Statistics	RangeIndex: 32 entries, 0 to 31 Data columns (total 2 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 position 32 non-null int64 1 team 32 non-null object	Kaggle
Fifa World Cup Player Statistics	Data columns (total 4 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 match_id 37784 non-null int64 1 team_initials 37784 non-null object 2 player_name 37784 non-null object 3 event 9069 non-null object	Kaggle
Fifa World Cup Match Statistics	Data columns (total 12 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 year 852 non-null int64 1 datetime 852 non-null object 2 stadium 852 non-null object 3 city 852 non-null object 4 home_team_name 852 non-null object 5 home_team_goals 852 non-null int64 6 away_team_goals 852 non-null int64 7 away_team_name 852 non-null object 8 attendance 850 non-null float64 9 match_id 852 non-null int64 10 home_team_initials 852 non-null object 11 away_team_initials 852 non-null object	Github, Kaggle

Extract and transform

```python
# convert camel to snake
def convert_to_snake_case(text: str) -> str:
    text = text.strip()
    return ''.join(['_' + s.lower() if (s.isupper() and i > 0 and text[i - 1] != ' ' and not text[i - 1].isupper()) \
                    else '_' if s == ' ' \
                    else s.lower() for i, s in enumerate(text)])

# lower all string columns
def lower_all_object_columns_of_df(df: pd.DataFrame):
    string_columns = df.select_dtypes(include='object').columns
    df[string_columns] = df[string_columns].apply(lambda x: x.str.lower().str.strip())
```

Custom cleansing functions

```python
world_cup_players = pd.read_csv("../../data/input_data/world_cup_players.csv",
                                usecols=['MatchID', 'Team Initials', 'Player Name','Event'])

world_cup_matches = pd.read_csv("../../data/input_data/world_cup_matches.csv",
                                usecols=['MatchID', 'Year', 'Datetime', 'Stadium', 'City', 'Home Team Name','Home Team Goals', 'Away Team Goals', 'Away Team Name',
                                         'Attendance', 'Home Team Initials', 'Away Team Initials'])

world_cup_countries = pd.read_csv("../../data/input_data/fifa_countries_2006.csv",
                                  usecols=['Position', 'Team'])

world_cup_players.columns = [convert_to_snake_case(col) for col in world_cup_players.columns]
world_cup_matches.columns = [convert_to_snake_case(col) for col in world_cup_matches.columns]
world_cup_countries.columns = [convert_to_snake_case(col) for col in world_cup_countries.columns]
lower_all_object_columns_of_df(world_cup_players)
lower_all_object_columns_of_df(world_cup_matches)
lower_all_object_columns_of_df(world_cup_countries)
```

Import data

```python
# clean world_cup_players df and add data to
world_cup_players = world_cup_players.rename(columns={'team_initials': 'player_team_initials'})
world_cup_players['event'] = world_cup_players['event'].fillna('')
world_cup_players['red_cards'] = world_cup_players['event'].str.count('r')
world_cup_players['yellow_cards'] = world_cup_players['event'].str.count(r'(?<!rs)y')
world_cup_players['goals'] = world_cup_players['event'].str.count('g')
# only for 2006 players
world_cup_players['player_name'] = world_cup_players['player_name'].replace(['uma\x00a m.', 'bola\x00os c.', 'nu\x00ez v.', 'c\x00ceres', 'acu\x00a', 'ca\x00iza',
       'nu\x00ez', 'caba\x00as', 'alvb\x00ge', 'k\x00llstr\x00m', 'allb\x00ck',
       'jo\x00o ricardo', 'andr\x00 macanga', 'akw\x00', 'sim\x00o', 'z\x00 kalanga',
       'loc\x00', 'lam\x00', 'fl\x00vio', 'm\x00rio', 'zuberb\x00hler', 'l\x00cio', 'kak\x00',
       'z\x00 roberto', 'luis\x00o', 'ca\x00izares'], ['umaña m.', 'bolaños c.', 'nuñez v.', 'céceres', 'acuña', 'cañiza',
 'nuñez', 'cabañas', 'alvbäge', 'kallström', 'allbäck',
 'joão ricardo', 'andrè macanga', 'akwá', 'simão', 'zé kalanga',
 'locó', 'lamá', 'flávio', 'mário', 'zuberbühler', 'lúcio', 'kaká',
 'zé roberto', 'luisão', 'cañizares'])
```

Data cleansing

Load

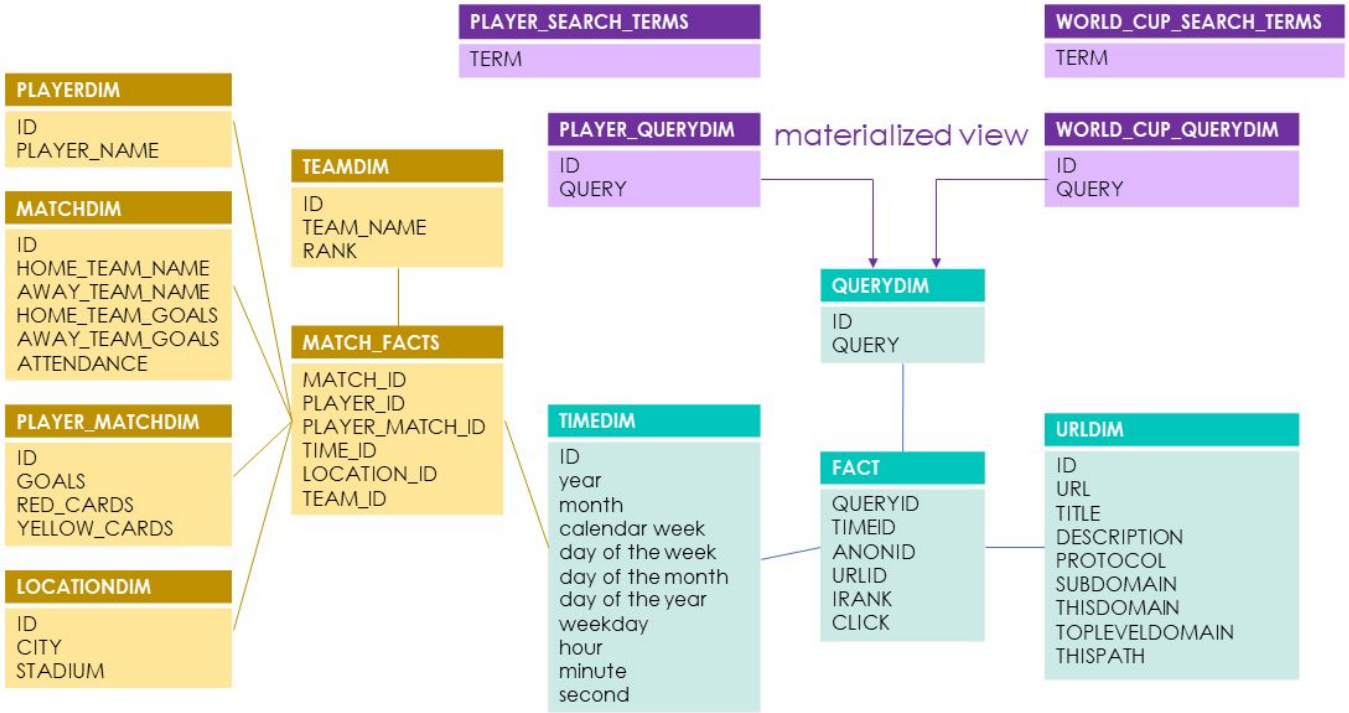
```
CREATE OR REPLACE TABLE AOL_SCHEMA.MATCH_FACTS (
    MATCH_ID DECIMAL(18) NOT NULL,
    PLAYER_MATCH_ID DECIMAL(18) NOT NULL,
    PLAYER_ID DECIMAL(18) NOT NULL,
    TIME_ID DECIMAL(18) NOT NULL,
    TEAM_ID DECIMAL(18) NOT NULL,
    LOCATION_ID DECIMAL(18) NOT NULL
);
IMPORT INTO AOL_SCHEMA.MATCH_FACTS
FROM LOCAL CSV FILE 'D:\Programmierung\wise_2324_bi_project\data\query_data\match_facts.csv';
```

Manual with sql scripts

```
import pyexasol
C = pyexasol.connect_local_config('standard_exasol', config_path='../../.pyexasol.ini', protocol_version = 1)


# create teamdim
C.execute("""CREATE OR REPLACE TABLE AOL_SCHEMA.MATCH_FACTS (
    MATCH_ID DECIMAL(18) NOT NULL,
    PLAYER_MATCH_ID DECIMAL(18) NOT NULL,
    PLAYER_ID DECIMAL(18) NOT NULL,
    TIME_ID DECIMAL(18) NOT NULL,
    TEAM_ID DECIMAL(18) NOT NULL,
    LOCATION_ID DECIMAL(18) NOT NULL
);""")
C.import_from_file('../../data/query_data/match_facts.csv', ("AOL_SCHEMA", "MATCH_FACTS"))
```

Automated with pyexasol and python scripts

Schema


PLAYER_SEARCH_TERMS
TERM
WORLD_CUP_SEARCH_TERMS
TERM
PLAYERDIM
ID
PLAYER_NAME
PLAYER_QUERYDIM
ID
QUERY
materialized view
WORLD_CUP_QUERYDIM
ID
QUERY
TEAMDIM
ID
TEAM_NAME
RANK
MATCHDIM
ID
HOME_TEAM_NAME
AWAY_TEAM_NAME
HOME_TEAM_GOALS
AWAY_TEAM_GOALS
ATTENDANCE
QUERYDIM
ID
QUERY
MATCH_FACTS
MATCH_ID
PLAYER_ID
PLAYER_MATCH_ID
TIME_ID
LOCATION_ID
TEAM_ID
PLAYER_MATCHDIM
ID
GOALS
RED_CARDS
YELLOW_CARDS
TIMEDIM
ID
year
month
calendar week
day of the week
day of the month
day of the year
weekday
hour
minute
second
FACT
QUERYID
TIMEID
ANONID
URLID
IRANK
CLICK
URLDIM
ID
URL
TITLE
DESCRIPTION
PROTOCOL
SUBDOMAIN
THISDOMAIN
TOPLEVELDOMAIN
THISPATH
LOCATIONDIM
ID
CITY
STADIUM

Materialized views

```python
def _missing_letter(s: str) -> list:
    typo_words = []
    for i in range(1, len(s) + 1):
        typo_words.append(s[:i - 1] + s[i:])
    return typo_words


def _insert_letter(s: str) -> list:
    typo_words = []
    for i in range(0, len(s) + 1):
        typo_words += [s[:i] + char + s[i:] for char in LETTERS]
    return typo_words


def _wrong_letter(s: str) -> list:
    typo_words = []
    for i in range(0, len(s)):
        typo_words += [s[:i] + char + s[i + 1:] for char in LETTERS]
    return typo_words
```

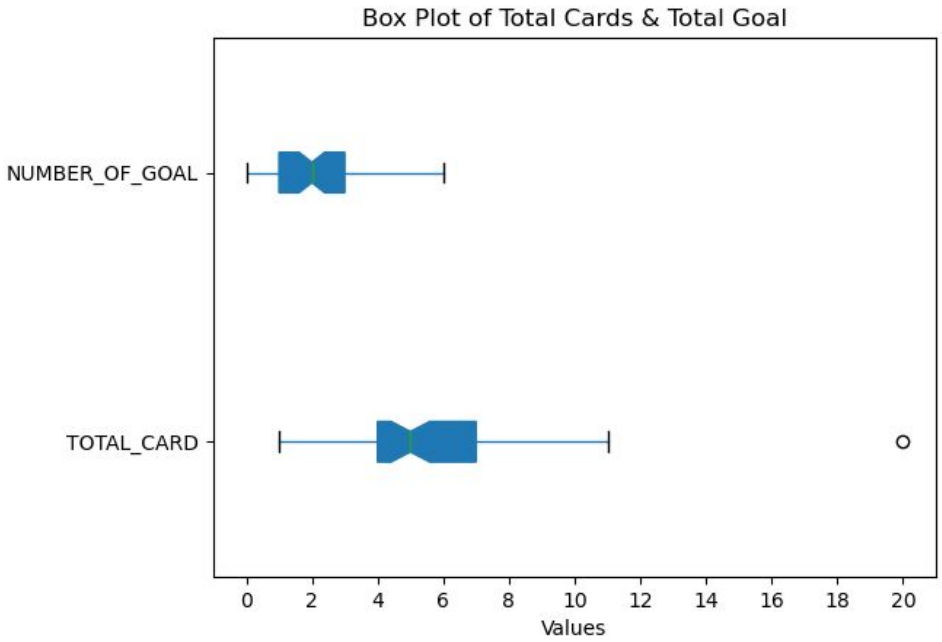
Generating search terms with typos

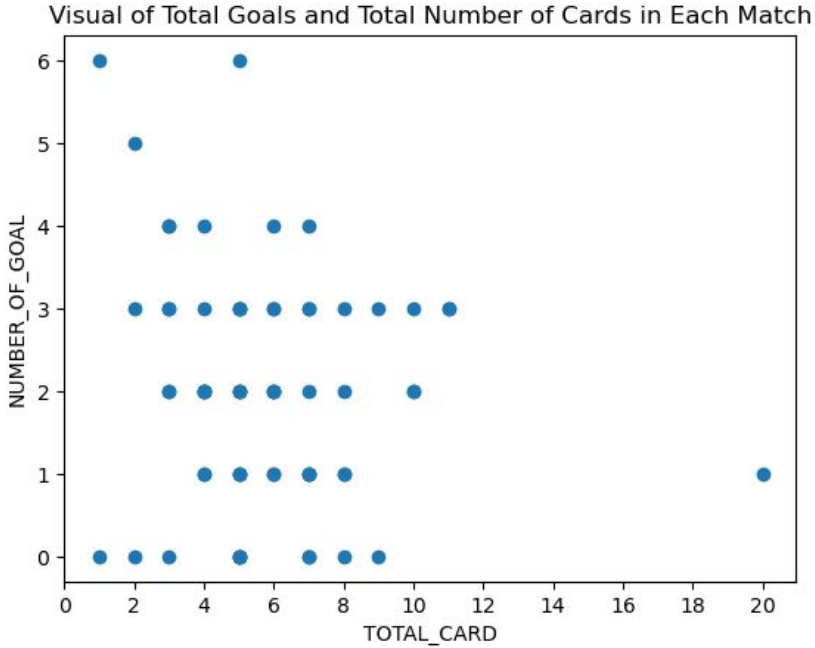
```sql
FUNCTION CHECK_WORDS_EXIST (input_string VARCHAR(1000), words_to_check VARCHAR(1000))
    RETURN BOOLEAN
    IS
        res BOOLEAN;
        pos INT;
        word VARCHAR(1000);
    BEGIN
        res := TRUE;
        pos := POSITION(' ' IN input_string);

        WHILE pos > 0
        DO
            word := SUBSTRING(input_string FROM 1 FOR pos - 1);

            IF POSITION(word IN words_to_check) = 0 THEN
                res := FALSE;
                RETURN res;
            END IF;

            input_string := SUBSTRING(input_string FROM pos + 1);
            pos := POSITION(' ' IN input_string);
        END WHILE;

        IF LENGTH(input_string) > 0 THEN
            IF POSITION(input_string IN words_to_check) = 0 THEN
                res := FALSE;
            END IF;
        END IF;

        RETURN res;
    END CHECK_WORDS_EXIST;
/

CREATE OR REPLACE TABLE AOL_SCHEMA.WC_QUERYDIM AS (
    SELECT qd.ID, qd.QUERY
    FROM AOL_SCHEMA.QUERYDIM AS qd
        INNER JOIN AOL_SCHEMA.WC_SEARCH_TERMS AS wc ON (
          CHECK_WORDS_EXIST(wc.TERM, qd.QUERY)
        )
);
```

Creating world cup querydim

QUERIES

Q:1 - Relationship between Number of Goal & Card in a Match During 2006 World Cup:





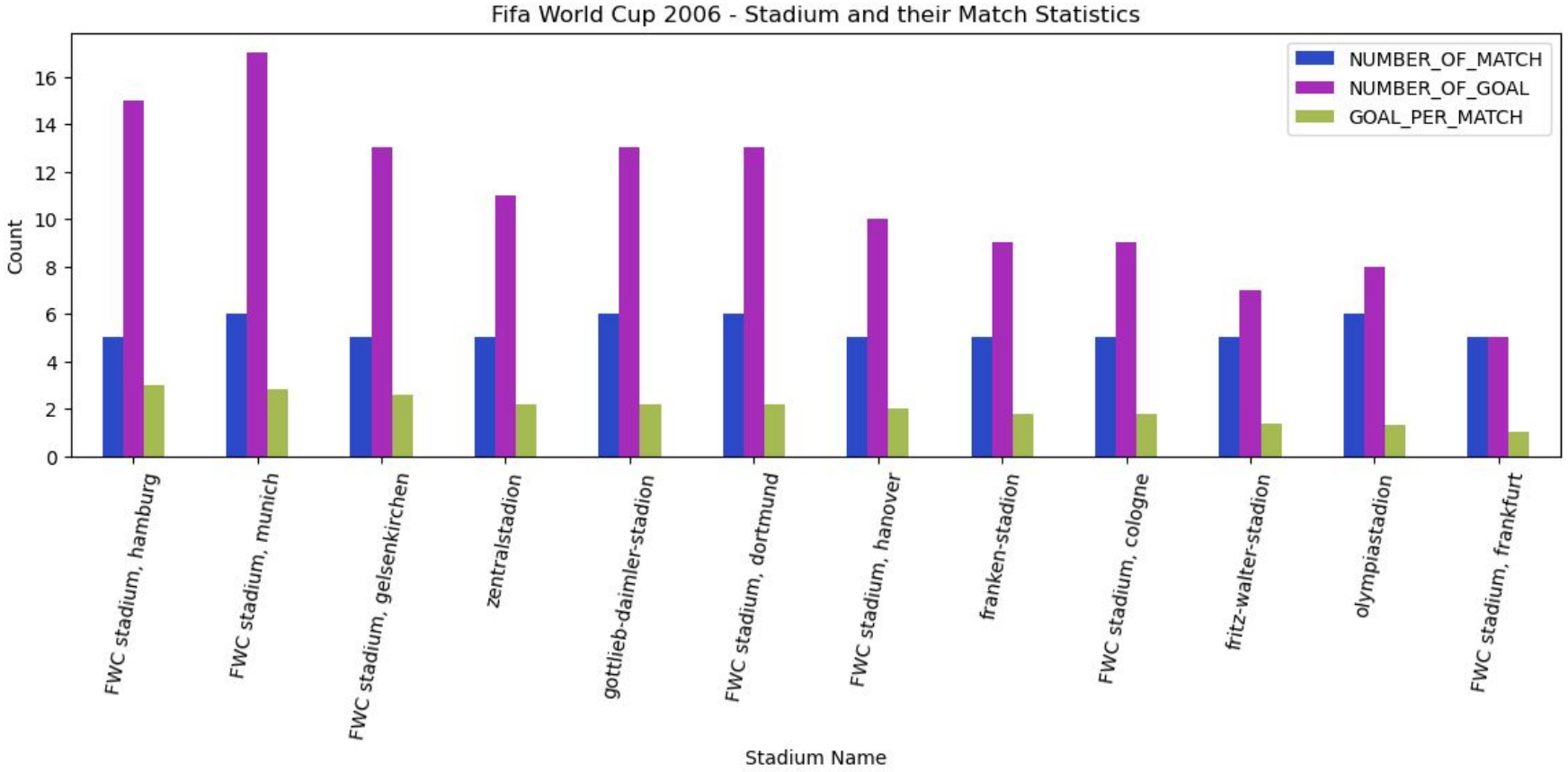

```
print(f'Correlation between goal and card: {df.NUMBER_OF_GOAL.corr(df.TOTAL_CARD)}')
```

```
Correlation between goal and card: -0.14656197504441218
```

Query - 1:

```
SELECT iq.MATCH_ID, iq.total_card, iq.number_of_goal

FROM (

    SELECT mf.MATCH_ID,

    SUM(PMD.RED_CARDS)+SUM(PMD.YELLOW_CARDS) AS total_card,

    SUM(pmd.GOALS) AS number_of_goal

    FROM AOL_SCHEMA.MATCH_FACTS mf

    JOIN AOL_SCHEMA.PLAYER_MATCHDIM pmd ON pmd.ID = mf.PLAYER_MATCH_ID

GROUP BY 1) AS iq

ORDER BY 1
```

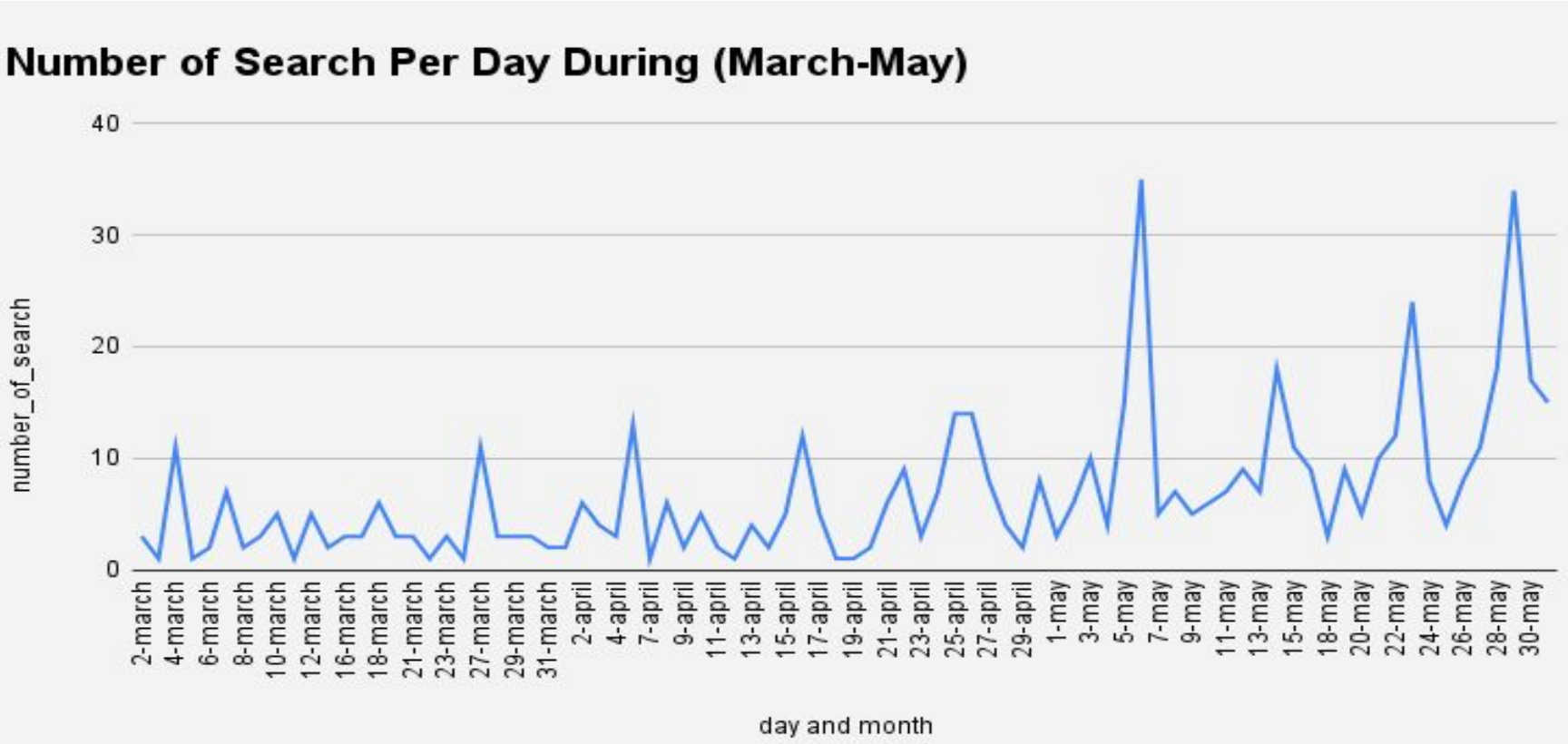
Q:2 - Basic Statistics of Fifa World Cup 2006 Stadiums:



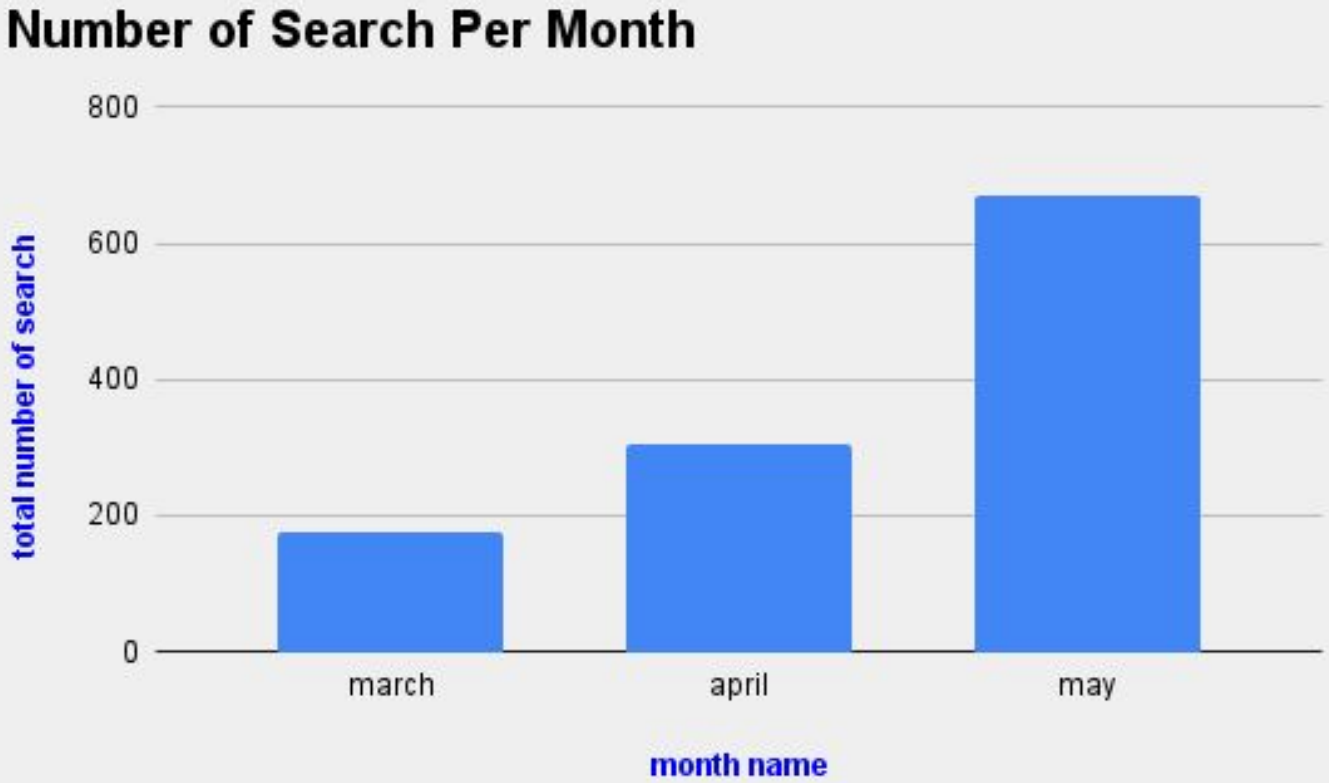
Query - 2:

```
SELECT loc.STADIUM,
    COUNT(DISTINCT mf.MATCH_ID) AS number_of_match,
    SUM(pmd.GOALS) AS number_of_goal,
    ROUND(SUM(pmd.GOALS) / COUNT(DISTINCT mf.MATCH_ID), 2) AS goal_per_match
FROM AOL_SCHEMA.MATCH_FACTS mf
JOIN AOL_SCHEMA.LOCATIONDIM loc ON loc.ID = mf.LOCATION_ID
JOIN AOL_SCHEMA.PLAYER_MATCHDIM pmd ON pmd.ID = mf.PLAYER_MATCH_ID
GROUP BY 1
ORDER BY 4 DESC
```

Q:3 Number of Times Users Searched for World Cup Related Queries (excluding those related to players)



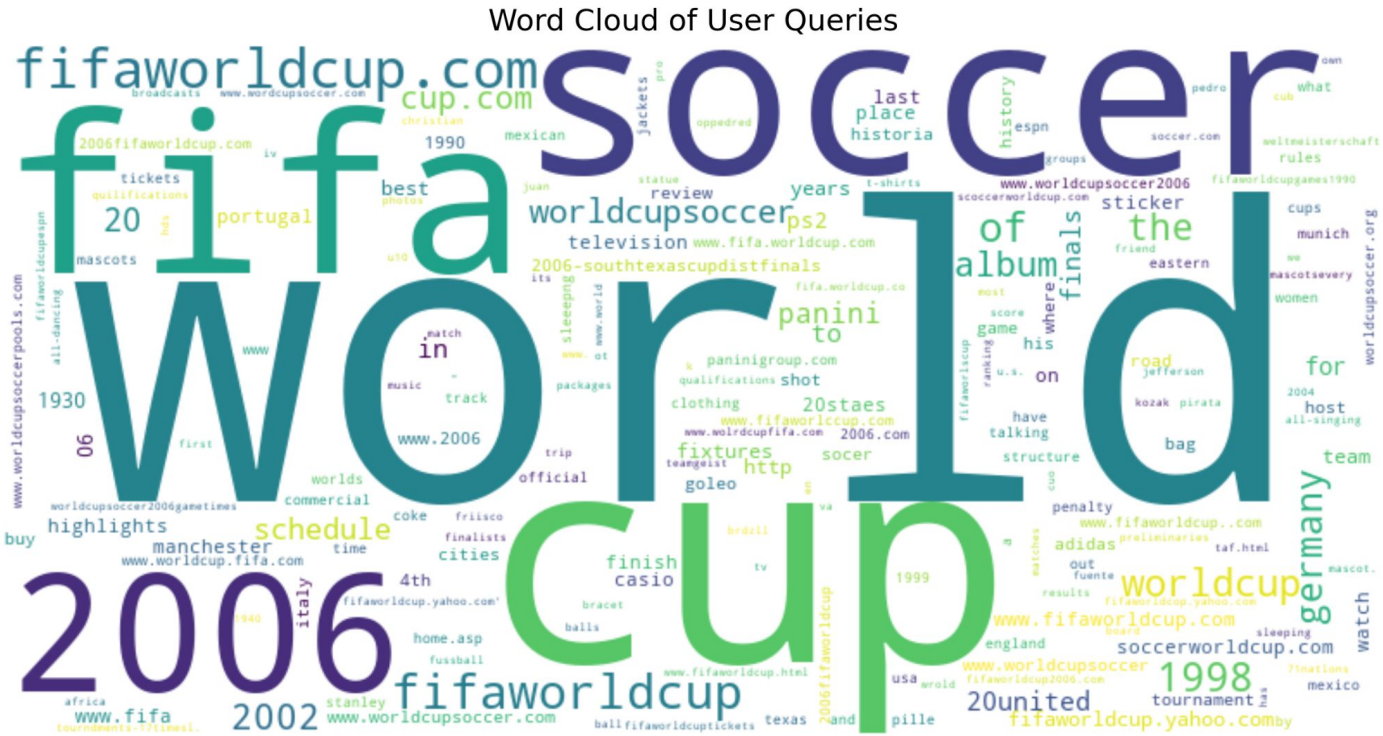
Q3: (Cont.) - Monthly Search Pattern



Query - 3 (a):

```
WITH
    SINGLE_QUERIES AS (
        SELECT ANONID, QUERYID, MIN(URLID) AS URLID, TIMEID FROM AOL_SCHEMA.FACTS
        GROUP BY ANONID, TIMEID, QUERYID)

SELECT t."day of the month", t."month", COUNT(*) AS searches
FROM AOL_SCHEMA.WC_QUERYDIM wq
JOIN SINGLE_QUERIES s ON wq.ID = s.QUERYID
JOIN AOL_SCHEMA.TIMEDIM t ON s.TIMEID = t.ID
WHERE t."month" IN ('april', 'march', 'may')
GROUP BY ROLLUP(t."month", t."day of the month") ORDER BY
    CASE
        WHEN t."month" = 'march' THEN 1
        WHEN t."month" = 'april' THEN 2
        WHEN t."month" = 'may' THEN 3
        ELSE 4
    END, t."day of the month";
```

Q3: (Cont.) - Word Cloud of Users Search Queries for Fifa World Cup 2006



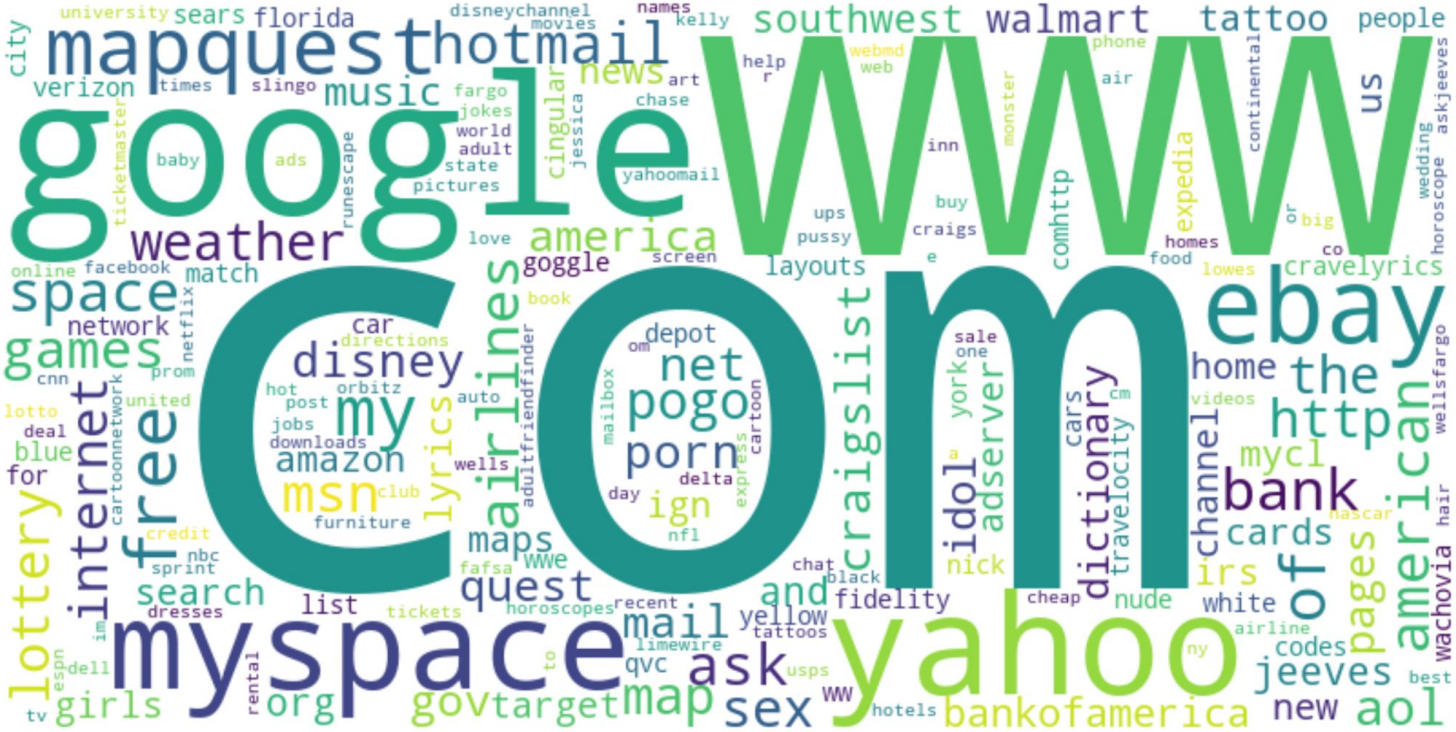
query	counts
world cup soccer	57
fifa world cup	38
fifaworldcup.com	35
soccer world cup	28
fifaworldcup	25
fifa world cup 1998	14
fifa world cup 2006	13
fifa world cup 2002	11
worldcupsoccer	11
2006 fifa world cup panini album	6

Query - 3 (b):

```
SELECT wq.QUERY, COUNT(DISTINCT f.TIMEID)

FROM AOL_SCHEMA.WC_QUERYDIM wq

JOIN AOL_SCHEMA.FACTS f ON wq.ID = f.QUERYID

GROUP BY 1

ORDER BY 2 DESC
```

Q:4 - User Search Queries Excluding Fifa World Cup queries during the same time-frame:

Word Cloud of User Queries Excluding Fifa World Cup

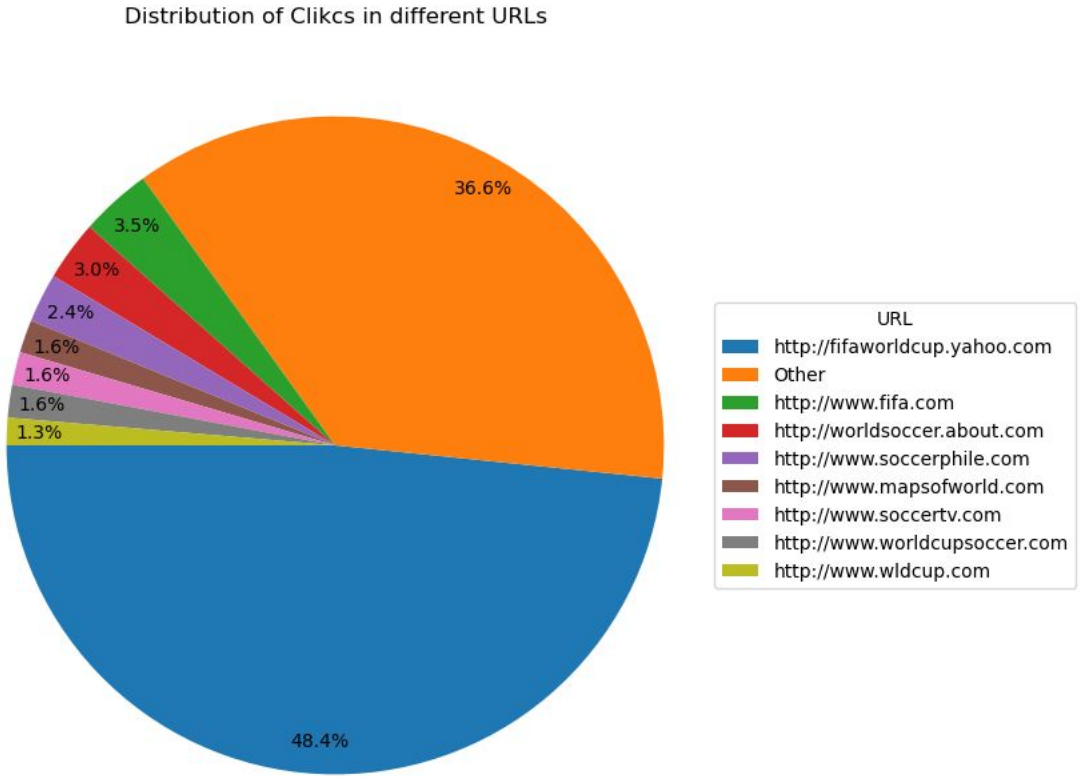


QUERY	COUNTS
google	302390
ebay	126592
yahoo	118891
yahoo.com	88941
mapquest	80495
google.com	72814
myspace.com	71079
myspace	69040
www.yahoo.com	39057
www.google.com	38853

Query - 4:

```
WITH
    SINGLE_QUERIES AS (
        SELECT DISTINCT tdm."day of the month", tdm."month"
        FROM AOL_SCHEMA.wc_querydim wqd
        JOIN AOL_SCHEMA.facts f ON wqd.ID = f.QUERYID
        JOIN AOL_SCHEMA.timedim tdm ON f.TIMEID = tdm.ID)
SELECT qd.QUERY query, COUNT(qd.ID) counts

FROM AOL_SCHEMA.QUERYDIM qd

JOIN AOL_SCHEMA.FACTS fct ON qd.ID = fct.QUERYID
JOIN AOL_SCHEMA.TIMEDIM td ON fct.TIMEID = td.ID
JOIN SINGLE_QUERIES SQ ON SQ."month" = td."month"
AND SQ."day of the month" = td."day of the month"

LEFT JOIN AOL_SCHEMA.wc_querydim wq ON wq.ID = qd.ID

WHERE wq.ID IS NULL AND qd.QUERY IS NOT NULL
GROUP BY 1
ORDER BY 2 DESC;
```

Q:5 - The Sites with the Highest Click Rates for All World Cup Queries:

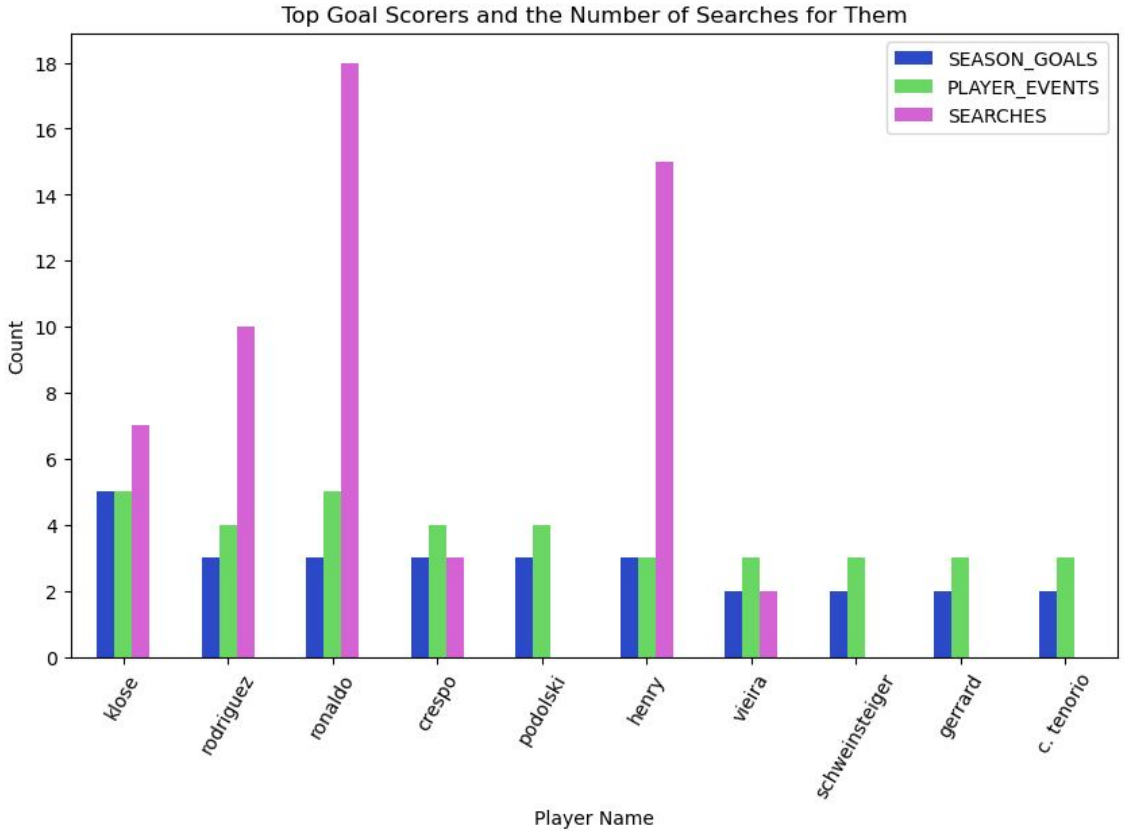


Total Unique Sites: 104

Query - 5:

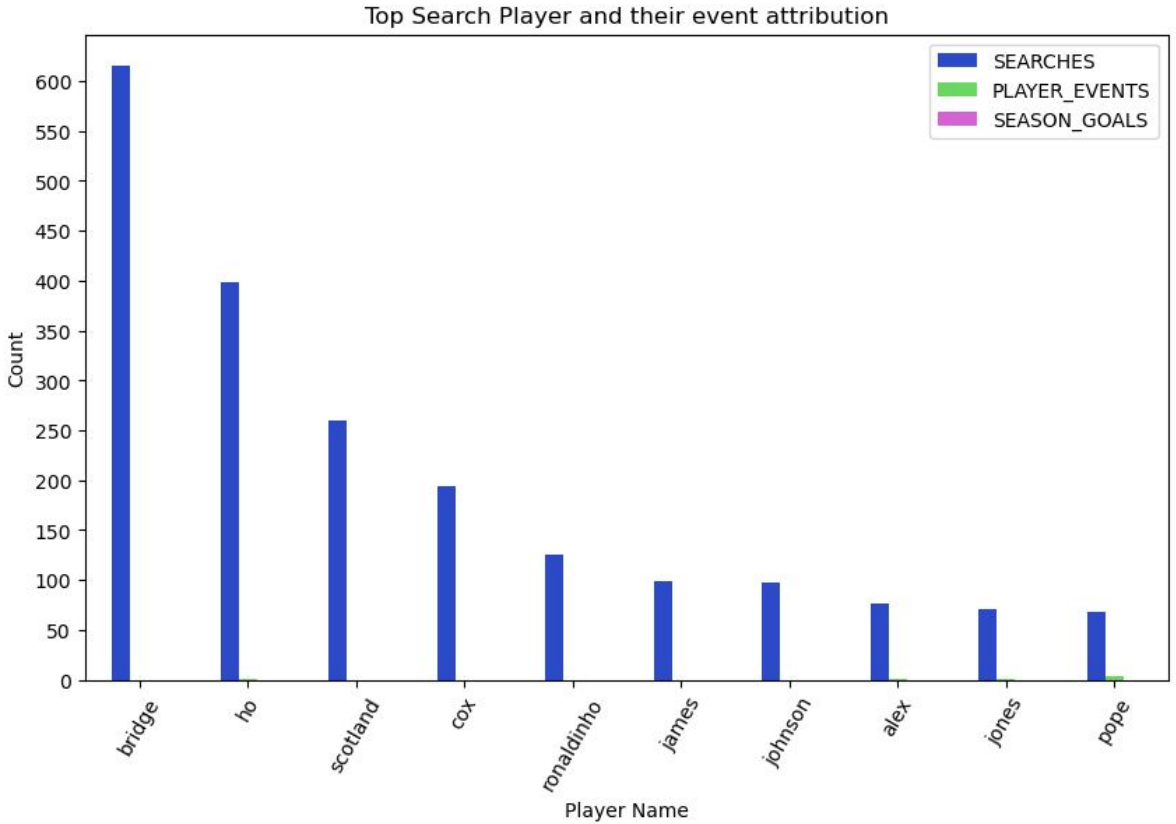
```
SELECT u.URL, u.THISDOMAIN, count(*) AS CLICKS,
        dense_rank() OVER (PARTITION BY CASE
        WHEN u.URL is null
             THEN 1
             ELSE 0
        END
        ORDER BY count(u.URL) DESC) AS URL_RANK,
        ROUND((COUNT(*) * 100.0 / SUM(COUNT(*)) OVER ()), 2) AS CLICK_PERCENTAGE
        FROM AOL_SCHEMA.WC_QUERYDIM wq
        JOIN AOL_SCHEMA.FACTS f ON wq.ID = f.QUERYID
        JOIN AOL_SCHEMA.URLDIM u ON f.URLID = u.ID
        WHERE u.THISDOMAIN IS NOT NULL
        GROUP BY GROUPING SETS ((u."THISDOMAIN"), (u."THISDOMAIN", u."URL"))
        ORDER BY url_rank, u.THISDOMAIN;
```

Q:6 - Top Goal Scorers and their search patterns:



PLAYER_NAME	SEASON_GOALS	SEARCHES	PLAYER_EVENTS
klose	5	7	5
rodriguez	3	10	4
ronaldo	3	18	5
crespo	3	3	4
podolski	3	0	4
henry	3	15	3
vieira	2	2	3
schweinsteiger	2	0	3
gerrard	2	0	3
c. tenorio	2	0	3

Q:6 (cont.) - Top User Search Player and their Event Contribution change



PLAYER_NAME	SEARCHES	PLAYER_EVENTS	SEASON_GOALS
bridge	616	0	0
ho	398	1	0
scotland	260	0	0
cox	194	0	0
ronaldinho	126	0	0
james	99	0	0
johnson	98	0	0
alex	76	1	0
jones	71	1	0
pope	68	3	0

Q:6 - Is user search correlated with the performance/event contribution of the player?

```
df['PLAYER_EVENTS'].corr(df['SEARCHES'])
```

```
-0.034868538631625 65
```

Query - 6:

```
WITH
    SINGLE_QUERIES AS (
        SELECT ANONID, QUERYID, MIN(URLID) AS URLID, TIMEID FROM AOL_SCHEMA.FACTS
        GROUP BY ANONID, TIMEID, QUERYID),

    PLAYER_PERFORMANCE AS (SELECT p.ID, p.PLAYER_NAME, SUM(pm.GOALS) SEASON_GOALS,
            SUM(pm.YELLOW_CARDS) SEASON_YELLOWS, SUM(pm.RED_CARDS) SEASON_REDS
     FROM AOL_SCHEMA.PLAYERDIM p JOIN AOL_SCHEMA.MATCH_FACTS mf ON p.ID = mf.PLAYER_ID
     JOIN AOL_SCHEMA.PLAYER_MATCHDIM pm ON pm.ID = mf.PLAYER_MATCH_ID GROUP BY p.ID, p.PLAYER_NAME)

SELECT p.PLAYER_NAME, SEASON_GOALS, SEASON_YELLOWS, SEASON_REDS, COUNT(pq.QUERY) SEARCHES,
     DENSE_RANK() OVER (PARTITION BY CASE WHEN p.PLAYER_NAME IS NULL THEN 0 ELSE 1 END
     ORDER BY COUNT(pq.QUERY) DESC) SEARCH_RANK,
     (SEASON_GOALS + SEASON_YELLOWS + SEASON_REDS) PLAYER_EVENTS,
     CASE WHEN p.PLAYER_NAME IS NULL THEN NULL
          ELSE DENSE_RANK() OVER (PARTITION BY
               CASE WHEN p.PLAYER_NAME IS NULL THEN 0 ELSE 1 END
               ORDER BY SEASON_GOALS + SEASON_YELLOWS + SEASON_REDS DESC)
     END PLAYER_EVENT_RANK
FROM AOL_SCHEMA.PLAYER_QUERYDIM pq JOIN SINGLE_QUERIES s ON pq.ID = s.QUERYID

RIGHT JOIN PLAYER_PERFORMANCE p ON TRIM(pq.QUERY) = p.PLAYER_NAME
GROUP BY CUBE (p.PLAYER_NAME, SEASON_GOALS, SEASON_YELLOWS, SEASON_REDS)

HAVING p.PLAYER_NAME IS NOT NULL AND SEASON_GOALS IS NOT NULL AND SEASON_YELLOWS IS NOT NULL AND
SEASON_REDS IS NOT NULL
ORDER BY PLAYER_EVENT_RANK, SEARCH_RANK;
```

Challenges

1. Few Fifa World Cup related queries.
2. In AOL-data set no queries found during the World Cup 2006 time frame.
3. Took longer time to perform complex query.

Questions?

Thank You

APPENDIX

Queries

Query	Special operators
Are the cards that a player receives related to his goals?	SLICE, DICE, CORR
Which stadium/team got the most goals?	SLICE
What are the most searched queries (only world cup related, excluding world cup)?	SLICE, DICE
How have the searches for the World Cup changed over time?	SLICE, DICE, ROLLUP
What are the most clicked search results for the World Cup and how frequently are they clicked?	SLICE, PARTITION BY, DENSE_RANK, GROUPING SETS
Who are the most searched players, and how many goals did they score? Is a player's search popularity related to their goals?	SLICE, PARTITION BY, DENSE_RANK, CUBE, CORR

https://github.com/AhmedDiderRahat/wise_2324_bi_project

Sources

1. FIFA World Cup 2006 Germany Logo: https://upload.wikimedia.org/wikipedia/de/thumb/c/cc/Logo_FIFA_World_Cup_2006_Germany.svg/1200px-Logo_FIFA_World_Cup_2006_Germany.svg.png [November 18, 2023]