

**Ano Letivo 2022/2023**

15 de dezembro de 2022

**ATP Players**

Base de Dados de Tenistas Profissionais



**UC** Armazenamento para Big Data

**Licenciatura** Ciência de Dados

Grupo 8, CDB1

**Docente**

Pedro Ramos

André Silvestre Nº104532

Diogo Catarino Nº104745

Francisco Gomes Nº104944

Rita Matos Nº104936

**Índice**

[Introdução 3](#_Toc121869933)

[Modelo Relacional e Não Relacional 3](#_Toc121869934)

[Base de Dados ATP Tennis Professionals 4](#_Toc121869935)

[Metodologia CRISP-DM 5](#_Toc121869936)

[1. *Business Understanding* 5](#_Toc121869937)

[2. *Data Understanding* 6](#_Toc121869938)

[Importação da Base de Dados 6](#_Toc121869939)

[Variáveis em Estudo 8](#_Toc121869940)

[3. *Data Preparation* 10](#_Toc121869941)

[Desenho da Base de Dados Relacional 10](#_Toc121869942)

[Variáveis Player\_Name e LinkPlayer 10](#_Toc121869943)

[Variável Born\_Country 11](#_Toc121869944)

[Casos Particulares 13](#_Toc121869945)

[Variável Height 14](#_Toc121869946)

[Variável Hand 15](#_Toc121869947)

[Variável Tournament\_Name 16](#_Toc121869948)

[Variável Tournament\_Country 17](#_Toc121869949)

[Casos Particulares 18](#_Toc121869950)

[Variável Tournament\_Date 19](#_Toc121869951)

[Variável Ground 20](#_Toc121869952)

[Variável Prize 20](#_Toc121869953)

[Variáveis Round, Opponent, Opponent\_Rank, Result e Score 21](#_Toc121869954)

[4. *Modeling* | *Evaluation* | *Deployment* 22](#_Toc121869955)

[Tabela Players 22](#_Toc121869956)

[Tabela Tournaments 24](#_Toc121869957)

[Tabela Games 25](#_Toc121869958)

[Duplicados 26](#_Toc121869959)

[Games 26](#_Toc121869960)

[Unique\_Games 27](#_Toc121869961)

[Diagrama do Modelo Relacional da Base de Dados 29](#_Toc121869962)

[Comandos SELECT e Respetivos Resultados 29](#_Toc121869963)

[Conclusões 33](#_Toc121869964)

[Bibliografia 34](#_Toc121869965)

# **Introdução**

## **Modelo Relacional e Não Relacional**

Estruturar devidamente a informação é uma condição essencial para o bom desempenho de um qualquer sistema de informação, esteja ele informatizado ou não. Quando a informação está bem estruturada, torna-se fácil e intuitivo consultá-la e assegurar a sua manutenção e integridade.

Ao longo dos anos foram propostas e utilizadas diversas técnicas e linguagens de auxílio à estruturação da informação. Atualmente o Modelo Relacional, divulgado nos anos 80, ainda é a forma privilegiada de estruturação da informação, sendo suportado pela quase totalidade das ferramentas de desenvolvimento de sistemas de informação.

O conceito de Modelo Relacional foi criado por Edgar Codd, matemático nos anos 70, e tornou-se popular desde os anos 80 até ao quotidiano. Atualmente, quase todas as bases de dados existentes são relacionais. Este tipo de modelos permite-nos armazenar dados em tabelas, garantindo a sua integridade através da utilização de chaves primárias e estrangeiras e definindo outras informações *a priori*. [1]

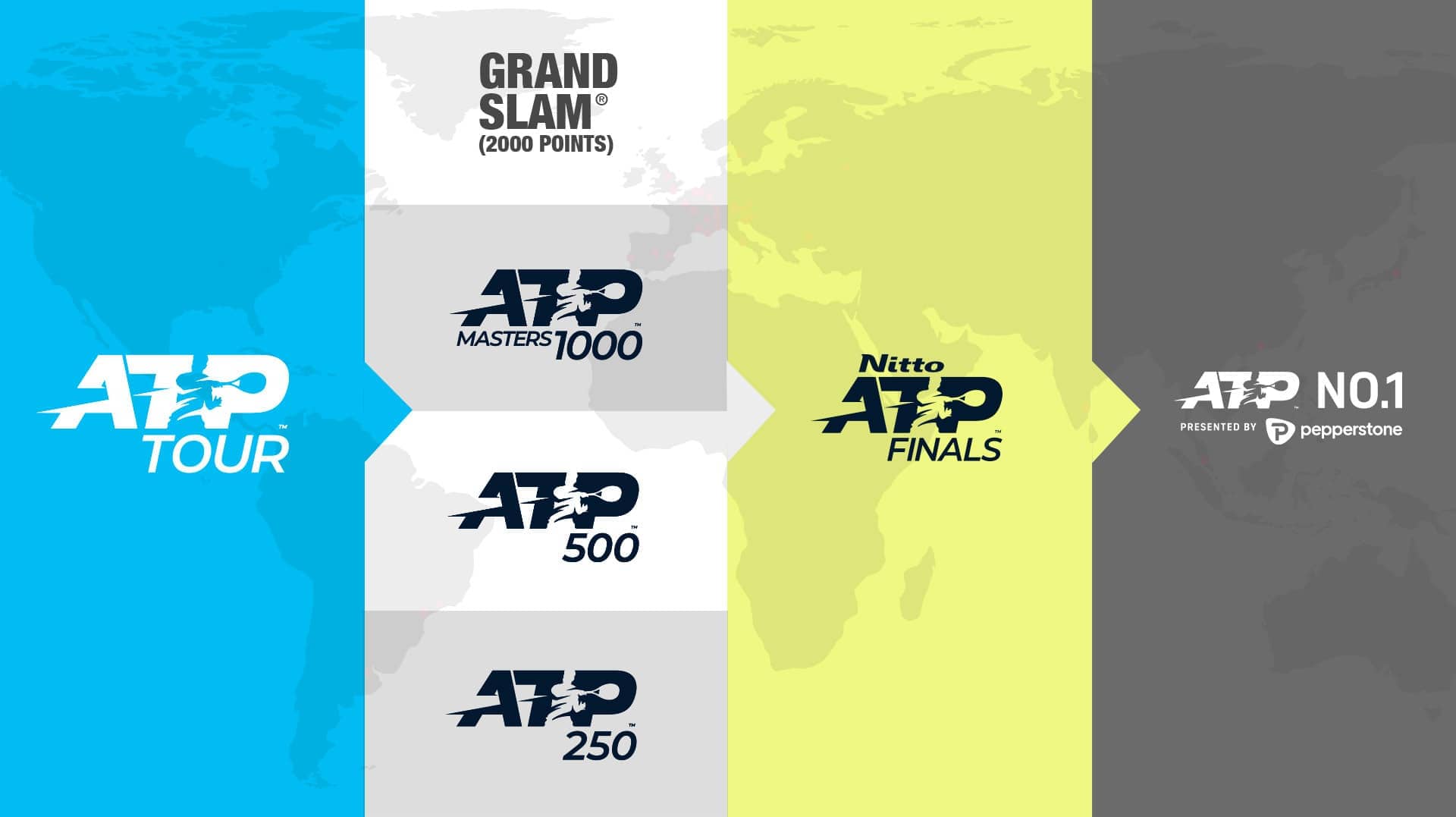
Todavia, nos últimos anos, as bases de dados *NoSQL* têm vindo a aumentar a sua presença nas empresas, mas estão longe de substituir as relacionais. Estes dois tipos de tecnologias são complementares e não mutuamente exclusivos. A vantagem preponderante ao usar Modelos Não Relacionais prende-se com a flexibilidade no seu uso, dado que não necessita de definir informação precedente à criação da base de dados, como o tipo de dados dos campos, chaves primárias, chaves estrangeiras, regras de integridade referencial, entre outras; e lidam satisfatoriamente com enormes quantidades de dados, contrariamente aos Modelos Relacionais. [2]

Neste sentido, foi proposto no âmbito da UC de *Armazenamento para Big Data* inserida na licenciatura de Ciência de Dados, um projeto que congrega ambos os tipos de Modelos, o Relacional e o Não Relacional, permitindo aprofundar e praticar competências no que diz respeito à limpeza e análise de uma base de dados real.

## **Base de Dados ATP Tennis Professionals**

Neste sentido, o *dataset* que iremos trabalhar refere-se aos ATP Players, no qual estão todos os jogos individuais de aproximadamente 10 000 jogadores profissionais de ténis masculinos (*TOP 500* jogadores que jogaram entre 28/03/1973 e 14/02/2022).

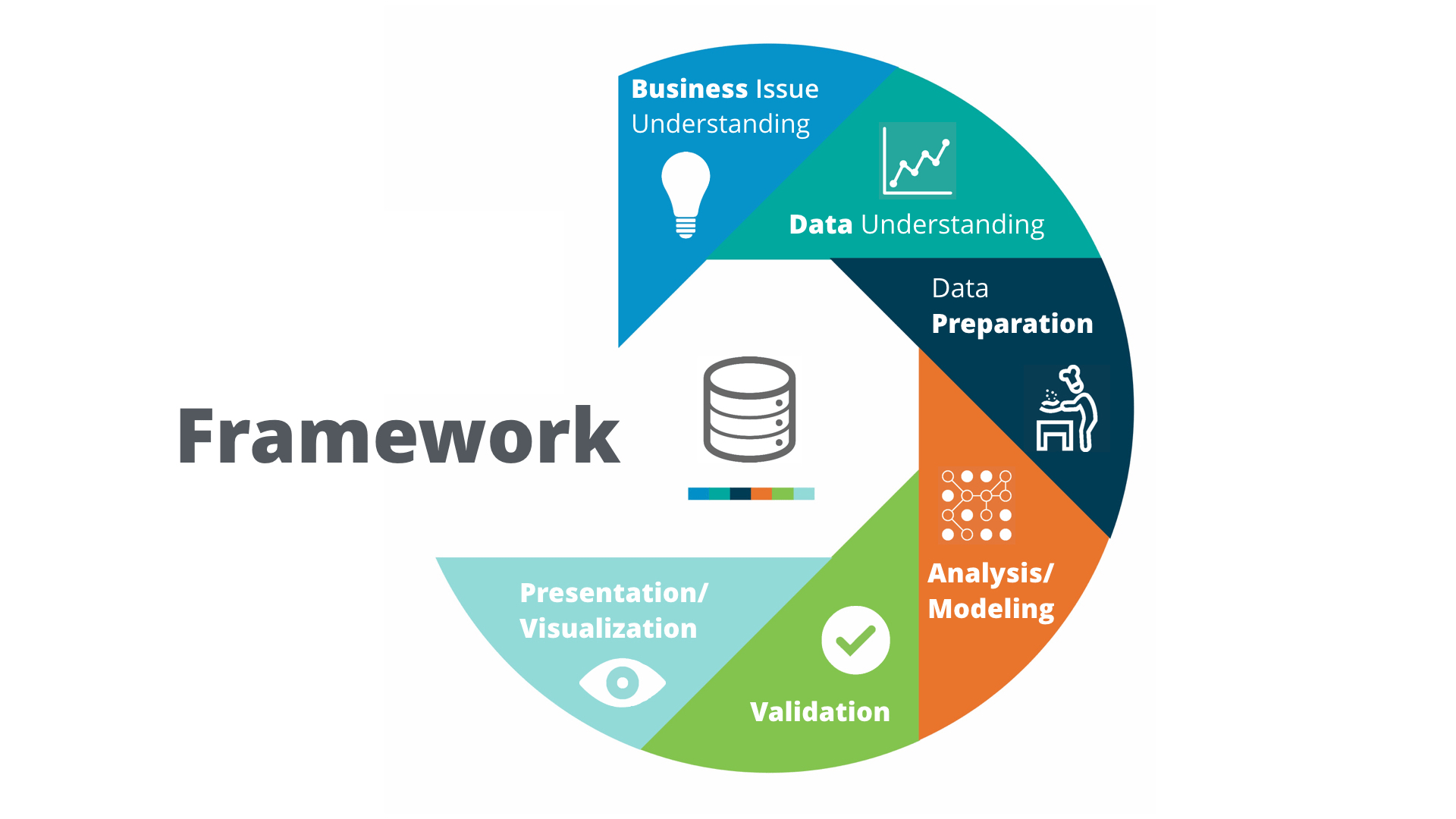
“*Como organismo global de ténis profissional masculino, a missão do ATP é servir ténis. Entretemos mil milhões de fãs globais, mostramos os melhores jogadores do mundo nos mais prestigiados torneios e inspiramos a próxima geração de fãs e jogadores.”* [3]



**Figura 1 |** Diferentes Temporadas até ao Título de *Melhor Tenista Profissional*.

**Fonte:** [3]

Para melhor tirar partido do *dataset* fornecido, decidimos que deveríamos aplicar a metodologia **CRISP-DM**. Esta metodologia é cíclica (**Fig.2**) e baseia-se na exploração e aperfeiçoamento das diferentes etapas, para obter melhores resultados, e assim tomadas de decisão mais eficazes e significativos. Por definição, esta metodologia é formada por seis fases: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation* e *Deployment*. [4]



**Dataset**

**CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining

**Figura 2 |** Ciclo da Metodologia CRISP-DM.

**Fonte:** Adaptado de [5]

# **Metodologia CRISP-DM**

## **1. *Business Understanding***

Primeiramente é imprescindível conhecer o tema da base de dados, de forma a poder inferir estatísticas e conclusões de forma sólida e consistente. Assim sendo, começámos por compreender a história e evolução do ténis profissional até à origem da **ATP (*A****ssociation of* ***T****ennis* ***P****rofessionals***).**

Lançada em 1972 por Jack Kramer, Donald Dell e Cliff Drysdale, para proteger os interesses dos tenistas profissionais, a **ATP** tornou-se o órgão regulador dos circuitos de ténis profissionais masculinos. A designação da associação foi alterada ao longo dos anos, passando por **ATP Tour (**1990**)**, apenas **ATP (**2001**)**, **ATP World Tour (**2009**)**, e mais recentemente voltou a denominar-se por **ATP Tour (**2019**)**, designação que perdura até à atualidade. [6]

O ATP Tour é o circuito mundial de tenistas profissionais masculino, e compreende o ATP Masters 1000, ATP 500 e ATP 250. A **ATP** também supervisiona o ATP Challenger Tour, um nível abaixo do ATP Tour.

Para regular a competição, a associação criou o sistema de classificação de jogadores profissionais, que permite publicar os diferentes Rankings de jogadores profissionais. [7]

Relativamente ao desporto de ténis é importante salientar ainda que cada jogo para além de Ranking, tem também Score. Este depende do número de *sets* em cada jogo, podendo conter mais do que 1 valor correspondente ao Score em cada set. [8]

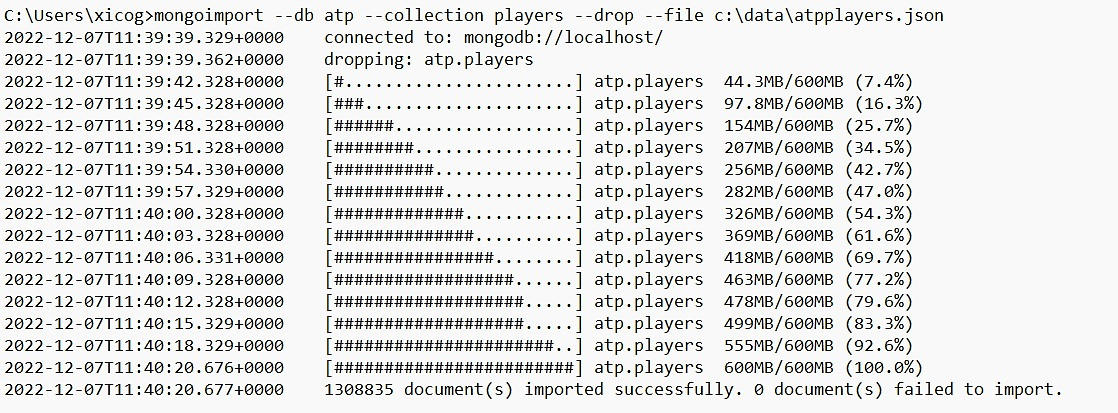
Por fim, e de modo a responder a uma das consultas pedidas no trabalho, é ainda relevante saber a que se refere o termo Grand Slam.

Os torneios de **Grand Slam**, também designados de *Majors*, são os quatro eventos anuais mais importantes do ténis. Nestes, os jogadores têm a oportunidade de receber um maior número de pontos no *ranking* mundial, prémios monetários, atenção do público e da comunicação social. Os quatro torneios que fazem parte do **Grand Slam** são o Australian Open, Roland Garros, Wimbledon e o US Open, que decorrem nesta ordem. [9]

## **2. *Data Understanding***

Depois de conhecermos e enquadrarmos melhor o domínio que iremos trabalhar, vamos perceber as variáveis presentes no *dataset*, visando saber que valores são esperados e o significado dos mesmos.

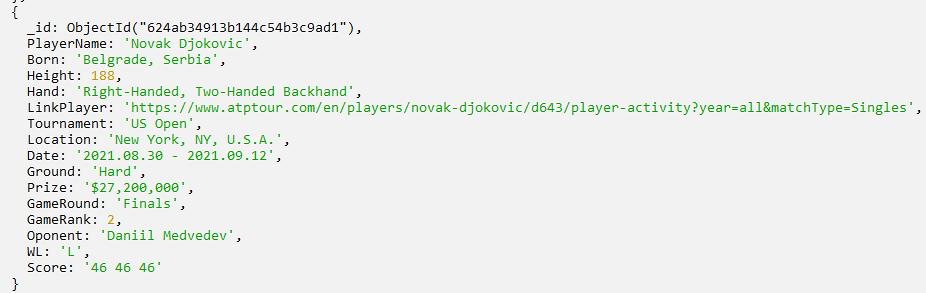
### **Importação da Base de Dados**

Assim sendo, começámos por importar a base de dados em formato .json. Numa linha de comandos, demos início ao servidor de **MongoDB** através do comando mongod e, noutra corremos o comando da **Figura 3**.

**Figura 3 |** *Import* em MongoDB da Base de Dados ATP Players.

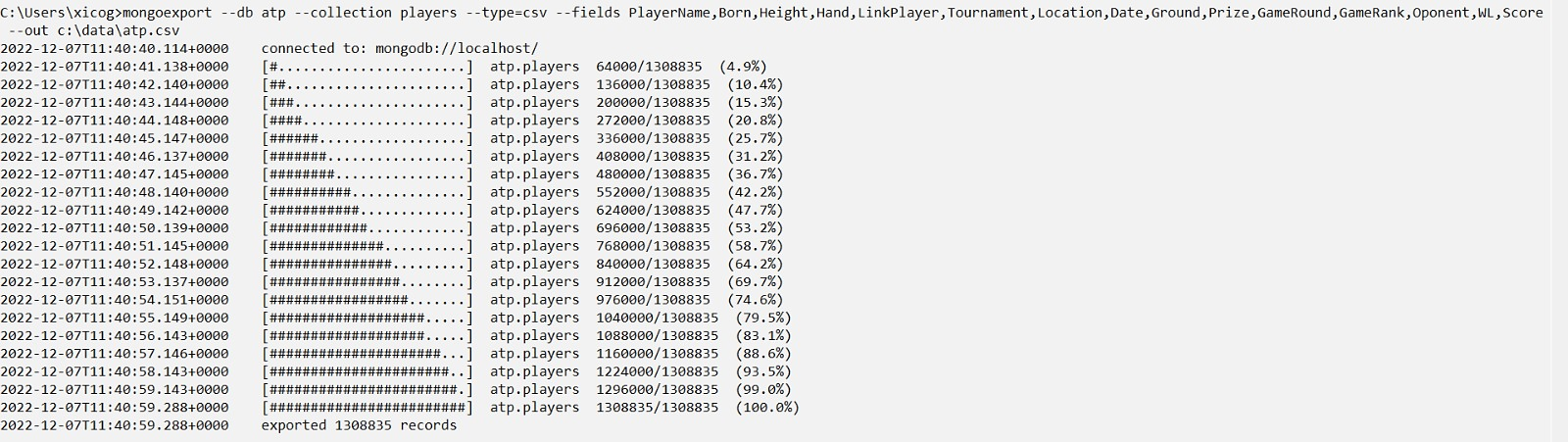
Verificamos que o *dataset* fornecido tem 1 308 835 documentos contidos numa única coleção, players.

De seguida verificámos que informação está em cada documento da coleção players da base de dados atp.



**Figura 4 |** Exemplo de um **Documento** do *dataset* ATP Players.

Através do *output* (**Fig. 4)** observa-se que a base de dados tem 15 variáveis para estudo (excluindo o \_id, que não tem significância para o objetivo do trabalho).

Desde logo optámos por fazer o trabalho de limpeza em SQL, pelo que exportámos a base de dados para o formato .csv (**Fig.5**), após verificarmos que não haveria nenhum caracter que pudesse interferir no processo de importação em SQL.

**Figura 5 |** *Export* em MongoDB da Base de Dados ATP Players.

Depois de exportada a base de dados atp com os 1 308 835 documentos, convertidas em linhas de .csv procedemos à sua implementação no SQL como uma única tabela, atp, com os comandos que podem ser consultados no ***script***.

Ponderámos exportar do MongoDB em ficheiros distintos, correspondentes às tabelas do modelorelacional, porém optámos por não o fazer, uma vez que não fizemos qualquer limpeza prévia, pelo que poderá existir duplicados entre outros problemas que ao estar numa só tabela serão solucionados para todos os campos de uma só vez, de forma mais rápida e eficaz.

Assim, a importação dos dados resultou na tabela da **Figura 6.**

Uma imagem com texto, captura de ecrã, interior, vários

Descrição gerada automaticamente

**Figura 6 |** Excerto da Tabela ATP criada em SQL.

Posteriormente, fomos conhecer as variáveis em estudo, enquadrando-as e procurando saber o seu significado.

### **Variáveis em Estudo**

PlayerName **–** Indica o nome do principal jogador da partida.

Born **–** Teoricamente, devolve a cidade e/ou o país de onde o jogador em causa é natural.

Height **–** Demonstra a altura dos jogadores, em *cm*.

Hand **–** Indica qual a mão possante do jogador, Right-Handed (destro), Left-Handed (canhoto) ou Ambidextrous (ambidestro); e que mãos utiliza para executar o serviço, One-Handed Backhand, Two-Handed Backhand ou Unknown Backhand).

LinkPlayer **–** Fornece o *link* do perfil do jogador no website do ATP Tour.

Tournament **–** Designa o nome do torneio.

Location **–** Revela a localização do torneio.

Date **–** Indica o intervalo de datas em que se realizou o torneio em questão.

Ground **–** Informa em que terreno é jogado o torneio.

Apenas pode apresentar os valores Hard, Grass, Clay ou Carpet.

O tipo de campo Hardé feito de materiais uniformes e rígidos com uma camada superficial de acrílico, usado nos torneios ***US Open*** e ***Australian Open*** de *Grand Slam*.

O tipo de campo Grasssão terrenos de relva natural, sendo o mais emblemático deste estilo o torneio de ***Wimbledon***.

O tipo de campo Clayé feito de pedaços muito pequenos de pedra, tijolo, xisto ou outro agregado mineral não ligado, dependendo do torneio, podendo ter a típica cor vermelha ou verde. O ***Roland Garros*** é o único torneio de *Grand Slam* a ter este terreno.

E, por último, o tipo de campo Carpet consiste em materiais têxteis ou poliméricos fornecidos em rolos ou folhas de produto acabado. A versão ao ar livre mais comum deste tipo é a relva artificial com preenchimento de areia.

Prize **–** Revela o valor do prémio, na moeda do país da realização do torneio.

GameRound **–** Permite saber em que fase do torneio está a ser disputado o jogo em questão.

A variável está organizada em qualificação e eliminatórias.

A escala de qualificação está compreendida entre 1st Round Qualifying e 3rd Round Qualifying e a escala das eliminatórias entre Finals e Round of 128. Existe ainda a Robin Round, que é um formato das eliminatórias, onde os jogadores são separados em grupos e jogam todos contra todos, dentro do grupo, de forma a encontrar os jogadores das semifinais. [7]

GameRank **–** Mostra qual o *rank* do jogador Opponent.

Uma imagem com texto, interior, captura de ecrã, branco

Descrição gerada automaticamenteEste *rank* é concretizado através dos pontos obtidos, conforme a **Figura 7,** pelo que podemos verificar que dependendo do torneio o jogador ganha quantidades diferentes de pontos.

**Figura 7 |** Tabela de Atribuição de Pontos conforme as regras da ATP.

Oponent **–** Indica o nome do jogador adversário.

Este jogador não tem de necessariamente de ter um Player\_Name e LinkName associado nesta base de dados, pois estes jogos são referentes ao *TOP 500* jogadores masculinos que jogaram entre 28/03/1973 e 14/02/2022, logo há casos de não correspondências.

WL **–** Revela o resultado do jogo, W (***Win*** - venceu) ou L (***Lost*** - perdeu).

Score **–** Apresenta a distribuição dos resultados dos *sets*.

São fornecidos números de 2 algarismos, sendo esse valor, o número de *sets* existentes no jogo em questão.

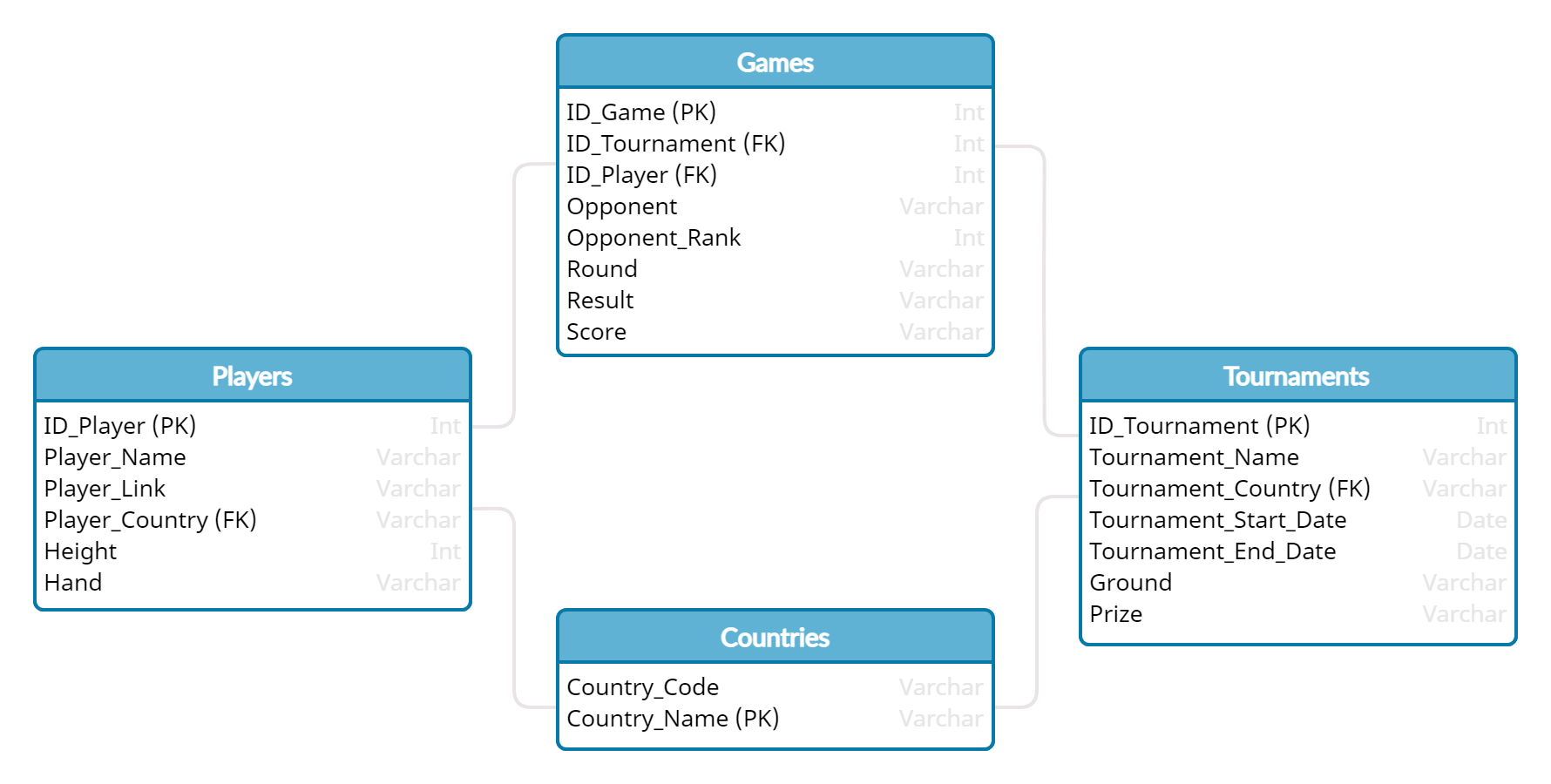
O primeiro algarismo de cada número apresentado indica quantos jogos venceu o jogador principal naquele *set*, já o segundo algarismo indica quantos jogos venceu o jogador adversário.

## **3. *Data Preparation***

Na fase de ***Data Preparation*** estudámos os dados e arranjámo-los para a fase seguinte.

### **Desenho da Base de Dados Relacional**

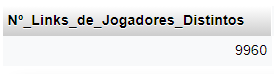
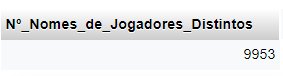
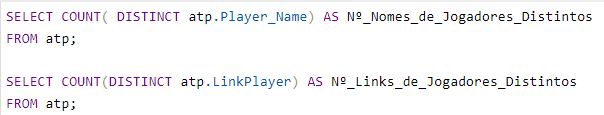
Assim sendo, tratámos e limpámos as variáveis individualmente. Este procedimento visa criar uma **Base de Dados Relacional**. Assim, esboçamos já uma base de dados relacional possível (**Fig.8**).



**Figura 8 |** Esboço da estrutura do **Modelo Relacional**.

### **Variáveis** Player\_Name **e** LinkPlayer

Relativamente às variáveis Player\_Name e LinkPlayer constatámos que existem 9953 Player\_Name distintos, todavia existem 9960 LinkPlayer distintos, logo concluímos que haveria jogadores com o mesmo nome (**Fig.9**), pelo que esta não poderia ser uma das variáveis utilizadas para selecionar diferentes jogadores, ou até, posteriormente, servir de chave primária (**PK**) na tabela Players.

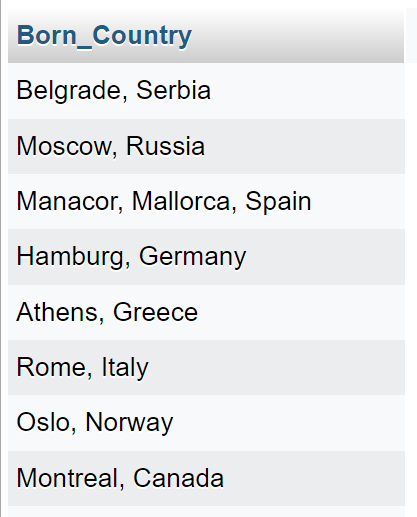


**Figura 9 |** Contagem dos Player\_Name e LinkPlayer distintos.

### **Variável** Born\_Country

Analisando a variável Born\_Country, verificámos que era necessário uniformizar este campo para ser possível criar a chave estrangeira (**FK**) para a tabela Countries, construída tendo por base o ficheiro countries\_net.csvfornecido, permitindo validar os países e efetuar as estatísticas finais.

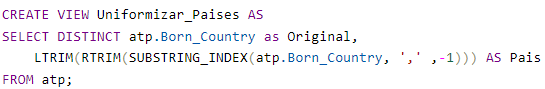
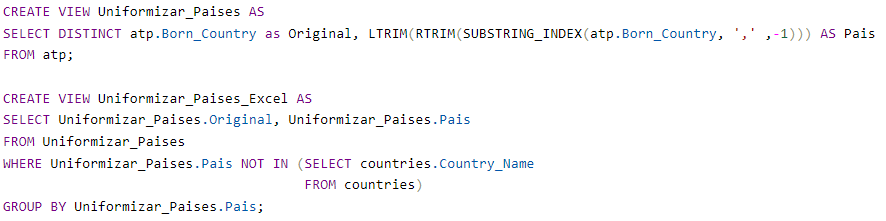
Começámos por verificar quantos valores distintos existem na variável Born\_Country e como é que estes estão registados (**Fig.10**).



**Figura 10 |** Resultados dos Valores Distintos na Variável Born\_Country.

Assim, verificámos que na maioria dos casos o campo é preenchido como Cidade, País, havendo exceções, sendo necessário extrair os países e atualizá-los de modo a padronizar consoante as designações da tabela Countries.

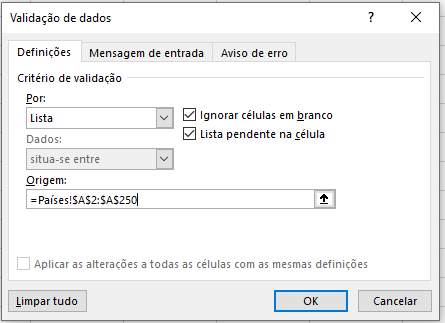
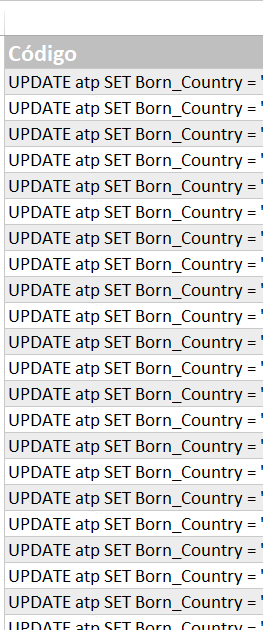
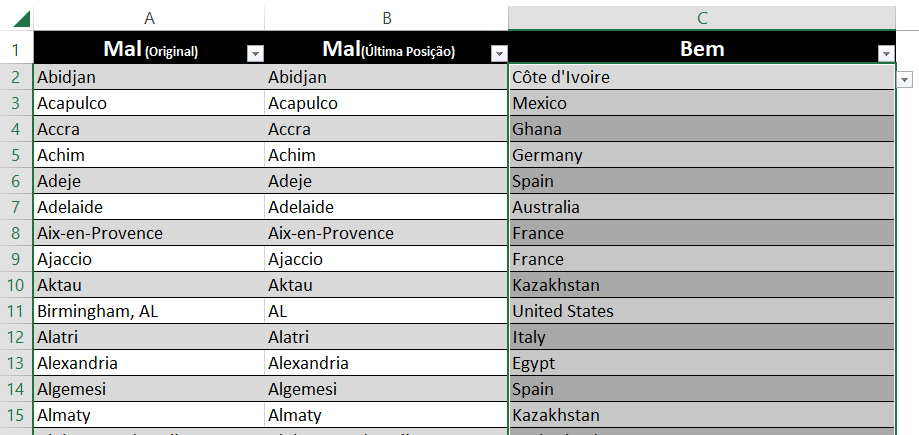
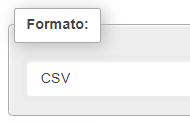
De modo a solucionar este problema e uma vez que não encontramos nenhum padrão que facilitasse a alteração de todas as designações, criámos duas VIEWs (**Fig.11**) na qual resulta 648 designações discordantes com a tabela Countries - cidades em vez de países, países com erros ortográficos ou escritos de forma díspar do que está definido.



**Figura 11 |** Views Uniformizar\_Paises e Uniformizar\_Paises\_Excel.

Nestas VIEWs utilizámos a função SUBSTRING\_INDEX(), para isolar apenas a última posição, que, em certos casos é o país, tal como referido anteriormente, utilizando como delimitador a ,. Adicionalmente as funções RTRIM() e LTRIM() com o prepósito de remover espaços indesejados à direita e à esquerda de uma *string*, respetivamente.

De seguida, exportámos os resultados e auxiliamo-nos ao software *Excel* para corrigir as designações discordantes, tal como esquematizado na **Figura 12**.



**Figura 12 |** Esquematização Sucinta do Processo de Uniformização dos Países em *Excel.*

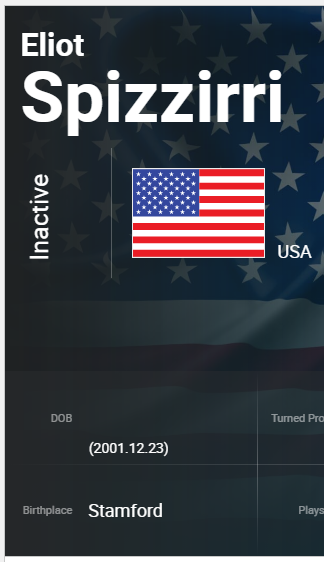
Depois de todas as designações estarem corrigidas, criámos o código de UPDATE, tal como ilustrado de seguida

UPDATE atp SET Born\_Country = "Côte d'Ivoire" WHERE LTRIM(RTRIM(Born\_Country)) = "Abidjan";

#### **Casos Particulares**

Todavia, certos valores, que contêm caracteres especiais, �, que o SQLnão reconhece, tivemos de adaptar o código com a LIKE tal como:

UPDATE atp SET Born\_Country = "France" WHERE Born\_Country LIKE "%ret";



Adicionalmente a estes casos houve necessidade de pesquisar os jogadores pelos links de modo a confirmar o país de origem, como por exemplo na cidade de Stamford que existe quer nos EUA, quer no Reino Unido. Neste exemplo, fomos pesquisar pelo link do jogador e verificar que país era atribuído ao mesmo. (**Fig. 13**)

**Figura 13 |** Perfil do jogador Eliot Spizzirri *.*

**Fonte:** *https://www.atptour.com/en/players/eliot-spizzirri/s0ja/overview*

Tivemos ainda de efetuar decisões, para deliberar a que país pertence certa cidade, como é o caso da Czechoslovakia e da Yuguslavia, para as quais optámos por uniformizar em Slovakia e Serbia, respetivamente.

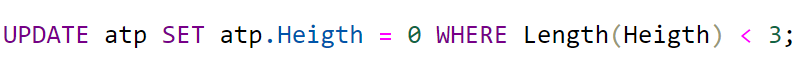
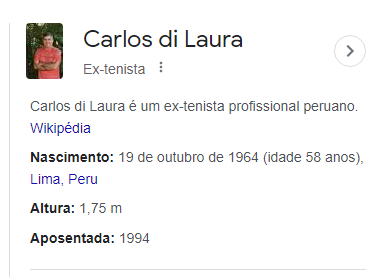
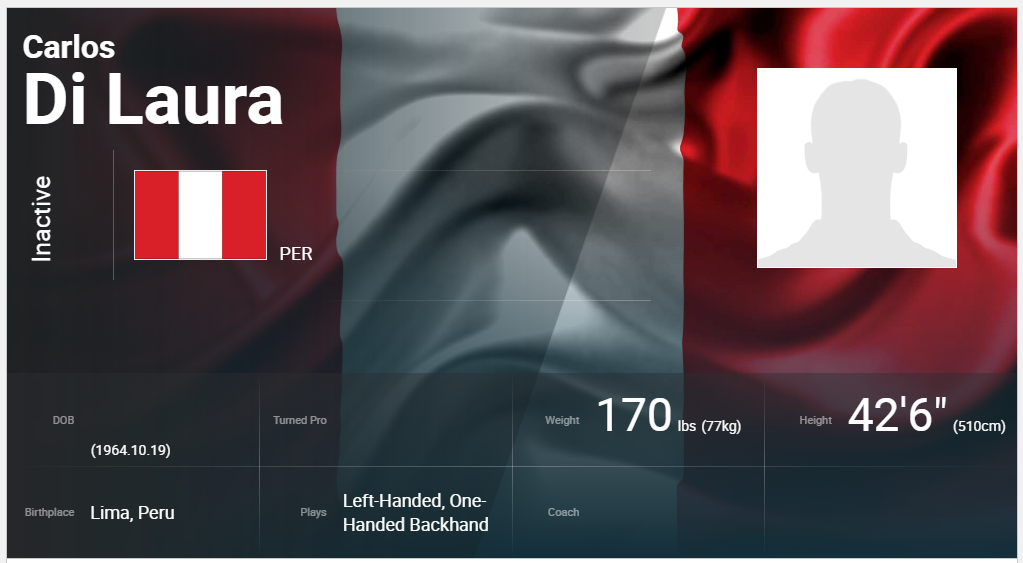
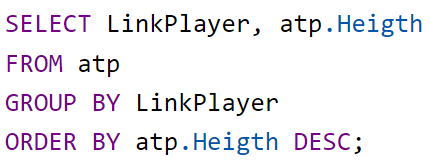
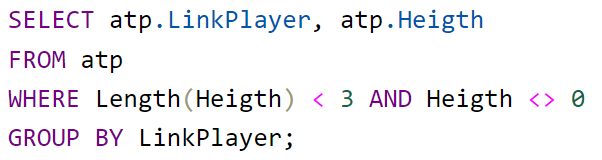
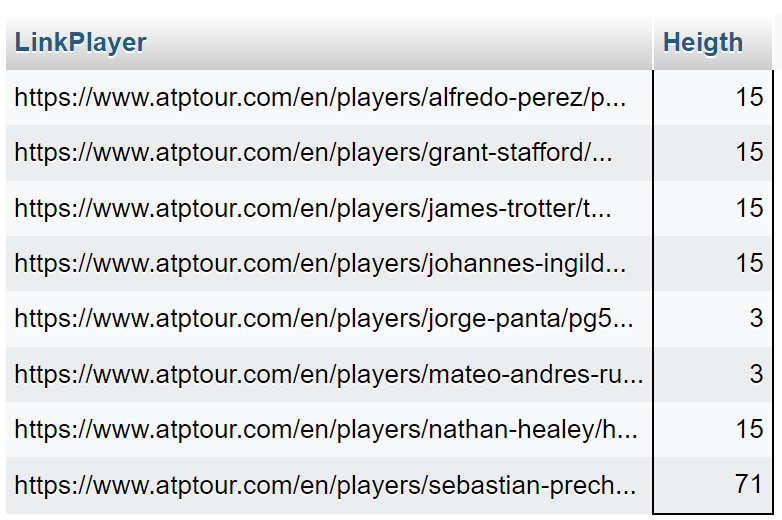
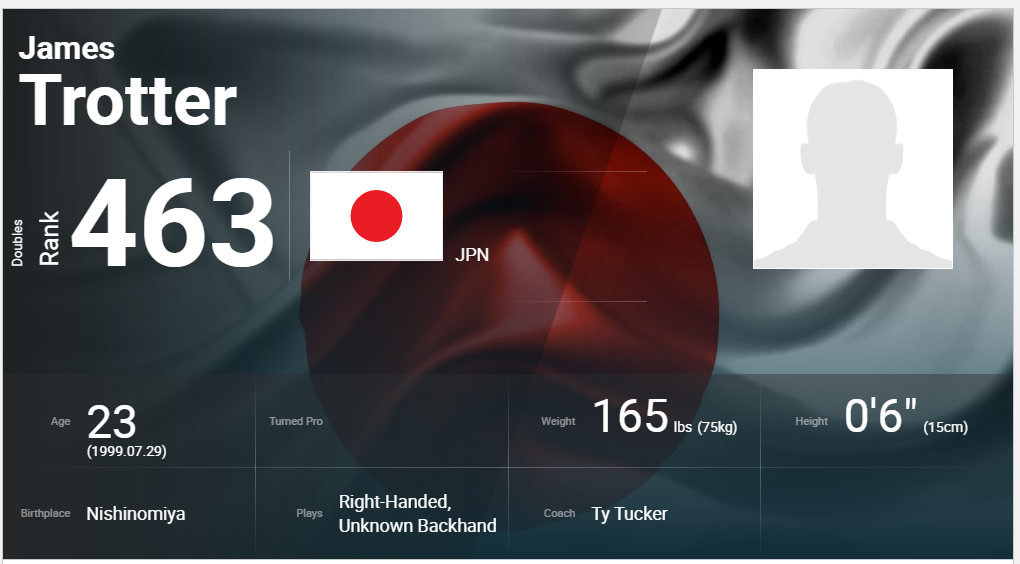
No final desta homogeneização tivemos **312 051 linhas afetadas** (23,84%).

### **Variável** Height

Quanto à variável Height, verificámos em que intervalo se encontrava a altura dos jogadores, de modo a encontrar irregularidades.

Assim, percebemos que existiam alturas inferiores a 1 metro, ou seja, jogadores com menos de 100 centímetros de altura e um jogador com altura de 510 centímetros.

Considerámos estes dados inverídicos e, para tentar ao máximo não eliminar dados, priorizámos a substituição pelos valores corretos, que apenas foi possível no caso dos 510cm, pelo que os substituímos por zero nos restantes casos (**Fig. 14**).



**Figura 14 |** Irregularidades na variável Height.

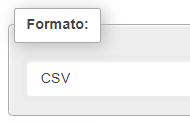
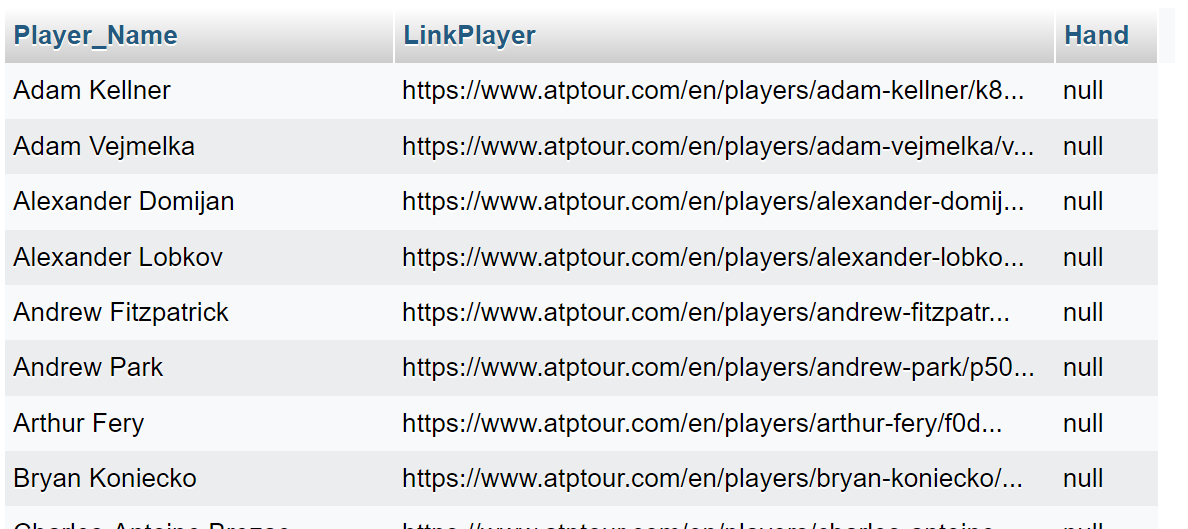
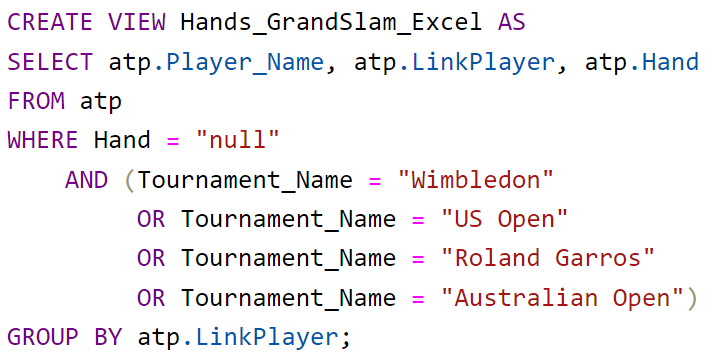
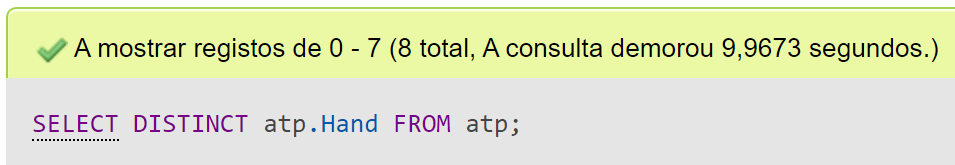
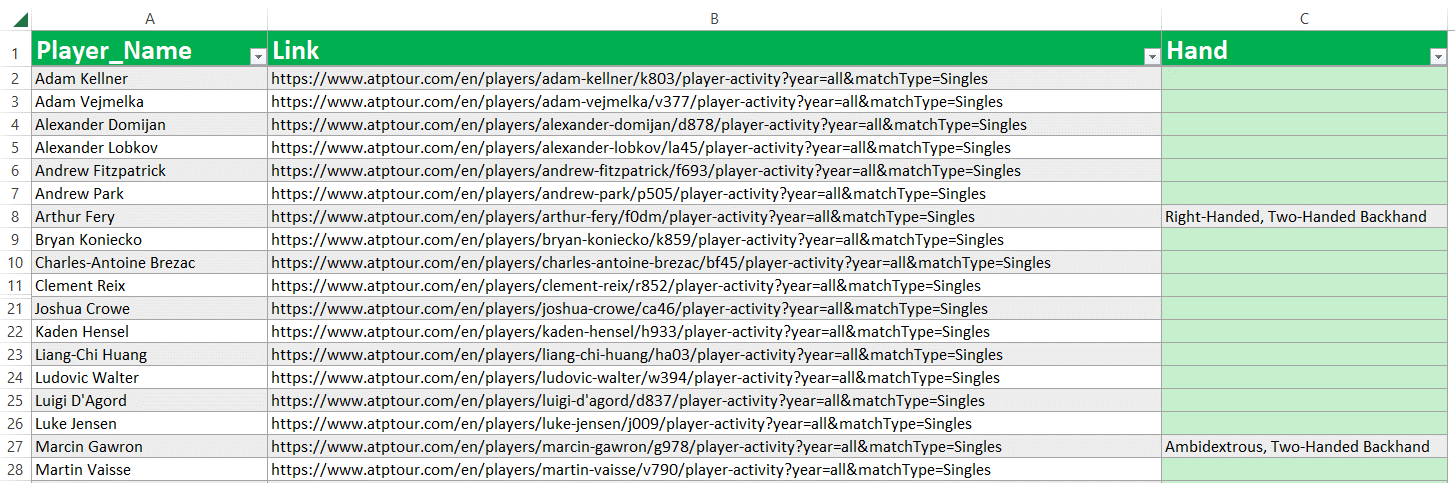
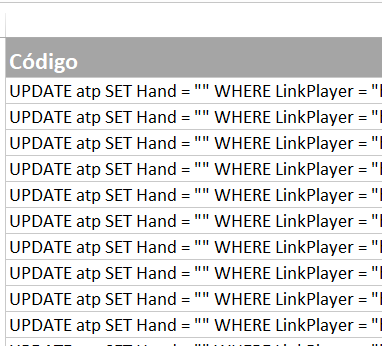
**Fonte:** *https://www.atptour.com/en/players/*

Neste processo foram afetadas **1491 linhas** (0,11%).

### **Variável** Hand

Relativamente à variável Hand, verificámos que os valores categóricos que pode tomar eram cumpridos, sendo que há 191 917 null.

Uma vez que iremos precisar desta variável para uma das estatísticas relacionada com os jogadores esquerdinos que participaram nos torneios de *Grand Slam* e, destes, apenas 49 tinham null na variável Hand, então tentámos preencher os valores omissos, utilizando o mesmo processo da variável Born\_Country, com auxílio ao *Excel.* (**Fig.15**)



**Figura 15 |** ViewHand\_GrandSlam\_Excel e processo em *Excel.*

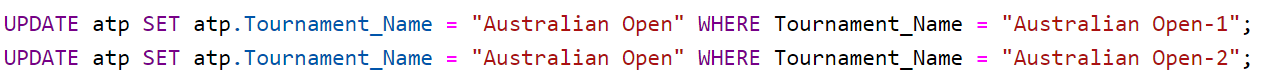
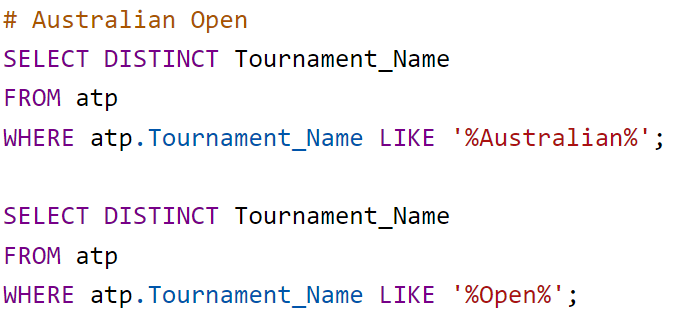
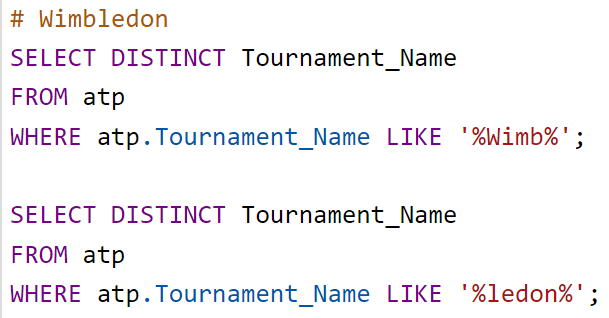
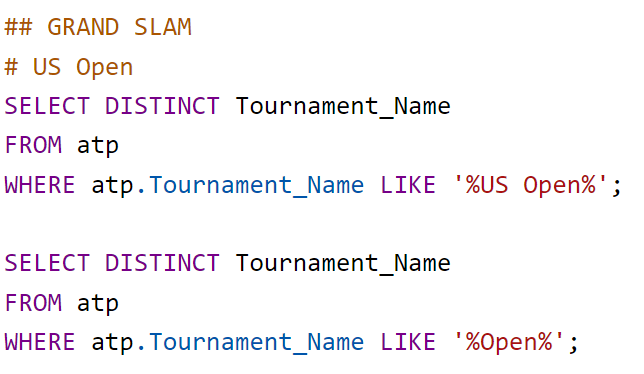
Apesar do esforço em tentar recuperar os valores para estes 49 jogadores, apenas foi possível adicionar este dado a 3 jogadores, cuja informação se encontrava no site da ATP Tour.

### **Variável** Tournament\_Name

Cada torneio tem o seu nome, sendo este representado na variável Tournament\_Name.

Visto que será necessário mais adiante usarmos esta variável para identificar os torneios do *Grand Slam* procurámos termos relacionados com estes torneios para unificar, caso haja.

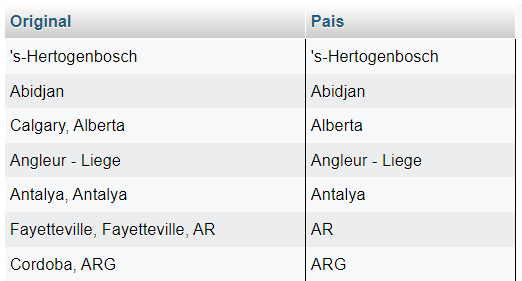
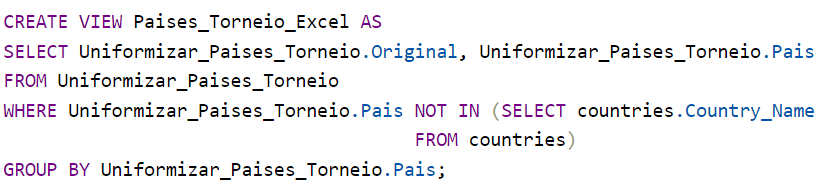
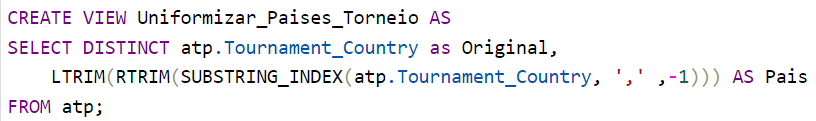
Como apresentado na **Figura 16**, o Australian Open estava escrito de maneiras diferentes então uniformizámo-lo.



**Figura 16 |** Uniformização da variável Tournament\_Name.

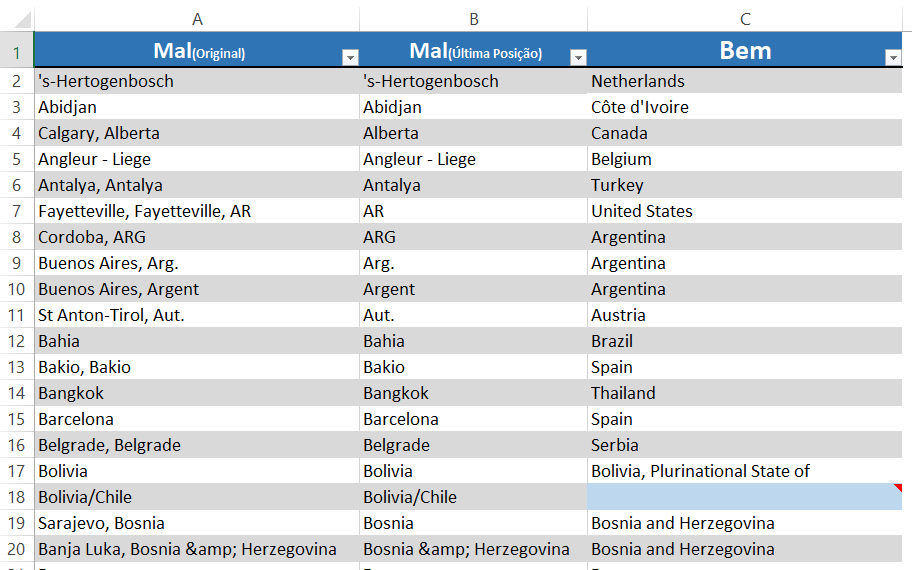
### **Variável** Tournament\_Country

Seguindo o mesmo raciocínio da variável Born\_Country, uniformizámos os países da variável Tournament\_Country, começando por perceber as incoerências (**Fig. 17**), criando as VIEW**s** e exportando para *Excel* (**Fig. 18**).



**Figura 17 |** Views Uniformizar\_Paises\_Torneio e Paises\_Torneio\_Excel.

Neste caso há 211 valores discordantes da designação da tabela Countries.

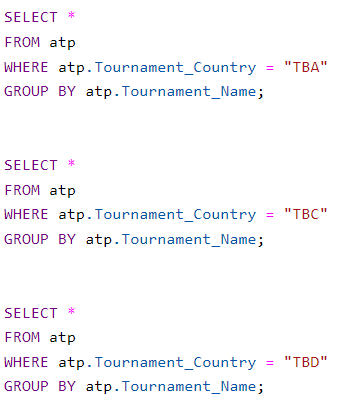


**Figura 18 |** Uniformização da variável Tournament\_Country em *Excel*.

#### **Casos Particulares**

Tal como ocorreu na variável Born\_Country, tivemos de tomar decisões e procurar mais informação para completar todas as designações. No que diz respeito a decisões que efetuámos, salientamos a designação Bolívia/Chile, na qual optámos pela Bolívia.

Adicionalmente, é relevante salientar os casos que encontrámos as terminologias TBA (To Be Announced), TBC (To Be Confirmed) e TBD (To Be Determined), entre os países de torneios que ocorreram. (**Fig. 19**)



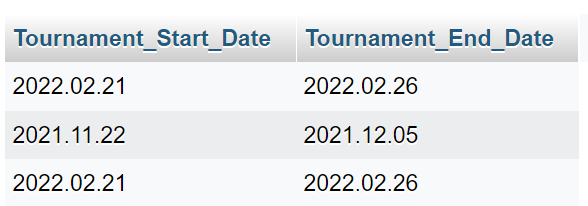
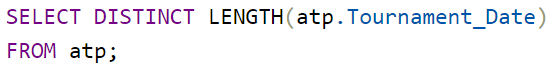
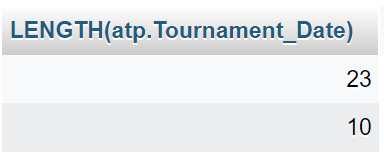
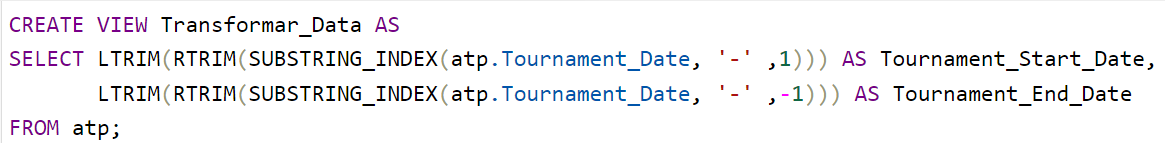
**Figura 19 |** Tournament\_Country com TBA, TBC e TBD.

Tal como visível nos *outputs* da **Figura 19**, pelo nome de cada torneio é deduzível a sua localização. Desta forma, alterámos todas as ocorrências com essas terminologias para o respetivo país em que ocorreu o torneio.

No final desta uniformização tivemos **300 465 linhas afetadas** (22,96%).

### **Variável** Tournament\_Date

Quanto à variável Tournament\_Date analisámos qual era o seu formato de modo a puder atribuir-lhe o tipo DATE aquando da criação das tabelas do modelo relacional, possibilitando assim a integridade desta variável (**Fig.20**).



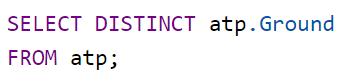
**Figura 20 |** Transformação da variável Tournament\_Date.

Verificámos que esta apresenta o formato aaaa.mm.dd - aaaa.mm.dd (o equivalente a 23 caracteres) caso o torneio seja de vários dias, ou aaaa.mm.dd (o equivalente a 10 caracteres) caso o torneio decorra em apenas um dia.

Por isso, optámos por dividir as datas em duas colunas, uma com o dia inicial do torneio, Tournament\_Start\_Date, e outra com o dia final do torneio, Tournament\_End\_Date, permitindo assim que posteriormente conseguiremos por esta variável do tipo DATE.

### **Variável** Ground

Uma vez que a variável Ground representa o tipo de terreno em que o campo do torneio de ténis é jogado, sendo que apenas pode assumir quatro valores diferentes, Hard, Grass, Clay ou Carpet, confirmámos se apenas existiam estes valores nesta variável, o que se verificou (**Fig.21**).



**Figura 21 |** Valores distintos da variável Ground.

### **Variável** Prize

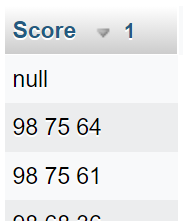
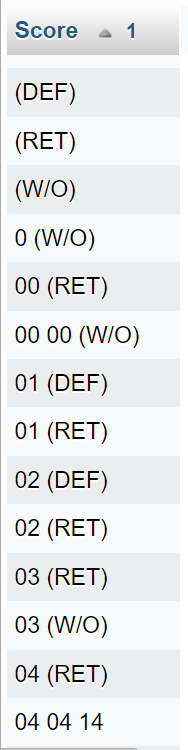
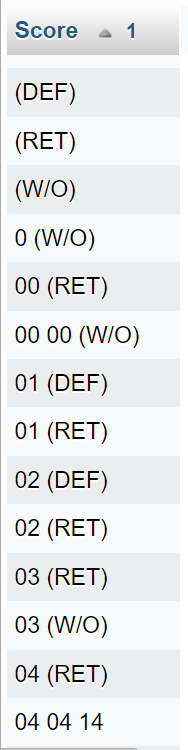
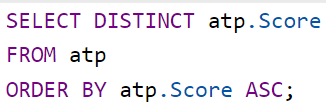
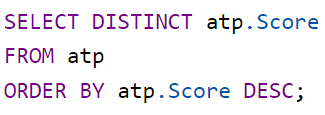
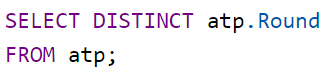
Na variável Prize verificámos que o prémio em dinheiro que o jogador campeão recebe contém caracter especial �.

Como os torneios estão espalhados por vários países, a moeda em que se recebe o prémio também difere, desde dólares a euros.

Logo, para uniformizar os prémios numa moeda teria de ser feito em conta o ano em que o prémio foi dado e a inflação da moeda atual. Devido a esta irregularidade nos dados e visto que é uma variável que não vamos usar para esta análise, decidimos não alterar.

### **Variáveis** Round**,** Opponent**,** Opponent\_Rank**,** Result **e** Score

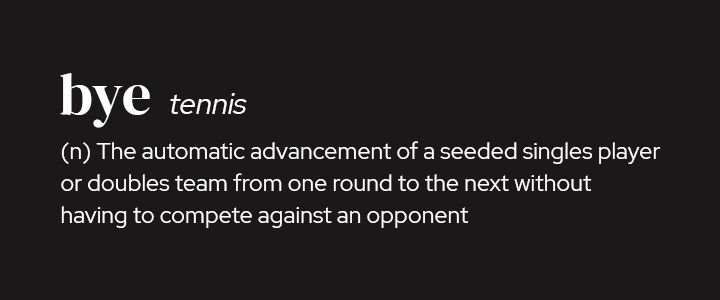
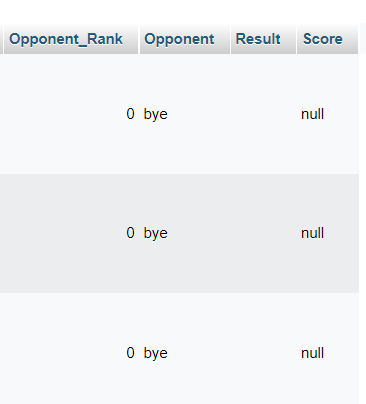
Por fim, nas variáveis Round, Opponent, Opponent\_Rank, Result e Score, verificámos os valores que poderiam assumir (**Fig.22**), sendo que, não houve nenhuma alteração a fazer, uma vez que não havia irregularidades.



**Figura 22 |** Valores distintos das variáveis Round, Score e Result.

É apenas de referir que há um Opponent com valor bye, que tem um significado característico em ténis, pois referência jogos cujo jogador passa para a ronda seguinte sem necessitar de jogar aquele jogo (**Fig.23**).

Logo, nestes casos, as restantes variáveis relacionadas com o jogo são atribuídos NULLs.



**Figura 23 |** Opponent com o termo bye e o seu significado.

**Fonte:** [10]

Por fim, além deste termo destacamos ainda o (W/o) de *without* que denota jogos sem classificação atribuída.

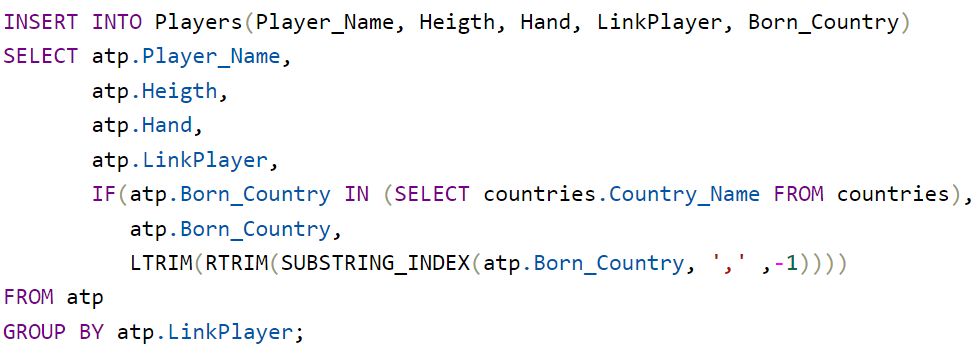
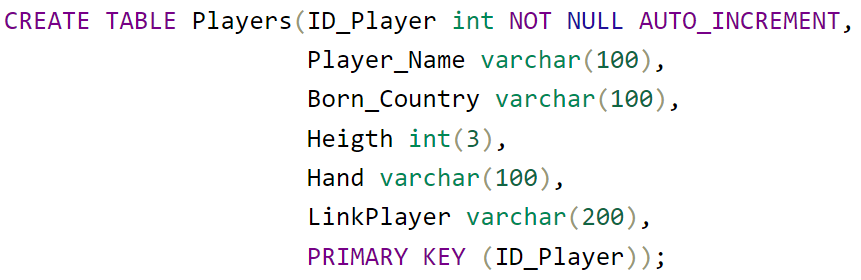
## **4. *Modeling* | *Evaluation* | *Deployment***

Após uniformizarmos todas as variáveis, as últimas fases do **CRISP-DM** adaptadas a este projeto consistem na construção de uma **Base de Dados Relacional** em SQL, garantindo todas as restrições de integridade e na apresentação do resultado das *queries* solicitadas.

### **Tabela** Players

Tal como se tinha idealizado no esboço do *Diagrama do Modelo Relacional* (**Fig.8**) para o *dataset* ATP Tour, criámos a tabela Players, que contém toda a informação pessoal acerca de todos os jogadores referenciados nesta base de dados.

Inicialmente, pensámos e preenchemos a tabela apenas com os jogadores da coluna Player\_Name, sendo que os agrupámos pelo Link\_Player, pois sabemos previamente que há jogadores com o mesmo nome, tal como já mencionado anteriormente, pelo que iriamos perder jogadores ao agrupá-los pelo nome. Para ser mais eficiente e rápido o preenchimento, criámos índices na tabela atp.

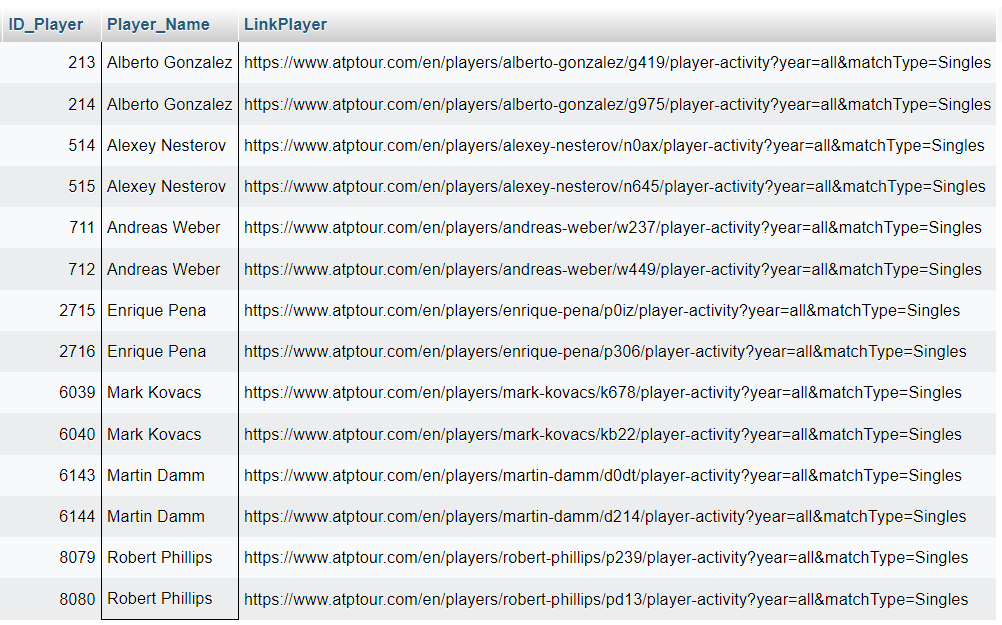
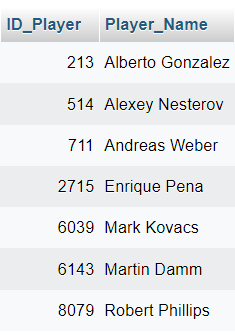
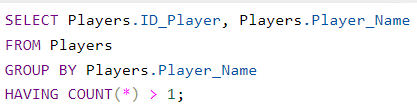


**Figura 24 |** Criação da Tabela Players e inserção dos jogadores.

Neste processo importámos 9960 linhas, o correspondente a jogadores distintos.

Destacamos a utilização da função IF, do RTRIM e do LTRIM, dado que não foram lecionados. O IF permitiu atribuir à variável Born\_Country a correta designação, uma vez que, em alguns casos, a condição em que podemos encontrar a variável é **[Cidade, País]**, e o que pretendemos é apenas o país. Já o RTRIM e o LTRIM removem espaços à direita e à esquerda de uma *string*, respetivamente.

Nesta etapa verificámos desde logo o pressuposto de haver jogadores com o mesmo Player\_Name e LinkPlayer distinto, ocorrendo com **14 jogadores**, tal como corroborado na **Figura 25**.



**Figura 25 |** Jogadores com o mesmo nome.

Por este motivo, o Player\_Name não pode ser a Chave Primária (**PK**) da tabela, pelo que criámos o ID.

Todavia, entendemos mais adiante, tal como será clarificado, que iriamos precisar que os jogadores Opponents tivessem um ID, logo considerámos que seria adequado aproveitar esta tabela, uma vez que a maioria dos jogadores oponentes não pertencem ao *TOP 500* dos anos em estudo.

Assim, adicionámos todos os jogadores oponentes (**Fig.26**), garantindo que não estavam já incluídos na lista, embora todas as restantes informações relativas a estes jogadores não estejam contidas no *dataset*, pelo que ficaram como NULL.

Salientamos ainda que ao contemplarmos todos os oponentes, foi também inserido o termo bye como sendo um jogador, pelo que será posteriormente eliminado.

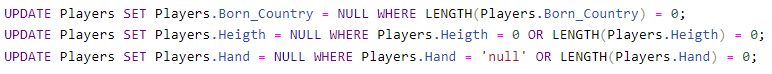
Uma imagem com texto, laranja

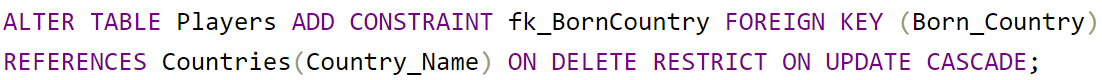
Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

**Figura 26 |** Inserção dos jogadores oponentes na tabla Players.

Findámos esta etapa totalizando 22 689 jogadores distintos.

Por fim, uniformizámos os valores omissos das várias variáveis na tabela Players, substituindo por NULL, e criámos a chave estrangeira (**FK**) para a variável Born\_Country (**Fig. 27**).



**Figura 27 |** Uniformização dos Valores Omissos da Tabela Players e criação da FK.

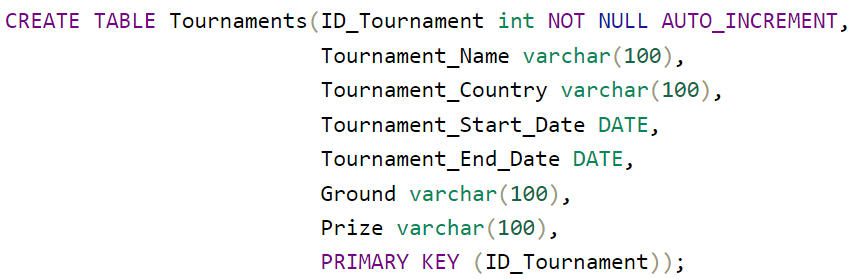
### **Tabela** Tournaments

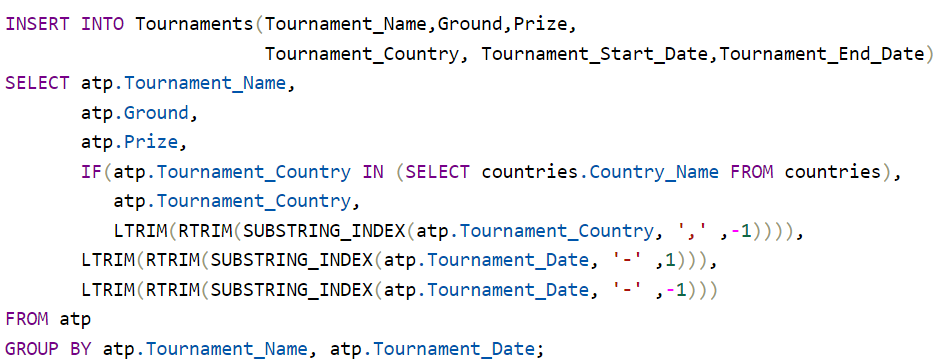
Uma vez mais, como idealizado no esboço do *Diagrama do Modelo Relacional* (**Fig.8**) para o *dataset* ATP Tour, criámos a tabela Tournaments, que contém toda a informação relativa aos torneios desta base de dados.

Tendo em base essa finalidade, começámos por criar a tabela Tournaments (**Fig.28**) e adicionámos as variáveis referentes aos torneios. Nestas, fizemos logo a divisão da data, tal como referido na limpeza das variáveis.

Para identificar jogos distintos, utilizamos a combinação de Tournament\_Name, Tournamente\_Date no GROUP BY.

Não havendo algo unívoco que identificasse eficientemente cada linha da tabela, uma vez que o nome e data do torneio seria pouco benéfico para o cumprimento dos objetivos do trabalho, optámos, uma vez mais, por criar a variável ID\_Tournament para Chave Primária (**PK**).





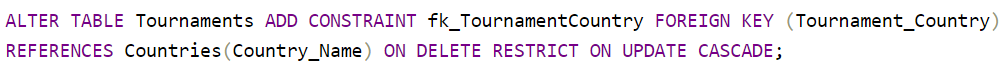
Uma imagem com texto, laranja

Descrição gerada automaticamente

**Figura 28 |** Criação da Tabela Tournament e Inserção das Variáveis.

Totalizámos 22 233 torneiros distintos.

Por fim, foi criada a Chave Estrangeira (**FK**) na variável Tournament\_Country, da mesma forma e com o mesmo propósito da criada na tabela Players; e uniformizámos os valores omissos por NULL (**Fig.29**).



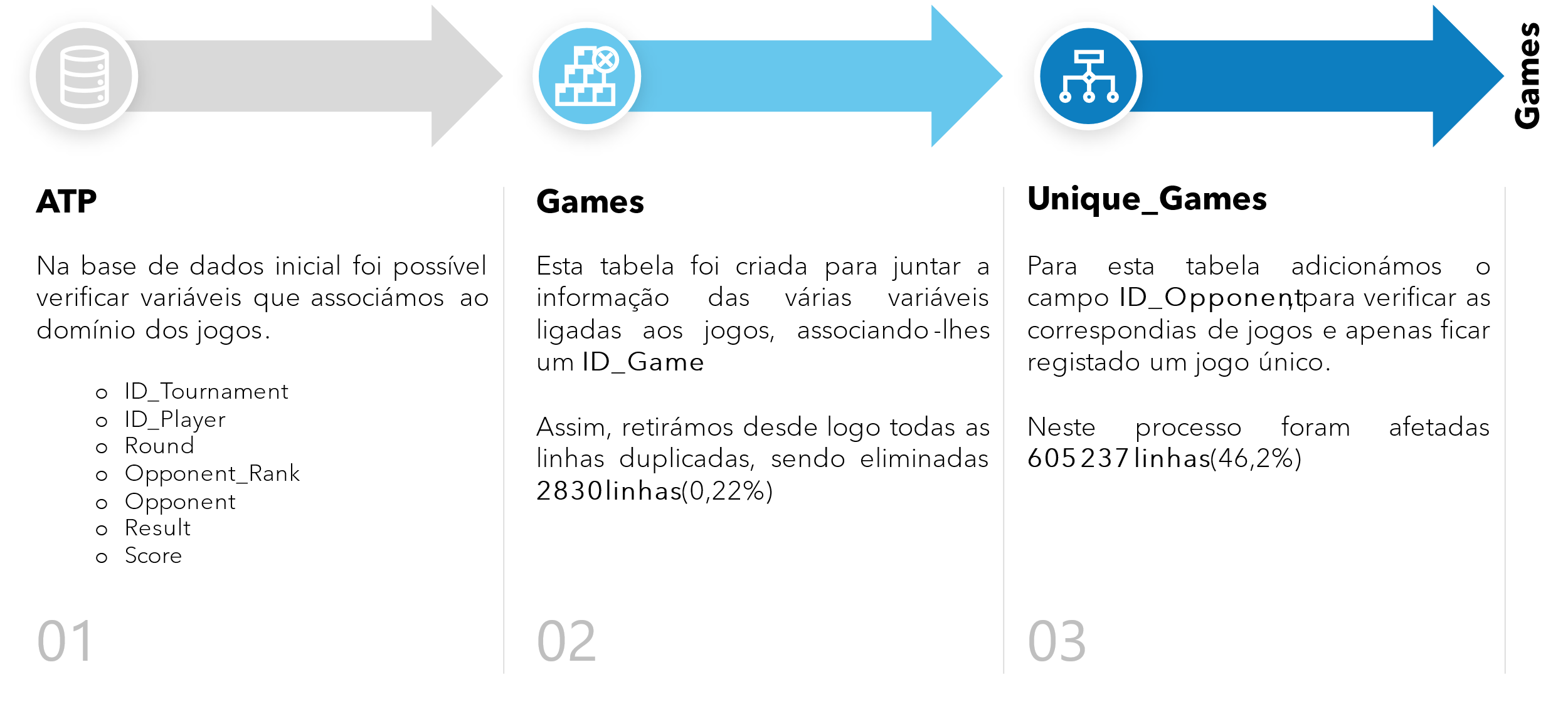
**Figura 29 |** Criação da FK da variável Tournament\_Country.

### **Tabela** Games

Por fim, falta-nos preencher a tabela relativa aos jogos. Inicialmente (**Fig.8**) julgámos que nesta tabela iriámos por todos os jogos da tabela ATP, porém, após uma análise daquilo que fora solicitado no enunciado do projeto, conferimos que logo para a primeira *query* teríamos de quantificar todos os jogos únicos realizados num certo país.

Dessa forma, procedemos à criação de uma tabela Games com jogos não duplicados, nos dois sentidos da questão: no que diz respeito a linhas que contém exatamente a mesma informação; e a linhas que correspondem ao mesmo jogo, apenas trocando o jogador com o oponente.

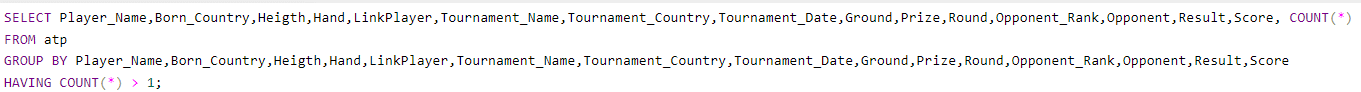
Todas fazes realizadas estão esquematizadas na **Figura 30**.



**Figura 30 |** Esquema da conceção da Tabela Games.

#### **Duplicados**

Tal como mencionado, apercebemo-nos da existência 2830 de jogos duplicados, que tinham exatamente a mesma informação, pelo que procedemos à sua eliminação (**0,216% de linhas eliminados**) na tabela Games numa primeira fase (**Fig.31**).



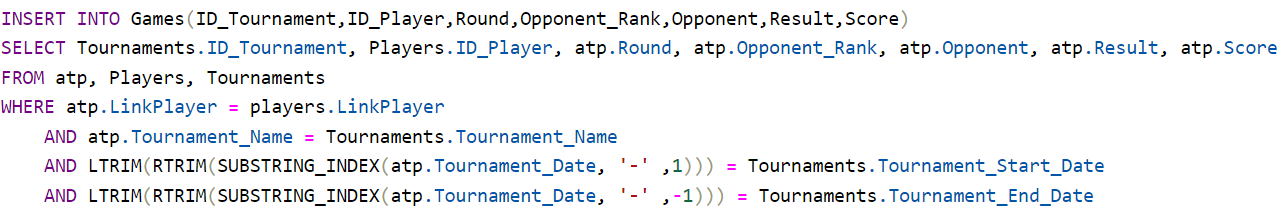
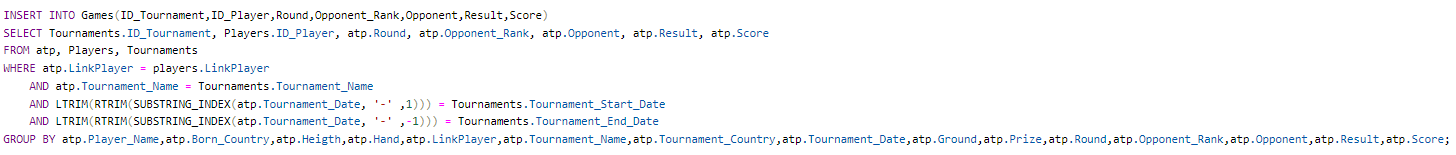
**Figura 31 |** Código e *Output* para verificar duplicados.

#### **Games**

Posto isto, criámos a tabela Games e preenchemos tal como mostrado na **Figura 32**. O nosso objetivo ao fazer esta primeira tabela é eliminar as 2830 linhas duplicadas e atribuir a cada jogo um ID\_Game para auxiliar dos jogos únicos numa fase posterior.

Uma imagem com texto

Descrição gerada automaticamente

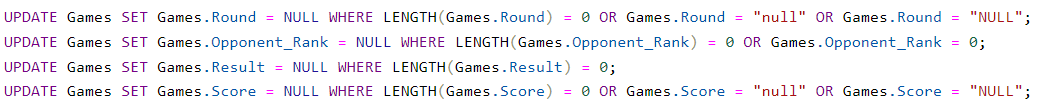




**Figura 32 |** Criação e Preenchimento da Tabela Games.

Assim, totalizámos 1 306 005 de jogos.

Aproveitámos para uniformizar já os valores omissos em NULL de modo a trabalhar com uma tabela mais limpa na fase seguinte (**Fig.33**).



**Figura 33 |** Uniformização dos Valores Omissos na Tabela Games.

#### **Unique\_Games**

A fim de procedermos à eliminação dos jogos, cujos Player\_Name e Opponent invertem papéis, tornando os jogos únicos, criámos a tabela Unique\_Games que contém, para além dos campos adicionados à tabela Games, o ID\_Opponent e o campo temporário Duplicado, para identificar as linhas duplicadas.

A criação do campo ID\_Opponent terá como principal propósito a identificação desses casos. Foi esta a principal razão pela qual adicionamos todos os oponentes à tabela Players.

É de notar que os jogos com oponente bye são considerados, pois não influenciam nas contagens, e não deixam de ser dados que poderão ser relevantes para posteriores análises.

A título de exemplo, o bye é útil quando queremos saber o percurso de um certo jogador num certo torneio, e através desta designação sabemos com certeza que a transição não sucessiva de uma qualificatória, para outra, dois níveis acima não se deveu a uma omissão de dados, mas sim, a uma auto qualificação premeditada.

Após a criação e preenchimento da tabela, que não alterou o número de linhas face à tabela Games, atualizámos todas as linhas conforme apresentado na **Figura 34** e eliminámos todos os casos, resultando numa tabela com jogos únicos.

Uma imagem com texto

Descrição gerada automaticamente

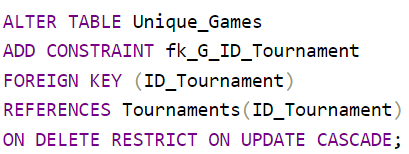
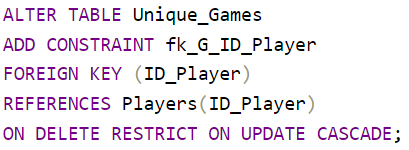
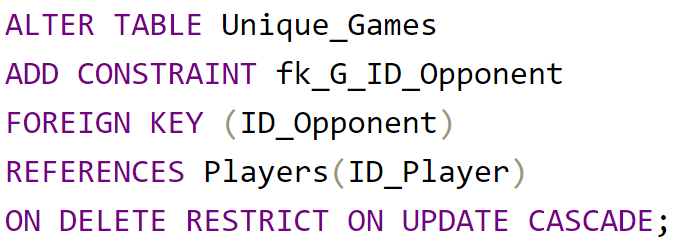




**Figura 34 |** Identificação e Eliminação dos jogos “duplicados”.

Neste procedimento foram eliminadas 605 237 **linhas** (46,34% dos registos), sendo contabilizados 700 768 **jogos únicos**.

Por fim, uma vez mais, criámos as Chaves Estrangeiras (**FK**) para as variáveis ID\_Player, ID\_Opponent e ID\_Tournament; removemos as tabelas ATP e Games; e, adicionalmente, renomeamos a tabela para Games, terminado assim a conceção do *Modelo Relacional*. (**Fig.35**)



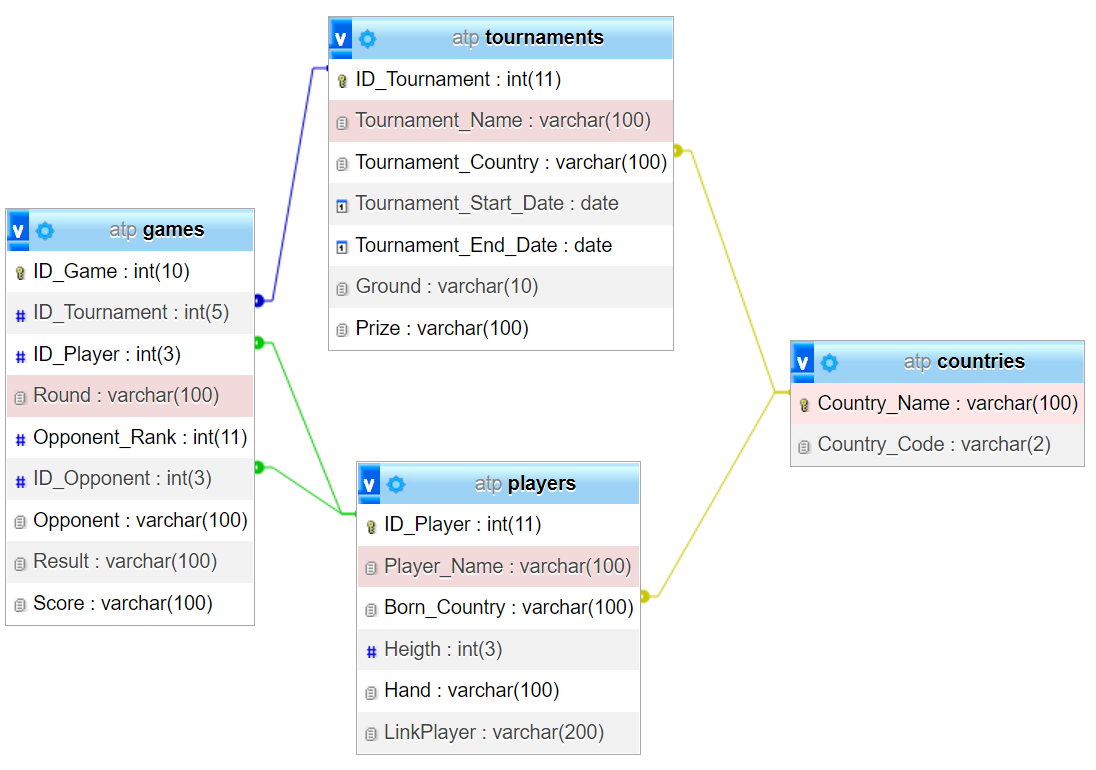
Uma imagem com texto

Descrição gerada automaticamente

**Figura 35 |** Criação das FKs da tabela Games e finalização do *Modelo Relacional*.

## **Diagrama do Modelo Relacional da Base de Dados**

Terminámos esta etapa com o seguinte diagrama.



**Figura 36 |** Diagrama do Modelo Relacional da Base de Dados ATP Tour.

## **Comandos** SELECT **e Respetivos Resultados**

**1 |** Para cada País (apenas um selecionado)

**a)** Número de **Jogadores**

**b)** Número de **Torneios**

**c)** Número de **Rondas**

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com mesa

Descrição gerada automaticamentePara o primeiro comando SELECT utilizámos conhecimentos prévios de SQL.

Ao comparar os resultados com os apresentados no enunciado, observamos que o Zimbabwe tem mais 1 torneio e 6 jogos. Este facto pode dever-se à adição do torneio ocorrido neste país que se encontrava como TBA no *dataset* inicial.

**2 |** Lista dos **10 melhores jogadores** com a sua % de jogos vencidos, por ordem por %

Uma imagem com texto

Descrição gerada automaticamente

Para o segundo SELECT utilizámos algumas funções que não foram abordadas em aula, tais como CASE WHEN … THEN e o LIMIT.

O nosso principal foco era tentar arranjar o código mais simples para resolver o problema em causa e para tal pensámos nos ciclos WHILE/IF de Python para calcular o número de vezes que o ID\_Player/ID\_Opponent ganha. Tendo em vista este raciocínio, pesquisámos uma conversão destes ciclos para SQL e apresentámos o código acima que contém um ciclo que soma 1 nas combinações em que um jogador ganha uma partida. Isto é, quando ele é ID\_Player e o resultado é W, e quando ele é ID\_Oppoent e o resultado é L.

Já o LIMIT serve apenas para que o *output* contenha 10 observações.

Aquando do visionamento, numa primeira instância, dos resultados deste e dos restantes SELECTs, que solicitam a mesma estatística, condicionada a fatores/características diferentes, verificámos que havia jogadores com *100%* de partidas vencidas, dado que apenas tinham jogado 1 partida e venceram-na.

De modo que as estatísticas não tivessem este enviesamento, decidimos que o número mínimo de partidas a considerar para que o jogador pudesse pertencer a estes *ranking* é de 30, pois verificámos que os jogadores competitivos jogam, em média, anualmente, 50-150 partidas, porém sabemos que esta base de dados poderá conter omissões de jogos, principalmente em torneios mais antigos, pelo que consideramos este o valor mais indicado, de modo a incluir todas as gerações de jogadores. [12]

Uma imagem com texto, recibo, captura de ecrã

Descrição gerada automaticamente

**3 |** Lista dos 10 melhores jogadores **canhotos** com a sua % de jogos vencidos em torneios do ***Grand Slam***

Uma imagem com texto

Descrição gerada automaticamente

Para o terceiro SELECT utilizámos a mesma parte inicial do anterior e acrescentámos as condições solicitadas, ou seja, os jogadores serem canhotos com o Players.Hand LIKE '%Left%' e serem apenas os jogos referentes ao *Grand Slam*, ou seja, Wimbledon, US Open, Roland Garros e Australian Open.

Uma vez mais, usámos o critério de ter no mínimo 30 jogos, para garantir que o resultado abrangesse apenas jogadores profissionais.

Uma imagem com mesa

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente**4 |** Lista dos 5 melhores jogadores com os seus jogos vencidos em **Hard Ground**

O quarto SELECT é semelhante ao anterior, mas consideramos todos os jogos cujo pavimento é do tipo Hard.

Uma imagem com mesa

Descrição gerada automaticamente

# **Conclusões**

Conclui-se com este trabalho, a preponderância da limpeza na base de dados para integridade dos dados e, por fim, na qualidade das estatísticas que dela possam advir.

Foi-nos possível estudar e conhecer técnicas no tratamento dos dados, pondo em prática o que fora lecionado previamente.

É de frizar apenas que as estatísticas efetuadas têm um enquadramento muito generalizado da realidade do ATP dado que dos jogadores do *TOP 500* que jogaram entre 28/03/1973 e 14/02/2022 havia, por exemplo, **6 270 jogadores** (o equivalente a dos jogadores) sem Born\_Country, pelo que os dados poderão ter enviesamentos significativos com a realidade.

Abstraindo isso, consideramos que tivemos dificuldades na parte referente à criação da tabela Games que fizesse face às necessidades do trabalho e no seu uso nos últimos SELECTs, para os quais não encontrámos índices otimizados a estes.

Todavia, em suma, consideramos que cumprimos com sucesso o objetivo pretendido neste projeto de criar um modelo relacional com o *dataset* inicial e de dar resposta às *queries* solicitadas.

# **Bibliografia**

1. Nogueira Ramos, P. (2008). *Desenhar Bases de Dados com UML* (2nd ed.). Edições Sílabo.
2. Moniruzzaman, A. B. M., & Hossain, S. A. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *ArXiv:1307.0191 [Cs]*. https://arxiv.org/abs/1307.0191
3. *Official Site of Men’s Professional Tennis | ATP Tour | Tennis*. (2019). ATP Tour. https://www.atptour.com/en
4. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O’Reilly.
5. P S, S. (2021, February 2). *Data Mining using CRISP-DM methodology*. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/data-mining-using-crisp-dm-methodology/
6. *History | ATP Tour | Tennis*. (n.d.). ATP Tour. https://www.atptour.com/en/corporate/history
7. ATP Tour. (2022). *The 2022 ATP® Official Rulebook*.
8. Hadlich, G. (2019, December 21). *How Does Tennis Scoring Work? (By Former PRO)*. My Tennis HQ. https://mytennishq.com/tennis-scoring-rules-the-ultimate-guide-explained/
9. ATP *Grand Slam All-Time Champions*. (2020). usopen.org. https://www.usopen.org/en\_US/visit/grand\_slam\_alltime\_champions.html
10. TennisCompanion. (2017, November 18). *Bye in Tennis | Definition, Examples, and Common Questions About The Bye*. TennisCompanion. https://tenniscompanion.org/bye/
11. Sport Science Committee of the United States Tennis Association (USTA). (n.d.). *A Guide to Improving Performance and Limiting Injury Through Improved Recovery*. https://www.usta.com/content/dam/usta/sections/texas/pdf/Recovery%20Booklet.pdf