UNIVERSITÀ DI PISA

DATA SCIENCE & BUSINESS INFORMATICS

DIPARTIMENTO DI INFORMATICA

Laboratory of Data Science

Progetto Gruppo 14

Studenti

Carlo PALADINO 537650

Francesco SALERNO 534622

Alessandro BONINI 604482

Part 1

Assignment 0

The database schema was developed on SQL Server Management Studio. As regards the types of the different attributes, the following choices were made:

- for the attributes of the string type, char (100) was used
- for the attributes of the numeric type, int or float was used, depending on the eventuality.

The primary and foreign keys have been defined as suggested by the scheme provided, in particular the connection between thetable *Player* and *Match* occurred with two different relationships: between the primary key *id_player* and the external one *winner_id* and again, between *id_playe*reloser_id.

Assignment 1

This assignment can be divided into two phases carried out in sequence, in the first, starting from the four csv files (*tennis.csv, male_players.csv, female_players.csv and country.csv*) we have prepared five new files containing the data organized in to be able to fill the tables previously created in the database.

- *Tournament.csv*: You have selected the attributes of interest from tennis.csv
- *Date.csv*: Using the attribute *tourney_date* from tennis.csv and creating the attributes *year, month, day and quarter.*
- *Match.csv*: the attribute was created *match_id* with the use of *tourney_id* and *match_num* and the other attributes of interest were taken from tennis.csv
- *Geography.csv*: created with the use of the csv file made available on didawiki for the language and geography.csv file.have been selected *Country_ioc, country, continen*te *language* as attributes and a slight data cleaning has been done.
- *Player.csv*: file created with the use of male_players.csv and female_players.csv to obtain theattribute *sex* and the attributes of interest are selected using a dictionary, furthermore, here too a first cleaning of the data has been made.

The second part was dedicated to cleaning data, for this purpose we used the pandas library. For each file described above, aversion was created *_cleaned* in which missing values, duplicates, outliers and various inconsistencies were managed. At the end of this process we got five csv files corresponding to the tables created with SQL Server Management Studio.

Assignment 2

To populate the database, a program has been written for each table (uploadCSV_ <tablename> .py), each of which has the same structure:

• reading the csv file, using the read_csv method;

• opening of the connection to the “database'Group_14_DB” and creation of the cursor;

• iteration on the lines of the file;

• writing the query in parametric form;

• file closing, cursor and connection

Part 2

Assignment 0

For every tournament, the players ordered by number of matches won.

The first node inserted into the data flow task is an OLE DB source that is required to access the Match fact table. Subsequently a group by was carried out on winner_id and tourney_id and a count on match_id, renamed *victories*. Then, the result was sorted by tourney_id with ascending sort type and wins, with descending sort type and output aliases_player *wins*. The result obtained was saved on a file in .txt format.

The first few lines are shown below:

```
tourney_id,winner_id,vittorie_giocatore
2016-0083,200282,7
2016-0083,104327,5
2016-0083,105906,5
2016-0083,105047,3
2016-0083,106072,3
2016-0083,106214,3
```

Assignment 1

A tournament is said to be ”worldwide” if no more than 30% of the participants come from the same continent. List all the worldwide tournaments.

For this assignment, the following steps were carried out:

- Collect all player_id by combining winners and losers by union
- Calculate the total number of players for tournament and continent
- Calculate the total number of players for tournament
- Calculate all tournaments where the maximum percentage of players coming from the same continent is <30%

The process devised on SSIS is therefore composed of four main parts:

- The first part is necessary to collect all the player_id. An OLE DB source node has been inserted to access the Match fact table, then a multicast with double sorting on tourney_id (with ascending sort order) and finally a union with input 1 winner_id and input 2 loser_id.
- To solve the second point, a search is performed with the Player table to obtain the country of each player and then another search to obtain the continent to be associated with each country. The aggregation is then performed by tourney_id and continent and the count on player_id to obtain the total number of players per continent. At this point, having all the information we need, we perform a multicast to carry out different operations.
- To calculate the total number of players per tournament we perform, thanks to the aggregation operation, a group by on tourney_id and a sum on TotalPlayerContinent to obtain *TotalPlayerByTournament* . At this point an ascending sort order is performed on tourney_id for the two parallel flows obtained from multicast and a merge join is performed afterwards.
- Next, we convert *TotalPlayerByContinent* and TotalPlayerByTournament as floating point values and divide them to get the value *Average* (TotalPlayerContinent / TotalPlayerByTournament). Finally, through the aggregation operator we perform a group by for tourney_id and the Maximum operation on Average. This is because, in the conditional subdivision, we will specify the condition "for each tournament, if the highest ratio between the number of players on the same continent and the total number of players, is less than 30%, then add this tournament to the final file ". The result obtained was saved on a file in .txt format.

The resulting lines are shown below:

```
tourney_id,Average
2019-W-ITF-RSA-01A-2019,0.27419356
2018-W-FC-2018-G2-AO-A-M-NZL-LBN-01,0.25
```

Assignment 2

For each country, list all the players that won more matches than the average number of won matches for all players of the same country.

For this assignment, the following steps were carried out:

- Calculate the total wins per player
- Calculate the total wins for each country
- Calculate the total players for each country
- Select all those players for which the total wins are greater than the ratio of total wins by country and total players for that country.

The first node inserted into the data flow task is an OLE DB source that is required to access the Match fact table. The Player table is then searched to get the country column. A multicast is

then applied to create two parallel streams. The left flow continues with an aggregation. Two group by are performed on winner_id and country and a count on match_id, named *PlayerWins*.

The right flow, on the other hand, continues with an aggregation in which a group by on country and a count on match_id, called *Vittorie_Per_country* is performed.. Subsequently, both streams are sorted by country in ascending order for a subsequent merge. First, however, the right stream performs a merge operation once more with the Player table. This table is loaded via OLE DB source, a group by on country and a count on id_player is performed through aggregation to determine the total players for each country. Finally, a sorting on country of increasing type is performed. At this point, a merge join is performed andare selected as columns *Country,* and *Wins_Per_countryPlayerPerCountry*. Then, the derived column is created *AverageWins*, obtained as the ratio between Vittorie_Per_country and PlayerPerCountry. Next, a merge join is performed between that stream and the left stream outgoing from the first multicast. Finally, through a conditional subdivision, all those players for which PlayerWins> AverageWins are saved on a .txt file.

The first few lines are shown below:

```
winner_id,PlayerWins,AverageWins,country
213972,78,10,ALG
214604,93,19,ARG
106057,68,19,ARG
144821,28,19,ARG
104216,55,19,ARG
206345,113,19,ARG
```

Part 3

Assignment 0

Build a datacube from the data of the tables in your database, defining the appropriate hierarchies for time and geography. Use the rank and rank points of the winner and loser as measure.

In this first phase, the olap Group 14 DB_Parte3 cube was built, which has two dimensions: *Tourney* and *Player*. Within Tourney, we find the non-flathierarchy *YearQuarterMonthTourneyId* consisting of Year -> Quarter -> Month -> Tourney_id.

Inside Player, on the other hand, we find the non-flat hierarchy *ContinetCountryIDPlayer* composed of Continent -> Country -> IDPlayer.

Assignment 1

Show the player that lost the most matches for each country

```
with member rnk as
rank(([Loser].[Country].currentmember,[Loser].[ContinetCountryIDPlayer].currentmember),
([Loser].[Country].currentmember,[Loser].[ContinetCountryIDPlayer].[Id Player] ),
[Measures].[n_match])

select [Measures].[n_match] on columns,
nonempty(filter(([Loser].[Country].[Country],[Loser].[ContinetCountryIDPlayer].[Id Player]), rnk = 1)) on rows
from [Group 14 DB_Parte3]
```

To solve the following assignment, a newm easure has been created *rnk,* which represents the sorting by function *rank*, of the number of games lost by each player (loser) for each country. The measure has been inserted on the *n_matchcolumns,* which would be the count of the games, while on the rows, for each row, there is the *country*, the*Id_player* and the *number of games of the player* which, in the ordering, results with a rank equal to 1 , that is, the one who has lost the most games.

The following shows the first few lines:

		n_match
Algeria	Ines Ibbou	63
Andorra	Victoria Jimenez Kasintseva	19
Argentina	Renzo Olivo	145
Armenia	Ani Amiraghyan	31
Australia	Marc Polmans	123
Austria	Sebastian Ofner	125
Bahamas	Kerrie Cartwright	12
Barbados	Darian King	90
Belarus	Uladzimir Ignatik	131
Belgium	Kimmer Coppejans	137

Assignment 2

For each tournament, show the loser with the lowest total loser rank points

```
SELECT [Measures].[Loser Rank Points] ON COLUMNS,
GENERATE(([Tourney].[Tourney Id].[Tourney Id], [Tourney].[Year].[Year]),
BOTTOMCOUNT(([Tourney].[Tourney Id].CURRENTMEMBER, [Tourney].[Year].CURRENTMEMBER,
nonempty([Loser].[Id Player].[Id Player])),
1,
[Measures].[Loser Rank Points])) ON ROWS
FROM [Group 14 DB_Parte3]
```

To solve the following assignment, on the columns has been inserted the measure *Loser Rank*Points,while on the rows, using function*Generate,* for every tournament, for every year, is performed a *BottomCount*. the loser with the lowest Loser Rank Points measure is chosen.

The first few lines are shown below:

			Loser Rank Points
Abu Dhabi	2021	Makoto Ninomiya	20
Acapulco	2016	Renata Zarazua	45
Acapulco	2017	Giuliana Olmos	47
Acapulco	2018	Alan Fernando Rubio Fierros	1
Acapulco	2019	Luis Patino	6
Acapulco	2020	Lucas Gomez	10
Acapulco	2021	Luis Patino	15
Adelaide	2020	Mikalai Haliak	10
Adelaide	2021	Kimberly Birrell	41
Agadir $15K	2017	Linda Puppendahl	3

Assignment 3

For each tournament, show the loser with the highest ratio between his loser rank points and the average winner rank points of that tournament.

```
with member AVG_Winner_rank_points as
(([Tourney].[Tourney Id].CURRENTMEMBER, [Loser].[Id Player].[All]),  [Measures].[Winner Rank Points]) /
(([Tourney].[Tourney Id].CURRENTMEMBER, [Loser].[Id Player].[All]), [Measures].[n_match] )

member Loser_Rank_Points as
([Tourney].[Tourney Id].CURRENTMEMBER, [Measures].[Loser Rank Points])

member Ratio as
Loser_Rank_Points / AVG_Winner_rank_points

select { AVG_Winner_rank_points, Loser_Rank_Points, Ratio}  on columns,
GENERATE([Tourney].[Tourney Id].[Tourney Id],
TOPCOUNT(([Tourney].[Tourney Id].CURRENTMEMBER,[Tourney].[Year].[Year], [Loser].[Id Player].[Id Player]), 1, Ratio )) on rows
from [Group 14 DB_Parte3]
```

To solve the following assignment, new measures have been created: *AVG_winner_rank_points, Loser_Rank_Points* and *Ratio. AVG_winner_rank_points*, defines for each Loser of each tournament the ratio between *Winner_rank_point s*and the *count of Matches* played. *Loser_Rank_Points* instead, it defines the total of each player's Loser Rank Points. Finally, *Ratio* is calculated as Loser_Rank_Points / AVG_winner_rank_points.

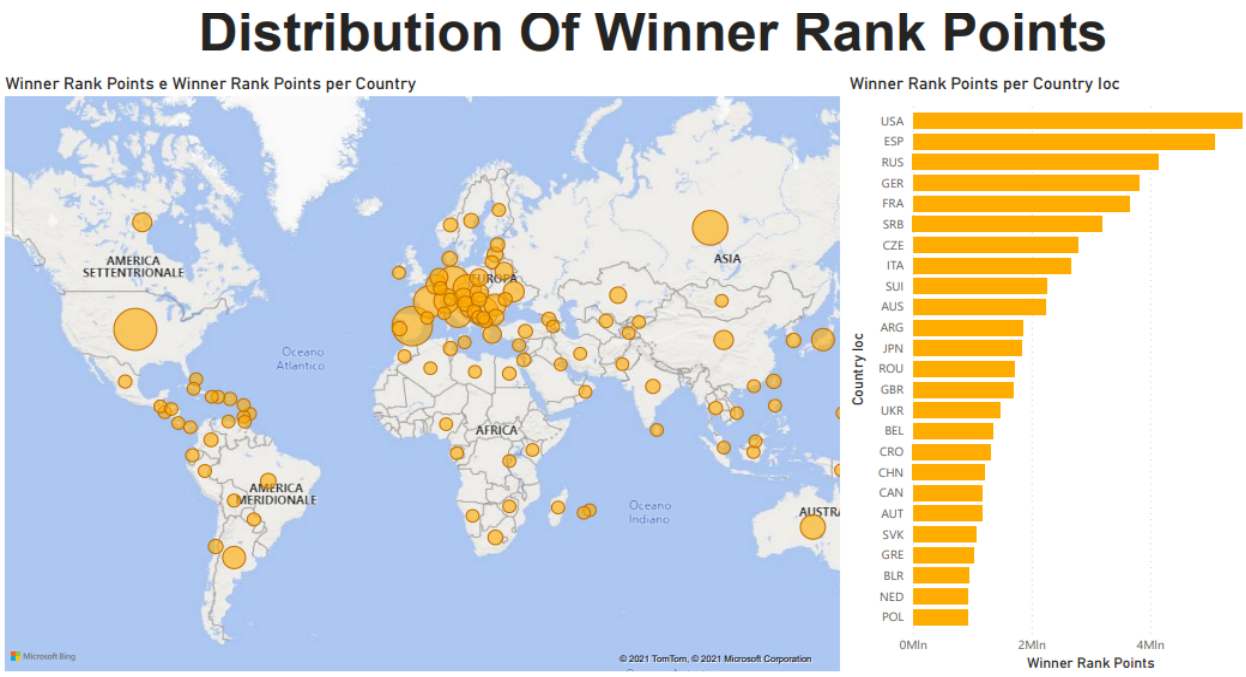
In the final query, on the columns, the set containing the three measures AVG_winner_rank_points, Loser_Rank_Points and Ratio is inserted. On the lines, on the other hand, for each line,is generated for each tournament, for each year the player with the highest Ratio.

The first few lines are shown below:

			AVG_Winner_rank_points	Loser_Rank_Points	Ratio
Abu Dhabi	2021	Sofia Kenin	1567.1724137931	5760	3.67540925893329
Acapulco	2016	Victoria Azarenka	946.020408163265	2935	3.10247006795383
Acapulco	2017	Kristina Mladenovic	830.632653061224	1580	1.90216456598118
Acapulco	2018	Alexander Zverev	1473.79591836735	4450	3.01941398028138
Acapulco	2019	Rafael Nadal	1477.10204081633	8320	5.63265080549338
Acapulco	2020	Alexander Zverev	1875.65306122449	3885	2.07127857508133
Acapulco	2021	Stefanos Tsitsipas	1702.96610169492	6765	3.97248071659617
Adelaide	2020	Felix Auger Aliassime	999.893617021277	1656	1.65617618895627
Adelaide	2021	Ashleigh Barty	1515.74418604651	9186	6.06038940116912
Agadir $15K	2017	Abir El Fahimi	83.6451612903226	132	1.57809487080602

Assignment 4

Create a dashboard that shows the geographical distribution of winner rank points and loser rank points.

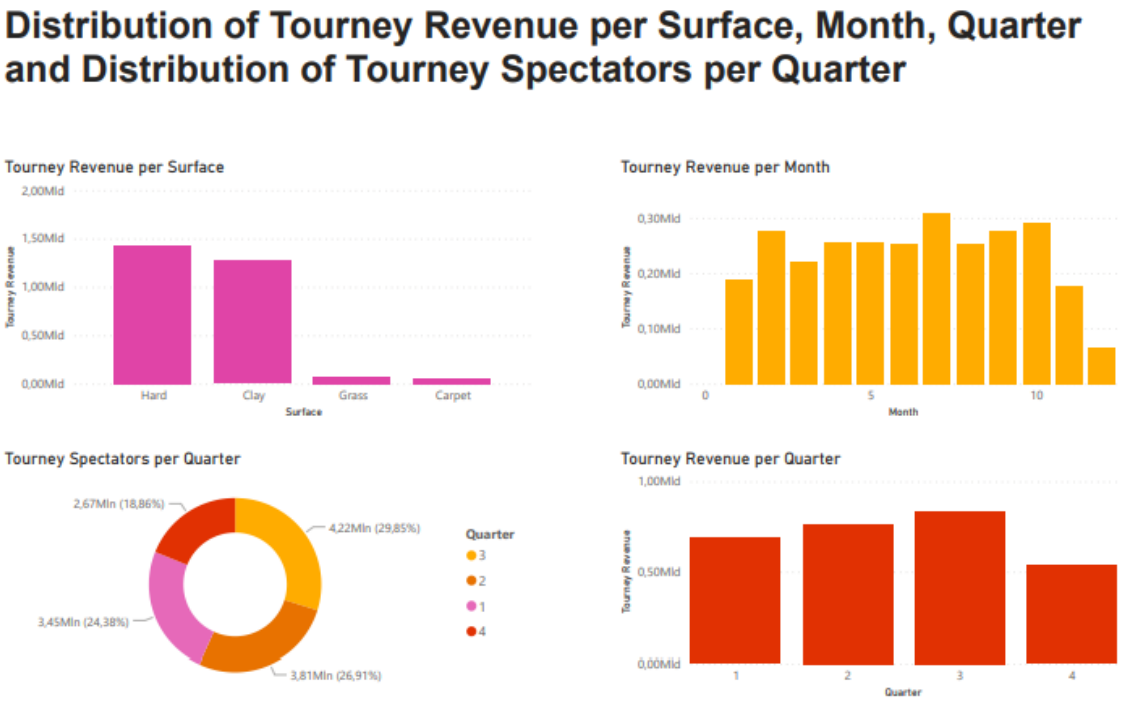


Power BI was used to perform this assignment. After connecting the software to our database, we represented the geographical distributions of winner rank point and loser rank point using two different graphs:

- A first representation of the results was presented by means of a *bubble distribution* on the geographical map which, through the size of the spheres , is able to suggest the amount of points per country.
- A second representation refers instead to a *bar graph*, which allows you to understand the relationships of magnitude and also identify numerically which of the nations has a greater distribution than the others, effectively creating a ranking.

Assignment 5

Create a plot / dashboard of your choosing, that you deem interesting wrt the data available in your cube



Power BI software was also used for this last request. One of information that we think may be interesting to know is the following:

- the *Tourney_Revenue* Distribution *Surface, Month,* Quarter,and also the *Distribution The spectators* attending the match for Quarter.

Both of these information are collected within the image described above. Three different bar charts were used for Surface, Month and Quarter, given their easy comprehension, while a pie chart was used to represent the number of spectators for each Quarter.