2

Introdução ao Modelo de Entidade e Relacionamento

O modelo de Entidade e Relacionamento (ER) é um modelo conceitual e deve estar o mais próximo possível da visão que o usuário tem dos dados, não se preocupando em representar como estes dados estarão realmente armazenados.

Este modelo tem por objetivo descrever quais dados devem ser armazenados pela aplicação e quais desses dados se relacionam.

Suponha que uma escola precise armazenar informações sobre seus alunos, professores, e disciplinas. O modelo da base de dados deve informar quais dados sobre alunos, professores e disciplinas são importantes para serem armazenados. Assim, sobre os alunos, pode-se armazenar informações como nome, data de nascimento e matrícula. Sobre os professores, armazenam-se informações como nome, telefone e matrícula. A respeito das disciplinas, pode-se armazenar informações como código da disciplina e o nome da disciplina. Além dessas informações, é necessário saber qual professor é responsável por qual disciplina e quais alunos fazem a disciplina. O modelo para esse pequeno exemplo ficaria como mostra a figura 2.1.

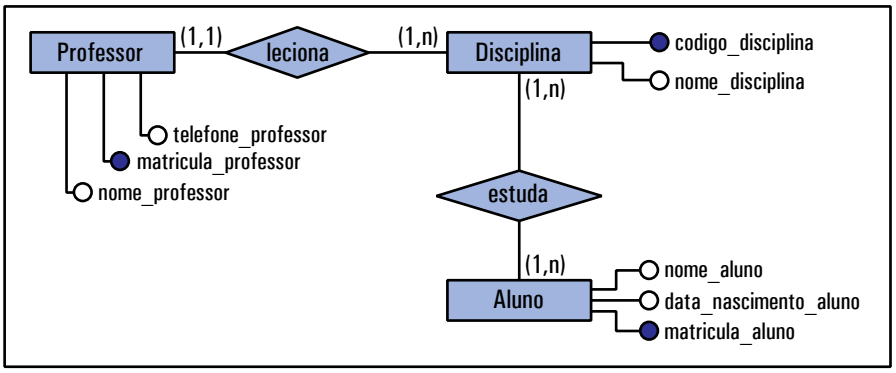


Figura 2.1 – Exemplo de modelo conceitual

Note que não estamos preocupados aqui em saber como os dados serão armazenados ou como eles devem ser implementados. A nossa preocupação é conseguir entender o que precisa ser armazenado e quais informações devem se relacionar.

A ferramenta utilizada para criação deste diagrama foi a **brModelo**. Esta é uma ferramenta para modelagem de base de dados gratuita e foi desenvolvida por Carlos Henrique Cândido, sob a orientação do Prof. Dr. Ronaldo dos Santos Mello.

Observe que o modelo da figura 2.1 informa que a base de dados contém informações sobre professores, alunos e disciplinas, mas não se preocupa em descrever o valor que esses dados armazenam.

O Modelo de Entidade e Relacionamento, apresentado na figura 2.1, utiliza uma representação gráfica chamada de Diagrama de Entidade e Relacionamento (DER).

É importante que se compreenda que o Modelo de Entidade e Relacionamento define os conceitos que serão aplicados no desenvolvimento de um Diagrama de Entidade e Relacionamento (DER). Assim, o DER será utilizado para representar graficamente o conjunto de objetos do Modelo de Entidade e Relacionamento, tais como entidades, atributos, atributos-chave, relacionamentos, restrições estruturais, etc.

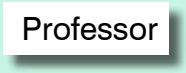
Um modelo de Entidade e Relacionamento consiste em um conjunto de objetos básicos chamados **entidades** e de **relacionamentos** entre as entidades.

Entidades e Atributos

Você desenvolve um Sistema de Informação para resolver um problema do mundo real. Assim, toda a informação referente ao problema que se quer solucionar representa parte deste mundo.

Uma **entidade** representa um conjunto de objetos do mesmo tipo do mundo real e sobre os quais se pretende armazenar dados. Por exemplo, ao desenvolver um Sistema de Informação para uma escola, as possíveis entidades desse sistema serão: professores, alunos, disciplinas, turmas e cursos.

Uma **entidade** é representada graficamente por um retângulo com o nome da entidade dentro do retângulo. Por exemplo:



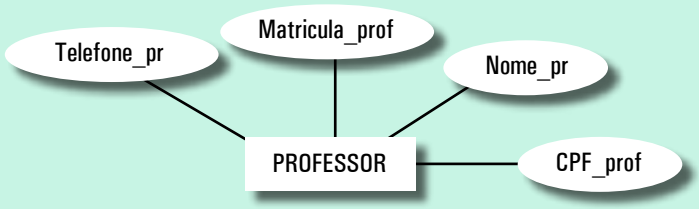
Cada uma dessas entidades armazenará um conjunto de objetos do mesmo tipo. Ou seja, ter uma entidade denominada professor significa que vários professores serão cadastrados nessa entidade e cada professor representa, portanto, um objeto da entidade.

Além de uma entidade representar objetos do mundo real, ela também deve possuir um conjunto de propriedades que a caracterize e a descreva, bem como aos seus objetos. A esse conjunto de propriedades dá-se o nome de **atributos**. Por exemplo, para a entidade Professor, é necessário armazenar dados como: CPF, nome, telefone, endereço, grau de escolaridade, número da matrícula, etc. Esses dados são os atributos da entidade Professor e são eles que identificam e caracterizam um objeto do tipo professor.

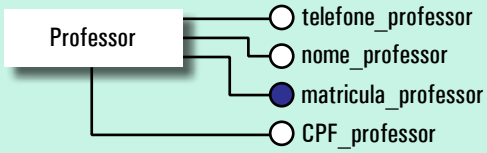
Outro exemplo, para uma entidade chamada Cadeira, os possíveis atributos dessa entidade serão: número de pernas, cor, tamanho, peso, altura, tecido, etc.

O nome dos atributos deve representar o que aquele atributo armazena.

Um atributo pode ser representado graficamente por uma elipse com o nome do atributo dentro da elipse. A elipse é ligada à entidade por uma linha, conforme exemplo:



Outra forma de representação utilizada por algumas ferramentas é representar o atributo como uma bolinha ligada à entidade e com o nome do atributo ao lado, conforme exemplo:



Uma entidade deve ter ao menos dois atributos. Uma entidade que possui apenas um atributo não é entidade e esse único atributo deveria estar em alguma outra entidade do modelo.

Todo atributo possui um tipo de dado que representa os valores permitidos para aquele atributo. A esse tipo de dados dá-se o nome de **domínio do atributo**. Por exemplo: o atributo "número de pernas" da entidade "Cadeira" é do tipo inteiro, ou seja, só permite que sejam armazenados valores inteiros para esse atributo.

Os tipos de dados dependem do SGBD que o desenvolvedor está utilizando. De forma geral, todos os SGBD disponibilizam tipos de dados como: *inteiro, caracter, real (ou float), data* e *hora*.

Quando se define o tipo de um atributo, pode-se definir inclusive o tamanho máximo que o atributo vai permitir armazenar. Por exemplo, o atributo "nome" é do tipo caracter (500), ou seja, armazena no máximo 500 caracteres.

Os atributos podem ainda ser divididos em 6 categorias: simples, compostos, monovalorado, multivalorado, derivado e nulo. É importante ressaltar que os atributos podem pertencer a mais de uma categoria ao mesmo tempo. Isso significa que é comum um único atributo ser simples, monovalorado e derivado ao mesmo tempo. A seguir, serão explicadas e exemplificadas cada uma das categorias.

- **Atributo simples:** é o atributo indivisível, que não pode ou não deve ser decomposto. Por exemplo: "CPF", "numero da matrícula", "RG", "preço do produto", etc.

- **Atributo composto:** é o atributo que pode ser decomposto em outros atributos simples. Por exemplo, o atributo "endereço" pode ser decomposto em "nome da rua", "número" e "complemento".
- **Atributo monovalorado:** é o atributo que permite apenas o armazenamento de um valor por vez. Por exemplo, o atributo "CPF" é monovalorado porque uma pessoa possui apenas um número de CPF. Caso o CPF seja alterado ele é substituído pelo novo valor. Assim, uma pessoa **nunca** terá cadastrado mais de um CPF no mesmo campo.

É interessante que atributos compostos sejam decompostos ainda no 1º modelo, que é o Modelo de Entidade e Relacionamento, uma vez que isso vai ter que ocorrer obrigatoriamente no modelo relacional.

Alguns atributos como, por exemplo, "nome do aluno" pode ser classificado como simples ou composto dependendo da aplicação. Se na aplicação forem realizadas consultas pelo sobrenome do aluno, é interessante que este atributo seja decomposto em dois atributos simples: "primeiro nome" e "sobrenome". Isso ocorre por questão de desempenho.

- **Atributo multivalorado:** é o atributo que permite armazenar mais de um valor ao mesmo tempo no mesmo campo. Por exemplo, o atributo e-mail pode ser multivalorado uma vez que uma pessoa possui, normalmente, mais de um endereço de e-mail.

O atributo multivalorado deve ser evitado sempre que possível. No entanto, em situações em que não é possível evitá-lo, ele deve ser representado no diagrama como multivalorado (ver como representar na figura 2.2). Quando formos passar o DER para o Modelo Relacional (capítulo 4), vamos entender o que acontece com esse atributo. Outro ponto importante: quem determina se o atributo é multivalorado ou não, muitas vezes, é o próprio usuário do sistema. No caso do exemplo do atributo "e-mail", o usuário pode determinar que somente é necessário armazenar um e-mail e sendo assim o atributo deixa de ser multivalorado e passa a ser monovalorado.

- **Atributo nulo:** é o atributo que permite que seja inserido um valor nulo para ele. Valor nulo representa a inexistência de um valor, ou seja, significa que o usuário não precisa cadastrar um valor para o atributo e pode deixá-lo vazio. Em algumas situações, é inevitável que permitamos **valores nulos** para os atributos. Vamos usar novamente o atributo "e-mail" como exemplo. Como nem todas as pessoas possuem e-mail, esse atributo deve permitir valores nulos, porque se ele não permitir algumas pessoas não poderão se cadastrar ou terão que criar um e-mail para poder efetivar o cadastro. Novamente é o usuário quem, muitas vezes, vai definir se um atributo é obrigatório ou não.

Valor nulo:
Valor nulo é diferente de valor zero!!! O valor nulo (representado por null em banco de dados) significa que aquele campo está vazio.

O valor nulo na base de dados pode levar o banco a ficar inconsistente e ter baixo desempenho. Mesmo que o atributo não seja obrigatório, é interessante que ele receba um valor padrão (*default)* via aplicação ou via SGBD para evitar os valores nulos.

- **Atributo derivado:** é o atributo cujo valor para ele deriva de outro(s) atributo(s). Por exemplo, suponha que a sua entidade se chame compra e que ela tenha os seguintes atributos: "número da compra", "data da compra", "valor da compra", "percentual de desconto" e "valor da compra com o desconto". O valor para este último atributo é calculado considerando-se o "valor da compra" e o "percentual de desconto". Assim, esse atributo é derivado porque seu valor deriva dos valores de outros atributos e é calculado automaticamente pela aplicação ou pelo SGBD.

A figura 2.2 apresenta formas de representação dos tipos dos atributos descritos anteriormente.

Representações de Atributos:

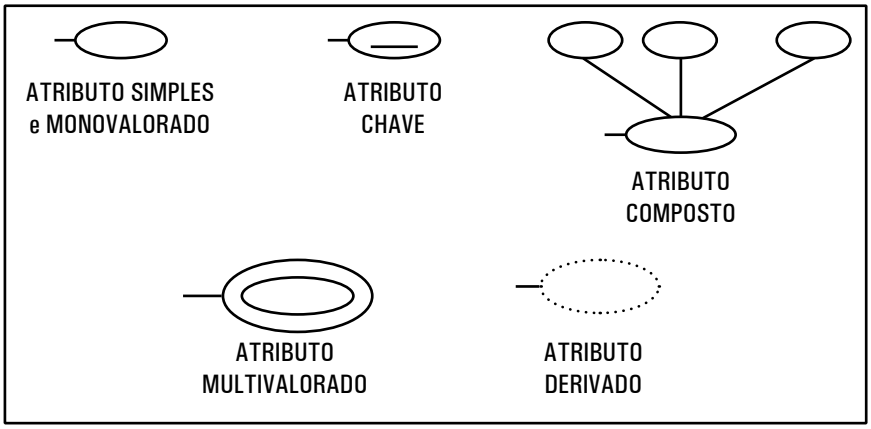


Figura 2.2 – Representação gráfica dos atributos

Chave Primária

Um conceito importante no Modelo de Entidade e Relacionamento é o conceito de chave primária (ou *Primary Key* ou ainda **PK**). Uma **chave primária** é um atributo da entidade que identifica apenas um objeto dessa entidade. Portanto, o valor dentro de uma chave primária não poderá se repetir e também não poderá receber um valor nulo.

Por exemplo, na entidade "Professor", tanto o atributo "CPF" quanto o atributo "matrícula" não se repetem, uma vez que esses atributos são únicos para cada indivíduo. Nesse caso, qualquer um dos dois atributos poderia ser definido como uma chave primária.

A pergunta é: Qual deles eu devo definir como uma chave primária? Bem, se mais de um atributo for único para a entidade, a escolha da chave primária vai depender do escopo do problema e de como serão realizadas as consultas. Considerando-se que a maioria dos SGBD vincula um índice à chave primária, é interessante que essa chave primária seja o atributo mais utilizado nas consultas. Para o exemplo da entidade "Professor", a maioria das consultas seriam feitas considerando-se a matrícula do professor na instituição e não o CPF. A opção por consultas que tenham como condição a matrícula do professor se deve por causa do escopo do problema, uma vez que a matrícula do professor é um atributo muito mais específico para a instituição de ensino. Sendo assim, o atributo matricula_professor seria a melhor opção para ser a chave primária.

Outro ponto importante a considerar durante a decisão de qual atributo deverá ser a chave primária é que se deve dar preferência a atributos numéricos (inteiros) em vez de atributos do tipo caractere, data ou hora. Como "CPF" é um atributo do tipo caractere, ele poderia ser descartado como chave primária por esse motivo também.

Tipos de Chave Primária

Uma chave primária pode ser simples ou composta. Uma chave primária simples é aquela que será formada por apenas um atributo. Por exemplo: matricula_professor.

Uma chave primária composta é formada por dois ou mais atributos. Por exemplo, imagine que tenhamos uma entidade chamada "Localização" e esta entidade tem os seguintes atributos: "nome da cidade", "nome do estado", "nome do país". Cada um desses atributos sozinhos não pode ser chave primária porque eles se repetem, como mostra a figura 2.3.

nome_cidade	nome_estado	nome_país
Curitiba	Paraná	Brasil
Maringá	Paraná	Brasil
Campo Grande	Mato Grosso do Sul	Brasil
Campo Grande	Rio de Janeiro	Brasil

Figura 2.3 – Exemplos de dados armazenados na entidade "Localização"

Como os atributos individualmente podem se repetir, vamos tentar encontrar uma chave primária composta. Sabemos que no Brasil um estado não tem duas cidades com o mesmo nome. Sendo assim, poderíamos criar uma chave primária composta do "nome da cidade" mais o "nome do estado", porque o valor para esses dois atributos juntos nunca vai se repetir.

Nesse caso, o atributo "nome da cidade" não é uma chave primária, e sim **faz parte** da chave primária.

É importante destacar que não existe mais de uma chave primária por entidade. Essa chave primária poderá ser simples ou composta. Mesmo composta, é uma **única** chave primária composta de mais de um atributo.

A figura 2.4, na página 24, identifica graficamente uma chave primária, pintando de azul o atributo que corresponde a chave primária. Outra forma de identificação de uma chave primária no diagrama é grifar o nome do atributo que é chave ou que compõe a chave, como mostra a figura 2.5.

Entidade Fraca

Entidade fraca é um tipo de entidade que não possui atributo chave primária por si só. Isso significa que não é possível definir uma chave primária, nem simples e nem composta, para a entidade.

Além disso, uma entidade fraca é dependente de uma outra entidade, e o relacionamento entre a entidade fraca e a outra entidade é normalmente **1:N**, e o N fica junto à entidade fraca.

1:N
Notação que indica cardinalidade. Lê-se 1 para N.

A entidade "Contato" da figura 2.4 é uma entidade fraca porque não possui um atributo (ou conjunto de atributos) que identifique um único objeto. Além disso, ela é dependente da entidade "Aluno", porque só existe um contato se houver o aluno para aquele contato. Finalmente, o relacionamento entre as duas entidades (Aluno e Contato) é 1:N, ou seja, um aluno pode ter vários contatos, mas um contato pertence a apenas um aluno, este conceito de 1:N será aprofundado no capítulo 3.

Na figura 2.4, a entidade fraca é representada por uma linha mais grossa (da entidade ao relacionamento).

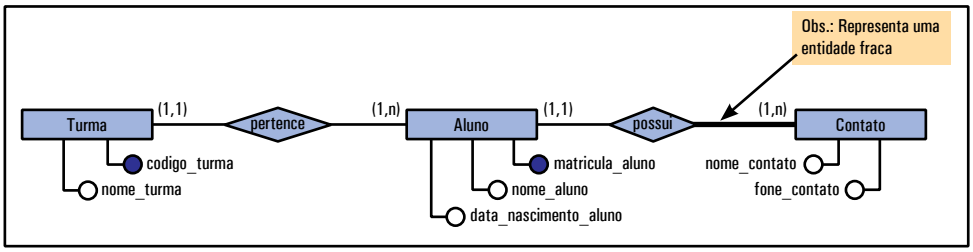


Figura 2.4 – Exemplo de uma entidade fraca

Alguns livros de banco de dados representam a entidade fraca por um retângulo duplo e o relacionamento entre entidade fraca e outra entidade por um losango duplo, como mostra a figura 2.5.

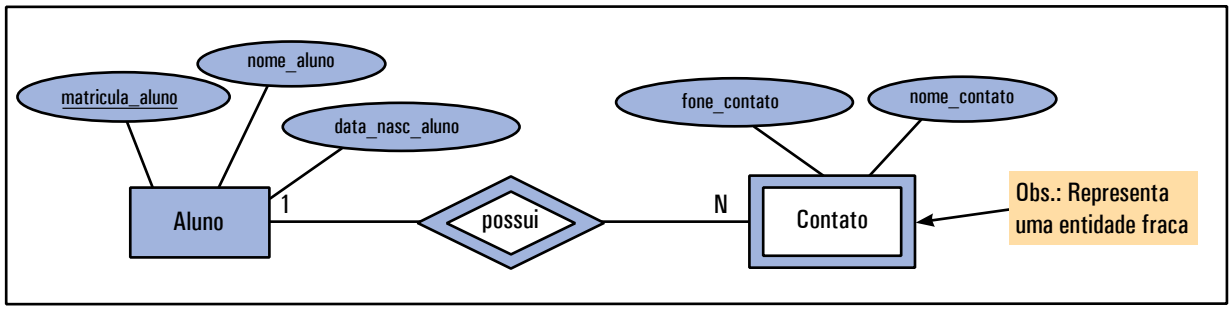


Figura 2.5 – Exemplo de uma entidade fraca usando outra notação

Atividades

1) Para que serve o Diagrama de Entidade e Relacionamento?

2) Quando um diagrama de ER deve ser construído? Quem é responsável pela sua construção?

3) Um DER pode mudar com frequência? Explique.

4) Defina o que é uma entidade e dê pelo menos três exemplos de entidades (diferentes dos apresentados no capítulo 2).

5) Para cada entidade que você apresentou no exercício 4, cite 4 atributos e diga qual o domínio de cada atributo.

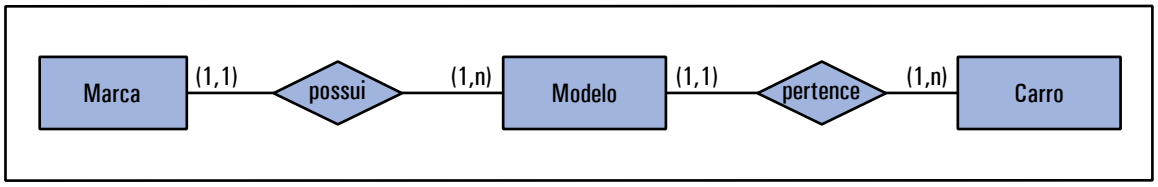
6) Explique quais os tipos de atributos que podemos ter. Para cada tipo de atributo, cite 3 exemplos.

7) Apresente 2 situações, diferentes das apresentadas no capítulo 2, em que se justifique o uso de uma entidade fraca.

8) Explique o que é uma chave primária e para que ela serve. Apresente 3 exemplos de atributos que poderiam ser chave primária e explique o porquê.

9) Uma chave primária pode assumir o valor nulo? Justifique sua resposta.

10) Dado o diagrama de ER abaixo, coloque os atributos para cada entidade e marque as chaves primárias para cada entidade.



11) Dado o diagrama abaixo, pode-se afirmar que a entidade "Endereço" possui três chaves primárias? Justifique sua resposta.