



Universidade de Évora

Departamento de Informática

Administração de Bases de Dados

Ano letivo 2019 - 2020

# FIFA 19 complete player dataset

Alunos:

Luís Ressonha - 35003

Rúben Teimas - 39868

Docentes:

Carlos Pampulim Caldeira

25 de Maio de 2020

## Índice

|          |                                    |          |
|----------|------------------------------------|----------|
| <b>1</b> | <b>Introdução</b>                  | <b>1</b> |
| <b>2</b> | <b>Geração e inserção de dados</b> | <b>2</b> |
| <b>3</b> | <b>Estudo da performance</b>       | <b>3</b> |
| <b>4</b> | <b>11 Regras de Mullins</b>        | <b>4</b> |
| <b>5</b> | <b>Referências</b>                 | <b>5</b> |

# 1 Introdução

Para esta última fase do projeto foi-nos pedido que fizéssemos a afinação da Base de Dados.

Esta afinação passa não só pelo estudo da indexação mas também pela concepção das *queries*.

As *queries* devem ser concebidas tendo sempre em vistas as 11 regras de *Mullins*. Embora algumas delas sejam bastante intuitivas, como o *KISS* (*Keep It Simple Stupid*) existem outras como *Judicious use of OR* que podem não surgir logo no pensamento e como tal demonstrámos com exemplos práticos no nosso trabalho.

O primeiro passo para realizar esta fase foi gerar dados para "popular" a nossa base de dados.

Inicialmente pensámos em utilizar o *DBForge*, pois permitia conectar-se directamente à base de dados e gerar todos os dados sem quaisquer restrições manuais, contudo tivemos alguns problemas com este *software* acabando por utilizar um outro, *Spawner*.

Ainda que este gerador de dados não seja tão poderoso como o anterior, permitiu-nos gerar os 10 milhões de dados para as tabelas. Para além dele utilizámos um pequeno *script* em *Python* feito por nós para gerar os dados da tabela "Estatura", pois estávamos a ter problemas com o separador decimal do *Spawner*, que é uma virgula.

Assim sendo, o *SQL Developer* assumia que estávamos a inserir 4 valores, em vez dos 3 pretendidos.

O estudo da indexação não correu como planeado, tendo existindo alguns problemas. Os resultados obtidos nem sempre foram consistentes ainda que tenhamos seguido todos os procedimentos e nunca nos tenhamos esquecido de atualizar as estatísticas!

Com um pouco mais de tempo talvez conseguimos ter percebido a origem do problema e fazer um melhor estudo da indexação, contudo acabámos por nos atrasar um pouco na inserção dos dados e tal não foi possível.

## 2 Geração e inserção de dados

Utilizamos a aplicação Spawner para gerar os dados, que foi uma das sugestões do docente. Com esta aplicação podemos gerar dez milhões de dados para cada tabela da nossa base de dados. O Spawner cria um ficheiro com comandos em SQL.

A aplicação é bastante completa, o que fez com que fosse muito fácil criar dados para cada tabela, tanto dados numéricos para os ID's dos jogadores como colunas com texto pré definido por nós.

A inserção destes dados no Oracle foi a partir de comandos INSERT, pois não nos foi possível importar a partir do excel devido à falta dessa opção no Spawner. Os comandos foram escritos num ficheiro sql e, de seguida, foi aberto no Oracle SQL Developer e corrido como script.

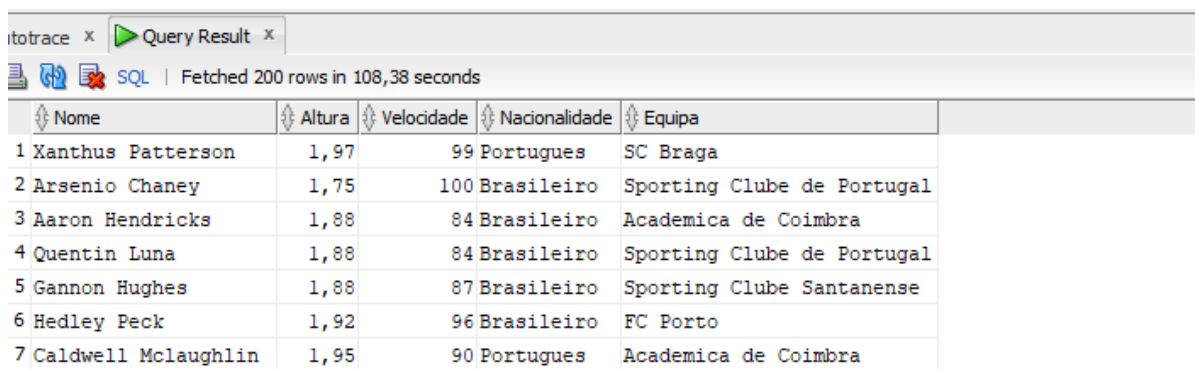
### 3 Estudo da performance

Criamos 2 queries e para cada uma delas criámos vários indexes. Na maioria dos testes obtivemos tempos de execução iguais ou ligeiramente melhores do que sem usar o índice, mas nada de muito significativo. Abaixo encontram-se as imagens com os tempos de execuções das queries.



|   | Nome                | Altura | Velocidade | Nacionalidade | Equipa                    |
|---|---------------------|--------|------------|---------------|---------------------------|
| 1 | Judah Lara          | 1,97   | 97         | Portugues     | SC Braga                  |
| 2 | Lucas Boyer         | 1,92   | 91         | Brasileiro    | FC Porto                  |
| 3 | Macon Preston       | 1,94   | 97         | Brasileiro    | Sporting Clube Santanense |
| 4 | Gannon Hughes       | 1,88   | 87         | Brasileiro    | Sporting Clube Santanense |
| 5 | Hedley Peck         | 1,92   | 96         | Brasileiro    | FC Porto                  |
| 6 | Caldwell McLaughlin | 1,95   | 90         | Portugues     | Academica de Coimbra      |
| 7 | Byron Lindsay       | 1,9    | 92         | Brasileiro    | Sporting Clube Santanense |
| 8 | Ashton Gregory      | 1,91   | 86         | Portugues     | Sporting Clube Santanense |
| 9 | Amal Vaughn         | 1,86   | 87         | Portugues     | Academica de Coimbra      |

Figura 1 Query 1 com Index 6



|   | Nome                | Altura | Velocidade | Nacionalidade | Equipa                     |
|---|---------------------|--------|------------|---------------|----------------------------|
| 1 | Xanthus Patterson   | 1,97   | 99         | Portugues     | SC Braga                   |
| 2 | Arsenio Chaney      | 1,75   | 100        | Brasileiro    | Sporting Clube de Portugal |
| 3 | Aaron Hendricks     | 1,88   | 84         | Brasileiro    | Academica de Coimbra       |
| 4 | Quentin Luna        | 1,88   | 84         | Brasileiro    | Sporting Clube de Portugal |
| 5 | Gannon Hughes       | 1,88   | 87         | Brasileiro    | Sporting Clube Santanense  |
| 6 | Hedley Peck         | 1,92   | 96         | Brasileiro    | FC Porto                   |
| 7 | Caldwell McLaughlin | 1,95   | 90         | Portugues     | Academica de Coimbra       |

Figura 2 Query 1 sem Index

Esta foi uma das poucas situações em que realmente se obteve uma melhoria substancial do tempo de execução, sendo que na criação de outros índices os resultados não eram significativos.

## 4 11 Regras de Mullins

Muitas das regras de Mullins são bastante intuitivas e como tal dispensam uma demonstração, ainda assim existem outras que devem ser vistas com exemplos práticos, como é o caso de:

```

1  ————— Judicious use of OR —————
2  SELECT "Jogador"."Nome", "Altura", "Velocidade", "Nacionalidade", "Equipa"
3  . "Nome" AS "Equipa"
4  FROM "Jogador" JOIN "Estatura" ON "Jogador"."ID" = "Estatura"."ID"
5  JOIN "Contrato" ON "Contrato"."ID" = "Jogador".ID
6  JOIN "Equipa" ON "Equipa"."Nome" = "Contrato"."Equipa"
7  WHERE "Jogador"."Velocidade" > 80 AND "Estatura"."Altura" > 1.7 AND "
8  Equipa"."País" = 'Portugal'
9  AND ("Jogador"."Nacionalidade" = 'Portugues' OR "Jogador"."Nacionalidade"
10 = 'Brasileiro');
11
12 — Forma correta de escrever esta query, seguindo a regra acima
13 SELECT "Jogador"."Nome", "Altura", "Velocidade", "Nacionalidade", "Equipa"
14 . "Nome" AS "Equipa"
15 FROM "Jogador" JOIN "Estatura" ON "Jogador"."ID" = "Estatura"."ID"
16 JOIN "Contrato" ON "Contrato"."ID" = "Jogador".ID
17 JOIN "Equipa" ON "Equipa"."Nome" = "Contrato"."Equipa"
18 WHERE "Jogador"."Velocidade" > 80 AND "Estatura"."Altura" > 1.7 AND "
19 Equipa"."País" = 'Portugal'
20 AND "Jogador"."Nacionalidade" IN ('Portugues', 'Brasileiro');

```

```

1  ————— Retrieve Only What is Needed —————
2  SELECT "Jogador"."Nome", "Altura", "Velocidade", "Nacionalidade", "Equipa"
3  . "Nome" AS "Equipa",
4  FROM "Jogador" JOIN "Estatura" ON "Jogador"."ID" = "Estatura"."ID"
5  JOIN "Contrato" ON "Contrato"."ID" = "Jogador".ID
6  JOIN "Equipa" ON "Equipa"."Nome" = "Contrato"."Equipa"
7  WHERE "Jogador"."Velocidade" > 80 AND "Estatura"."Altura" > 1.7 AND "
8  Equipa"."País" = 'Portugal'
9  AND "Jogador"."Nacionalidade" IN ('Portugues', 'Brasileiro');
10
11 — Seleccionar a Nacionalidade é dispensável
12 — Pois sabemos que será Portugues ou Brasileiro
13 SELECT "Jogador"."Nome", "Altura", "Velocidade", "Equipa"."Nome" AS "
14 Equipa"
15 FROM "Jogador" JOIN "Estatura" ON "Jogador"."ID" = "Estatura"."ID"
16 JOIN "Contrato" ON "Contrato"."ID" = "Jogador".ID
17 JOIN "Equipa" ON "Equipa"."Nome" = "Contrato"."Equipa"
18 WHERE "Jogador"."Velocidade" > 80 AND "Estatura"."Altura" > 1.7 AND "
19 Equipa"."País" = 'Portugal'
20 AND "Jogador"."Nacionalidade" IN ('Portugues', 'Brasileiro');

```

## 5 Referências

- Para o desenvolvimento deste documento foram usados alguns dados do site "*kaggle.com*"  
<https://www.kaggle.com/karangadiya/fifa19>