Laboratory of Data Science Project

Presentato da:
Giuseppe Mirko Milazzo
Francesco Santucciu

Data Science & Business Informatics
A.A 2021/2022

Road-map of contents

1. Datasets presentation
2. Data wrangling and data cleaning
3. Loading into Database
4. SSIS
5. Data Cube Creation
6. Queries
7. Data visualization

1. Datasets presentation

Dataset	Dimension	Description
Tennis.csv	186073 rows 49 columns	Datasets containing the description of each match
countries.csv	124 rows 3 columns	Dataset containing the IOC code of the country, the name and the continent
country_list.csv	233 rows 6 columns	Dataset containing the name of the country and the speaking language
male_players.csv	55208 rows 2 columns	List containing first name and last name of all male players
female_players.csv	46172 rows 2 columns	List containing first name and last name of all female players

2. Data wrangling and data cleaning

Table	Primary keys
Tournament	Tourney_ID

Table	Primary keys
Date	Tourney_date

Define_table

Table	Primary keys
Match	{ tourney_ID, match_number }

Table	Primary keys
Geography	{ winner_IOC, loser_IOC }

Transform functions	Features created/trasformed
Transform_player	{ "byear", "sex" }
Transform_geography	{ "continent", "language" }
Transform_match	{ "match_id" }
Transform_date	{ "year", "month", "day", "quarter" }

Missing Values

Attributes	Percentage	Cleaning method
"Hand"	0.33%	Imputation with the mode
"Winner"/"loser_rank_points"	Winner ➡ 7,65% Loser ➡ 15,80%	Simulation of a normal distribution of "winner"/"loser_rank_points" of the player for each tourney
"Winner"/"loser_rank"	Winner ➡ 7,64% Loser ➡ 15,79%	Random forest regression as input the "winner"/"loser_rank_points"
"Byear"	20.93%	"Dropping" rows with value ‘?’
"Surface"	1.28%	Imputation with the mode
"Score"	0.09%	Imputazione con la moda
(20 Attributi con missing values >= 50%)		Dropping of the attribute

Geography

ISO COUNTRY CODE	ISO LANGUAGE
"SGP" (Singapore)	"cmn" (Mandarino)
"DEU"Germania	"de" (Tedesco)
"GRC" (Grecia)	"el" (Hellenic)
"UNK"(Kosovo)	"al" (Albanese)
"MNE" (Montenegro)	"mo" (montenegrino)
"NLD"(olanda)	"nl" (olandese)
"NGA"(nigeria)	"en" (inglese)
"PHL" (filippine)	"tl" (tagalog)
"TRI" (Trinidad del Tobago)	"en" (inglese)
"POC" (Pacific Ocean Countries)	"unknown"

The missing and/or incorrect data were compared withe the dataset at the following link: https://github.com/datasets/country-codes/blob/master/data/country-codes.csv
On the left are the codes of the languages absent in the link and manually added country by country

3. Loading into Database

The function **"load_csv"** takes as parameters:

- The csv file
- The connection and the name of the table in the DB

By doing so, it is possible to generalize the SQL command for insertion.

When it is in the header:

Through the "**get_header**“ function, we become aware of which attributes to insert into the table and through "**get_sql_params**“ we determine how many values need to be inserted. This allows us to create a parametrized SQL command.

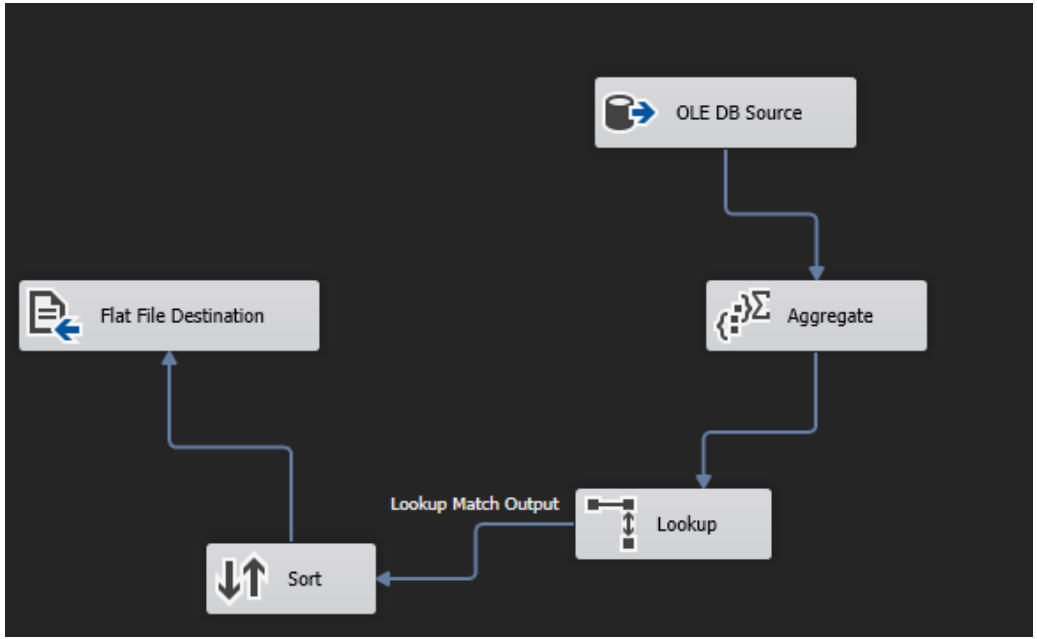
When it is not in the header:

it takes the row, splits the values, and inserts them into the database table.

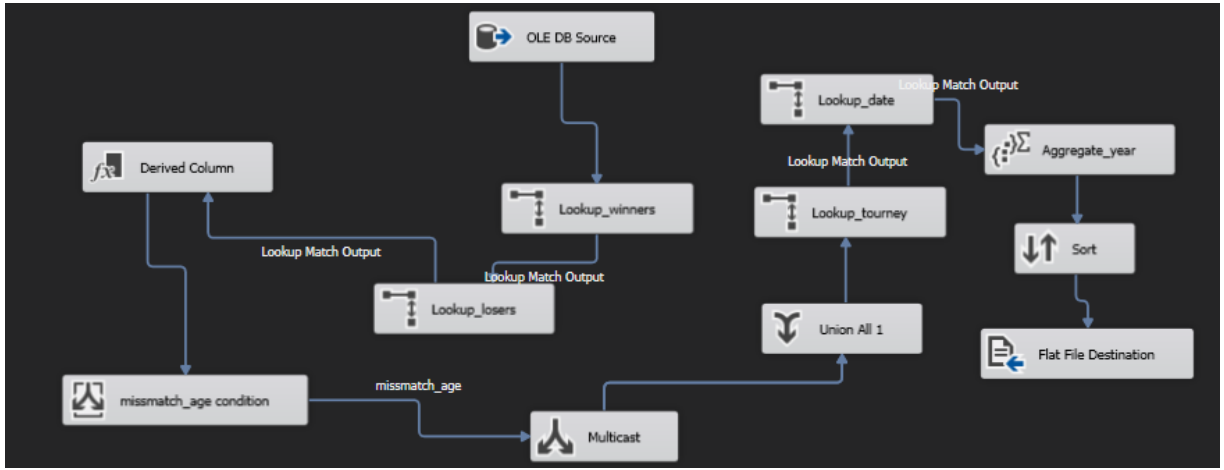
4. SSIS

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

Processo
1 Connection to the table **"match"**
2 Grouping by **"winner_id"** with counting
3 Performing a join with the **"player"** table to obtain the country and the name of the player
4 Sorting by country and by count
5 Output as flat file



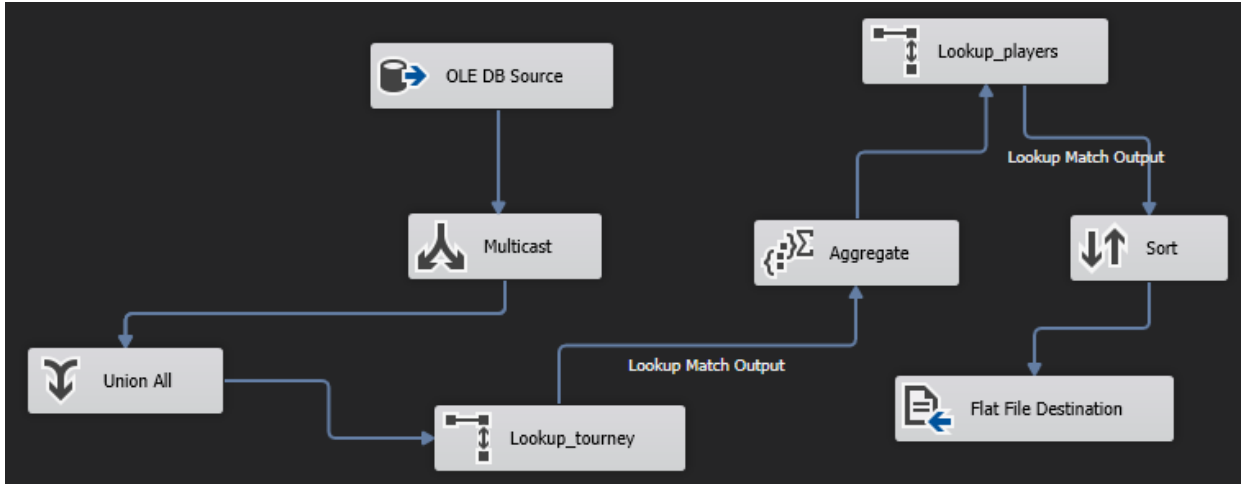
For each year, list the player that participated in the most age mismatches.



Processo

1 Connection to the table **"match"**	5 Join with tourney and date to obtain the year
2 Join with the table **"player"** to obtain the **"byear"**	6 Grouping by year, player and count
3 Derivation of the column related to the age difference and selection of matches with a difference grater than or equal to 6	7 Sorting by year and by counting
4 Transformation of the pair **"winner_id"** and **"loser_id"** in one unique column**"player_id"**	8 Output as flat file

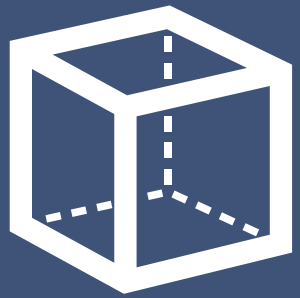
Calculate for each player the total number of spectators that he performed in front of



Processo

1 Connection to the table **"match"**	5 Joining with the table "**player**" to obtain the name of the player
2 Transformation of the pair **"winner_id"** and **"loser_id"** in one column **"player_id"**	6 Sorting by total number of spectators
3 Performing a **"join"** with the table "tournament" to obtain the number of spectators	7 Output as flat file
4 Grouping by player using as a measure the sum of the spectators count	

5. Data Cube Creation



Dimensione	Gerarchia	Composizione gerarchia
Player	Geography	Continent -> Country_IOC -> Player_ID
Tourney	DayMonthQuarterYear	Year -> The quarter -> The month -> Date_ID
Tourney	YearTournament	Tourney_name -> Tourney_ID

6. Queries

MDX

Query 1

```
-- Q1 Show the percentage increase in loser rank points with respect
-- to the previous year for each loser.
WITH MEMBER wrt_previous_year AS

    ( ( [Measures].[Loser Rank Points] * 100 ) /
                        ([Tourney].[DayMonthQuarterYear].CURRENTMEMBER.LAG(1),
                                                [Measures].[Loser Rank Points]  ) ) -100,
                                                        FORMAT_STRING = "STANDARD"

SELECT { [Measures].[Loser Rank Points], wrt_previous_year } ON COLUMNS,
    NONEMPTY(([Tourney].[DayMonthQuarterYear].[Year], [Loser].[Player Id].[Player Id])) ON ROWS

FROM [Group 30_tennis];
```

The formula **"wrt_previous_year"** was obtained through the following proportion:

$$loser\_rank\_points_{t-1} : 100 = loser\_rank\_points_t : x$$

Because it was necessary the previous year, we exploited the hierarchy **"DayMonthQuarterYear"** to obtain it through the **"LAG"** function

Query 2

```
-- Q2 For each tournament show the total winner rank points in percentage with respect
-- to the total winner rank points of the corresponding year of the tournament.

WITH MEMBER grand_total_tourney AS

    ([Tourney].[YearTournament].PARENT, [Measures].[Winner Rank Points])

MEMBER perc AS

    ([Measures].[Winner Rank Points] / grand_total_tourney), FORMAT_STRING = "percent"

SELECT {[Measures].[Winner Rank Points], grand_total_tourney, perc} ON COLUMNS,
    ([Tourney].[Year].[Year], [Tourney].[YearTournament].[Tourney Id]) ON ROWS

FROM [Group 30_tennis];
```

Through the hierarchy **"YearTournment"** we were able to distinguish different yearly editions of that tourney and viceversa

In this way we were able to compute the total **"winner_rank_points"** of all the edition of that tourney and finally to obtain the percentage of that editions of the tourney with respect to all of its editions

Query 3

```
-- Q3 Show the winners having a total winner rank points greater than the average winner
-- rank points in each continent by continent and year.
WITH MEMBER grand_total AS

    ([Winner].[geography].PARENT.PARENT, [Tourney].[DayMonthQuarterYear].PARENT, [Measures].[Winner Rank Points])

MEMBER avg_rank_points AS

    grand_total / ([Winner].[geography].PARENT.PARENT, [Tourney].[DayMonthQuarterYear].PARENT,
                                                        [Measures].[Match Count])

SELECT { [Measures].[Winner Rank Points], avg_rank_points } ON COLUMNS,
    FILTER( ([Winner].[Continent].[Continent],[Tourney].[DayMonthQuarterYear].[Year], [Winner].[geography].[Player Id]),
                              [Measures].[Winner Rank Points] > avg_rank_points ) ON ROWS

FROM [Group 30_tennis];
```

Inserting **"player_id"** into the hierarchy **"geography"** we were able to scale it to obtain the continent of origin of that player

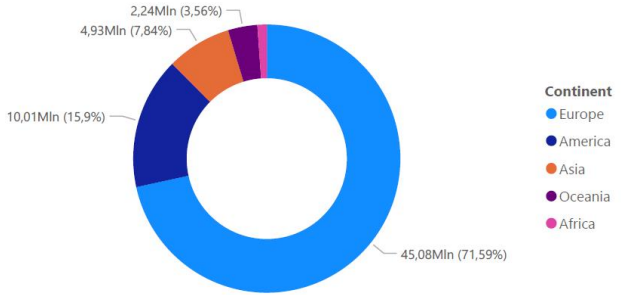
By summing all the **"winner_rank_points"** of matches from the players's continent of origin across all years, and dividing it by the number of matches played, we computed the average **"rank_points"**

7. Data visualization

Assignment 4

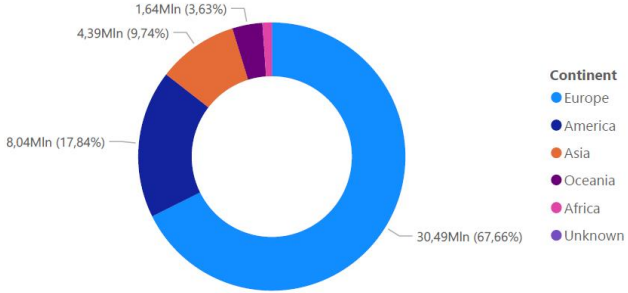
Winner Rank Points per Country loc

Winner Rank Points per Continent



Loser Rank Points per Country loc

Loser Rank Points per Continent

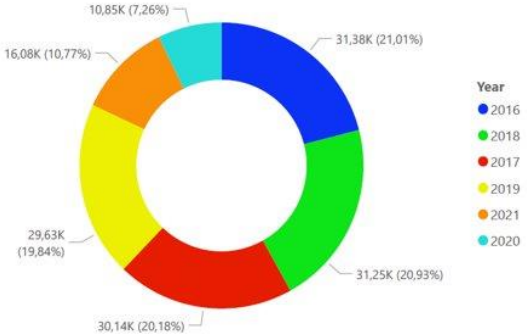


Assignment 5

Match Count per Country loc e Sex

Sex ● F ● M

Match Count per Year



Grazie dell attenzione!