

1. Bases de Dados

1.1. Dados

Antes de avançarmos para as bases de dados, vamos começar por analisar os dados que nelas irão ser guardados. Começemos por observar os conceitos de dados apresentados por alguns autores.

Começemos por analisar a origem da palavra dado. “A palavra dado deriva do Latim do verbo dar, então dado é realmente um facto que se deu sobre o qual se podem deduzir factos adicionais” (DATE, C.J., 2004, Página 15).

De seguida podemos analisar o que são dados, “Dados – É qualquer informação que valha a pena preservar” (GARCIA-Molina, HECTOR ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; TRAD: SOUZA, Vandenberg D. De, 2001, Pagina 2). Ou se preferirem, “Dados – São factos conhecidos que podem ser registados e que têm um significado implícito” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 4).

Se pretendermos relacionar os dados com a informação teremos que “Dados - São os mais pequenos elementos da informação.” (FRIÉDÉRICH, Micheline; LANGLOIS, Georges, 2005, Página 63), ou se preferirem podemos diferenciá-los “Dados é aquilo que é realmente armazenado na base de dados. Informação é a mensagem, o significado, que se pode retirar dos dados.” (DATE, C.J., 2004, Página 6).

Mas será que já nos encontramos preparados para compreender o sentido dos dados em toda a sua amplitude? “Parafraseando Codd, Capturar o sentido dos dados é uma tarefa sem fim, podemos esperar continuar a ver desenvolvimentos nesta área, à medida que a nossa compreensão for evoluindo” (DATE, C.J., 2004, Página 410).

1.2. Base de dados

Vistos os dados isolados, vamos agora tentar compreender o que será uma base de dados. Vamos verificar o que afirmam diferentes autores sobre este conceito. Alguns autores mostram uma preocupação relacionada com o acesso e modificação dos dados. “Base de dados é uma colecção de dados organizados para facilitar o acesso e a modificação, preservada durante um longo período” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. De, 2001 Pagina 2) ou ainda, “Base de

dados é um conjunto de dados relacionados entre si, armazenados de uma forma conjunta em estruturas tipo ficheiro, com redundância controlada, que suportam acessos directos através de processos bem definidos ou através de programas desenvolvidos para o efeito” (COSTA PINTO, Manuel Luís Da, 1996. Pagina 7).

Uma base de dados deve ser sempre capaz de manter os dados ao longo do tempo. “Base de dados é uma colecção de dados persistentes que são usados pelo sistema de aplicações de uma determinada organização” (DATE, C.J., 2004, Página 11).

Qualquer base de dados apresenta um conjunto de propriedades. “A base de dados tem as seguintes propriedades, representa alguns aspectos do mundo real, é uma colecção lógica e coerente de dados com algum sentido intrínseco, é desenhada, construída e carregada com dados para um fim específico.”(ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 4).

As relações existentes entre os dados, são suficientemente importantes para serem consideradas em alguns conceitos. “Quando os dados organizacionais são inter-relacionados e armazenados juntos para servirem a uma ou mais aplicações, esta colecção de dados é geralmente chamada de base de dados”. (WETHERBE, James C., 1987, Página 235).

Não nos resta qualquer duvida que o facto de organizarmos os dados em bases de dados nos irá trazer enormes vantagens. “As vantagens de utilização de uma base de dados são, os dados podem ser partilhados, a redundância pode ser reduzida, a inconsistência pode ser evitada, o suporte transaccional pode ser fornecido, a integridade pode ser mantida, a segurança pode ser reforçada, os requisitos geradores de conflitos podem ser estabilizados, a standardização pode ser reforçada.” (DATE, C.J., 2004, Página 18).

Outra ideia fundamental, no conceito de base de dados, é a partilha da informação. “Um dos pressupostos fundamentais da utilização das bases de dados é que os dados de uma organização devem estar disponíveis, de forma partilhada, a todas as aplicações ou utilizadores que deles necessitem” (PEREIRA, José Luís, 1998, Página 291).

Mas não nos podemos esquecer que o objectivo de uma base de dados não se restringe apenas ao armazenamento de dados e processamento de transacções. “Aplicações de bases de dados podem ser classificadas, a grosso modo, em sistemas de processamento de transacções e sistemas de suporte à decisão” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 702).

1.3. SGBD – Sistemas de Gestão de Base de Dados

Começámos por analisar os dados isoladamente, seguiram-se as bases de dados, vamos agora ver o que será um sistema de gestão de base de dados. Alguns autores demonstram preocupações com o armazenamento e actualização da informação, “Um sistema de base de dados é basicamente somente um sistema computadorizado de manutenção de registros, por outras palavras é um sistema computadorizado cujo objectivo é armazenar informação permitindo aos utilizadores aceder e actualizar essa informação. (DATE, C.J., 2004, Página 6).

Para outros autores é importante acrescentarmos o conceito de sistema de gestão de bases de dados, preocupações com a segurança dos dados. “Um SGBD é uma ferramenta poderosa para criar e gerir grandes quantidades de dados de forma eficiente e permitir que esses dados persistam durante longos espaços de tempo com segurança.” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. De, 2001, Pagina 1).

Para outros autores a preocupação maior prende-se com o desenho da base de dados e a definição de procedimentos. “Sistemas de Gestão de Bases de dados, são programas que permitem a gestão do enorme volume de dados que pode existir numa organização e ser passível de tratamento, com o objectivo de gerar informação importante para o funcionamento da própria organização.

Essa gestão é uma gestão que envolve o desenho e definição global de toda a base de dados , como a criação de procedimentos para a sua manipulação, sem esquecer os importantíssimos procedimentos de segurança, quer de acesso quer da existência de cópias dos dados para prevenir danos à organização, resultantes da perda acidental dos mesmos” (COSTA PINTO, Manuel Luís Da, 1996, Pagina 7).

Outros autores consideram que um SGBD é um software. “O Software responsável pela implementação de uma base de dados é designado por Sistema de Gestão de Base de Dados” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 62). “O Sistema de Gestão de Base de Dados é o software que trata todos os acessos à base de dados” (DATE, C.J., 2004, Página 44).

Temos ainda alguns autores que misturam algumas das ideias anteriores. “Um Sistema de Gestão de Base de Dados é uma colecção de programas que permitem aos utilizadores criar e manter a base de dados. O SGBD é um software que facilita os processos de definição,

construção e manipulação da base de dados” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 5).

Chapter 1 / Databases and Database Users

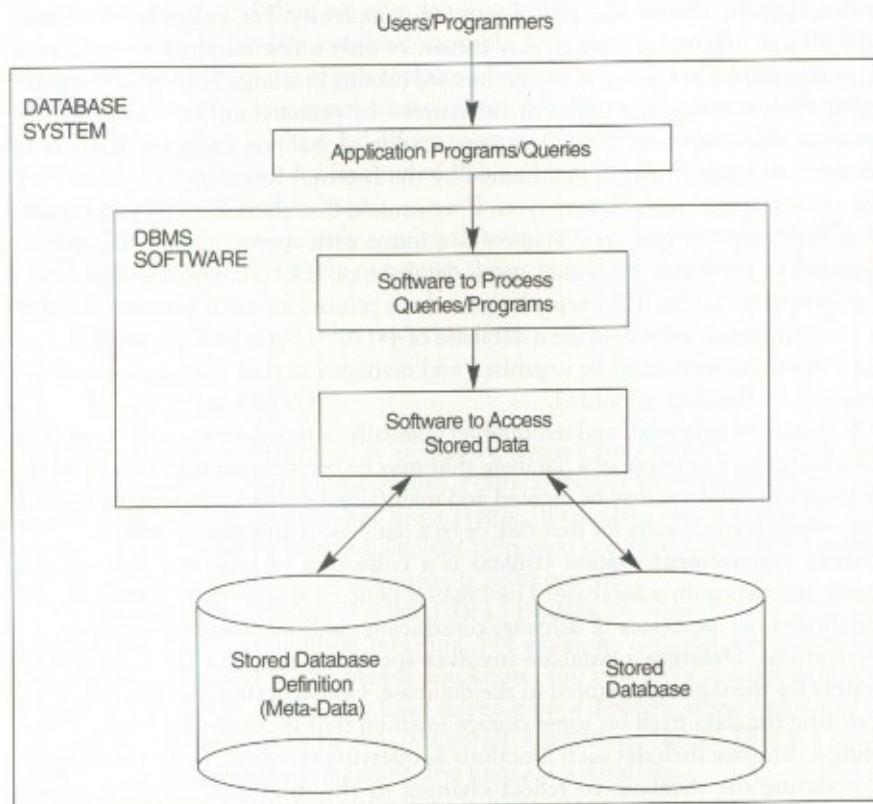


Figure 1.1 A simplified database system environment, illustrating the concepts and terminology discussed in Section 1.1.

Fig. Sistema de Gestão de Base de Dados

Fonte: ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 6

Alguns autores, referem o nível aplicacional, “O Sistema de Gestão de Base de Dados é um conjunto de software, destinado a gerir todo o armazenamento e manipulação dos dados do sistema, fazendo a interface entre o nível aplicacional e a base de dados propriamente dita” (PEREIRA, José Luís, 1998, Página 33).

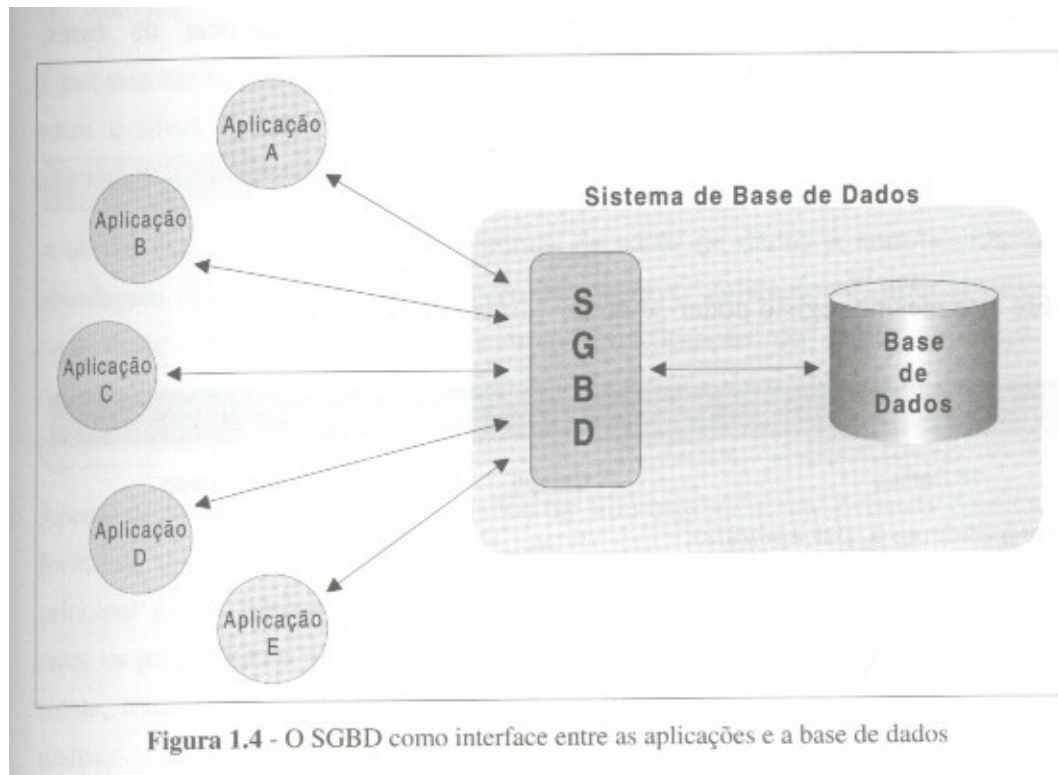


Fig. Sistema de Gestão de Base de dados
 Fonte: (PEREIRA, José Luís, 1998, Página 25)

1.3.1. Funções de um SGBD

Analiseemos de seguida quais são as funções que um SGBD terá que executar. “Um SGBD tem que ser capaz de aceitar definições de dados, manipular os dados, otimizar e executar as tarefas requeridas, garantir a segurança e integridade dos dados, gerir a concorrência e recuperação dos dados, providenciar um dicionário de dados que contém dados sobre os dados, manter a performance do sistema” (DATE, C.J., 2004, Página 45 a 47)

1.3.2. Componentes de um SGBD

Para conseguir executar as funções anteriormente descritas, o SGBD conta com quatro componentes. “Um sistema de gestão de base de dados envolve quatro componentes principais, os dados, o hardware, o software e os utilizadores. Os dados podem ser integrados ou partilhados. O hardware é composto por discos magnéticos utilizados para armazenar dados, memória, controladores e outros componentes. O software do SGBD é o mais importante, mas não é o único. Podem se encontrar outras ferramentas de desenvolvimento como, ferramentas para desenho da base de dados, para construção de relatórios ou de gestão

de transacções. Em relação aos utilizadores, há três tipos de utilizadores. Os programadores são responsáveis pela construção da base de dados. Os utilizadores finais, além de manterem a informação da base de dados actualizada, podem também construir consultas para extrair informação. Os Administradores da base de dados, devem pôr a funcionar um controle técnico que reforça as decisões políticas da administração dos dados. Têm obrigação de manter todo o sistema a funcionar”(DATE, C.J., 2004, Página 6 a 11).

1.3.3. Utilizadores de um SGBD

Considerámos importante este pequeno apontamento sobre os utilizadores de um SGBD, para chamar a atenção para a subdivisão existente dentro dos utilizadores finais. Existem utilizadores finais menos sofisticados e outros mais sofisticados. Os menos sofisticados estariam no nível transaccional, os mais sofisticados já se posicionariam no nível decisional. analisemos então os diferentes tipos de utilizadores.

“Utilizadores de sistemas de bases de dados:

- Administrador da base de dados - é o responsável máximo pelo bom funcionamento de todo o sistema. Tem como principais responsabilidades a especificação do esquema conceptual da base de dados e a sua manutenção. Sendo responsável pelo funcionamento do sistema, define procedimentos de backup e recuperação, monitoriza a sua utilização e afina o seu desempenho. Controla a segurança, dando e retirando acessos aos utilizadores.

- Utilizadores

- Utilizadores finais - Acedem ao sistema para consultar, alterar, adicionar ou remover dados, não poderão nunca alterar a estrutura da base de dados. Dentro deste grupo surgem os menos sofisticados, que se limitam a correr aplicações sobre o sistema, e os que, casualmente, nos casos em que o próprio SGBD dispõe dessa facilidade, utilizam linguagens de interrogação para obter, interactivamente, respostas específicas do sistema.

- Programadores - Utilizando linguagens de alto nível e tendo por base vistas sobre o esquema conceptual da base de dados, desenvolvem as aplicações que os utilizadores finais irão depois utilizar” (PEREIRA, José Luís, 1998, Página 92)

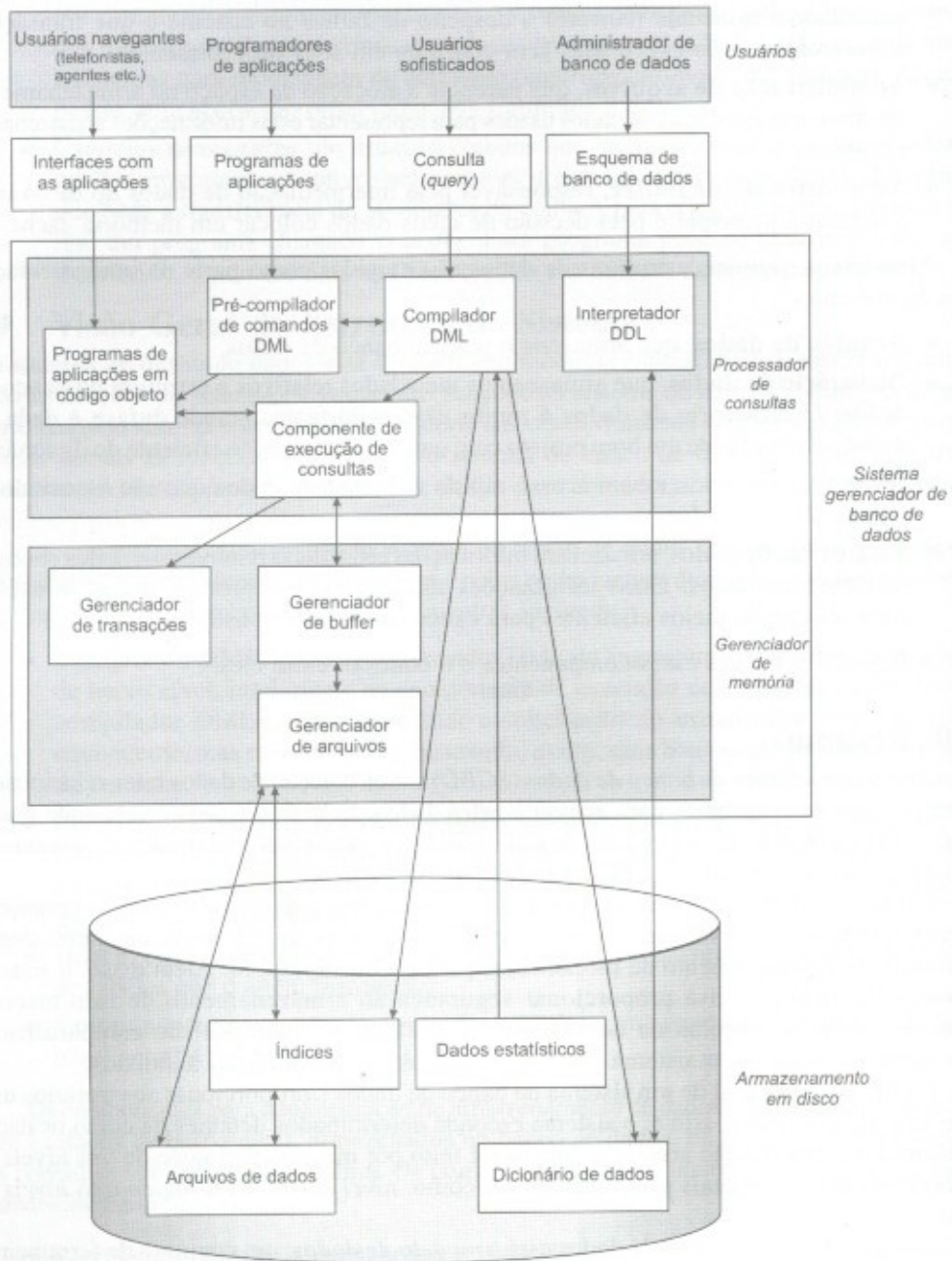


Figura 1.6 Estrutura do sistema.

Fig. Utilizadores do Sistema de Gestão de Base de Dados

Fonte: SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 18

1.3.4. Objectivos de um SGBD

Analisemos de seguida quais são os objectivos de um SGBD.

“Os Objectivos de um SGBD são os seguintes:

- Criar e manter a estrutura física dos dados em computador, ao nível do software, escondendo dos utilizadores as complexidades de gestão interna dos dados, possibilitando-lhes uma visão lógica da informação compatível com o modelo conceptual.
- Disponibilizar um interface que permita aos utilizadores e às aplicações aceder ao conteúdo da base de dados, para operações de consulta e de actualização.

Num SGBD podem identificar-se e distinguir-se os seguintes elementos:

- A base de dados propriamente dita, constituída por um conjunto de tabelas fisicamente registadas na base de dados, bem como pelas tabelas virtuais que resultam de operações efectuadas sobre essas tabelas.
- O motor da base de dados, constituída por um conjunto de programas responsáveis pela execução de todas as operações de criação, consulta e actualização da informação.
- O interface que disponibiliza o acesso dos utilizadores à definição da estrutura das bases de dados, bem como à consulta e actualização dos dados nela contidos.

A criação de um sistema de bases de dados, permite definir e implementar um conjunto de objectivos. Dentre os mais importantes, podem salientar-se os seguintes:

- A independência entre as aplicações e os dados

Num sistema de bases de dados, o software do SGBD proporciona um interface que permite um certo grau de isolamento entre as aplicações e as estruturas através dos quais os dados são fisicamente armazenados. As aplicações não actuam directamente sobre os dados. Os acessos são sempre realizados através do SGBD.

- Redução da redundância e da inconsistência

Temos uma redundância quando a informação referente a um mesmo facto é armazenada em dois pontos distintos de uma mesma base de dados. A inconsistência verifica-se quando temos, na mesma base de dados, informação diferente sobre o mesmo facto.

- O estabelecimento de regras de integridade

Visam garantir, se não a correcção, pelo menos a plausibilidade dos dados.

- O estabelecimento de mecanismos de segurança

Cada utilizador ou grupo de utilizadores é detentor de um conjunto de direitos ou privilégios, que lhe garantem determinadas formas de acesso aos objectos do sistema.” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 62 a 67)

1.3.5. Recursos de um SGBD

Para os utilizadores conseguirem atingir os seus objectivos, naturalmente necessitam de ter à sua disposição alguns recursos.

“Os recursos que um SGBD oferece ao utilizador são:

Armazenamento Persistente – Do mesmo modo que um sistema de arquivos, um SGBD admite o armazenamento de quantidades muito grandes de dados que existem independentemente de quaisquer processos que estejam a utilizar esses dados. Porém o SGBD vai muito além do sistema de arquivos ao proporcionar flexibilidade, como estruturas de dados que permitem o acesso eficiente a quantidades muito grandes de dados.

Interface de Programação – Um SGBD permite ao utilizador aceder e modificar dados através de uma linguagem de consulta poderosa. A vantagem de um SGBD sobre um sistema de arquivos é a flexibilidade para manipular dados armazenados de formas muito mais complexas que a leitura e gravação de arquivos.

Gestão de transacções – Um SGBD admite o acesso concorrente a dados, isto é o acesso simultâneo por muitos processos distintos, chamados transacções, ao mesmo tempo. Para evitar algumas das consequências indesejáveis do acesso simultâneo, o SGBD admite o isolamento, a aparência de que as transacções são executadas uma de cada vez, e a atomicidade, o registo de que as transacções sejam executadas completamente ou não sejam executadas de forma alguma. Um SGBD também admite a resiliência, a capacidade de se recuperar de muitos tipos de falhas ou erros.” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001 Pagina 1)

1.3.6. Propriedades das transacções de um SGBD

Para que um SGBD atinja os seus objectivos, e os seus utilizadores se encontrem satisfeitos, as transacções terão que possuir algumas propriedades.

“As transacções devem ter as seguintes propriedades:

- **Atomicidade** – A transacção deve ser executada numa base de tudo ou nada, ou seja ou é realizada na sua totalidade ou não é realizada de todo.
- **Isolamento** – Cada transacção deve aparentar ser executada como se nenhuma outra transacção estivesse em execução ao mesmo tempo.

- Durabilidade – O efeito de uma transacção sobre o banco de dados nunca deve ser perdido, depois que a transacção tiver sido concluída.
- Consistência – Todas as bases de dados têm restrições de consistência ou expectativas sobre relacionamentos entre elementos de dados. Espera-se que as transacções preservem a consistência da base de dados” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Pag. 10).

1.3.7. Vantagens da utilização de um SGBD

Como seria de esperar, a utilização de um SGBD traz-nos enormes vantagens.

“As vantagens da utilização de um SGBD são:

- Controle da redundância.
- Restrição a acessos não autorizados.
- Permitir um armazenamento persistente para os objectos dos programas e estruturas de dados.
- Permitir a inferência em acções seguindo regras.
- Permitir múltiplos interfaces para os utilizadores.
- Permite representar relacionamentos complexos entre os dados.
- Permite reforçar as restrições de integridade.
- Permite a recuperação de dados.

(ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 14)

1.4. Níveis de Abstracção de Dados

Existe uma relação obvia entre os diferentes tipos de utilizadores de um SGBD, e os diversos níveis de abstracção de dados.

“Podemos considerar três níveis de abstracção de dados

- Físico - É o nível mais baixo da descrição dos dados e corresponde à forma última de como se encontram organizados em memória, ao nível dos bits e bytes.
- Conceptual - É a este nível que se definem e descrevem exactamente os dados e as estruturas que a base de dados vai conter e as relações entre eles. Depois de estabelecer os dados que descrevem cada entidade fundamental, é necessário descrever as relações entre

elas. Os processos de estabelecimento de relações entre os dados dependem da organização da base de dados e do próprio sistema de gestão de base de dados.

- De vista - É o nível mais alto de abstracção. Proporciona, como o próprio nome indica, uma visão, ou uma vista da base de dados sob uma determinada perspectiva, aquela que mais interessa a um utilizador específico.” (COSTA PINTO, Manuel Luís Da, 1996, Página 11)

Outros autores preferem falar em níveis de arquitectura, mas igualmente os relacionam com os diferentes tipos de utilizadores. “Há três níveis de arquitectura:

- Nível Interno – É o nível mais baixo de representação de toda a base de dados, consiste em muitas ocorrências de muitos tipos de registos internos.

- Nível Conceptual – A vista conceptual é uma representação de toda a informação contida na base de dados.

- Nível externo – é o nível do utilizador individual. Este utilizador pode ser um programador ou um utilizador final de qualquer grau de sofisticação. Cada utilizador terá uma linguagem diferente à sua disposição” (DATE, C.J., 2004).

Na figura denominada níveis de independência de dados, estão representados os 3 níveis da arquitectura ANSI/SPARC.

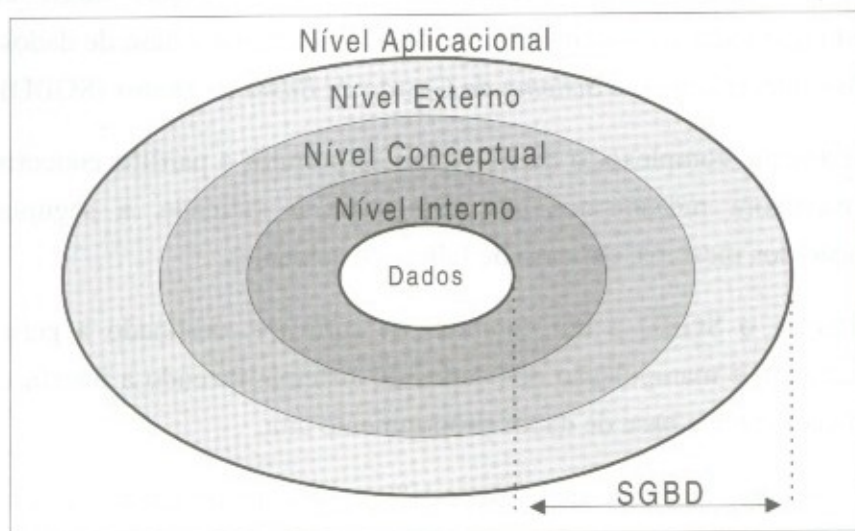


Figura 2.1 - Os três níveis da arquitectura ANSI/SPARC

Fig. Níveis de independência dos dados
Fonte: PEREIRA, José Luís, 1998, Página 34

Temos outros autores que preferem falar em níveis de independência de dados.

“Há três níveis de independência dos dados:

- O nível interno – Descreve a estrutura física da base de dados.
- O nível conceptual – Esconde os detalhes do armazenamento físico e concentra-se na descrição de entidades, tipos de dados, relacionamentos, operações e restrições dos utilizadores.
- O nível externo ou de vista – Descreve a parte da base de dados adstrita a um determinado grupo de utilizadores e esconde o resto da base de dados desse grupo” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 27)

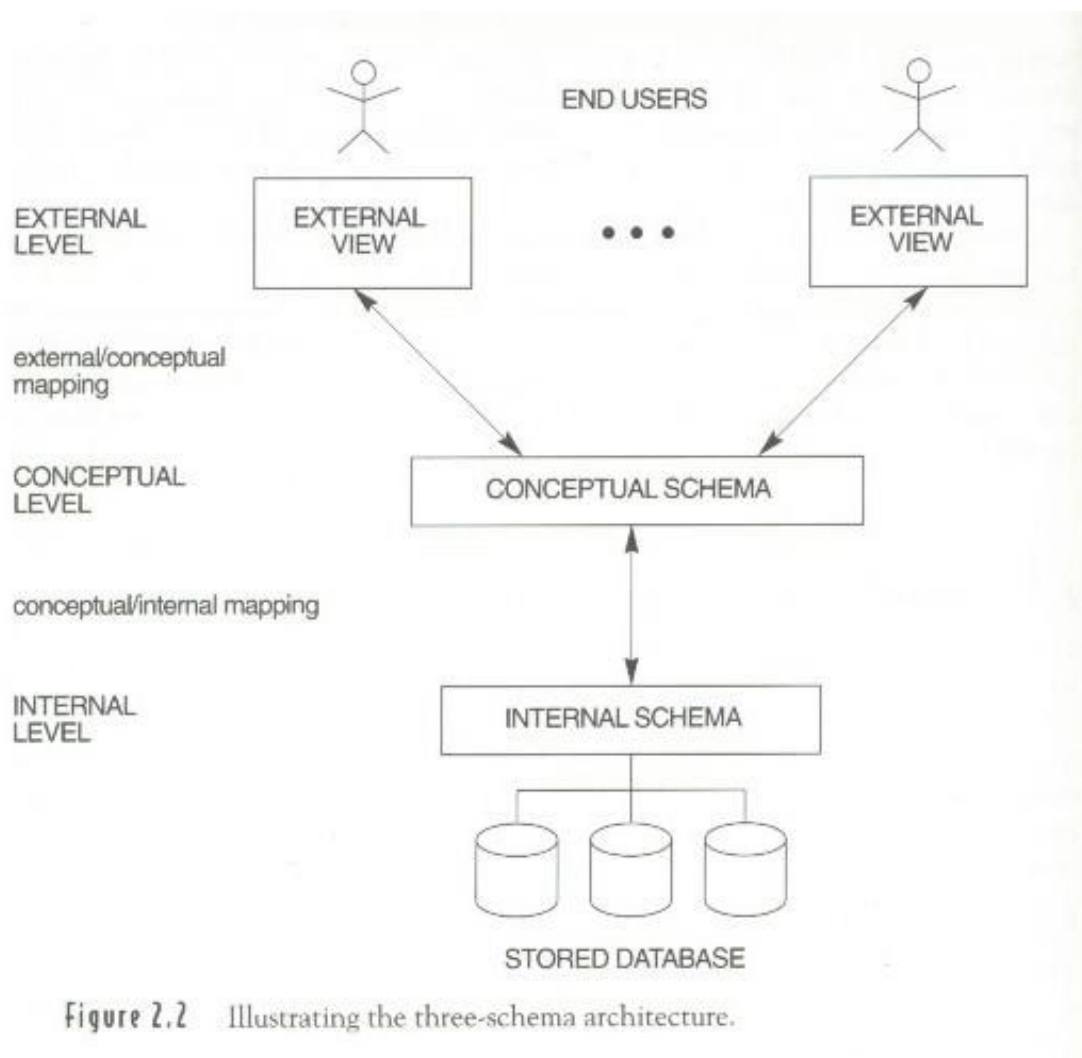


Fig. Níveis da arquitectura dos dados
 Fonte: ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 27

1.5. Modelo Relacional

1.5.1. Componentes do modelo relacional

“O Modelo relacional é sem margem para dúvidas, o fundador das tecnologias modernas das bases de dados. É a partir deste modelo que este campo de conhecimento se transformou numa ciência. O modelo relacional preocupa-se com três aspectos dos dados, a estrutura dos dados, a manipulação dos dados, a integridade dos dados.

É importante compreender que este modelo não é um modelo estático, tem evoluído ao longo dos anos e continua a desenvolver-se” (DATE, C.J., 2004, Página 109).

O modelo relacional é considerado de uma enorme importância, isto deve-se ao facto dele assentar noutras teorias já enraizadas. “Existem vários modelos conceptuais de bases de dados, mas o modelo actualmente dominante é o modelo relacional, baseado na teoria dos conjuntos. Numa síntese rápida, o modelo relacional pode ser descrito do seguinte modo:

- a informação é organizada em tabelas (também designadas por relações, na terminologia original do modelo).
 - Cada tabela é estruturada por forma a conter os dados referentes a entidades e relacionamentos que, na situação real, produzem a informação que a base de dados deve registar, actualizar e manter.
 - A estrutura da base de dados é vista pelo utilizador apenas como um conjunto de tabelas relacionadas, independentemente das técnicas de organização interna da informação, ao nível do suporte físico.
 - As operações de consulta e actualização da base de dados são logicamente equivalentes a operações envolvendo tabelas, em última análise, operações envolvendo conjuntos”.
- (CARRIÇO, José António ; CARRIÇO, António João, 1997, Pagina 62)

No modelo relacional como noutros campos da ciência é importante definir a utilização da terminologia. “Na terminologia formal do modelo relacional, uma linha é chamada de tupla, uma coluna é um atributo, e uma tabela é uma relação. Os valores que podem aparecer em cada coluna são chamados de dominio. O dominio é um conjunto de valores atómicos.

Valores atómicos porque cada valor do dominio é indivisível. Para cada dominio deve ser definido um formato e, ou um tipo de dados. O esquema da relação é constituído pelo nome da relação e uma lista de atributos. $R(A_1, A_2, A_3, \dots, A_n)$

Cada atributo pode tomar os valores de um determinado dominio. $Dom(A_i)$. O grau de uma relação, é o numero de atributos da relação. A relação é definida como uma série de tuplas sem qualquer ordem. Uma relação representa factos a um nivel lógico ou abstracto. Uma tupla pode ser considerada como uma série de valores de diferentes atributos, relativos ao

mesmo facto. Os valores de uma tupla terão que ser atómicos, ou seja não são aceites valores compostos, ou com diversos valores, num atributo. Por definição todos os elementos de uma série são distintos, portanto todas as tuplas de uma relação devem ser distintas. Isto significa que não podem existir duas tuplas com a mesma combinação de valores em todos os seus atributos. Algumas relações podem representar factos relativos a entidades, outras relações podem representar factos relativos a relacionamentos” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 196 a 201)

No modelo relacional a relação assume um papel central e fundamental. Uma relação deve conter certas propriedades. “cada linha contém um valor para cada coluna, não existe uma ordenação das colunas da esquerda para a direita, não há uma ordenação das linhas de cima para baixo, não há linhas duplicadas”(DATE, C.J., 2004, Página 148).

Há medida que o modelo relacional se for transformando numa base de dados, vamos verificar que normalmente as tabelas da base de dados surgem a partir das relações do modelo relacional. “Há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 61).

(DATE, C.J., 2004, Página 64)

O modelo relacional contém cinco componentes:

- Uma colecção aberta de tipos de escalas incluindo as escalas da álgebra de Boole¹
- Um gerador de relações (listas) com capacidade para as interpretar
- Capacidade para definir relações de variáveis.
- Uma operação relacional capaz de atribuir valores às relações de variáveis.
- Uma colecção aberta de operadores relacionais²

(DATE, C.J., 2004, Página 111 e 112)

O conceito de tipo é fundamental, cada valor, cada variável, cada parâmetro, cada operador e cada atributo relacional são de um determinado tipo. Um tipo representa um conjunto de valores que podem ser tomados. Qualquer tipo está associado a determinados operadores. O operador soma está associado ao tipo número, mas nunca ao tipo letra.

¹ Álgebra de Boole – Escalas tipicamente baseada em valores verdadeiros e falsos

² Operadores relacionais de álgebra relacional

1.5.2. Regras de Codd

(COSTA PINTO, Manuel Luís da, 1996, Pagina 20 e 21)

Em 1985, Ted Codd definiu e publicou as regras às quais uma base de dados deveria obedecer para ser considerada uma base de dados relacional. Estas foram o resultado de um trabalho de Codd sobre o modelo relacional e representam mais uma definição ideal do que uma definição real do que deve ser efectivamente uma base de dados relacional.

Estas regras dividem-se em três grandes grupos:

- Estruturais – modelo de como estão armazenados os dados.
- De independência – forma como os dados constituintes da base de dados são independentes das alterações efectuadas à base de dados.
- De linguagem – capacidades da linguagem que permitem ao utilizador final manipular os dados da base de dados.

Regra da informação – Toda a informação de uma base de dados está representada explicitamente ao nível lógico, mediante tabelas organizadas em linhas e colunas.

Regra do acesso garantido – Garante-se que todos e cada um dos dados, numa base de dados relacional, sejam acessíveis, recorrendo a uma combinação do nome da tabela. Do valor da chave primária e do nome da coluna.

Tratamento sistemático de valores nulos – Os valores nulos são suportados pelos SGBD'S completamente relacionais para representar a falta de informação, a informação inaplicável de um modo sistemático e independente do tipo de dados.

Catálogo permanente dinâmico baseado no modelo relacional – A própria base de dados é representada ao nível lógico por tabelas que descrevem a sua estrutura, da mesma forma que as tabelas criadas na base de dados contêm dados de entidades e acontecimentos que os descrevem. Desta forma os utilizadores autorizados podem operar com os dados da estruturação da base de dados, da mesma forma que o fazem com os outros dados criados na própria base de dados.

Regra da sublinguagem completa e compreensível dos dados – Um sistema relacional pode suportar várias linguagens e vários modos de utilização. Tem que, pelo menos, existir uma linguagem com instruções que se podem expressar de uma forma bem definida e que suporte a programação dos seguintes itens: definição de dados, definição de vistas, manipulação de dados, restrições de integridade, autorização, medidas de segurança em casa de falha de sistema.

Regra da actualização de vistas – O SGBD deve possuir forma de determinar, no momento de definição de uma vista, quando é que esta pode ser utilizada para inserir linhas, eliminar linhas, ou actualizar colunas das tabelas e guardar os resultados.

Inserção, actualização e supressão de alto nível – A capacidade de manejar uma base de dados relacional, ou uma relação derivada, não se aplica apenas à recuperação de dados, mas também à inserção, actualização e supressão de dados.

Independência física dos dados – Os programas de aplicação e as operações interactivas permanecem logicamente inalteradas quaisquer que sejam as trocas efectuadas nas representações de armazenamento e métodos de acesso.

Representação lógica dos dados – Os programas de aplicação e as operações interactivas permanecem logicamente inalteradas quaisquer que sejam as trocas efectuadas nas representações de armazenamento e métodos de acesso.

Representação lógica dos dados – Os programas de aplicação e as operações interactivas permanecem logicamente inalteradas pelas eventuais alterações na organização dos dados que não envolvam perda de informação.

Independência dos dados – As restrições de integridade específicas para uma base de dados particular devem ser definidas na sublinguagem de dados relacional.

Independência de distribuição – Os programas de aplicação e as operações interactivas não precisam de ser modificados, quer um sistema suporte ou não distribuição de base de dados entre vários computadores.

Regra da não subversão – Se um sistema de gestão de base de dados relacional tem uma linguagem de baixo nível procedimental, essa linguagem não deve subverter, ou não cumprir, as características de integridade e segurança expressas na linguagem relacional de alto nível.

1.5.3. Notação do modelo relacional

“Em qualquer modelo é sempre necessário definir a notação a utilizar.

R – Relação $R(A_1, A_2, \dots, A_n)$

A – Atributo

t – Tupla

v – valores de uma determinada tupla

A_i – Atributo i da relação $R(A_1, A_2, \dots, A_i, \dots, A_n)$

v_i – valor i da tupla $t = \langle v_1, v_2, \dots, v_i, \dots, v_n \rangle$

$t[A_i]$ – valor v_i em t para o atributo A_i

$t[A_u, A_w, \dots, A_z]$ onde A_u, A_w, \dots, A_z são uma lista de alguns atributos da relação R , refere-se a uma subtupla de valores $\langle v_u, v_w, \dots, v_z \rangle$ da tupla t , correspondentes aos atributos especificados na lista.

Q, R, S são nomes de relações

q, r, s são estados de relações

t, u, v são tuplas”

(ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página201)

1.5.4. Vantagens do modelo relacional

“De uma forma sintética podemos considerar o seguinte conjunto de vantagens na utilização do modelo de dados relacional:

- base teórica sólida baseada na teoria matemática das relações
- estruturas de dados simples
- pequeno numero de conceitos
- operadores simples
- suporte de linguagem SQL
- mecanismo de vistas
- independência lógica e física dos dados
- facilidade no desenvolvimento de aplicações
- Definição de dados dinâmica
- facilidade de instalação e operação
- simplificação do desenho da base de dados
- suporte de base de dados distribuídas
- performance
- capacidade de crescimento”

(COSTA PINTO, Manuel Luís da, 1996, Pagina 22)

O modelo relacional ao apresentar as enorme vantagens atrás descritas, pode ser considerado aquele que melhor se adapta ao mundo das empresas. “Relativamente aos modelos convencionais, particularmente no que diz respeito ao modelo relacional, dadas as suas qualidades de simplicidade e estabilidade, estes continuarão a ser largamente utilizados na área do processamento de dados de gestão onde, tradicionalmente, se encontram bem adaptados e com grande sucesso” (PEREIRA, José Luís, 1998, Página 268).

1.6. Modelo Entidade - Relacionamento

1.6.1. Modelo ER

Comecemos por ver o que é o modelo entidade-relacionamento, “O modelo de dados entidade-relacionamento (E-R) tem por base a percepção do mundo real como um conjunto de objectos básicos, chamados entidades, e do relacionamento entre eles” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 7).

Ou segundo outro autor, “O modelo Entidade Relacionamento corresponde a uma colecção de objectos designados por entidades e um conjunto de relacionamentos entre esses objectos. A entidade pode ser descrita por um conjunto de dados ou atributos. Ao conjunto de valores permitidos de um dado atributo chamamos domínio do atributo. Podemos então considerar que cada entidade é descrita, não por um conjunto de atributos, mas por um conjunto de pares (atributo, valor). Um conjunto de entidades é uma série de entidades do mesmo tipo, descritas exactamente pelos mesmos atributos, mas com valores diferentes. Uma base de dados, à luz deste modelo, será portanto uma colecção de conjuntos de entidades, cada uma das quais contém um numero qualquer de entidades do mesmo tipo” (COSTA PINTO, Manuel Luís da, 1996, Pagina 12).

Ambos os autores falam em objectos denominados entidades e no relacionamento entre esses objectos. Comecemos por analisar o que são as entidades.

1.6.2. Entidade

Vamos começar por observar diferentes conceitos de entidade apresentados por vários autores.

“Uma entidade é uma coisa ou um objecto do mundo real que pode ser identificado por outros objectos” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 7).

“Uma entidade organizacional é algo com interesse duradouro para a organização e sobre a qual pode ser armazenada informação” (AMARAL, Luís Alfredo Martins do ; VARAJÃO, João Eduardo Quintela, 2000, Página 102).

“Entidade é um objecto distinto representado na base de dados sobre o qual é necessário guardar informação” (DATE, C.J., 2004, Página 12 e 13).

Uma entidade, como qualquer objecto, terá diversas características ou propriedades. A essas características ou propriedades, no modelo entidade-relacionamento, dá-se o nome de atributo. “A entidade representa um objecto ou conceito do mundo real. O atributo representa uma propriedade ou característica que descreve a entidade” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 25)

Tanto as entidades como os atributos podem ser de diferentes tipos. “Uma entidade pode ser um objecto:

- Com uma existência física, uma pessoa, um carro, uma casa.
- Com uma existência conceptual, uma empresa, um emprego, um curso.

Cada entidade tem vários atributos, são os atributos que descrevem as propriedades ou características das entidades. A entidade pessoa pode ser descrita pelos atributos nome, idade, endereço e emprego. Os atributos são preenchidos por valores, estes valores serão a maior parte dos dados armazenados na base de dados.

Os atributos podem ser:

- Compostos – quando podem ser subdivididos – ex: morada.
- Simples ou atômicos – quando não podem ser subdivididos – ex: país, cidade, rua.

Os atributos podem ter:

- Um valor único – ex: nome do cão – Rex.
- Vários valores – ex: cor do cão – castanho e preto.

Os atributos podem ser:

- Armazenados – São aqueles que estão armazenados na base de dados.
- Derivados ou Calculados – São aqueles que são calculados com base nos outros atributos – ex: com base na data de nascimento e a data de hoje podemos calcular a idade da pessoa” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 45 a 48).

Analisemos a visão de outro autor sobre os atributos. “As entidades e os relacionamentos têm propriedades ou atributos. As propriedades ou atributos podem ser simples ou compostas, um atributo pode ser composto por vários atributos, por exemplo o nome completo pode ser composto por nome e apelido. podem ser chaves quando são atributos únicos. Podem ser únicos ou com múltiplos valores. podem ser sem valores. E ainda, com valores base ou derivados dos valores base” (DATE, C.J., 2004, Página 414 a 417).

É fundamental ficar bem claro que existem diferenças entre as entidades, os dados e os acontecimentos. “Entidade pode ser uma pessoa, lugar, objecto ou conceito acerca do qual uma organização necessita registar dados. Dados são elementos que descrevem as entidades e

acontecimentos. Acontecimento é uma ocorrência relativa a uma entidade da qual é necessário também registrar dados” (COSTA PINTO, Manuel Luís da, 1996, Página 5).

Por sua vez devemos distinguir as entidades fortes das entidades fracas. “Entidades fracas são entidades que não possuem no conjunto dos seus atributos um ou mais do que um, capazes de por si só constituírem uma chave primária. Entidades fortes são entidades que contêm uma chave primária entre os seus atributos” (COSTA PINTO, Manuel Luís da, 1996, Pagina 14). Ou se preferirem “Entidade fraca é uma entidade que é dependente de outra entidade, ou seja, uma entidade fraca não pode existir, se a outra entidade forte não existir. Entidade forte ou regular é uma entidade que não é fraca” (DATE, C.J., 2004, Página 414 a 417).

1.6.3. Relacionamento

Vistas as entidades e os seus atributos, vamos nos debruçar agora sobre os relacionamentos³.

Vamos poder observar que o conceito de relacionamento não difere muito de autor para autor. “Um relacionamento é uma associação entre entidades” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 7). “O relacionamento entre duas ou mais entidades representa uma interacção entre entidades” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 25). “Relacionamentos são o elo de ligação entre as entidades” (DATE, C.J., 2004, Página 12 e 13).

Visto o conceito de relacionamento, podemos de seguida analisar os diferentes tipos de relacionamentos. “Relacionamentos são associações entre entidades. Entidades participantes são entidades envolvidas num determinado relacionamento. O Grau identifica o numero de participantes envolvidos num relacionamento. O relacionamento pode ser total ou parcial. No total, todos os registos de uma entidade têm um registo relacionado na outra entidade. No parcial, nem todos os registos de uma entidade têm um registo relacionado na outra entidade. Os relacionamentos entre entidades podem ser, de um para um, de um para muitos, ou de muitos para muitos.

³ Em Portugal a nomenclatura utilizada, por vezes provoca alguma confusão. Alguns autores utilizam a palavra relação para defenir os relacionamento entre várias entidades. Mas também utilizam a palavra relação para referir uma entidade. A utilização da palavra relação para descrever duas situações distintas, pode provocar alguma confusão. Os anglo-saxónicos utilizam a palavra relation quando se referem à relação, entidade ou tabela. E utilizam a palavra relationship quando se referem ao relacionamento entre duas relações, entidades os tabelas. Os brasileiros utilizam a palavra relação, para definir a relação, entidade ou tabela. E utilizam a palavra relacionamento, quando se referem ao relacionamento entre duas relações, entidades os tabelas. Uma vez que a nomenclatura portuguesa pode gerar alguma confusão, vamos seguir a nomenclatura brasileira.

Tipos, subtipos e supertipos – Qualquer entidade é de um determinado tipo, mas também pode ser de vários tipos simultaneamente.

- Subtipo – Quando todos os atributos de uma entidade se aplicam, a uma outra entidade, podemos dizer que a segunda entidade é um subtipo da primeira entidade.

Supertipo - Quando todos os atributos de uma entidade se aplicam a uma outra entidade, podemos dizer que a primeira entidade é um supertipo da segunda entidade.

Os atributos e relacionamentos aplicadas à entidade supertipo, são herdadas pela entidade subtipo” (DATE, C.J., 2004, Página 414 a 417)

• 1 – 1 (relação de uma para uma)

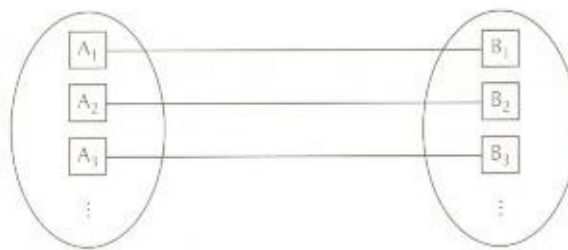


Fig. Relação uma para uma

Fonte: COSTA PINTO, Manuel Luís da, 1996, Página 13

Um elemento do conjunto de entidades de A está associado a um e um só elemento do conjunto de entidades de B e vice-versa.

• 1 – n (relação de uma para muitas)

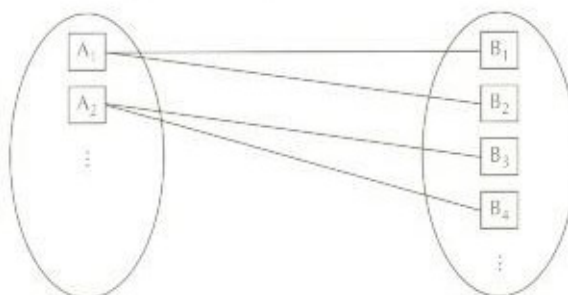


Fig. Relação uma para muitas

Um elemento de A está associado a um qualquer numero de entidades de B. Um elemento de B está somente associado com uma entidade de A.

• $n - 1$ (relação de muitas para uma)

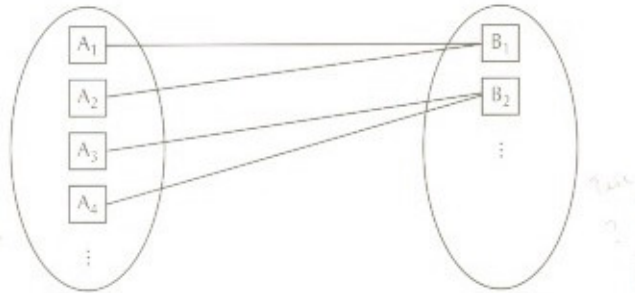


Fig. Relação muitas para uma

Fonte: COSTA PINTO, Manuel Luís da, 1996, Página 13

Um elemento de A está associado a uma e uma só entidade de B, enquanto que um elemento de B pode estar associado a qualquer numero de entidades de A.

• $n - n$ (relação de muitas para muitas)

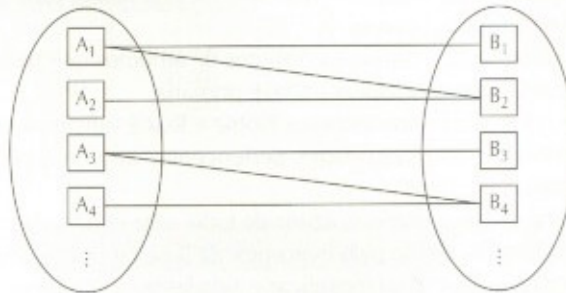


Fig. Relação de muitas para muitas

Fonte: COSTA PINTO, Manuel Luís da, 1996, Página 13

Um elemento de A está associado a um qualquer numero de entidades de B, e um elemento de B pode estar associado a um qualquer numero de entidades de A.

Um relacionamento não é necessariamente entre apenas duas entidades. O numero de entidades envolvida num relacionamento define o grau. “O grau de um relacionamento é medido pelo numero de entidades que participam nesse relacionamento. Pode ser binário

quando participam duas entidades, ou ternário quando participam três entidades” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 53 a 55).

1.6.4. Diagrama Entidade Relacionamento (ER)

O diagrama ER serve para representar graficamente a estrutura lógica da base de dados.

“Toda a estrutura lógica da base de dados pode ser expressa graficamente por meio do diagrama E-R, cujos construtores dos seguintes componentes são:

- Rectângulos que representam os conjuntos de entidades.
- Elipses que representam os atributos.
- Losangos que representam os relacionamentos entre os conjuntos de entidades.
- Linhas que unem os atributos aos conjuntos de entidades e o conjunto de entidades aos seus relacionamentos.
- Elipses duplas que representam atributos multivalorados
- Linhas duplas que indicam participação total de uma entidade em um conjunto de relacionamentos” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 8 e 34).

Podemos observar a mesma notação segundo outro autor. “O diagrama entidade relacionamento consiste numa técnica de representação da estrutura lógica da base de dados na forma de uma figura. Seguindo a máxima que uma figura vale por mil palavras.

As entidades são representadas por rectângulos contendo o nome da entidade..Uma entidade fraca deve ser representada com um rectângulo duplo. Os atributos são ilustrados por elipses contendo o nome da respectiva propriedade e estão ligadas à entidade por uma linha sólida. Cada relacionamento é representado por um losango contendo o nome do relacionamento. Utiliza-se um losango duplo se o relacionamento em questão é entre uma entidade fraca e a entidade da qual a sua existência depende. Os participantes num relacionamento estão ligados ao relacionamento por linhas sólidas. Cada linha terá uma etiqueta indicado 1 ou M consoante o relacionamento em causa.

Os relacionamentos podem ser de um para um, um para muitos ou muitos para muitos” (DATE, C.J., 2004, Página 418 e 419).

“Uma entidade tipo é representada no diagrama ER com uma caixa rectangular com o nome da entidade. Para uma entidade escolhe-se sempre um nome singular e nunca um plural porque o nome da entidade aplica-se a cada registo individual dessa entidade.

Entidade fraca é uma entidade que não tem nenhum atributo chave. No diagrama ER a entidade fraca é representada por uma caixa com uma linha dupla. Entidade forte é uma entidade que tem um atributo chave.

Os atributos são representados numa elipse oval com o seu nome, são ligados à respectiva entidade por linhas rectas. Os atributos com vários valores são representados com duas elipses ovais. Os valores do atributo chave podem ser utilizados para identificar cada entidade de forma unívoca. Por vezes é necessário juntar vários atributos para formar a chave. Uma entidade pode não ter chave, às entidades que não têm chave, chamamos entidades fracas. Os atributos chave são representados no diagrama ER sublinhando o nome do atributo dentro da elipse oval.

Sempre que um atributo de uma entidade se refere a outra entidade, algum relacionamento existe. No diagrama ER os relacionamentos são representados por um losango, losango esse que estará ligado por linhas rectas às entidades que pretende relacionar. O nome do relacionamento é colocado no centro do losango. Para o nome de um relacionamento, normalmente utiliza-se um verbo.

Um relacionamento entre uma entidade forte ou dominante e uma entidade fraca ou subordinada é chamado relacionamento identificador. No diagrama ER o relacionamento identificador é representado por um losango com linha dupla. (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 49 a 53 e 59 e 60 e 62)

3.7 ER Diagrams, Naming Conventions, and Design Issues


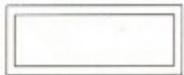



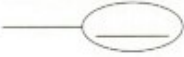


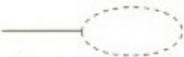
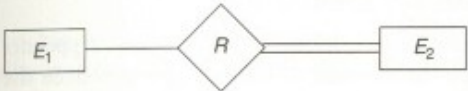

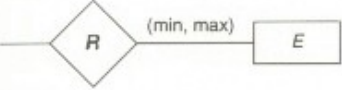
Symbol	Meaning
	ENTITY
	WEAK ENTITY
	RELATIONSHIP
	IDENTIFYING RELATIONSHIP
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E_2 IN R
	CARDINALITY RATIO 1 : N FOR $E_1 : E_2$ IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Figure 3.14 Summary of ER diagram notation.

Fig. Convenções do diagrama ER

Fonte: ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 63

“Uma entidade é uma coisa ou um objecto no mundo real que pode ser identificada de forma unívoca em relação a todos os outros objectos. Uma entidade tem um conjunto de propriedades, e os valores para alguns conjuntos dessas propriedades devem ser únicos. Uma entidade pode ser concreta ou pode ser abstracta.

Um relacionamento é uma associação entre uma ou várias entidades.

Mapeamento das cardinalidades:

- Um para Um - Uma entidade A está associada no máximo a uma entidade em B, e uma entidade em B está associada a no máximo uma entidade em A.
- Um para Muitos - Uma entidade em A está associada a várias entidades em B. Uma entidade em B, entretanto deve estar associada no máximo a uma entidade em A.
- Muitos para Um - Uma entidade em A está associada a no máximo uma entidade em B. Uma entidade em B, entretanto, pode estar associada a um numero qualquer de entidades em A.
- Muitos para muitos - Uma entidade em A está associada a qualquer numero de entidades em B e uma entidade em B está associada a um numero qualquer de entidades em A” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 21 e 24 e 29).

“Rácio de Cardinalidade - O rácio de cardinalidade dos relacionamentos especifica o numero de relacionamentos em que uma entidade pode participar. Os rácios de cardinalidade possíveis num relacionamento binário são:

- 1 para 1
- 1 para N
- N para 1
- M para N

O rácio de cardinalidade dos relacionamentos binários são representados no diagrama ER, colocando um 1, M ou N no losango que representa o relacionamento.

Relacionamentos recursivos – São relacionamentos em que a mesma entidade participa mais do que uma vez, a desempenhar papéis diferentes.

A dependência da participação das entidades no relacionamento pode ser:

- Total – Todos os valores da entidade A, têm obrigatoriamente um valor relacionado na entidade B.
- Parcial – Nem todos os valores da entidade A, têm um valor relacionado na entidade B.

No diagrama ER os relacionamentos parciais são representados por uma linha simples e os relacionamentos totais por uma linha dupla.

Os relacionamentos também podem ter atributos similares aos das entidades. Os atributos dos relacionamentos de 1 para 1 e 1 para N podem migrar para uma das entidades envolvidas no relacionamento.

- Nos relacionamentos de 1 para 1 podem migrar para uma das entidades envolvidas no relacionamento.

- Nos relacionamentos de 1 para N o atributo só pode migrar para a entidade que se encontra no lado N do relacionamento.

- Nos relacionamentos de N para M, alguns atributos podem ser determinantes na determinação da combinação das entidades envolvidas no relacionamento. Estes atributos devem ser especificados no diagrama, como atributos do relacionamento” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 53 a 58)

1.7. Normalização

“A Normalização é um processo que consiste em estruturar a informação em tabelas da forma mais adequada tendo em consideração as operações que podem vir a ser executadas sobre essa informação. O processo de normalização permite evitar redundâncias desnecessárias e eliminar problemas que podem ocorrer em operações de inserção, eliminação e actualização de dados” (CARRIÇO, José António; CARRIÇO, António João, 2004, Página 49).

“Normalização consiste em definir o formato lógico adequado para as estruturas de dados das tabelas de uma base de dados relacional, identificadas no projecto lógico do sistemas, com o objectivo de minimizar o espaço utilizado pelos dados e garantir a integridade e confiabilidade das informações” (MACHADO, Felipe Nery Rodrigues, 2004, Página 182).

“O processo de normalização leva ao refinamento das entidades, retirando delas grande parte das redundâncias e inconsistências. Naturalmente, para que haja uma associação entre entidades, é preciso que ocorram redundâncias mínimas de atributos que evidenciam esses relacionamentos. Sem essas redundâncias não haveria relacionamentos entre entidades” (MACHADO, Felipe Nery Rodrigues, 2004, Página 206).

Relativamente à normalização podemos afirmar que se “trata de um processo sistemático, conduzido por regras que traduzem os relacionamentos existentes entre os dados, constituído por decomposições sucessivas de relações maiores em relações cada vez menores que, ao longo de vários passos, tenta identificar e remover a redundância própria das associações dos dados tal e qual existem no mundo real. Por outro lado, organizando esses dados, a

normalização dá origem a modelos relacionais mais flexíveis e menos sujeitos a posteriores problemas na sua manipulação” (PEREIRA, José Luís, 1998, Página 179)

“Na prática, a normalização não deve ser levada às últimas consequências ao nível do desempenho global do sistema. Como é evidente, quanto mais espalhados estiverem os dados por várias relações, mais junções entre relações vão ser necessárias para obter a mesma informação. Ou seja, a obtenção de informação pode, em algumas situações, tornar-se um processo demorado, dado o número de junções que é necessário efectuar entre relações. Deparam-se, então dois objectivos frequentemente conflituosos: por um lado pretendem-se sistemas flexíveis, sem problemas de redundância, por outro lado exigem-se sistemas com alto desempenho. É pois necessário estabelecer um compromisso. Pretende-se um esquema equilibrado que nunca ponha em risco a integridade da base de dados, mas que, simultaneamente, tenha um desempenho razoável, pois só assim será utilizado” (PEREIRA, José Luís, 1998, Página 192).

1.7.1. Dependências

(CARRIÇO, José António; CARRIÇO, António João, 2004, Pagina 51)

- Dependência Funcional - O conceito de dependência funcional será apresentado tendo como referência a tabela A com a seguinte estrutura: $A(x,y,z,w)$

Diz-se que o atributo z é funcionalmente dependente do atributo x , se, para um dado valor de x o valor de z é sempre o mesmo. Ou seja, conhecido o valor de x sabe-se automaticamente qual é o valor que está na coluna relativa ao atributo z . O atributo x designa-se por determinante e o atributo z designa-se por dependente.

- Dependência Funcional Elementar - Existe uma dependência funcional elementar entre o conjunto de atributos $\{x,y\}$ e o atributo w , quando o atributo w depende do conjunto $\{x,y\}$ como um todo, e não depende de nenhum dos seus elementos (ou subconjuntos) tomados isoladamente. No caso de existir dependência funcional elementar entre $\{x,y\}$ e z , pode igualmente dizer-se que o atributo w é irredutivelmente dependente de $\{x,y\}$.

(PEREIRA, José Luís, 1998, Pág. 181)

Dependências funcionais, multivalor e de junção.

- Dependências funcionais - Por definição diz-se que existe uma dependência funcional $X \rightarrow Y$ entre dois conjuntos de atributos X e Y , se uma instância de valores dos atributos de X

determina ou identifica univocamente uma instância de valores dos atributos de Y. Ou seja, não existem duas instâncias distintas de Y, para uma mesma instância de X.

Se $X \rightarrow Y$, então existe uma dependência funcional entre X e Y em que X determina Y ou, o que é o mesmo, Y depende de X.

Se $X \rightarrow Y$, então a correspondência entre X e Y é do tipo 1:1 (um para um) ou M:1 (muitos para um).

A manipulação de dependências funcionais obedece aos chamados axiomas de Armstrong.

- Reflexividade - Se $X \rightarrow Y$, então $X \rightarrow X$

- Aumentatividade - Se $X \rightarrow Y$, então $XZ \rightarrow YZ$

- Transitividade - Se $X \rightarrow Y$ e $Y \rightarrow Z$, então $X \rightarrow Z$

Estes por sua vez dão origem às seguintes regras de inferência

- Decomposição - Se $X \rightarrow YZ$ então $X \rightarrow Y$ e $X \rightarrow Z$

- União - Se $X \rightarrow Y$ e $X \rightarrow Z$, então $X \rightarrow YZ$

- Pseudotransitividade - Se $X \rightarrow Y$ e $YW \rightarrow Z$, então $XW \rightarrow Z$

- Dependência multivalor - Por definição dada uma qualquer relação $R(X,Y,Z)$ diz-se que existe uma dependência multivalor $X \twoheadrightarrow Y$ (X multidetermina Y) se, para cada par de tuplos de R contendo os mesmos valores de X, também existe em R um par de tuplos correspondentes à troca dos valores de Y no par original.

- Dependências de junção - Numa relação existe uma dependência de junção se, dadas algumas projecções sobre essa relação, apenas se reconstrói a relação inicial através de algumas junções bem específicas, mas não de todas.

Formas Normais

(DATE, C.J., 2004, Pág. 351 a 364)

Uma relação de variáveis só pode estar na segunda forma normal se já satisfizer todas as condições da primeira forma normal.

- Primeira forma Normal - Uma relação de variáveis está na primeira forma normal se cada tupla contiver um valor para cada atributo.

- Segunda forma normal - Uma relação de variáveis encontra-se na segunda forma normal se estiver na primeira forma normal e todos os atributos que não são chave primária, dependem irredutivelmente da chave primária.

- Terceira forma normal – Uma relação encontra-se na terceira forma normal se já estiver na segunda forma normal e todos os seus atributos que não forem chave primária forem não transitivamente dependentes da chave primária.

(PEREIRA, José Luís, 1998, Página 185 a 188)

- 1ª Forma Normal (1FN) - Este primeiro passo do processo de normalização visa eliminar os grupos de valores repetidos que, eventualmente possam existir em estruturas não normalizadas. Em múltiplas situações surgem casos em que para os mesmos valores de vários atributos existem diferentes valores de outros atributos. A 1FN resolve este problema decompondo a estrutura inicial em tantas relações quantos os grupos de atributos que se repetem e identificando as chaves de cada uma dessas relações, eliminando-se a redundância existente na versão não normalizada.

- 2ª Forma Normal (2FN) - Uma relação na 2FN é uma relação em que, além de estar na 1FN, todos os atributos não pertencentes a qualquer chave candidata (atributos não primos) devem depender da totalidade da chave e não apenas de parte dela. Se os atributos presentes na relação estão envolvidos em duas ou mais dependências funcionais a relação ainda não se encontra na 2FN. A solução passa por decompor a relação, de acordo com as dependências funcionais, em várias relações.

- 3ª Forma Normal (3FN) - Uma relação na 3FN é uma relação em que, além de estar na 2FN, não existem dependências funcionais entre os atributos não chave. Ou seja cada atributo deve depender apenas da chave primária da relação. A solução é decompor a relação em duas relações de acordo com as dependências funcionais existentes.

- Boyce-Codd Normal Form (BCFN) - Por definição uma relação está na BCNF se todos os atributos são funcionalmente dependentes da chave, de toda a chave e nada mais do que a chave.

- 4ª Forma Normal - Diz-se que uma relação está na 4FN se está na BCNF e não existem dependências multivalor. Se numa relação ainda existem dependências multivalor, esta relação deve ser decomposta.

- 5ª Forma Normal - Uma relação estará na 5FN se não puder ser mais decomposta sem perda de informação. Se puder ser reconstruída sem perda de informação a partir de algumas das suas projecções, então existe uma dependência de junção e, portanto, a relação não se encontra na 5FN, devendo ser decomposta segundo essa dependência.

(CARRIÇO, José António; CARRIÇO, António João, 2004, Página 50)

1FN - Uma tabela encontra-se na primeira forma normal se todos os seus atributos ou colunas estiverem definidos em domínios que contenham apenas valores atômicos. Isto significa que um atributo só pode admitir valores elementares e não conjuntos de valores.

2FN - A condição para que a tabela se encontre na 2FN é a de que todos os atributos que não pertençam à chave dependam da chave através de uma dependência funcional elementar. Há uma dependência funcional elementar quando um determinado atributo depende dum conjunto de atributos como um todo, e não depende de nenhum dos seus elementos (ou subconjuntos) tomados isoladamente.

3FN - A condição para que uma tabela se encontre na terceira forma normal é que nenhum dos atributos que não fazem parte da chave pode ser funcionalmente dependente de qualquer combinação dos restantes. Cada atributo depende apenas da chave e não depende de qualquer outro atributo ou conjunto de atributos. Esta condição visa eliminar as chamadas dependências transitivas. Para que uma tabela esteja na 3FN é necessário que todos os atributos que não pertençam à chave sejam mutuamente independentes. Na prática, os atributos são mutuamente independentes se puderem ser actualizados independentemente uns dos outros.

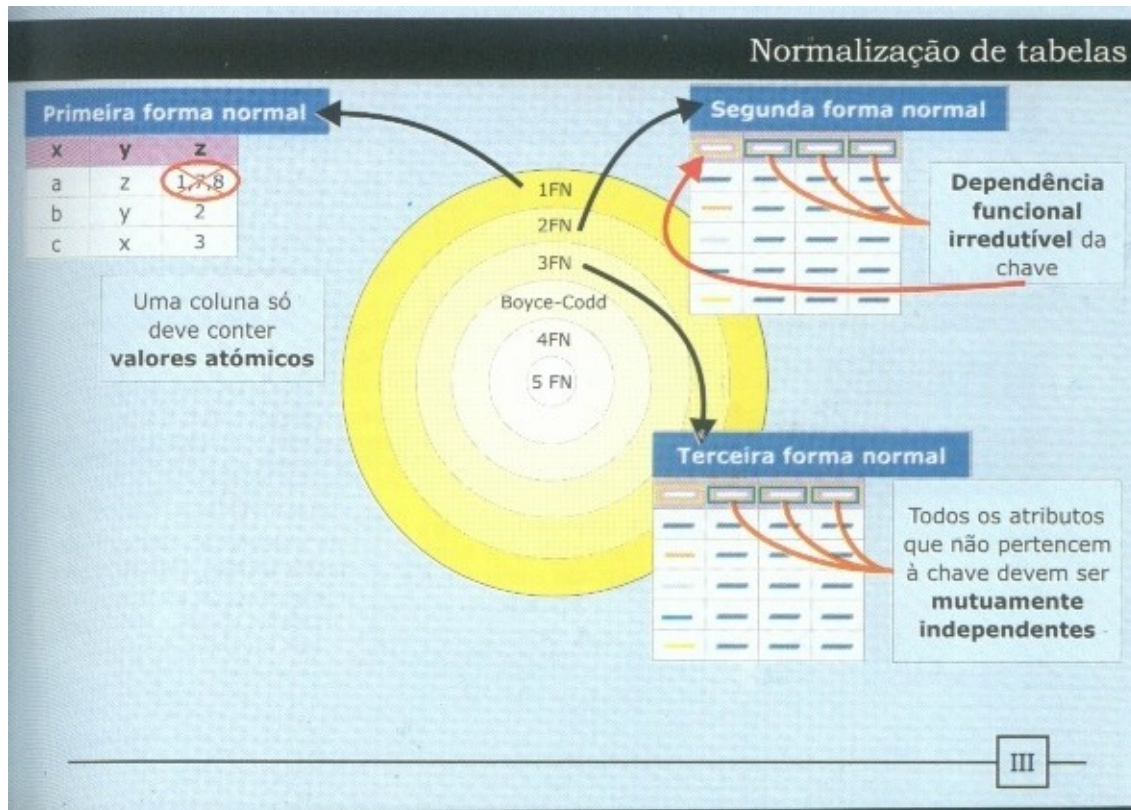


Fig. Formas Normais

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004, Página 50

Na figura denominada formas normais, podemos observar graficamente as diversas fases de normalização descritas no texto.

(MACHADO, Felipe Nery Rodrigues, 2004, Página 204 a 206)

Roteiro de Aplicação da Normalização

- Aplicação da 1FN

- Decompor a entidade em uma ou mais entidades, sem grupos repetitivos.
- Destacar um ou mais atributos como chave primária das novas entidades, e este será concatenado com a chave primária da entidade original.
- Estabelecer o relacionamento e a cardinalidade entre as novas entidades geradas e a entidade geradora.
- Verificar a questão da variação temporal de certos atributos e criar relacionamentos 1:N entre a entidade original e a entidade criada por questões de histórico.

- Aplicação da 2FN

- Para entidades que contenham chaves primárias concatenadas, destacar os atributos que tenham dependência parcial em relação à chave primária concatenada.
- Criar uma nova entidade que conterá esses atributos, e que terá como chave primária os atributos dos quais se tenha dependência parcial.
- Serão criadas tantas entidades quanto forem os atributos da chave primária concatenada, que gerem dependência parcial.
- Estabelecer o relacionamento e a cardinalidade entre as novas entidades geradas e a entidade geradora.

- Aplicação da 3FN

- Verificar se existem atributos que sejam dependentes transitivos de outros que não pertencem à chave primária, sendo ela concatenada ou não, bem como atributos que sejam dependentes de cálculo realizado a partir de outros atributos.
- Destacar os atributos com dependência transitiva, gerando uma nova entidade com esse atributo e cuja chave primária é o atributo que originou a dependência.
- Eliminar os atributos obtidos através de cálculos realizados a partir de outros atributos.

- Aplicação FNBC

- Só aplicável em entidades que possuam chaves primárias concatenadas.

- Verificar se alguma chave candidata concatenada é um determinante, e em caso afirmativo, criar uma entidade com os que dependam funcionalmente desse determinante e cuja chave primária é o próprio determinante.
- Aplicação da 4FN
 - Verificar se a entidade possui atributos que não sejam participantes da chave primária e que sejam multivalorados e independentes em relação a um mesmo valor da chave primária.
 - Retirar esses atributos não chaves e multivalorados, criando novas entidades para cada um deles, herdando a chave primária da entidade desmembrada.
- Aplicação da 5FN
 - A ocorrência deste tipo de forma normal está vinculada aos relacionamentos múltiplos ou entidades que possuam chave primária concatenada com três ou mais atributos.
 - Verificar se é possível reconstruir o conteúdo do elemento original a partir de elementos decompostos desta.
 - Se não for possível, o elemento observado não está na 5FN, caso contrário, os elementos decompostos representam um elemento da 5FN.

1.8. Componentes das Bases de Dados

Se a base de dados fosse um edifício, as tabelas seriam os pilares, os relacionamentos entre as tabelas seriam as vigas, as consultas seriam as lajes e as paredes, os formulários seriam as portas de entrada, os relatórios seriam as portas de saída, e as pessoas que lá habitam seriam os dados.

Embora existam outros componentes numa base de dados, vamos aqui incidir a nossa análise apenas sobre as tabelas e as consultas. São as tabelas, os relacionamentos entre elas e as consultas que melhor reflectem os modelo de dados.

1.8.1. Tabelas

Começemos por analisar o que dizem alguns autores sobre a tabela. “Uma tabela representa o conjunto dos elementos de uma entidade abstracta do mundo real. As entidades do mundo real representam coisas, seres, acontecimentos geradores de informação. Essa informação

resulta da observação e registo dos seus atributos” (CARRIÇO, José António; CARRIÇO, António João, 1997, Página 68)

“A Tabela é uma organização de dados em uma tabela bidimensional, na qual as linhas (tuplas) representam entidades básicas ou factos de algum tipo, e colunas (atributos) representam propriedades dessas entidades” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer; Trad: SOUZA, Vandenberg D. de, 2001, Pagina 2).

Combinando informação da página 15 com informação da página 95 podemos afirmar que uma tabela é constituída por linhas e colunas. Uma linha pode também ser chamada de registo ou tupla. Uma coluna pode igualmente ser designada de campo ou atributo. (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. DE, 2001, Página 15 e 95)

A tabela é composta por campos e registros. “Campo é a unidade mais pequena de dados armazenados. A base de dados irá conter muitas ocorrências de cada um, dos vários tipos de campos armazenados. Registo é uma colecção de campos relacionados armazenados” (DATE, C.J., 2004, Página 22).

A denominada relação, normalmente dará origem a uma tabela. há autores que vão mais longe e afirmam que “*Relation is just a mathematical term for a table*” (DATE, C.J., 2004, Página 64).

No entanto reconhecem que há diferenças entre tabelas e relações. “ A tabela pode conter valores duplicados mas uma relação não. As tabelas devem ter pelo menos uma coluna, uma relação pode não ter nenhum atributo. As tabelas podem incluir valores nulos, as relações não. As tabelas têm duas dimensões, as relações podem ter n dimensões” (DATE, C.J., 2004, Página 151).

1.8.1.1. Superchave

O relacionamento entre as tabelas só é possível devido à existência de chaves. Começemos por analisar a superchave. “Uma superchave é um identificador único de uma tupla, baseado em vários atributos. Se t_1 e t_2 forem duas quaisquer tuplas de uma relação R , se S e K forem dois atributos da mesma relação R . Então a dupla de atributos $[SK]$ forma uma superchave, se qualquer tupla t_1 for diferente de qualquer tupla t_2 na dupla de atributos S e K .

$t_1[SK] \neq t_2[SK]$

Todas as relações têm pelo menos uma superchave, baseada no conjunto de todos os seus atributos. Uma superchave pode ter atributos redundantes” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 203).

“A Superchave surge da associação de um ou mais atributos cujos valores, em conjunto, identificam univocamente cada tuplo. Como é evidente, no limite, a associação de todos os atributos de uma relação constitui uma superchave” (PEREIRA, José Luís, 1998, Página 166).

1.8.1.2. Chave

Passemos agora ao conceito de chave. “ Chave é uma superchave, à qual não podemos retirar nenhum atributo, sob pena de deixar de ser uma superchave” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 203)

1.8.1.3. Chave candidata

Por vezes as tabelas apresentam mais do que uma chave, “se uma relação tiver mais que uma chave, cada uma das diferentes chaves, chama-se chave candidata” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 203)

Uma chave candidata é o “subconjunto dos atributos de uma superchave que, sendo ainda superchave, não pode ser reduzido sem perder essa qualidade” (PEREIRA, José Luís, 1998, Página 166).

“Uma chave candidata a chave primária terá que possuir as seguintes propriedades:

- Ser unívoca - O atributo candidato não pode conter duas tuplas com o mesmo valor.
- Ser Irredutível – Se for subdividida deixa de ser unívoca, portanto não pode ser reduzida” (DATE, C.J., 2004, Página 269).

1.8.1.4. Chaves Primárias

De entre as diversas chaves candidatas, é necessário escolher uma para chave primária. “Chave primária é a chave candidata escolhida, cujos valores vão ser utilizados para identificar as tuplas da relação. Quando existem várias chaves candidatas o processo de escolha da chave primária torna-se arbitrário. No entanto costuma-se escolher para chave

primária a chave candidata que tiver menor numero de atributos e que não tenha valores nulos” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 203 e 204).

“Escolher das chaves candidatas uma para chave primária pode ser uma boa ideia em muitos casos, possivelmente na maioria dos casos, mas não pode ser justificado em todos os casos” (DATE, C.J., 2004, Página 272)

“Uma chave primária é um conjunto de um ou mais atributos que permitem identificar unicamente uma entidade de entre um conjunto de entidades” (COSTA PINTO, Manuel Luís da, 1996, Pagina 13).

“Cada elemento de uma entidade representa um objecto com uma identidade própria, um objecto identificável. O conjunto dos atributos definidos para uma entidade deve incluir pelo menos um atributo que identifique, de forma exclusiva, cada elemento dessa entidade, no contexto da base de dados. Um atributo identificador é um atributo que não admite valores repetidos. Não podem existir duas linhas numa tabela que tenham o mesmo valor no atributo identificador. Numa tabela podem existir vários atributos identificadores. Qualquer atributo que seja identificador é designado por chave candidata. Dentre as chaves candidatas, o analista deve escolher uma que funcione como atributo identificador dessa tabela na relação com as outras tabelas da base de dados. A chave escolhida para funcionar como atributo identificador da tabela é designado por chave primária. Uma chave primária pode ser constituída por um só atributo ou por mais do que um atributo” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 68,69).

Índices unidimensionais pressupõem uma chave de pesquisa única. A chave de pesquisa é um campo único. Nas aplicações geográficas, os dados são elementos dum mundo bidimensional, ou às vezes tridimensional. (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Página 201 e 202)

1.8.1.5. Chaves Estrangeiras ou Externas

As chaves estrangeiras têm um papel fundamental nos relacionamentos entre as tabelas.

“Numa tabela designam-se por chaves externas os atributos que são chaves primárias de outras tabelas. Uma chave externa constitui o elemento que relaciona a linha da tabela onde está inserida com a linha de outra tabela” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 70)

“Uma chave estrangeira é um conjunto de atributos de uma relação cujos valores coincidem com os valores da chave candidata de outra relação. A chave estrangeira pode ser simples ou composta dependendo da chave candidata” (DATE, C.J., 2004, Página 272).

“Chave estrangeira ou chave importada - trata-se de um conjunto constituído por um ou mais atributos que é chave primária numa outra relação. A existência de uma chave estrangeira numa relação prende-se com a necessidade de manter a interligação entre essa relação e a relação onde esse conjunto de atributos é chave principal” (PEREIRA, José Luís, 1998, Página 167)

Se preferirmos, numa linguagem mais matemática temos a seguinte “igualdade:

$$t1[FK] = t2[PK]$$

t1 – tupla da relação 1

t2 – tupla da relação 2

FK – chave estrangeira

PK – chave primária

t1[FK] – tupla da chave estrangeira da relação 1

t2[PK] – tupla da chave primária da relação 2

Uma chave estrangeira também se pode referir à sua própria relação” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 207).

Não nos podemos esquecer que uma tabela pode conter mais do que uma chave estrangeira. “A uma coluna de uma tabela, cujo valor coincide com a chave primária de outra tabela, designa-se por chave estrangeira. Uma tabela pode conter mais de uma chave estrangeira, pois pode estar relacionada com mais de uma tabela adicional” (COSTA PINTO, Manuel Luís da, 1996, Pagina 20).

Na figura denominada chave estrangeira podemos observar que a tabela produtos encomendados contem duas chaves estrangeiras ligadas a duas chaves primárias de outras duas tabelas. O campo numero de encomenda encontra-se ligado ao seu homólogo na tabela encomenda. O campo código de produto encontra-se ligado ao seu homólogo na tabela do produto.

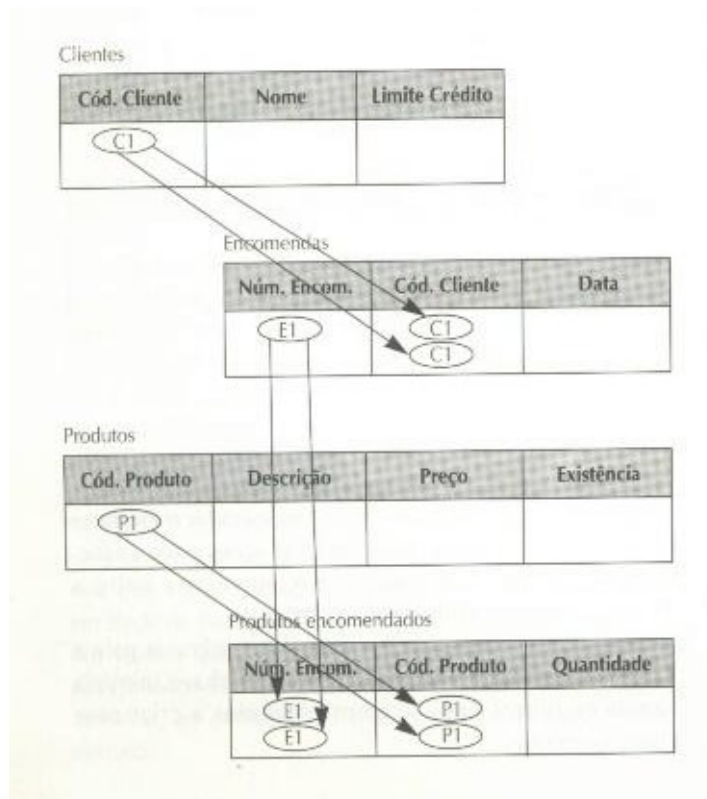


Fig. Chave estrangeira

Fonte: COSTA PINTO, Manuel Luís da, 1996, Pagina 20

1.8.1.6. Relacionamentos

O funcionamento das bases de dados baseadas no modelo ER, só é possível devido à existência de relacionamentos entre as tabelas. “Um relacionamento é definido como uma associação entre várias entidades. O mapeamento da associação entre entidades pode corresponder a uma das seguintes 4 situações:

- Relacionamento de uma para uma – Um elemento de do conjunto de entidades A está associado a um e só um elemento do conjunto de entidades de B e vice versa.
- Relacionamento de um para muitas – Um elemento de A está associado a um qualquer número de entidades de B. Um elemento de B está somente associado com uma entidade de A.
- Relacionamento de muitas para uma – Um elemento de A está associado a uma e só uma entidade de B, enquanto que um elemento de B pode estar associado a qualquer numero de entidades de A.
- Relacionamento de muitas para muitas – Um elemento de A está associado a um qualquer numero de entidades de B, e um elemento de B pode estar associado a um qualquer numero de entidades de A” (COSTA PINTO, Manuel Luís da, 1996, Página 12 e 13)

1.8.1.7. Integridade Referencial

O conceito de integridade referencial é essencial para permitir proteger a integridade dos dados. “O princípio da integridade referencial estabelece que o valor de uma chave externa tem de existir na tabela onde essa chave é primária. Por outras palavras, não pode ser inscrito um valor num campo de uma chave externa se esse valor não existir já na tabela de origem” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 70). Podemos verificar na figura sobre integridade referencial um exemplo bem explicito duma falha ocorrida, por não ter sido reforçada a integridade referencial.

JOGADORES			JOGOS			
IdJogador	Nome	Data de nascimento	IdJogo	Estádio	Adversário	Data
J1	Alves	1970 - 10 - 08	1	Estádio A	Equipa A	1996 - 01 - 05
J2	Peres	1972 - 08 - 20	2	Estádio B	Equipa C	1996 - 02 - 15
J3	Paulo	1974 - 01 - 20	3	Estádio C	Equipa Y	1996 - 02 - 25

PARTICIPAÇÕES		
IdJogador	IdJogo	Golos
J1	1	0
J2	1	1
J3	2	0
J1	2	1
J152	5	2

Fig. Integridade referencial

Fonte: CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 70

Quando a integridade referencial está reforçada, “ A base de dados não pode conter na chave estrangeira valores não relacionados, ou seja, não pode conter na chave estrangeira valores que não existam na chave candidata. Se o valor existe na chave estrangeira terá que existir na chave primária” (DATE, C.J., 2004, Página 274).

Posto de outra forma, podemos afirmar que “as restrições da integridade referencial são especificadas entre duas relações e são usadas para manter a consistência entre as tuplas das duas relações. Informalmente podemos afirmar que a restrição da integridade referencial, especifica que uma tupla numa relação que se refere a outra relação, deve referir-se a uma tupla que exista nessa outra relação” (ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 206).

“Frequentemente, desejamos garantir que um valor que aparece em uma relação para um dado conjunto de atributos também apareça para um certo conjunto de atributos de outra relação. Essa condição é chamada de integridade referencial” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 193).

1.8.2. Vistas ou Consultas

Conforme afirmámos anteriormente, se a base de dados fosse um edifício, as tabelas seriam os pilares, os relacionamentos entre as tabelas seriam as vigas, as consultas seriam as lajes e as paredes. Tal como os pilares e as vigas condicionam as lajes e as paredes de um edifício, as tabelas e os seus relacionamentos também condicionam as consultas. Mas comecemos por analisar o que é uma consulta, “Uma consulta à base de dados consiste na execução de uma ou mais operações sobre as tabelas” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Pagina 67). Ou se preferirem “consulta é uma operação que extrai dados especificados da base de dados” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Pagina 2).

O resultado final de uma consulta ou vista, é semelhante ao de uma nova tabela ou relação. “Qualquer relação que não faça parte do modelo lógico, mas é visível para o utilizador como uma relação virtual, é chamada de vista” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 99).

“As vistas servem para providenciar:

- Uma capacidade semelhante a um atalho ou uma macro.
- Que os mesmos dados sejam vistos por utilizadores diferentes de formas distintas ao mesmo tempo.
- Segurança automática para dados escondidos.
- Independência lógica dos dados, ou seja a estrutura lógica da base de dados é imune a utilizadores e programas” (DATE, C.J., 2004, Página 298).

Embora pareçam relações, as vistas não o são, e conseqüentemente não devem ser tratadas com tal. “Vista é o resultado da aplicação de uma expressão a uma relação. As vistas são uma técnica legítima utilizada para extracção de informação, mas não devem ser utilizadas para actualizar informação” (DATE, C.J., 2004, Página 302 e 303).

1.8.2.1. O Processador de Consultas

“O processador de consultas é representado por dois componentes:

- O compilador de consultas – Converte a consulta para uma forma interna chamada plano de consulta. Plano de consulta é uma sequência de operações a serem executadas sobre os dados. Com frequência, as operações num plano de consulta são implementações de álgebra relacional. O compilador de consultas usa metadados e estatísticas sobre os dados para decidir que sequência de operações provavelmente será a mais rápida.
- O mecanismo de execução – Tem a responsabilidade de executar cada uma das etapas do plano de consultas escolhido” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Página 11).

1.8.2.2. Álgebra Relacional

“A álgebra relacional é uma linguagem de consultas *procedural*. Consiste em um conjunto de operações tendo como entrada uma ou duas relações produzindo, como resultado, uma nova relação” (SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDAESHAN, S., 1999, Página 69).

“A álgebra relacional é uma coleção de operadores que toma as relações como seus operandos e devolve relações como seus resultados. A primeira versão da álgebra relacional foi definida por Codd com base em oito operadores divididos em dois grupos de quatro. O primeiro grupo de quatro operadores tradicionais compostos por união, intersecção, diferença e produto cartesiano. O segundo grupo de quatro operadores relacionais compostos por selecção, projecção, junção e divisão” (DATE, C.J., 2004, Página 173).

“A álgebra relacional foi projectada originalmente como se as tabelas fossem conjuntos. Na verdade as tabelas em SQL são conjuntos múltiplos, isto é, a mesma linha pode aparecer mais de uma vez numa relação de SQL. Desse modo, devemos introduzir a álgebra relacional como uma álgebra sobre tabelas. Os operadores de álgebra relacional são:

- União – No caso de R união com S, um registo t está no resultado tantas vezes quanto o numero de vezes que ele está em R mais o numero de vezes que ela está em S.
- Intersecção – No caso de R intersecção com S, um registo t está no resultado no mínimo o numero de vezes que ela está em R e S.
- Diferença - No caso de R menos S, um registo está no resultado o numero de vezes que ele está em R menos o numero de vezes que está em S, mas não um numero de vezes menor que zero.

- Seleccção – Este operador produz uma nova tabela a partir de uma antiga, seleccionando algumas linhas da tabela antiga com base em alguma condição ou algum predicado. Corresponde em linhas gerais à cláusula WHERE duma consulta de SQL.
- Projecção – Este operador produz uma nova tabela a partir de uma antiga, escolhendo algumas colunas, como a cláusula SELECT de uma consulta de SQL.
- Produto – Este operador é o produto cartesiano da teoria dos conjuntos, que constrói registos emparelhando as linhas de duas tabelas de todas as maneiras possíveis. Corresponde em SQL à lista de tabelas duma cláusula FROM, cujo produto forma a relação à qual são aplicadas a condição da cláusula WHERE e a projecção da cláusula SELECT.
- Junção – Existem vários tipos de operadores de junção, que correspondem a operadores como JOIN, NATURAL JOIN e OUTER JOIN do padrão SQL2.
 - A junção mais simples e mais comum é a junção natural. A junção natural das tabelas R e S é uma abreviação da projecção de L da selecção de C do produto de R por S onde C é uma condição que compara todos os pares de atributos de R e S que têm o mesmo nome. L é uma lista de todos os atributos de R e S, excepto pelo facto de uma cópia de cada par de atributos comparada ser omitida.
 - A junção theta ou equivalente é uma abreviação da selecção C do produto de R por S. A condição C é um único termo da forma x igual a y , onde x é um atributo de R e y é um atributo de S” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Página 257 a 264)

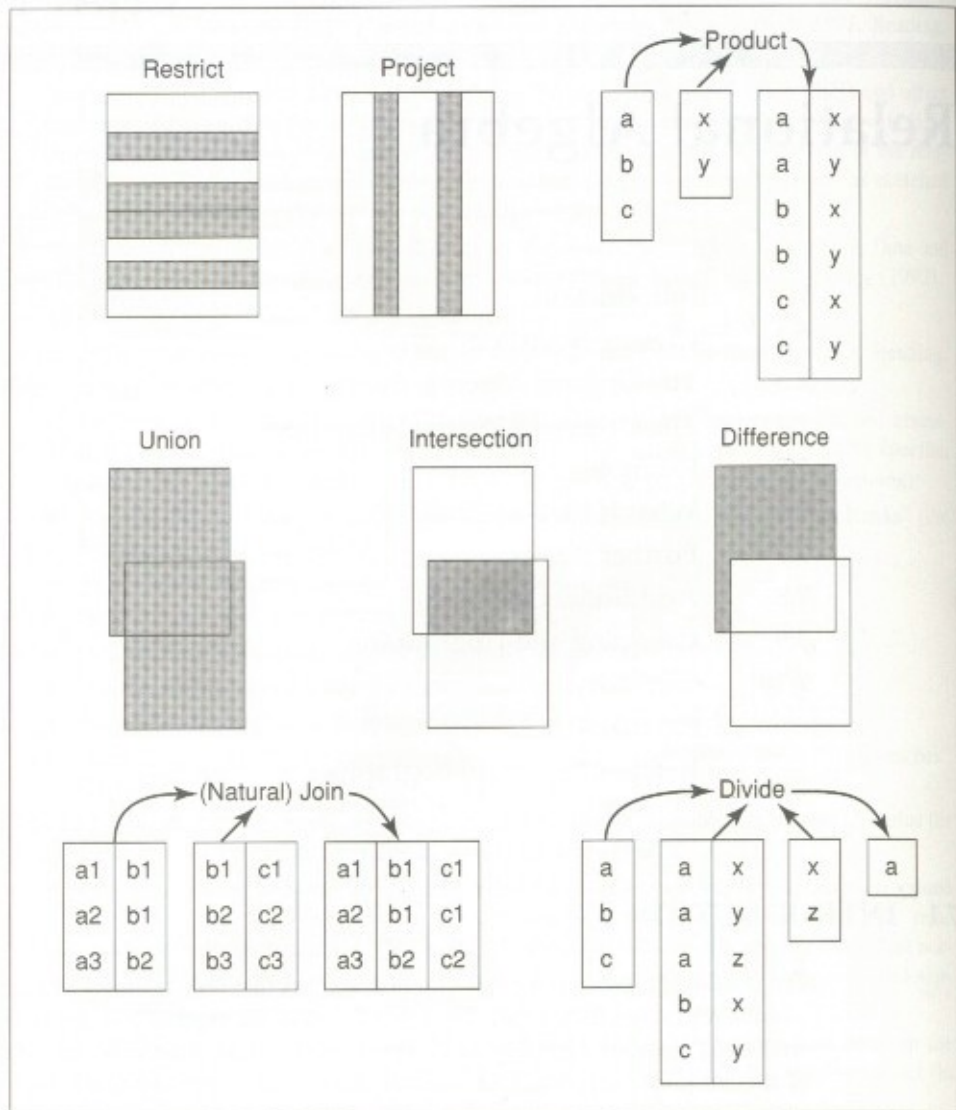


Fig. 7.1 The original eight operators (overview)

Fig. As oito operações originais
 Fonte: DATE, C.J., 2004, Página 178

Como podemos observar na figura das oito operações originais, na álgebra relacional temos uma semântica que define cada uma das operações. “Semântica da álgebra original:

- União – A união de dois conjuntos é o conjunto de todos os elementos que pertencem a ambos os conjuntos. O resultado da união de duas relações será o conjunto das linhas das duas relações. A união de duas relações só é possível quando os atributos das duas relações são os mesmos.

- Intersecção – Tal como na união, a intersecção de duas relações só é possível quando os atributos das duas relações são os mesmos. A intersecção de duas relações, dá origem a uma nova relação composta somente pelas tuplas que surgem em ambas as relações.

- Diferença - Tal como na união, e na intersecção, a diferença de duas relações só é possível quando os atributos das duas relações são os mesmos. O resultado da diferença entre duas relações é uma relação contendo as tuplas da primeira relação que não constam na segunda relação.

- Produto – Em termos matemáticos o produto cartesiano de dois conjuntos é o conjunto de todos os pares ordenados, nos quais, o primeiro elemento vem do primeiro conjunto e o segundo elemento vem do segundo conjunto. O produto cartesiano de duas relações será um conjunto de pares ordenados das duas tuplas, nas quais, a primeira tupla vem da primeira relação e a segunda tupla vem da segunda relação.

- Restrições – Quando se aplica um operador com uma restrição condicional, a uma dada relação, o resultado será uma nova relação, composta pelas tuplas da primeira relação que satisfazem a condição imposta.

- Projecção – O resultado da projecção de uma relação, é a mesma relação apenas com os atributos mencionados, depois de eliminadas as tuplas duplicadas.

- Junção – Há diversos tipos de junções, mas a mais importante é conhecida por junção natural. Para se poder proceder à junção natural, é necessário que as duas relações tenham um atributo em comum. O resultado da junção natural será uma relação com todos os atributos das duas relações, em que cada tupla será composta pelo atributo comum das duas relações. Além do atributo comum, cada tupla contará com os respectivos atributos das duas relações respeitantes à tupla do atributo comum.

- Divisão – Há dois tipos de divisão, a grande divisão e a pequena divisão.

Na pequena divisão, o resultado da divisão de uma relação chamada dividendo por outra relação chamada divisor com base numa relação chamada mediador, consiste numa nova relação composta pelos valores do dividendo que quando associados ao divisor, existem no mediador” (DATE, C.J., 2004, Página 180 a 189)

Relativamente à nomenclatura, “A maioria dos textos de bases de dados utilizam a seguinte notação, selecção (δ), junção (\bowtie), intersecção (\cap) e projecção (π)” (DATE, C.J., 2004, Página 178)

Table 7.1 Operations of Relational Algebra

Operation	Purpose	Notation
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\langle \text{selection condition} \rangle}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\langle \text{attribute list} \rangle}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$, or $R_1 \bowtie_{(\langle \text{join attributes 1} \rangle, \langle \text{join attributes 2} \rangle)} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 \ltimes_{\langle \text{join condition} \rangle} R_2$, or $R_1 \ltimes_{(\langle \text{join attributes 1} \rangle, \langle \text{join attributes 2} \rangle)} R_2$ or $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2 ; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2 ; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2 ; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2 .	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

Fig. Numenclatura utilizada nas operações

Fonte: ELMASRI, Ramez ; NAVATHE, Shamkant B., 1997, Página 234

“A intenção fundamental da álgebra relacional é permitir escrever expressões relacionais. De seguida são descritas algumas aplicações para essas expressões:

- Definição do âmbito de extracção de informação – Definir os dados a retirar numa operação de extracção de informação.
- Definir o âmbito de actualização – Definir os dados a serem inseridos, alterados ou apagados numa operação de actualização.
- Definir as restrições de integridade – Definir algumas restrições que a base de dados deve satisfazer.
- Definir as relações de variáveis derivadas – definir os dados a serem incluídos numa vista.
- Definir a estabilidade dos pedidos – Definir quais são os dados abrangidos por uma operação de controle de concorrência.

- Definir as restrições de segurança – Definir os dados, aos quais o acesso é garantido por algum tipo de autorização” (DATE, C.J., 2004, Página 193).

(DATE, C.J., 2004, Página 194)

A álgebra relacional ainda permite utilizar outras propriedades dos relacionamentos entre as relações (listas).

- Propriedade associativa

Se a, b e c são relações (listas) então

(a união b) união c

é equivalente a

a união (b união c)

- Propriedade comutativa

a união b

É equivalente a

b união a

1.8.2.3. SQL – Structured Query Language – Linguagem Estruturada de Consulta

A linguagem SQL é uma linguagem extremamente importante no mundo dos sistemas de gestão de bases de dados. “o SQL chegou para dominar o mercado dos SGBD. A razão prende-se com o facto do SQL se basear no modelo de dados relacional” (DATE, C.J., 2004, Página 26). “O SQL é a linguagem standard para sistemas relacionais, e é suportada por praticamente todos os produtos de bases de dados existentes hoje no mercado” (DATE, C.J., 2004, Página 85).

A linguagem SQL serve acima de tudo para extrair dados de uma base de dados, “a SQL é uma linguagem especial, criada especificamente para o acesso a bases de dados relacionais. Em última análise, o conjunto de comandos SQL traduz-se numa ou várias operações efectuadas sobre tabelas. O resultado de uma operação de consulta aparece sempre sobre a forma de uma tabela” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Pagina 65).

Como o SQL é uma linguagem que trabalha sobre a base de dados, “usa o termo tabela no lugar de relação, o termo linha no lugar de tupla e o termo coluna no lugar de atributo. O comando CREATE TABLE especifica o nome da tabela a ser criada, o nome e tipos das colunas dessa tabela, as chaves primárias e as chaves estrangeiras. O comando SELECT

permite executar diversas funções, entre elas, impor restrições, efectuar projecções e junções” (DATE, C.J., 2004, Página 86).

“A linguagem de base de dados SQL tem um grande numero de recursos, inclusive instruções que consultam e modificam a base de dados. A modificação da base de dados é feita através de três comandos, INSERT, DELETE e UPDATE.

O resultado duma consulta pode ser calculado:

- Tomando-se todas as combinações possíveis de linhas das tabelas na cláusula FROM.
- Eliminando-se tudo o que não satisfaz a condição da clausula WHERE.
- Agrupando-se as linhas restantes, de acordo com os seus valores nas colunas mencionadas na cláusula GROUP BY.
- Testando-se cada grupo de acordo com a condição na cláusula HAVING e rejeitando-se todos os grupos que não satisfazem essa condição.
- Calculando-se linhas de colunas e agrupamentos de colunas especificados, da maneira especificada pela cláusula SELECT.
- Ordenando-se as linhas resultantes de acordo com os valores na lista de atributos da cláusula ORDER BY” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Página 16).

De seguida podemos observar uma figura sobre as queries, nesta figura podemos observar alguns exemplos de consultas. Nos primeiros exemplos são impostas condições, restrições aos dados, nos exemplos seguintes é reunida na mesma consulta informação de várias tabelas.

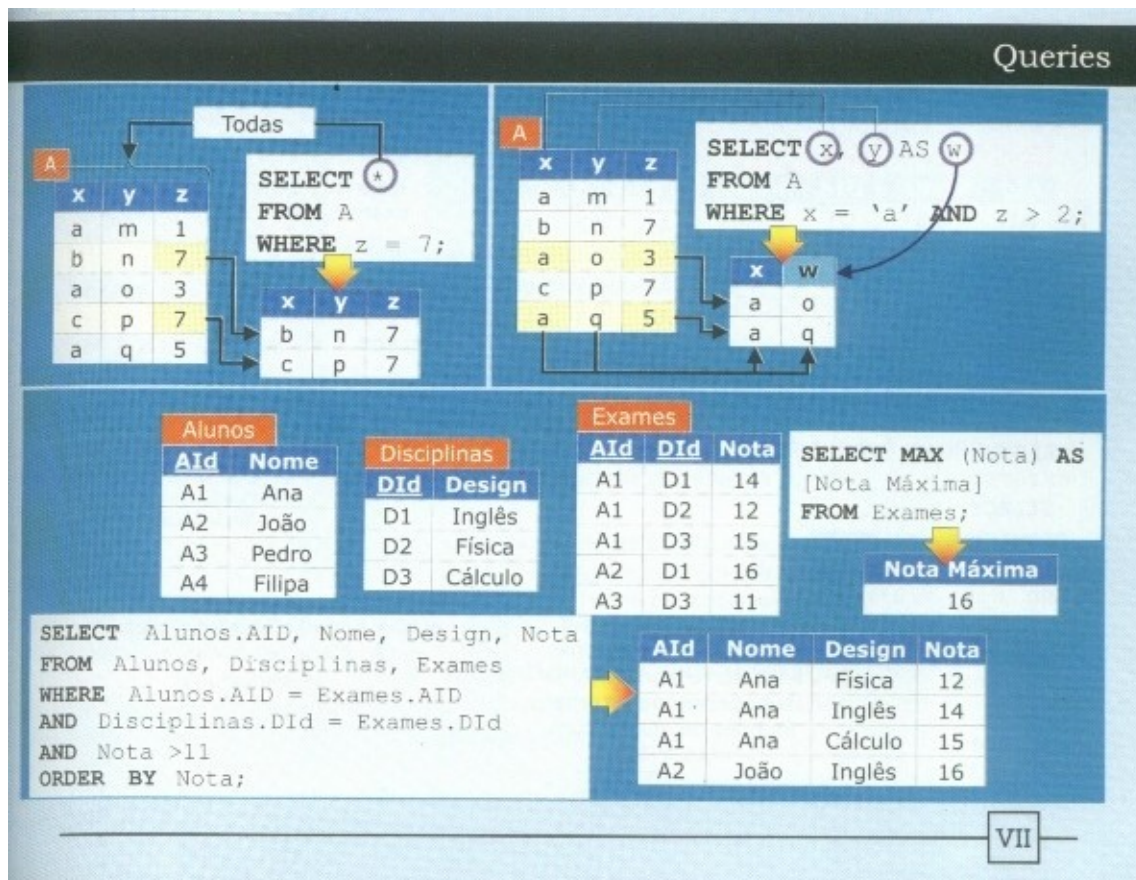


Fig. Queries

Fonte: GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ;

Trad: SOUZA, Vandenberg D. de, 2001, Página 16

Restrições

“Uma restrição é uma operação aplicada sobre uma tabela com o objectivo de extrair um subconjunto das suas linhas. O subconjunto de linhas que obedecem a uma ou mais condições especificadas na operação de restrição” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Pagina 71).

Em SQL o comando "WHERE condição filtra dados indesejados do resultado de uma consulta, devolvendo apenas os dados que satisfazem a condição” (KLINE, Kevin E.; KLINE, Daniel; HUNT, Brand, 2004, Página 384).

Na figura operação restrição, podemos observar um exemplo de uma restrição imposta aos dados de uma coluna de uma tabela.

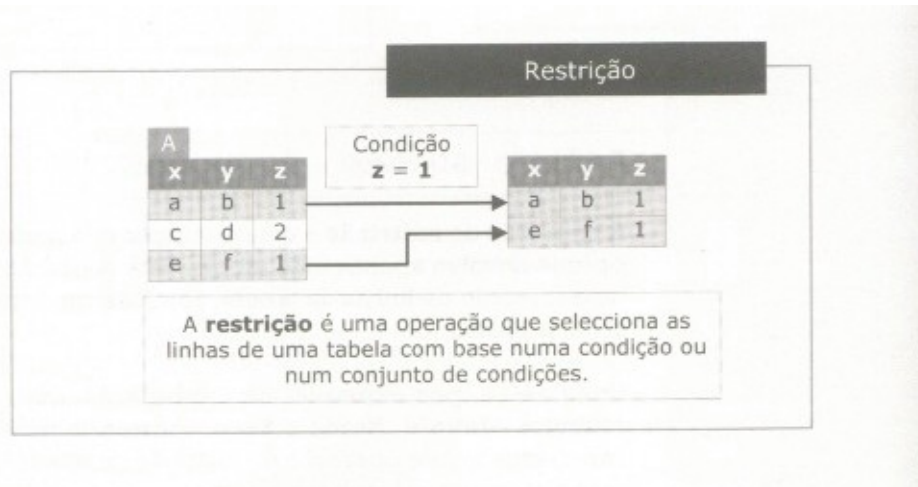


Fig. Operação Restrição

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004, Página 90

Projecções

“Uma projecção é uma operação que selecciona uma ou mais colunas de uma tabela” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 71).

No SQL o comando “SELECT item pode ser utilizado como função, calculo matemático, parâmetro, variável ou uma subconsulta, mas normalmente é utilizado para definir uma coluna duma tabela ou duma vista” (KLINE, Kevin E.; KLINE, Daniel; HUNT, Brand, 2004, Página 382).

Na figura da operação de projecção podemos visualizar um exemplo em que retiramos da tabela A as colunas X e Y, e lhes mudamos o nome para K e M respectivamente.

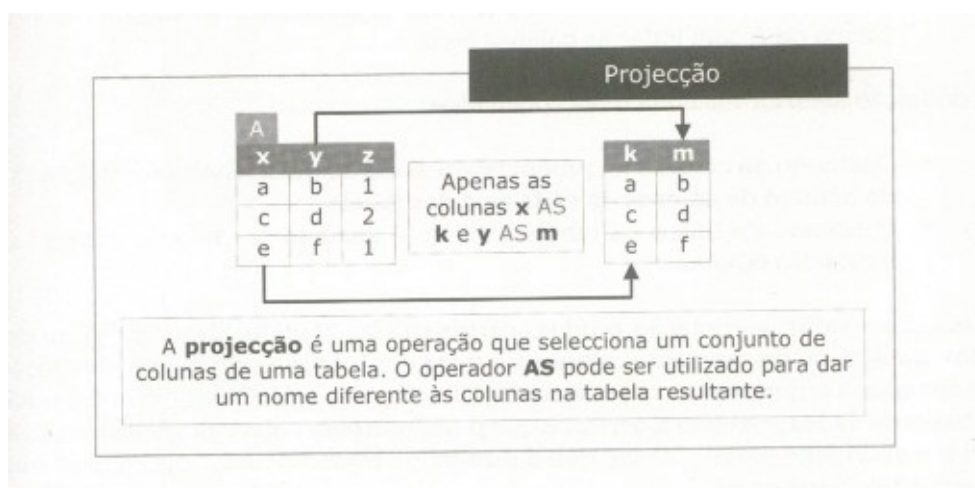


Fig. Operação Projecção

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004 Página 91

Uniões

“A união é uma operação através da qual duas tabelas dão origem a uma terceira tabela, constituída pelas linhas que pertencem a uma ou a outra tabelas. Para que se possa efectuar a união, é necessário que as duas tabelas tenham uma estrutura idêntica. É evidente que a união pode estender-se a mais do que duas tabelas” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 72).

Em SQL o comando “UNION permite juntar duas tabelas com o mesmo numero de colunas. O tipo de dados não tem que ser obrigatoriamente o mesmo desde que a conversão dos mesmos esteja implícita” (KLINE, Kevin E.; KLINE, Daniel; HUNT, Brand, 2004, Página 459).

Na figura da operação união temos um exemplo do funcionamento de uma união entre duas tabelas. É importante referir que as tabelas do exemplo, apresentam-se à união com os mesmos campos. É de referir igualmente que o resultado final também apresenta os mesmos campos.

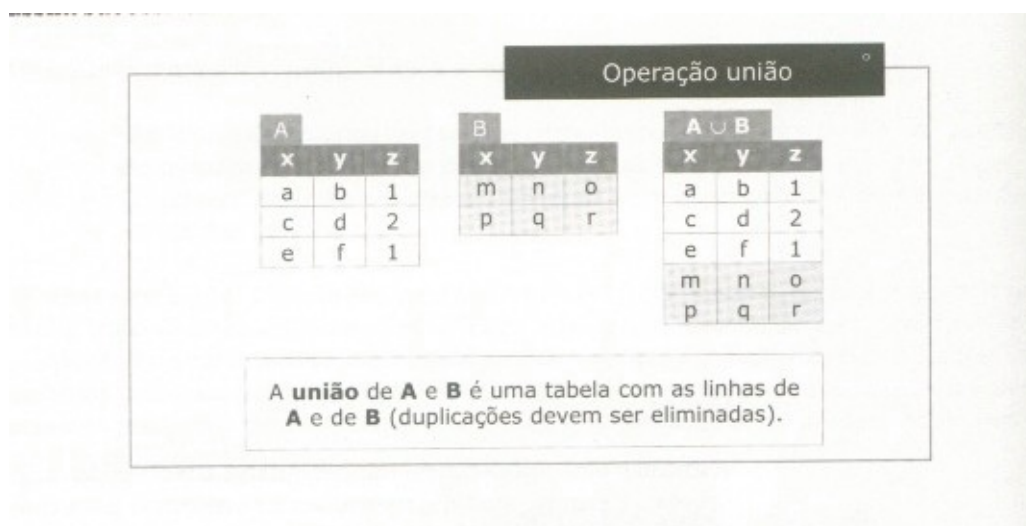


Fig. Operação união

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004, Página 94

Intersecções

“A intersecção conjuga duas tabelas, originando uma terceira tabela contendo as linhas que são comuns. Também neste caso, é necessário que as tabelas tenham a mesma estrutura, para que a operação se possa realizar” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 72)

O comando “INTERSECT escolhe as linhas de duas ou mais tabelas em que a linha da primeira tabela e da segunda tabela são idênticas” (KLINE, Kevin E.; KLINE, Daniel; HUNT, Brand, 2004, Página 326)

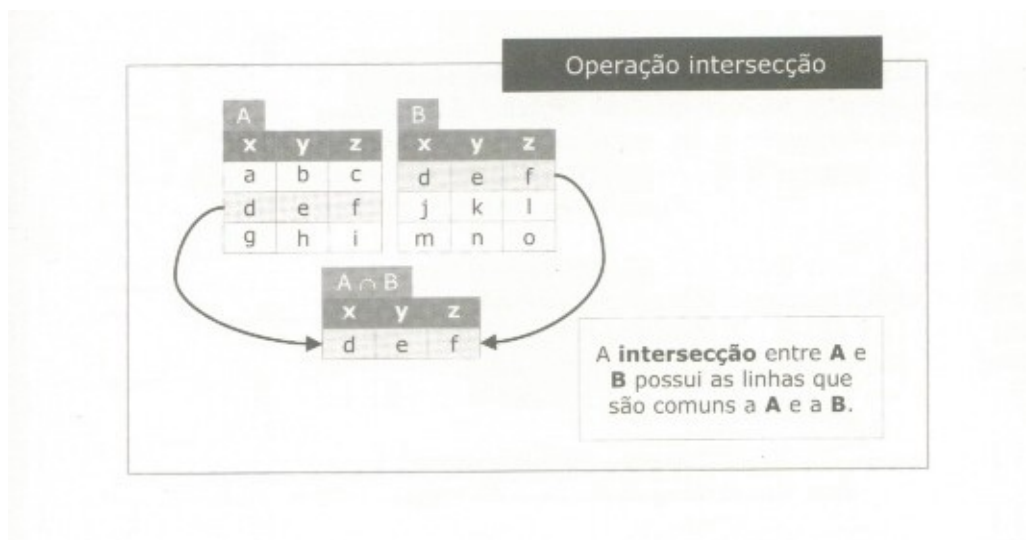


Fig. Operação intersecção

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004, Página 95

Junção

“A operação junção é uma das operações mais frequentes utilizadas em bases de dados relacionais. Informalmente, pode definir-se como a junção dos elementos de tabelas tomando por base um atributo comum.” (CARRIÇO, José António ; CARRIÇO, António João, 1997, Página 72).

O comando “JOIN condição - Junta informação de duas tabelas que tenham um campo comum. Normalmente o campo comum é chave primária numa das tabelas e chave estrangeira na outra tabela. Há vários tipos de joins:

- Cross join - Provoca o produto de tabelas, juntando para cada registo da primeira tabela, todos os registos da segunda tabela, criando potencialmente um resultado gigantesco.

- Inner join - Especifica que todas as linhas que não tiverem o mesmo dado nas duas tabelas, não terão lugar na consulta. Por defeito é este join que é utilizado.
- Left outer join - Devolve todos os valores da tabela da esquerda. Quando a tabela da direita não tiver dados condizentes com a da esquerda, os dados são preenchidos com valores nulos.
- Right outer join - Devolve todos os valores da tabela da direita. Quando a tabela da esquerda não tiver dados condizentes com a da direita, os dados são preenchidos com valores nulos.
- Full outer join - Devolve todos os valores das duas tabelas preenchendo com valores nulos todos os valores que não existem na outra tabela” (KLINE, Kevin E.; KLINE, Daniel; HUNT, Brand, 2004, Página 383 e 386).

“A operação Join pode ser representada através do símbolo \bowtie . A tabela que resulta do Join da tabela A e da tabela B pode ser representado através da expressão $A \bowtie B$. Esta tabela é o resultado de uma projecção, de uma restrição de um produto cartesiano. O produto cartesiano é $A \times B$. O resultado deste produto é objecto de uma restrição. Essa restrição selecciona apenas as linhas onde se verifique a igualdade dos valores do atributo comum. Finalmente, uma projecção extrai um subconjunto do conjunto das tabelas produto” (CARRIÇO, José António; CARRIÇO, António João, 2004, Página 98).

Na figura da operação de junção podemos observar a junção da tabela A com a tabela B, com base nos dados da coluna Z. É de salientar que a coluna Z é comum nas duas tabelas.

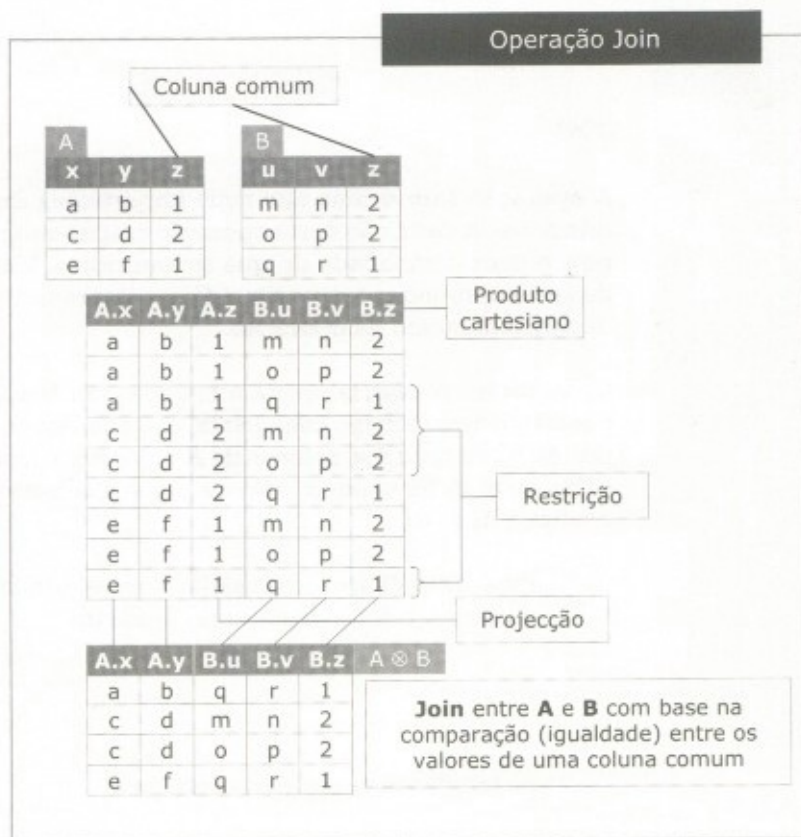


Fig. Operação junção

Fonte: CARRIÇO, José António; CARRIÇO, António João, 2004, Página 98

Agrupamento e Agregação

GROUP BY

O comando GROUP BY do SQL é utilizado para agrupar informação numa consulta. Quando juntamos na mesma consulta dados de duas tabelas que têm entre si uma relação de um para muitos, temos como resultado uma consulta com registos que normalmente faz sentido serem agrupados. Estes dados depois de agrupados, são normalmente sujeitos a determinadas operações, que fazem sentido ser aplicadas a cada um dos grupos obtidos. “Existe uma família de características em SQL que funcionam em conjunto para permitir consultas envolvendo agrupamento e agregação.

- Operadores de Agregação – Os cinco operadores AVG, SUM, COUNT, MIN e MAX produzem os valores médio, total, de contagem mínimo ou máximo, respectivamente, do atributo ao qual são aplicados. Esses operadores são utilizados em cláusulas SELECT.

-Agrupamento – Uma cláusula GROUP BY numa consulta de SQL faz a tabela construída pelas cláusulas FROM e WHERE ser agrupada de acordo com o valor do atributo ou dos atributos mencionados na cláusula GROUP BY. As agregações são então aplicadas ao grupo.

- HAVING – A cláusula HAVING deve seguir uma cláusula GROUP BY e fornece um condição a que um grupo deve satisfazer, a fim de contribuir para o resultado da consulta” (GARCIA-MOLINA, Hector ; ULLMAN, Jeffrey D.; WIDOM, Jennifer ; Trad: SOUZA, Vandenberg D. de, 2001, Página 266).

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.