Laboratory of Data Science a.a. 2021/2022

Gruppo 2
Giada Traina 616682
Mario Proia 616679

Panoramica

- Processo ETL
- Analisi e trasformazione dei dati (SSIS)
- Analisi multidimensionale con cubo OLAP
- Reportistica

Processo ETL

- Manipolazione dei file CSV
- Gestione missing values e duplicati
- Creazione schema
- Caricamento del server

Manipolazione dei file csv

- Creazione della tabella dimensionale **Match**:

  1. Creazione Match_id = concatenazione match_number e tourney_id

  2. Eliminazione id duplicati

  3. Correzione delle righe con valori di riga diversi e uguale Match_id

Manipolazione dei file csv

- Creazione della tabella dimensionale **Players**:

  1. Creazione dei file males e females tramite concatenazione del nome e cognome

  2. Attribuzione del sesso e correzione dei record con Nome giocatore uguale e sesso diverso

  3. Eliminazione righe duplicate

  4. Correzione altri errori

Manipolazione dei file csv

- Creazione tabella dimensionale **Tournament:**

  1. Recupero di tutte le colonne necessarie dal file tennis.csv

  2. Creazione campo “tourney_id” concatenando tourney_id, tourney_level e tourney_name.

Manipolazione dei file csv

- Creazione tabella **Date**:

  1. Creazione dei campi “anno”, “mese” e “giorno” tramite l’utilizzo della stringa “date_id”

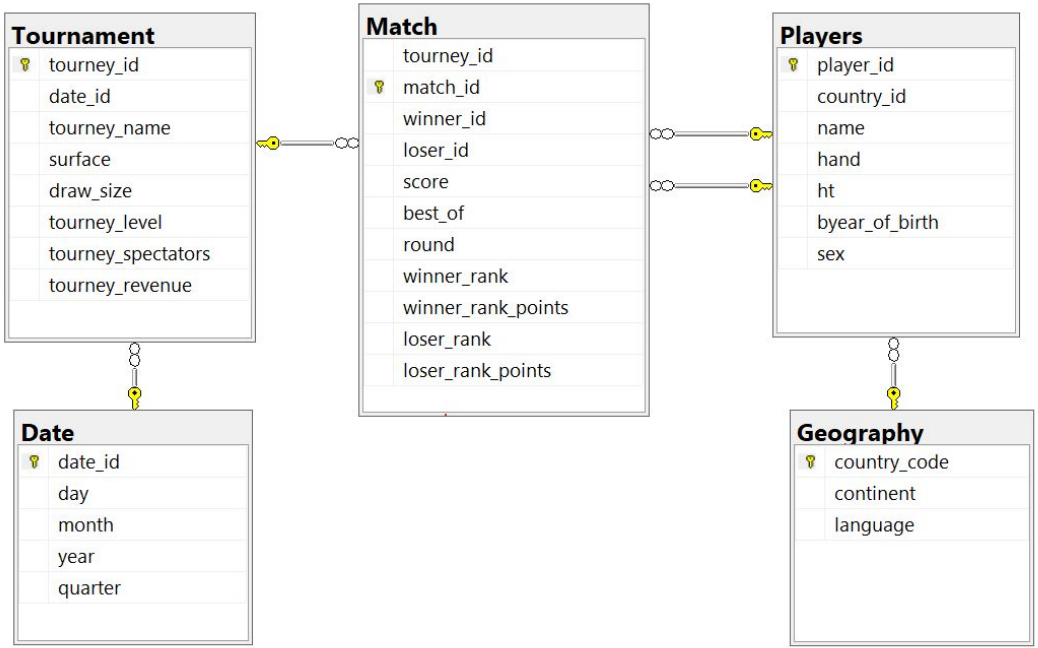
  2. Suddivisione dei mesi in intervalli trimestrali per la creazione delle strighe: “Quarter-1”,“Quarter-2”,“Quarter-3”,“Quarter-4”

Manipolazione dei file csv

- Creazione tabella **Geography**:

  1. Recupero delle lingue corrispondenti al country_ioc dal file country_list.csv
  2. Integrazione manuale delle lingue mancanti
  3. Integrazione dei paesi mancanti tramite il file players.csv

Creazione dello schema su SSMS



Gestione dei missing values e duplicati

- Le tabelle “**Geography**” e “**Date**” non presentano missing values o duplicati

- **“Players”**:
  - Eliminazione righe duplicate
  - Correzione id relativo a giocatori con nome diverso e stesso id
  - Sostituzione missing values in *sex* (controllo manuale), *hand* (distribuzione percentuale), *ht* (genere e continent)
  - Sostituzione outliers in *ht*
  - Sostituzione campi nulli in *byear_of_birth* con valore “-1”

Gestione dei missing values e duplicati

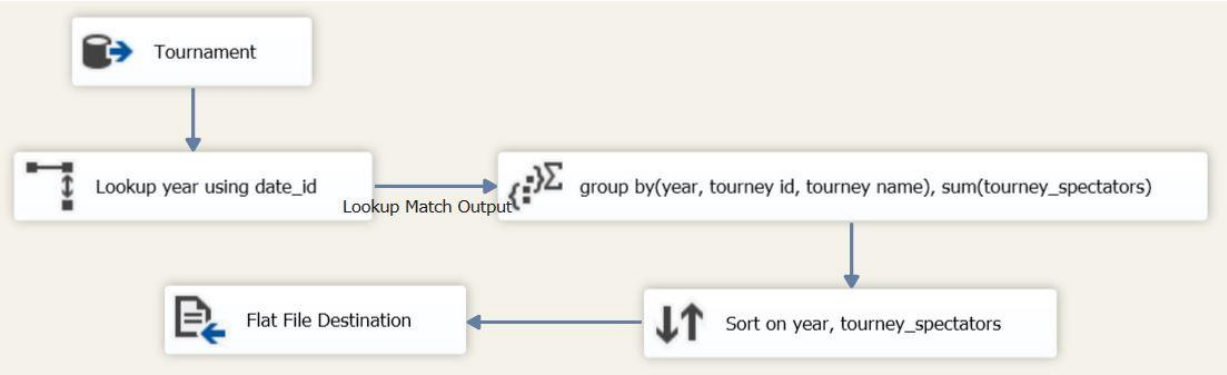
- **“Match”:**
  - Rimozione delle colonne con oltre 103000 missing values
  - Rimozione delle righe contenenti *score* mancanti
  - Sostituzione dei missing values di *winner_rank*, *loser_rank*, *winner_rank_points* e *loser_rank_points* con la media, raggruppando per anno e id del giocatore
  - Rimanenti valori non imputabili sostituiti con “-1” per evitare altre eliminazioni
- “**Tournament**”:
  - Sostituzione dei missing values di “surface” con la moda

Scrittura dei dati su SQL Server

```python
import pyodbc
import csv

# Credenziali di accesso per la stringa di connessione
server = 'tcp:131.114.72.230'
database="Group_2_DB"
username="Group_2"
password="ROJQAAGH"

connectionString = 'DRIVER={ODBC Driver 17 for SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+ password
cnxn = pyodbc.connect(connectionString)
cursor = cnxn.cursor()

count = 0
while count < 190: # Eseguo il commit un chunk alla volta
    file = open(r'C:\Users\Mario\Desktop\UniPi\2° anno\1° semestre\Lab\Progetto\Datasets\Tabelle da inserire\tab da inserire\chunk_match\
    csv_file = csv.DictReader(file, delimiter = ",")

    # Query di inserimento
    sql = 'INSERT INTO Match(tourney_id, match_id, winner_id, loser_id, score, best_of, round, winner_rank, winner_rank_points,loser_rank
    i=1
    print('--- Inizio file ', count, ' ---')
    for row in csv_file:
        val = (row["tourney_id"], row['match_id'], row['winner_id'],
               row['loser_id'], row['score'], row['best_of'],
               row['round'], row['winner_rank'], row['winner_rank_points'],
               row['loser_rank'], row['loser_rank_points'])
        cursor.execute(sql, val)

        print("Inserimento row ", i) # Tengo traccia di quale riga è stata inserita fin'ora
        i=i+1

    file.close()
    cnxn.commit()
    print('Commit file ', count, '\n')
    count+=1

cursor.close()
cnxn.close()
print("Connection closed.")
```

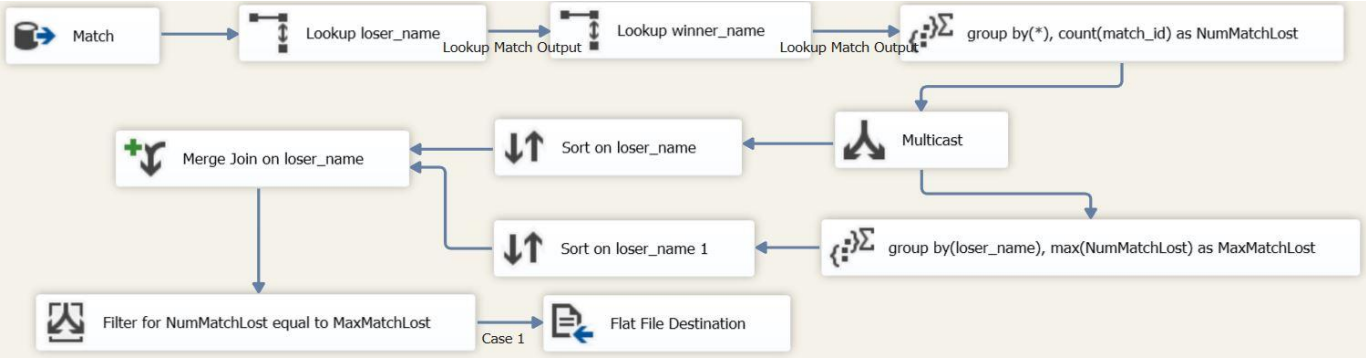
SSIS

Assignment 0: *Per ogni anno, i “tournaments” ordinati per numero di spettatori.*



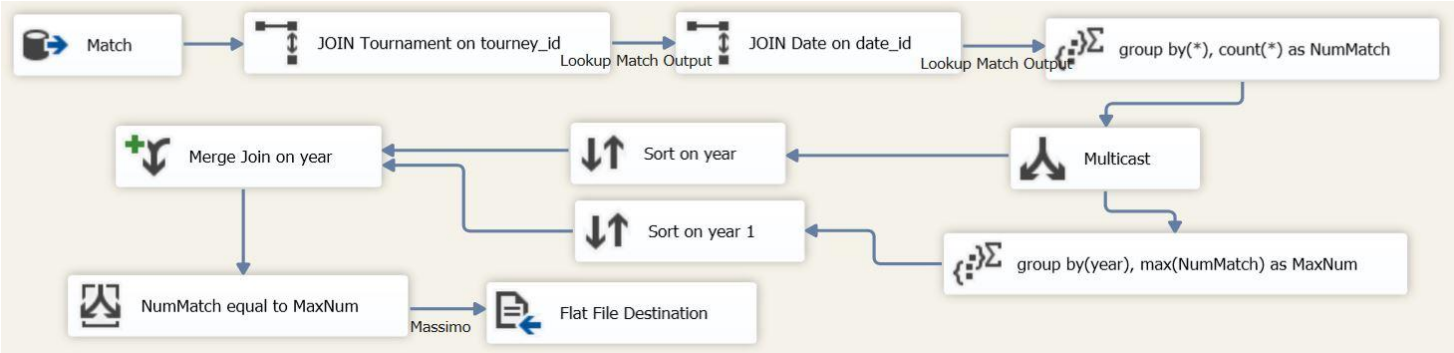
SSIS

Assignment 1: *Per ogni player, la sua "nemesi" rappresenta il player contro cui lui/lei ha perso più match. Elencare ogni player con la rispettiva nemesi e il numero dei match persi.*



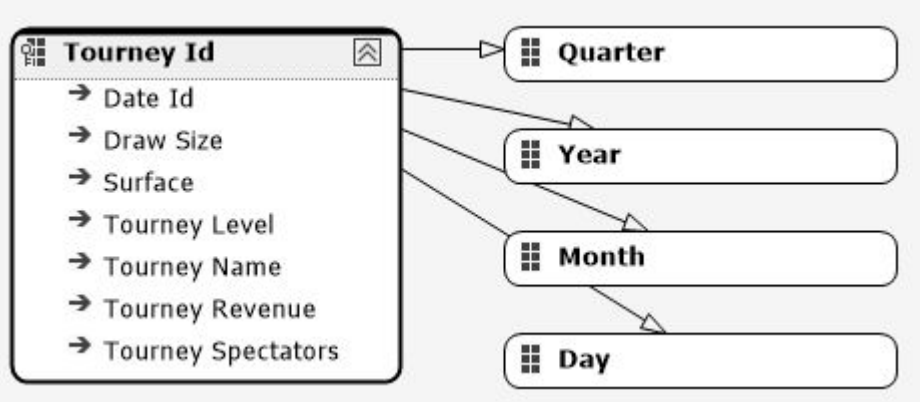
SSIS

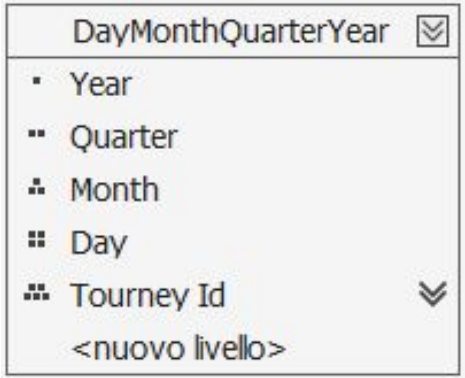
Assignment 2: *Per ogni anno, il nome del torneo con il maggior numero di match giocati.*



CUBO OLAP

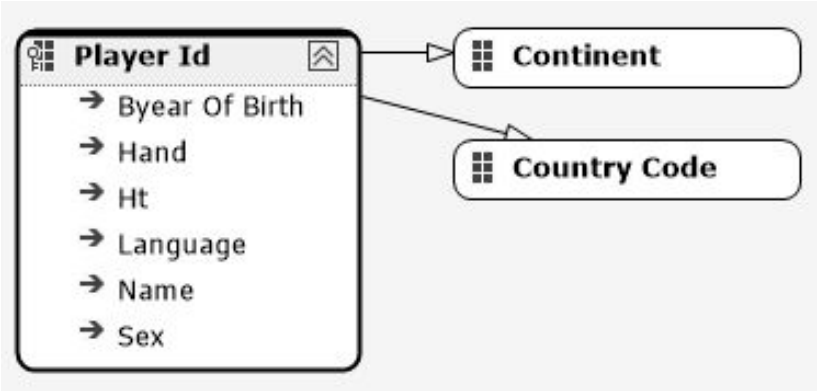
Focus su **Time**





CUBO OLAP

Focus su **Geography**



CUBO OLAP

Query 1: mostrare il numero dei n_distinct_winners in relazione al paese e il totale rispetto ad ogni continente.

```
select [Measures].[n_distinct_winners] on columns,
nonempty(([Winner].[Geography].children, [Winner].[Country Code].members)) on rows
from [Group 2 DB]
```

CUBO OLAP

Query 2: mostrare il totale dei winner_rank_points per ogni anno e la somma cumulata annuale dei winner rank points dei giocatori Europei.

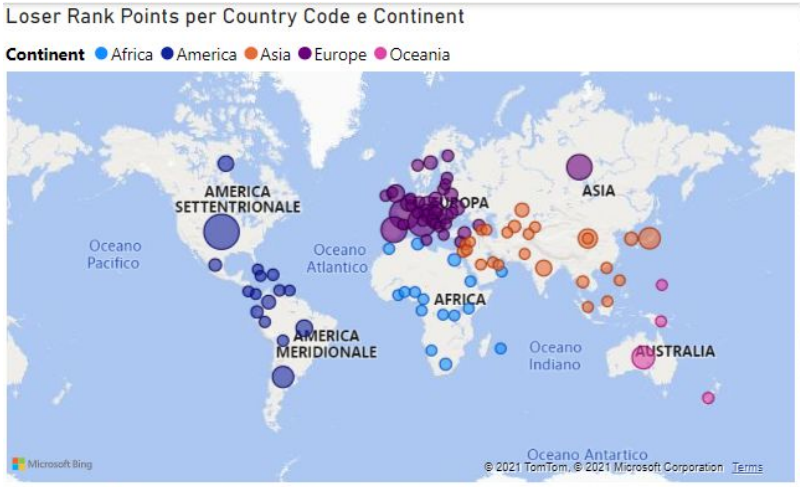
```
with member EU_running_total as
(
sum( PERIODSTODATE ([Tournament].[DayMonthQuarterYear].[All].level,
[Tournament].[DayMonthQuarterYear].currentmember),
(([Winner].[Geography].[Continent].&[Europe],[Measures].[Winner Rank Points]))
)
)

member European_Members as
(([Winner].[Geography].[Continent].&[Europe],[Measures].[Winner Rank Points])
)

select {[Measures].[Winner Rank Points], European_Members, EU_running_total} on columns,
nonempty(([Tournament].[DayMonthQuarterYear].[Year])) on rows
from [Group 2 DB]
```

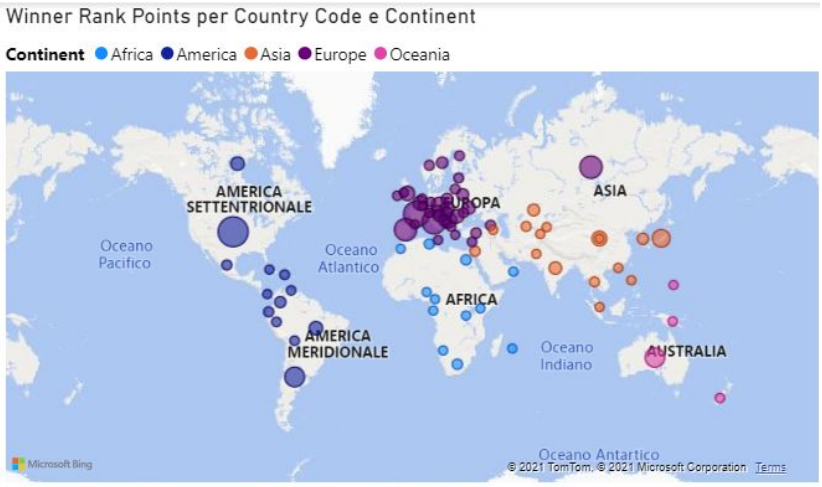
CUBO OLAP

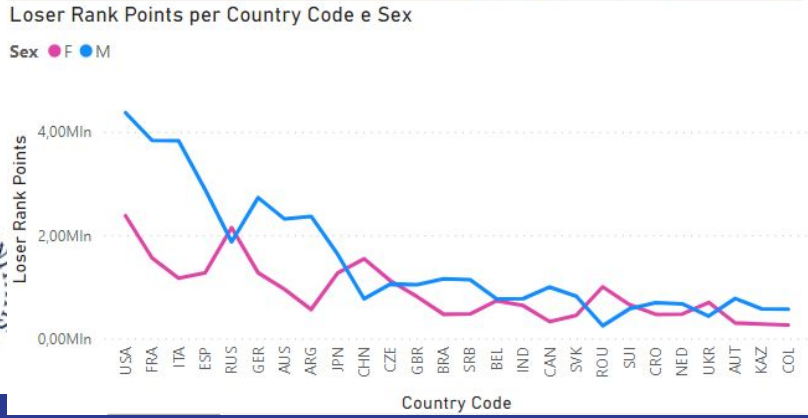
Query MDX 3: mostrare il rapporto tra il totale annuale dei winner rank points, rispetto al totale dell'anno precedente.

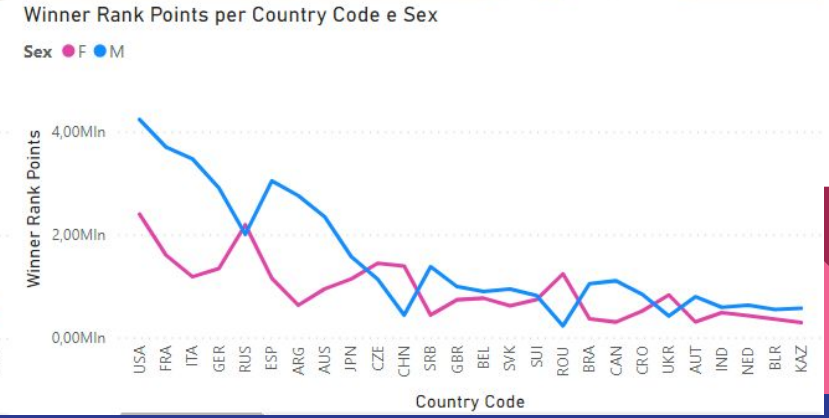
```
with member Previous_year_ratio as (
    iif (([Tournament].[DayMonthQuarterYear].currentmember.lag(1), [Measures].[Winner Rank Points])=0, 0,
        Round([Measures].[Winner Rank Points] / ([Tournament].[DayMonthQuarterYear].currentmember.lag(1),
        [Measures].[Winner Rank Points]), 4)
    )
)

select {[Measures].[Winner Rank Points], Previous_year_ratio} on columns,
nonempty([Tournament].[DayMonthQuarterYear].[Year].members) on rows
from [Group 2 DB]
```

REPORTISTICA

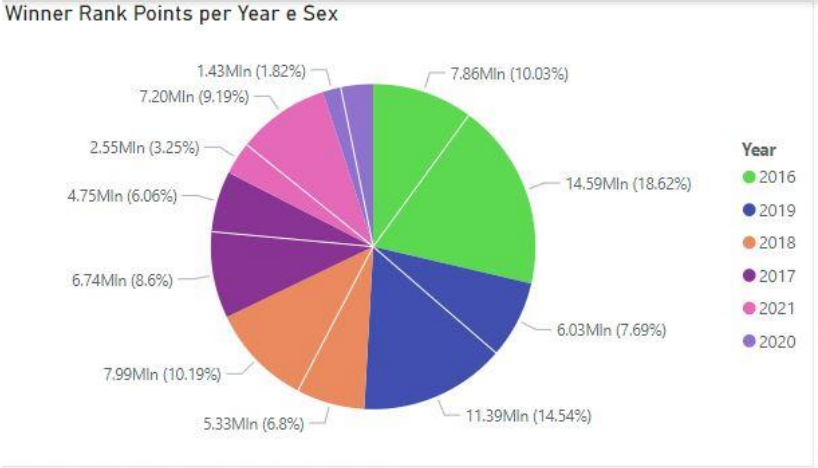


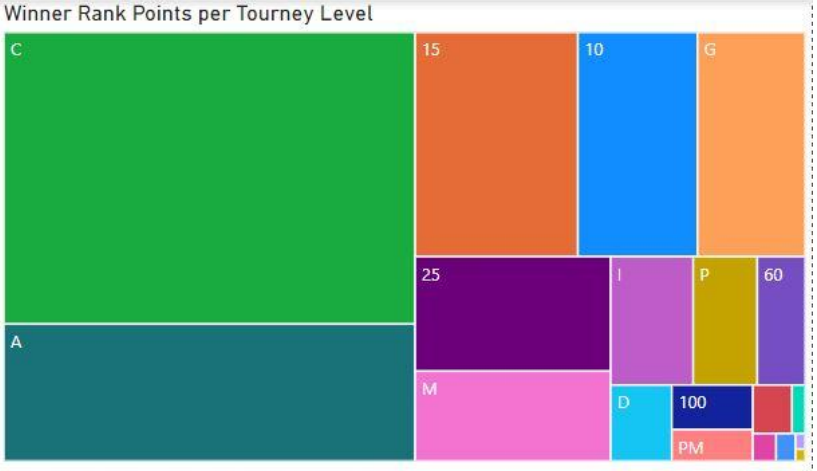


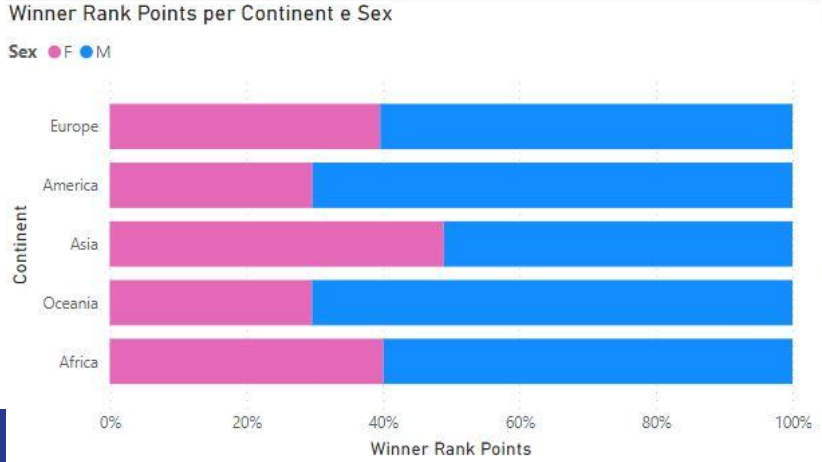


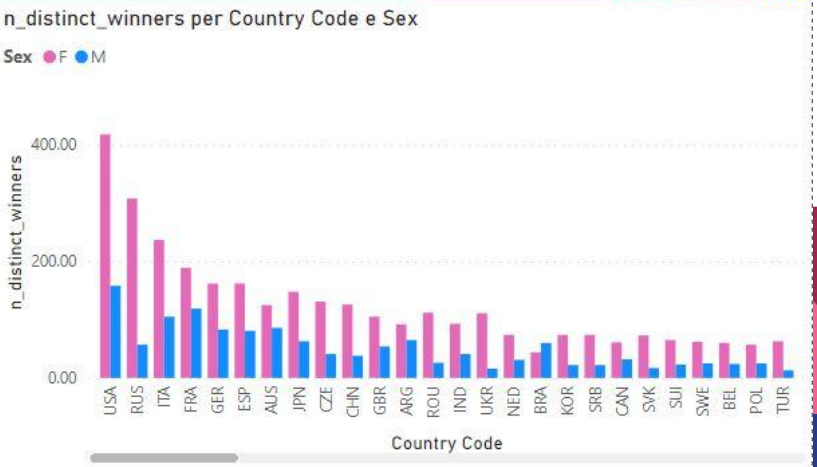


REPORTISTICA

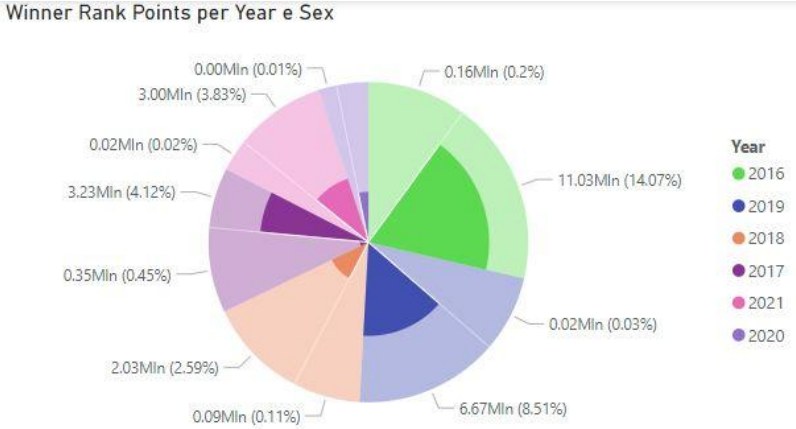


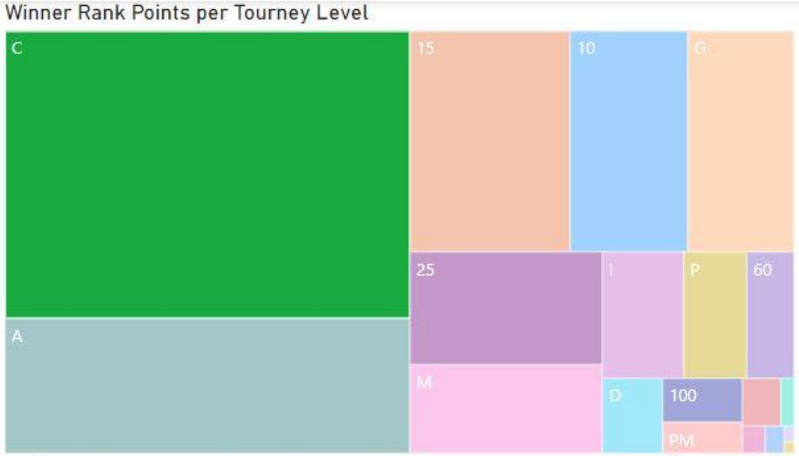


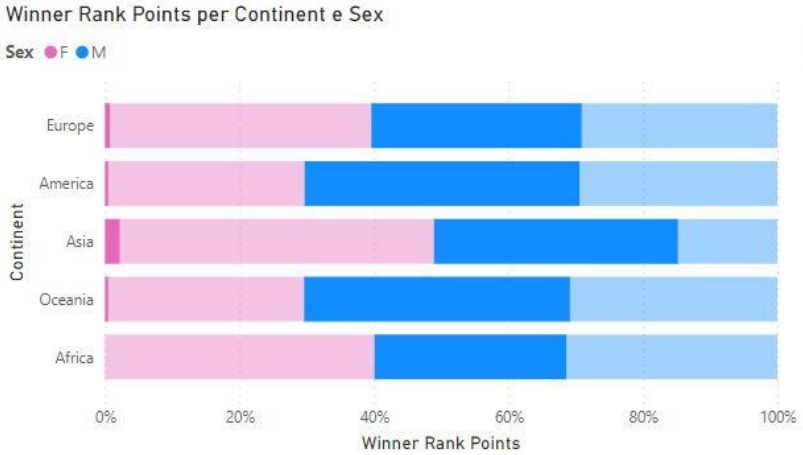


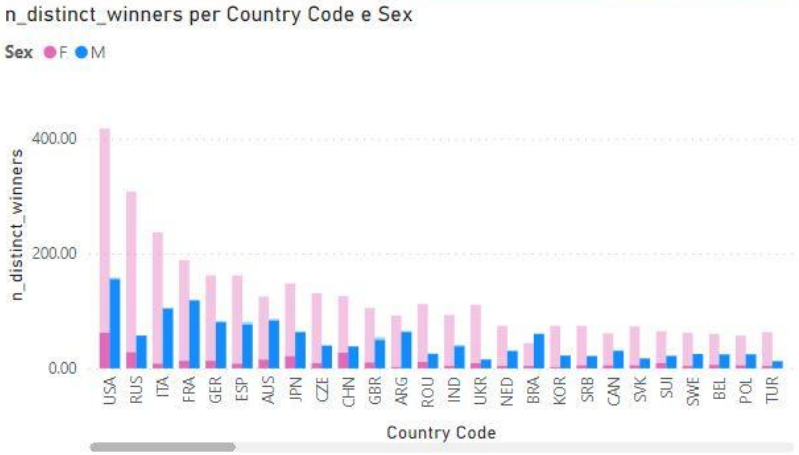


REPORTISTICA









Grazie per l’attenzione