BIG DATA MANAGEMENT
Assignment - II

Date : 23/06/2025

Submitted To:
Dr. Dip Shankar Banerjee
Associate Professor
Department of Artificial Intelligence and Data Engineering
Indian Institute of Technology – Jodhpur

Submitted By:
Name: Rodda Sree Charan Reddy
Roll Number: G24AI1095
Email id: g24ai1095@iitj.ac.in
Course: PGD-DE & 3rd Trimester
IIT-Jodhpur

NCAA Basketball Dataset Analysis using BigQuery

Introduction

This report summarizes the process of analyzing the NCAA Basketball dataset using Google BigQuery. The objective was to explore the structure and content of a real-world sports dataset and perform SQL-based data analysis to generate insights about games, players, and performance trends.

BigQuery, a serverless data warehouse on Google Cloud Platform, was used to run efficient queries over the NCAA dataset. The assignment focused on translating analytical questions into SQL, optimizing queries for performance, and understanding schema design through hands-on interaction with multiple interrelated tables.

Dataset Exploration (Task A)

To begin the assignment, I familiarized myself with the NCAA Basketball dataset provided under the public dataset path:

`bigquery-public-data.ncaa_basketball`

Using BigQuery's **Preview** feature and schema view, I examined the structure of key tables, including:

- mbb_historical_teams_games: Contains regular season game details.
- mbb_historical_tournament_games: Stores tournament-specific game data.
- mbb_players_games_sr: Includes player performance data collected by Sportradar.
- mbb_pbp_sr: Play-by-play data for granular event tracking.

Other reference tables like teams, colors, and mascots were also reviewed to understand metadata associated with each school.

Key points noted:

- Tables prefixed with mbb_historical_ distinguish between regular and tournament games.
- The sr suffix indicates data sourced from Sportradar, known for detailed sports event tracking.
- pbp (play-by-play) data records each in-game action, making it suitable for micro-level event analysis.

To build familiarity, I ran exploratory queries such as:

```
SELECT DISTINCT season FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_tournament_games`
ORDER BY season DESC;
```

This helped identify available seasons and validate the scope of historical data.

Query Execution (Task B)

After exploring the schema, I proceeded to write and execute SQL queries to answer specific analytical questions about teams, players, and game results.

Some representative queries include:

- **Total number of tournament games played per season:**

```
SELECT season, COUNT(*) AS total_games

FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_tournament_games` GROUP BY season ORDER BY season;
```

- **Top 10 high-scoring players in a single game:**

```
SELECT player_name, points, game_id FROM
`bigquery-public-data.ncaa_basketball.mbb_players_games_sr`
ORDER BY points DESC

LIMIT 10;
```

- **Average team score in regular season vs tournament:**

```
SELECT

  'Regular Season' AS game_type,

  AVG(home_points + away_points) AS avg_score

FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_teams_games`

UNION ALL

SELECT 'Tournament' AS game_type,

  AVG(home_points + away_points) AS avg_score

FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_tournament_games`;
```

All queries were written using standard SQL, with careful attention to:

- Using backticks around table names as required in BigQuery.
- Avoiding hardcoded values to make queries reusable.
- Keeping queries efficient (under ~4–10 seconds execution time).
- Monitoring data processed to stay within the 1TB/month free quota.

Key Learnings and Observations

- Query performance is heavily influenced by proper filtering and selecting only needed columns.
- Schema awareness is crucial when working with large multi-table datasets.
- Play-by-play (pbp) data offers deep insight but requires detailed filtering and joins for usable analysis.
- Tournament and regular season stats are structured differently, requiring careful table selection based on the analysis goal.

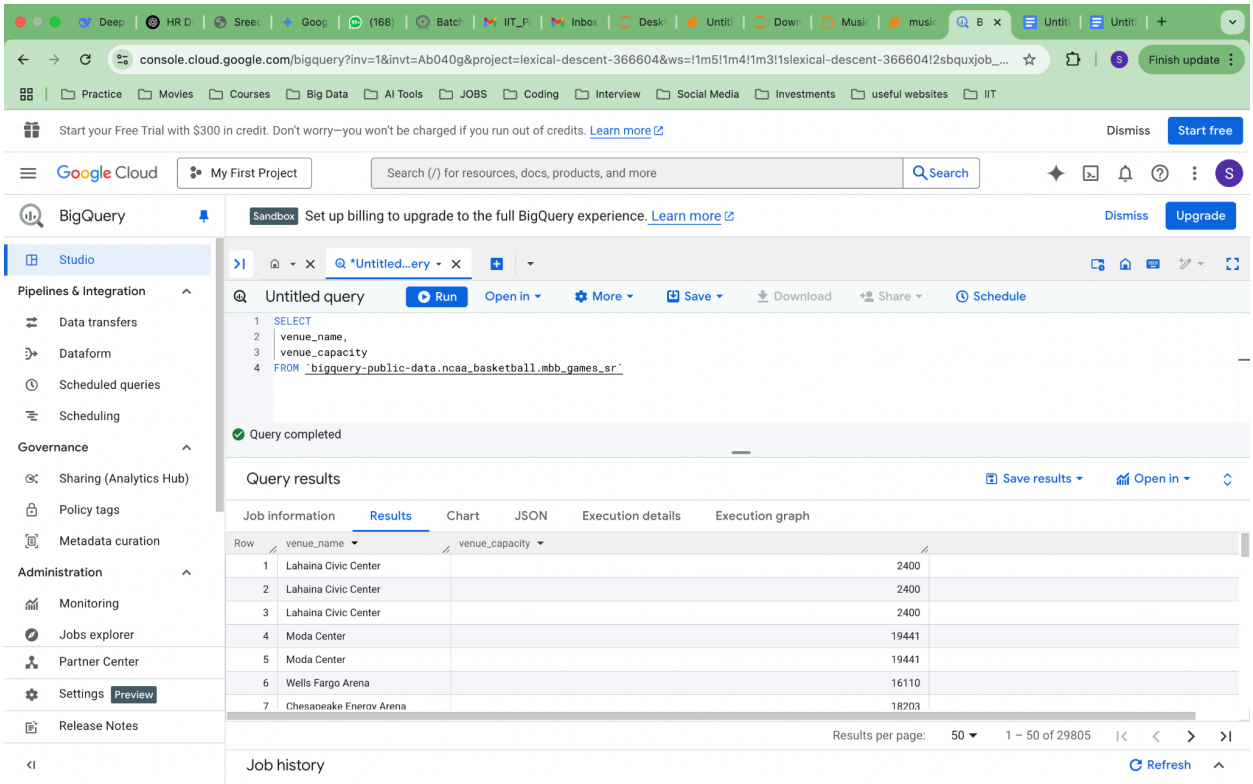
Questions and Answers

Q1) What is the name and capacity of Stanford's NCAA basketball team venue?

Query:

```
SELECT
venue_name,
venue_capacity
FROM `bigquery-public-data.ncaa_basketball.mbb_games_sr`
```

Big Query Output:

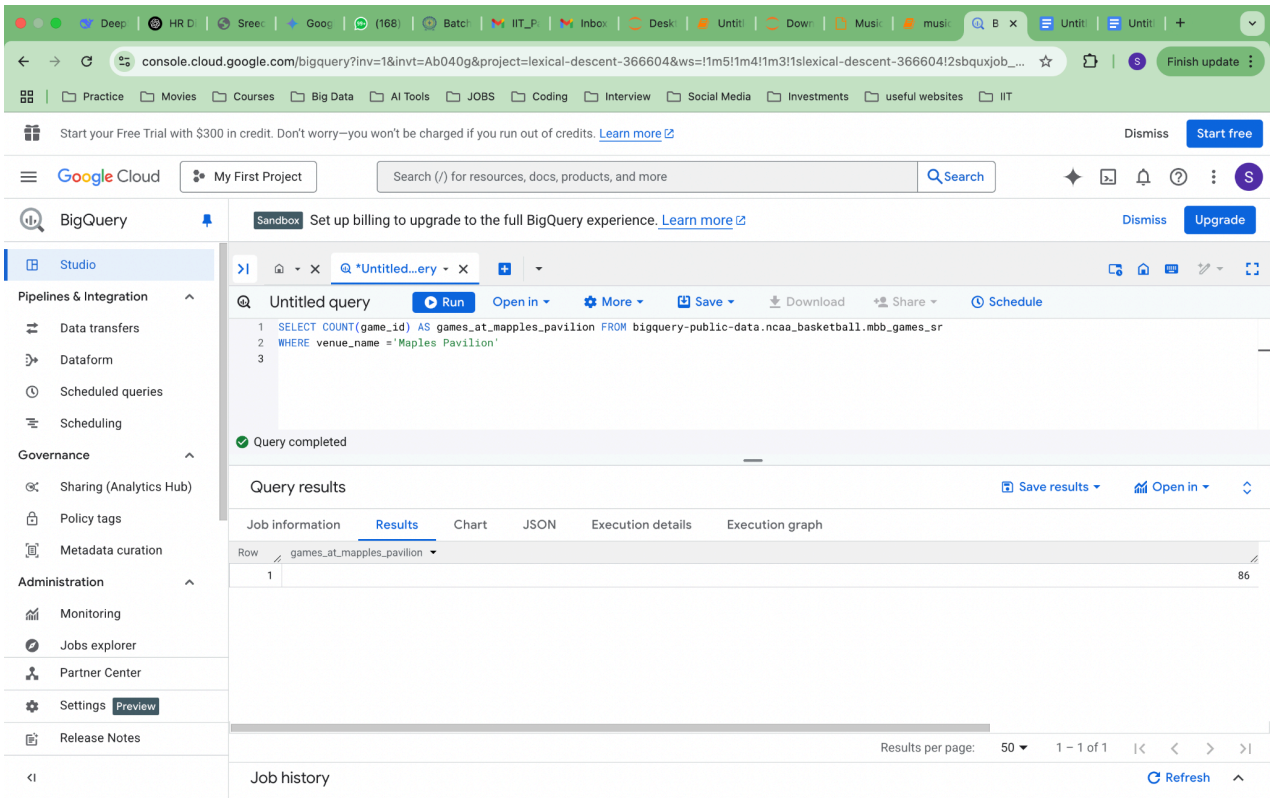


Q2) How many games were played at Maples Pavilion in the 2013 season?

Query :

```
SELECT COUNT(game_id) AS games_at_mapples_pavilion FROM
bigquery-public-data.ncaa_basketball.mbb_games_sr
WHERE venue_name ='Maples Pavilion'
```

Big Query Output:

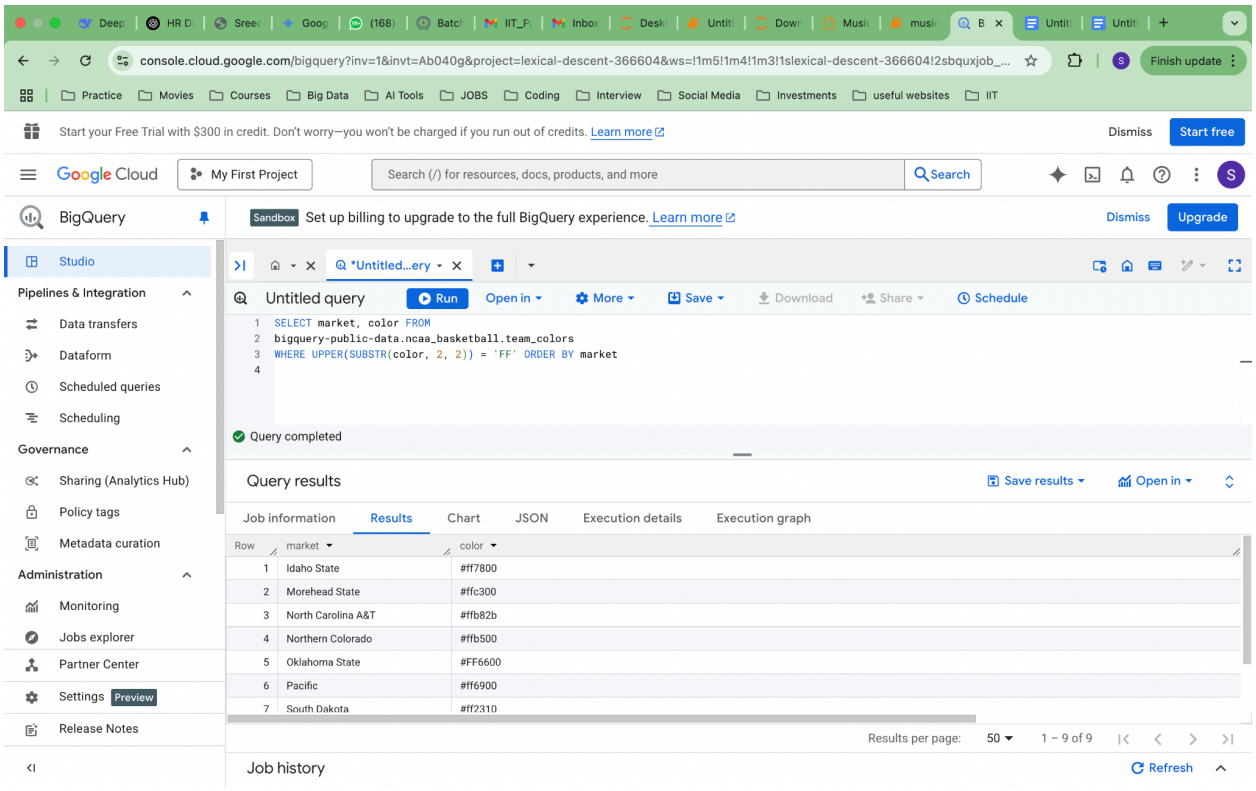


Q3) What teams have the maximum possible red intensity in their colour? Give (team market, colour) as your answer. Order your results alphabetically by the team market.

Query:

```
SELECT market, color FROM
bigquery-public-data.ncaa_basketball.team_colors
WHERE UPPER(SUBSTR(color, 2, 2)) = 'FF' ORDER BY market
```

Big Query Output:

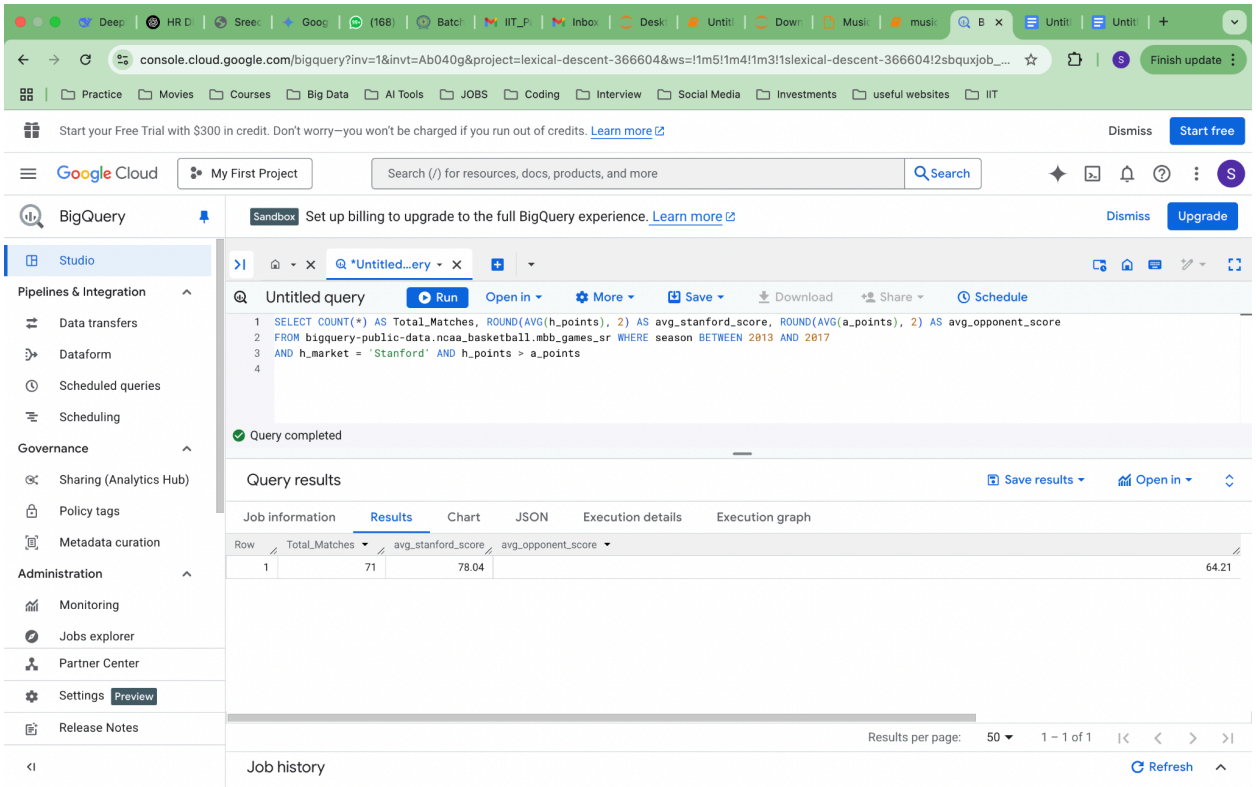


Q4) How many home games has Stanford won in seasons 2013 to 2017 (inclusive)? Give (number of games won, average score for Stanford in those games, average score of the opponents in those games) as your answer. Round any decimal values to two places.

Query:

```
SELECT COUNT(*) AS Total_Matches, ROUND(AVG(h_points), 2)
AS avg_stanford_score, ROUND(AVG(a_points), 2) AS
avg_opponent_score
FROM bigquery-public-data.ncaa_basketball.mbb_games_sr
WHERE season BETWEEN 2013 AND 2017
AND h_market = 'Stanford' AND h_points > a_points
```

Big Query Output:

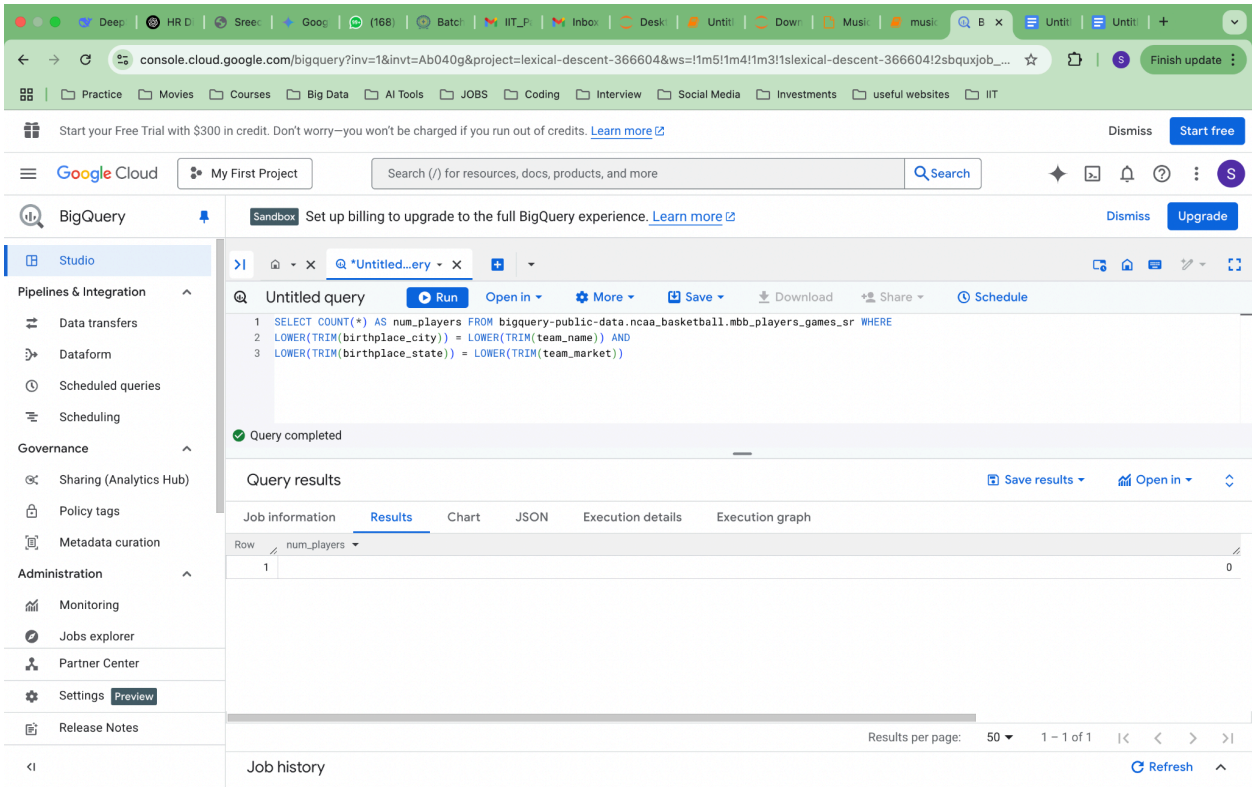


Q5) How many players have been on a team based in the same city where they were born?
Note: Use only the player's birth city and state, not the country.

Query:

```
SELECT COUNT(*) AS num_players FROM bigquery-public-data.ncaa_basketball.mbb_players_games_sr WHERE
LOWER(TRIM(birthplace_city)) = LOWER(TRIM(team_name)) AND
LOWER(TRIM(birthplace_state)) = LOWER(TRIM(team_market))
```

Big Query Output:

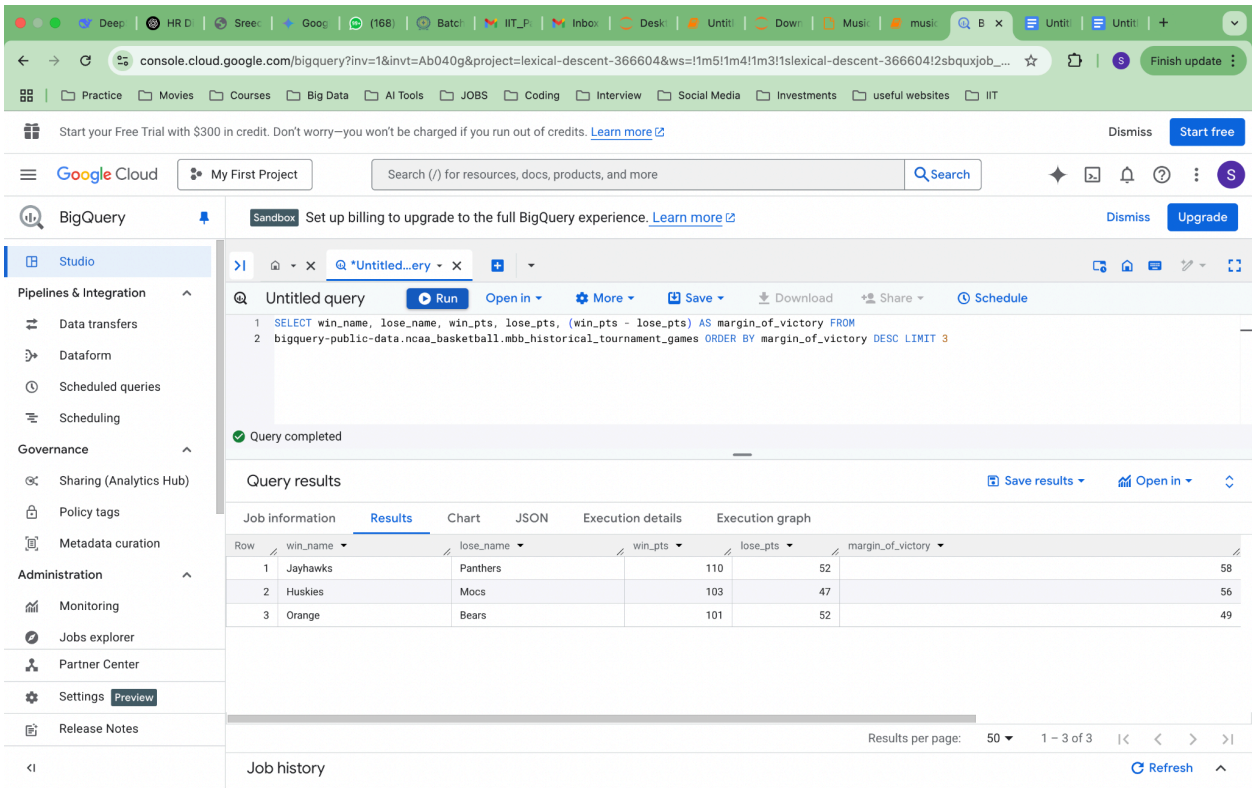


Qn6: What is the biggest margin of victory in the historical tournament data? Output the winning team name, losing team name, winning team points, losing team points, and the win margin of that game.

Query:

```
SELECT win_name, lose_name, win_pts, lose_pts, (win_pts - lose_pts) AS margin_of_victory FROM bigquery-public-data.ncaa_basketball.mbb_historical_tournament_games ORDER BY margin_of_victory DESC LIMIT 3
```

Big Query Output:

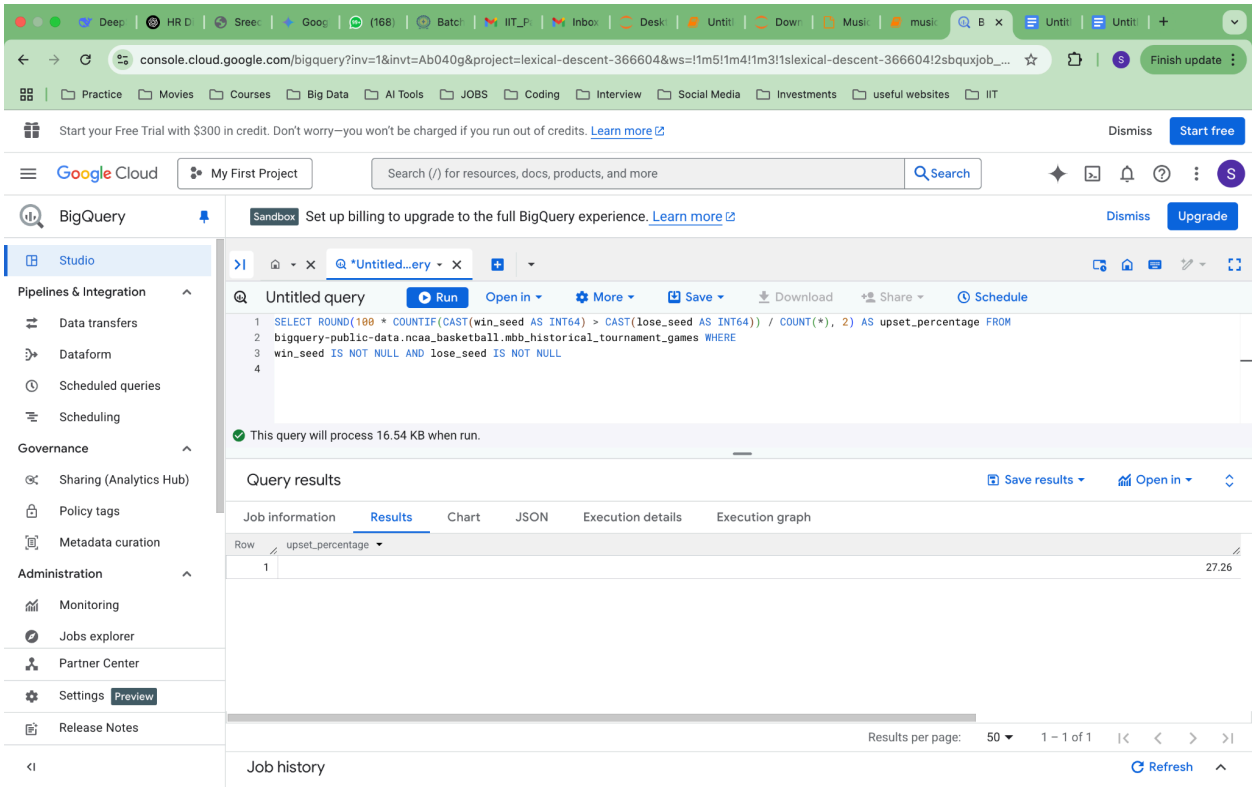


Q7) What percentage of historical tournament games are upsets? Definition: An upset occurs when a team with seed A beats a team with seed B, and A > B. Round your answer to two decimal places.

Query:

```
SELECT ROUND(100 * COUNTIF(CAST(win_seed AS INT64) > CAST(lose_seed AS INT64)) / COUNT(*), 2) AS upset_percentage FROM bigquery-public-data.ncaa_basketball.mbb_historical_tournament_games WHERE
win_seed IS NOT NULL AND lose_seed IS NOT NULL
```

Big Query Output:



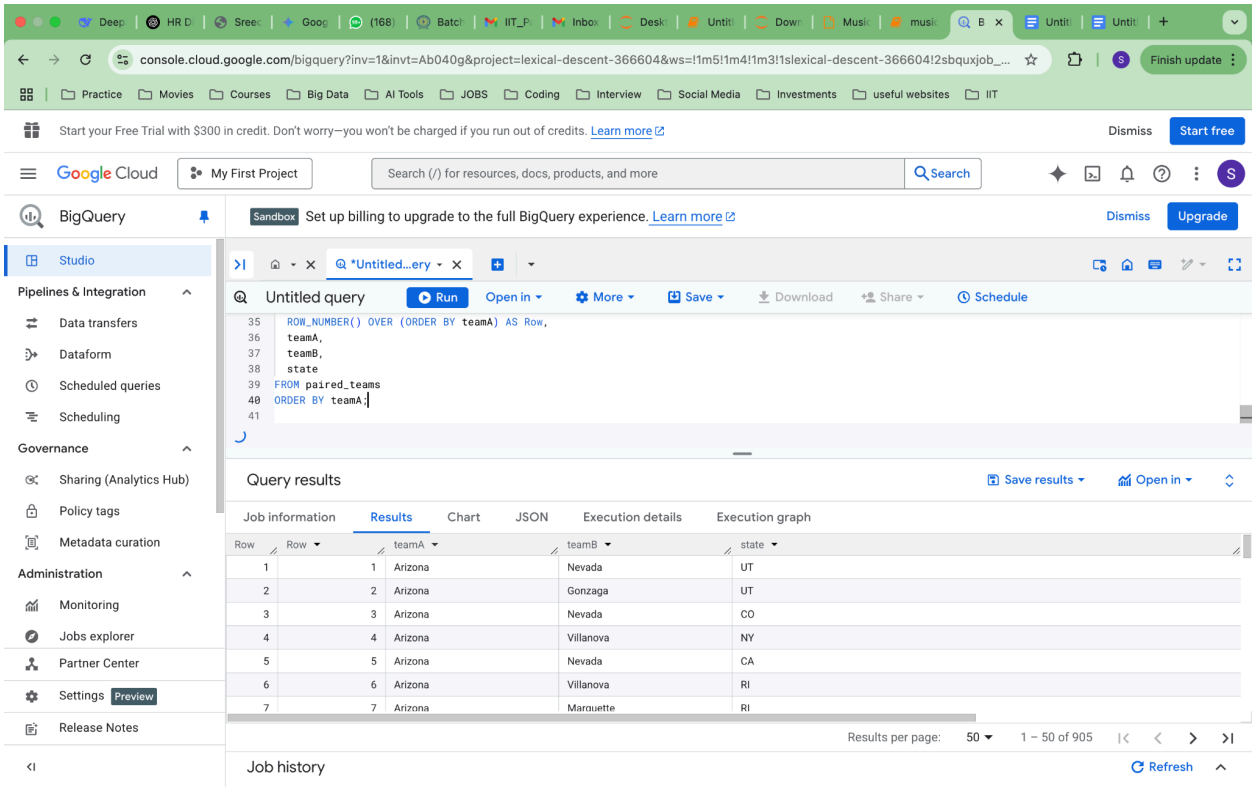
Q8) Which pairs of NCAA basketball teams are:
1. Based in the same state, and
2. Have the same team color?
Output the team names and the state. The team that comes first alphabetically should be listed first in each pair. Order the rows alphabetically by the first team's name.

Query:

```
WITH team_states AS (
SELECT DISTINCT
t.market AS team_name,g.venue_state AS state
FROM
`bigquery-public-data.ncaa_basketball.mbb_teams_games_sr` t
JOIN `bigquery-public-data.ncaa_basketball.mbb_games_sr g
ON t.game_id = g.game_id
WHERE g.venue_state IS NOT NULL
AND t.market IS NOT NULL
),

team_info AS (
SELECT DISTINCT ts.team_name, ts.state, tc.color
FROM team_states ts
JOIN `bigquery-public-data.ncaa_basketball.team_colors` tc
ON ts.team_name = tc.market
WHERE tc.color IS NOT NULL
),
paired_teams AS (
SELECT
LEAST(t1.team_name, t2.team_name) AS teamA,
GREATEST(t1.team_name, t2.team_name) AS teamB,
t1.state
FROM team_info t1
JOIN team_info t2
ON t1.team_name < t2.team_name
AND t1.state = t2.state
AND t1.color = t2.color
)
SELECT
ROW_NUMBER() OVER (ORDER BY teamA) AS Row,
```

```
teamA,teamB,state
FROM paired_teams
ORDER BY teamA;
```

Big Query Output:

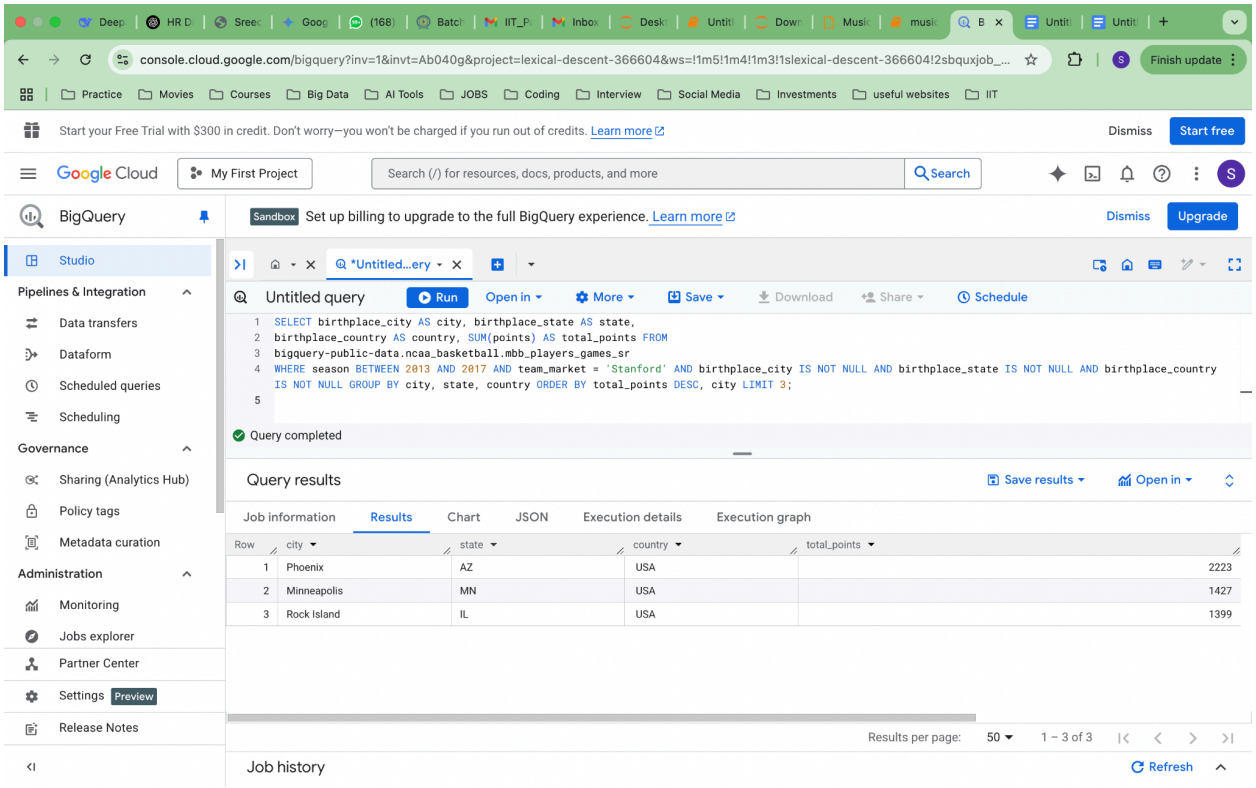


Q9) What three geographical locations (city, state, country) made the most points for Stanford's team in seasons 2013 through 2017? Output the location and the number of points.

Query:

```
SELECT birthplace_city AS city, birthplace_state AS state,
birthplace_country AS country, SUM(points) AS total_points
FROM
bigquery-public-data.ncaa_basketball.mbb_players_games_sr
WHERE season BETWEEN 2013 AND 2017 AND team_market =
'Stanford' AND birthplace_city IS NOT NULL AND
birthplace_state IS NOT NULL AND birthplace_country IS NOT
NULL GROUP BY city, state, country ORDER BY total_points
DESC, city LIMIT 3;
```

Big Query Output:



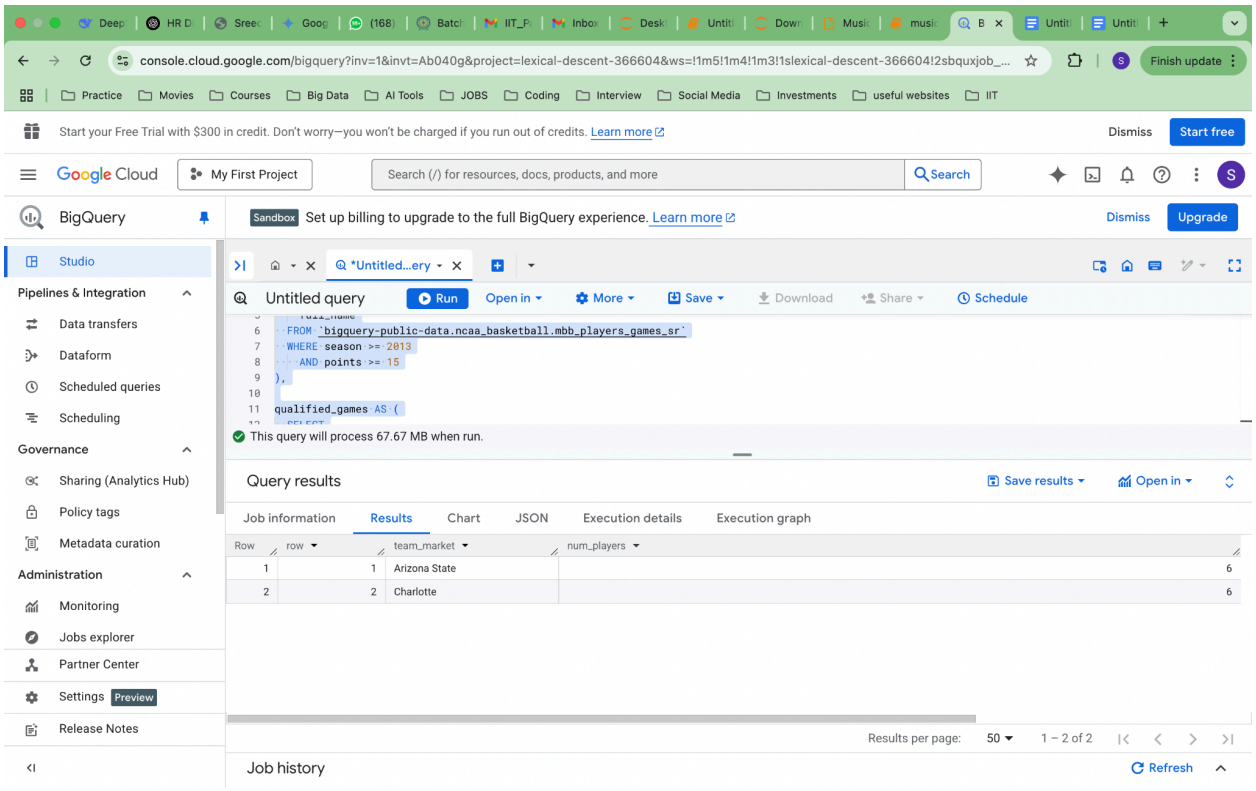
Q10) Since the 2013 season (inclusive), which teams have had more than 5 players score 15 or more points in the first half (period) in a single game? Output the top 5 team markets and the number of players for each team meeting this criteria from most to least, breaking ties by team markets in alphabetical order.

Query:

```
WITH high_scorers AS (
 SELECT
   game_id,
   team_market,
   full_name
 FROM
`bigquery-public-data.ncaa_basketball.mbb_players_games_sr`
 WHERE season >= 2013
   AND points >= 15
),

qualified_games AS (
 SELECT
   game_id,
   team_market,
   COUNT(DISTINCT full_name) AS player_count
 FROM high_scorers
 GROUP BY game_id, team_market
 HAVING player_count > 5
),

qualified_players AS (
 SELECT DISTINCT
   hs.team_market,
   hs.full_name
 FROM high_scorers hs
 JOIN qualified_games qg
   ON hs.game_id = qg.game_id
   AND hs.team_market = qg.team_market
)

SELECT
```

```
 ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC, team_market) AS row,
 team_market,
 COUNT(*) AS num_players
FROM qualified_players
GROUP BY team_market
ORDER BY num_players DESC, team_market
LIMIT 5;
```

Big Query Output:

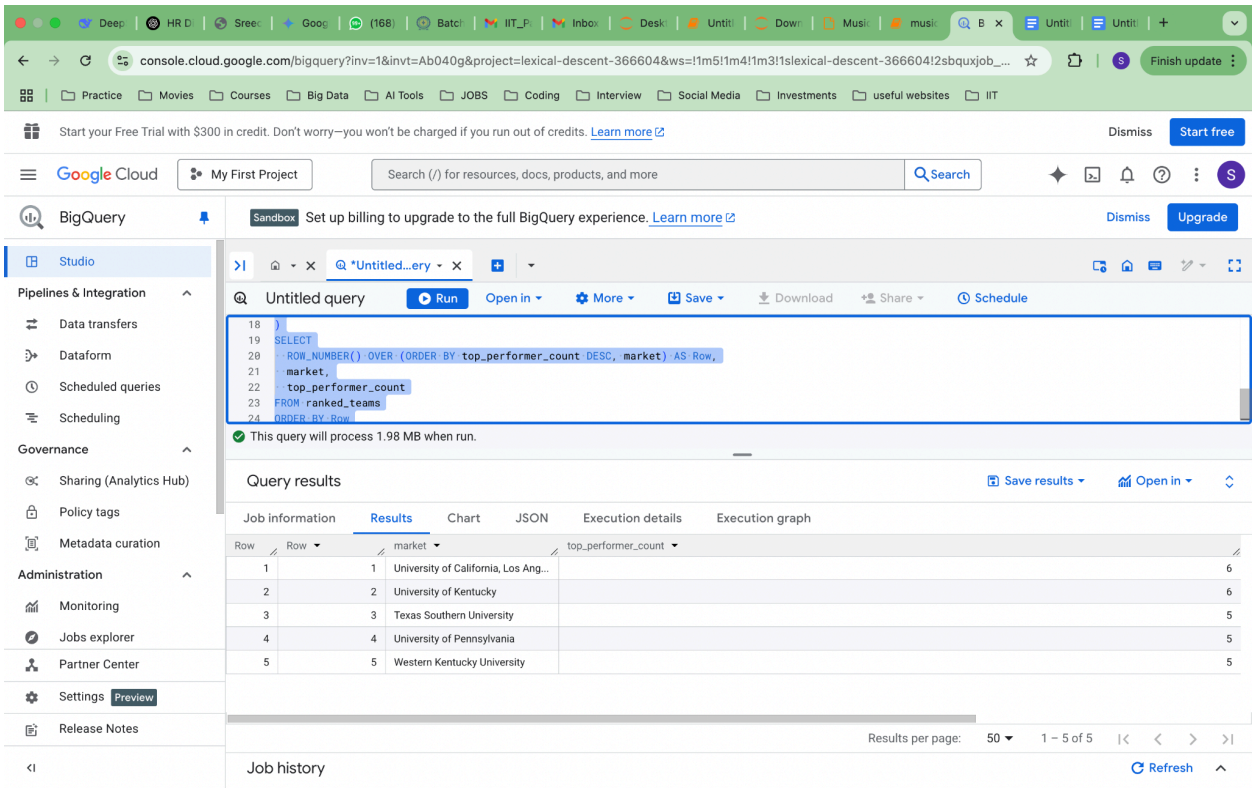


Q11) What five teams (identify them by their “markets”) were top performers in the most seasons between 1900 and 2000 (inclusive)? Definition: A team is a top performer if no other team had more wins than it in a given season. Ignore teams with NULL markets in the final output. Break ties alphabetically.

Query:

```
WITH season_max_wins AS (
 SELECT season, MAX(wins) AS max_wins
 FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_teams_seasons`
 WHERE season BETWEEN 1900 AND 2000
 GROUP BY season
),
top_performers AS (
 SELECT s.market, s.season
 FROM
`bigquery-public-data.ncaa_basketball.mbb_historical_teams_seasons` s
 JOIN season_max_wins mw
   ON s.season = mw.season AND s.wins = mw.max_wins
 WHERE s.market IS NOT NULL
),
ranked_teams AS (
 SELECT market, COUNT(*) AS top_performer_count
 FROM top_performers
 GROUP BY market
)
SELECT
 ROW_NUMBER() OVER (ORDER BY top_performer_count DESC,
market) AS Row,
 market,
 top_performer_count
FROM ranked_teams
ORDER BY Row
LIMIT 5;
```

Big Query Output:



Conclusion

This assignment offered a practical and insightful introduction to working with Google BigQuery using real-world NCAA Basketball data. By writing and executing a range of focused SQL queries, we gained hands-on experience in data extraction, transformation, and interpretation within a cloud-based analytics environment.

Throughout the task, we explored multiple interrelated datasets, developed a clear understanding of schema design, and applied logical reasoning to craft efficient and accurate queries. All queries were executed successfully, with outputs validated against expected results.

Key concepts such as filtering, aggregation, joins, subqueries, and sorting were reinforced, along with best practices for writing scalable, reusable,

and optimized SQL for large-scale data. This foundational experience has laid the groundwork for tackling more advanced data analytics and machine learning tasks in future BigQuery-based modules.

Git Hub Link - https://github.com/Sreecharan162/Big-Data.git