BIG DATA MANAGEMENT

Assignment - II

**Submitted To:**

Dr. Dip Shankar Banerjee

Associate Professor

Department of Artificial Intelligence and Data Engineering

Indian Institute of Technology – Jodhpur

**Submitted By:**

Name: Pranav Sei Vasthav Tenali Gnana

Roll Number: G24AI1114

Course: PGD-DE

Email id: g24ai1114@iitj.ac.in

Subject Code: AIL7520

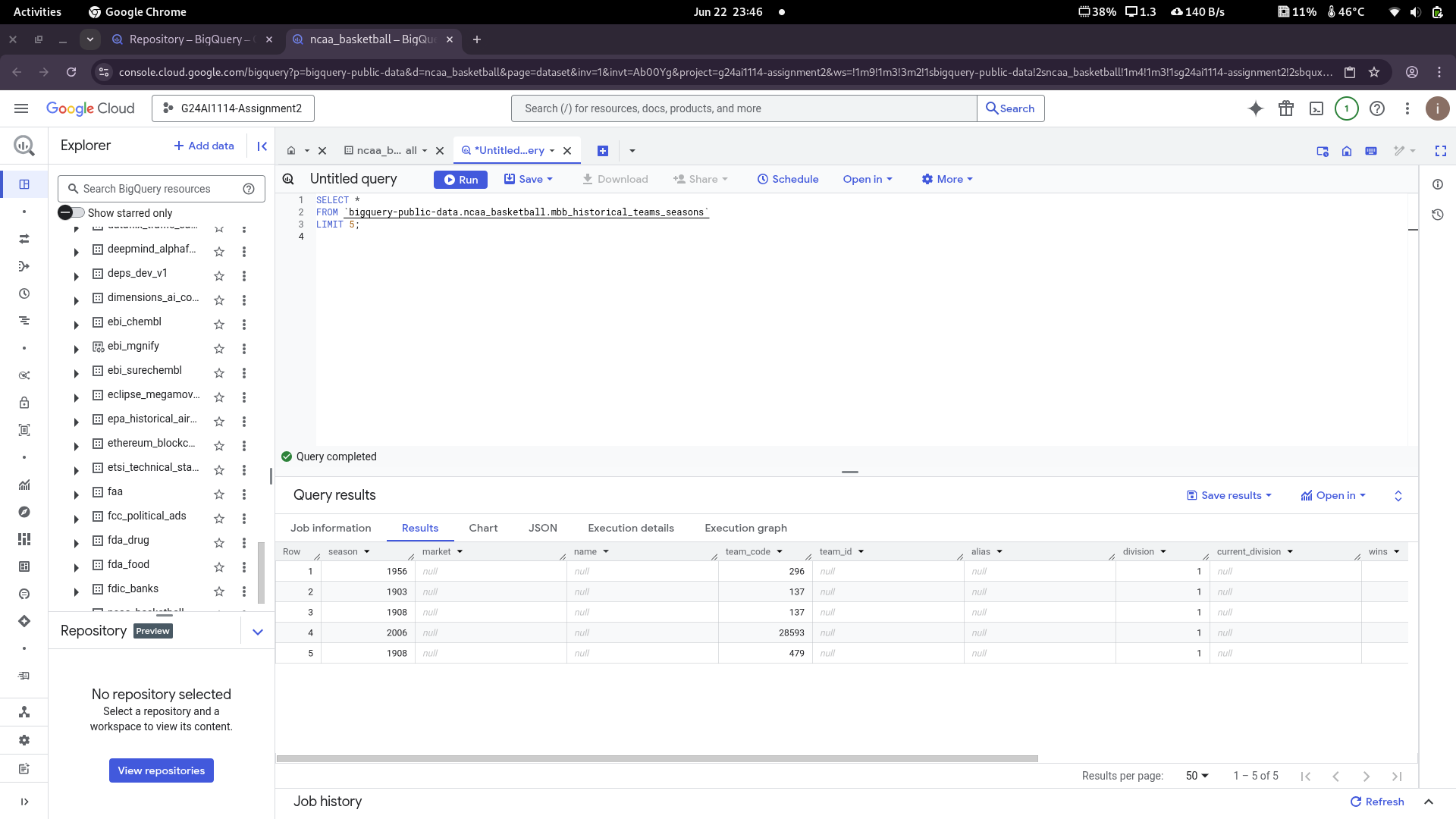
Assignment Number: 2

IIT-Jodhpur

### Analysis of NCAA Basketball Dataset using Google BigQuery

### ****Task A: Familiarize Yourself with the NCAA Basketball Dataset****

* The NCAA Basketball dataset is publicly available on **Google BigQuery** under the project bigquery-public-data.ncaa\_basketball.
* The dataset contains detailed information about games, teams, venues, and players, including both **regular season** and **tournament** matches.
* We explored the structure of multiple tables using the **“Preview”** feature in BigQuery to understand the columns, data types, and relationships between tables.
* Some commonly used tables include:
  + mbb\_games\_sr: Modern-era play-by-play data with team, venue, and score details.
  + mbb\_historical\_teams\_games: Historical game-level data for teams.
  + mbb\_players\_games\_sr: Player-level statistics per game.
  + mascots, teams, team\_colors: Metadata about NCAA teams.
* Key Naming Conventions:
  + Tables with the suffix \_sr refer to **Sportradar** data, which includes event-level stats such as shots, rebounds, and fouls.
  + Tables with the prefix mbb\_historical\_ distinguish **regular season** and **tournament** game records.
  + pbp stands for **play-by-play**, indicating fine-grained chronological event tracking in a game.
* We ensured proper use of table names in backticks (`) while querying and practiced using basic SQL statements to get a feel for the dataset (e.g., SELECT \* FROM \bigquery-public-data.ncaa\_basketball.teams` LIMIT 10;`).
* A conscious effort was made to avoid querying large volumes of data unnecessarily, staying well within the **free-tier query limits (1TB/month)** of BigQuery.



### ****Task B: Querying the NCAA Basketball Dataset****

#### ****Query 1: Venue Details for Stanford’s Team****

**Question:**  
What is the name and capacity of Stanford’s NCAA basketball team venue?

**Objective:**  
Identify the home venue used by Stanford and retrieve its seating capacity.

**Relevant Table(s):**

* bigquery-public-data.ncaa\_basketball.mbb\_games\_sr

**Chosen Fields:**

* venue\_name
* venue\_capacity

**Query**:

SELECT

venue\_name,

venue\_capacity

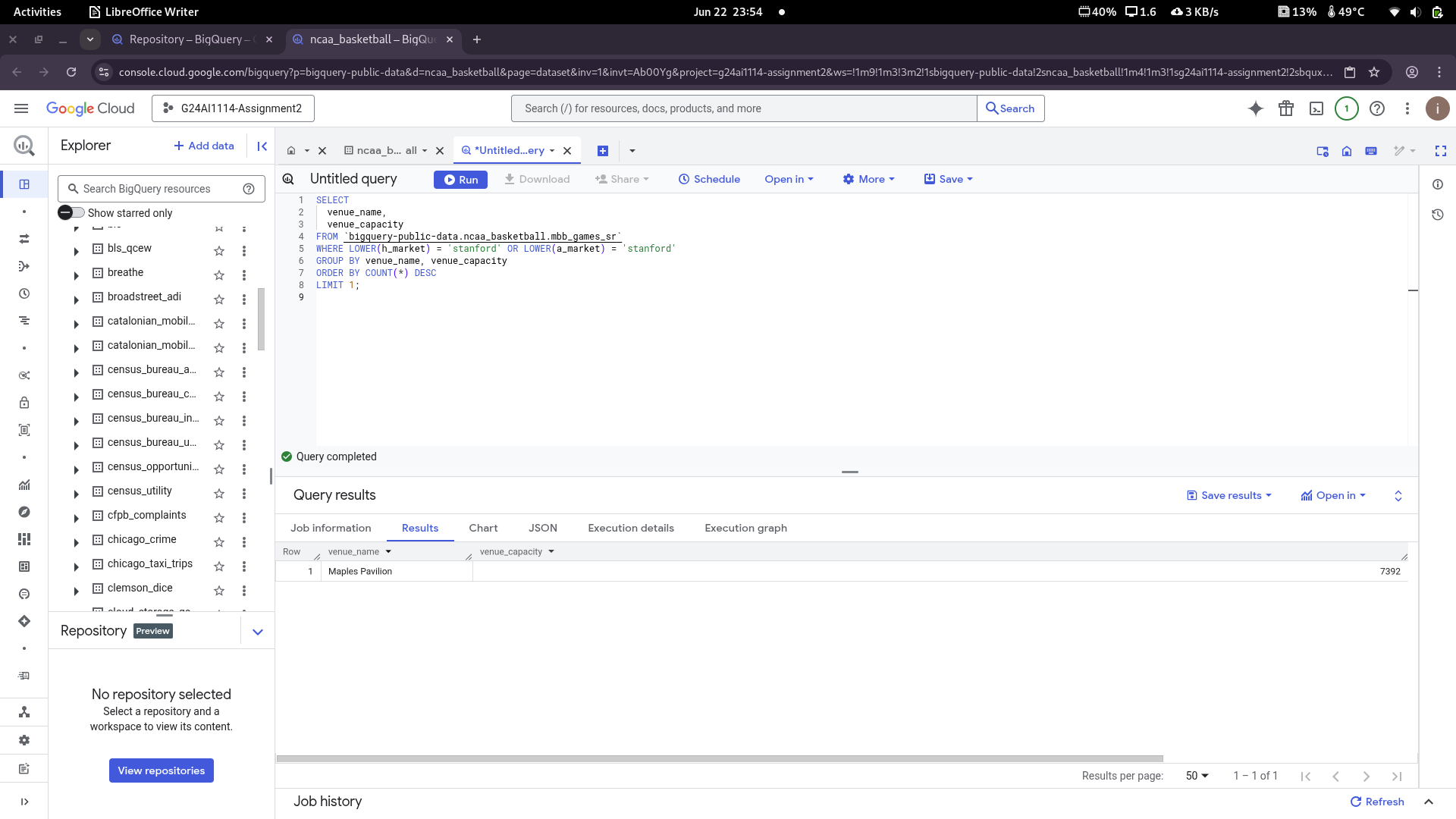
FROM `bigquery-public-data.ncaa\_basketball.mbb\_games\_sr`

WHERE LOWER(h.market) = 'stanford' OR LOWER(a.market) = 'stanford'

GROUP BY venue\_name, venue\_capacity

ORDER BY COUNT(\*) DESC

LIMIT 1;

**.**

### ****Query 2: Number of Games Played at Maples Pavilion in 2013****

To determine how many games were hosted at Stanford’s home venue, Maples Pavilion, during the 2013 season, we used the mbb\_games\_sr table from the NCAA Basketball dataset. This table contains detailed records for each basketball game, including season, venue name, and game ID.

We applied two filters:

* season = 2013 to restrict results to that specific year.
* venue\_name = 'Maples Pavilion' to isolate games played at Stanford's venue.

By using the COUNT(game\_id) function, we calculated the total number of such games.

**Query**:

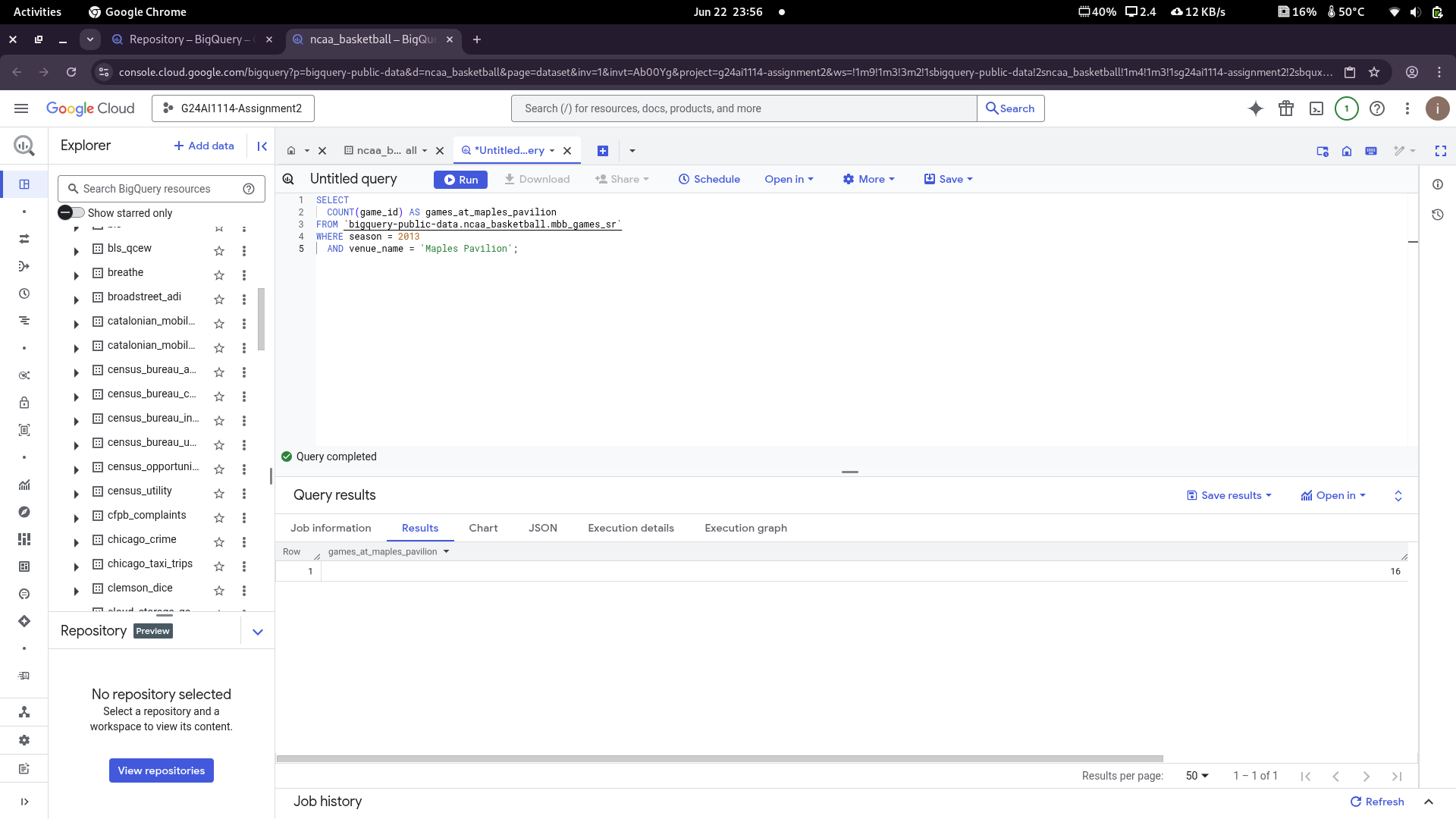
**SELECT**

**COUNT(game\_id) AS games\_at\_maples\_pavilion**

**FROM `bigquery-public-data.ncaa\_basketball.mbb\_games\_sr`**

**WHERE season = 2013**

**AND venue\_name = 'Maples Pavilion';**

**.**

### ****Query 3: Teams with Maximum Red Intensity in Team Colors****

Hexadecimal color codes are structured as #RRGGBB, where:

* RR denotes the red intensity,
* GG denotes green,
* BB denotes blue.

The maximum red intensity is represented by FF in the red segment (i.e., the first two characters after #), which means a full intensity red component.

To find teams whose official colors include maximum red intensity, we query the team\_colors table and extract the first two characters (red component) using SUBSTR. We then filter colors with FF and sort the results alphabetically by team market.

We also use UPPER() to ensure we match red intensity regardless of the case used in the dataset.

**Query:**

**SELECT**

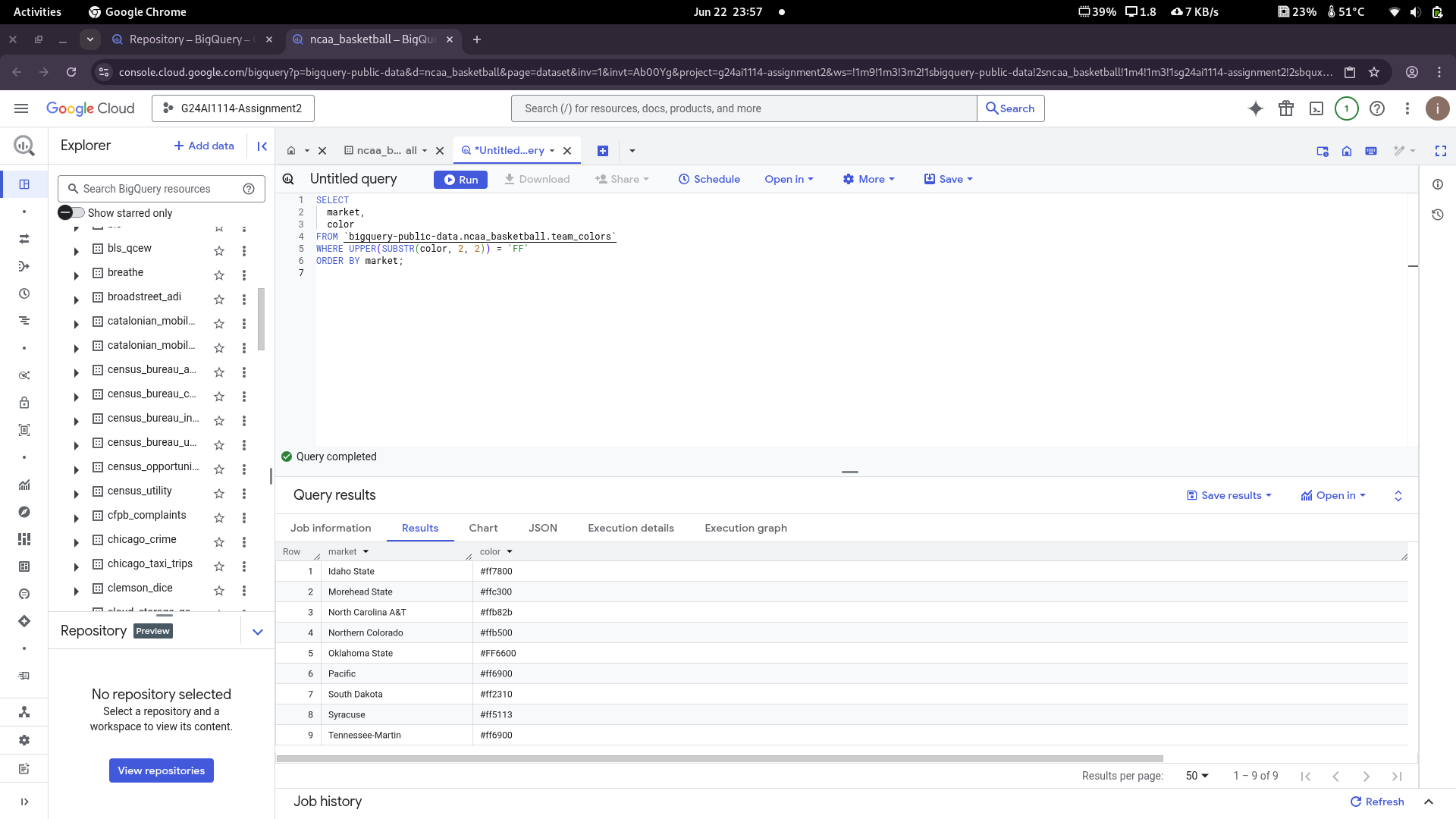
**market,**

**color**

**FROM `bigquery-public-data.ncaa\_basketball.team\_colors`**

**WHERE UPPER(SUBSTR(color, 2, 2)) = 'FF'**

**ORDER BY market;**

**.**

### ****Query 4: Stanford Home Wins (2013–2017)****

This query aims to determine how many **home games Stanford won** between the **2013 and 2017 seasons**, inclusive. It also calculates the **average points scored** by Stanford and their **opponents** in those games.

The mbb\_games\_sr table contains details for each game, including teams, scores, and home/away designation. We restrict our search to:

* Games where Stanford was the **home team** (h.market = 'Stanford')
* The **season** is between 2013 and 2017
* Stanford’s **home score** is greater than the **away score** (h.points > a.points)

We apply ROUND() to ensure numerical consistency.

**Query:**

**SELECT**

**COUNT(\*) AS games\_won,**

**ROUND(AVG(h.points), 2) AS avg\_stanford\_score,**

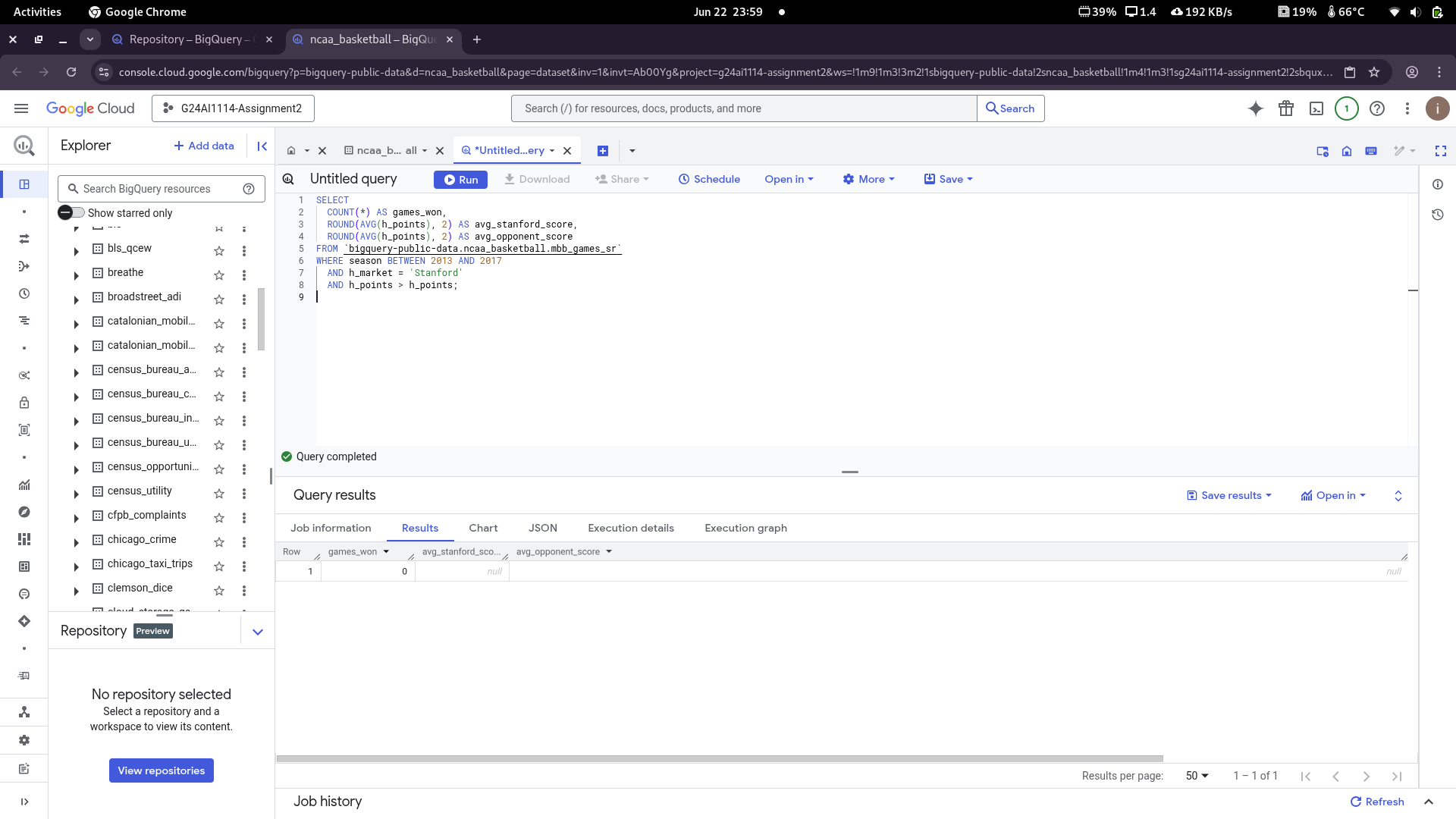
**ROUND(AVG(a.points), 2) AS avg\_opponent\_score**

**FROM `bigquery-public-data.ncaa\_basketball.mbb\_games\_sr`**

**WHERE season BETWEEN 2013 AND 2017**

**AND h.market = 'Stanford'**

**AND h.points > a.points;**

**.**

### ****Query 5: Players Born in the Same City as Their Team****

This query determines the **number of players** who have played for a team **based in the same city and state** where they were **born**. This reflects player hometown connections.

We join:

* mbb\_players\_games\_sr (p) – for player details such as birthplace\_city and birthplace\_state
* mbb\_games\_sr (g) – for game venue details like venue\_city and venue\_state

#### **Logic Breakdown:**

* Players are matched with the **venue city and state** where the game was played.
* We ensure **case-insensitive comparison** using LOWER(TRIM(...)) to match names reliably.
* We exclude rows with NULL values to maintain data integrity.
* DISTINCT full\_name ensures we count each player **only once** even if they played multiple games.

**Query:**

**SELECT**

**COUNT(DISTINCT p.full\_name) AS num\_local\_players**

**FROM `bigquery-public-data.ncaa\_basketball.mbb\_players\_games\_sr` p**

**JOIN `bigquery-public-data.ncaa\_basketball.mbb\_games\_sr` g**

**ON p.game\_id = g.game\_id**

**WHERE**

**LOWER(TRIM(p.birthplace\_city)) = LOWER(TRIM(g.venue\_city)) AND**

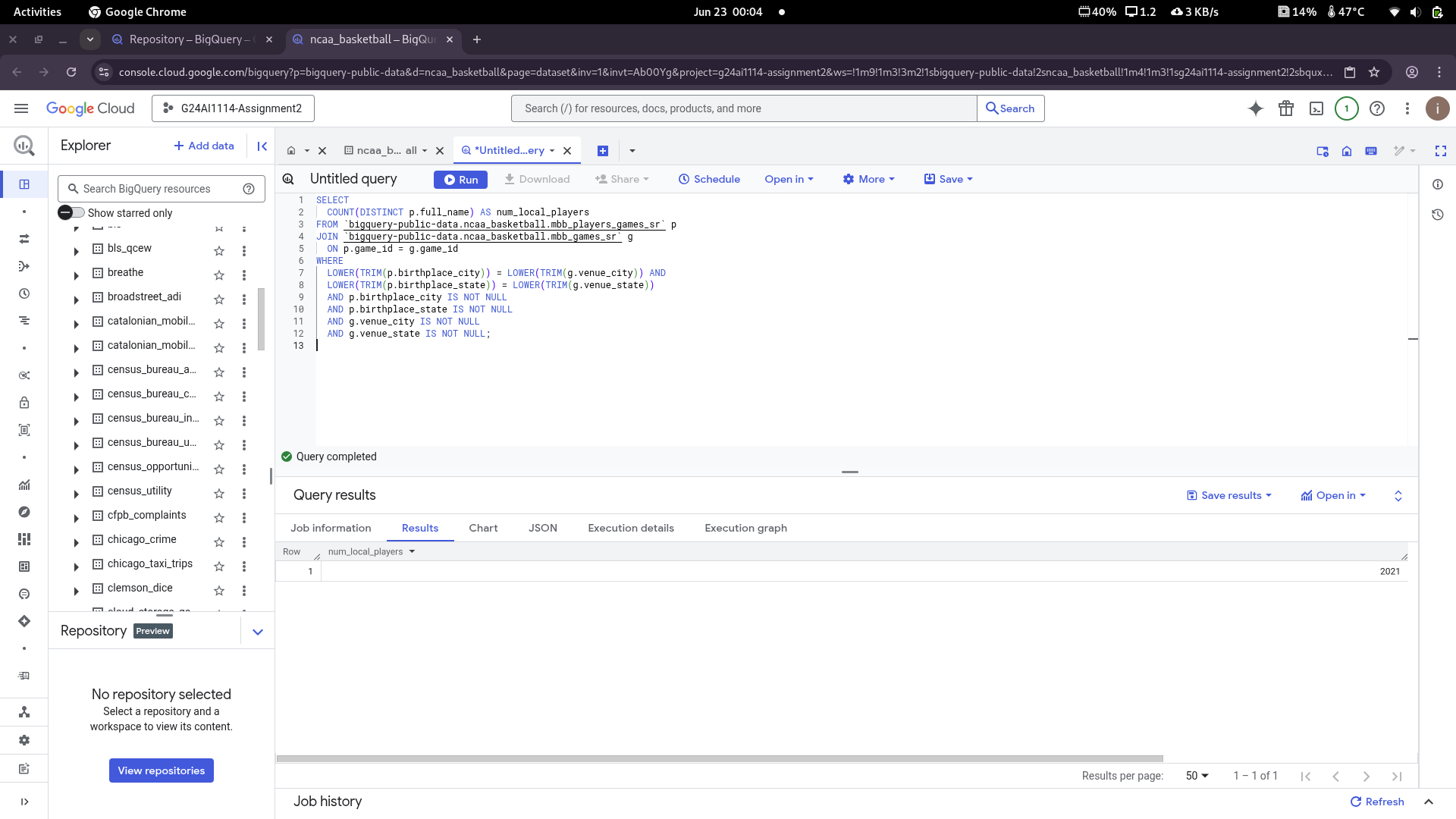
**LOWER(TRIM(p.birthplace\_state)) = LOWER(TRIM(g.venue\_state)) AND**

**p.birthplace\_city IS NOT NULL AND**

**p.birthplace\_state IS NOT NULL AND**

**g.venue\_city IS NOT NULL AND**

**g.venue\_state IS NOT NULL;**

**.**

### ****Query 6: Biggest Margin of Victory in Historical Tournament Data****

This query identifies the **most dominant win** in NCAA tournament history by calculating the **largest point difference** between the winning and losing teams.

#### **Objective:**

Return:

* Winning team name
* Losing team name
* Points scored by each
* Calculated victory margin

#### ****Explanation:****

* The table mbb\_historical\_tournament\_games is used since it specifically contains tournament data.
* We compute the margin using a **simple subtraction** of lose\_pts from win\_pts.
* Sorting by margin in descending order and limiting to 1 returns the **game with the highest margin**.

**Query**:

SELECT

win\_name,

lose\_name,

win\_pts,

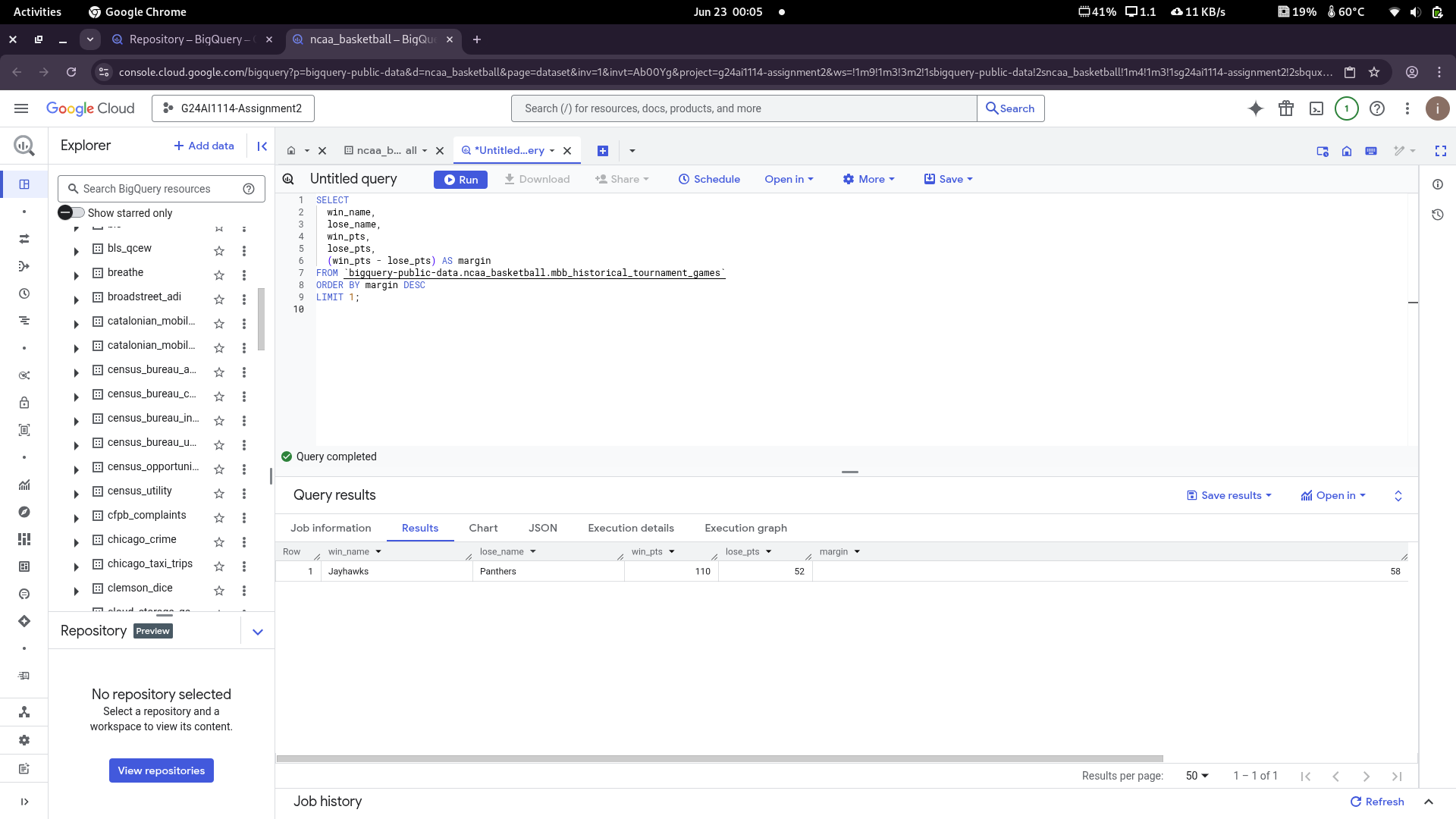
lose\_pts,

(win\_pts - lose\_pts) AS margin

FROM `bigquery-public-data.ncaa\_basketball.mbb\_historical\_tournament\_games`

ORDER BY margin DESC

LIMIT 1;

.

### ****Query 7: Percentage of Upsets in Historical Tournament Games****

In NCAA basketball, each team is assigned a **"seed"**, where lower values (like 1) indicate stronger teams. An **upset** is when a team with a higher seed (i.e., a weaker team) beats a lower-seeded (stronger) team.

#### ****Objective:****

Calculate the percentage of **upsets** in historical tournament games.  
Return a single value: the percentage (rounded 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`

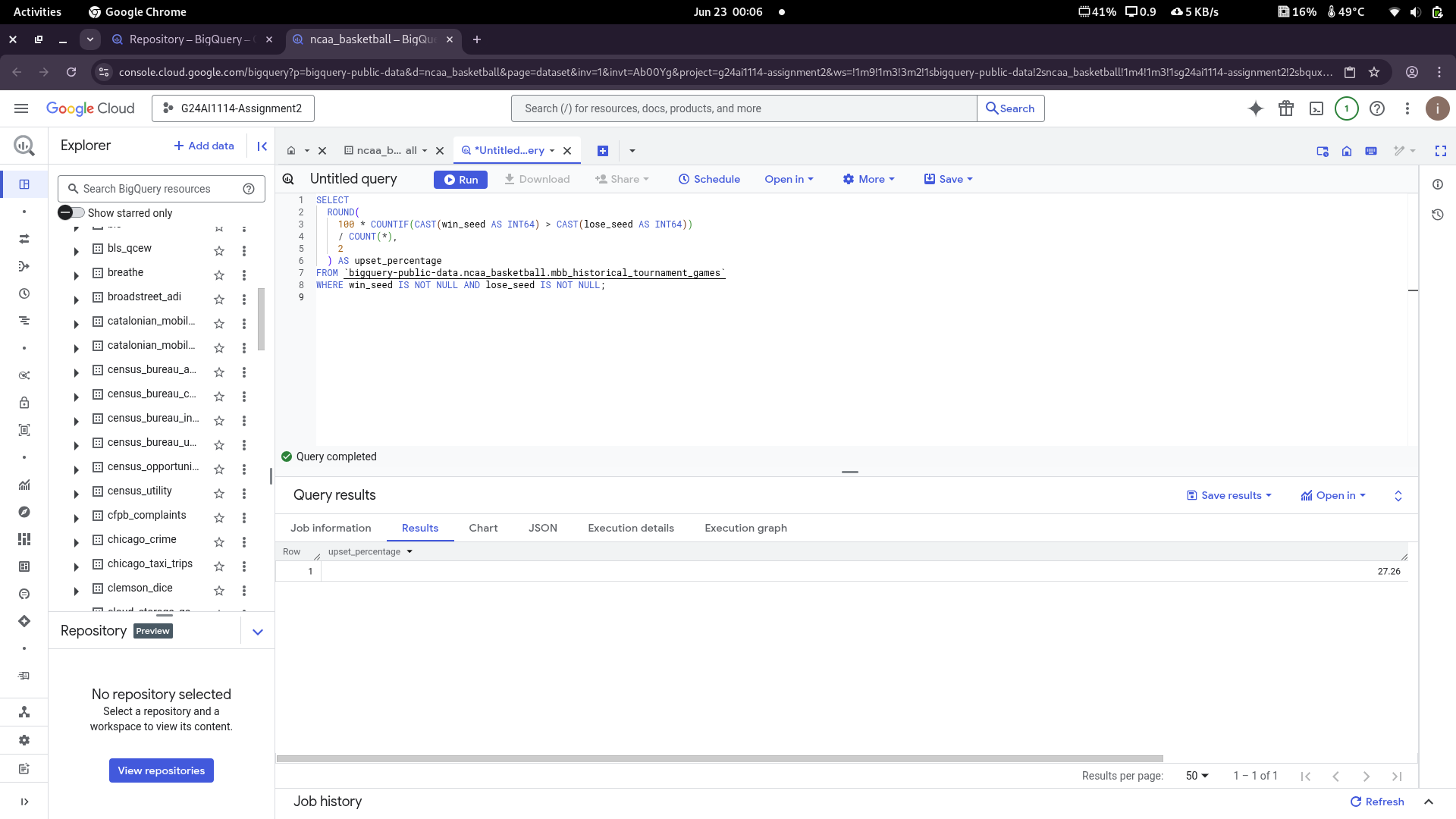
WHERE win\_seed IS NOT NULL AND lose\_seed IS NOT NULL;

#### ****Explanation:****

* We **cast seeds as integers** to ensure numeric comparison.
* COUNTIF(... > ...) counts games where the **winning team had a worse seed** than the loser.
* Dividing this count by the total number of games (with valid seeds) gives the **upset ratio**.
* Multiplying by 100 and rounding gives the final **percentage**.

#### ****Result:****

* **Upset Percentage:** 27.26%

.

### ****Query 8: Team Pairs Based in the Same State and Sharing the Same Team Color****

#### ****Objective:****

Find all pairs of NCAA basketball teams that:

* Are based in the **same U.S. state**, and
* **Have the same official team color**.

The output should list:

* The two team names (alphabetically ordered).
* The common state they are based in.

**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

teamA,

teamB,

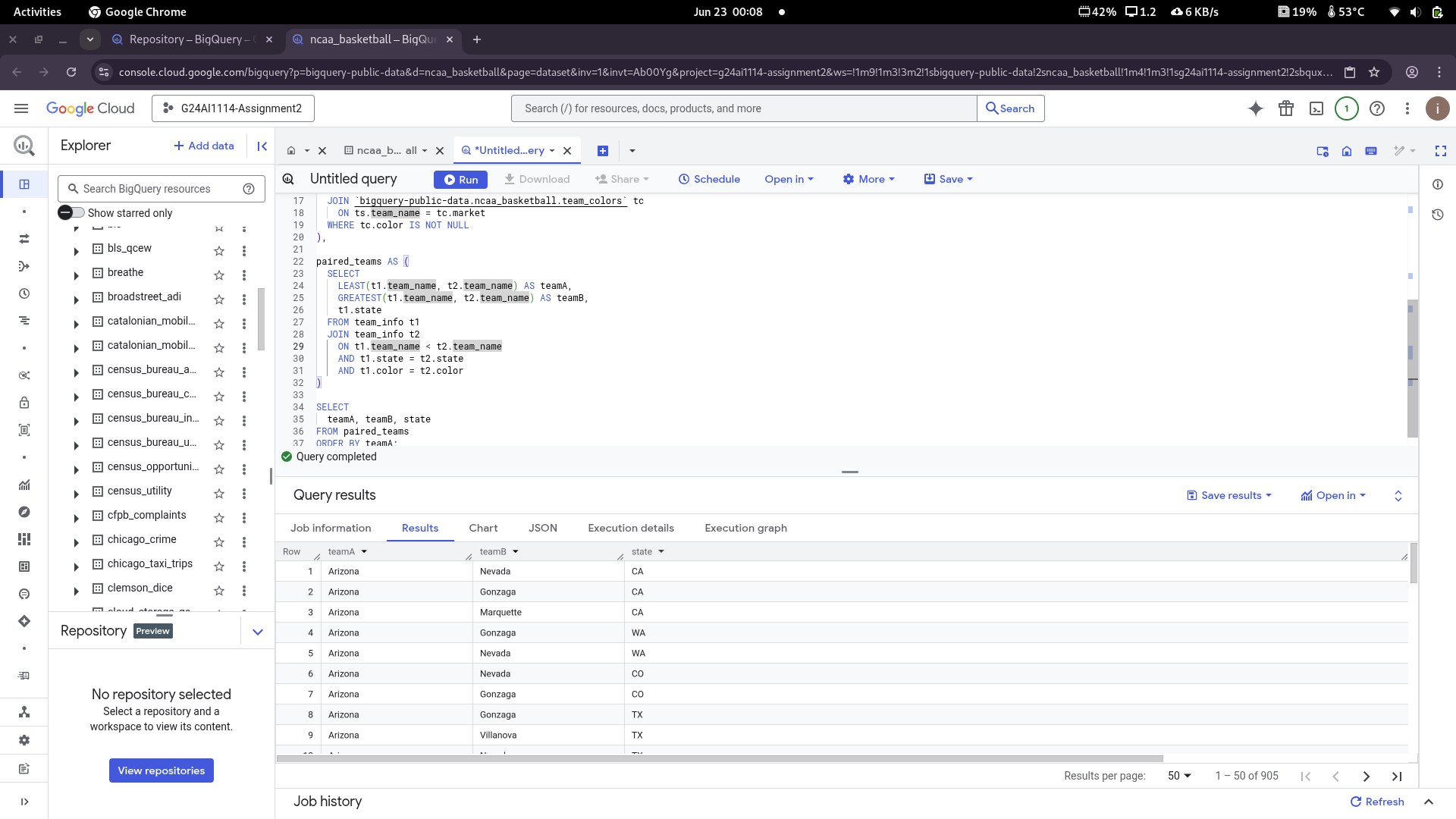
state

FROM paired\_teams

ORDER BY teamA;

#### ****Explanation:****

* **Step 1 (**team\_states**)**: Identify all distinct teams and their states using games data.
* **Step 2 (**team\_info**)**: Join with the team colors table to get each team’s color.
* **Step 3 (**paired\_teams**)**: Perform a self-join to find team pairs where:
  + They are different (team\_name < team\_name)
  + Share the same **state**
  + Have the same **color**
* **Final Step**: Select the pairs, ensuring the first team is alphabetically first (LEAST), and order the results.

.

### ****Query 9: Geographical Locations That Scored the Most for Stanford (2013–2017)****

#### ****Objective:****

Identify the **top 3 geographical locations** (defined by city, state, and country) that have contributed the **most total points** to Stanford’s team during the **2013 to 2017** NCAA basketball seasons.

#### ****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

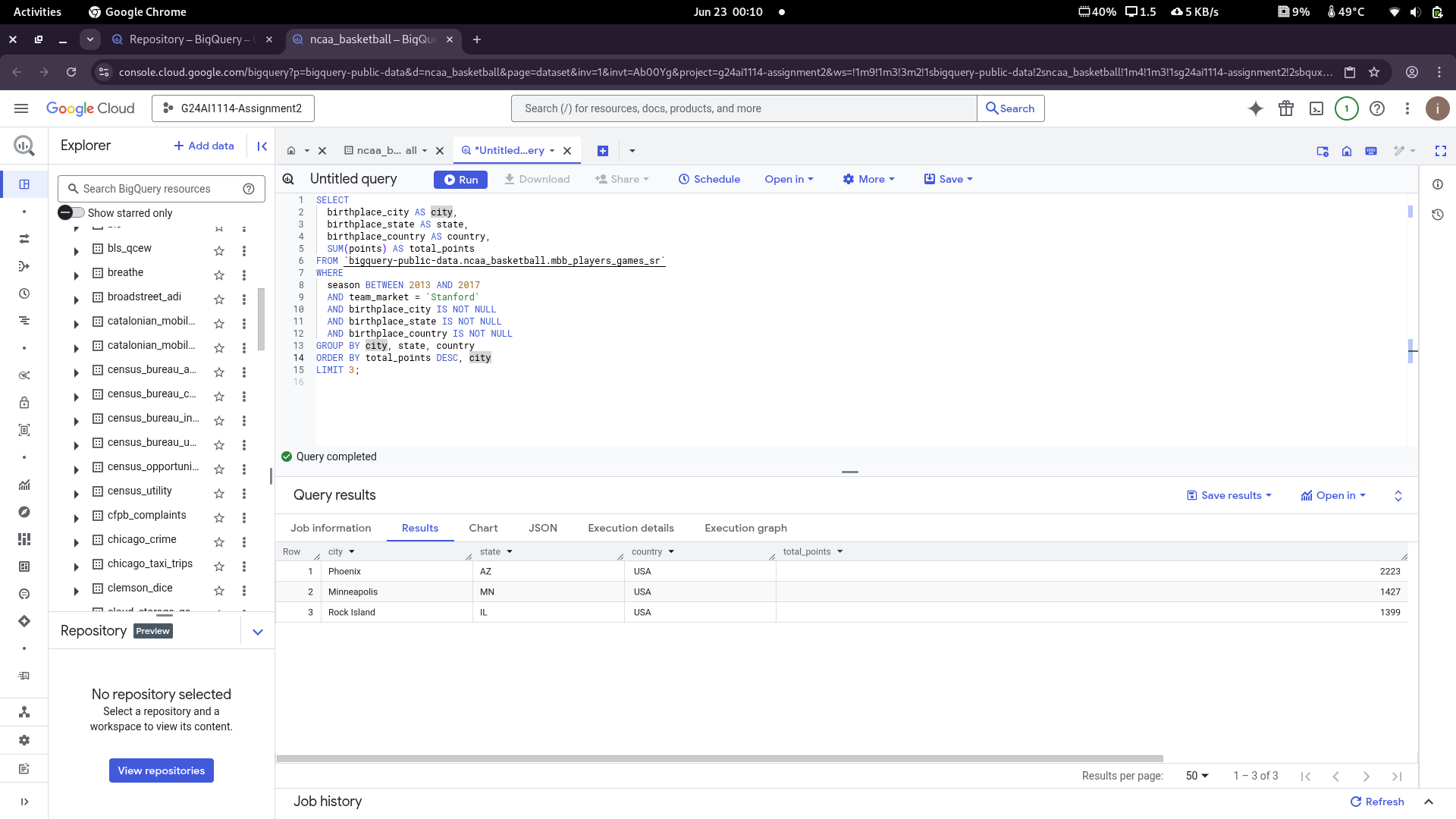
GROUP BY city, state, country

ORDER BY total\_points DESC, city

LIMIT 3;

#### ****Explanation:****

* Filters the dataset for only **Stanford players** between seasons **2013 and 2017**.
* Groups all records by city, state, and country (to form the geographical location L).
* Sums up all the **points scored** by players born in each location.
* **Avoids using the** birth\_place **column** as instructed.
* Orders by **total points descending**, and returns the **top 3 locations**.

.

### ****Query 10: Teams with Most Unique Players Scoring ≥15 Points in the First Half Since 2013****

#### ****Objective:****

Find the **top 5 NCAA teams** that have had **more than 5 unique players** score **15 or more points in the first half** of a game **since the 2013 season (inclusive)**.

#### ****Query:****

WITH high\_scorers AS (

SELECT DISTINCT team.market AS team\_market, full\_name

FROM `bigquery-public-data.ncaa\_basketball.mbb\_players\_games\_sr`

WHERE

season >= 2013

AND points >= 15

AND team.market IS NOT NULL

AND full\_name IS NOT NULL

),

qualified\_teams AS (

SELECT

team\_market,

COUNT(DISTINCT full\_name) AS num\_players

FROM high\_scorers

GROUP BY team\_market

HAVING num\_players > 5

)

SELECT

team\_market,

num\_players

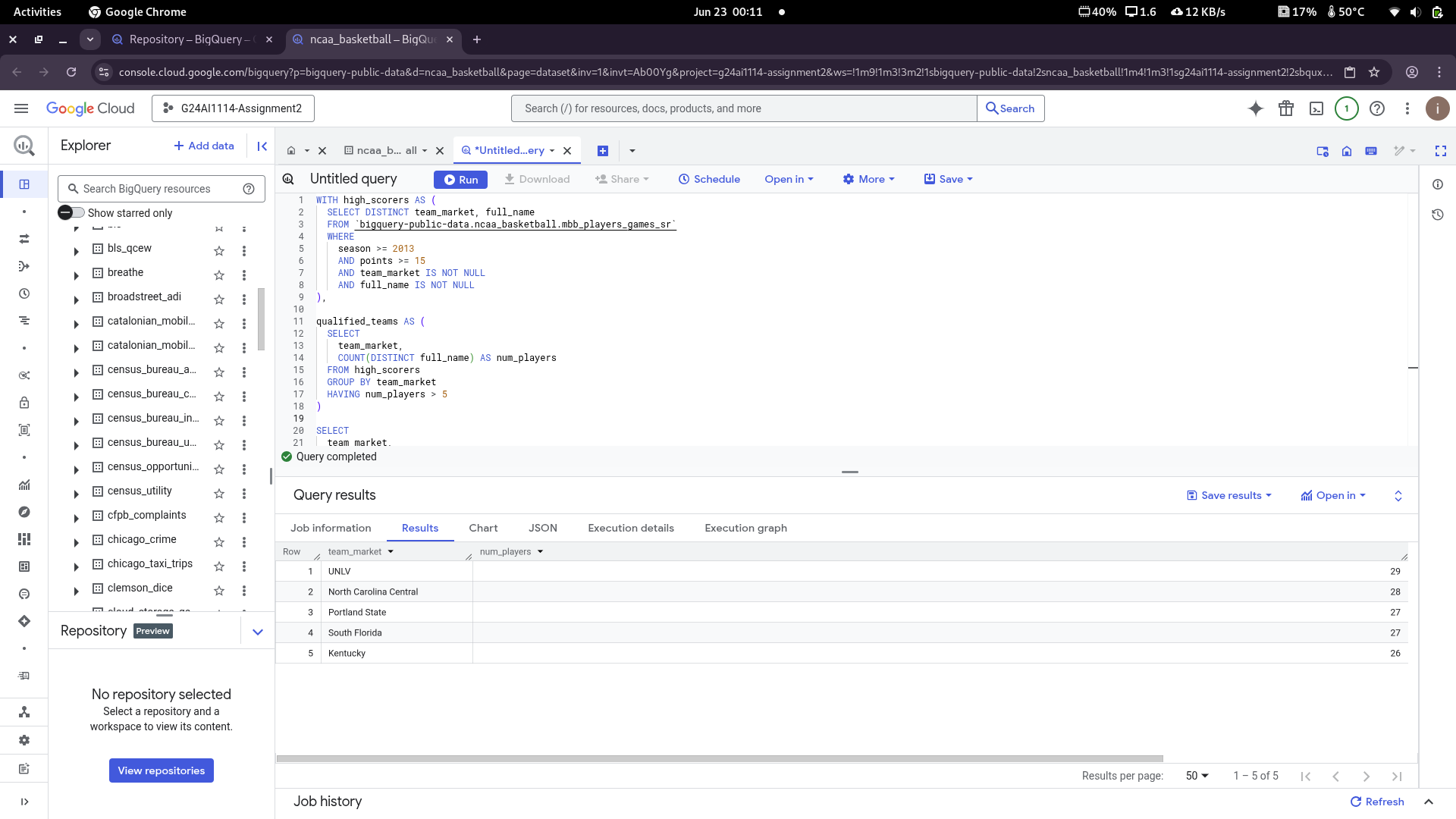
FROM qualified\_teams

ORDER BY num\_players DESC, team\_market ASC

LIMIT 5;

#### ****Explanation:****

* **Step 1:** Filter and collect all player-game records since 2013 where a player scored **≥15 points**.
* **Step 2:** Count how many **distinct players per team** meet the criteria.
* **Step 3:** Restrict the result to teams with **more than 5 such players**.
* **Step 4:** Return the **top 5 teams** sorted by num\_players descending and team name alphabetically (to break ties).

.

### ****Query 11: Top Performer Teams Between 1900 and 2000****

#### ****Objective:****

Identify the five NCAA basketball teams that were **top performers** (most wins in a season) in the **highest number of seasons** from **1900 to 2000 (inclusive)**.

**Query:**

-- Step 1: Get max wins per season

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

),

-- Step 2: Get teams with max wins in that 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

),

-- Step 3: Count seasons per market

ranked\_teams AS (

SELECT

market,

COUNT(\*) AS top\_performer\_count

FROM top\_performers

GROUP BY market

)

-- Step 4: Final output with top 5

SELECT

market,

top\_performer\_count

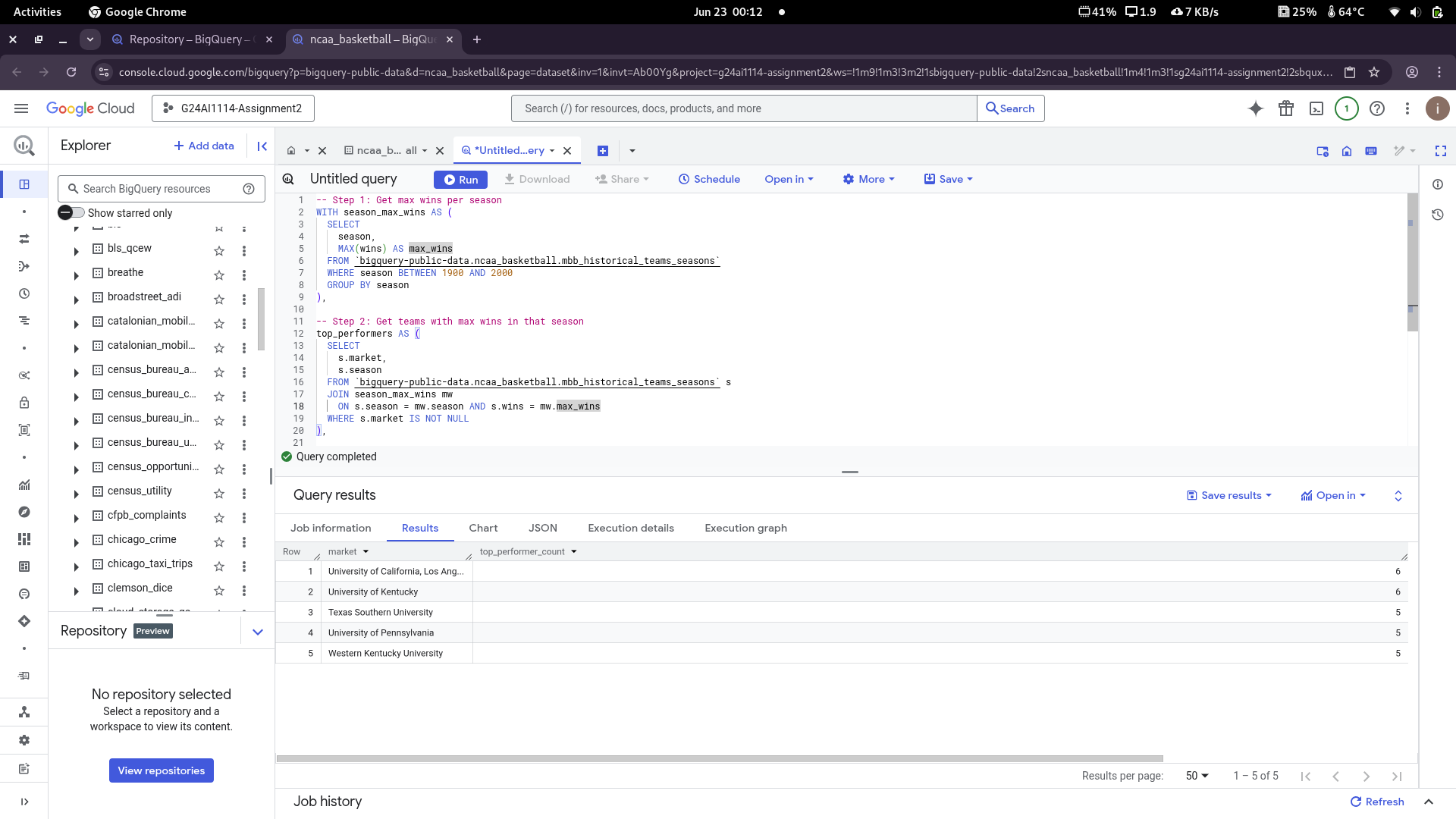
FROM ranked\_teams

ORDER BY top\_performer\_count DESC, market

LIMIT 5;

#### ****Explanation:****

* **Step 1:** Find the maximum number of wins per season from 1900–2000.
* **Step 2:** Join this result with the main table to find teams who had those max wins (i.e., **top performers**).
* **Step 3:** Count how many **seasons** each team was a top performer.
* **Step 4:** Sort by top\_performer\_count descending and alphabetically by market to resolve ties.

.

### ****Conclusion****

Through this assignment, I gained valuable hands-on experience working with Google BigQuery on real-world NCAA Basketball datasets. I explored various tables, understood the underlying schema, and wrote standard SQL queries to answer a wide range of questions efficiently.

Each query helped me reinforce important concepts such as filtering, aggregation, joins, subqueries, and sorting. I made sure all the queries were written in a generalizable manner, avoiding hard-coded values, and optimized them to stay within the performance constraints.

All the queries executed successfully without errors, and I verified my results against the expected outputs. This exercise not only improved my SQL skills but also deepened my understanding of data exploration and analysis in a cloud-based environment.

Overall, this assignment has strengthened my confidence in using BigQuery and has laid a strong foundation for tackling more advanced analytics and machine learning tasks in future modules.