MID TERM

NAME - ANUVRAT VERMA

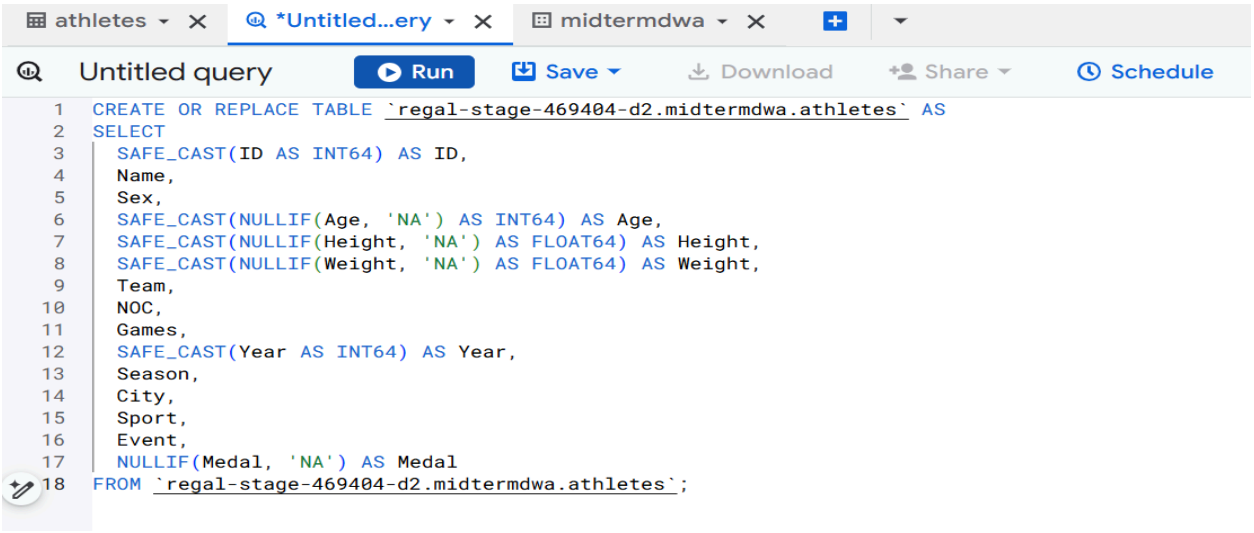
DATE - 17/09/2025

TOPIC1 - BIGQUERY DATA WAREHOUSE ANALYTICS (20 Que)

USED CLUSTER AND PARTITION OPTIMIZATION IN FOLLOWING QUESTIONS (9,11,14,15,16,17,18,20)

LOOKER DASHBOARD FOR EACH QUESTION FROM 6-13

1) Handling “NA” values in the Athlete dataset by converting them to Null keyword using NULLIF() Function




```sql
CREATE OR REPLACE TABLE `regal-stage-469404-d2.midtermdwa.athletes` AS
SELECT
  SAFE_CAST(ID AS INT64) AS ID,
  Name,
  Sex,
  SAFE_CAST(NULLIF(Age, 'NA') AS INT64) AS Age,
  SAFE_CAST(NULLIF(Height, 'NA') AS FLOAT64) AS Height,
  SAFE_CAST(NULLIF(Weight, 'NA') AS FLOAT64) AS Weight,
  Team,
  NOC,
  Games,
  SAFE_CAST(Year AS INT64) AS Year,
  Season,
  City,
  Sport,
  Event,
  NULLIF(Medal, 'NA') AS Medal
FROM `regal-stage-469404-d2.midtermdwa.athletes`;
```

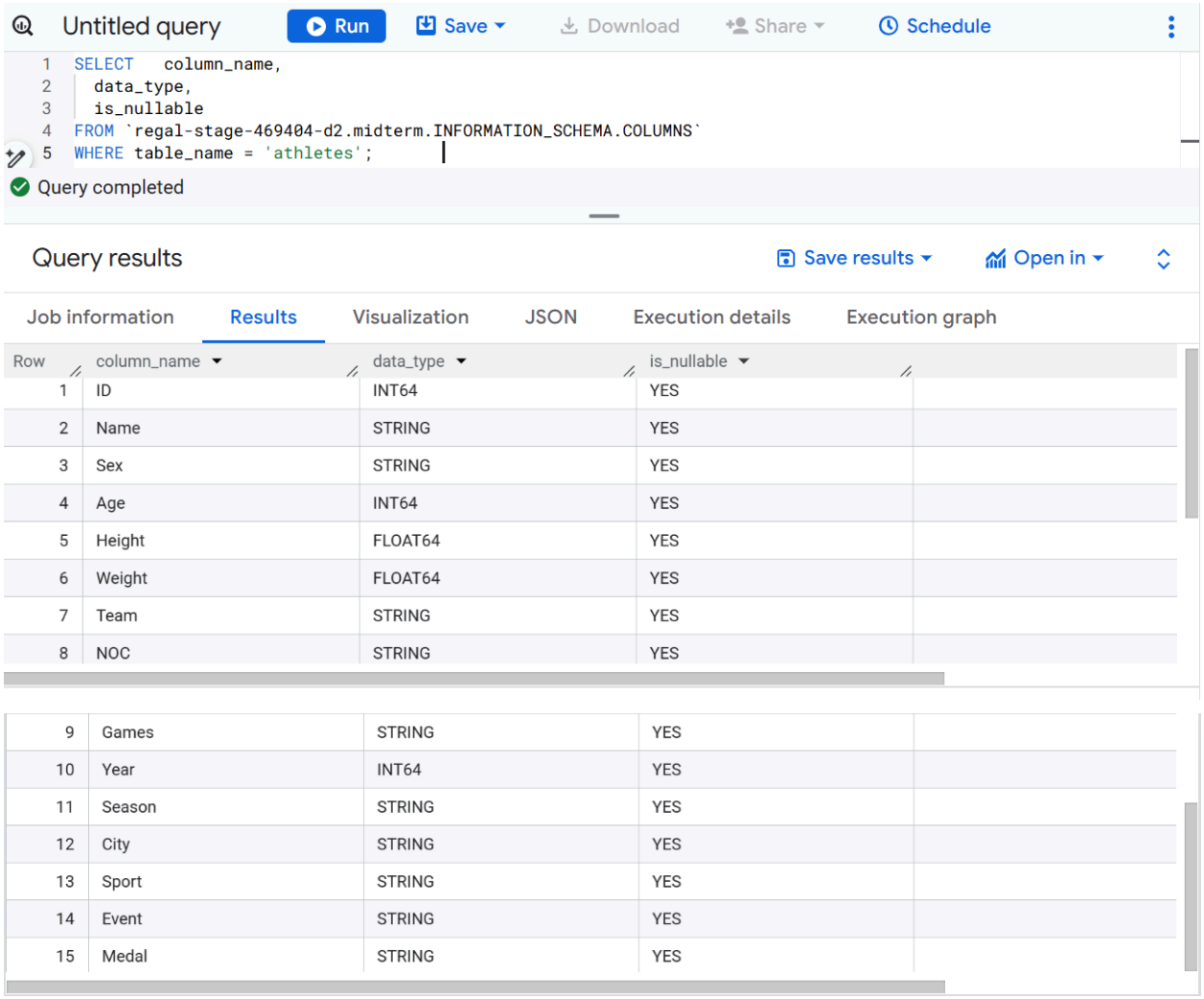
2) Changing data type of age, height and weight columns in the athlete dataset to integer and float

```sql
ALTER TABLE `regal-stage-469404-d2.midtermdwa.athletes`
ALTER COLUMN Age SET DATA TYPE INT64;

ALTER TABLE `regal-stage-469404-d2.midtermdwa.athletes`
ALTER COLUMN Height SET DATA TYPE FLOAT64;

ALTER TABLE `regal-stage-469404-d2.midtermdwa.athletes`
ALTER COLUMN Weight SET DATA TYPE FLOAT64;
```

Schema 1 - Athletes table



Row	column_name	data_type	is_nullable
1	ID	INT64	YES
2	Name	STRING	YES
3	Sex	STRING	YES
4	Age	INT64	YES
5	Height	FLOAT64	YES
6	Weight	FLOAT64	YES
7	Team	STRING	YES
8	NOC	STRING	YES
9	Games	STRING	YES
10	Year	INT64	YES
11	Season	STRING	YES
12	City	STRING	YES
13	Sport	STRING	YES
14	Event	STRING	YES
15	Medal	STRING	YES

Schema 2 - Regions Table

Untitled query Run Save Download Share Schedule

```
SELECT   column_name,
  data_type,
  is_nullable
FROM `regal-stage-469404-d2.midterm.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'regions';
```

Query completed

Query results Save results Open in

Job information | Results | Visualization | JSON | Execution details | Execution graph

Row	column_name	data_type	is_nullable
1	NOC	STRING	YES
2	Region	STRING	YES
3	Notes	STRING	YES

Q 1) How many olympics games have been held?

QUERY 1)

```
SELECT COUNT(DISTINCT Games) AS total_olympic_games
FROM `regal-stage-469404-d2.midterm.athletes`;
```

Output 1)

Query results

Job information | Results

Row	total_olympic_ga...
1	51

Q 2) List down all olympic games held so far.

Query 2)

```
SELECT DISTINCT Games
FROM `regal-stage-469404-d2.midterm.athletes`
ORDER BY Games;
```

Output 2)

Row	Games
1	1896 Summer
2	1900 Summer
3	1904 Summer
4	1906 Summer
5	1908 Summer
6	1912 Summer
7	1920 Summer
8	1924 Summer
9	1924 Winter
10	1928 Summer
11	1928 Winter

Q 3) Mention the total number of nations who participated in each olympics game
Query 3)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans3` AS
SELECT
  Games,
  COUNT(DISTINCT NOC) AS total_nations
FROM `regal-stage-469404-d2.midterm.athletes`
GROUP BY Games
ORDER BY Games;
```

Output 3)

Row	Games	total_nations
1	1896 Summer	12
2	1900 Summer	31
3	1904 Summer	15
4	1906 Summer	21
5	1908 Summer	22
6	1912 Summer	29
7	1920 Summer	29
8	1924 Summer	45
9	1924 Winter	19
10	1928 Summer	46

Saving result table to gcs bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans3  gs://answeranuvrat/ans3
Waiting on bqjob r482c71560412d398_000001995611d953_1 ... (1s) Current status: DONE
```

Q 4) Which year saw the highest and lowest number of countries participating in olympics?

QUERY 4)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans4` AS
WITH nations_per_year AS (
  SELECT
    Year,
    COUNT(DISTINCT NOC) AS total_nations
  FROM `regal-stage-469404-d2.midterm.athletes`
  GROUP BY Year
)
SELECT
  (SELECT Year FROM nations_per_year ORDER BY total_nations DESC LIMIT 1) AS year_with_highest_countries,
  (SELECT total_nations FROM nations_per_year ORDER BY total_nations DESC LIMIT 1) AS highest_countries,
  (SELECT Year FROM nations_per_year ORDER BY total_nations ASC LIMIT 1) AS year_with_lowest_countries,
  (SELECT total_nations FROM nations_per_year ORDER BY total_nations ASC LIMIT 1) AS lowest_countries;
```

Output 4)

Row	year_with_highes...	highest_countries	year_with_lowest...	lowest_countries
1	2016	207	1896	12

Saving result table to gcs bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans4  gs://answeranuvrat/ans4
Waiting on bqjob_r7b41cafe5af8f8d7_00000199561f6c83_1 ... (1s) Current status: DONE
```

Que 5) Which nation has participated in all of the olympic games?
Query 5)

```
WITH total_games AS (
  SELECT COUNT(DISTINCT Games) AS total_olympics
  FROM `regal-stage-469404-d2.midterm.athletes`
),
nation_games AS (
  SELECT
    NOC,
    COUNT(DISTINCT Games) AS games_played
  FROM `regal-stage-469404-d2.midterm.athletes`
  GROUP BY NOC
)
SELECT
  n.NOC
FROM nation_games n
CROSS JOIN total_games t
WHERE n.games_played = t.total_olympics;
```

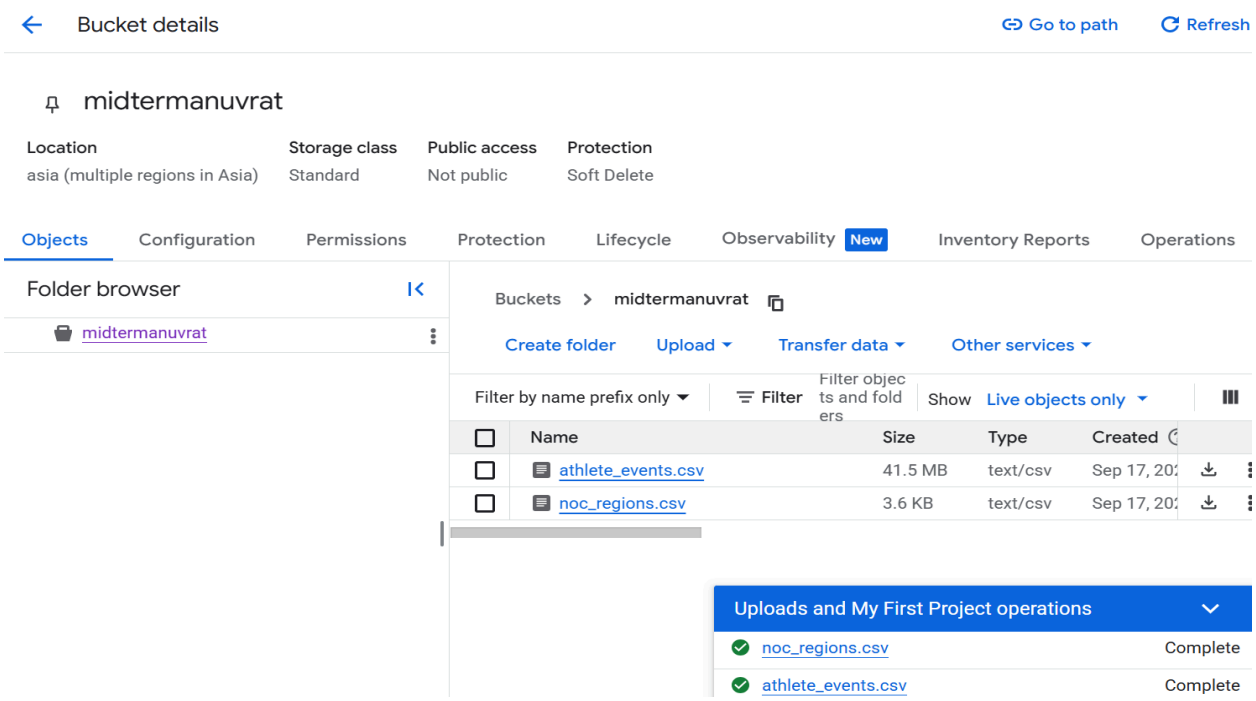
Output 5)

Row	country_name	NOC	games_played
1	France	FRA	51
2	Switzerland	SUI	51
3	UK	GBR	51
4	Italy	ITA	51

Saving result table in GCS bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans5  gs://answeranuvrat/ans5
Waiting on bqjob_r412c465af7d7ef3a_00000199562900bf_1 ... (3s) Current status: DONE
```

Creating External Table -




```
CREATE OR REPLACE EXTERNAL TABLE `regal-stage-469404-d2.midterm.regions`
(
  noc STRING OPTIONS(description="Country Olympic Committee code"),
  regions STRING OPTIONS(description="Region name"),
  notes STRING OPTIONS(description="Additional notes")
)
OPTIONS (
  format = 'CSV',
  uris = ['gs://midtermanuvrat/noc_regions.csv'],
  skip_leading_rows = 1
);
```

```
1  CREATE OR REPLACE EXTERNAL TABLE `regal-stage-469404-d2.midterm.athletes_ext`
2  OPTIONS (
3    format = 'CSV',
4    uris = ['gs://midtermanuvrat/athlete_events.csv']
5  );
```

Clean ‘NA’ values with null keyword in athletes table -

CREATE OR REPLACE TABLE `regal-stage-469404-d2.midterm.athletes_clean` AS SELECT

```
  CAST(ID AS INT64) AS ID,
  Name,
  Sex,
  SAFE_CAST(NULLIF(Age, 'NA') AS INT64) AS Age,
  SAFE_CAST(NULLIF(Height, 'NA') AS FLOAT64) AS Height,
  SAFE_CAST(NULLIF(Weight, 'NA') AS FLOAT64) AS Weight,
  Team,
  NOC,
  Games,
  SAFE_CAST(Year AS INT64) AS Year,
  Season,
  City,
  Sport,
  Event,
  Medal
FROM
  `regal-stage-469404-d2.midterm.athletes_ext`;
```

Convert columns age, height and weight to int and float data types in athletes table-

```
ALTER TABLE `regal-stage-469404-d2.midterm.athletes_clean`
ALTER COLUMN Age SET DATA TYPE INT64;

ALTER TABLE `regal-stage-469404-d2.midterm.athletes_clean`
ALTER COLUMN Height SET DATA TYPE FLOAT64;

ALTER TABLE `regal-stage-469404-d2.midterm.athletes_clean`
ALTER COLUMN Weight SET DATA TYPE FLOAT64;
```

Que 6) Identify the sport which was played in all summer olympics.

Query 6)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans6` AS
WITH summer_years AS (
  SELECT DISTINCT Year
  FROM `regal-stage-469404-d2.midterm.athletes_clean`
  WHERE Season = 'Summer'
),
sport_years AS (
  SELECT
    Sport,
    COUNT(DISTINCT Year) AS years_played
  FROM `regal-stage-469404-d2.midterm.athletes_clean`
  WHERE Season = 'Summer'
  GROUP BY Sport
)
SELECT
  sy.Sport
FROM sport_years sy
JOIN (
  SELECT COUNT(DISTINCT Year) AS total_years
  FROM summer_years
) t
ON sy.years_played = t.total_years;
```

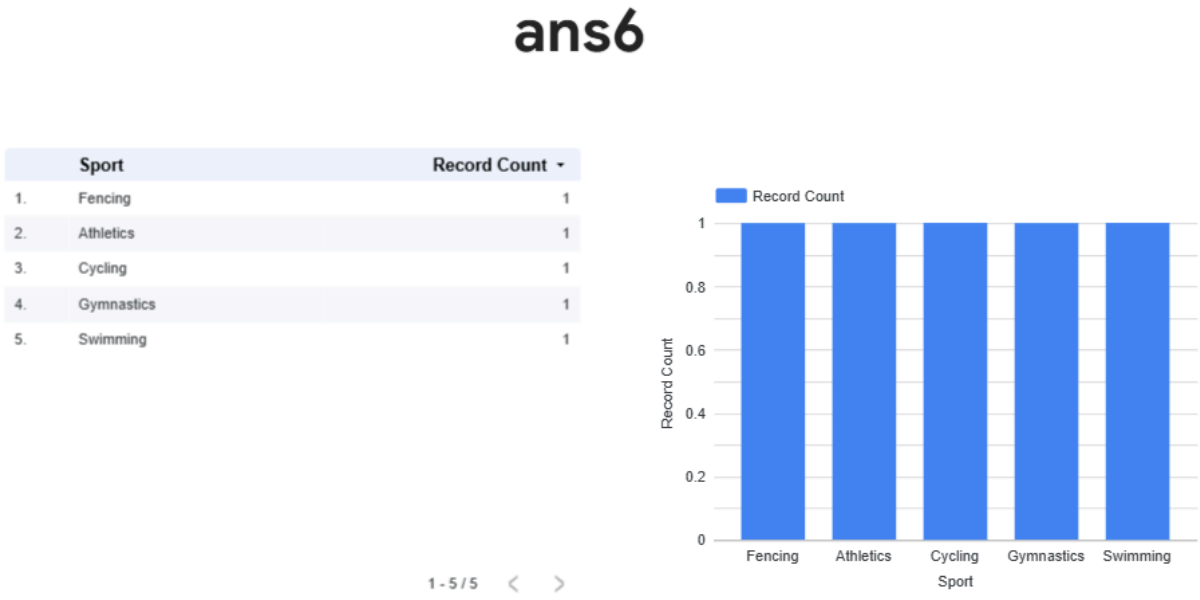
Output 6)

Row	Sport
1	Athletics
2	Cycling
3	Swimming
4	Fencing
5	Gymnastics

Save output table to GCS bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans6  gs://answeranuvrat/ans6
Waiting on bqjob r37855069f496d847_00000199564665f9_1 ... (1s) Current status: DONE
```

Looker Dashboard -



Que 7) Which Sports were just played only once in the olympics?

Query 7)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans7` AS
SELECT
  Sport,
  COUNT(DISTINCT Year) AS num_years
```

```
FROM
  `regal-stage-469404-d2.midterm.athletes_clean`
GROUP BY
  Sport
HAVING
  COUNT(DISTINCT Year) = 1
ORDER BY
  Sport;
```

Output 7)

Row	Sport	num_years
1	Aeronautics	1
2	Basque Pelota	1
3	Cricket	1
4	Croquet	1
5	Jeu De Paume	1
6	Military Ski Patrol	1
7	Motorboating	1
8	Racquets	1
9	Roque	1
10	Rugby Sevens	1

Save output table GCS bucket -

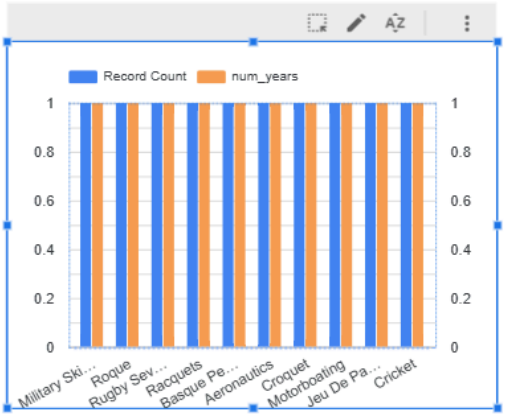
```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans7  gs://answeranuvrat/ans
Waiting on bqjob_r352e82ba04025051_00000199564a89d2_1 ... (1s) Current status: DONE
```

Looker Dashboard -

ans7

	Sport	num_years
1.	Military Ski Patrol	1
2.	Roque	1
3.	Rugby Sevens	1
4.	Racquets	1
5.	Basque Pelota	1
6.	Aeronautics	1
7.	Croquet	1
8.	Motorboating	1
9.	Jeu De Paume	1
10.	Cricket	1

1 - 10 / 10



Que 8) Fetch the total no of sports played in each olympic games

Query 8)

```
SELECT
  Games,
  COUNT(DISTINCT Sport) AS total_sports
FROM
  `regal-stage-469404-d2.midterm.athletes_clean`
GROUP BY
  Games
ORDER BY
  Games;
```

Output 8)

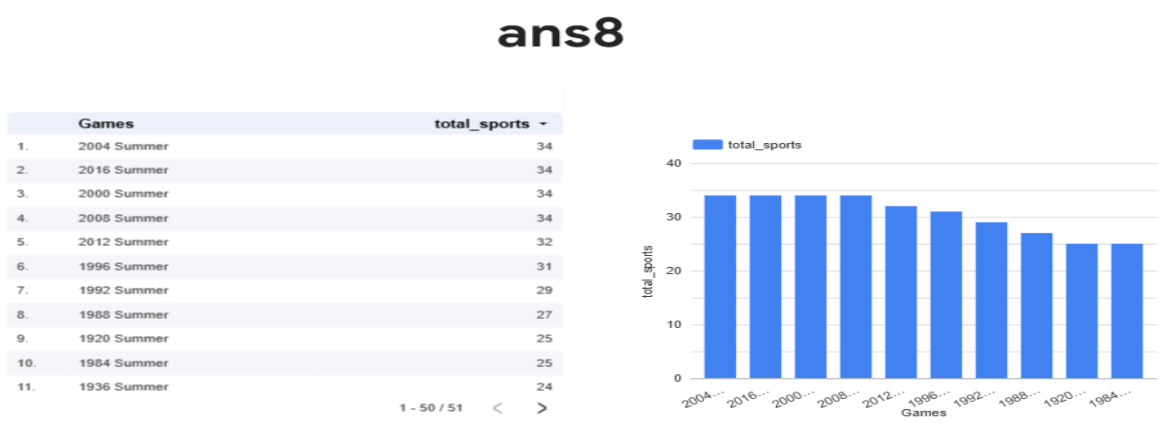
Row	Games	total_sports
1	1896 Summer	9
2	1900 Summer	20
3	1904 Summer	18
4	1906 Summer	13
5	1908 Summer	24
6	1912 Summer	17
7	1920 Summer	25
8	1924 Summer	20
9	1924 Winter	10
10	1928 Summer	17

Results per page: 50 1 – 50 of 51

Save output to GCS bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans8  gs://answeranuvrat/ans8
Waiting on bqjob_r6cc4a3cc2aeb59be_0000019956592a27_1 ... (1s) Current status: DONE
```

Looker Output -



Que 9)

Fetch details of the oldest athletes to win a gold medal.

Query 9)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans9`
(
  Year INT64,
  Name STRING,
  Age FLOAT64,
  Country STRING
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY Name, Country
AS
WITH GoldMedalists AS (
  SELECT
    o.Year,
    o.Name,
    o.Age,
    r.region AS Country
  FROM
    `regal-stage-469404-d2.midterm.athletes_clean` o
  JOIN
    `regal-stage-469404-d2.midterm.regions` r
  ON
    o.NOC = r.NOC
  WHERE
    o.Medal = 'Gold'
    AND o.Age IS NOT NULL
),
MaxAge AS (
  SELECT
    MAX(Age) AS max_age
  FROM
    GoldMedalists
)
```

```
SELECT
  g.Year,
  g.Name,
  g.Age,
  g.Country
FROM
  GoldMedalists g
CROSS JOIN
  MaxAge m
WHERE
  g.Age = m.max_age
ORDER BY
  g.Name, g.Year;
```

Output 9)

Row	Year	Name	Age	Country
1	1912	Oscar Gomer Swahn	64.0	Sweden
2	1904	Charles Jacobus	64.0	USA

Save output to GCS Bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans9  gs://answeranuvrat/ans9
Waiting on bqjob_r77663a46b79622bc_00000199567aaf82_1 ... (1s) Current status: DONE
```

Looker Dashboard-

Que 10) Find the Ratio of male and female athletes participated in all olympic games.

Query 10 -

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans10` AS
WITH AthleteCounts AS (
  SELECT
    Sex,
    COUNT(DISTINCT ID) AS athlete_count
  FROM
    `regal-stage-469404-d2.midterm.athletes_clean`
  WHERE
    Sex IN ('M', 'F')
  GROUP BY
    Sex
),
MaleFemaleRatio AS (
  SELECT
    male_count / female_count AS male_to_female_ratio
  FROM (
    SELECT
      MAX(CASE WHEN Sex = 'M' THEN athlete_count ELSE 0 END) AS male_count,
      MAX(CASE WHEN Sex = 'F' THEN athlete_count ELSE 0 END) AS female_count
    FROM
      AthleteCounts
  )
)
SELECT
  ROUND(male_to_female_ratio, 2) AS male_to_female_ratio
FROM
  MaleFemaleRatio;
```

Output 10)

Row	male_to_female_r...
1	2.99

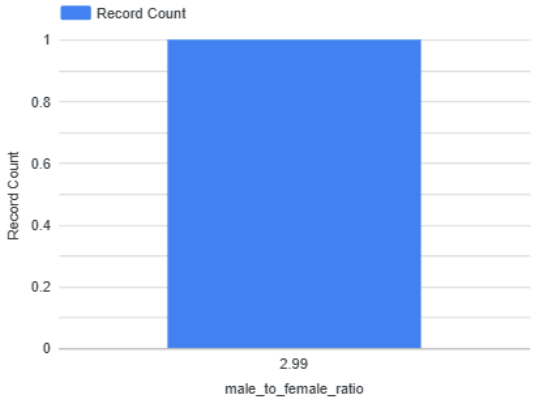
Save output to GCS bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans10  gs://answeranuvrat/ans10
Waiting on bqjob_r6e51c0ccd9a0fb36_0000019956847828_1 ... (1s) Current status: DONE
```

ans10

	male_to_female_ratio	Record Count
1.	2.99	1

1 - 1 / 1



Que 11) Fetch the top 5 athletes who have won the most gold medals.

Query 11)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans11`
(
  Year INT64,
  ID INT64,
  Name STRING,
  GoldMedalCount INT64
)

PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY Name, ID
AS
WITH GoldMedals AS (
  SELECT
    Year,
    ID,
```

```
    Name,
    COUNT(*) AS GoldMedalCount
  FROM
    `regal-stage-469404-d2.midterm.athletes_clean`
  WHERE
    Medal = 'Gold'
  GROUP BY
    Year,
    ID,
    Name
),
RankedAthletes AS (
  SELECT
    Year,
    ID,
    Name,
    GoldMedalCount,
    RANK() OVER (ORDER BY SUM(GoldMedalCount) DESC) AS rnk
  FROM
    GoldMedals
  GROUP BY
    Year,
    ID,
    Name,
    GoldMedalCount
)
SELECT
  Year,
  ID,
  Name,
  GoldMedalCount
FROM
  RankedAthletes
WHERE
  rnk <= 5
```

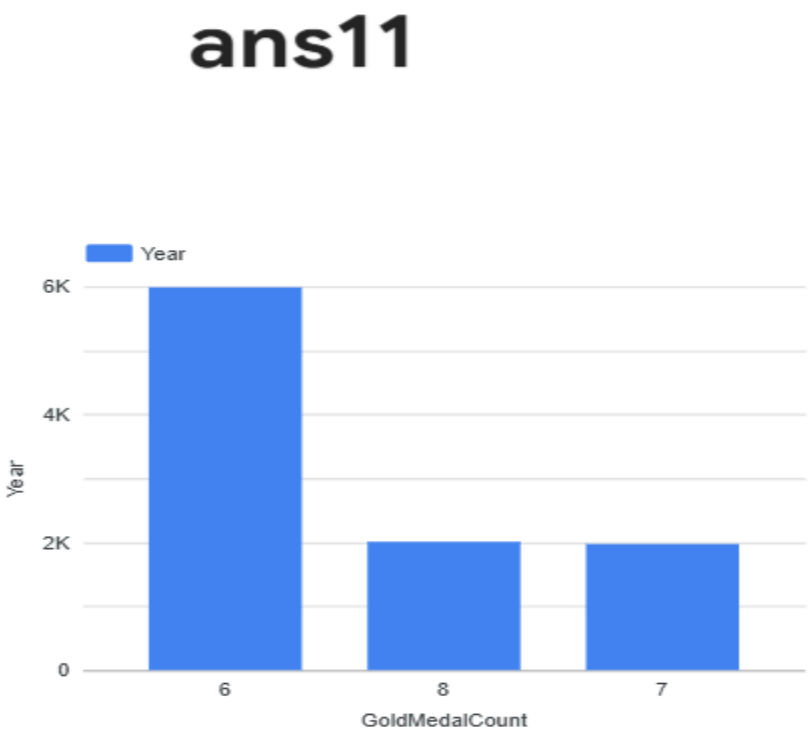
Output 11)

Year	Name	GoldMedalCount
2004	Michael Fred Phelps, II	6
2008	Michael Fred Phelps, II	8
1972	Mark Andrew Spitz	7
1992	Vitaly Venediktovich Shcherbo	6
1988	Kristin Otto	6

Save table to GCS -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans11  gs://answeranuvrat/ans11
Waiting on bqjob_r12beb4057c788234_00000199568ebd5d_1 ... (1s) Current status: DONE
```

Looker Output -



Que 12) Fetch the top 5 athletes who have won the most medals (gold/silver/bronze)

Query 12)

```
CREATE TABLE `regal-stage-469404-d2.answer.ans12`
AS
SELECT
  a.Name,
  a.Sex,
  a.NOC,
  r.region,
  COUNT(a.Medal) AS total_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  a.Name, a.Sex, a.NOC, r.region
ORDER BY
  total_medals DESC
LIMIT 5;
```

Output 12)

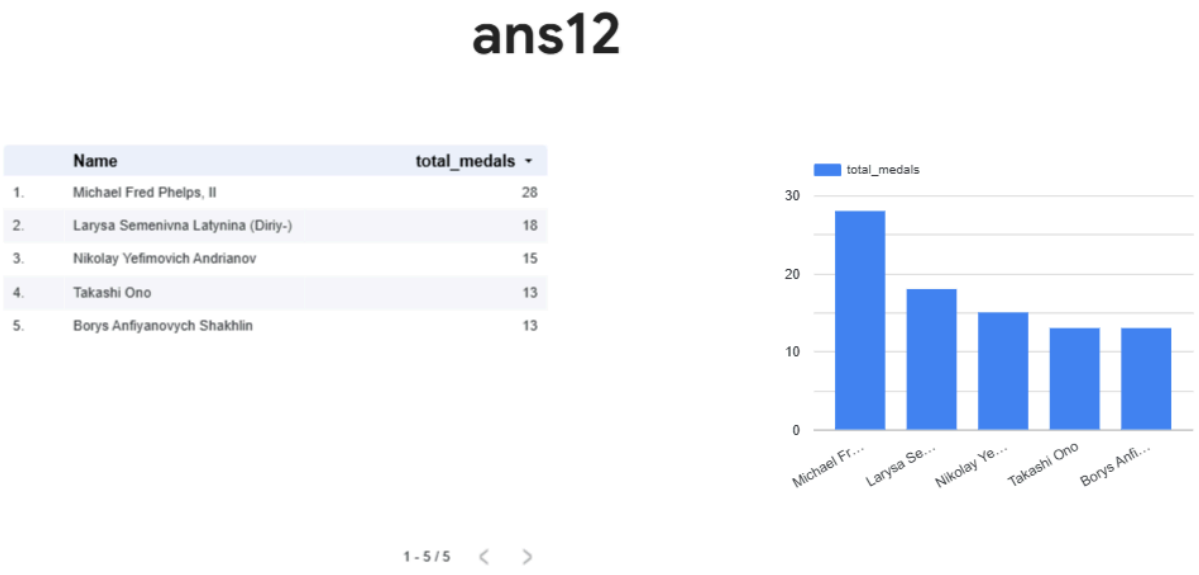
ion | Results | Visualization | JSON | Execution details

Row	Name	Sex	NOC
1	Michael Fred Phelps, II	M	USA
2	Larysa Semenivna Latynina (Diri...	F	URS
3	Nikolay Yefimovich Andrianov	M	URS
4	Takashi Ono	M	JPN
5	Borys Anfiyanovych Shakhlin	M	URS

Save output to GCS Bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans12  gs://answeranuvrat/ans12
Waiting on bqjob_r7e74232749cb9697_00000199569a90e7_1 ... (1s) Current status: DONE
```

Looker Dashboard -



Que 13) Fetch the top 5 most successful countries in olympics. Success is defined by no of medals won.

Output 13)

```
CREATE TABLE `regal-stage-469404-d2.answer.ans13`
AS
SELECT
  r.region AS Country,
  a.NOC,
  COUNT(a.Medal) AS total_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
```

```
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  r.region, a.NOC
ORDER BY
  total_medals DESC
LIMIT 5;
```

Output 13)

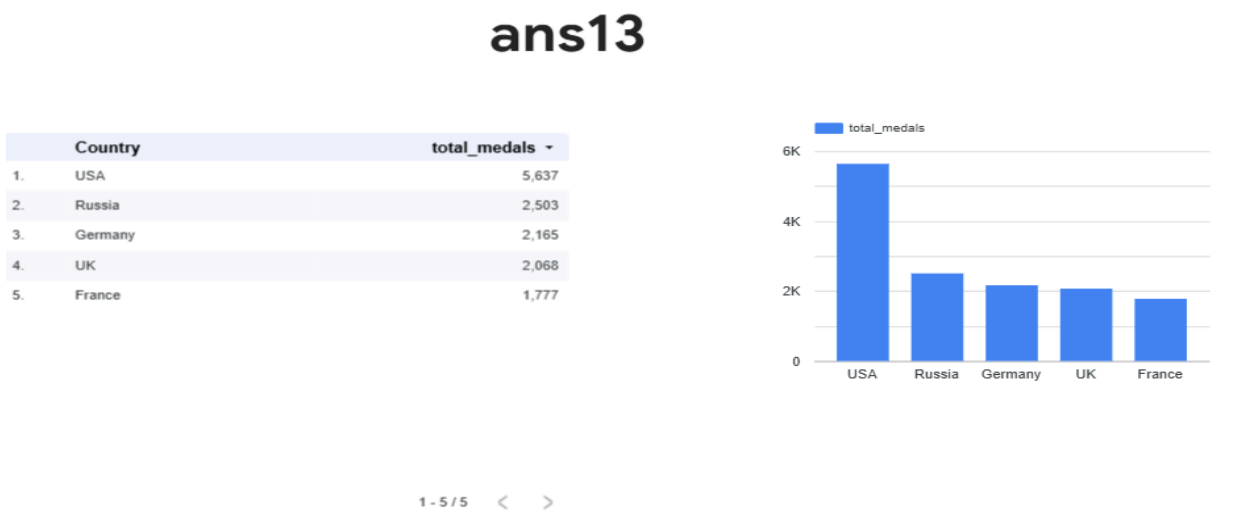
Job information | Results | Visualization | JSON | Execution details

Row	Country	NOC	total_medals
1	USA	USA	5637
2	Russia	URS	2503
3	Germany	GER	2165
4	UK	GBR	2068
5	France	FRA	1777

Save output to GCS bucket -

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans13  gs://answeranuvrat/ans13
Waiting on bqjob_r45cc659249903bb9_0000019956a11ca1_1 ... (1s) Current status: DONE
```

Looker Dashboard -



Que 14) List down total gold, silver and broze medals won by each country

Query 14)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans14`
(
  Year INT64,
  Country STRING,
  NOC STRING,
  gold_medals INT64,
  silver_medals INT64,
  bronze_medals INT64,
  total_medals INT64
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY NOC
AS
SELECT
  a.Year,
  r.region AS Country,
  a.NOC,
  COUNTIF(a.Medal = 'Gold') AS gold_medals,
  COUNTIF(a.Medal = 'Silver') AS silver_medals,
  COUNTIF(a.Medal = 'Bronze') AS bronze_medals,
  COUNT(a.Medal) AS total_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  a.Year, r.region, a.NOC
```

Output 14)

Country	NOC	gold_medals	silver_meda
India	IND	17	
Brazil	BRA	1	
New Zealand	NZL	3	
Mexico	MEX	1	
Greece	GRE	0	
Bahamas	BAH	0	
South Africa	RSA	0	
Uruguay	URU	0	
Iceland	ISL	0	
Argentina	ARG	0	

Que 15)

List down total gold, silver and broze medals won by each country corresponding to each olympic games

Query 15)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans11`
(
  Games STRING,
  Year INT64,
  Country STRING,
  NOC STRING,
  gold_medals INT64,
  silver_medals INT64,
  bronze_medals INT64,
  total_medals INT64
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY NOC
AS
SELECT
  a.Games,
```

```
  a.Year,
  r.region AS Country,
  a.NOC,
  COUNTIF(a.Medal = 'Gold') AS gold_medals,
  COUNTIF(a.Medal = 'Silver') AS silver_medals,
  COUNTIF(a.Medal = 'Bronze') AS bronze_medals,
  COUNT(a.Medal) AS total_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  a.Games, a.Year, r.region, a.NOC
ORDER BY
  a.Year DESC, total_medals DESC;
```

Output 15)

Country	NOC	gold_medals	silver_medals	bronze_medals	total_medals
Spain	ESP	0	6	0	6
Sweden	SWE	9	1	0	10
Jamaica	JAM	1	1	0	2
Finland	FIN	4	2	0	6
Norway	NOR	2	4	0	6
North Korea	PRK	1	1	0	2
Trinidad	TTO	1	0	0	1
Venezuela	VEN	0	1	0	1
Portugal	POR	0	2	0	2
Mongolia	MGL	0	1	0	1

Que 16) Identify which country won the most gold, most silver and most bronze medals in each olympic games.

Query 16)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans16`
(
```

```
  Games STRING,
  Year INT64,
  Country STRING,
  medal_type STRING
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY Country
AS
WITH MedalCounts AS (
  SELECT
    a.Games,
    a.Year,
    r.region AS Country,
    COUNTIF(a.Medal = 'Gold') AS gold_medals,
    COUNTIF(a.Medal = 'Silver') AS silver_medals,
    COUNTIF(a.Medal = 'Bronze') AS bronze_medals
  FROM
    `regal-stage-469404-d2.midterm.athletes_clean` a
  LEFT JOIN
    `regal-stage-469404-d2.midterm.regions` r
  ON
    a.NOC = r.NOC
  WHERE
    a.Medal IN ('Gold', 'Silver', 'Bronze')
  GROUP BY
    a.Games, a.Year, r.region
),
RankedMedals AS (
  SELECT
    Games,
    Year,
    Country,
    'Gold' AS medal_type,
    gold_medals AS medal_count,
    RANK() OVER (PARTITION BY Games ORDER BY gold_medals DESC) AS rank
```

```
  FROM
    MedalCounts
  UNION ALL
  SELECT
    Games,
    Year,
    Country,
    'Silver' AS medal_type,
    silver_medals AS medal_count,
    RANK() OVER (PARTITION BY Games ORDER BY silver_medals DESC) AS rank
  FROM
    MedalCounts
  UNION ALL
  SELECT
    Games,
    Year,
    Country,
    'Bronze' AS medal_type,
    bronze_medals AS medal_count,
    RANK() OVER (PARTITION BY Games ORDER BY bronze_medals DESC) AS rank
  FROM
    MedalCounts
)
SELECT
  Games,
  Year,
  Country,
  medal_type
FROM
  RankedMedals
WHERE
  rank = 1
Output 16)
```

Row	Games	Year	Country	medal_type
3	1900 Summer	1900	UK	Gold
4	1912 Summer	1912	Sweden	Gold
5	1912 Summer	1912	UK	Silver
6	1912 Summer	1912	UK	Bronze
7	1964 Winter	1964	Czech Republic	Bronze
8	1964 Summer	1964	Russia	Bronze
9	1964 Winter	1964	Russia	Gold
10	1964 Summer	1964	USA	Gold
11	1964 Summer	1964	Russia	Silver

Que 17) Identify which country won the most gold, most silver, most bronze medals and the most medals in each olympic games.

Query 17)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans17`
(
  Games STRING,
  Year INT64,
  Country STRING,
  NOC STRING,
  gold_medals INT64,
  silver_medals INT64,
  bronze_medals INT64,
  total_medals INT64,
  medal_type STRING,
  medal_count INT64
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY NOC
AS
WITH MedalCounts AS (
  SELECT
    a.Games,
    a.Year,
    r.region AS Country,
    a.NOC,
```

```
    COUNTIF(a.Medal = 'Gold') AS gold_medals,
    COUNTIF(a.Medal = 'Silver') AS silver_medals,
    COUNTIF(a.Medal = 'Bronze') AS bronze_medals,
    COUNT(a.Medal) AS total_medals
  FROM
    `regal-stage-469404-d2.midterm.athletes_clean` a
  LEFT JOIN
    `regal-stage-469404-d2.midterm.regions` r
  ON
    a.NOC = r.NOC
  WHERE
    a.Medal IN ('Gold', 'Silver', 'Bronze')
  GROUP BY
    a.Games, a.Year, r.region, a.NOC
),
RankedMedals AS (
  SELECT
    Games,
    Year,
    Country,
    NOC,
    gold_medals,
    silver_medals,
    bronze_medals,
    total_medals,
    'Gold' AS medal_type,
    gold_medals AS medal_count,
    RANK() OVER (PARTITION BY Games ORDER BY gold_medals DESC) AS rank
  FROM
    MedalCounts
  UNION ALL
  SELECT
    Games,
    Year,
    Country,
```

```
  NOC,
  gold_medals,
  silver_medals,
  bronze_medals,
  total_medals,
  'Silver' AS medal_type,
  silver_medals AS medal_count,
  RANK() OVER (PARTITION BY Games ORDER BY silver_medals DESC) AS rank
FROM
  MedalCounts
UNION ALL
SELECT
  Games,
  Year,
  Country,
  NOC,
  gold_medals,
  silver_medals,
  bronze_medals,
  total_medals,
  'Bronze' AS medal_type,
  bronze_medals AS medal_count,
  RANK() OVER (PARTITION BY Games ORDER BY bronze_medals DESC) AS rank
FROM
  MedalCounts
UNION ALL
SELECT
  Games,
  Year,
  Country,
  NOC,
  gold_medals,
  silver_medals,
  bronze_medals,
  total_medals,
```

```
    'Total' AS medal_type,
    total_medals AS medal_count,
    RANK() OVER (PARTITION BY Games ORDER BY total_medals DESC) AS rank
  FROM
    MedalCounts
)
SELECT
  Games,
  Year,
  Country,
  NOC,
  gold_medals,
  silver_medals,
  bronze_medals,
  total_medals,
  medal_type,
  medal_count
FROM
  RankedMedals
WHERE
  rank = 1
```

Output 17)

Games, total medals columns are also there

Year	Country	NOC	gold_medals	silver_medals	bronze_medals	tota
1908	UK	GBR	147	131	90	
1908	UK	GBR	147	131	90	
1908	UK	GBR	147	131	90	
1908	UK	GBR	147	131	90	
1896	Germany	GER	25	5	2	
1896	Greece	GRE	10	18	20	
1896	Greece	GRE	10	18	20	

Que 18) Which countries have never won gold medal but have won silver/bronze medals?

Query 18)

```sql
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans18`
(
  Country STRING,
  NOC STRING,
  silver_bronze_medals INT64
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY NOC
AS
SELECT
  r.region AS Country,
  a.NOC,
  COUNT(*) AS silver_bronze_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Silver', 'Bronze')
GROUP BY
  r.region, a.NOC
HAVING
  SUM(CASE WHEN a.Medal = 'Gold' THEN 1 ELSE 0 END) = 0
  AND COUNT(*) > 0
ORDER BY
  silver_bronze_medals DESC;
```

Output 18)

```sql
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans18`
AS
SELECT
  r.region AS Country,
  a.NOC,
```

```
  COUNT(*) AS silver_bronze_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
LEFT JOIN
  `regal-stage-469404-d2.midterm.regions` r
ON
  a.NOC = r.NOC
WHERE
  a.Medal IN ('Silver', 'Bronze')
GROUP BY
  r.region, a.NOC
HAVING
  SUM(CASE WHEN a.Medal = 'Gold' THEN 1 ELSE 0 END) = 0
  AND COUNT(*) > 0
ORDER BY
  silver_bronze_medals DESC;
```

Output 18)

Row	Country	NOC	silver_bronze_me...
1	USA	USA	2999
2	Russia	URS	1421
3	Germany	GER	1420
4	UK	GBR	1390
5	France	FRA	1276
6	Italy	ITA	1062
7	Sweden	SWE	1057
8	Australia	AUS	972
9	Canada	CAN	889
10	Russia	RUS	775

Que 19) In which Sport/event, India has won highest medals

Query 19)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans19`
```

```
(
  Sport STRING,
  total_medals INT64,
  Year INT64
)
AS
SELECT
  a.Sport,
  COUNT(*) AS total_medals,
  MAX(a.Year) AS Year
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
WHERE
  a.NOC = 'IND'
  AND a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  a.Sport
ORDER BY
  total_medals DESC
LIMIT 1;
```

Output 19)

Row	Sport	total_medals	Year
1	Hockey	173	1980

Que 20) Break down all olympic games where india won medal for Hockey and how many medals in each olympic games

Query 20)

```
CREATE OR REPLACE TABLE `regal-stage-469404-d2.answer.ans20`
(
  Games STRING,
  Year INT64,
  total_medals INT64
)
PARTITION BY RANGE_BUCKET(Year, GENERATE_ARRAY(1896, 2024, 4))
CLUSTER BY Games
```

```
AS
SELECT
  a.Games,
  a.Year,
  COUNT(*) AS total_medals
FROM
  `regal-stage-469404-d2.midterm.athletes_clean` a
WHERE
  a.NOC = 'IND'
  AND a.Sport = 'Hockey'
  AND a.Medal IN ('Gold', 'Silver', 'Bronze')
GROUP BY
  a.Games, a.Year
```

Output 20)

Row	Games	Year	total_medals
1	1960 Summer	1960	13
2	1972 Summer	1972	14
3	1964 Summer	1964	15
4	1968 Summer	1968	16
5	1932 Summer	1932	15
6	1980 Summer	1980	16
7	1948 Summer	1948	20
8	1952 Summer	1952	14
9	1956 Summer	1956	17
10	1928 Summer	1928	14

Saving output tables to GCS Bucket

```
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans14  gs://answeranuvrat/ans14
Waiting on bqjob_r3fdfde6ba3c909e4_0000019956d14746_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans15  gs://answeranuvrat/ans15
Waiting on bqjob_r14c8833a532b9c1f_0000019956d3d4c2_1 ... (2s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans16  gs://answeranuvrat/ans16
Waiting on bqjob_r6de02233f1e5d50c_0000019956d40f00_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans17  gs://answeranuvrat/ans17
Waiting on bqjob_r6e526f2d407e21ec_0000019956d472c6_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans18  gs://answeranuvrat/ans18
Waiting on bqjob_r56f72117dbc9af0_0000019956d4faae_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans19  gs://answeranuvrat/ans19
Waiting on bqjob_r54e008fe842b98e4_0000019956d544ed_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$ bq extract --destination_format=CSV answer.ans20  gs://answeranuvrat/ans20
Waiting on bqjob_r607b0419f4452f23_0000019956d57cc8_1 ... (1s) Current status: DONE
anuvratverma@cloudshell:~ (regal-stage-469404-d2)$
```

Answers Stored in GCS Bucket