Unidade II

3 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

O diagrama E-R é uma ferramenta para modelagem conceitual de banco de dados amplamente utilizada no projeto de banco de dados, sendo considerada praticamente um padrão para modelagem conceitual. É aconselhado como padrão por ser de fácil compreensão e apresenta poucos conceitos.

Um diagrama E-R é uma representação gráfica, na forma de um diagrama, no qual são utilizados apenas três conceitos: entidades, relacionamentos e atributos, que serão descritos a seguir.

3.1 Entidades do DER

Representam categorias de fatos do mundo real, sejam eles concretos ou abstratos, como empregados, departamentos, despesas etc. São representados por retângulos nomeados por substantivos, que indicam um conjunto de ocorrências da entidade. Sugerem-se nomes no plural.

3.2 Relacionamentos do DER

Representam associações entre entidades, sendo, em cada associação, indicadas as cardinalidades, ou seja, o número de ocorrências de uma entidade que se relaciona com uma ocorrência de outra entidade.

Por exemplo, como representado na Figura 8.

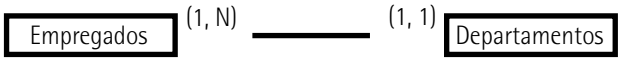


Figura 8 – cardinalidade de um relacionamento

- Indica que 1 empregado trabalha no mínimo em 1 departamento e no máximo em 1 departamento.
- Indica que em 1 departamento trabalha no mínimo 1 empregado e no máximo N empregados.

3.2.1 Relacionamentos binários

Representam uma associação entre duas entidades, representada por um losango nomeado com linhas para as duas entidades envolvidas. Um relacionamento indica uma função cumprida pelas duas entidades envolvidas. Sugere-se que esse nome seja um substantivo no plural, uma vez que o uso de verbos limita mais a criatividade para a determinação de nomes.

Quando se define um relacionamento, deve-se definir também a cardinalidade dessa associação, por meio de dois pares de valores, no qual o valor da esquerda indica o número mínimo esperado de ocorrências de uma entidade que se relaciona com a entidade questionada, e o valor da direita o número máximo. Esses valores de máximo e mínimo podem ser constantes, quando se souber exatamente esses valores.

3.2.2 Relacionamentos reflexivos

É um tipo de associação que envolve ocorrências de uma mesma entidade (parte e chega da mesma entidade). Nesse caso, recomenda-se a indicação explícita do papel assumido por cada lado da associação, para tornar mais clara a interpretação do relacionamento. Por exemplo, em uma relação de chefia entre um empregado e outro, é interessante a indicação dos papéis de superior e subordinado.

Alguns controles de integridade, não expressos no diagrama E-R, podem ser necessários, como a proibição de um empregado ser chefiado por ele mesmo (valores iguais no relacionamento) ou uma hierarquia de gerência com ciclos. É interessante que esses controles, se desejados, sejam explicitamente comentados como um anexo ao diagrama E-R.

3.2.3 Relacionamentos ternários

Relacionam três ocorrências de entidades. Só é interessante utilizar esse tipo de relacionamento quando realmente é obrigatório associar, ao mesmo tempo, um par de entidades com uma terceira. Por exemplo, um empregado que trabalha num projeto necessariamente realiza alguma tarefa nesse trabalho. Esses três fatos estão sempre relacionados. Quando não ocorre essa obrigatoriedade, recomenda-se o uso da agregação.

A determinação da cardinalidade de um relacionamento ternário é feita questionando um par em relação à terceira entidade envolvida. Por exemplo, no caso de um empregado trabalhando em um projeto, temos o par empregado-projeto que realiza de 1 a N tarefas. Logo, a cardinalidade (1,N) é colocada ao lado da entidade TAREFAS.

Outros relacionamentos acima de ternários podem ocorrer em um diagrama E-R, porém são raros e deve-se avaliar cuidadosamente se serão necessários. A determinação da cardinalidade é semelhante ao comentado para ternários, ou seja, questiona-se um conjunto de entidades associadas em relação àquela cuja cardinalidade se deseja determinar.

3.3 Atributos do DER

Representam uma propriedade de uma entidade ou de um relacionamento, como salário de um empregado ou o tempo que um empregado estará alocado em um projeto.

São representados por círculos nomeados, ligados por linhas a uma entidade ou relacionamento. Atributos identificadores de entidades, ou que sejam necessários na identificação de relacionamentos, são indicados por um círculo preenchido ou pelo nome sublinhado.

3.3.1 Atributos opcionais

Atributos com propriedades que podem assumir NULL. São indicados por um traço que corta a linha que liga o atributo à entidade ou relacionamento.

3.3.2 Atributos compostos

Representam uma abstração de outros atributos, como um endereço que abstrai (agrega) outros dados como rua, CEP, cidade etc. É representado por uma elipse, com ligações (linhas) para os atributos componentes (convencionais).

3.3.3 Atributos multivalorados

Atributos com propriedades que podem assumir mais de um valor, como os números de telefone de um departamento. São representados por uma cardinalidade mínima e máxima que é colocada ao lado do nome do atributo, indicando as quantidades de valores mínimos e máximos permitidos.

3.4 Entidades fracas

Entidades cujas existências dependem da existência de outra entidade, dita *fork*. É representada por um retângulo duplo, podendo, opcionalmente, ser indicada uma seta que parte dela e chega à sua entidade forte, para tornar mais clara a dependência.

A cardinalidade de uma ocorrência de uma entidade fraca com a forte é sempre (1,1), indicando que ela sempre depende de uma ocorrência da entidade forte.

3.5 Especialização

Salienta que algumas ocorrências de uma entidade assumem papéis específicos, ou seja, são especializações, como o fato de um empregado ser um motorista ou secretário, inclusive com a possibilidade de terem atributos específicos. São classificadas em dois tipos, descritos a seguir.

3.5.1 Especialização com ou sem exclusão mútua

Indica se uma ocorrência de uma entidade genérica pode assumir apenas uma ou várias especializações, respectivamente.

No primeiro caso, representa-se por meio de um único triângulo que recebe uma linha da entidade genérica, sendo que da sua base partem linhas para várias entidades especializadas, dando a ideia de uma escolha que deve ser feita.

No segundo caso, desenha-se um triângulo para cada ligação entre a entidade genérica e uma especialização.

3.5.2 Especialização com ou sem obrigatoriedade

Indica se uma especialização é obrigatória ou não. Em caso negativo, diz-se que a especialização é parcial. A parcialidade é indicada pelo uso da letra P colocada ao lado do triângulo.

3.6 Agregação

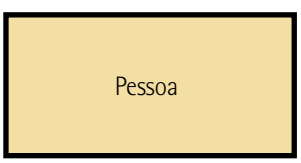
Um agregado representa uma porção de um diagrama E-R que envolve um relacionamento entre várias entidades. É utilizado para indicar situações em que entidades relacionadas podem ainda se relacionar com outras entidades. Essas entidades relacionadas são consideradas uma entidade de alto nível, chamada agregado, e são recomendadas para representar situações em que existe parcialidade na relação entre três ou mais ocorrências de entidade. Por exemplo, uma pessoa que tem a posse de um automóvel e que pode ou não ter contratado seguro para o carro. Um agregado possível é representado pelo conjunto de pessoas que tem automóvel com seguro.

Uma agregação é representada por um polígono fechado que envolve as entidades e o relacionamento entre elas, determinando a entidade agregada. Essa entidade agregada é considerada, então, uma entidade normal do diagrama E-R, que pode se relacionar com outras entidades.

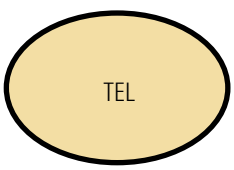
4 MODELO ENTIDADE RELACIONAMENTO (MER)

É um modelo abstrato desenvolvido pelo professor Peter Chen, a fim de representar as estruturas de dados de forma mais natural e mais próxima do mundo real dos negócios. Seus aspectos estruturais formalizam matematicamente a maneira como os dados estão organizados no modelo relacional (a seguir, serão descritos os conceitos básicos de um modelo relacional).

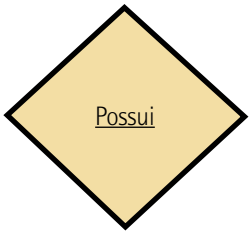
É composto de entidades, que são caracterizadas por atributos e que se relacionam entre si. Dentro do MER, utilizamos algumas formas geométricas para a representação das entidades, seus atributos e seus relacionamentos, conforme apresentados a seguir.



Retângulos representam as entidades



Elipses representam os atributos de uma entidade



Losangos representam as relações entre as entidades

Figura 9

4.1 Domínio

O domínio define o universo de valores permitidos para certo item de dado. Por exemplo, o domínio para o item de dado salário pode ser o conjunto dos números reais; o domínio do item de dado idade pode ser o conjunto dos números inteiros no intervalo [0,150]; o domínio do item sexo pode ser o conjunto limitado (masculino, feminino).

Um domínio compreende um tipo de dado predefinido mais um conjunto de restrições de integridade que limitam os valores permitidos para esse tipo de dado.

Domínios podem ser simples, ou seja, possuem um único valor, como um inteiro, ou compostos, quando possuem vários valores, como uma data. Os valores dos itens de dados que apresentam domínios compostos são abstraídos e tratados como um único valor. Uma data poderia ser manipulada como uma *string*.

Observação

A noção de domínio de um item de dados assemelha-se à noção de domínio de um conjunto na matemática.

Lembrete

MER é o modelo teórico inventado para transportar estruturas existentes no mundo real para o banco de dados. DER é a representação gráfica do MER.

4.2 Atributo do MER

Um atributo é um nome dado a um domínio utilizado para representar um item de dado. Matematicamente falando, um atributo é o nome de um conjunto.

Em uma tabela, representa uma coluna. Um atributo pode apresentar um valor condizente com o domínio associado a ele ou NULL, que indica a ausência de valor ou valor desconhecido.

Um atributo pode conter apenas um valor atômico, ou seja, um valor indivisível.

4.3 Tupla

É um conjunto de pares (atributo – valor) que define uma linha da tabela, ou seja, uma ocorrência de uma entidade ou relacionamento. Se imaginarmos uma tabela como um conjunto, uma tupla seria um elemento desse conjunto.

4.4 Relação

Pode ser definida como um subconjunto do produto cartesiano dos domínios (d1, d2, d3,..., dn) que correspondem aos atributos (a1, a2, a3,...,an) de uma tabela. O produto cartesiano gera (d1 x d2 x d3 x...x dn) tuplas e a relação mantém apenas aquelas tuplas que contêm valores de atributos relacionados que estão presentes na realidade.

Uma relação, portanto, é um subconjunto. É vista como uma tabela no modelo relacional, composta de um cabeçalho e um corpo. O cabeçalho é formado por um conjunto fixo de atributos que possuem nomes distintos, para evitar ambiguidade na localização de um item de dado. O grau de uma relação significa seu número de atributos.

O corpo de uma relação é formado por um número variável de tuplas, cuja ordem não é significativa, ou seja, dada uma relação R1 com três tuplas, nesta ordem: t1, t2 e t3, e uma relação R2 com três tuplas nesta ordem: t2, t3 e t1, teremos R1 igual a R2. A cardinalidade de uma relação significa o número de tuplas da relação.

O conceito de relação é ligeiramente diferente do conceito de tabela. Relação equivale à noção de conjunto, ou seja, um agrupamento de elementos sem repetição. Tabela equivale à noção de coleção, ou seja, um agrupamento de elementos em que é permitida a repetição. Os Sistemas Gerenciadores de Banco de Dados, na prática, lidam com o conceito de tabela.

4.5 Chave

O conceito de chave é fundamental no modelo para garantir a identificação de tuplas e estabelecer relacionamentos entre relações.

4.5.1 Chave primária (PK) – *primary key*

Atributo ou grupo de atributos que permite a identificação única de uma tupla em uma relação, ou seja, o valor da PK (*primary key*) de uma tupla nunca se repetirá nas demais tuplas da mesma relação.

Uma PK é escolhida entre várias chaves candidatas, que podem estar presentes na relação. As chaves candidatas não selecionadas são ditas chaves alternativas. A cada chave alternativa deve ser associada uma Relação de Integridade (RI) que garanta que seus valores sejam únicos.

Observação

Quando a chave primária tem um atributo, dizemos que é uma chave primária simples, e, quando ela tem dois ou mais atributos, dizemos que é uma chave primária composta.

Utilizamos uma chave primária simples para armazenar o ESTADO (UF) porque eles são únicos.

UF	
PK	SIGLA UF
	NOME_UF

Figura 10 – Exemplo de chave primária simples

Utilizamos uma chave composta para CIDADE porque pode-se ter mais de uma cidade com o mesmo nome, por isso temos que associar o nome da cidade com o estado ao qual ela pertence, para poder garantir a unicidade da tupla.

Cidade	
PK	UF CIDADE_NOME

Figura 11 – Exemplo de chave primária composta

4.5.2 Chave estrangeira (FK) – *foreign key*

Atributo ou grupo de atributos de uma relação R1 que estabelece um relacionamento de equivalência, por valor, com uma PK (*primary key*) de uma relação R2.

O domínio da FK (*foreign key*) de R1 deve ser compatível com o domínio da PK (*primary key*) de R2. Nada impede que R1 e R2 sejam a mesma relação.

4.5.3 Chave artificial ou delegada (SK) – *surrogate key*

Serve para substituir a chave original da tabela. Normalmente composta de números sequenciais, é muito utilizada em Modelos Multidimensionais, em que o processo de dimensionamento associa a chave natural do modelo relacional, que vão ser apenas atributos da dimensão, e para garantir a unicidade é criada a chave delegada (*surrogate key*).

Saiba mais

Para aprofundar os estudos sobre o Modelo Multidimensional, recomendamos a leitura dos livros de Ralph Kimball. E não deixe de visitar frequentemente este *site* indicado a seguir, se quiser entrar na área de Inteligência Corporativa e aprender mais sobre a tecnologia de Datawarehouse.

www.kimballgroup.com

4.6 Restrições de integridades básicas

Os aspectos de integridade do modelo relacional estão associados aos conceitos de chave primária e chave estrangeira. Garantir a integridade de um esquema relacional significa assegurar o acesso individualizado a todas as tuplas de uma relação, assim como a relacionamentos válidos e condizentes com a realidade.

O modelo relacional é regido por duas regras de integridade básicas: a de integridade de entidade e a de entidade referencial, descritas a seguir.

4.6.1 Regra de integridade de entidade

Diz respeito à chave primária de uma relação. Ela diz que nenhum atributo que faz parte da chave primária pode ter NULL em alguma tupla. A manutenção dessa regra garante que toda tupla possa ser identificada unicamente.

4.6.2 Regra de integridade referencial

Diz respeito às chaves estrangeiras de uma relação. Ela diz que o valor de um atributo que faz parte de uma chave estrangeira pode assumir NULL desde que não faça parte da chave primária. Ainda, esse mesmo atributo pode assumir um valor qualquer, condizente com o seu domínio, desde que esse mesmo valor exista na chave primária de uma tupla da relação referida por ele. Com isso, nunca existirão relacionamentos incorretos entre dados.

Para que essas regras sejam sempre respeitadas, o SGBD deve implementar rotinas de verificações automáticas. Isso implica testes e ações a serem realizados pelo SGBD toda vez que uma operação de atualização ocorrer.

No caso da regra de integridade de entidade, o seguinte algoritmo deve ser executado toda vez que for incluída uma nova tupla em uma relação ou for alterado um valor de um atributo de uma tupla que faz parte da chave primária de uma relação:

SE a chave primária da tupla for NULL

OU existir tupla com o mesmo valor da chave primária

ENTÃO impedir a efetivação da operação

SENÃO efetivar a operação

Já para o caso da regra de integridade referencial, três alternativas de ações são geralmente tomadas pelo SGBD quando se percebe que uma violação irá ocorrer:

- **Impedimento** → a operação não é efetivada.
- **Cascata** → para o caso de exclusões de tuplas ou alterações de chave primária na relação referida, realizar a mesma operação em todas as tuplas que se referem a ela.
- **Anulação** → para o caso de exclusões de tuplas ou alterações de chave primária na relação referida, altera-se para NULL o(s) atributo(s) que compõe(m) a chave estrangeira que estabelece o relacionamento com essa relação. Uma variante dessa alternativa é anular apenas a chave estrangeira de uma única tupla.

A aplicação dessas ações depende da operação de atualização submetida e da relação que sofre a operação (a referida ou a que possui uma referência). Se uma operação de atualização é submetida à relação que possui a referência (possui a chave estrangeira), o seguinte algoritmo deve ser executado toda vez que for incluída uma nova tupla ou for alterado o valor de um atributo que faz parte da chave estrangeira:

SE a chave estrangeira da tupla for NULL

ENTÃO

SE a chave estrangeira fizer parte da chave primária

ENTÃO impedir a efetivação da operação

SENÃO efetivar a operação

SENÃO SE não existir uma tupla na relação referida com valor de chave primária igual ao valor da chave estrangeira desta tupla

ENTÃO SE a chave estrangeira fizer parte da chave primária

ENTÃO aplicar impedimento

SENÃO aplicar impedimento

OU aplicar anulação

SENÃO efetivar a operação

Se uma operação de atualização é submetida à relação referida (possui a chave primária), o seguinte algoritmo deve ser executado toda vez que for excluída uma tupla ou for alterado o valor de um atributo que faz parte da chave primária:

SE a chave estrangeira fizer parte da chave primária

ENTÃO aplicar impedimento

OU aplicar cascata

SENÃO aplicar impedimento

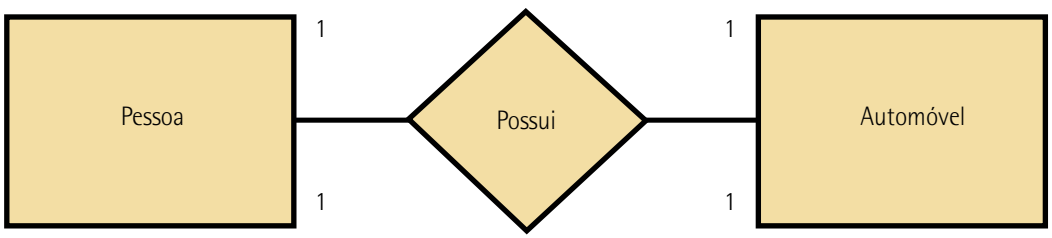
OU aplicar cascata

OU aplicar anulação

Lembrete

Entidades, do original *entity type*, servem para representar objetos, coisas, pessoas que existem na realidade. No MER, os relacionamentos são identificados com um verbo, conforme pode ser observado na figura 12.

Uma pessoa possui um automóvel.

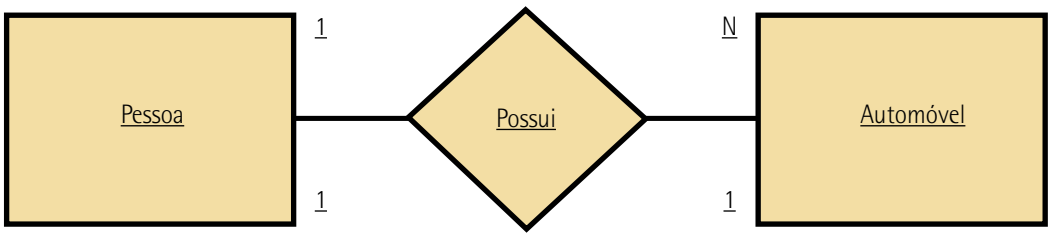


Um automóvel pertence a uma pessoa.

Figura 12 – Exemplo de um relacionamento 1:1

Dentro de um modelo relacional, temos que ver como as entidades se relacionam entre si, ou seja, ver como a entidade Pessoa se relaciona com a entidade Automóvel e como a entidade Automóvel se relaciona com a entidade Pessoa, e podemos concluir que:

Uma pessoa possui vários automóveis.



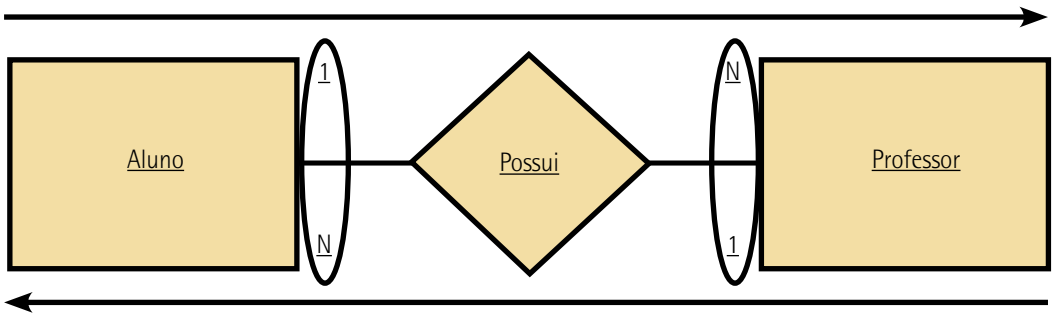
Vários automóveis pertencem a uma pessoa.

Figura 13 – Exemplo de relacionamento 1:N

Aqui fazemos os mesmos passos para a obtenção da cardinalidade, verificamos o relacionamento de Pessoa com Automóvel e de Automóvel com Pessoa.

- 1 Pessoa possui N (vários) automóveis.
- 1 Automóvel que pertence a 1 Pessoa.

Um aluno possui vários professores.



Um professor possui vários alunos.

Figura 14 – Exemplo de relacionamento N:N

No exemplo da Figura 14, temos a seguinte definição: 1 aluno possui N (vários) professores e 1 professor possui N (vários) alunos.

Para obter a cardinalidade final, devemos usar a multiplicação. Em cada um dos lados, pegamos os dois números, que estão em destaque ao lado da entidade. Nesse caso, em aluno temos 1 e N, 1 x N = N (qualquer número multiplicado por 1 é ele mesmo). Já em professor, temos N e 1. N x 1 = N. Assim, obtemos um relacionamento de N para N.

4.7 Tipos de relacionamento

Quando fazemos o relacionamento entre duas ou mais tabelas, esse relacionamento pode ser classificado de duas formas:

- **Identificado**: também conhecido como relacionamento forte. Identifica que a chave estrangeira da tabela "pai" faz parte da chave da tabela "filha".

- **Não identificado**: também conhecido como relacionamento fraco. Identifica que a chave estrangeira da tabela "pai" não faz parte da chave primária da tabela filha, sendo esse apenas mais um atributo.

A seguir, mostramos a representação desses tipos de relacionamento, com exemplos, nas figuras 15 e 16.

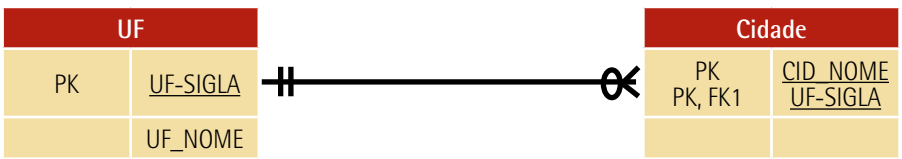


Figura 15 – Exemplo de relacionamento forte

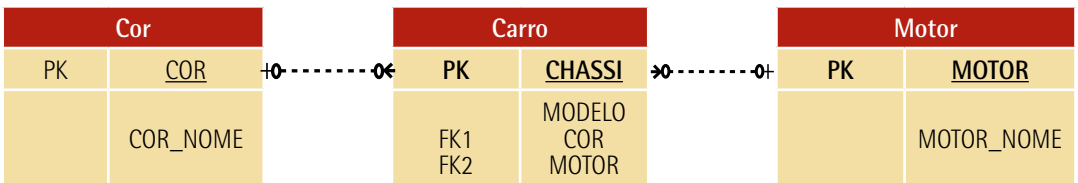


Figura 16 – Exemplo de relacionamento fraco

4.8 Normalização

Normalização de dados é o processo formal passo a passo que examina os atributos de uma entidade, com o objetivo de evitar redundâncias e anomalias observadas na inclusão, na exclusão e na alteração de registros.

4.8.1 Primeira Forma Normal (1FN)

Uma relação estará na Primeira Forma Normal 1FN se, e somente se, todos os domínios básicos contiverem somente valores atômicos (não contiver grupos repetitivos).

Procedimentos:

- identificar a chave primária da entidade;
- identificar o grupo repetitivo e removê-lo da entidade;
- criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo;
- a chave primária da nova entidade será obtida pela concatenação da chave primária da entidade inicial e a do grupo repetitivo.

Exemplo:

ID	Nome	Telefone	Endereço
1	João	(11)1234-5678 (11)2345-6789	Rua Seis, 85 Morumbi 12536-965
2	Maria	(11)3456-7890 (11)4567-8901	Rua Onze, 64 Moema 65985-963
3	José	(11)5678-9012 (11)6789-0123	Praça Ramos Liberdade 68858-456

Figura 17

Aqui temos os mesmos grupos de dados agrupados para todas as linhas.

ID	Nome	Telefone	Rua	Bairro	CEP
1	João	(11)1234-5678 (11)2345-6789	Rua Seis, 85	Morumbi	12536-965
2	Maria	(11)3456-7890 (11)4567-8901	Rua Onze, 64	Moema	65985-963
3	José	(11)5678-9012 (11)6789-0123 (11)7890-1234	Praça Ramos	Liberdade	68858-456

Figura 18

Foi resolvido o problema do endereço, quebrando a informação em três colunas; porém, ainda persiste o problema da coluna telefone e, diferentemente do que se dá no caso anterior, não tem a mesma repetição do outro grupo.

Para resolvermos esse caso, teremos que quebrar essa tabela em duas.

ID	Nome	Rua	Bairro	CEP
1	João	Rua Seis, 85	Morumbi	12536-965
2	Maria	Rua Onze, 64	Moema	65985-963
3	José	Praça Ramos	Liberdade	68858-456

Figura 19

A primeira, sem a informação de telefone.

Telefone	ID
(11)1234-5678	1
(11)2345-6789	1
(11)3456-7890	2
(11)4567-8901	2
(11)5678-9012	3

Telefone	ID
(11)6789-0123	3
(11)7890-1234	3

Figura 20

A segunda, apenas com a informação de telefone, associada com a primeira pelo código do cliente.

4.8.2 Segunda Forma Normal (2FN)

Uma relação estará na Segunda Forma Normal 2FN se já estiver na 1FN e não existir nenhum atributo que não seja dependente de todo da chave da tabela.

Procedimentos:

- identificar os atributos que não são funcionalmente dependentes de toda a chave primária;
- remover da entidade todos esses atributos identificados e criar uma nova entidade com eles.

Exemplo:

	C_Produto	N_Produto	Qtd	Vl_Unitário	Subtotal
1	1234	Impressora Matricial	4	100,00	400,00
2	2345	Impressora Jato de Tinta	3	200,00	600,00
3	3456	Impressora Laser	2	1000,00	2000,00
4	5678	Impressora Multifuncional	1	350,00	350,00

Figura 21

Não está na Segunda Forma Normal porque temos atributos que não dependem da chave primária.

N_Pedido	C_Produto	Qtd	Vl_Unitário	Subtotal
1	1234	4	100,00	400,00
2	2345	3	200,00	600,00
3	3456	2	1000,00	2000,00
4	5678	1	350,00	350,00

Figura 22

Para resolver, separamos a informação em duas tabelas.

C_Produto	N_Produto
1234	Impressora Matricial
2345	Impressora Jato de Tinta
3456	Impressora *Laser*
5678	Impressora Multifuncional

Figura 23

Da tabela de pedidos, foi retirada a descrição do produto, porque o atributo é dependente apenas do código do produto. Foi criada uma nova tabela chamada de **Produto**, composta de **Código do Produto** e **Nome do Produto**. Foi mantido o **Código do Produto** na tabela de **Pedido** para o relacionamento.

4.8.3 Terceira Forma Normal (3FN)

Uma tabela está na Terceira Forma Normal 3FN se estiver na 2FN e se nenhuma coluna não chave depender de outra coluna não chave.

Procedimentos:

- identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave e removê-los;
- a chave primária da nova entidade será o atributo do qual os atributos removidos são funcionalmente dependentes.

N_Pedido	C_Produto	Qtd	Vl_Unitário	Subtotal
1	1234	4	100,00	400,00
2	2345	3	200,00	600,00
3	3456	2	1000,00	2000,00
4	5678	1	350,00	350,00

Figura 24

Ainda existe um atributo que está relacionado a outro atributo não chave.

O atributo **SUBTOTAL** é derivado do cálculo da quantidade que multiplica o valor unitário.

N_Pedido	C_Produto	Qtd	Vl_Unitário
1	1234	4	100,00
2	2345	3	200,00
3	3456	2	1000,00
4	5678	1	350,00

Figura 25

Nesse caso, removemos a coluna, já que, entre outras coisas, ela era derivada de um cálculo.

Saiba mais

Na internet, existe muita informação sobre o fascinante mundo do DBA. Para começar a navegar nesse mundo, sugerimos alguns *sites*. Boa viagem!

http://certificacaobd.com.br

http://silasmendes.com/dba

http://www.mcdbabrasil.com.br

http://www.sybase.com/resources/*blogs*

http://dbaforums.org/oracle

Resumo

Nesta unidade, apresentamos o modelo relacional, que foi formalmente definido por E. Codd, no laboratório da IBM em San José, na Califórnia, em 1970. O projeto inicial foi denominado de **Sistema R** e definia a organização dos dados e linguagens formais para a sua manipulação. Com base nessas linguagens formais, a primeira versão da linguagem SQL foi definida. Essa linguagem é, atualmente, um padrão para gerenciamento de dados em um SGBD relacional.

Entidades e relacionamentos são representados nesse modelo por relações que equivalem ao conceito matemático de conjunto, ou seja, um agrupamento de elementos sem repetição.

Uma relação é vista como uma tabela, em que as colunas indicam os campos e as linhas, as ocorrências (valores). Relacionamentos entre relações são estabelecidos por igualdade de valor de campos. Relacionamentos M:N são modelados como relações que mantêm os campos que identificam as duas relações envolvidas mais os dados pertinentes ao relacionamento, caso existam.

Uma das vantagens em adotar o modelo relacional é a representação uniforme das entidades e dos relacionamentos. É um conceito único e simples para a organização de dados. Um esquema no modelo relacional é dito um esquema do mais alto nível, pois abstrai uma estrutura de

implementação, como estruturas em árvore ou grafos, em que o usuário deve se preocupar em percorrer apontadores adequadamente.

O modelo relacional é o primeiro modelo de dados que oferece linguagens de manipulação de dados cujos comandos podem ser executados independentemente da aplicação, pois não estão obrigatoriamente vinculados ao código da aplicação. São, ainda, linguagens de alto nível, ou seja, um pequeno comando de manipulação equivale a um procedimento de acesso a dados, se comparado com os modelos anteriores. Linguagens com essa característica são ditas declarativas, ou seja, o usuário preocupa-se em dizer o que eles desejam obter do banco de dados, e não como ele deseja obter.

A normalização é uma metodologia para projeto de banco de dados relacionais, uma vez que sugere uma organização de dados em tabelas. Assim sendo, a normalização é uma ferramenta para projeto lógico de banco de dados relacionais. O objetivo dessa técnica é evitar problemas de redundância e anomalias de atualização, que podem estar presentes em uma relação. A solução para resolver esses problemas é a decomposição de uma relação em uma ou mais relações, com base na aplicação de certas regras de normalização (formas normais). Essa proliferação de relações nem sempre é ideal do ponto de vista de *performance*, devendo ser balanceadas vantagens e desvantagens antes da efetivação dos resultados de uma forma normal.

Exercícios

Questão 1. Um banco de dados é um componente fundamental para o êxito dos sistemas de informação (SI) na atualidade. Todavia, para ter um banco de dados consistente, torna-se necessário o projeto do banco de dados, que é uma atividade essencial na fase de implementação de um SI. Projetar bancos de dados tem se tornado uma atividade importante e que exige profissionais especializados, principalmente quando os sistemas manipulam grandes quantidades de dados. Frequentemente, quando não há uma abordagem adequada para o projeto de um banco de dados, pode-se incorrer em resultados indesejáveis e até em grandes prejuízos financeiros para as empresas. Os autores indicam que a ausência de um projeto do banco de dados acarreta uma falta de clareza para entender a natureza exata dos dados em um nível conceitual (abstrato). As atividades do projeto de um banco de dados incluem: o projeto conceitual, o projeto lógico e o projeto físico.

Considerando os conceitos sobre Banco de Dados e Projetos de Banco de Dados, examine as afirmações a seguir e indique a alternativa **incorreta**:

A) O projeto conceitual de um banco de dados é uma atividade do projeto de banco de dados que usa como base a especificação dos requisitos de um projeto de SI, produzindo como resultado o esquema conceitual do banco de dados.

B) Um esquema conceitual é uma descrição em alto nível da estrutura do banco de dados, independente do Sistema de Gerenciamento de Banco de Dados (SGBD) adotado para implementá-lo.

C) O propósito do projeto conceitual de um banco de dados é descrever o conteúdo de informação do banco de dados em vez das estruturas de armazenamento que serão necessárias para gerenciar essa informação.

D) O projeto lógico de um banco de dados tem por objetivo avaliar o esquema conceitual frente às necessidades de uso do banco de dados pelos usuários e/ou aplicações, realizando possíveis refinamentos para alcançar maior desempenho das operações sobre o banco de dados.

E) O projeto físico de um banco de dados não toma como base o projeto lógico para construir o esquema físico do banco de dados.

Resposta correta: alternativa E.

Análise das alternativas

De acordo com a teoria de projeto de banco de dados, as atividades de projeto de BD envolvem, em ordem sequencial, o projeto conceitual, o projeto lógico e o projeto físico do banco de dados. Dessa forma, o projeto físico utiliza o esquema lógico gerado durante o projeto lógico para construção das estruturas físicas do banco de dados.

A) Alternativa correta.

Justificativa: projeto conceitual de dados é uma atividade que tem como ponto de partida os requisitos de informação e as regras de negócio inerentes a um determinado problema ou necessidades de automação de uma área de negócio.

B) Alternativa correta.

Justificativa: o esquema conceitual de dados é constituído de: a) Diagrama de Entidade e Relacionamento (DER), que modela o aspecto estrutural do mundo real; b) Regras de Restrição de Integridade (RI) e c) um conjunto de Regras de Derivação (RD) que modelam o aspecto comportamental do mundo real.

C) Alternativa correta.

Justificativa: durante o projeto conceitual é gerado o esquema conceitual do banco de dados e através dele se descrevem as características dos dados ou informações que estarão contidos no banco de dados.

D) Alternativa correta.

Justificativa: a atividade do projeto lógico de banco de dados tem como objetivo transformar o modelo conceitual obtido durante o projeto conceitual de dados em um modelo lógico. O modelo lógico

define como o banco de dados será implementado em um SGBD específico, tal como o Oracle, o DB2, o MySQL etc.

E) Alternativa incorreta.

Justificativa: o projeto físico do banco de dados toma por base o esquema lógico para construir o esquema físico. Um esquema físico é uma descrição da implementação do banco de dados em memória disco magnético ou memória secundária; ele descreve as estruturas de armazenamento e métodos de acesso usados para efetivamente realizar o acesso aos dados. O projeto físico é direcionado para um SGBD específico (por exemplo: Oracle, Sybase, OpenIngres, Access).

Questão 2. O modelo de banco de dados relacional apresenta o banco de dados como uma coleção de tabelas. O conceito de tabela, embora seja simples e intuitivo, apresenta uma forte correspondência com o conceito matemático de uma relação. Dessa forma, um banco de dados relacional consiste em uma coleção de relações (tabelas), cada qual associada a um nome único (identificação da relação ou tabela). Para a manipulação das tabelas e dos Bancos de Dados Relacionais utiliza-se a linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada), que é uma linguagem usada para a consulta, atualização, criação e gerenciamento de banco de dados relacionais. É um método para selecionar determinados registros de um banco de dados, seguindo um critério especificado pelo usuário ou por uma aplicação. A SQL é considerada uma linguagem padrão para o gerenciamento de banco de dados relacionais.

Considerando os conceitos sobre Banco de Dados Relacionais e linguagens de manipulação de BD, examine as afirmações a seguir e indique a alternativa **incorreta**:

A) A linguagem SQL é considerada a linguagem padrão ANSI (*American National Standards Institute* - Instituto Nacional de Padronização Americano) para a operação em bancos de dados relacionais.

B) A linguagem SQL realmente tornou-se um padrão na indústria de Banco de Dados relacionais e os fornecedores (empresas desenvolvedoras de BD relacionais) garantem esse padrão ao longo do tempo.

C) Pode-se afirmar que SQL não é a mesma coisa que SQL Server, pois SQL é a denominação universal para a linguagem de manipulação de BD relacional e SQL Server é o nome de um SGBD específico de um determinado fornecedor de Banco de Dados.

D) A linguagem SQL possui dois tipos de comandos fundamentais para que o usuário ou uma determinada aplicação faça uso do BD relacional, os comandos DDL e os comandos chamados de DML.

E) Na linguagem SQL, o comando SELECT é o comando mais utilizado, pois é através dele que o BD retorna os dados de uma ou mais tabelas. Como ele não gera nenhuma modificação nos dados, é utilizado para a busca de informações de uma tabela ou de um conjunto de tabelas de acordo com as necessidades do usuário ou de uma funcionalidade de uma aplicação.

Resposta correta: alternativa B.

Análise das alternativas

A) Afirmativa correta.

Justificativa: a linguagem SQL foi criada para atender a todos os bancos de dados relacionais e permitir que usuários possam acessar qualquer banco usando a mesma base de conhecimento.

B) Afirmativa incorreta.

Justificativa: atualmente, existem diversas diferenças na implementação específica da linguagem SQL nos bancos de dados de fornecedores diferentes. Todavia, a estrutura básica da SQL foi mantida, permitindo a execução dos comandos SQL nas diversas soluções do mercado com poucas modificações.

C) Afirmativa correta.

Justificativa: SQL é uma linguagem universal para trabalhar com banco de dados relacional, e SQL Server é um *software* de banco de dados da empresa americana Microsoft.

D) Afirmativa correta.

Justificativa: a linguagem SQL possui diversos conjuntos de comandos, que permitem a criação do banco de dados, sua manipulação e sua gestão. Os dois comandos principais são: os comandos DDL (*Data Definition Language*) incluem o *create*, o *alter* e o *drop*. E os comandos DML (*Data Manipulation Language*) incluem o *select*, o *insert*, o *update* e o *delete*.

E) Afirmativa correta

Justificativa: para fazer uma modificação em uma tabela do banco de dados relacional, utiliza-se o comando *select*, e, após os dados serem alterados na memória do computador, torna-se necessário o uso do comando *update* para efetuar a modificação no banco de dados. Isso evita o risco de realizar alterações indevidas.