Estatísticas de Basquetebol universitário

Trabalho realizado por:
- Maria Helena Ferreira
- Miguel Rosa

Grupo 6

Índice

1. Subject description. Assignment requirements.

Os nossos dados (data set) estão relacionados com Basquete universitário entre os anos de 2009 e 2021 e incluem estatísticas avançadas da NBA. Cada estatística está associada a um jogador. Para cada jogador a única informação que não se altera é o nome, dado que a restante informação pode variar com cada ano, nomeadamente a equipa, o número de jogos feitos e as estatísticas associadas. O CollageYear refere-se ao ano de escolaridade, estando este escrito com siglas de nomes de anos escolares americanos. Este dataset possui algumas das estatísticas mais usuais no Basquete, como pontos, bloqueios, etc. No entanto, também inclui algumas estatísticas avançadas que empregam equações complicadas, como classificação ofensiva e PORPAG. Por exemplo, o atributo box plus / minus é uma métrica que estima a contribuição de um jogador de basquete para o equipa. Muitas das estatísticas são apresentadas em percentagem.

O número de fatos é superior ao mínimo de 15000 possuindo pelo menos uma medida aditiva e uma não aditiva. Especificamente, existem 61061 factos correspondentes sendo único para todos os factos a combinação (jogador, ano). Um exemplo de uma medida aditiva é o número de jogos feitos por um jogador num ano, o número de jogadores numa equipa por ano, entre outros. A maior parte das medidas não são aditivas porque estamos a lidar com estatísticas em percentagem, das quais só se pode fazer médias.

Finalmente existem mais do que 4 dimensões, e pelo menos uma das quais é temporal. A medida temporal é o ano que varia entre 2009 e 2021, sendo o ano da estatística em causa. As medidas vão ser detalhadas no próximo tópico.

2. Planning: dimensional bus matrix, dimensions and facts dictionary

Dimenções	Ano	Ano de escolaridade	Equipa	Conferência	Draft Number	Recruit Rank	Estastísticas
Estastisticas dos jogadores que foram drafted	x	x	x	x		x	x
Estastisticas dos jogadores que foram recrutados	x	x	x	x	x		x
Estastisticas dos jogadores por conferencia	x	x	x		x	x	x
Estratisticas dos jogadores por ano		x	x	x	x	x	x
Estastisticas dos jogadores por ano de escolaridade	x		x	x	x	x	x

Fig. 1: Dimensional Bus matrix

O nosso projeto tem 5 dimensões, ano, equipa, se o jogador foi escolhido no draft, se o jogador foi recrutado e ano escolar do jogador.
A nossa tabela de factos contém estatísticas sobre a performance de um jogador num ano específico. Aqui estão algumas estatísticas que existem na tabela de factos: Número de jogos jogados, número de minutos jogados, estatísticas de cestos (número de tentativas, número de sucessos e percentagem de sucesso), ratings (ofensivo, defensivo, etc) e eventos que aconteceram ao longo do jogo (ressaltos, assistências, perda da bola, bloqueios, roubos, pontos, etc). Os UMLs abaixo mostram especificamente todos os atributos das dimensões.

No NBA profissional, existem atualmente 30 equipas na NBA. A liga é dividida em duas conferências, a Conferência Leste e a Conferência Oeste. A conferência Leste tem três divisões chamadas Atlântico, Central e Sudeste. A conferência Western também tem três divisões, que são Noroeste, Pacífico e Sudoeste. No NBA universitário as conferências são diferentes e são representadas neste dataset como siglas do seu nome.

A tabela de factos é constituído pelos seguintes elementos não aditivos: minutes %, offensive rating, usage, effective field goal %, true shooting %, offensive rebound %, defensive rebound %, assist %, turnover %, free throws %, 2-pointers %, 3-pointers %, block %, steal %, free throw rate, defensive rating, adjusted defensive rating, defensive points over replacement per adjusted game, box plus/minus, offensive box plus/minus, defensive box plus/minus, gbpm, offensive gbpm, defensive gbpm, assist / turnover ratio, year, collage year.

A tabela de factos é constituído pelos seguintes elementos aditivos: free throws made, free throws attempted, 2-pointers made, 2-pointers attempted, 3-pointers made, 3-pointers attempted, offensive rebounds, defensive rebounds, total rebounds, assists, steals, blocks, points, games played, stops, minutes played.

3. MultiDim Conceptual model

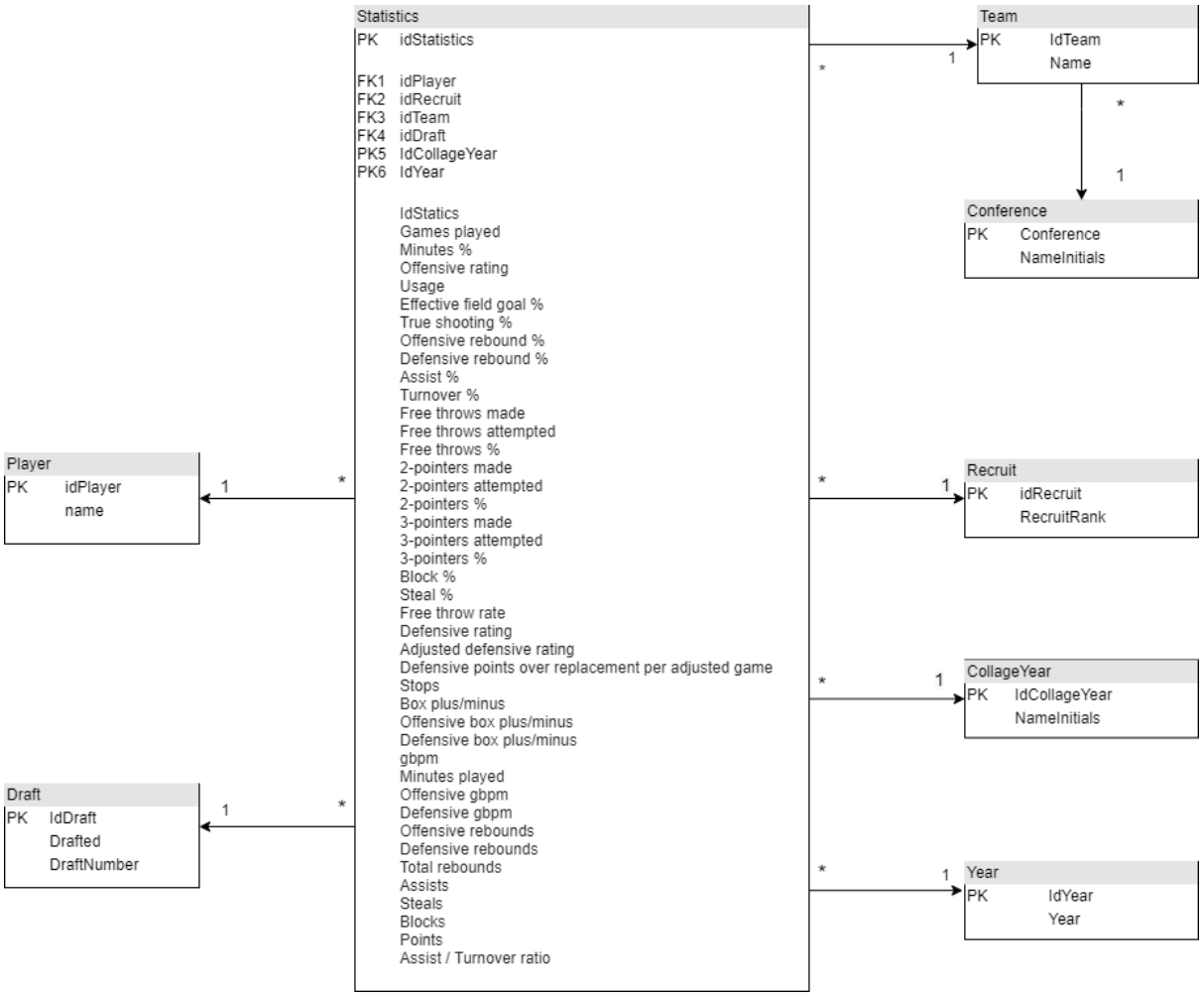

Fig. 2: Multi Dimensional Conceptual model

Tal como podemos observar, existem 8 classes: Player, Draft, Recruit, CollageYear, Year, Team, Statistics e Conference. Tal como já foi referido, o ano varia entre 2009 e 2021. O CollageYear pode ter as siglas para os seguintes nomes: Jr (junior), Fr (freshman), So (sophomore), Sr (senior) ou N/A. Na classe Statistics, todas as estatísticas em percentagens tomam os esperados valores em percentagem.

4. Dimensional data model

4.1. Relational model of the Data Warehouse

Nesta subseção iremos usar a notação: tableName (at1, at2, at3 -> tabela referenciada).

Player (idPlayer, name)
Team (idTeam , conference, name)
Draft(idDraft, draft Number)
Recruit(idRecruit, Recruit Rank)
Statics(IdStatics, Games played, Minutes %, Offensive rating, Usage, Effective field goal %, True shooting %, Offensive rebound %, Defensive rebound %, Assist %, Turnover %, Free throws made, Free throws attempted, Free throws %, 2-pointers made, 2-pointers attempted, 2-pointers %, 3-pointers made, 3-pointers attempted, 3-pointers %, Block %, Steal %, Free throw rate, Defensive rating, Adjusted defensive rating, Defensive points over replacement per adjusted game, Stops, Box plus/minus, Offensive box plus/minus, Defensive box plus/minus, gbpm, Minutes played, Offensive gbpm, Defensive gbpm, Offensive rebounds, Defensive rebounds, Total rebounds, Assists, Steals, Blocks, Points, Assist / Turnover ratio, Year, CollageYear, idPlayer -> Player, idTeam -> Team, idDraft -> Draft, idRecruit -> Recruit)

4.2. Esquema em estrela (Star schema)

No diagrama em estrela passamos a ter apenas 4 classes. Os valores nas classes que desapareceram passaram a estar na classe estatística e equipa. Dado que as conferências estão associadas à presença de uma equipa, os valores nesta classe passaram a integrar a classe equipa. Apesar do atributo ano de escolaridade referir-se ao jogador, este também está relacionado com o jogador num ano específico, dado que no ano seguinte o valor irá, provavelmente, mudar. Portanto as classes ano e ano de escolaridade passaram a fazer parte da classe estatística.

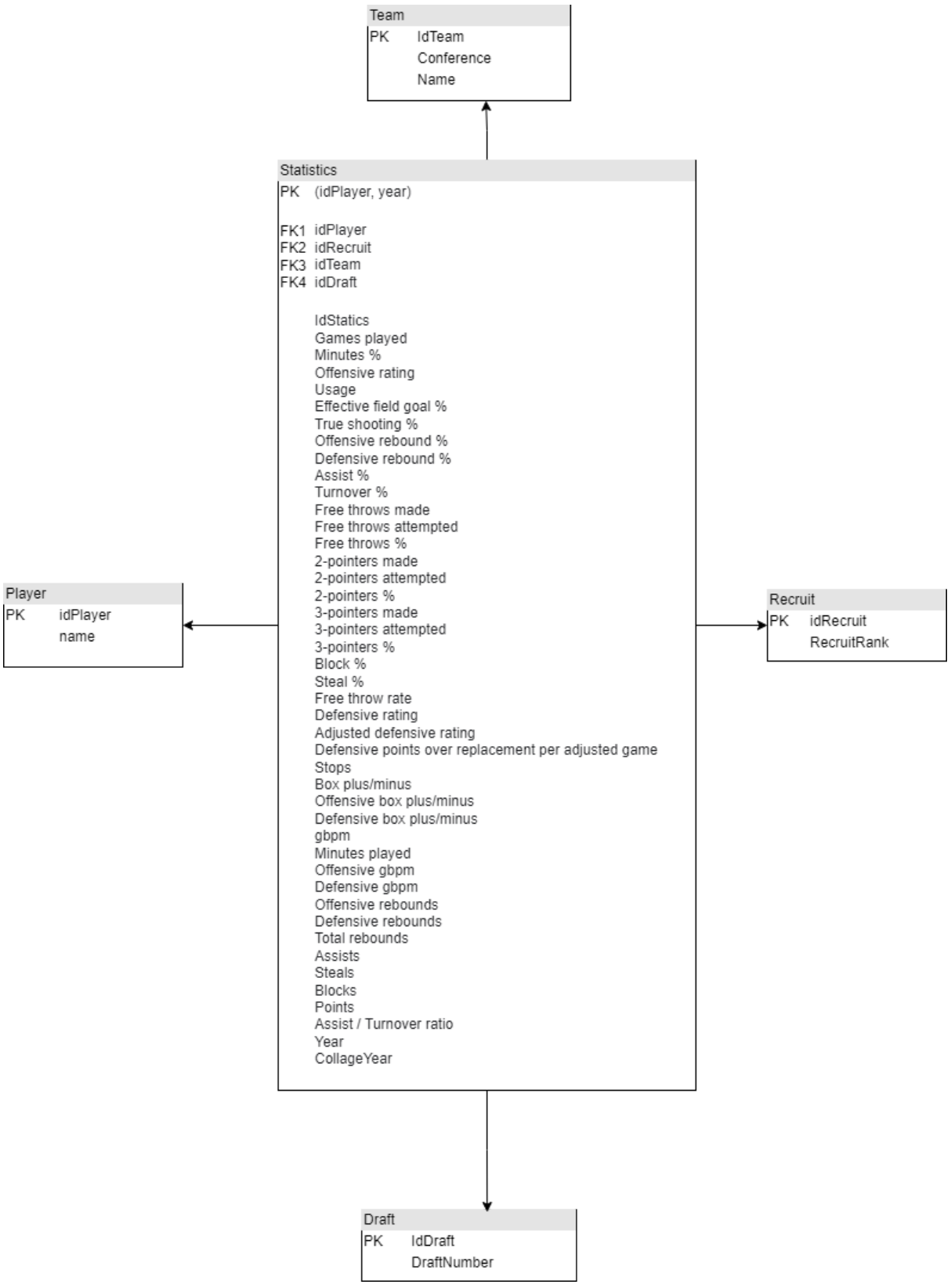

Fig. 3: Diagrama em estrela

4.3. Script SQL

```
DROP TABLE IF EXISTS grupo6.player;
DROP TABLE IF EXISTS grupo6.team;
DROP TABLE IF EXISTS grupo6.draft;
DROP TABLE IF EXISTS grupo6.recruit;
DROP TABLE IF EXISTS grupo6.statics;



CREATE TABLE grupo6.player (
    idPlayer INT PRIMARY KEY,
    name VARCHAR(30) NOT NULL
);

CREATE TABLE grupo6.team (
    idTeam INT PRIMARY KEY,
    conference VARCHAR(5),
    name VARCHAR(30) NOT NULL
);

CREATE TABLE grupo6.draft(
    idDraft INT PRIMARY KEY,
    draftNumber INT
);

CREATE TABLE grupo6.recruit(
    idRecruit INT PRIMARY KEY,
    recruitRank FLOAT
);

CREATE TABLE grupo6.statics(
    gamesPlayed INT,
    minutesInPercentage FLOAT,
    offensiveRating FLOAT,
    usageStat FLOAT,
    effectiveFieldGoalPercentage FLOAT,
    truesShootingPercentage FLOAT,
    offensiveReboundPercentage FLOAT,
    defensiveReboundPercentage FLOAT,
    assistPercentage FLOAT,
    turnoverPercentage FLOAT,
    freeThrowsMade INT,
    freeThrowsAttempted INT,
    freeThrowsPercentage FLOAT,
```

```
    twoPointersMade INT,
    twoPointersAttempted INT,
    twoPointersPercentage FLOAT,
    threePointersMade INT,
    threePointersAttempted INT,
    threePointersPercentage FLOAT,
    blockInPercentage FLOAT,
    stealPercentage FLOAT,
    freeThrowRate FLOAT,
    defensiveRating FLOAT,
    adjustedDefensiveRating FLOAT,
    defensivePointsOverReplacementPerAdjustedGame FLOAT,
    stops FLOAT,
    boxPplusOrminus FLOAT,
    offensiveBoxPlusOrminus FLOAT,
    defensiveBoxPlusOrminus FLOAT,
    gbpm FLOAT,
    minutesPlayed FLOAT,
    offensiveGbpm FLOAT,
    defensiveGbpm FLOAT,
    offensiveRebounds FLOAT,
    defensiveRebounds FLOAT,
    totalRebounds FLOAT,
    assists FLOAT,
    steals FLOAT,
    blocks FLOAT,
    points FLOAT,
    assistOrTurnoverRatio FLOAT,
    year INT,
    collageYear VARCHAR(4),
    idPlayer INT REFERENCES player(idPlayer),
    idTeam INT REFERENCES team(idTeam),
    idDraft INT REFERENCES draft(idDraft),
    idRecruit INT REFERENCES recruit(idRecruit),
    CONSTRAINT PK_statics PRIMARY KEY(idPlayer, year)
);
```

5. Data sources selection

Este projeto contém uma única fonte de dados. A fonte escolhida foi o dataset “College Basketball 2009-2021 + NBA Advanced Stats” presente no kaggle.com (especificamente no link https://www.kaggle.com/adityak2003/college-basketball-players-20092021).
Esta base de dados contém algumas estatísticas mais simples, como pontos, bloqueios, etc. No entanto, os dados também contém algumas estatísticas mais avançadas que empregam equações complicadas, como classificação ofensiva e PORPAG. Todas essas estatísticas precisam ser consideradas com cautela, pois algumas podem ser enganosas.

Existem estatísticas para os anos de 2009 a 2021. Cada jogador tem no máximo uma estatística em cada ano.

6. Transformations including incremental loadings

Esta seção apresenta as Transformações ETL necessárias para carregar os dados presentes num ficheiro CSV para o MYSQL. O software utilizado foi o Kettle (Pentaho Data Integration). Na primeira transformação estamos a carregar os dados provenientes do ficheiro CSV que foi extraído da fonte de dados mencionada na seção anterior. Algumas das colunas não serão utilizadas em nenhuma tabela, dado terem mais valores em falta do que valores presentes. No passo Combination Team, a tabela Team é preenchida, sendo o idTeam um parâmetro criado incrementalmente no preenchimento da tabela. O mesmo preenchimento automático incremental acontece na etapa Combination Draft e Combination Recruit. A tabela de output statics irá conter chaves estrangeiras para as tabelas Team, Draft, Recruit e Player. Para além destas chaves estrangeiras, a tabela contém múltiplos campos extraídos do ficheiro CSV. No fluxo de baixo, o mesmo ficheiro csv é carregado. Neste fluxo as colunas são ordenadas pelo campo pid, que representa o id do Player. Temos que usar este pid pois dois jogadores diferentes podem ter o mesmo nome. Em oposição às outras tabelas, este id é extraído do CSV. Dado que o mesmo jogador pode aparecer em várias estatísticas entre 2009 e 2021, estamos interessados em remover entradas repetidas, pois representam o mesmo jogador. Para remover as linhas repetidas foi utilizada a etapa “Unique rows”. Na etapa “Table Statics”, o id de jogador a ser utilizado é o campo pid extraído do csv.

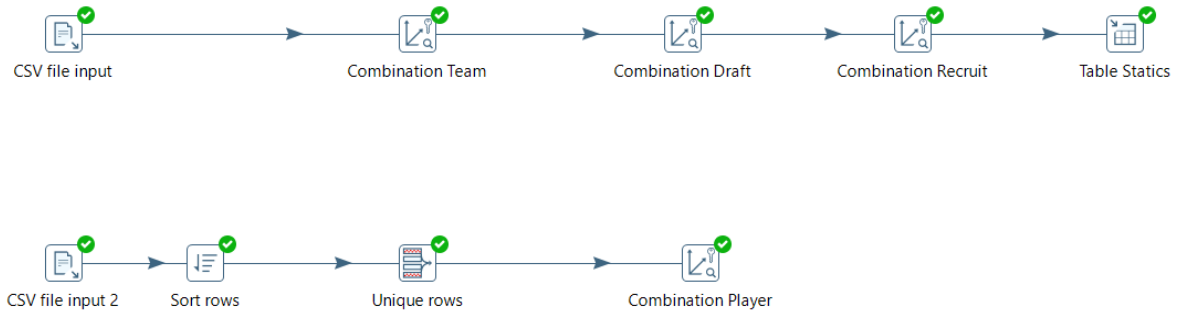

Fig. 4: Transformações ETL

Para confirmar que as tabelas tinham sido carregadas corretamente, fizemos várias queries à base de dados. Nomeadamente a visível em baixo.

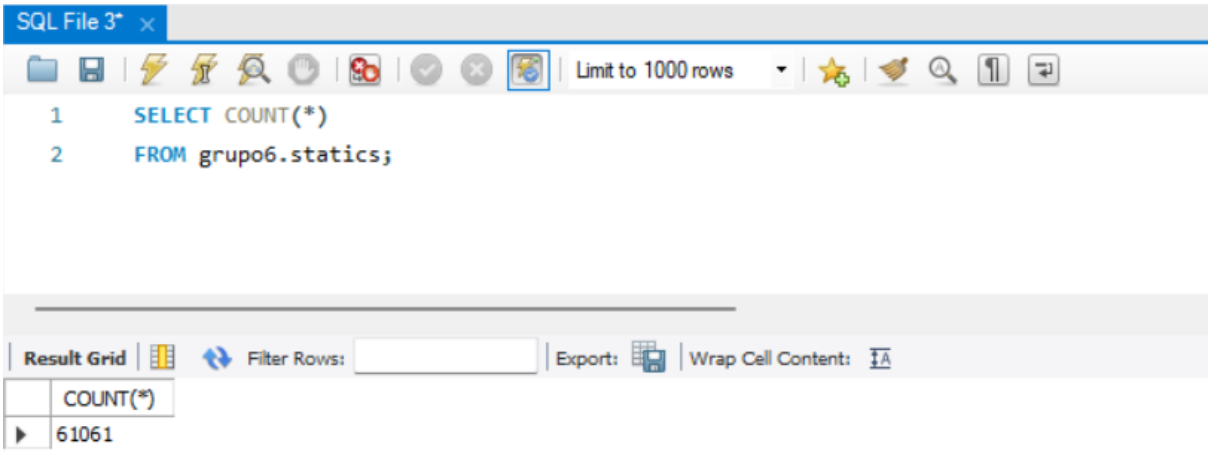

Fig. 5: Query de confirmação dos dados

Para além da transformação anteriormente descrita, também foi criada uma transformação auxiliar que é usada na execução do trabalho de atualização da base de dados. Esta transformação tem como função evitar linhas que já foram adicionadas anteriormente, apagando-as.

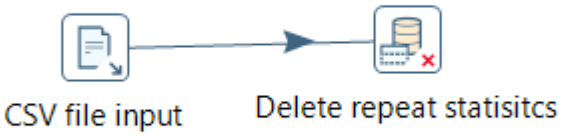

Fig. 6: Transformação auxiliar

Alternativamente, poderia ser adicionado no CSV um campo “created at”. Nesta alternativa a transformação filtrava os dados pelo valor do campo “created at”, adicionando apenas as linhas com valor superior ou igual à última hora e data de update neste campo. Desta maneira, não se eliminavam os valores antigos, só adicionamos os novos dados, i.e., não há update. Apesar de mais custosa, na alternativa implementada as estatísticas podem ser atualizadas, desde que mantenham o mesmo pid e ano. Isto porque apagamos a estatísticas comparado o pid e o ano apenas, dado que representam a chave primária.

7. Jobs

Esta seção fala dos Trabalhos desenvolvidos para possibilitar a atualização da base de dados com frequência. Trabalhos são uma ótima ferramenta para executar diversas transformações, pois executam código em série em vez de em paralelo como nas transformações, evitando possíveis conflitos. Este trabalho é composto por duas transformações, sendo elas descritas na seção anterior. Para além destas transformações,

o trabalho também anota quando foi executada uma atualização na base de dados e se esta falhou.

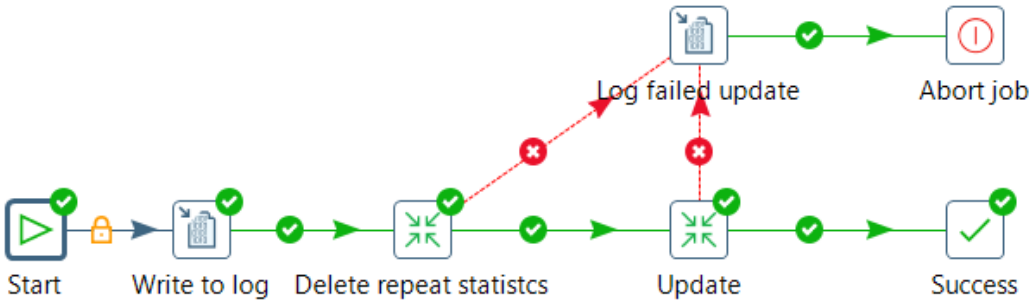

Fig. 5: Trabalho de atualização da base de dados

8. Multi-dimensional modeling

Cada cubo OLAP representa uma tabela de fatos. Portanto, neste trabalho, temos apenas um cubo, chamado de statistics. O cubo é apresentado por meio de medidas e dimensões, o que permite a análise multidimensional dos dados. Nas dimensões temos 4 hierarquias principais: Draft, Player, Recruit e Team, como podemos ver na Figura 6. O cubo será detalhado no resto da seção. A imagem do mesmo cubo no workbench encontra-se na Seção 8.3.

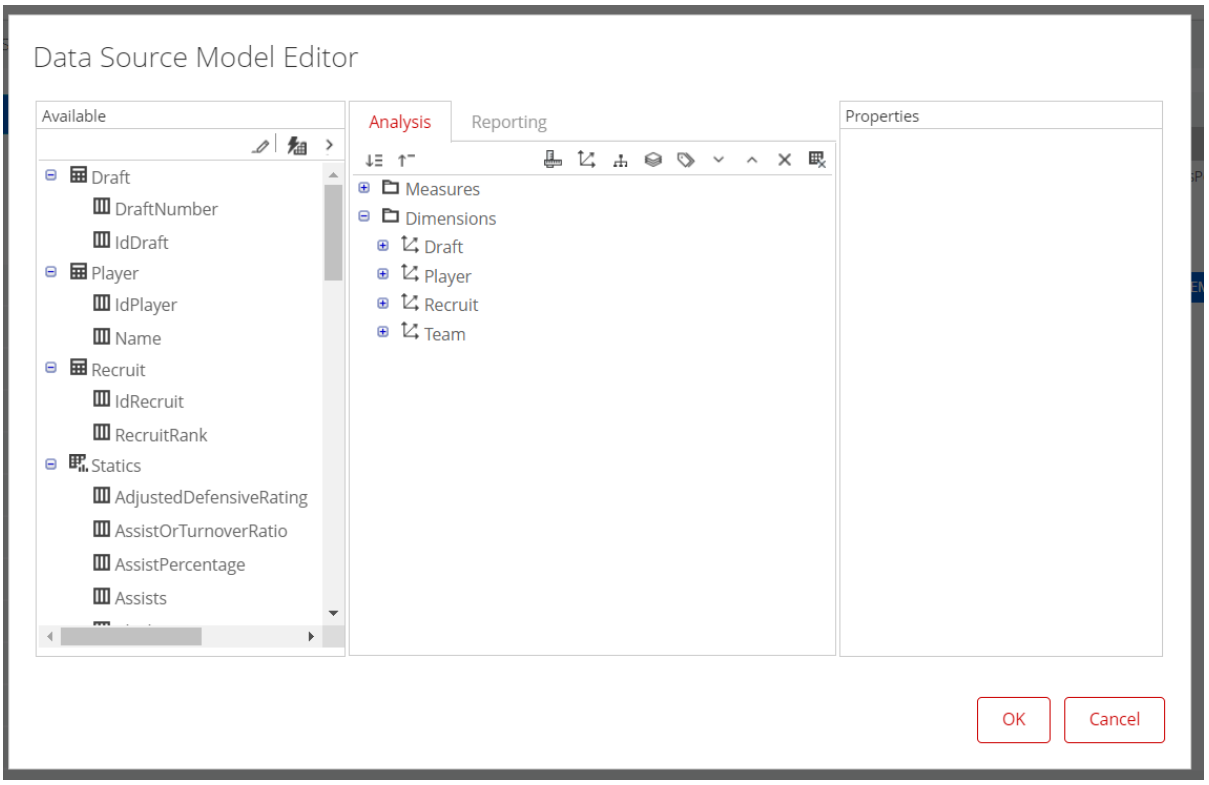

Fig. 6: Vista resumida do cubo statistics

8.1 - Dimensões

A dimensão draft é constituída pelos elementos DraftNumber e IdDraft. A dimensão Player é constituída por IdPlayer e Name, sendo o idPlayer o campo pid referido anteriormente e o name o nome do jogador. As dimensões podem ser vistas em detalhe na figura 7.

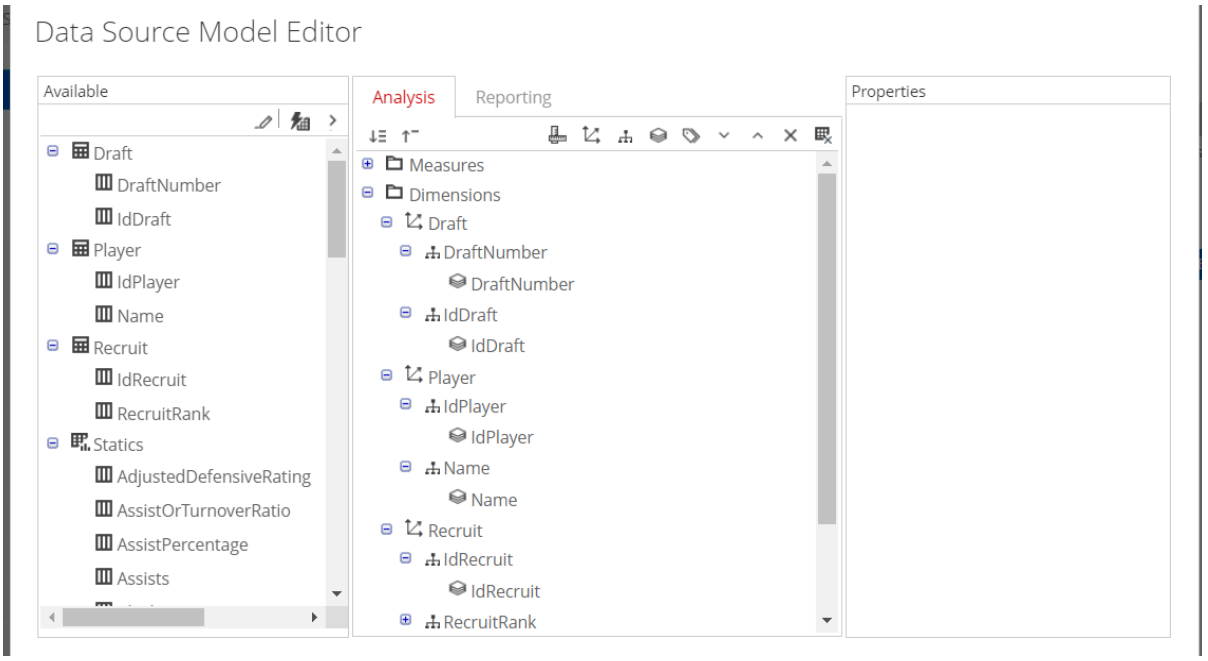

Fig. 7: Dimensões Draft e Player detalhadas

Por fim, temos as dimensões Recruit e Team. A dimensão Recruit é constituída por IdRecruit e RecruitRank e a dimensão Team é constituída por 3 hierarquias: conference, idTeam e Name.

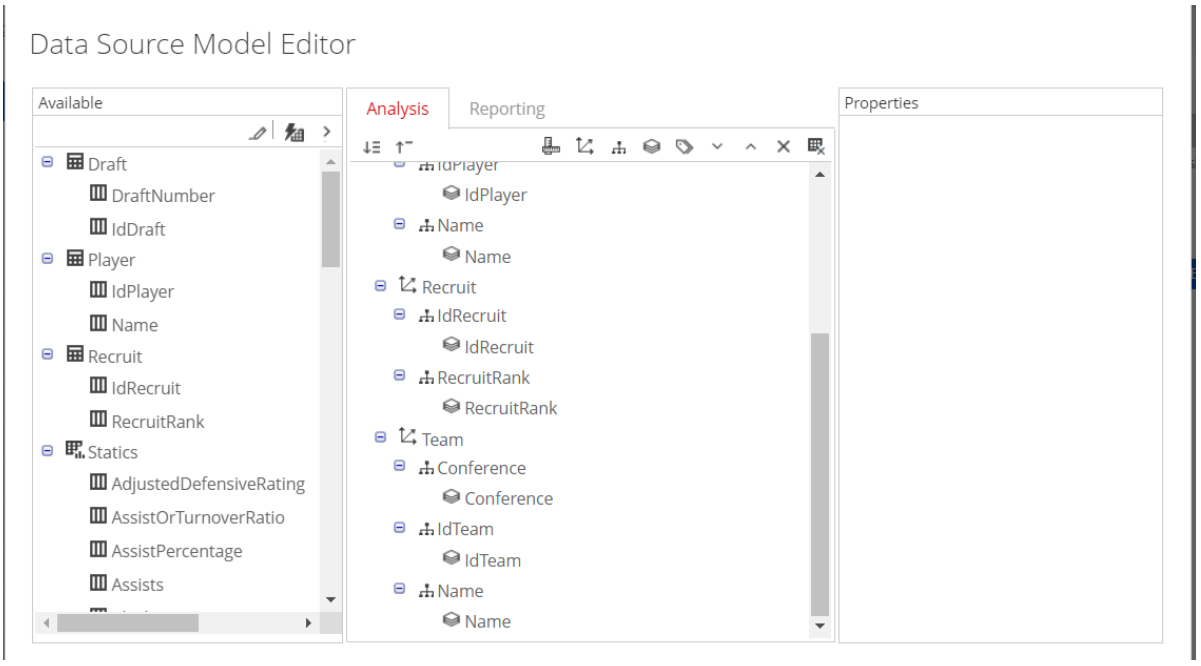

Fig. 8: Dimensões Recruit e Team detalhadas

8.2 - Medidas

A longa lista de medidas do cubo statistics pode ser vista na extensa lista de figuras que se seguem.

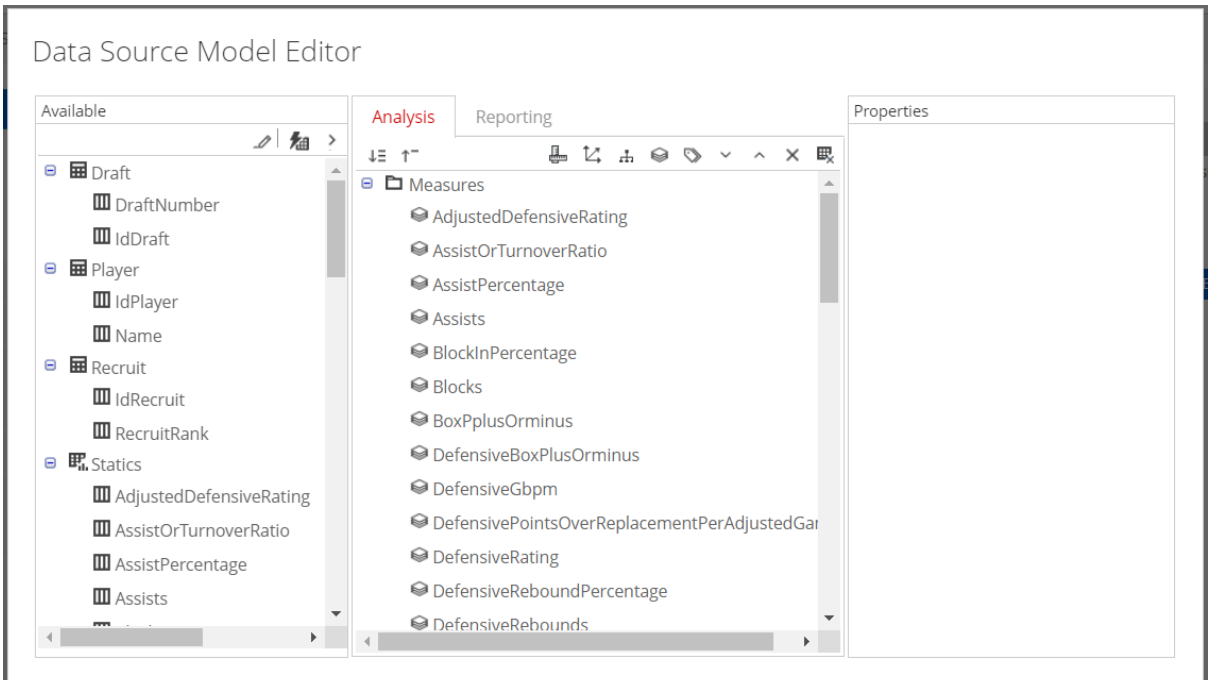

Fig. 9: Medidas do cubo (Parte 1/4)

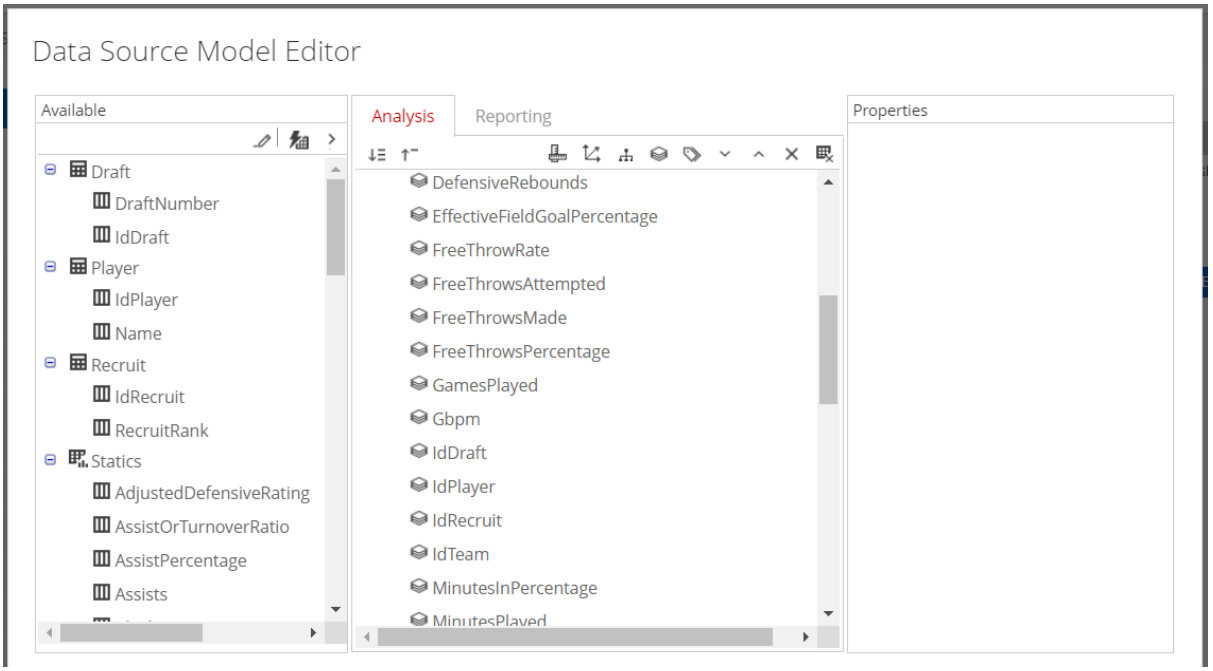

Fig. 10: Medidas do cubo (Parte 2/4)

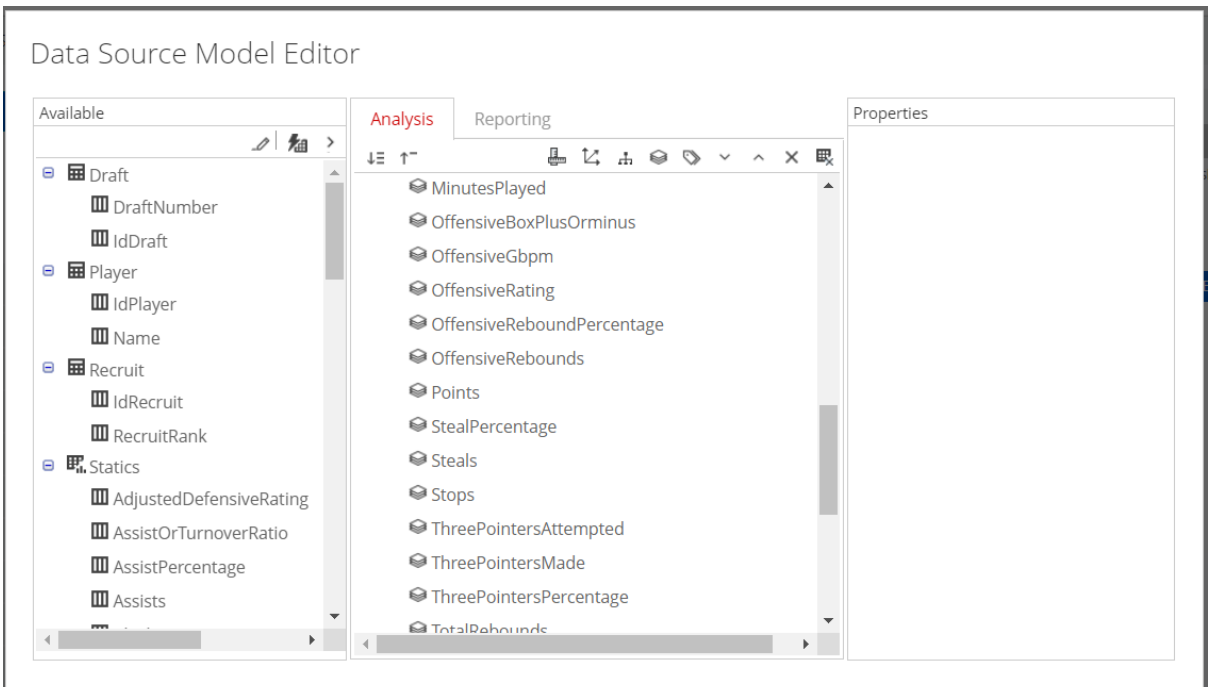

Fig. 11: Medidas do cubo (Parte 3/4)

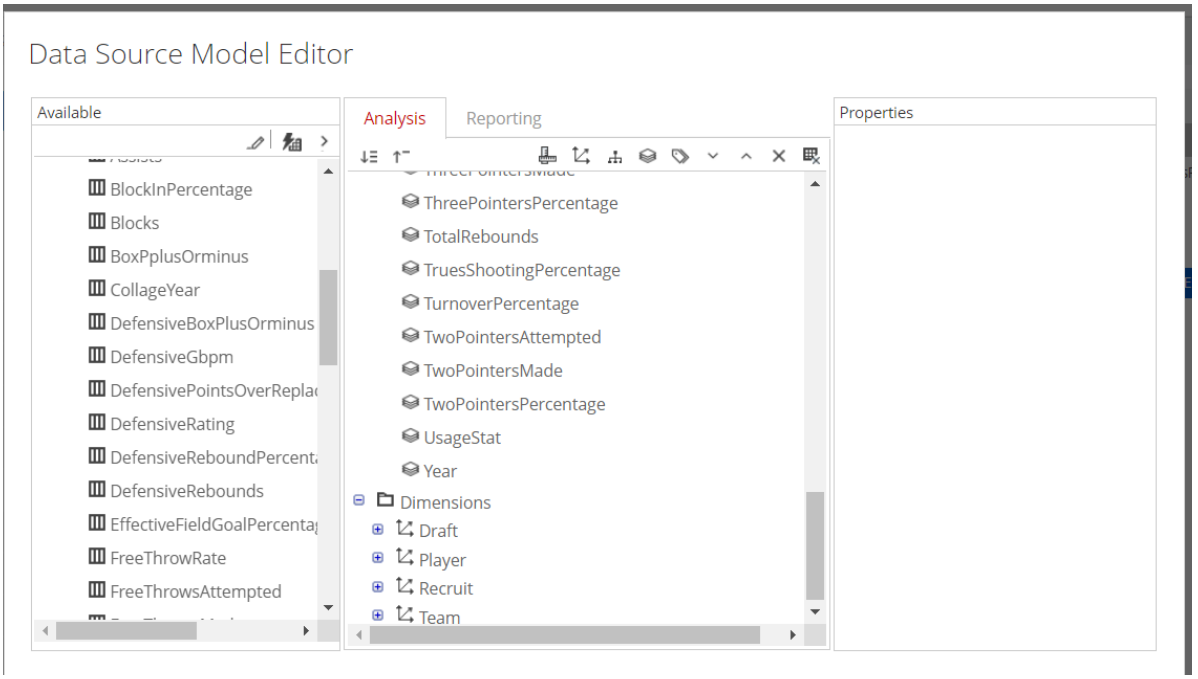

Fig. 12: Medidas do cubo (Parte 4/4)

8.3 - Cubo no Workbench

O esquema do cubo no Workbench pode ser visto nas duas imagens que se seguem. Este cubo foi publicado com sucesso.

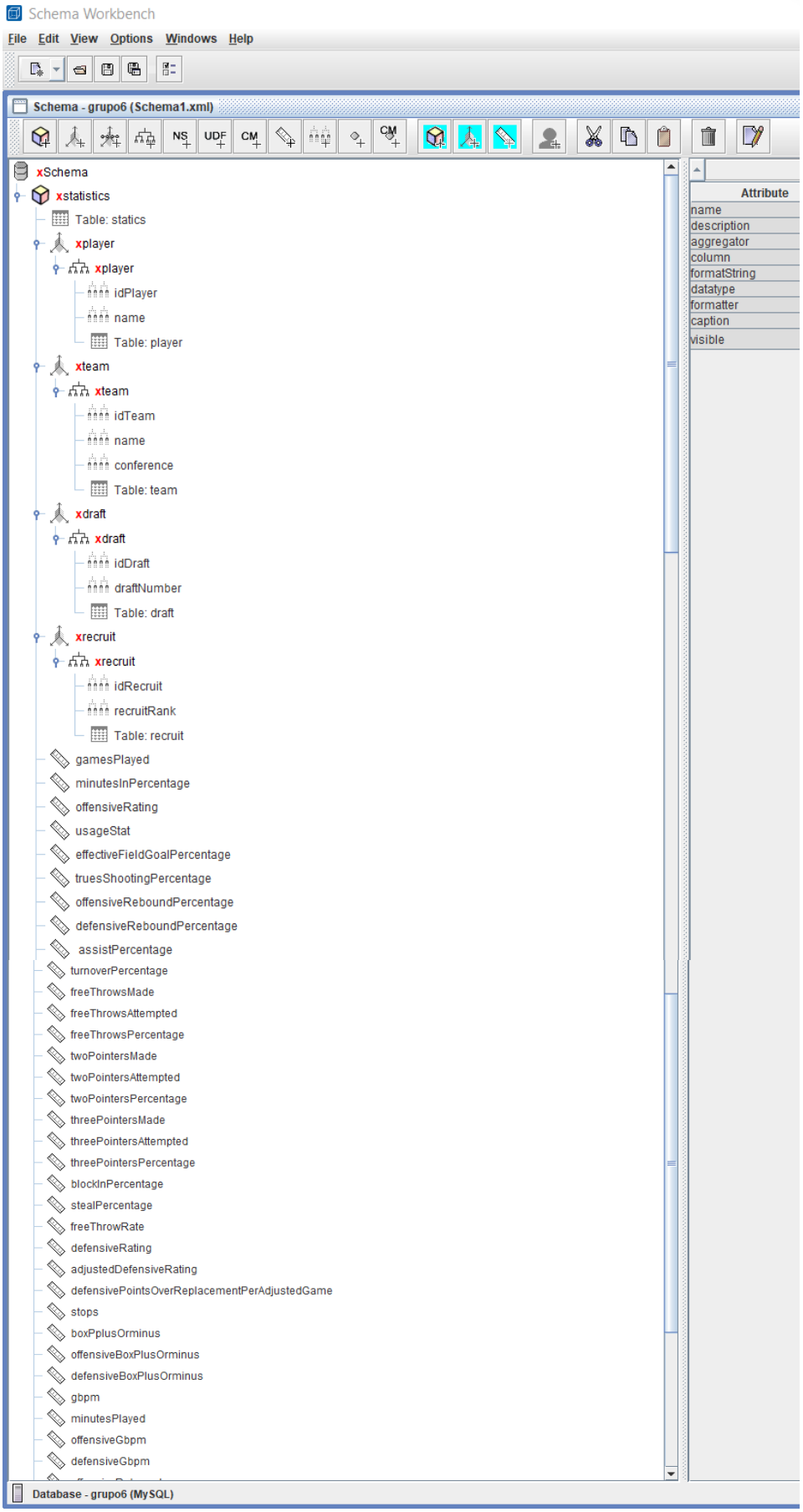

Fig. 13: cubo no Workbench (Parte 1/2)

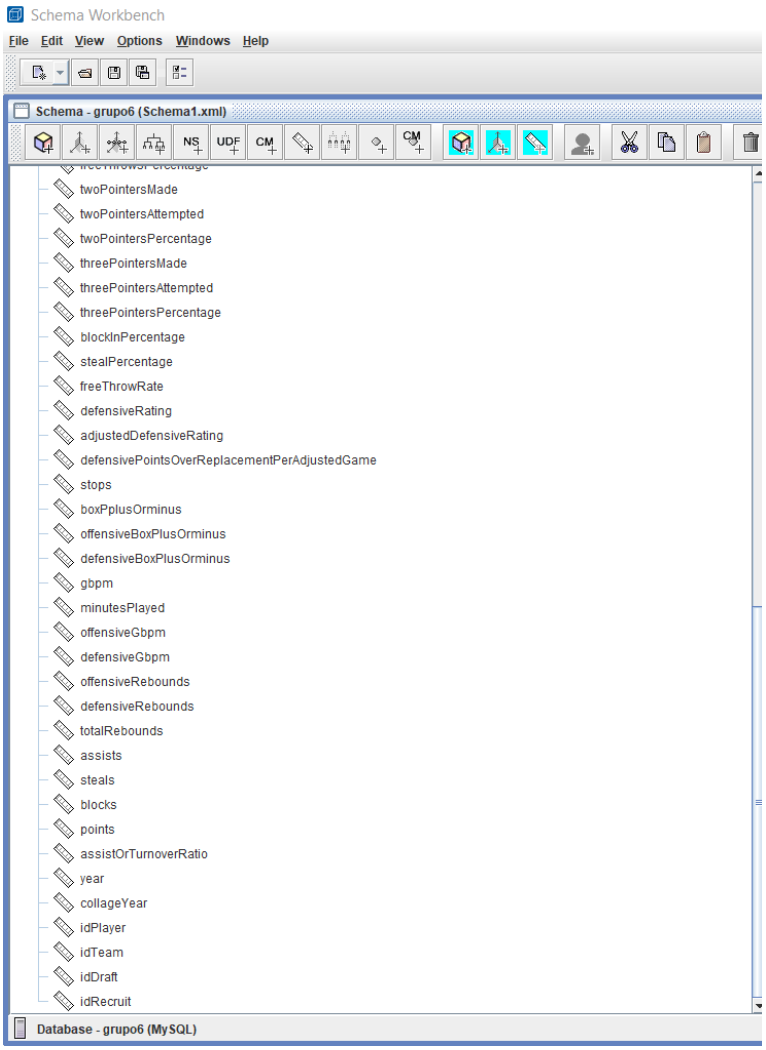

Fig. 14: cubo no Workbench (Parte 1/2)

A criação deste cubo possibilitou a construção do dashboard descrito na seção seguinte.

9. Data analysis

Nesta seção iremos apresentar o dashboard e as queries MDX utilizadas.

9.1 .Dashboards

Os dashboards são uma ótima ferramenta para análise de negócios, dado que uma representação gráfica é normalmente mais sintética e simples de entender. Neste trabalho queremos poder analisar o jogador, a equipa e as conferências em maior detalhe.

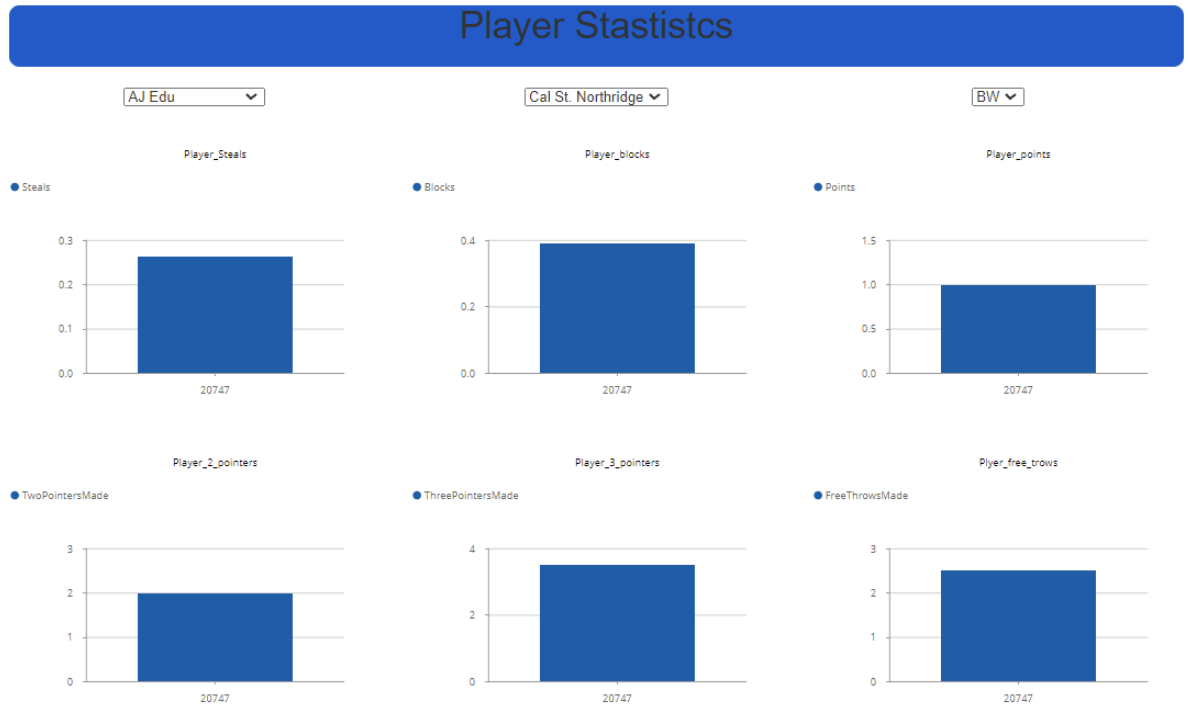

Fig. 15: Estatísticas relacionadas com o jogador

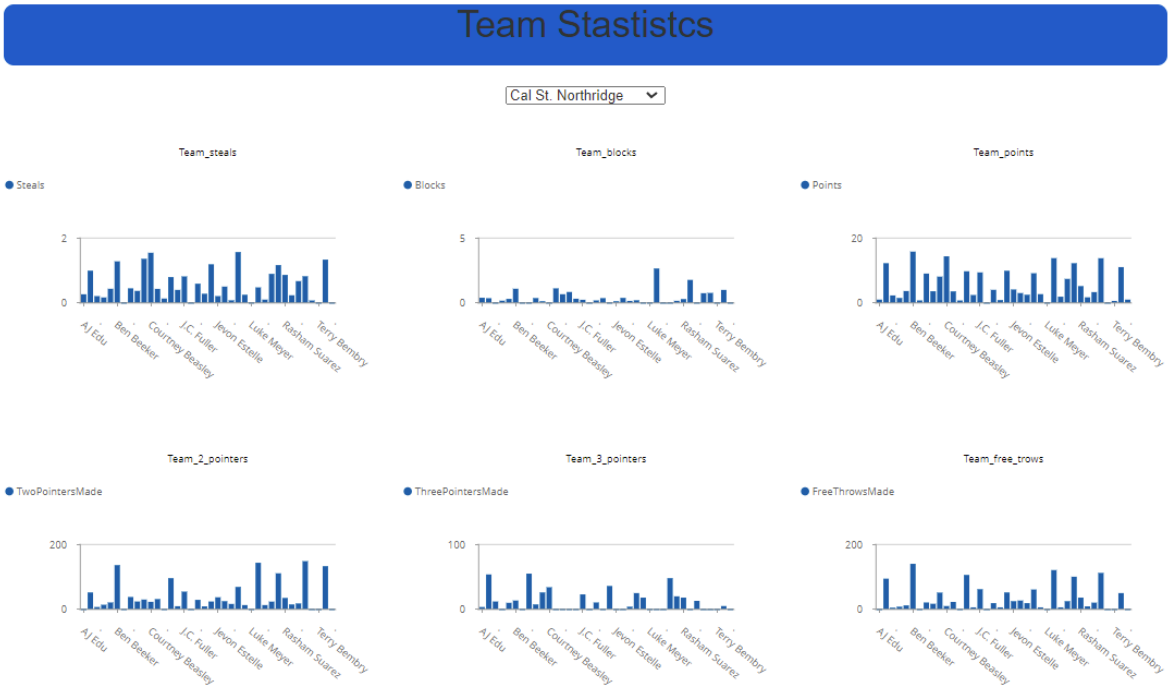

Fig. 16: Estatísticas relacionadas com a equipa

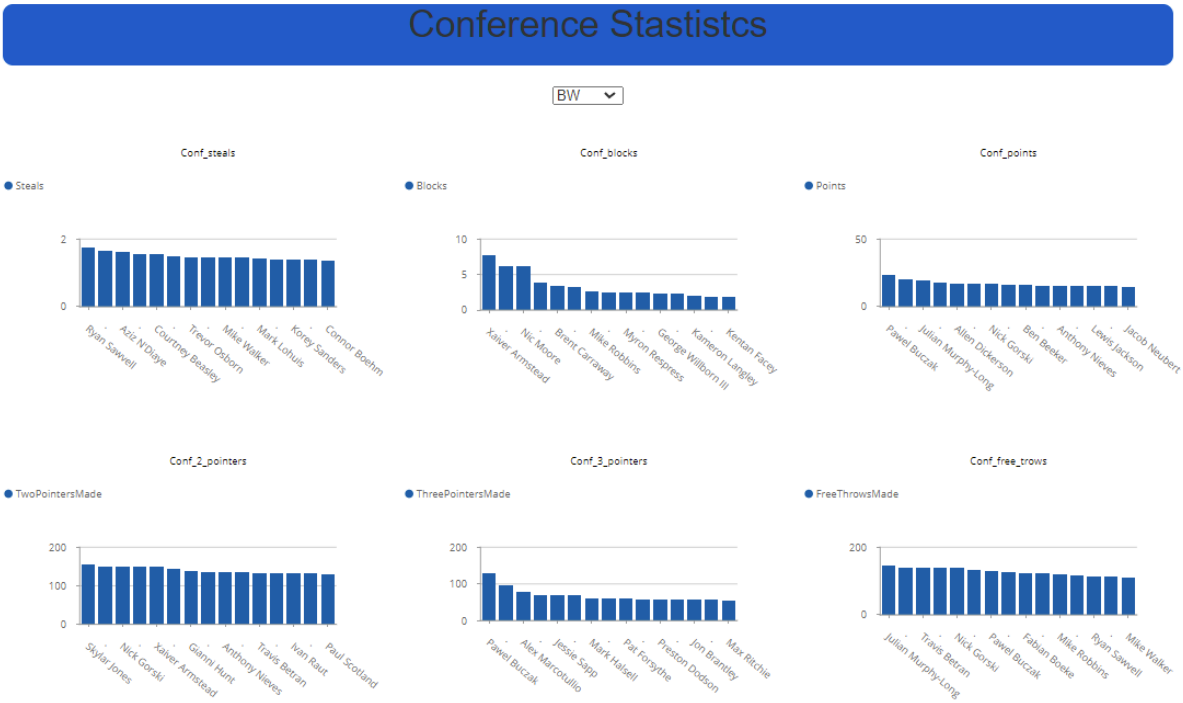

Fig. 17: Estatísticas relacionadas com as conferências

9.2. MDX queries

As queries usadas foram as seguintes:

Player name selector query

```
with member [Measures].[Name] as '[Player.Name].CurrentMember.UniqueName' select
TopCount( filter({Descendants([Player.Name].[All Player.Names] ,[Player.Name].[Name])},
not isempty(([Player.Name].CurrentMember)) ) , 50) on ROWS,
 {[Measures].[Name]} on Columns
 from [group6]
```

Player team query

```
with member [Measures].[Name] as '[Team.Name].CurrentMember.UniqueName' select
TopCount( filter({Descendants([Team.Name].[All Team.Names] ,[Team.Name].[Name])}, not
isempty(([Team.Name].CurrentMember)) ) , 50) on ROWS,
 {[Measures].[Name]} on Columns
 from [group6]
 where (${Player_name_selectorParameter})
```

Player Conf query

```
with member [Measures].[Name] as '[Team.Conference].CurrentMember.UniqueName'
select TopCount( filter({Descendants([Team.Conference].[All Team.Conferences]
,[Team.Conference].[Conference])}, not isempty(([Team.Conference].CurrentMember)) ) , 50)
on ROWS,
 {[Measures].[Name]} on Columns
 from [group6]
 where (${Player_name_selectorParameter})
```

Team name selector query

```
with member [Measures].[Name] as '[Team.Name].CurrentMember.UniqueName' select
TopCount( filter({Descendants([Team.Name].[All Team.Names] ,[Team.Name].[Name])}, not
isempty(([Team.Name].CurrentMember)) ) , 50) on ROWS,
 {[Measures].[Name]} on Columns
 from [group6]
```

Team steals query

```
 select NON EMPTY({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}) on ROWS,
 NON EMPTY({[Measures].[Steals]}) on Columns
 from [group6]
 where (${Team_name_selectorParameter})
```

Team blocks query

```
 select NON EMPTY({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}) on ROWS,
 NON EMPTY({[Measures].[Blocks]}) on Columns
 from [group6]
 where (${Team_name_selectorParameter})
```

Team points query

```
 select NON EMPTY({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}) on ROWS,
 NON EMPTY({[Measures].[Points]}) on Columns
 from [group6]
 where (${Team_name_selectorParameter})
```

Conference name selector query

```
with member [Measures].[Name] as '[Team.Conference].CurrentMember.UniqueName'
select TopCount( filter({Descendants([Team.Conference].[All Team.Conferences]
,[Team.Conference].[Conference])}, not isempty(([Team.Conference].CurrentMember)) ) , 50)
on ROWS,
 {[Measures].[Name]} on Columns
 from [group6]
```

Conference steals query

```
 select NON EMPTY(TopCount({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}, 15, [Measures].[Steals])) on ROWS,
 NON EMPTY({[Measures].[Steals]}) on Columns
 from [group6]
 where (${Conf_name_selectorParameter})
```

Conference blocks query

```
 select NON EMPTY(TopCount({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}, 15, [Measures].[Blocks])) on ROWS,
 NON EMPTY({[Measures].[Blocks]}) on Columns
 from [group6]
 where (${Conf_name_selectorParameter})
```

Conference points query

```
 select NON EMPTY(TopCount({Descendants([Player.Name].[All Player.Names],
[Player.Name].[Name])}, 15, [Measures].[Points])) on ROWS,
 NON EMPTY({[Measures].[Points]}) on Columns
 from [group6]
 where (${Conf_name_selectorParameter})
```

Player steals query

```
 select NON EMPTY(TopCount({Descendants([Player.IdPlayer].[All Player.IdPlayers],
[Player.IdPlayer].[IdPlayer])}, 1, [Measures].[Steals])) on ROWS,
 NON EMPTY({[Measures].[Steals]}) on Columns
 from [group6]
 where (${Player_name_selectorParameter})
```

Player blocks query

```
 select NON EMPTY(TopCount({Descendants([Player.IdPlayer].[All Player.IdPlayers],
[Player.IdPlayer].[IdPlayer])}, 1, [Measures].[Blocks])) on ROWS,
 NON EMPTY({[Measures].[Blocks]}) on Columns
 from [group6]
 where (${Player_name_selectorParameter})
```

Player points query

```
 select NON EMPTY(TopCount({Descendants([Player.IdPlayer].[All Player.IdPlayers],
[Player.IdPlayer].[IdPlayer])}, 1, [Measures].[Points])) on ROWS,
 NON EMPTY({[Measures].[Points]}) on Columns
 from [group6]
 where (${Player_name_selectorParameter})
```

10. Conclusion

Este projeto teve diferentes componentes, nomeadamente a criação de um Data Warehouse, processos ETL, consultas MDX e um Dashboard. O dashboard criado mostra diversas estatísticas interessantes relativamente ao Basquete universitário entre os anos de 2009 e 2021 incluindo estatísticas avançadas da NBA. Desta maneira, o produto final pode ser utilizado por selecionadores de equipas universitárias para decisões mais informadas relativamente a contratações futuras. Para além disso, o produto final pode dar uma perspetiva geral do desempenho das equipas e dos jogadores para interessados em basquete e para os coordenadores das equipas. Para o grupo o trabalho foi um êxito.