INSTITUTO UNIVERSITÁRIO DE LISBOA

Armazenamento para Big Data

2023/2024

Tennis Professionals Dataset

Afonso Lourenço, nº 111487
Afonso Santos, nº 111431
Filipe Rego, nº 111533
Leonor Laborinho, nº 111287
CDB2

Índice

Introdução

Este relatório descreve o processo de importação, limpeza e tratamento de uma base de dados disponibilizada. Esta base de dados, atp, contém diversas informações relacionadas com jogadores de ténis e com jogos e torneios onde os mesmos participaram.

O principal objetivo deste relatório é mostrar, detalhadamente, os passos tomados desde a importação e limpeza da base de dados no Mongo, até à importação da mesma base de dados, agora limpa, para o Sql.
O relatório destaca os procedimentos adotados para a limpeza dos dados, abrangendo correções de dados inconsistentes, remoção de dados duplicados, entre outros.

Desta forma, para alcançar o propósito pretendido, divide-se o trabalho em 4 fases: Importação da base de dados, Limpeza da base de dados no Mongo, Exportação da base de dados e Importação dos arquivos CSV para o Sql.
Em cada fase aborda-se os desafios encontrados e as soluções escolhidas para os enfrentar, sempre com o propósito de preservar a integridade e coerência dos dados.

-

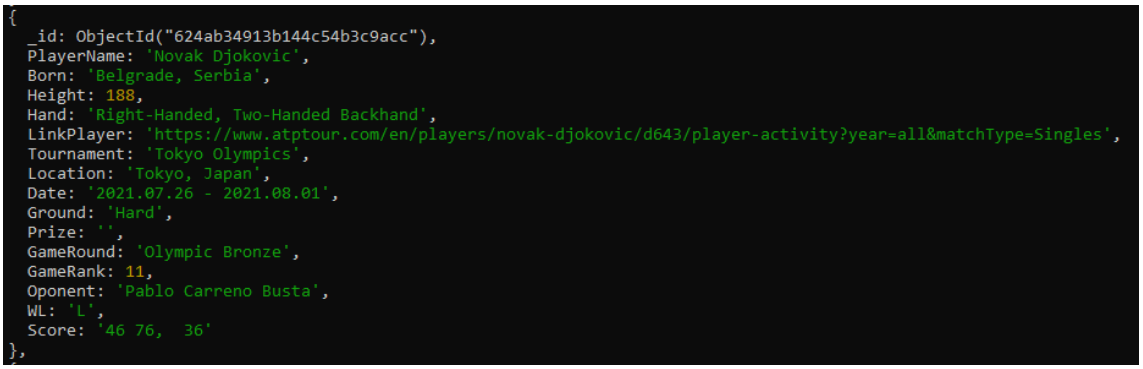
Importação da Base de Dados

Para consultar a base de dados fornecida, procedeu-se à importação da mesma para o Mongo, a partir do seguinte código:

mongoimport --db atp --collection atp --file c:\atp\atpplayers.json

Este comando garante a importação dos dados do ficheiro JSON "atpplayers.json" para a collection atp na base de dados atp.

Após esta importação, executam-se os comandos "use atp" e "db.atp.find()" na Shell do Mongo (Mongosh) para se visualizar toda a base de dados.



A imagem acima mostra um exemplo de uma linha da base de dados.

Podemos observar a existência dos vários campos pertencentes à base de dados: '_id', 'PlayerName', 'Born', 'Height', 'Hand', 'LinkPlayer', 'Tournament', 'Location', 'Date', 'Ground', 'Prize', 'GameRound', 'GameRank', 'Oponent', 'WL' e 'Score'.

Limpeza da Base de Dados no Mongo

Ao analisar a base de dados no Mongo Shell, identificamos a existência de certos erros e inconsistências, entre eles a presença de dados duplicados, dados em falta, erros de formatação, etc. Assim, de modo a garantirmos que a base de dados está corrigida, e em conformidade com os nossos objetivos, mostrou-se necessário realizar a limpeza da mesma. Decidiu-se realizar essa limpeza no Mongo.

Países:

Para dar início à limpeza dos dados, optou-se por começar por corrigir os países, dado que alguns demonstravam certos aspetos que necessitavam de ser retificados.

Para realizar esta correção, utilizou-se, como referência, a tabela disponibilizada no enunciado, com os nomes dos países e os respetivos códigos. Esta tabela serviu como base para estabelecer uma correspondência consistente entre os países mencionados na base de dados e os seus códigos.

Uma vez que existem linhas que não contêm nenhuma string que corresponde ao país, procedeu-se à correção dessas mesmas linhas, de modo a perder-se o mínimo possível de dados.

Dado que o campo 'Born' é maioritariamente do tipo 'Cidade:País', o critério de correção escolhido envolve apenas o texto que está após a última vírgula na string. Assim, estabeleceu-se como critério corrigir tudo o que está à direita da vírgula no campo 'Born', independentemente do tipo de conteúdo (pode ser um país, uma cidade ou até um estado mal escrito).
Desta forma, utilizou-se um código, no Mongo, que devolve todos os valores do campo 'Born' que estão depois da última vírgula, caso esta exista. Esse código é o seguinte:

db.atp.aggregate([{ $match: { Born: /,/ } }, { $project: { lastPart: { $arrayElemAt: [{ $split: ['$Born', ','] }, { $subtract: [{ $size: { $split: ['$Born', ','] } }, 1] }] } } }, { $group: { _id: '$lastPart' } }])

Assim, este código começa por filtrar as linhas cujo campo 'Born' contém uma vírgula, e acaba por devolver os valores que se encontram após a vírgula.

De seguida, para facilitar a alteração dos dados desejados, utilizou-se a ferramenta Excel, onde foram datados e corrigidos os casos que resultaram do código anterior. Desta forma, o documento Excel passou a conter uma coluna, 'Born', com todos os valores resultantes do código anterior, corrigidos. Além disso, foi adicionada uma segunda coluna, 'BornCountry', com os códigos correspondentes aos países (Retirados da tabela do enunciado).

Desta forma, pretende-se adicionar este novo campo, 'BornCountry', à collection atp. Com o auxílio do Excel, criou-se uma função que compõe o texto para o update. Ou seja, criou-se uma string do:

db.atp.updateMany({ Born: /Afghanistan/i }, { $set: { BornCountry: "AF" } })

para cada país ou respetiva correção.

Este código atualiza, de acordo com um determinado critério (Born: …) certas linhas da collection atp. Neste caso, procura as linhas cujo campo 'Born' contém a string 'Afghanistan', e define-se um novo campo, 'BornCountry' com o valor 'AF'. Ou seja, adiciona-se um novo campo, 'BornCountry', ao qual se atribui o código do país correspondente a 'Afghanistan', que é 'AF'.

A folha de Excel utilizada serviu também como auxílio para criar o campo 'LocationCountry', que contém os códigos dos países presentes em 'Location'. Assim, criou-se para cada país ou correção do mesmo uma string do tipo:

db.atp.updateMany({ Location: /Afghanistan/i }, { $set: { LocationCountry: "AF" } })

Correções pontuais (casos específicos):

Durante a análise da base de dados, identificaram-se duas situações particulares, que necessitavam de ser alteradas para garantir consistência e uniformidade dos dados.
Em primeiro lugar, observou-se a existência de dois jogadores, considerados por nós distintos: "Horacio De La Pena" e "Horacio de la Pena".
Concluiu-se que, apesar de no Mongo eles serem vistos como dois jogadores diferentes, no Sql o mesmo não iria acontecer (uma vez que o Sql não faz distinção entre maiúsculas e minúsculas). Deste modo, modificamos o nome "Horacio De La Pena" para "Horacio Pena", em 'PlayerName' e em 'Oponent', utilizando os códigos:

db.atp.updateMany({ PlayerName:"Horacio De La Pena" }, { $set: { PlayerName:"Horacio Pena" } })

db.atp.updateMany({ Oponent:"Horacio De La Pena" }, { $set: { Oponent:"Horacio Pena" }})

Além disto, chegou-se também à conclusão de que existe um torneio chamado "Valencia" que ocorre simultaneamente em Espanha e nos Estados Unidos da América. Para resolver esta situação, alterou-se o nome do torneio que ocorre nos EUA para "Valencia US", a partir do seguinte código:

db.atp.updateMany ({ Tournament:"Valencia", Location:"Valencia, CA, U.S.A." }, { $set:{ Tournament:"Valencia US" } }

Criação de collections:

De modo a facilitar a posterior criação do modelo relacional em Sql, criou-se as seguintes collections: Players, Tournament e Games.

Os códigos responsáveis pela criação das três collections encontram-se na folha dos scripts, junto dos outros códigos.

- Players:

Relativamente à collection Players, esta é constituída pelos campos:' _id,' 'PlayerName', 'Born', 'BornCountry', 'Height', 'Hand', 'HandTechnique' e 'PlayerLink'.

Definiu-se como critério chave que jogadores com nomes distintos são jogadores distintos. Isto dá-se devido à existência de jogadores 'Oponent' que não são 'PlayerName', o que nos impossibilita de distinguir nomes iguais para jogadores diferentes. Esta decisão baseia-se no facto de sabermos que apenas existem, com certeza, 7 jogadores distintos com nomes iguais (se utilizarmos o 'PlayerLink' como critério de diferença).

Para criar a collection Players, utilizou-se a função aggregate, onde se começou por filtrar as linhas cujo 'PlayerName' existe, ou seja, não é nulo. De seguida, indicou-se os campos a extrair para a nova collection, entre eles 'PlayerName', 'Born', 'BornCountry', 'Height', 'Hand', 'HandTechnique' e 'PlayerLink'.

Todos estes campos pertencem à collection original (atp), e foram copiados da mesma, com exceção de Hand e HandTechnique que resultam da divisão do campo Hand (O critério utilizado foi a separação pela vírgula (,) respetivamente pela ordem anterior).

Os campos são agrupados por PlayerName.

De seguida, dado que constatamos que existem jogadores que pertencem ao campo 'Oponent' mas não ao campo 'PlayerName', utiliza-se um código que adiciona à nova collection Players, os jogadores que apenas estão em 'Oponent'. Assim, sempre que estes ainda não tenham sido previamente inseridos nesta collection, serão inseridos. Neste caso, todos os campos exceto 'PlayerName' serão null. Isto deve-se ao facto de não querermos perder jogos (que por consequência perderíamos jogadores) e ao facto de apenas os 'PlayerName' terem a restante informação.

- Tournament:

A collection Tournament é constituída pelos campos '_id', 'LocationCountry' (contém os códigos dos países - criado anteriormente), 'Tournament' e 'Date'.

Para esta collection foram copiados os respetivos campos da collection original (atp) para esta.

- Games:

A collection Games é constituída pelos seguintes campos: '_id', 'Tournament', 'GameRound', 'Date', 'Ground', 'Players', 'Count', 'Player1', 'Player2', 'PlayerWin'.

A criação desta collection revela alguns desafios, uma vez que se verifica a existência de jogos em espelho, jogos repetidos e jogos sem 'Oponent'.

Para resolver esta situação, definiu-se que apenas são considerados jogos válidos, aqueles em que existe um 'Oponent', um 'PlayerName' e um jogador vitorioso.

Relativamente ao problema dos jogos em espelho, existem casos na collection atp como o seguinte:

'PlayerName': A 'Oponent': B 'WL': W

'PlayerName': B 'Oponent': A 'WL': L

Podemos observar que estas duas linhas correspondem ao mesmo jogo, simplesmente os nomes dos jogadores estão trocados.

Para facilitar esta situação, concantenou-se os nomes dos dois jogadores ('PlayerName' e 'Oponent') de modo a ser o nosso critério de equivalência de jogos em espelho.

De seguida, utilizou-se o "count" para contar quantos jogos cumprem o critério referido. Por exemplo, se 'count:2' isto pode significar que existe um jogo em espelho ('PlayerName': A, 'Oponent': B e 'PlayerName': B, 'Oponent': A) ou repetido ('PlayerName': A, 'Oponent': B e 'PlayerName': A, 'Oponent': B).

Assim, o count é utilizado para verificar se um certo par de jogadores já tem o jogo registado ou não. Após esta contagem, apenas se mantêm os primeiros valores encontrados, ou seja, os jogos (a mais) repetidos ou em espelho são descartados, só permanece o primeiro jogo encontrado.

Além disto, na collection Games, criou-se novos campos 'Player1', 'Player2' e 'PlayerWin', provenientes de três campos da collection atp (´PlayerName', 'Oponent' e 'WL').

Desta forma, 'Player1' e 'Player2' são os jogadores do jogo, sendo esta ordem arbitrária ('Player1': A, 'Player2': B e 'Player1': B, 'Player2': A não acontece, pois, violaria o princípio acima).

O campo 'PlayerWin' corresponde ao jogador vencedor. Este campo verifica se 'WL' é igual a W. Se for o caso, atribui-se 'PlayerName' a 'PlayerWin', caso contrário atribui-se 'Oponent'.

Exportação da Base de Dados

Após ter sido realizada a limpeza da base de dados "atp" no Mongo, é necessário importá-la para o Sql, onde vai ser construído o modelo relacional.

Para tal, deve-se exportar os dados do Mongo, das collections Players, Games e Tournament, para um formato CSV, o que é feito utilizando os seguintes códigos:

mongoexport --db atp --collection Players --type=csv --fields "PlayerName","Height","Born","BornCountry","Hand","HandTechnique","PlayerLink" --out C:\data\Players.csv

mongoexport --db atp --collection Games --type=csv --fields "_id.Tournament","_id.GameRound","_id.Date","_id.Ground","_id.Players","Player1","Player2","PlayerWin" --out C:\data\Games.csv

mongoexport --db atp --collection Tournament --type=csv --fields "Tournament","LocationCountry","Date" --out C:\data\Tournament.csv

Para além de serem utilizados estes três CSV´s exportados do Mongo, será também utilizado um CSV que contém a tabela com o nome de todos os países e o respetivo código (Retirada do link presente no enunciado).

Importação dos CSV Para o Sql

Para dar início à criação da base de dados em Sql, procedeu-se à importação dos quatro arquivos CSV para o Sql.

Durante o processo de importação, decidiu-se, por base, atribuir a todas as colunas o tipo VARCHAR(150), com exceção das colunas que contêm siglas de países (‘Code’, ‘BornCountry’ e ‘LocationCountry’), onde se optou por VARCHAR(2), e da coluna que contém as alturas dos jogadores (‘Height’), onde foi atribuído o tipo VARCHAR(3).

Os códigos relativos à importação para Sql encontram-se na folha de códigos.

Ao importar os arquivos para Sql, obtem-se quatro tabelas: Players, Games, Tournament e Country, cujas colunas correspondem aos campos de cada collection no Mongo (Com exceção da tabela Country que foi retirada do enunciado).

Assim, a tabela Players contém as colunas: ‘PlayerName’, ‘Height’, ‘Born’, ‘BornCountry’, ‘Hand’, ‘HandTechnique’, e ‘PlayerLink’.
A tabela Games contém as colunas: ‘GameRound’, ‘Date’, ‘Ground’, ‘Players’, ‘Player1’, ‘Player2’ e ‘PlayerWin’.
A tabela Tournament contém as colunas: ‘Tournament’, ‘LocationCountry’ e ‘Date’.
A tabela Country contém as colunas: ‘Name’ e ‘Code’.

Na tabela Country foi definida como chave primária a coluna ‘Code’, ou seja, os códigos dos países.

Relativamente à tabela Players, a chave primária escolhida é a coluna ‘PlayerName’.
Além disso, a tabela contém uma chave estrangeira, na coluna BornCountry, ligada à chave primária da tabela Country (‘Code’).

Na tabela Tournament assumimos como chave primária a combinação das colunas 'Tournament' e 'Date', uma vez que existem torneios com o mesmo nome, mas em datas diferentes. Desta forma, sendo a chave primária composta por ambas as colunas, garante-se que os valores são únicos.
Adicionalmente, a outra coluna existente nesta tabela, 'LocationCountry' funciona como chave estrangeira, estando ligada à chave primária da tabela Country ('Code').

Por fim, na tabela Games, a chave primária é composta pela junção entre as colunas 'GameRound', 'Date' e 'Players'.
Esta escolha foi a maneira mais eficaz de assegurarmos a unicidade de cada jogo.
Para além disto, temos uma chave estrangeira na coluna 'PlayerWin' ligada à chave primária da tabela Players ('PlayerName'), e temos também outra chave estrangeira na coluna 'Tournament', ligada à chave primária da tabela Tournament (´Tournament´).

A imagem seguinte proporciona uma visão clara da base de dados atp e das ligações entre as quatro tabelas, estabelecidas por meio das chaves estrangeiras.