

Departamento de Informática

Introdução às Bases de Dados e à Normalização

Prof. Carlos Pampulim Caldeira

Évora, Setembro de 2008

Tabela de Conteúdos

1.	<i>Introdução às Bases de Dados Relacionais</i>	<i>1</i>
1.1	Historial	1
1.2	Definição de Base de Dados Relacional	3
1.3	Inconvenientes dos ficheiros clássicos	4
1.4	Vantagens (e Desvantagens) dos Sistemas de Base de Dados	5
1.5	Desenvolvimento de Bases de Dados	6
1.5.1	Conceptualização de Modelos de Dados	6
2.	<i>Modelo de Dados Relacional</i>	<i>10</i>
2.1	Introdução	10
2.2	Domínios, Relações e Atributos	11
2.2.1	Propriedades das Relações	12
2.3	Diagrama do Modelo de Dados Relacional e Restrições de Integridade	14
2.3.1	Diagrama do Modelo de Dados Relacional	14
2.3.2	Restrições de Integridade nos Diagramas das Bases de Dados Relacionais	14
2.4	Dependências Funcionais	18
3.	<i>Descrição do Processo de Normalização</i>	<i>19</i>
3.1.1	Primeira Forma Normal	19
3.1.2	Segunda Forma Normal	21
3.1.3	Terceira Forma Normal	22

1. Introdução às Bases de Dados Relacionais

1.1 Historial

A evolução do processamento de dados pode ser vista da seguinte forma (Figura 1-1):

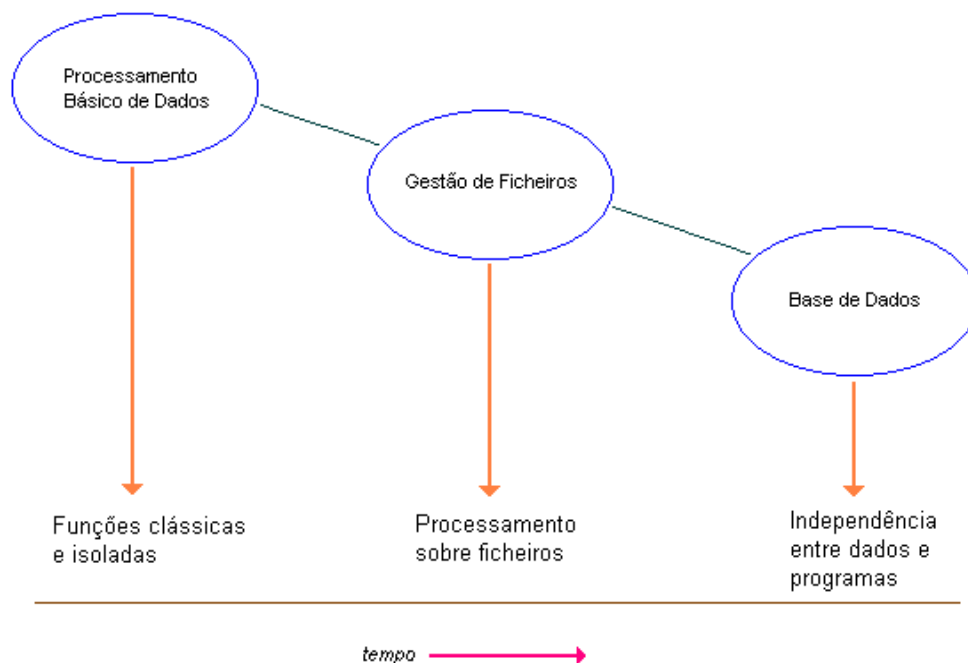


Figura 1-1: A evolução no processamento de dados.

1. **Processamento básico** (ficheiros elementares, anos 50/60). Caracterizou-se por trabalhos isolados de programação; cada programa tinha os seus ficheiros. A manipulação dos dados estava reduzida às funções mais simples: ordenação, classificação, e realização de somatórios.

software pouco mais fazia do que o *input/output* sobre o mecanismo de armazenamento, normalmente numa banda magnética. Qualquer alteração à forma como os dados deveriam estar armazenados, implicava modificações nos programas,

a sua recompilação e teste. A alteração num dado (como por exemplo um novo insecticida contra o escaravelho da batateira) conduzia à criação dum novo ficheiro. O antigo continuava a existir e assim sucessivamente. A grande maioria dos ficheiros era utilizada numa só aplicação. Havia, portanto, um alto nível de redundância, com os mesmos dados multiplicados por n ficheiros.

2. **Gestão de ficheiros** (anos 60/70). Os procedimentos isolados de programação foram integrados em funções. Começaram a aparecer os primeiros casos de partilha de ficheiros entre programas diferentes. Ainda não era possível o acesso aos campos, só aos registos no seu todo. Por esta altura deram-se os primeiros passos, no sentido de isolar as aplicações dos efeitos perversos das alterações de hardware.

Tal como no caso anterior também aqui os ficheiros eram, de uma forma geral, desenvolvidos com um único propósito. Desenvolvia-se, por exemplo, um conjunto de { ficheiros + programas } para a gestão da aplicação de fertilizantes, e outro conjunto com as características desses mesmos fertilizantes. Muita da informação estava repetida e era incoerente entre os ficheiros, tendo que haver vários programas com finalidades bastante semelhantes.

3. **Gestão de base de dados** (a partir dos anos 80). Nasceram os sistemas de gestão de base de dados que gerem os dados independentemente dos programas. As tabelas podem ser alteradas sem que isso obrigue à recompilação de todos os programas.

A noção de modelo de dados tornou-se essencial para o desenvolvimento de bases de dados. Aos dados passaram a ser aplicados dois níveis de independência, a *lógica* e a *física*.

A *independência lógica* significa que a estrutura lógica dos dados pode ser alterada sem consequências a nível de todos os programas. Por exemplo: adicionar novos campos a uma tabela, ou criar uma nova tabela.

A *independência física* verifica-se quando a organização física dos dados pode ser alterada sem que isso acarrete uma modificação global na estrutura lógica dos dados

e nos programas. Por exemplo: adicionar uma nova chave a uma tabela, ou distribuir a base de dados por dois ou mais computadores.

A independência lógica é a mais difícil de atingir dado que os programas são altamente dependentes de estrutura lógica.

1.2 Definição de Base de Dados Relacional

O termo base de dados está intimamente associado à noção de “uma colecção de informação”. De um ponto de vista mais teórico pode-se afirmar que uma base de dados é um conjunto estruturado de informação. Uma base de dados é uma colecção de dados formalmente definida, informatizada, partilhável e sujeita a um controlo central.

Uma base de dados é uma colecção de dados *inter-relacionados* com múltiplas utilizações. Uma base de dados relacional (daqui para diante a expressão *base de dados* é usada como sinónimo de *base de dados relacional*) é um sistema de gestão de informação relativamente complexo.

Dado que a base de dados é a componente central do sistema, uma boa técnica de desenho é crucial para a eficácia do sistema.

Se a função duma base de dados fosse simplesmente a de armazenar dados, a sua organização seria relativamente simples. A complexidade estrutural das bases de dados resulta do facto de que ela deve também mostrar as relações que existem entre os dados.

Uma base de dados é composta por um conjunto de tabelas e associações entre as tabelas. A associação entre os dados é o ponto forte dos sistemas relacionais. As tabelas são formadas por linhas e colunas onde figuram os dados. Numa base de dados relacional os dados estão todos representados como valores nas colunas das tabelas.

Neste tipo de aplicação os dados e os programas estão completamente separados. Já o mesmo não se passa, por exemplo, nas folhas de cálculo em que os dados e procedimentos estão frequentemente misturados.

Uma vantagem importante da *tabela* resulta do facto duma tabela poder ter mais do que uma finalidade e dos seus dados poderem ser *vistos* com diferentes formas e formatos, ao contrário de um ficheiro.

Os sistemas de gestão de bases de dados relacionais (SGBDR) são aplicações informáticas complexas, mas essenciais em muitas áreas científicas, nomeadamente em Agronomia, onde grandes quantidades de informação necessitam de ser combinadas ou exploradas, de diversas formas nem todas fáceis de prever.

1.3 Inconvenientes dos ficheiros clássicos

Um sistema de ficheiros clássicos tem os seguintes inconvenientes:

- **Redundância e inconsistência na informação.** A mesma informação aparece por vezes duplicada devido à forma desorganizada de criar ficheiros e programas. Veja-se, por exemplo, o caso em que a composição de um herbicida pode estar registada simultaneamente nos ficheiros “HERBICIDAS” e “HERBICIDAS MILHO”. Esta redundância é uma fonte potencial de inconsistência nos dados. A inconsistência pode verificar-se, por exemplo, quando a composição do herbicida A é modificada, e essa alteração só é feita no ficheiro “HERBICIDAS MILHO”.
- **Dificuldade em aceder à informação.** Nos ficheiros não é fácil obter-se a informação que se quer, não só porque não se sabe onde está, bem como é preciso programar para a pôr à disposição dos utilizadores. É um processo fortemente dependente de pessoal especializado e no qual os utilizadores não se revêm.

- **Dados isolados.** Em consequência da distribuição dos dados em múltiplos ficheiros, a mesma informação pode estar com formatos diferentes em cada um dos sítios. Isto torna particularmente difícil a programação de aplicações.
- **Integridade da informação.** Os valores que os dados podem assumir estão normalmente sujeitos a certos tipos de *restrições de integridade* como por exemplo: uma plantação não pode ter “-2”; ou não se deve aplicar mais de 2.25 Kg de um dado fungicida por hectare. Num ficheiro esta regra só pode ser assegurada através de linhas de código suplementares em cada um dos programas que aceda àqueles dados. Se em vez de uma, forem três ou quatro regras, então a dificuldade para conseguir manter a integridade dos dados aumenta exponencialmente.

1.4 Vantagens (e Desvantagens) dos Sistemas de Base de Dados

As principais vantagens dum SGBDR, face a um vulgar sistema de ficheiros, são:

- **Resposta rápida aos pedidos de informação.** Como os dados estão integrados numa única estrutura (a base de dados) a resposta a questões complexas processa-se mais velozmente.
- **Acesso múltiplo.** O software de gestão de base de dados permite que os dados sejam acedidos de diversíssimas maneiras. Nomeadamente, os dados podem ser visualizados através de pesquisas sobre qualquer um dos campos da tabela.
- **Flexibilidade.** Em consequência da independência entre dados e programas, qualquer alteração num desses elementos não implica modificações drásticas no outro.
- **Integridade da informação.** Dada a absoluta exigência de não permitir a redundância, as modificações de dados são feitas num só sítio, evitando-se assim possíveis conflitos entre diferentes versões da mesma informação.

- **Melhor gestão da informação.** Em consequência da localização central dos dados, sabe-se sempre como e onde está a informação.

A principal, e mais significativa, desvantagem dos sistemas de gestão de base de dados é o seu custo, não tanto em termos de preço do software de base, mas fundamentalmente em despesas de desenvolvimento. É um tipo de software altamente sofisticado que requer, para o seu desenho e desenvolvimento, pessoal com uma formação adequada.

Os custos resultantes duma má conceptualização são enormes. A construção duma base de dados deficiente tem consequências nefastas numa organização. Tanto em custos directos, como em termos psicológicos, ficando na memória da organização uma animosidade contra as “geniais” inovações tecnológicas.

1.5 Desenvolvimento de Bases de Dados

1.5.1 Conceptualização de Modelos de Dados

Uma base de dados é um *modelo* das necessidades em informação do utilizador final, i.e., do sistema real.

Por forma a atingir esse objectivo, torna-se imprescindível a construção de um modelo de dados onde serão explicitamente referidos todos os temas e sua estrutura, incluindo as relações existentes entre eles.

Os modelos de dados não se referem apenas a dados. São eles próprios feitos de dados. Da mesma forma que o modelo de um tractor pode ser metálico, tal como o seu original, assim também um modelo de dados é constituído por dados.

O desenvolvimento do modelo conceptual é, sem dúvida, o passo mais importante na criação de uma base de dados relacional. Este passo deve ser sempre a primeira etapa do projecto.

As duas propriedades mais importantes a que o esquema conceptual deve obedecer são as seguintes:

1. Simplicidade

Simplicidade compreende a facilidade de se deixar compreender e manipular, devendo obedecer a algumas condições:

- O número de objectos básicos deve ser pequeno.
O esquema conceptual deve ser construído com recurso a somente meia dúzia de conceitos e objectos.
- Não se devem misturar nem agrupar os conceitos.
Para que um conceito seja claro não deve ser utilizado para aglutinar outros objectos ou conceitos.
- A redundância deve ser sempre eficazmente controlada.
Não deve ser permitido que o mesmo facto seja mencionado em dois ou mais lugares. Um facto representa a associação entre uma *entidade* e alguma das suas *propriedades*. Por exemplo, a associação entre a batata-semente e o seu preço numa certa data.

2. Base Teórica

Dada a importância deste nível conceptual, é essencial que ele tenha como suporte um conjunto sólido de conceitos.

A abordagem relacional satisfaz globalmente todos esses requisitos.

- Toda a informação é representada por um só conceito, a *relação*. A *relação* é um conceito familiar ao utilizador - há várias centenas (talvez milhares) de anos que as pessoas utilizam tabelas. A relação é um conceito sólido facilmente manipulável.
- A teoria da normalização (conjunto de normas associadas ao modelo relacional) garante que um mesmo *facto* não apareça em mais de um sítio.

O desenvolvimento de uma base de dados é uma tarefa complexa, dado que não existe um único grupo de especificações¹, mas vários conjuntos que se sobrepõem uns aos outros.

A Figura 1-2 mostra uma descrição simplificada do processo de desenho de uma base de dados. O primeiro passo centra-se na recolha e análise das especificações. Para isso, a equipa de desenvolvimento conduzirá uma série de entrevistas aos utilizadores, e procurará reunir toda a documentação existente sobre as actividades em estudo.

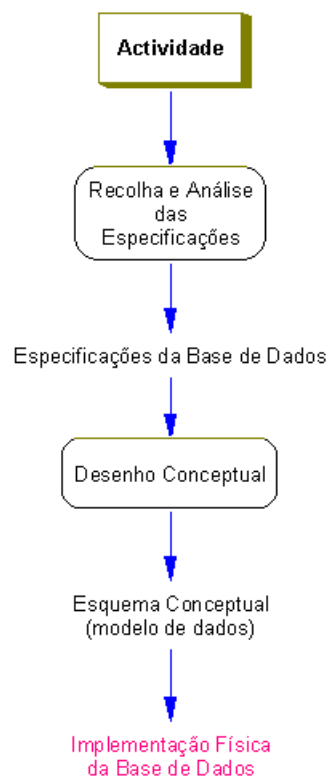


Figura 1-2: Processo de desenho duma base de dados.

Depois da análise das especificações, o passo seguinte é na direcção da criação de um esquema conceptual da futura base de dados. Este esquema será estabelecido através de um modelo de dados conceptual de alto nível. O

¹ Aquilo que o utilizador final pretende que a base de dados *faça*.

esquema conceptual é uma descrição rigorosa das necessidades (em dados e informação) dos utilizadores.

A última etapa consiste na construção física da base de dados através de um sistema de gestão de base de dados.

2. Modelo de Dados Relacional

2.1 Introdução

O modelo de dados relacional é uma ferramenta de modelação importante porque os seus conceitos básicos são simples e gerais e porque o seu desenho não depende de nenhum tipo de programa informático.

O objecto fundamental do modelo relacional, como o seu próprio nome indica, é a relação. O modelo relacional apresenta os dados como um conjunto de relações e tem um sólido fundamento teórico, com base na Teoria Matemática dos Conjuntos e na Álgebra Relacional.

Uma relação pode ser visualizada como uma tabela (Figura 2-2; pág. 14). De toda a forma, deve-se ter sempre o cuidado de tomar em consideração que a tabela é, nem mais nem menos, o modelo físico dum conceito matemático: a relação. A tabela é uma forma familiar de representação de informação. Veja-se os horários dos transportes públicos, ou por exemplo as estatísticas agrícolas do Instituto de Estruturas Agrárias e Desenvolvimento Rural (Quadro 2-1).

Quadro 2-1: Máquinas Agrícolas - 1994.

	Tractores	Ceifeiras Debulhadoras	Motoceifeiras	Moto-cultivadoras	Motoenxadas
CONTINENTE	147 882	3 996	30 036	48 455	14 894
Entre Douro e Minho	27 643	5	13 897	9 528	765
Trás-os-Montes	17 521	314	3 324	1 203	140
Beira Litoral	31 122	318	8 702	11 330	6 477
Beira Interior	12 402	139	1 447	18 397	5 842
Ribatejo e Oeste	32 030	654	1 474	18 397	5 842
Alentejo	19 216	2 535	668	1 198	151
Algarve	7 948	31	524	1 383	830

O modelo de dados relacional tem ainda as seguintes vantagens:

- é independente das linguagens de programação;
- é independente dos sistemas de gestão de bases de dados;
- e é independente dos sistemas operativos².

2.2 Domínios, Relações e Atributos

O modelo relacional baseia-se em três conceitos básicos: domínio, relação e atributo.

O domínio é um conjunto de valores que possuem determinadas propriedades em comum; ao conjunto de todos os valores possíveis para um determinado atributo dá-se o nome de domínio. O domínio engloba dados *atómicos* ou *simples*, porque não podem sofrer mais nenhuma decomposição.

Uma relação representa um conjunto de objectos dum tipo particular. Os objectos que pertencem à relação são, no essencial, os elementos que obedecem às propriedades da relação. Assim, a relação é um conjunto com propriedades próprias, em que os seus elementos se designam por *tuplas* ou linhas.

A relação pode-se definir da seguinte forma: dados os conjuntos C_1, C_2, \dots, C_n , R é uma relação sobre esses n conjuntos se for um conjunto de n -tuplas, em que o seu primeiro elemento provém de C_1 , o segundo elemento de C_2 e assim sucessivamente. Isto é: R é um subconjunto do produto cartesiano $C_1 \times C_2 \times \dots \times C_n$.

Uma relação define-se sobre um certo número de domínios e engloba duas componentes: o cabeçalho e o corpo. O cabeçalho é um conjunto de atributos em que cada atributo corresponde a um domínio. O corpo corresponde a um conjunto de tuplas (mais precisamente ' n -tuplas' em que n é o número total de atributos). Cada tupla é um conjunto de valores: um por cada atributo constante do cabeçalho.

² Conjunto de programas de controlo que fazem a gestão dos recursos de um computador.

Uma relação engloba elementos (atributos) provenientes de um ou mais conjuntos. Os conjuntos numa relação funcionam como domínios, fornecem o intervalo de valores que cada um dos seus atributos pode assumir. Retomando a analogia entre relação e tabela, os termos atributo e coluna são vulgarmente aceites como sinónimos.

Cada linha da tabela, representa uma proposição sobre determinada entidade. Enquanto que as linhas representam os valores actuais dos atributos, os domínios simbolizam todos os valores possíveis. As linhas variam com as circunstâncias enquanto que os domínios e os atributos são constantes.

Uma relação representa-se formalmente através do seu *esquema*:

O esquema da relação **R**, que se escreve **R(A₁, A₂, ..., A_n)**, é um conjunto de atributos **R** = {**A₁, A₂, ..., A_n**}, em que cada atributo **A_i** tem o nome do papel desempenhado por um domínio **D** no esquema **R**; **D** denomina-se domínio de **A_i** e tem a notação **dom(A_i)**. O esquema duma relação serve para descrever a relação em que **R** é o nome da relação e **A₁, A₂, ..., A_n** são os nomes dos atributos. Na Figura 2-1. apresenta-se o exemplo do esquema da relação “Cultivar”:

CULTIVAR (nome da cultivar, código da cultivar, representante legal, código da cultura)

Figura 2-1: Esquema da relação “CULTIVAR”.

Uma relação pode conter qualquer número de linhas e de atributos. Se a relação tiver um único atributo ela diz-se ‘*unária*’; se tiver dois então é ‘*binária*’, e assim sucessivamente. O número de atributos designa-se por grau da relação. O total de linhas é a sua cardinalidade.

2.2.1 Propriedades das Relações

Uma relação caracteriza-se por:

- Ter um nome único dentro do mesmo diagrama de modelo de dados relacional;

- Ter de zero a n linhas, cuja ordenação é indiferente dado que não são identificadas pela sua posição. Uma relação sem nenhuma linha diz-se vazia.
- Ser composta por um ou mais atributos, onde a ordem não é importante, pois identificam-se pelo nome e não pela sua posição;
- Cada um dos atributos contém valores retirados dum domínio particular, o que quer dizer que num mesmo atributo os dados são obrigatoriamente todos do mesmo tipo;
- Numa mesma relação não podem existir dois atributos com o mesmo nome;
- Cada relação tem que ter uma *chave primária*. Uma chave primária é um atributo, ou combinação de atributos, cujos valores proporcionam uma identificação unívoca das tuplas numa relação, ou seja, um certo valor para a chave só pode aparecer uma única vez em cada relação. Significa ainda que as tuplas numa relação são todas diferentes entre si, i.e., não são permitidas linhas duplicadas. Considerando a relação representada na Figura 2-2 não faz qualquer sentido que seja permitida a duplicação de uma linha dessa relação, pois os dados perderiam a sua coerência;
- A intersecção de uma coluna com uma linha corresponde a um dado ou valor. Não é permitida a existência de grupos de atributos repetidos;
- Os dados, ou valores, são sempre de tipo atómico.

Uma tabela que cumpra estes requisitos pode-se considerar equivalente a uma relação. Estas tabelas, de tipo especial, poderão receber a designação de R-tabelas. Uma base de dados relacional é composta por um conjunto de tabelas deste tipo muito particular.

TRACTOR AGRÍCOLA				
Designação do Tractor	Marca	Modelo	Potência	
Tractor de Lagartas	New Holland	55-85 SOM	55	
Tractor Vinhateiro	John Deere	5500N	73	
Ceifeira-Debulhadora	Massey Ferguson	8560	190	
Tractor Clássico	New Holland	7610S	95	

Figura 2-2: Exemplo duma tabela.

2.3 Diagrama do Modelo de Dados Relacional e Restrições de Integridade

2.3.1 Diagrama do Modelo de Dados Relacional

O diagrama dum modelo de dados relacional contém, normalmente, numerosas relações que se encontram associadas entre si através de *tuplas* comuns. O diagrama dum modelo de dados relacional (DMDR) é um conjunto de esquemas de relações sujeitos a um conjunto de restrições de integridade.

2.3.2 Restrições de Integridade nos Diagramas das Bases de Dados Relacionais

Há três tipos de restrições de integridade que se especificam sobre os diagramas de modelos de dados relacionais: 1) a chave de uma relação, 2) a integridade da tabela e 3) a integridade referencial. As regras ou restrições de integridade asseguram que o modelo de dados reflecta adequadamente a realidade, sem qualquer ambiguidade ou redundância.

2.3.2.1 A Chave de uma Relação

A *chave* é um atributo cujo valor identifica uma e uma só linha numa relação. Por vezes essa identificação unívoca só é possível através duma *chave composta* que consiste na menor combinação de atributos com essa propriedade.

Uma relação tem sempre, pelo menos, uma chave: a *chave primária* que é escolhida do conjunto das chaves candidatas. Para a inserção na relação de uma linha de dados ter êxito, é obrigatório fornecer os valores para os atributos componentes da chave primária. Os valores dos atributos que não pertencem àquela chave podem não ser conhecidos na altura da inserção e podem ser representados por *valores nulos*. A construção duma chave primária não traz qualquer tipo de limitação no acesso aos dados, pois as linhas continuam a poder ser encontradas através da indicação de valores para qualquer um dos seus atributos. Como no modelo relacional a associação da informação se faz através da comparação de valores contidos nos atributos das relações, a chave primária é a única forma de se fazer associação de informação sem que haja lugar a qualquer tipo de ambiguidade.

Uma relação é definida como um conjunto de tuplas. Por definição, todos os elementos de um conjunto são distintos, pelo que as tuplas têm que ser necessariamente diferentes entre si. Isto quer dizer que numa relação não pode haver duas tuplas com a mesma combinação de valores para todos os seus atributos. No entanto, e em termos práticos existe sempre um subconjunto de atributos que não se repetem. A essa combinação de atributos dá-se o nome de super-chave da relação *R*.

Uma chave é uma super-chave a que não se pode retirar nenhum atributo sob pena de já não ser assegurada a integridade dos dados.

Quadro 2-2: Relação ALUNO

ALUNO	Nº	Nome	Nº B.I.	Morada

Qualquer combinação de atributos que inclua o número de aluno (por exemplo: n° + nome + morada) é uma super-chave. No entanto, não é uma chave pois $\{n^\circ, \text{nome}, \text{morada}\}$ não é uma chave pois removendo o nome ou a morada, ou ambos, continua-se a ter uma super-chave.

Um esquema relacional pode ter mais de uma chave, a que se dá o nome de chave candidata. A relação ALUNO tem duas chaves candidatas: N° e N° B.I. a partir das quais se escolhe uma como chave primária da relação. Quando um esquema relacional tem várias chaves candidatas a escolha deve recair sobre aquela que é composta pelo menor número de atributos ou que melhor se insira no ambiente da relação.

2.3.2.2 A Integridade da Relação

O(s) atributo(s) que compõem uma chave primária não podem conter o valor *nulo*³, pois numa relação dum diagrama de modelo de dados relacional utilizam-se os valores constantes nesses atributos para identificar as respectivas linhas. Se, por exemplo, duas ou mais linhas tiverem o valor nulo nos atributos não seria possível distingui-las entre si. Por definição, um *valor nulo* não tem capacidade de identificação, pelo que uma chave primária nunca pode ser nula. Se a chave for composta nenhum dos seus elementos pode ter o valor nulo, o cumprimento desta regra assegura que é sempre possível identificar cada um das tuplas numa relação.

2.3.2.3 A Integridade Referencial

A chave primária e a integridade da relação são especificadas individualmente em cada relação. A integridade referencial é uma restrição estabelecida entre duas relações com o objectivo de manter a consistência dos dados entre as tuplas dessas relações. De um modo simples, pode-se dizer que a integridade referencial determina que uma linha numa relação **A** que está referida a uma outra relação **B** tem que obrigatoriamente estar ligada a uma linha existente na relação **B**. Veja-se, por exemplo, a Figura 2-3 em que a relação

³ Tome-se em consideração que nulo é diferente de zero, significando que o valor não é conhecido.

“INIMIGO DA CULTURA” contém o atributo “Código da Cultura” para cada praga e cujo valor em cada uma das tuplas de “INIMIGO DA CULTURA” tem que coincidir com o valor do atributo “Código da Cultura” na relação “CULTURA”.

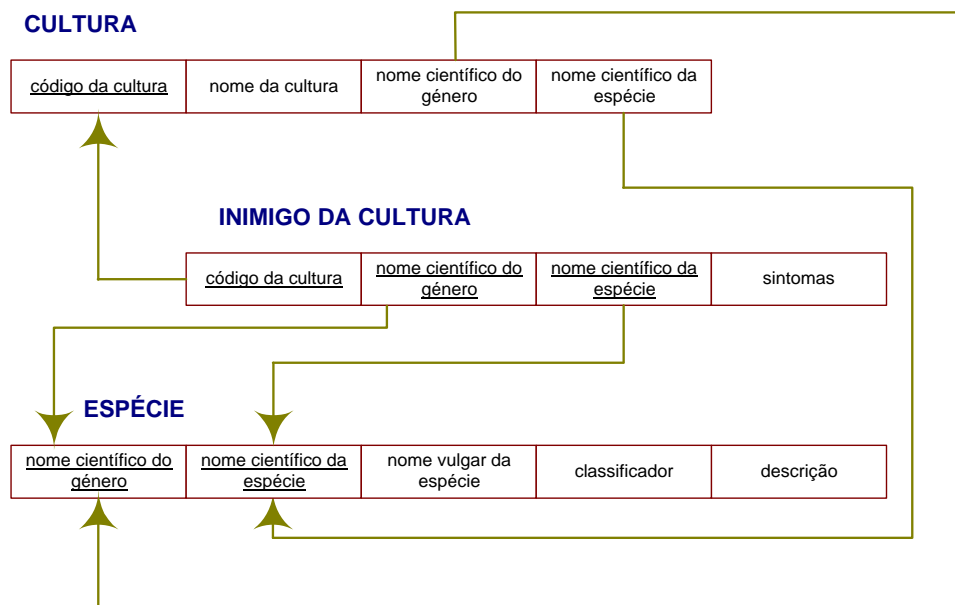


Figura 2-3: O DMDR "PRAGAS" com inclusão de algumas regras de integridade.

A associação entre duas relações faz-se através da chamada *chave estrangeira*, que é um atributo, ou grupo de atributos, que desempenham o papel de chave primária noutra relação, servindo portanto de elo de ligação entre elas. Dependendo das situações é possível que uma chave estrangeira aceite valores nulos, o que significa que essa linha não está referida à chave da outra relação.

A Figura 2-3 contém um DMDR com a indicação das chaves primárias (atributos sublinhados) e das restrições de integridade referencial (arcos com as setas apontando para a relação de nível superior, i.e, a chave estrangeira aponta no sentido do(s) atributo(s) de referência).

No esquema de uma relação a chave primária apresenta-se sublinhada e o nome de um atributo que seja uma chave estrangeira é antecedido pelo sinal “#” (Figura 2-4).

CULTIVAR (nome da cultivar, código da cultivar, representante legal, #código da cultura)

Figura 2-4: Esquema da relação “CULTIVAR” com a indicação da chave primária e de uma chave estrangeira.

2.4 Dependências Funcionais

A dependência funcional (DF) é um dos conceitos fundamentais no desenho dos modelos de dados relacionais. A dependência funcional é uma associação que se estabelece entre dois ou mais atributos duma relação e define-se do seguinte modo: Se **A** e **B** são atributos, ou conjuntos de atributos, da relação **R**, diz-se que **B** é funcionalmente dependente de **A** se cada um dos valores de **A** em **R** tem associado a si um e um só valor de **B** em **R**; a DF tem a notação: **A** → **B**

Na Figura 2-5 é apresentada a notação gráfica relacionada com a dependência funcional. A dependência funcional é representada por uma linha horizontal que parte do(s) atributo(s) mais à esquerda, terminando com setas nos atributos dependentes, localizados à direita. Todos os atributos que não fazem parte da chave primária de uma relação são funcionalmente dependentes dela.

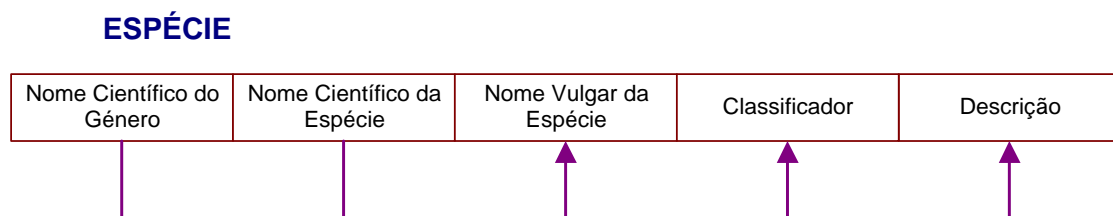


Figura 2-5: As dependências funcionais na relação “ESPÉCIE”.

3. Descrição do Processo de Normalização

3.1.1 Primeira Forma Normal

Uma relação diz-se na primeira forma normal quando todos os atributos contêm unicamente valores singulares e não existem grupos de atributos repetidos.

Quadro 3.1: Tabela com dados sobre tractores agrícolas.

Marca	Designação	Modelo	Concessionário	Concessionário	Potência
New Holland	Tractor de Lagartas	55-8550M	AGROMAQ	RIOMAQ	55
	Tractor Clássico	7610S	“	“	95
	Tractor Clássico	8970	“	“	240
John Deere	Tractor Vinhateiro	5500N	AGROMAQ	TRACTAGRI	73
	Tractor Clássico	9500	“	“	190
Massey Ferguson	Ceifeira-Debulhadora	8560	RIOMAQ	V-MAQ	190
	Tractor Clássico	MF 292	“	“	105

Chave Primária: “Marca”.

A tabela do Quadro 3.1 não é uma relação pois viola algumas das normas do modelo de dados relacional:

- tem um atributo repetido (“Concessionário”);
- algumas colunas pertencentes à chave primária contêm o valor nulo;
- as colunas e linhas só podem ser lidas na sequência apresentada;
- alguns dos atributos que não são chave não dependem funcionalmente da chave primária;

- alguns atributos que não fazem parte da chave são funcionalmente dependentes de outros atributos que também não pertencem à chave primária.

Quadro 3.2: Relação “TRACTOR AGRÍCOLA” na Primeira Forma Normal.

Marca	Designação	Modelo	Concessionário	Potência
New Holland	Tractor de Lagartas	55-8550M	AGROMAQ	55
New Holland	Tractor Clássico	7610S	AGROMAQ	95
New Holland	Tractor Clássico	8970	AGROMAQ	240
New Holland	Tractor de Lagartas	55-8550M	RIOMAQ	55
New Holland	Tractor Clássico	7610S	RIOMAQ	95
New Holland	Tractor Clássico	8970	RIOMAQ	240
John Deere	Tractor Vinhateiro	5500N	AGROMAQ	73
John Deere	Tractor Clássico	9500	AGROMAQ	190
John Deere	Tractor Vinhateiro	5500N	TRACTAGRI	73
John Deere	Tractor Clássico	9500	TRACTAGRI	190
Massey Ferguson	Ceifeira-Debulhadora	8560	RIOMAQ	190
Massey Ferguson	Tractor Clássico	MF 292	RIOMAQ	105
Massey Ferguson	Ceifeira-Debulhadora	8560	V-MAQ	190
Massey Ferguson	Tractor Clássico	MF 292	V-MAQ	105

Chave Primária: “Marca” + “Modelo”.

No Quadro 3.2 mostra-se uma estrutura de dados proveniente do Quadro 3.1 já na primeira forma normal: 1) cada ocorrência foi transformada numa linha; 2) a chave primária foi modificada passando a incluir as colunas “Marca” e “Modelo”; 3) o atributo repetido foi eliminado.

3.1.2 Segunda Forma Normal

Uma relação na primeira forma normal poderá ter ainda problemas, por exemplo, de duplicação de dados, pelo que deverá ser imediatamente sujeita às regras da segunda forma normal, que se baseia no conceito de *total dependência funcional* sobre a chave primária.

A segunda forma normal tem a ver com as relações entre os atributos chave, e os restantes atributos duma relação (atributos não chave). Um atributo não chave só pode fornecer informação sobre o conjunto completo dos atributos chave e só sobre eles. Uma relação R diz-se na segunda forma normal quando todos os atributos não chave são total e funcionalmente dependentes da chave primária de R . A segunda forma normal (2FN) é violada quando um atributo não chave contém informação sobre um sub-conjunto da chave primária.

A tabela que representa a relação do Quadro 3.2 não está na 2FN dado que o atributo “Concessionário” depende funcionalmente apenas de parte da chave primária: do campo “Marca”.

A dependência funcional $X \rightarrow Y$ é total se a remoção de um qualquer atributo A de X implicar o fim da dependência.

Isto é: para qualquer atributo $A \in X$, $(X - \{A\}) \not\rightarrow Y$.

A dependência funcional $X \rightarrow Y$ é parcial se existir um qualquer atributo $A \in X$ que possa ser removido de X mantendo-se a dependência.

Isto é: para qualquer atributo $A \in X$, $(X - \{A\}) \rightarrow Y$.

A dependência funcional referida no parágrafo anterior implica a decomposição dessa relação em duas outras (Quadro 3.3 e Quadro 3.4) cada uma das quais satisfaz as restrições inerentes à 2FN.

Quadro 3.3: Relação “CONCESSIONÁRIO”.

Marca	Concessionário
New Holland	AGROMAQ
John Deere	AGROMAQ
New Holland	RIOMAQ
John Deere	TRACTAGRI
Massey Ferguson	V-MAQ
Massey Ferguson	RIOMAQ

Chave Primária: “Marca” + “Concessionário”.

Quadro 3.4: Relação “TRACTOR AGRÍCOLA” na Segunda Forma Normal.

Marca	Designação	Modelo	Potência
New Holland	Tractor de Lagartas	55-8550M	55
New Holland	Tractor Clássico	7610S	95
New Holland	Tractor Clássico	8970	240
New Holland	Tractor de Lagartas	55-8550M	55
New Holland	Tractor Clássico	7610S	95
New Holland	Tractor Clássico	8970	240
John Deere	Tractor Vinhateiro	5500N	73
John Deere	Tractor Clássico	9500	190
John Deere	Tractor Vinhateiro	5500N	73
John Deere	Tractor Clássico	9500	190
Massey Ferguson	Ceifeira-Debulhadora	8560	190
Massey Ferguson	Tractor Clássico	MF 292	105
Massey Ferguson	Ceifeira-Debulhadora	8560	190
Massey Ferguson	Tractor Clássico	MF 292	105

Chave Primária: “Marca” + “Modelo”.

3.1.3 Terceira Forma Normal

A terceira forma normal (3FN) verifica-se quando um atributo que não é chave é um facto sobre outro atributo que também não pertence à chave primária da relação. A 3FN baseia-se no conceito de *dependência transitiva*.

A dependência funcional $X \rightarrow Y$ no esquema relacional R é uma *dependência transitiva* se existir um conjunto de atributos Z que não faça parte do sub-conjunto de qualquer chave de R , e ambos $X \rightarrow Z$ e $Z \rightarrow Y$ se mantêm.

No Quadro 3.5 a dependência “Adubo” \rightarrow “Custo Unitário” afigura-se como uma dependência transitiva, pois a dependência de “Custo Unitário” na chave primária é transitiva via “Adubo”.

Quadro 3.5: Relação “ADUBAÇÃO”.

Cultura	Adubo	Custo Unitário
Batata	N	47
Milho	P-K	63
Trigo	N	47
Girassol	P	39
Beterraba	K	29

Chave Primária: “Cultura”.

Para normalizar a relação “ADUBAÇÃO” procede-se, como anteriormente, à sua divisão em novas relações, tal como se mostra no Quadro 3.6 e Quadro 3.7.

Quadro 3.6: Relação “ADUBO”.

Cultura	Adubo
Batata	N
Milho	P-K
Trigo	N
Girassol	P
Beterraba	K

Chave Primária: “Cultura”.

Quadro 3.7: Relação “CUSTO DE ADUBO”.

Adubo	Custo Unitário
N	47
P-K	63
N	47
P	39
K	29

Chave Primária: “Adubo”.