

Chapter 4: Complex Calculations

Exercise 2: Average total country medals by region

Layered calculations are when you create a basic query with an aggregation, then reference that query as a subquery to run an additional calculation. This approach allows you to run aggregations on aggregations, such as a MAX() of a COUNT() or an AVG() of a SUM().

In this exercise, your task is to pull the average total_golds for all countries within each region. This report will apply only for summer events.

To avoid having to deal with null handling, we have created a summer_games_clean table. Please use this when building the report.

Instructions for the exercises:

- Query 1:
 - Set up a query that pulls total_golds by region and country_id from the summer_games_clean and countries tables.
 - GROUP BY the unaggregated fields.
- Query 2:
 - Alias your query as subquery and add a layer that pulls region and avg_total_golds that outputs the average gold medal count for all countries in the region.
 - Order by avg_total_golds in descending order.

Sample database:

query result	summer_games_clean	countries					
sport	event	year	athlete_id	country_id	bronze	silver	gold
Gymnastics	Gymnastics Men's Individual All-Around	2016-01-01	51	173	0	0	0
Gymnastics	Gymnastics Men's Floor Exercise	2016-01-01	51	173	0	0	0
Gymnastics	Gymnastics Men's Parallel Bars	2016-01-01	51	173	0	0	0
Gymnastics	Gymnastics Men's Horizontal Bar	2016-01-01	51	173	0	0	0
Gymnastics	Gymnastics Men's Rings	2016-01-01	51	173	0	0	0
Showing 100 out of 4937 rows							

query result	summer_games	countries	country_stats	athletes	
id	country	region			
1	AFG - Afghanistan	ASIA (EX. NEAR EAST)			
2	ALB - Albania	EASTERN EUROPE			
3	ALG - Algeria	NORTHERN AFRICA			
4	A.Sa - American Samoa	OCEANIA			
5	AND - Andorra	WESTERN EUROPE			
6	ANG - Angola	SUB-SAHARAN AFRICA			
Showing 100 out of 203 rows					

Query given as the solution:

Query 1:

```
-- Query total_golds by region and country_id
SELECT
    region,
    country_id,
    SUM(gold) AS total_golds
FROM summer_games_clean
JOIN countries
ON summer_games_clean.country_id = countries.id
GROUP BY region, country_id;
```

Query 2:

```
-- Pull in avg_total_golds by region
```

```
SELECT
    region,
    AVG(total_golds) AS avg_total_golds
FROM
    (SELECT
        region,
        country_id,
        SUM(gold) AS total_golds
    FROM summer_games_clean AS s
    JOIN countries AS c
    ON s.country_id = c.id
    -- Alias the subquery
    GROUP BY region, country_id) AS subquery
GROUP BY region
-- Order by avg golds in descending order
ORDER BY avg_total_golds DESC;
```