# LABORATORY OF DATA SCIENCE

LUCA MARIANO (624991) LORENZO TESTA (564492)
RICCARDO TOTI (537037)

December 27, 2021


UNIVERSITÀ DI PISA

## CONTENTS

Questo breve report descrive i passaggi che ci hanno portato alla creazione e al popolamento del database *Group23HWMart*, che contiene informazioni su partite, giocatori e tornei di tennis. Il report è così strutturato: nella sezione 1 descriviamo come abbiamo creato lo schema del database *Group23HWMart*; nella sezione 2 riportiamo la nostra strategia per la creazione delle tabelle geography.csv, player.csv, tournament.csv, date.csv e match.csv principalmente a partire dal file tennis.csv; nella sezione 3, invece, descriviamo i passaggi di popolamento del database. Per semplicità di espressione, d'ora in avanti indicheremo i file omettendo il loro tipo, a meno di potenziali equivoci. Nella sezione 4, sfruttiamo il database appena costruito per effettuare delle query SSIS. Nella sezione 5, infine, utilizziamo il database per rispondere a delle business questions attraverso interrogazioni MultiDimensional eXpressions (MDX).

# 1 CREAZIONE DEL DATABASE

Utilizzando SQL Server Management Studio, abbiamo caricato lo schema del database *Group23HWMart* sul server lds.di.unipi.it. Per farlo, abbiamo dovuto utilizzare le nostre credenziali e accedere al database. Abbiamo perciò ricreato fedelmente la struttura richiesta per poter ospitare i nostri dati, preoccupandoci di definire omogeneamente i data type per tutte le variabili, le *Primary Key* e le *Foreign Key* ed infine delle relazioni fra le tabelle.

# 2 CREAZIONE DELLE TABELLE

Dopo aver creato lo schema del database, ci occupiamo di effettuare gli split necessari per estrarre le informazioni richieste dal file tennis, distribuendo, correggendo e analizzando gli attributi secondo le richieste dell'assignment.

## 2.1 Geography

Per creare il file geography e riempirlo con gli attributi richiesti, abbiamo innanzitutto controllato il *country IOC* dei giocatori presenti nel dataset tennis. Successivamente, al fine di creare una tabella che contenesse *country IOC* (nello standard ISO3), il rispettivo continente e la lingua parlata in ciascun Paese, abbiamo eseguito un parsing del file Countryinfo.txt in un file .csv (utilizzando txt2csv). Il risultato è stato una tabella che aggregava le informazioni geografiche, come auspicato.

In questa tabella abbiamo notato la presenza di due missing values, per i quali abbiamo applicato un'imputazione manuale. Questa scelta è ovviamente dipesa dalla dimensione minimale del problema di assenza di informazioni. In particolare, risultavano mancanti le features *Continent* e *Languages* delle due *country IOC* **URU** e **POC**. Per la prima, abbiamo proceduto attingendo continente e lingua da *Wikipedia*. Per quanto concerne **POC**, invece, abbiamo scoperto che corrisponde al gruppo di piccole nazioni dell'Oceano Pacifico meridionale, sudoccidentale, centrale e occidentale, che competono collettivamente come un unico paese nei tornei di tennis di Coppa Davis (uomini) e Fed Cup (donne), probabilmente in funzione della loro ridotta popolazione. Di conseguenza, abbiamo attribuito a **POC** il

continente **Oceania** e le lingue parlate nella maggior parte di queste nazioni come *Languages*.

## 2.2 Player

Per la tabella player sono stati estratti gli identificativi dei giocatori dalle colonne *winner id* e *loser id* di tennis. In questo caso, abbiamo prestato estrema attenzione alla possibile presenza di duplicati. Di conseguenza, abbiamo selezionato non solo gli identificativi ma anche i nomi dei giocatori. Ciò ha evitato due potenziali errori (che si sono verificati in circa 50 casi):

**OMONIMIA** Perdere tennisti diversi (con nomi diversi) che però condividono lo stesso identificativo;

**SINONIMIA** Contare più volte il medesimo tennista che possiede identificativi diversi.

### 2.2.1 Sesso e data di nascita

A questo punto, abbiamo estratto l'attributo *Sex* facendo il match dei giocatori con i file male_player.csv e female_player.csv, utilizzando come chiave il nome dei giocatori. Abbiamo notato che per circa 30 tennisti il matching non consentiva di reperire l'attributo. Pertanto, abbiamo provveduto imputando manualmente i valori mancanti, trovando le informazioni su *Wikipedia* e *ATP*.

Per quanto concerne la feature *Year of birth*, abbiamo sfruttato la presenza di età dei giocatori e date (anni) dei tornei per calcolarne la differenza, in modo tale da ottenere una stima arrotondata dell'età dei giocatori.

### 2.2.2 Problematiche

Dopo aver raccolto e organizzato informazioni su sesso e anno di nascita dei giocatori, abbiamo cercato di risolvere alcune problematiche riscontrate nel nostro file player.

In primo luogo, abbiamo notato l'assenza di 33 valori di *hand*. Abbiamo imputato i missing values reperendo manualmente le informazioni su *ATP*.

In secondo luogo, abbiamo sistemato alcuni dati errati relativi ad *ht*, le altezze dei giocatori, che presentavano valori assolutamente incompatibili con la distribuzione osservata delle altezze dei giocatori (e in generale con l'altezza di soggetti umani). Anche in questo caso abbiamo proceduto con un retrieval manuale delle informazioni sostitutive.

Infine, abbiamo risolto i problemi di disambiguazione precedentemente citati, controllando l'effettiva esistenza di giocatori multipli con stesso identificativo e nomi diversi o, viceversa, l'effettiva non-esistenza di giocatori con stesso nome e identificativi diversi.

## 2.3 Tournament

Da tennis abbiamo selezionato le colonne *tourney id*, *tourney name*, *surface*, *draw size*, *tourney level*, *tourney date*, *tourney spectators*, *tourney* e *revenue*. Controllando i missing values, abbiamo trovato 62 valori mancanti per la colonna *surface*, che contiene la tipologia di superficie del campo da tennis su cui si è disputato un torneo. Dato il numero esiguo di missing value e la loro concentrazione nei tornei Fed Cup 2016, Fed Cup 2017 e Davis Cup 2017, abbiamo deciso di imputare la tipologia di surface controllando manualmente l'informazione su Wikipedia.

### 2.4 Date

Per creare la tabella contenente i riferimenti temporali a ciascun torneo abbiamo creato un identificativo unico per ogni data. A questo punto, abbiamo eseguito uno split sulle stringhe contenenti le date per estrarre anno, mese e giorno. Ottenute le informazioni singole, abbiamo controllato per il mese e inserito la colonna *quarter* per definire in quale quadrimestre si fosse svolto il torneo.

### 2.5 Match

La tabella match è stata estratta selezionando direttamente da tennis le colonne desiderate, con l'accortezza di aggiungere la colonna *match id*, che abbiamo ottenuto concatenando *match num* con *tourney id*.

A questo punto, abbiamo mappato gli identificativi dei giocatori secondo quanto contenuto nella tabella player. Inoltre, abbiamo omogeneizzato alcuni valori nella colonna *score* (**?**, $>$, **Def**), recuperando dai siti *ATP* e *Wikipedia* le informazioni corrispondenti.

Ci siamo poi occupati di trattare i 175 missing values nella colonna *score*. Tramite una ricerca incrociata sui vari siti specializzati, abbiamo scoperto che le partite cui mancano i risultati non sono effettivamente mai esistite. Riteniamo dunque che ci sia stato un problema di creazione del data set, dato che si sono verificati episodi di crawling di dati inesatti.

Sono stati rilevati delle incongruenze (presenza di duplicati) nella colonna *match id*. Abbiamo dunque deciso di eliminarla, considerato il fatto che probabilmente esse si devono semplicemente a problematiche di joining.

Rimangono delle colonne che presentano dei missing values, talvolta con informazioni assolutamente parziali. Davanti a questo problema, possiamo procedere attraverso due strade. La prima prevede l'imputazione dei missing values utilizzando algoritmi di imputazione, come MICE (Multiple Imputation by Chained Equations) o Random Forest. Anche se questi algoritmi garantirebbero la completezza dei dati da inserire nel database, noi abbiamo deciso di proseguire attraverso una seconda strada, che non crea valori artificiali ma lascia vuote le celle con informazione mancante. Riteniamo infatti che questa scelta sia più prudente, perché non crea deformazioni nelle distribuzioni e non modifica le correlazioni tra variabili. Inoltre, la nostra scelta è supportata dal fatto che le statistiche descrittive computate sui dati imputati sarebbero distorte, mentre questo problema non si pone sui dati originali. Dato il fatto che non dobbiamo eseguire fit di modelli complessi sui dati a nostra disposizione, non dobbiamo preoccuparci nemmeno di avere abbastanza osservazioni per stimare correttamente i parametri. Pertanto, la nostra scelta ci sembra la più robusta sia da un punto di vista prettamente statistico sia da un punto di vista di buon senso generale.[1]

## 3 POPOLAMENTO DEL DATABASE

Una volta ottenuti i vari file, non rimane che popolare il database. Per farlo, abbiamo utilizzato il pacchetto pyodbc, che offre un'interfaccia semplice per instaurare un dialogo tra Python e database.[2] Abbiamo proceduto con

[1] Yu, B., Kumbier, K., *Veridical data science*, Proceedings of the National Academy of Sciences (2020), DOI: 10.1073/pnas.1901326117

[2] https://github.com/mkleehammer/pyodbc/wiki

semplici query *"INSERT INTO"* per popolare tutte le sezioni del database. L'unica accortezza che abbiamo dovuto implementare è stata quella che riguarda match, che, date le sue dimensioni importanti, non veniva caricato con completezza nel database. Pertanto, abbiamo diviso match in dieci subset, e abbiamo caricato individualmente ciascuno di loro, riuscendo così a completare le operazioni di popolamento del database.

# 4 QUERIES

Per risolvere le task contenute in questa sezione abbiamo utilizzato SSIS.

## 4.1 Player ordinati in base alle vittorie

Per ottenere quanto richiesto dalla traccia, ossia di ottenere per ogni tournament i plauer ordinati in base al numero di matches vinti, abbiamo innanzitutto e separatamente eseguito l'accesso ai due dataset Match e Player. A questo punto, abbiamo ordinato le righe di Match utilizzando winner_id, e abbiamo effettuato la medesina operazione di ordering su Player, utilizzando la colonna player_id. A questo punto, abbiamo effettuato un join tra i due data set ordinati. Il join è stato eseguito su player_id. A questo punto, abbiamo raggruppato il data set emerso dall'operazione di join (group by) per tournament.counting. A questo punto, una volta ordinato il risultato per tour e number of win, abbiamo ottenuto quanto richiesto dalla traccia, e dunque abbiamo salvato il nostro risultato in un file .csv.

## 4.2 I tournament c.d. worldwide

Per rispondere alla seconda traccia, che chiedeva di listare tutti i tournament così detti "worldwide", abbiamo innanzitutto eseguito l'accesso a Match e Match_avg (anche questa volta in maniera separata). Per quanto riguarda Match, dopo aver sfruttato la funzione di look-up (che ci permette di effettuare ricerche con corrispondenza) su country, abbiamo raggruppato l'output per country.count win. In seguito abbiamo effettuato un order su country_ioc. Per quanto riguarda Match_avg, abbiamo effettuato un look-up su country_avg e, in seguito, un group by su country.count win_avg. A questo punto, abbiamo preso la media di win per ogni country, e abbiamo ordinato il risultato per country_ioc. Una volta ordinati i risultati per entrambi i data set, li abbiamo mergiati. A questo punto, per rispondere alla traccia, abbiamo sleezionato i top player, e salvato l'output.

## 4.3 Giocatori above-the-average

Al fine di risolvere anche la terza traccia, abbiamo eseguito l'accesso a Match_winner e Match_loser. Sul primo abbiamo effettuato due look-up, il primo usando winner e il secondo usando winner_continent. Anche su Match_loser abbiamo effettuato due look-up, uno su loser e uno su loser_continent. A questo punto, abbiamo eseguito un join sui due output, e abbiamo eliminato i duplicati che si presentassero in tale output. A questo punto, abbiamo effettuato un hot-encoding in modo tale da mappare in 1 e 0 le caratteristiche di continente. Abbiamo raggruppato il data set aumentato dall'encoding per tourney e calcolato la percentuale di giocatori

provenienti da ciascun continente (per ogni torneo). A questo punto, usando una suddivisione condizionata rispetto alla percentuale appena calcolata, abbiamo salvato solo la partizione di tourney che soddisfavano la richiesta della traccia.

# 5 MDX PER RISPONDERE A BUSINESS QUESTIONS

Prima di rispondere alle domande contenute in questa sezione, abbiamo costruito il cubo-dati partendo dal nostro database. Abbiamo posto particolare attenzione affinché tempo e geografia fossero ordinati secondo la relativa gerarchia. Le misure impiegate sono state i rank points di winners e losers.

## 5.1 Aumento percentuale nei winner rank points di ciascun winner rispetto all'anno precedente

Definiamo innanzitutto *prev_rank*, utilizzando *ParallelPeriod*. In questo modo, possiamo ottenere il valore laggato di un anno rispetto a quello corrente di una misura a nostra scelta (nel nostro caso, winner rank points). A questo punto, definiamo la percentuale di crescita per tutti i winners, tranne che per il primo anno a nostra disposizione (2016), perché ovviamente non deteniamo informazioni sul 2015 per calcolare il percentage increase. A questo punto, costruiamo una tabella dove inseriamo i winner rank points dell'anno corrente, quelli dell'anno precedente e la variazione percentuale come colonne, e anno e nome dei giocatori vincitori come righe. Per motivi di leggibilità, eliminiamo dalla tabella appena ottenuta tutte le righe che hanno 2016 come anno, poiché appunto rappresentano valori che non possiamo calcolare.

## 5.2 Per ogni continente, i total winner rank points come una percentuale del totale di winner rank points del continente corrispondente

Per risolvere questo task, aggreghiamo winner rank points per continente e per country. A questo punto, calcoliamo la percentuale di winner rank points per ciascuna country rispetto al continente in cui si trova. Infine creiamo una tabella con winner rank points della country, del continente e relativo rapporto come colonne, e country IOC come righe.

## 5.3 Losers che hanno un total loser rank maggiore del 10% del totale dei loser rank points in ciascun continente, per continente e anno

In questo caso, creiamo *tot_rank* aggregando i loser rank points. Calcoliamo quindi la percentuale dei loser rank points di ciascun giocatore rispetto a *tot_rank*. Creiamo una tabella dove inseriamo i loser rank points, il totale relativo al continente e la percentuale dei punti rispetto al continente come colonne, e per quanto riguarda le righe applichiamo un filtro. Il filtro consiste nella generazione di tutti nomi dei giocatori loser tali che abbiano una percentuale di punti rispetto a quelli aggregati per continente e anno maggiore del 10%, per tutte le coppie continente-anno.

# 6 DASHBOARDS

Per costruire le dashboards allegate a questo documento, abbiamo utilizzato Microsoft PowerBi.