

CEBDSIG 2021/2022 – Criação e Estruturação de Bases de Dados em SIG

Docentes: Patrícia Abrantes e Jorge Rocha

Aluno: Jorge Urbano Pereira, Nº 4687

**Trabalho prático – Relatório**

**Normalização de base dados relacional**

Com base no ficheiro (Base\_de\_dados\_TP\_2021\_2022.xlsx)

Indique os processos de normalização que seriam necessários efetuar para atingir a 3ª forma normal

Indique quais as novas tabelas que criaria, qual a sua estrutura, e construa o esquema relacional das tabelas, indicando o tipo de relações efetuadas e as chaves primárias utilizadas. Caso existam, deve indicar igualmente as chaves candidatas e estrangeiras.

O trabalho deverá focar três pontos:

1. Descrição dos processos de normalização da base de dados
2. Tratamento que seria necessário efetuado à base de dados.
3. Modelo relacional, esquema relacional, tipo de relações e chaves utilizadas

**Objectivos**

Este relatório tem enfoque na descrição e aplicação do processo de normalização de uma base dados relacional. Tendo por base o ficheiro *Base\_de\_dados\_TP\_2021\_2022.xlsx* que é uma tabela que será reconfigurada e subdividida em outras tabelas com a informação reorganizada de modo a ser possível relacionarem-se entre si através de atributos comuns, criando assim a estrutura do modelo relacional dando destaque às chaves utilizadas, ao tipo de relações e por fim ao esquema relacional.

**Procedimentos**

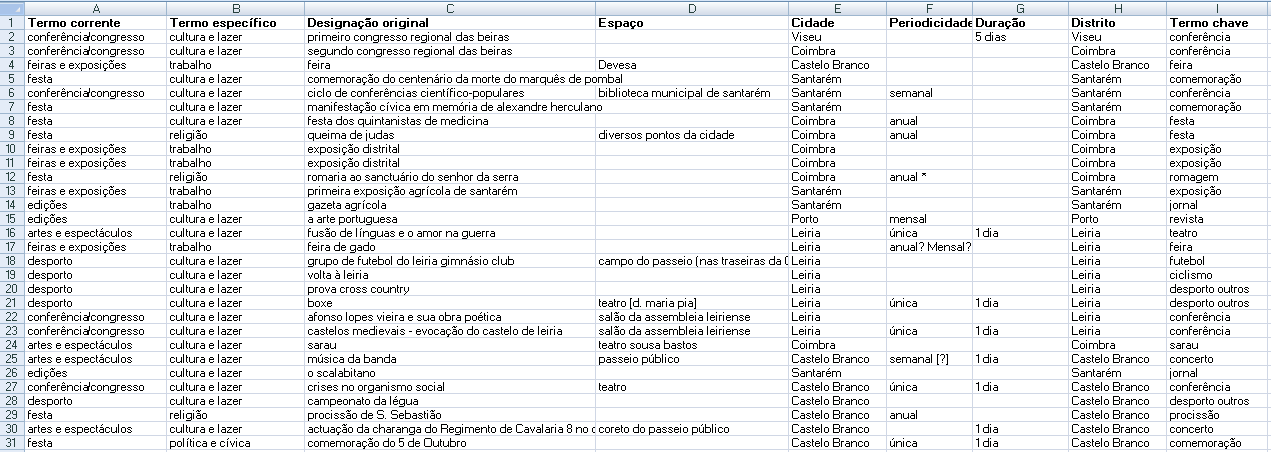
**Preparação e tratamento da tabela**

Antes de iniciar o processo de normalização da base de dados relacional foi necessário em primeiro lugar reconfigurar, reorganizar e tratar os dados da tabela de base dada.

Esta tabela apresenta a seguinte estrutura, é constituida por 9 colunas/ atributos, este valor é o grau da relação, e por 2590 registos/ tuplos sendo este valor a cardinalidade da relação. O *tipo* de todos os nove atributos é *string* destinado para caracteres ou texto

**Actividade** (Termo corrente, Termo específico, Designação original, Espaço, Cidade, Periodicidade, Duração, Distrito, Termo chave)

Quadro 1 – Tabela dada para o exercício



A análise da tabela permite perceber que esta apresenta bastantes situações irregulares, está desorganizada, existem muitos valores em branco, registos duplicados, informação estruturada de forma ineficaz e contém muitos caracteres especiais.

Avaliou-se a percentagem de preenchimento dos registos por atributo que é apresentada no quadro 2 para perceber a quantidade de informaçãode para cada atributo.

Quadro 2 - Percentagem de preenchimento dos registos por atributo

|  |  |
| --- | --- |
| **Atributo** | **%** preenchida |
| Termo corrente | 94,6 |
| Termo específico | 94,6 |
| Designação original | 98,3 |
| Espaço | 40,9 |
| Cidade | 80,1 |
| Periodicidade | 41,4 |
| Duração | 35,5 |
| Distrito | 95,3 |
| Termo chave | 93,8 |

Com esta análise surgem um conjunto de situações relacionadas com os registos, atributos, caracteres especiais e espaços vazios a eliminar e ou a corrigir, e assim sendo, quanto aos registos foram eliminados:

28 registos sem valor para o atributo [Designação original]

23 registos com a palavra *\*APAGAR?!* e *\*FICHA ERRADA!!!*

714 registos duplicados

143 registos com 4 ou menos colunas preenchidas

Quanto aos atributos, foram eliminados:

Por um lado, o atributo [Duração] apresenta a percentagem mais baixa (35,5%) de preenchimento dos registos e a informação apresentada é do tipo “5 Dias”, assim não sendo um atributo do tipo *number* (para valores numéricos) não é possível inquirir a BD para por exemplo saber qual o evento com maior duração.

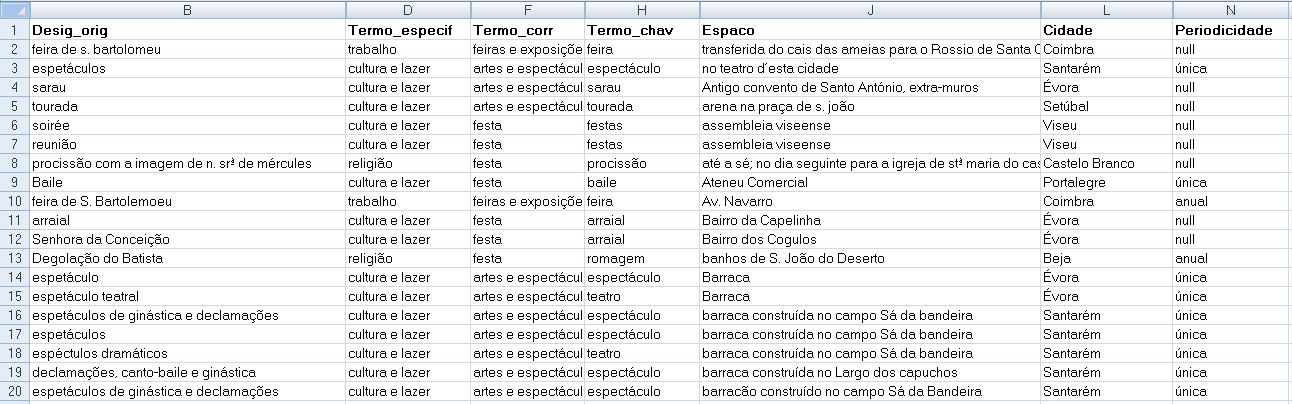
E por outro, os atributos [Distrito] juntamente com [cidade] apresentam percentagens de preenchimento dos registos elevadas com valores muito próximos 95,3% e 80,1% respectivamente, os registos destes dois atributos são repetidos, assim sendo fez-se um *merge* a estes dois atributos e denominou-se [cidade], tendo desta forma a taxa de preenchimento dos registos mais elevada sem prejuízo da integridade dos dados.

Quanto aos caracteres especiais do tipo ( ) “ $ ? ! / \* … [ ] ? foram retirados e por fim a substituição de campos vazios por NULL.

Após este trabalho de tratamento dos dados tabulares, ficamos com uma tabela ou relação que permite iniciar o processo de normalização à luz do modelo relacional. A relação passou a ser denominada *Eventos* (quadro 3) com a seguinte estrutura, é constituída por 7 atributos ou seja este valor é o grau da relação, e por 1682 tuplos sendo este valor a cardinalidade da relação. O tipo de todos os atributos é *string* destinado para caracteres ou texto.

**Eventos** (Desig\_orig, Termo\_especif, Termo\_corr, Termo\_chav, Espaco, Cidade, Periodicidade)

Quadro 3 – Relação Eventos



**Modelo Relacional**

Enumeramos algumas características e conceitos deste modelo, que apresenta-se como uma estrutura tabular composta por várias tabelas com informação distinta que se podem relacionar através de atributos que sejam comuns. É uma estrutura simples que pretende evitar a duplicação.

A estrutura fundamental do modelo relacional é a relação, também designada por tabela. Ou seja, os dados são estruturados em tabelas, formalmente designadas de relações.

Neste modelo as relações/tabelas, as colunas são os campos ou atributos e as linhas representam os registos ou tuplos. Cada registo é unívoco.

Os campos constituem informação elementar, são os atributos da relação e constituem informação que não pode ser decomposta, i.e. tem tantas colunas como o numero de atributos.

Numa relação a ordem dos tuplos e dos atributos é arbitrária para o sistema, ou seja, podemos alterar a ordem sem que afecte o conteúdo da tabela;

Ao número de atributos que constituem o esquema de relação designa-se por grau de relação;

O número de tuplos de uma relação designa-se por cardinalidade da relação.

Cada atributo apresenta um mesmo *domínio*, ou gama de valores possíveis de assumir nesse atributo. O exemplo seguinte é da relação Eventos.

Termo\_especif = {cultura e lazer, mutualidade e beneficência, política cívica, religião, trabalho}

Quadro 4 – Domínio da Relação Termo Específico

|  |  |
| --- | --- |
| **Cod\_TE** | **Termo\_especif** |
| TE01 | cultura e lazer |
| TE02 | mutualidade e beneficência |
| TE03 | política e cívica |
| TE04 | religião |
| TE05 | trabalho |

A intersecção de cada coluna com cada linha da tabela apenas contém um valor.

O valor de alguns atributos num tuplo podem ser desconhecidos ou não existirem, assumindo-se o valor especial designado *null*.

A definição do *tipo* de atributo costuma ser feita mediante a escolha de um entre vários tipos predefinidos, como por exemplo:

1. *string* ou texto – para caracteres ou texto

2. *number* – para valores numéricos

3. *date* – para datas

4. *memo* – para textos um pouco mais extensos

A Base de Dados Relacional utiliza diversas tabelas que podem relacionar-se através das chaves.

**Chave** – consiste em um ou mais atributos que determinam outros atributos de uma relação. Exemplo, se souber o valor de A (DTCC) sei o valor de B (Cidade), mas o contrário não acontece. São definidas para gerir e manipular relações na BD Relacional.

**Superchave** – associação de um ou mais atributos (Chave composta), cujos valores identificam inequivocamente um tuplo (DTCC, CONCELHO, POPULACAO, SOLO URBANO)

**Chave candidata** – subconjunto dos atributos de uma superchave que, sendo ainda uma superchave, não pode ser reduzido sem perder essa qualidade (ex. DTCC ou CONCELHO). É portanto uma candidata a chave primária.

**Chave primária** (ou principal) – chave selecionada de entre as chaves candidatas (ex. DTCC; CONCELHO);

**Chave estrangeira** (ou importada) – conjunto de um ou mais atributos que é chave primária noutra relação.

A existência de uma chave estrangeira numa relação deve-se à necessidade de manter uma ligação (relacionamento) entre essa relação e a relação em que essa chave estrangeira é chave primeira.

**Chave primária**

Chave que é escolhida de entre as chaves candidatas para identificar de forma única cada tuplo de uma relação.

Uma chave primária deve respeitar as seguintes regras:

1. **Ser unívoca** – os campos que desempenham o papel de chave primária, por definição, têm de ter um valor único para cada registo concreto;

2. **Não nula** – nenhum dos campos que formam uma chave primária poderá conter um valor nulo em nenhum registo;

3. **Não redundante** – no caso de uma chave primária ser composta, não devem ser incluídos mais campos do que os mínimos necessários para identificar os registos de modo unívoco. Um campo de uma chave composta não pode ser retirado dessa chave, pois se o for, o campo ou campos restantes deixam de ser unívocos (DTCC, POPULACAO)

**Construção de tabelas**

Para que uma tabela esteja correctamente constituída deve respeitar as seguintes regras:

1. Em toda a base de dados, não pode haver dois campos com o mesmo nome. Cada campo é identificado de modo único;

2. Não devem haver campos vazios. Caso o valor de um campo seja desconhecido ou não aplicável, então deve ser preenchido com um valor null ;

3. O domínio de todos os campos deve ser constituído por valores atómicos. Não é permitido incluir mais do que um valor em cada campo;

4. Cada linha da tabela representa uma entidade ou ocorrência única. Por isso, não podem haver registos duplicados.

**Processo de normalização de uma base de dados relacional**

Processo pelo qual se organiza uma BDR numa estrutura mais simples, minimizando redundâncias.

Quando se cria uma base de dados, temos um conjunto de campos que deverão estar presentes na base de dados.

Esses campos geralmente estão todos em conjunto, pertencendo ou não à mesma tabela.

Com a inclusão de campos normalmente existem dados duplicados e incoerências dentro da base de dados, situação que não se deve verificar.

Os problemas de redundância numa base de dados relacionam-se com:

- Problemas de manutenção (apagar, atualizar)

- Custos de armazenamento

- Desempenho nas consultas/ queries

Para que uma tabela seja útil e haja eficiência num sistema de base de dados relacional, deve-se respeitar um conjunto de regras, designadamente as formas normais.

Estas dizem respeito às relações entre atributos de uma relação. Visam organizar os dados em diferentes tabelas, minimizando redundâncias. Implica esquematizar que tipo de relações e como serão relacionadas.

A normalização converte cada entidade gradualmente para “**Formas Normais**”, através da aplicação sucessivas de regras que alteram o formato dos dados da 1ª Forma normal até à 5ª Forma normal.

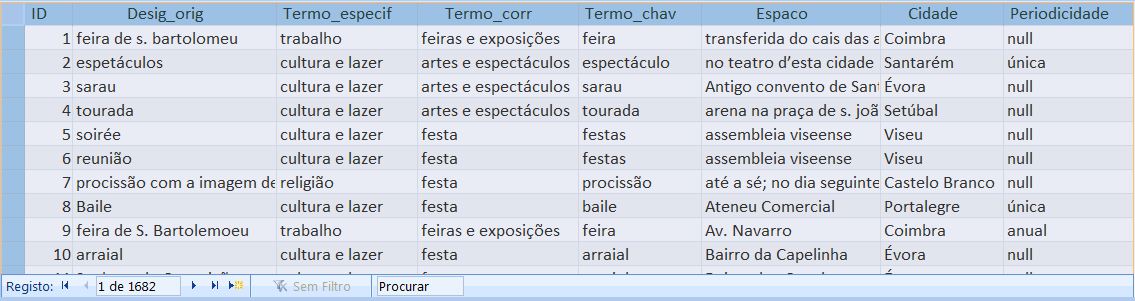
A normalização consiste na subdivisão das relações existentes noutras de menor grau e baseia-se na análise das dependências funcionais (A -> B) entre os atributos de todas as relações envolvidas.

A normalização da base de dados pode ser conseguida dando os seguintes passos:

**Forma não normal** (FNN)

A relação está na forma não normal, que é o momento antes de iniciar o processo de normalização. Foi adicionado um novo atributo, **ID** para se relaciona directamente com o atributo **Desig\_orig** este novo atributo é relevante para o passo seguinte a 1ª forma normal.

Quadro 5 – Relação Eventos com o atributo ID



**Eventos** (ID, Desig\_orig, Termo\_especif, Termo\_corr, Termo\_chav, Espaco, Cidade, Periodicidade)

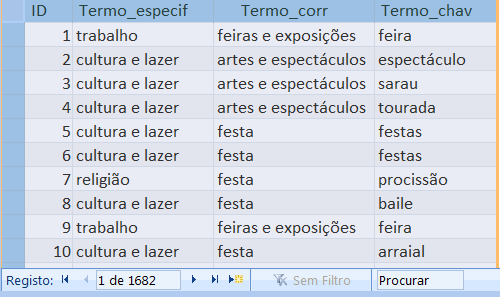
**Primeira forma normal (1FN)**

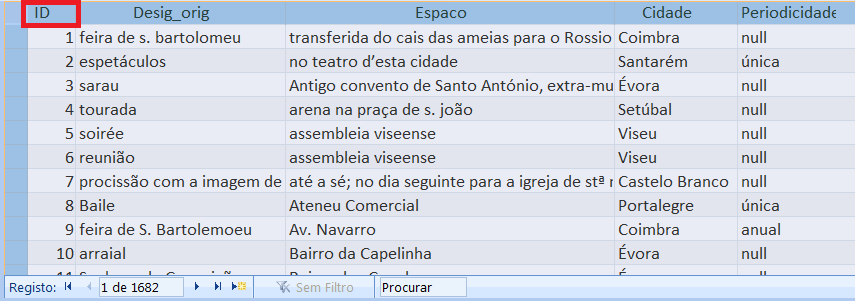
A normalização de uma base de dados na 1ª forma normal exige que:

- todos os seus campos sejam atómicos (um campo representa um único valor de atributo)

- não exista grupos de valores repetidos.

Eventos Termo (temp)





**Eventos** (**ID**, Desig\_orig, Espaco, Cidade, Periodicidade)

Neste relatório teremos as seguintes notações para as chaves utilizadas. A chave primária na **cor vermelho** e a chave estrangeira na **cor verde** e a chave composta a cor de laranja.

O novo atributo **ID** relaciona-se directamente com o atributo **Desig\_orig**, é o número de ordem de cada um dos tuplos do atributo Desig\_orig.

Desta forma o atributo ID obedece às regras para ser chave primária. O atributo Desig\_orig tem valores únicos no entanto são designações muito extensas que levariam a possiveis erros de introdução de novos registos é desta forma não é aconcelhavel ter o papel de chave primária.

A relação **Termo** (temp) (**ID**, Termo\_especif, Termo\_corr, Termo\_chav) é temporária, ou seja, é uma etapa para as seguintes 3 novas relações:

**Termo\_especif** (**Cod\_TE**, Termo\_especif)

**Termo\_corr** (**Cod\_TC**, Termo\_corr)

**Termo\_chav** (**Cod\_TCh**, Termo\_chav)

Com estes procedimentos verificam-se as condições para primeira forma normal. Os registos são atómicos e não há repetição de grupos de valores (por exemplo sempre que se insere novos dados) pois foram criadas três novas relações, como também não há atributos que causam redundância. Estas três últimas relações ligar-se-ão à relação **Eventos** por meio de um tabela de ligação **Tab\_lig**  (ID, Cod\_TE, Cod\_TC, Cod\_TCh).

Nesta situação a chave composta por dois atributos (a chave primária mais a chave estrangeira) apresenta três possibilidades:

**Tab\_lig**  (**ID, Cod\_TE**, Cod\_TC, Cod\_TCh).

A **chave primária** da tabela **Tab\_lig** é composta por dois atributos: ID + Cod\_TE. Só este par identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre os **Eventos** e sobre os **Termo\_especif**. Na relação **Tab\_lig** a chave estrageira Cod\_TE é a chave primária na relação **Termo\_especif.**

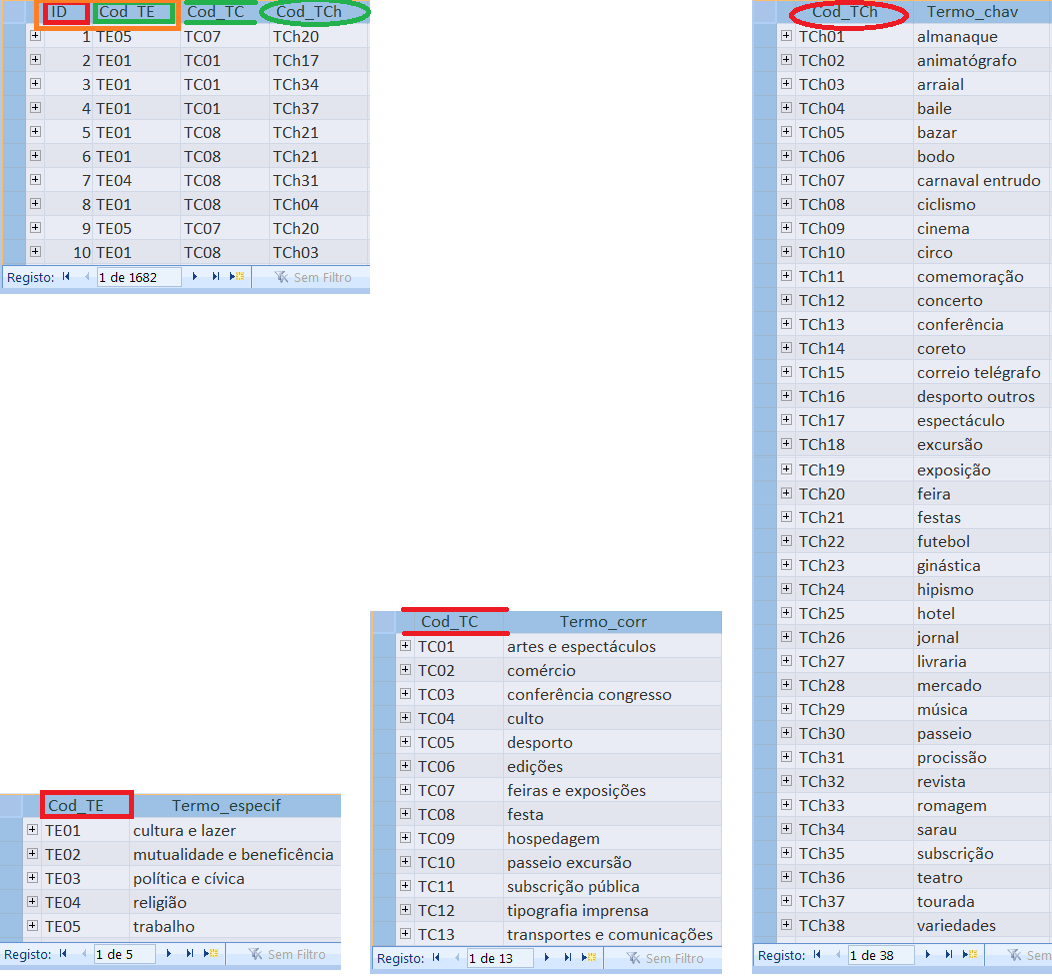
**Tab\_lig**  (**ID,** Cod\_TE, **Cod\_TC,** Cod\_TCh).

A **chave primária** da tabela **Tab\_lig** é composta por dois atributos: ID + Cod\_TC. Só este par identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre os **Eventos** e sobre os **Termo\_corr**. Na relação **Tab\_lig** a chave estrageira Cod\_TC é a chave primária na relação **Termo\_corr.**

**Tab\_lig**  (**ID,** Cod\_TE, Cod\_TC, **Cod\_TCh**).

A **chave primária** da tabela **Tab\_lig** é composta por dois atributos: ID + Cod\_TCh. Só este par identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre os **Eventos** e sobre os **Termo\_chav**. Na relação **Tab\_lig** a chave estrageira Cod\_TC é a chave primária na relação **Termo\_chav.**

**Tab\_lig**  (**ID, Cod\_TE**, Cod\_TC, Cod\_TCh)



**Termo\_especif**  **Termo\_corr**  **Termo\_chav**

No exemplo anterior, o campo produtos, porque pode conter vários tipos de produtos para a mesma unidade industrial, implica a repetição de linhas, apenas diferentes em relação ao

produto.

Então, há necessidade de remover o(s) atributo(s) que viola(m) a primeira forma normal e coloca-los numa nova relação, juntamente com a chave primária da relação original (ID).

A chave primária da nova relação é composta pela chave primária da relação original e por uma chave candidata dos atributos da nova relação.

Os registos são atómicos

Não há repetição de grupos de valores pois foi criada uma nova tabela dos atributos que causavam redundância

A **chave primária** da tabela (Linha\_Produtos) é composta por dois campos: ID + CodProduto. Só este par identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre os produtos e sobre as unidades industriais.

**Segunda forma normal (2FN)**

A normalização de uma base de dados na 2 FN exige que:

- todos os seus campos estejam na 1 FN;

- exista total (todos os campos) dependência da chave primária composta (e não apenas de uma parte);

A normalização de uma base de dados na 2 FN remete para o conceito de dependência funcional. No modelo relacional, todos os atributos que não participam na chave primária dependem desta.

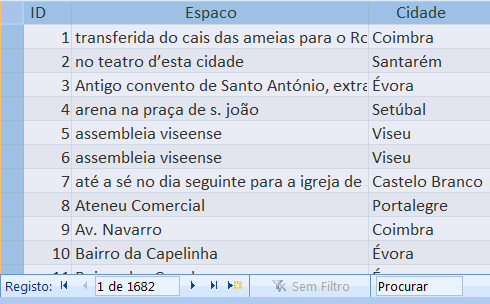
A 2FN exige que todos os seus campos estejam na 1FN e que haja a total dependência da chave primária composta e não apenas de uma parte dessa chave. Isto significa que

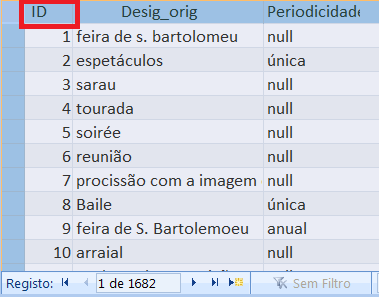
campos parcialmente dependentes ou não dependentes terão que de ser removidos.

Mas será que todos atributos que não pertencem à chave composta dependem funcionalmente desta?

O Atributo Produto é dependente do código de produto (CodProduto) e não do Atributo ID associado à Unidade industrial.

A 2FN passa por criar uma nova tabela com os dados que não dependam exclusivamente da chave composta.





**Eventos** (**ID**, Desig\_orig, Periodicidade) **Localiz (temp)** (**ID**, Espaco, Cidade)

Com estes procedimentos verificam-se as condições para 2ª forma normal.

Para além de estar na 1FN, existe total dependência da chave primária composta, para tal foi criada uma nova relação. Esta relação **Localiz** (**ID**, Espaco, Cidade) ligar-se-à diarectamente à relação **Eventos** (**ID**, Desig\_orig, Periodicidade)

Nesta situação a **chave primária** da relação **Eventos** é o atributo **ID** que identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre os **Eventos** e sobre a **Localiz (temp)** onde o atributo **ID** é **chave estrageira**.

A tabela **Localiz (temp)** (**ID**, Espaco, Cidade) é temporária pois é convertida no seguinte passo de normalização para duas tabelas a **Espaco** (**Cod\_Esp**, Espaco) e a **Cidade** (**DTCC**, Cidade) esta alteração prende-se com as dependencias funcionais entre atributos que não são chave primária.

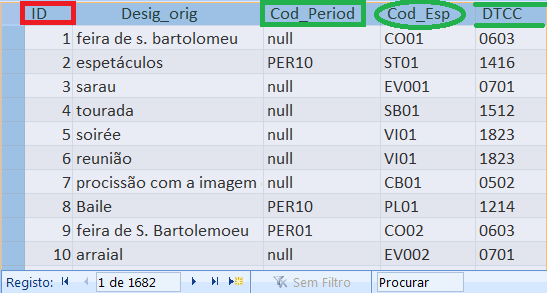
**Terceira forma normal (3FN)**

A normalização de uma base de dados na 3 FN exige que:

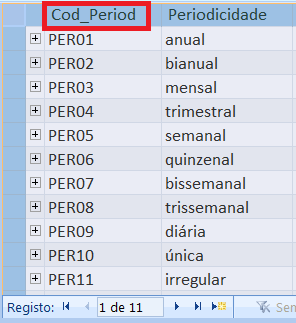
- todos os seus campos estejam na 2 FN

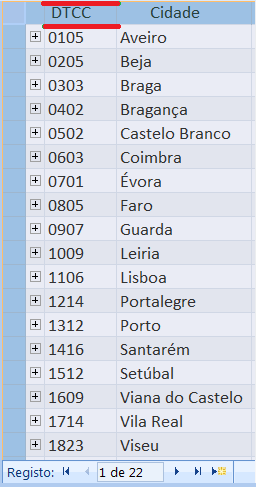
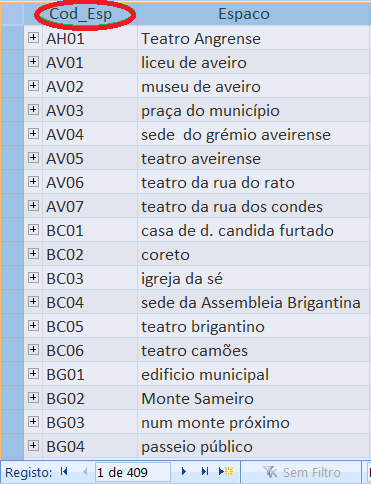
- não existam ***dependências funcionais***entre os atributos que não são chave primária.

**Eventos** (**ID**, Desig\_orig, **Cod\_Period**, **Cod\_Esp**, **DTCC**)



**Espaco** (**Cod\_Esp**, Espaco) **Cidade** (**DTCC**, Cidade) **Periodicidade** (**Cod\_Period**, Periodicidade)





Com estes procedimentos verificam-se as condições para 3ª forma normal.

Para além de estar na 2FN, não existamdependências funcionaisentre os atributos que não são chave primária, para tal foram criadas três novas relações. Estas três últimas relações ligar-se-ão à relação **Eventos** por meio de atributos código com função de **chave primária.** De acordo com isto indicam-se seguidamente as chaves das ligações entre estas relações.

**Eventos** (**ID**, Desig\_orig, **Cod\_Period**, **Cod\_Esp**, **DTCC**)

**Espaco** (**Cod\_Esp** Espaco)

Nesta situação a **chave primária** da relação **Espaco** é o atributo **Cod\_Esp** que identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre o **Espaco** e sobre **Eventos** onde o atributo **Cod\_Esp** é **chave estrageira**.

**Cidade** (**DTCC**, Cidade)

Nesta situação a **chave primária** da relação **Cidade** é o atributo **DTCC** que identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre o **Cidade** e sobre **Eventos** onde o atributo **DTCC** é **chave estrageira**.

**Periodicidade** (**Cod\_Period**, Periodicidade)

Nesta situação a **chave primária** da relação **Periodicidade** é o atributo **Cod\_Period** que identifica de modo unívoco os registos da tabela. A chave primária desta tabela permite obter informação sobre o **Periodicidade** e sobre **Eventos** onde o atributo **Cod\_Period** é **chave estrageira**.

**Tipo de Relações**

**Relações um para um** (1:1): quando um elemento de uma entidade só tem ligação com um elemento de outra entidade. (Por exemplo relação entre o ID de Eventos e o ID de Tab\_lig)

**Relações um para vários** (1:n): quando um elemento de uma entidade tem ligação com vários elementos da outra entidade. (Por exemplo relação entre o Termo\_especif e Cod\_TE)

**Relações de vários para vários** (n:n): quando um elemento da entidade A tem vários elementos na entidade B, e um elemento da entidade B tem vários elementos na entidade A.

Este tipo de relação terá que ser desmontado em duas relações do tipo um para vários, recorrendo a uma chave composta. Este tipo de relação não verifica neste trabalho de normalização.

Com o objectivo da normalização alcançado, o avaliar a qualidade do desenho de tabelas e transformá-lo num desenho de conjunto de tabelas equivalente, menos redundante, mais estável e que permite consultas mais satisfatórias à BDR e também a possibilidade e alterar, apagar e introduzir novos dados de modo ágil.

Esta BDR normalizada possibilita a distribuição espacial dos eventos através de atributos com características espaciais como o código INE para distrito e concelho (DTCC) associado à cidade onde decorre o evento, podemos representar graficamente a sua distribuição num mapa.

Ficheiro Access: TrabPratico\_Database.accdb

<https://github.com/Norge0/CEBDSIG>

**Esquema Relacional**

• Relações 1:n são relacionados de forma que a chave primária do lado “1” seja representada do lado “n” como chave estrangeira;

