The choosen domain of interest is the sports one, we took the point of view of a company that wants to invest in sports-related products using the results of the last 3 summer olympic games to get informations from the business domain (we choose to consider only summer olympic games because they are more popular than winter olympic games).

1)Operational data sources inspection and profiling

According to the initial purpose, our business questions are the following ones:

- We want to conduce a popularity analysis in order to understand what were the most discussed disciplines during the summer olympic games of the last decade, this will be quite useful because we have to know what sport material we have to invest in.
- Our aim is to identify also in which state our campaign will produce better results, so we need to know the most discussed states and, for each of these states, if there are particularly discussed athletes belonging to that state we can use as testimonials for our products.
- Once we have choosen our candidate testimonials, we have to evaluate the reasons behind their popularity, for this purpose it's very important to know if the athlete is popular in its own or if it's fame is due to the results obtained during the competitions; we want to conduce this analysis because, if we have to differentiate between the two cases, we will be able to distinguish between always popular athletes (mainly good performing athletes whose popularity is due to their results, they will be useful if we want to conduce a long-term campaign spreaded over the years) and athletes who are the “hot topic” of the moment (that may be useful for a shorder but potentially bigger amount of sales in the short term).
- On the other hand, we also have to keep in consideration the “quality” of our candidate testimonials' audience, so we have to compare the number of mentions they received during the olympic games period with the number of followers they have on their socials, in order to identify the percentage of active interested followers they have and deduce if there's a relation between followers and mentions.
- Finally, we will conduce some analysis considering the temporal aspect, for each athlete we will disocver if they took part at previous editions of the games so, if their popularity is due to their recent results, we will be able to make assumptions about how long their trail of popularity will last (athletes whose popularity is due to their results may produce better effects as testimonials if they are continuing to perform well across several editions of the games, so we can prefer them as testimonials instead of athletes who performed well only in the last edition of the games).
  We also need to do temporal analysis at coarser levels (per state, per discipline).

In order to answer our business questions, we started gathering data about participants, results and popularity metrics (such as number of followers and mentions) of the last 3 summer olympic games, the datasets we found are the following ones:

- **2012 London Olympic games' data**: informations about the participants can be found in this article (the dataset we used is linked at the end of the article) , we also needed to know about medals won by each athlete and we found that information on the ESPN official website ;

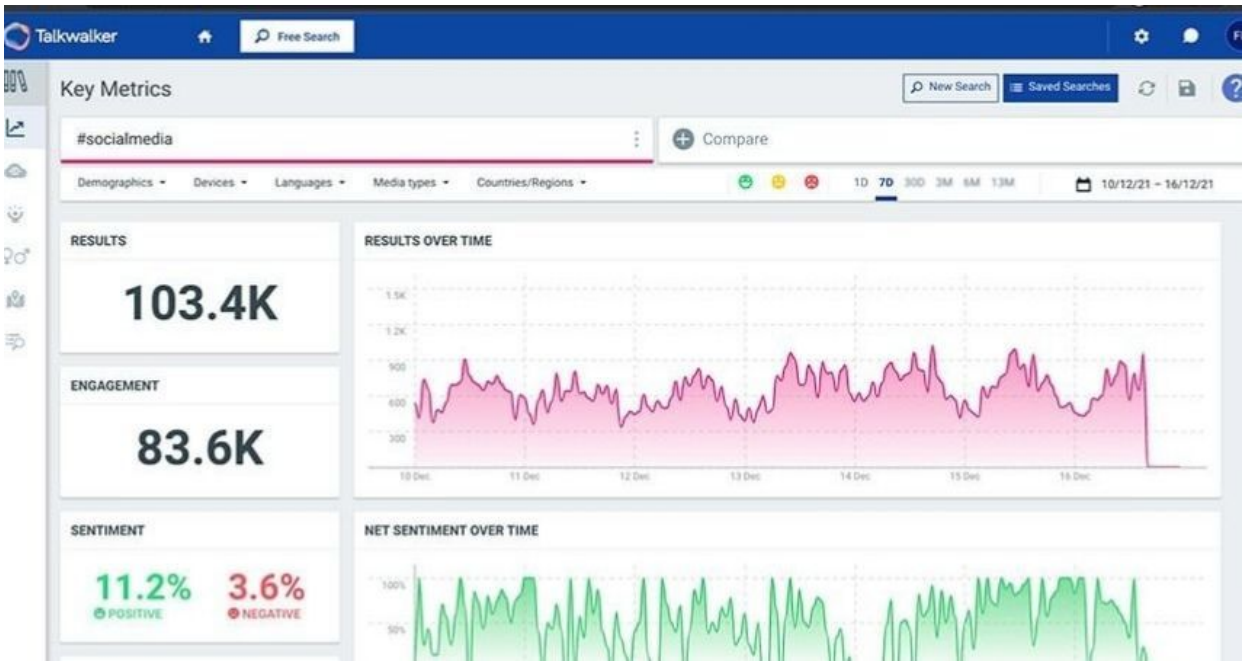
- **2016 Rio Olympic games' data**: the table athletes.csv of this kaggle dataset contains all the informations we need about 2016's participants and medals;

- **2020 Tokyo Olympic games' data**: the table athletes.csv of this kaggle dataset contains the informations about 2020's participants, the dataset also has a table medals.csv containing all the informations about medals won by each athlete but we found another one containing better summarized data about medals;

- For each athlete, whe only need to know the games they participated to, the state they represents, the discipline they practices and the medals they won (we also need to know something about the popularity metrics but we will discuss them in the following point), some of the datsets we have found contains also more data (e.g. height, weight and so on), we cut out those columns from our downloaded csv files because we don't need them for our analysis: we used OpenOffice Calc to open the csv files as excel tables, then we deleted the columns we didn't need;
  We also had to modify a column of the 2012's athletes' dataset because the other 2 athletes' datasets refer to nations using their country code (e.g ITA for Italy and son on) but this one named them using only the full names of the states, so we had to perform a multiple find and replace using the official country codes table to discover all the country codes we need; we also had to rely on find and replace for all of the “special characters” that may be included in athletes' names (for example, substituting À with A, deleting the usages of the ' and “ characters and so on), we also turned all the character in lower case in order to partially avoid name conflicts between different data source.
  Starting from the above datasets, we ended up with 3 csv files for the athletes' informations (one for each year) and 3 csv files for medals' informations (as before, one for each year).
  [these 6 csv files can be found in the directory “1_starting_files”]
  NB: unfortunately datasets of the real world are not perfect as we want, we were able to resolve the most critical conflicts but minor ones still remains and finding all of them requires a row by row analysis we weren't able to perform efficiently so we had to accept to work with slightly noisy data.

- The last piece of information we need is the one related to the popularity metrics, we have to know for each athlete the number of followers they have during the year they took part to the games and how many times they were mentioned on the social networks during the games' period.
  Unofrtunately, we couldn't find any already filled dataset containing the data we were looking for but we discovered some on-line services that can be used for

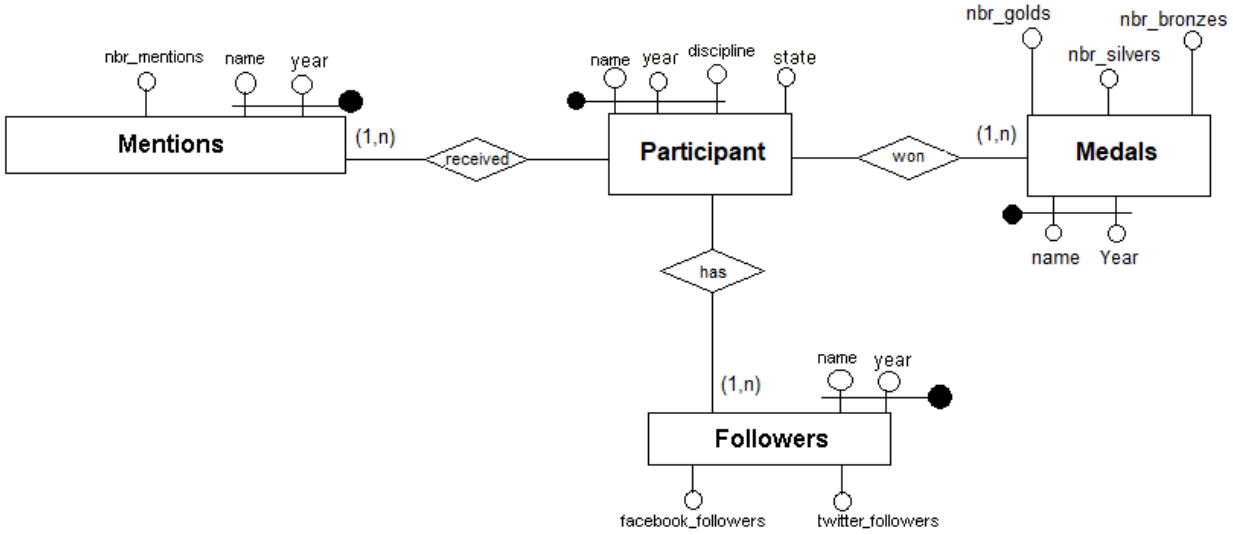
extracting the data we need from the web, one of these services is Talkwalker.



Talkwalker is a powerful on-line tool that allows to do hashtags and keywords analysis, among its several functionalities there is the possibility to count the number of times a certain hashtag was used, it's also possible to set a time-span specifying the time period to consider for the analisys;
Our aim was to use it to get, for each athlete, the number of times the hashtag with their name was used during the period of the games (and repeat it for each of the three editions of the games we are considering, using also some scripts in order to avoid manual research for all the names), unfortunately all the services like this offer only a short free demo (the full version that allows unlimited usage of the tool is reserved to real companies and it has a cost in money) so we coulnd't get the data we need but we were able to observe some real samples we could use to generate real-like data;
For the followers' count, we got the same problem so we had to generate data about it too; our aim is to show a methodology that would produce very interesting and relevant results in presence of the right data, also the most of our datasets are made up of real data and we resolved almost all the conflicts so our results will be realistic too even if they won't be 100% accurate respect to what happened in the real world.
The logical schema for the 4 tables is:

-participants(name,year,discipline,state)
-medals(name,year,golds,silvers,bronzes)
-mentions(name,year,total_mentions)
-followers(name,year,facebook_followers,twitter_followers)

[sql scripts for tables creation can be found in the directory “2_db_creation”]
NB: the 6 csv files in “1_starting_files” don't have the column “year” since it wasn't included in the original datasets, we had to add it during the mapping from csv to sql inserts (see later)
The relation between the above tables is expressed by the following ER schema:



a few notes about the schema:

-discipline has to be a part of participant's key because we discovered that the same athlete during the same year can participate multiple times for multiple disciplines (but only one time for each discipline);

-the omitted cardinalities are intended to be (1,1);

-the cardinality of mentions and followers is (1,n) because, if the same athlete participated during the same year for multiple disciplines, then the tuple with followers and mentions with that name and year as key will be associated with more than one tuple, that's the same for the medals associated with that name and year because each tuple of the medals table is intended to be a sum-up of all the medals won by that athlete during that year;

Information about athletes and medals can be found in the tables participants and medals we created; in order to perform the inserts and add the missing “year” column, we used a python script for each table for extracting data from csv files and reorganize them in a sql file (in this case, participants.py and medals.py).

[these two python scripts and the generated sql files with the inserts can be found in the directory “3_insert_scripts”]

After doing this, we used the generated participants.sql and medals.sql files to fill the tables we created on the DB, then we had to generate the number of followers and mentions for each athlete during each year they participated.

We used the script generator.py to randomize realistic data that we put in our followers and mentions tables, this script needs a list of pairs {name,year} and for each name and year generates a corresponding insert in both the tables.

After running the script, followers.sql and mentions.sql files are generated and ready to be used.

[the generator script, the list of pairs used by the script and the generated sql files with the inserts can be found in the directory “4_missing_data_generation”]

Now we have done with data cleaning and our source data was mapped into an appropriate integrated data source.

2)Data Warehouse conceptual design

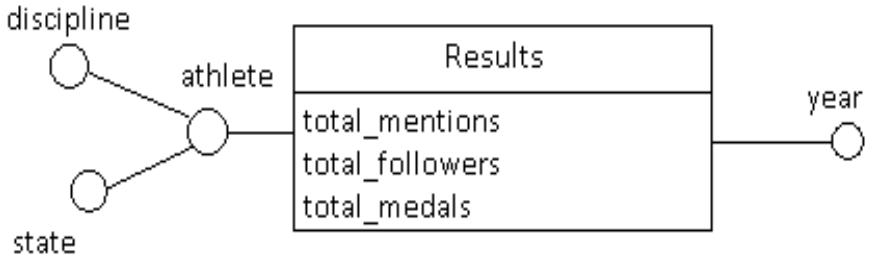
In this section we will discuss the transition from operational DB to DW.
We started identifying the measures we were interested in, according to our business questions; we divided the measures in two main types:

- **performance measures**: relative to the results achieved during the competition, we identified the number of gold,siver and bronze medals as performance measures;
- **popularity measures**: relative to the “share” obtained during the games, we identified the total number of followers and mentions on the socials as popularity measures;

Then, we had to decide the dimensions we would use to aggregate those measures:

- **temporal dimension**: degenerate dimension limited to the “year” aggregation level since we didn't need a finer one;
- **geographical dimension**: since we mentioned disciplines and states in our business questions as well as we mentioned athletes, we had to derive significative outcomes for them too, so we put athletes,states and discipline together in the same dimension (note that the dimensional attributes in this hierarchy are partially ordered since both discipline and state can be used to aggregate measures at a coarser level than athlete, but none of the two can be compared to the other)

According to what we stated above, our proposed DFM is the following:



NB:with “athlete” we mean the “name” attribute in our starting tables, also, there is no dinamicity in dimensions (once a medal in a certain discipline is registered as won by an athlete, no one else can win that medal during the same year).

data volume of the different aggregation patterns:

-{athlete,year}:	32510 records	-{athlete}:	29420 records
-{state,year}:	748 records	-{state}:	294 records
-{discipline,year}:	102 records	-{discipline}:	39 records
-{year}:	3 records	-{ }:	1 record

about the workload:

- our queries often use aggregation patterns including the “year” attribute.
- we never use the all query and the {year} aggregation pattern.
- we include the “athlete” dimensional attribute in the aggregation patterns used by our queries as much as “state” and “discipline”.

3)DataWarehouse ROLAP logical design

Starting from the DFM created in the previous section, we now have to derive a ROLAP logical schema from it;

We decided to put “year” into the FT since it's the only degenerate dimension, it isn't worth to create a junk DT for it even if the cardinality of the “year” dimension is limited to only three values because:

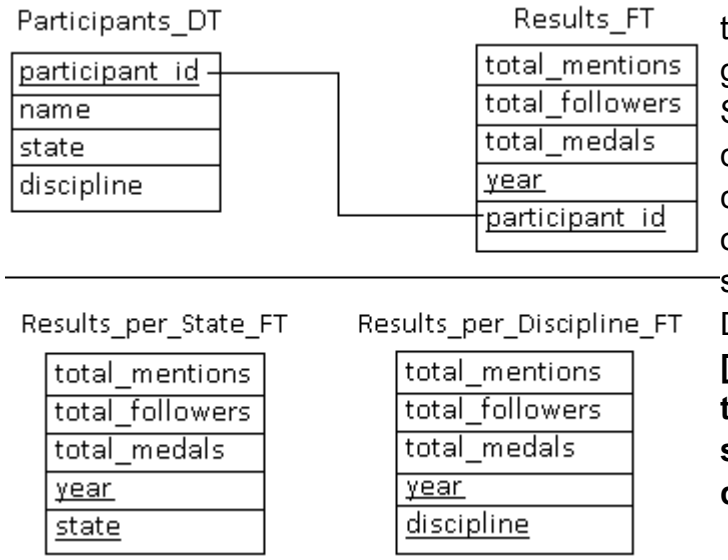
- doing it will avoid redundancy of the “year” attribute but we won't save space since we need surrogate keys (e.g. junkID) in order to link the junkDT with the FT.
- one more DT means one more join to do if we include “year” in the aggregation pattern of our queries, and we have many queries that need it.

For the geographical dimension, the cardinality gap between “athlete” and the other dimensional attributes is quite high so we had to trade between efficiency and space saving because:

- if we put all of the attributes in a single DT we will optimize joins between the DT and the FT but we will have a lot of redundancy for states and disciplines' names.
- If we put all the athletes in a primary DT and the other dimensional attributes in a secondary DT linked to it, we will save space due to the cardinality gap but the computational cost of queries involving attributes in the secondary DT will raise.

Considering both the cases, we have choosen to rely on the first possibility: we make use of “discipline” and “state” dimensional attributes quite often so we may prefer to achieve a better level of performance when accessing those data more than save space; also, space saving is not even crucial considering how “small” our DW is (big DWs consist in millions of records while ours has not more than 35.000 records).

Now we have done with the decisions about primary FT and its related DT; having also to include secondary events, we decided to design our DW as multiple independent star schemas with two secondary FTs for the {state,year} and {discipline,year} grouping sets/aggregation patterns. Secondary FTs have only two degenerate dimensions each, we have choosen to include them in the FTs in order to avoid joins between the secondary FTs and their hypotetical DTs.



[the sql scripts for the definition of the multiple independent star schemas is contained in the directory “5_dw_and_queries”]

4-5)OLAP queries (+ SparkSQL optional part)

We used several queries in order to compare multiple measures at different aggregation levels and extract the outcomes we were interested in, the sql file contains all the queries we have performed but in this report we will discuss in detail only 5 of them (one for each requested category of the assignment).
We have classified the queries we have perfomed in 2 main groups:

- queries we used for an overall analysis of the data in order to obtain something we could use to produce the graphical representation of our outcomes (discussed in the Tableau section of the documentation, seen later).
- queries we used for an in-depht analysis, consisting in directly comparing different intances in case of doubts among what state or what testimonials choose (discussed in this section of the documentation).

[the sql script with all the queries is contained in the directory “5_dw_and_queries”]

For these 5 queries, we also did a corresponding PySpark SQL script that resolves them using advanced DataFrames concepts in a large scale data processing environment.
[the pyspark scripts are contained in the directory “extra_part_spark_sql”]

What follows is a short explanation of the 5 queries with discussion of the results, results will be shown both on postgresql and on the cluster.

Query 1: Show the total number of mentions received by athletes divided by state and discipline and the total number of mentions for each state, for the year 2020.

Data Output | Explain | Messages | Notifications

	state text	discipline text	sum real
1	AFG	athletics	46729
2	AFG	shooting	23252
3	AFG	swimming	26558
4	AFG	taekwondo	27393
5	AFG	[null]	123932
6	ALB	athletics	15256
7	ALB	gymnastics	22659
8	ALB	judo	15379
9	ALB	shooting	18813
10	ALB	swimming	33082
11	ALB	weightlifting	55689
12	ALB	[null]	160878

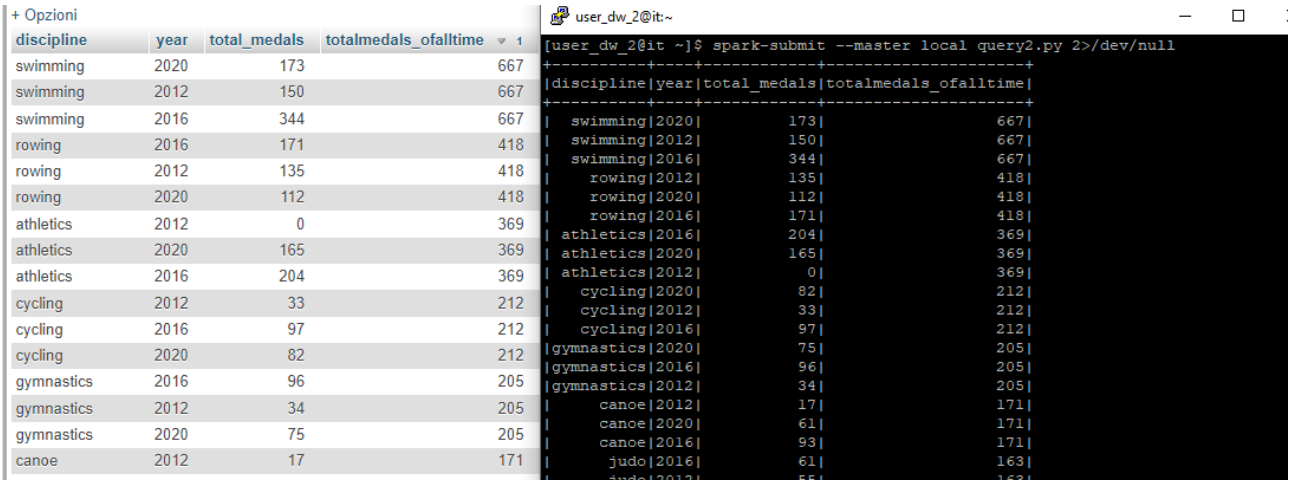
user_dw_2@it:~

```
[user_dw_2@it ~]$ spark-submit --master yarn query1.py 2>/dev/null
+-----+-------------+--------------------+
|state|   discipline|sum(total_mentions)|
+-----+-------------+--------------------+
| null|         null|           238906851|
|  AFG|         null|              123932|
|  AFG|    athletics|               46729|
|  AFG|     shooting|               23252|
|  AFG|     swimming|               26558|
|  AFG|    taekwondo|               27393|
|  ALB|         null|              160878|
|  ALB|    athletics|               15256|
|  ALB|   gymnastics|               22659|
|  ALB|         judo|               15379|
|  ALB|     shooting|               18813|
|  ALB|     swimming|               33082|
|  ALB|weightlifting|               55689|
|  ALG|         null|              839830|
|  ALG|    athletics|              108960|
|  ALG|       boxing|              186364|
```

The resulting table can be used to compare the popularity of a certain discipline in a certain state with the popularity of states at the same time, for example in this extract of the result table we can observe that “athletics” is more popular in Afghanistan than in Albania but Albania has a bigger general audience than Afghanistan.
So, if for example we decide to invest in that discipline we should do it in the state with the highest number of mentions for that discipline but if we want to invest in several different disciplines we should prefer the state with the biggest general audience.
For this query, we are considering only mentions received during the last edition of the games so these results are useful in a short term campaign in which we want to rely on the

popularity of athletes, discipline and states that are the “hot topic” of the moment.

Query 2: For each discipline and year, show the number of medals assigned to that discipline during that year and compare it with the total number of medals won by athletes who practice to that discipline across the three editions of the games.



+ Opzioni

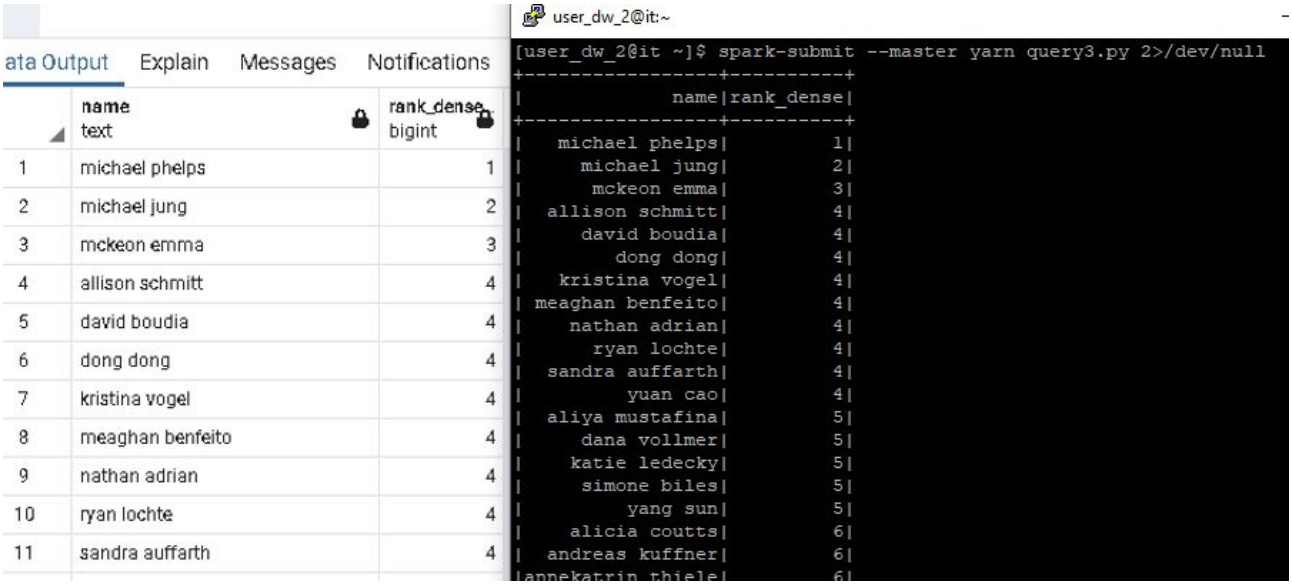
discipline	year	total_medals	totalmedals_ofalltime
swimming	2020	173	667
swimming	2012	150	667
swimming	2016	344	667
rowing	2016	171	418
rowing	2012	135	418
rowing	2020	112	418
athletics	2012	0	369
athletics	2020	165	369
athletics	2016	204	369
cycling	2012	33	212
cycling	2016	97	212
cycling	2020	82	212
gymnastics	2016	96	205
gymnastics	2012	34	205
gymnastics	2020	75	205
canoe	2012	17	171

NB:In the screenshot on the left, phpmyadmin is used instead of postgresql, since we worked on different devices, we tried different DB-management app too.

The resulting table can be quite useful if we have to know if a certain discipline is popular on his own or because of the performance metric too.

Each discipline is divided in sub-categories and each of them has its own competition, so the disciplines with an higher number of sub categories will have an higher total number of medals assigned, this is like a sort of “cheat” we used for inferring which disciplines have the highest number of sub-categories since they are often more popular than ones that have a few categories.

Query 3: Create a ranking of athletes based on the medals they won across the three editions of the games.



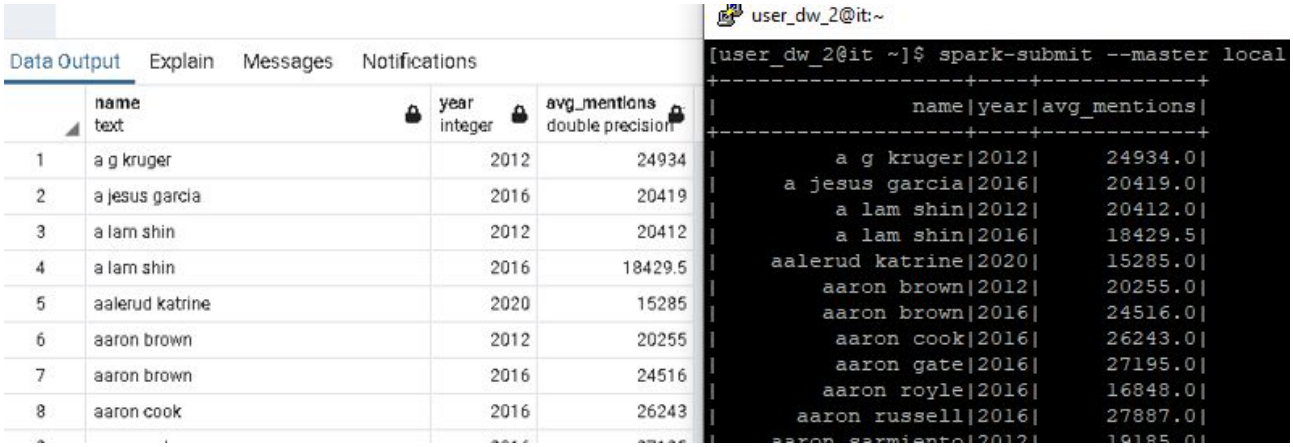
ata Output | Explain | Messages | Notifications

	name text	rank_dense bigint
1	michael phelps	1
2	michael jung	2
3	mckeon emma	3
4	allison schmitt	4
5	david boudia	4
6	dong dong	4
7	kristina vogel	4
8	meaghan benfeito	4
9	nathan adrian	4
10	ryan lochte	4
11	sandra auffarth	4

Particularly useful if we have to choose an athlete as a testimonial for a long term

campaign since this ranking considers results of all the three editions so we suppose athletes on the top of this table would perform well for at least two other editions.

Query 4: Compute the average number of mentions received by athletes across the three editions of the games.



Data Output | Explain | Messages | Notifications

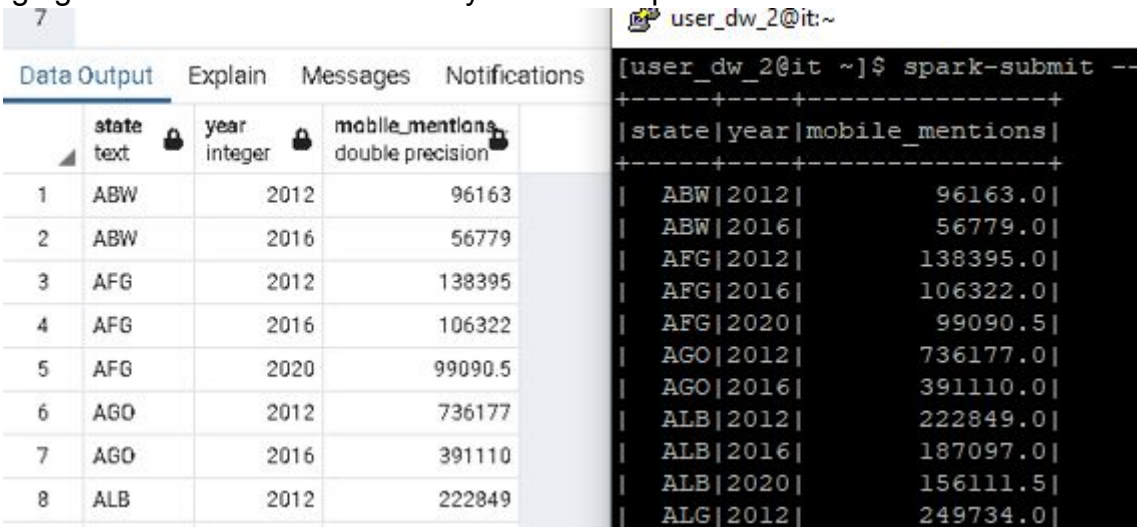
	name text	year integer	avg_mentions double precision
1	a g kruger	2012	24934
2	a jesus garcia	2016	20419
3	a lam shin	2012	20412
4	a lam shin	2016	18429.5
5	aalerud katrine	2020	15285
6	aaron brown	2012	20255
7	aaron brown	2016	24516
8	aaron cook	2016	26243

user_dw_2@it:~

```
[user_dw_2@it ~]$ spark-submit --master local
+--------------------+----+------------+
|                name|year|avg_mentions|
+--------------------+----+------------+
|         a g kruger|2012|     24934.0|
|     a jesus garcia|2016|     20419.0|
|        a lam shin|2012|     20412.0|
|        a lam shin|2016|     18429.5|
|    aalerud katrine|2020|     15285.0|
|       aaron brown|2012|     20255.0|
|       aaron brown|2016|     24516.0|
|        aaron cook|2016|     26243.0|
|        aaron gate|2016|     27195.0|
|       aaron royle|2016|     16848.0|
|      aaron russell|2016|     27887.0|
```

We stated that the number of mentions received by an athlete is a “volatile” metric, since it's not guaranteed that mentions will continue to grow over the years for that athlete, from this extract of the results we can observe two different cases with two different outcomes: the athlete “a lam shin” had a popularity decrease from 2012 to 2016 while the athlete “aaron brown” increased his average number of mentions from 2012 to 2016, so if we choose to conduce a long term campaign and we are in doubt between these two athletes, this query helps us deciding to consider “aaron brown” more than “a lan shin”; this query is intended to be used for in-depht analysis, so in case of doubts between two or more athletes we will run this query adding a “where name in (<list of names>)” clause.

Query 5: For each state, compute the average number of mentions received by athletes belonging to that state between each year and the previous one.



7

Data Output | Explain | Messages | Notifications

	state text	year integer	mobile_mentions double precision
1	ABW	2012	96163
2	ABW	2016	56779
3	AFG	2012	138395
4	AFG	2016	106322
5	AFG	2020	99090.5
6	AGO	2012	736177
7	AGO	2016	391110
8	ALB	2012	222849

user_dw_2@it:~

```
[user_dw_2@it ~]$ spark-submit --
+-----+----+---------------+
|state|year|mobile_mentions|
+-----+----+---------------+
|  ABW|2012|        96163.0|
|  ABW|2016|        56779.0|
|  AFG|2012|       138395.0|
|  AFG|2016|       106322.0|
|  AFG|2020|        99090.5|
|  AGO|2012|       736177.0|
|  AGO|2016|       391110.0|
|  ALB|2012|       222849.0|
|  ALB|2016|       187097.0|
|  ALB|2020|       156111.5|
|  ALG|2012|       249734.0|
```

This last query is particularly important for confirming that the number of mentions is often a volatile metric, even if we continue discending the table we can observe that almost all of the states have a “peak moment” in which they are at the maximum of their popularity and then we can note a gradual decrease of the measure.

In conclusion, if we have to compare Standard SQL execution with SparkSQL execution, we can say:

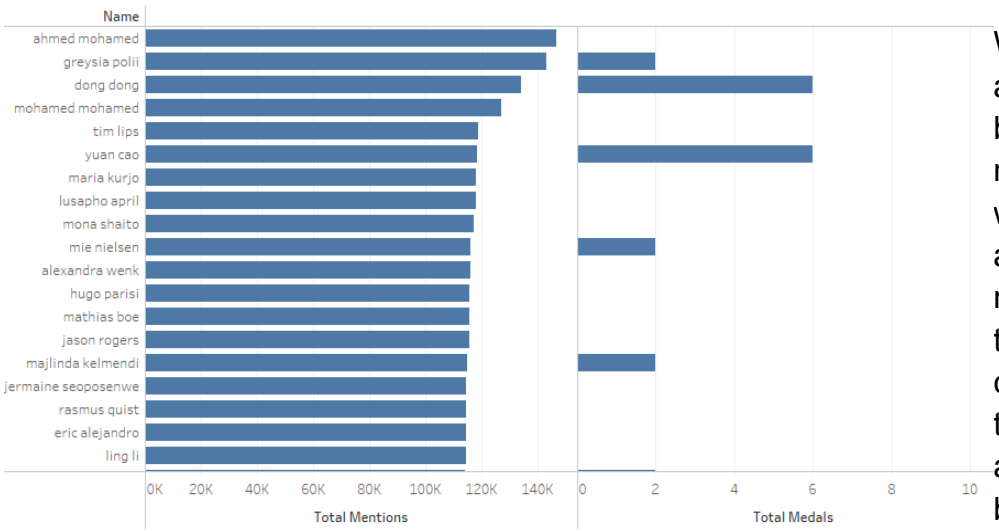
- Standard SQL allow us to query our data using a simple and intuitive syntax but with big tables the computational cost can rise a lot.
- SparkSQL can reduce the computational time by partitioning the data and parallelizing the query execution among all the partitions but the syntax is more complex.

If we want to have the benefit of the SparkSQL large scale data processing environment without having to learn a more complex syntax we can rely on HiveSpark: Hive stores tables in a distribuited file system (es: HDFS) allowing the user to query it providing a console that accepts commands in Standard SQL form.

6)Tableau

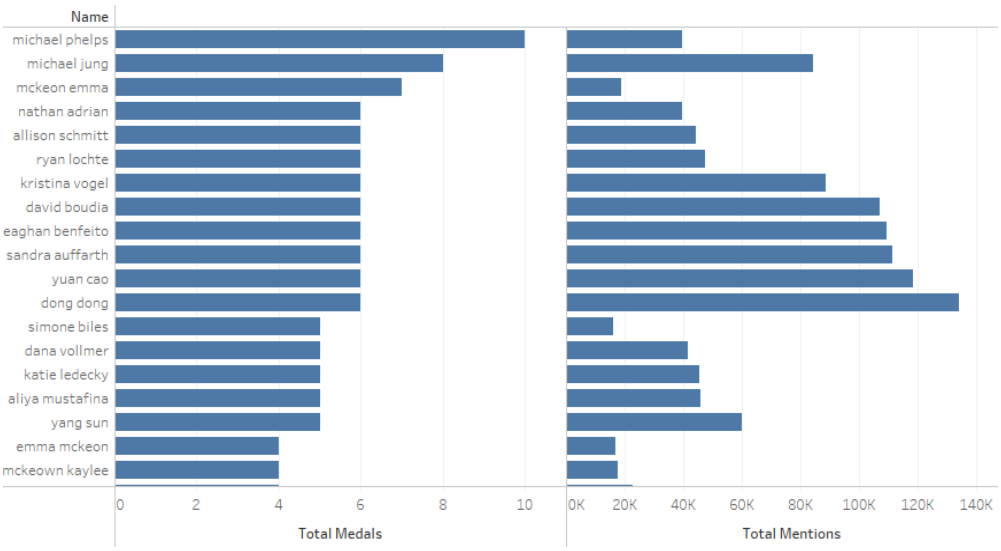
In the last part of this project we will make use of Tableau in order to obtain a graphical representation of our data, what follows is a short explanation of some of the outcomes we wanted to discuss embedded with the corresponding diagram/graph produced by Tableau (we will discuss almost all of the outcomes, the missing ones are in the presentation) **[the tables we have extracted from our DW and imported in Tableau are contained in the directory "6_tableau"]**

Important outcomes – performance vs popularity

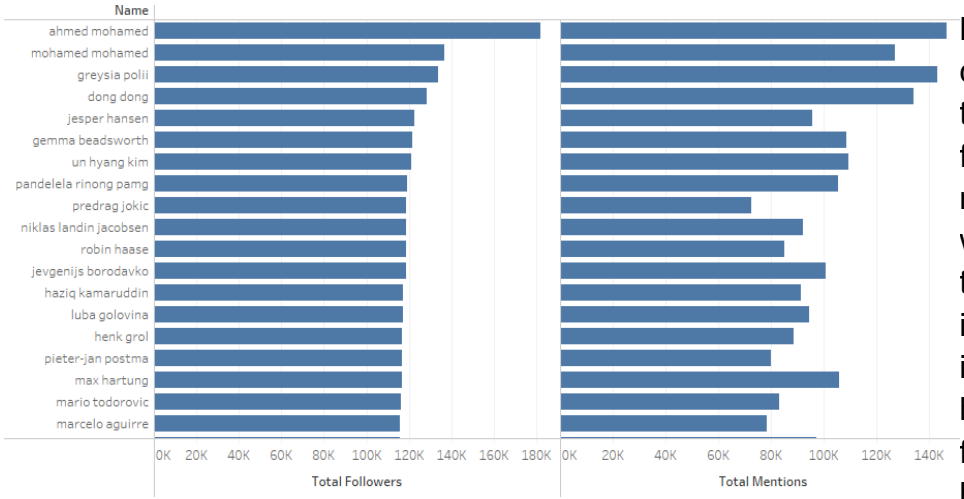


We started by doing a comparison between the total number of medals won by each athlete and the total number of mentions they received, we can observe that the two measures are not related because the most mentioned athlete is clearly far away to be the best performing one, this confirms also that the popularity factor is very volatile at the level of the single athlete because popularity often comes out of nowere and we can't predict how long it will last, same is for states. The situation is easier if we have to measure popularity at the discipline aggregation pattern, popularity related to disciplines is more long lasting and it could be taken in consideration if we have to choose in what material invest.

We took also in consideration the opposite case in order to demonstrate that the best performing athletes are not necessarily the most popular ones.

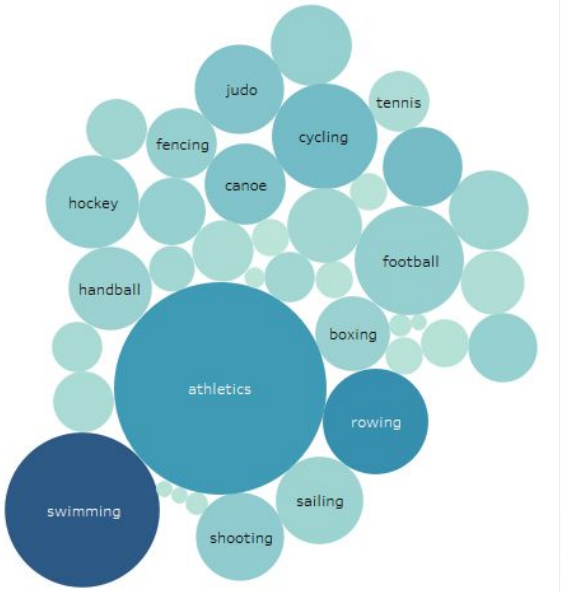


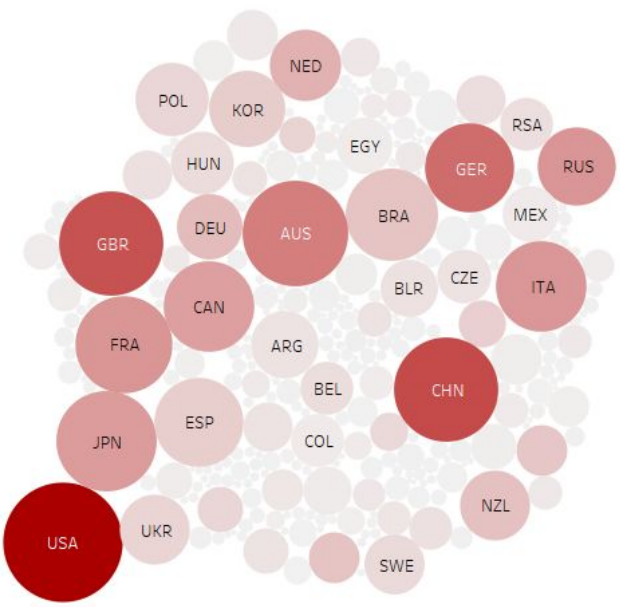
Important outcomes – audience vs mentions



By making a comparison between the number of followers and the number of mentions we can observe that the two measures are in a sort of relation, indeed athletes with high number of followers received an high average number of mentions too.

Important outcomes – relation among measures and aggregation pattern



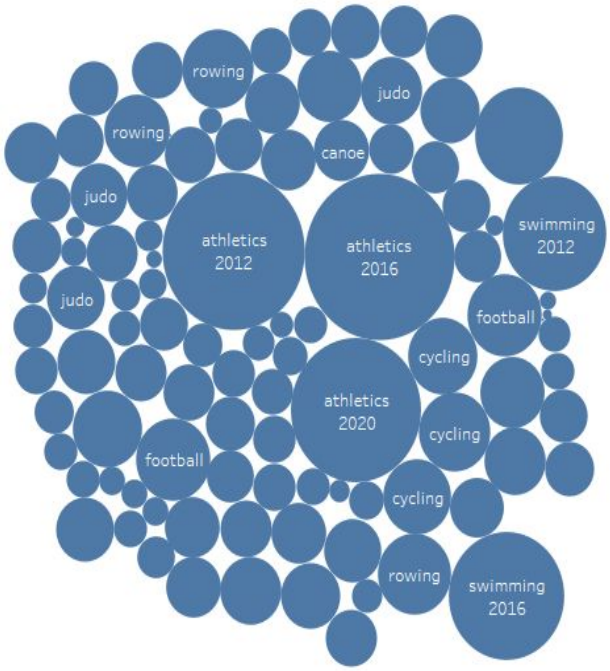


Legend of the two graphics above:

- The bigger the circle is, the more mentions athletes belonging to that discipline/state have received in total.
- The darker the color of the circle is, the more medals were won by athletes belonging to that discipline/state in total.

Whe saw that popularity and performance are not related at the “athlete” level, but we can't say the same for coarser aggregation patterns such as “state” and “discipline”.
The two graphics above show that, considering dimensional attributes at an higher level in the geographical dimension, there is a clear relation between the number of mentions and the number of medals won.

Important outcomes – time coherence



Legend: the bigger the circle is, the more mentions athletes belonging to that discipline have received during that year in total.

With this last graphic we can observe that the most popular disciplines maintain their popularity among the years, so we can assume that the popularity measures are not so volatile anymore when refereed to the discipline aggregation pattern.

Final considerations:

- popularity measures are volatile across the years if referred to single athletes or states but they are not so volatile anymore when we consider the discipline aggregation pattern;
- athletes with a good audience received also a good number of mentions, since mentions for athletes are volatile we can deduce also the number of their followers is, and this is coherent since followers are a popularity measure too.
- popularity measures are not related with performance if referred to single athletes, but when we consider coarser aggregation patterns the situation changes;