

Modelagem de Entidades e Relacionamentos



Introdução

O **Modelo Entidade Relacionamento** (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (**entidades**) envolvidos em um domínio de negócios, com suas características (**atributos**) e como elas se relacionam entre si (**relacionamentos**).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação.

Entidades

Os objetos ou partes envolvidas um domínio, também chamados de entidades, podem ser classificados como físicos ou lógicos, de acordo sua existência no mundo real.

Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, como um cliente (uma pessoa, uma empresa) ou um produto (um carro, um computador, uma roupa).

Já as **entidades lógicas** são aquelas que existem geralmente em decorrência da interação entre ou com entidades físicas, que fazem sentido dentro de um certo domínio de negócios, mas que no mundo externo/real não são objetos físicos (que ocupam lugar no espaço).

Entidades

Podemos classificar as entidades segundo o motivo de sua existência:

Entidades fortes: são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir.

Entidades fracas: ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a *entidade venda* depende da *entidade produto*, pois uma venda sem itens não tem sentido.

Entidades

Entidades associativas: esse tipo de entidade surge quando há a necessidade de associar uma entidade a um relacionamento existente.

Na modelagem Entidade-Relacionamento não é possível que um relacionamento seja associado a uma entidade, então tornamos esse relacionamento uma entidade associativa, que a partir daí poderá se relacionar com outras entidades.

Para melhor compreender esse conceito, tomemos como exemplo uma aplicação de vendas em que existem as entidades *Produto* e *Venda*, que se relacionam na forma *muitos-para-muitos*, uma vez que em uma venda pode haver vários produtos e um produto pode ser vendido várias vezes (no caso, unidades diferentes do mesmo produto).

Relacionamentos

Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas. De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classificá-los de três formas:

- **Relacionamento 1..1 (um para um);**
- **Relacionamento 1..n ou 1..* (um para muitos);**
- **Relacionamento n..n ou *..* (muitos para muitos).**

Relacionamentos

- **Relacionamento 1..1 (um para um)**: cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.

Relacionamentos

- **Relacionamento 1..n ou 1..* (um para muitos):** uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma unidade da outra entidade.

Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal. Note que temos apenas duas entidades envolvidas: usuário e dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado.

Relacionamentos

- **Relacionamento n..n ou *..* (muitos para muitos)**: neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice versa.

Atributos

Atributos são as características que descrevem cada entidade dentro do domínio. Por exemplo, um cliente possui nome, endereço e telefone. Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e consequentemente armazenar apenas as informações que serão úteis futuramente.

Uma pessoa possui atributos pessoais como cor dos olhos, altura e peso, mas para um sistema que funcionará em um supermercado, por exemplo, essas informações dificilmente serão relevantes.

Atributos

Os atributos podem ser classificados quanto à sua função da seguinte forma:

Descritivos: representam característica intrínsecas de uma entidade, tais como nome ou cor.

Nominativos: além de serem também descritivos, estes têm a função de definir e identificar um objeto. Nome, código, número são exemplos de atributos nominativos.

Referenciais: representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, uma venda possui o CPF do cliente, que a relaciona com a entidade cliente.

Atributos

Quanto à sua estrutura, podemos ainda classificá-los como:

Simples: um único atributo define uma característica da entidade. Exemplos: nome, peso.

Compostos: para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.

Exemplos

1 – Biblioteca

Entidades:

Aluno (id_aluno, nome, curso)

Livro (id_livro, título, autor)

Empréstimo (id_emprestimo, data_emprestimo, data_devolucao)

Relacionamentos:

Aluno realiza Empréstimo (1:N)

Livro pertence a Empréstimo (1:N)

Exemplos

2 – Sistema de Cinema

Entidades:

Cliente (id_cliente, nome, email)

Filme (id_filme, título, gênero)

Sessão (id_sessao, data, hora, sala)

Ingresso (id_ingresso, id_cliente,
id_sessao, valor)

Relacionamentos:

Cliente compra Ingresso (1:N)

Sessão disponibiliza Ingresso (1:N)

Filme é exibido em Sessão (1:N)

Exercício

- 1. Sistema Bancário (Internet Banking)**
- 2. Sistema de RH – Solicitação de Férias**
- 3. Sistema de Universidade**
- 4. Sistema de Hotelaria / Reservas Online**