

Modelo de Entidades e Relacionamentos

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. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de **bases de dados relacionais**.

Observação: Nem sempre criaremos modelos para um sistema completo, pois isso poderia resultar em um modelo muito extenso e difícil de interpretar. Dependendo da magnitude do que estaremos desenvolvendo, podemos criar modelos apenas para uma parte do sistema, um módulo, ou mesmo uma funcionalidade. Imagine, por exemplo, um sistema ERP de grande porte que contemple vendas, finanças, recursos humanos, etc. Várias entidades estão presentes em mais de uma parte do sistema, mas não seria muito interessante, e provavelmente nem mesmo necessário, criar um único modelo para todo o sistema, por isso pode-se dividir a modelagem em várias partes menores.

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). São exemplos disso uma venda ou uma classificação de um objeto (modelo, espécie, função de um usuário do sistema).

As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Exemplos práticos de entidades comuns em vários sistemas são Cliente, Produto, Venda, Turma, Função, entre outros.

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 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). Em determinado momento, a empresa passou a entregar brindes para os clientes que comprassem um determinado produto. A entidade Brinde, então, está relacionada não apenas com a Venda, nem com o Produto, mas sim com o item da venda, ou seja, com o relacionamento entre as duas entidades citadas anteriormente. Como não podemos associar a entidade Brinde com um

relacionamento, criamos então a entidade associativa "Item da Venda", que contém os atributos identificadores das entidades Venda e Produto, além de informações como quantidade e número de série, para casos específicos. A partir daí, podemos relacionar o Brinde com o Item da Venda, indicando que aquele prêmio foi dado ao cliente por comprar aquele produto especificamente.

Mais adiante veremos um exemplo prático onde poderemos observar a existência dessas entidades de forma mais clara.

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): cada uma das duas entidades envolvidas referencia 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.

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 a 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.

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.

Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores.

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, estas informações dificilmente serão relevantes.

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.

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.

Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF. A estes chamamos de Chave Primária.

Série: **MVC e Regras de negócio.**

Já os atributos referenciais são chamados de Chave Estrangeira e geralmente estão ligados à chave primária da outra entidade. Estes termos são bastante comuns no contexto de bancos de dados. Mantendo o exemplo anterior, a entidade cliente tem como chave primária seu CPF, assim, a venda possui também um campo “CPF do cliente” que se relaciona com o campo CPF da entidade cliente.

Diagrama Entidade Relacionamento

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras.

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado.

Em sua notação original, proposta por Peter Chen (idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

Observe na **Figura 1** um exemplo simples de um diagrama para um sistema de imobiliárias.

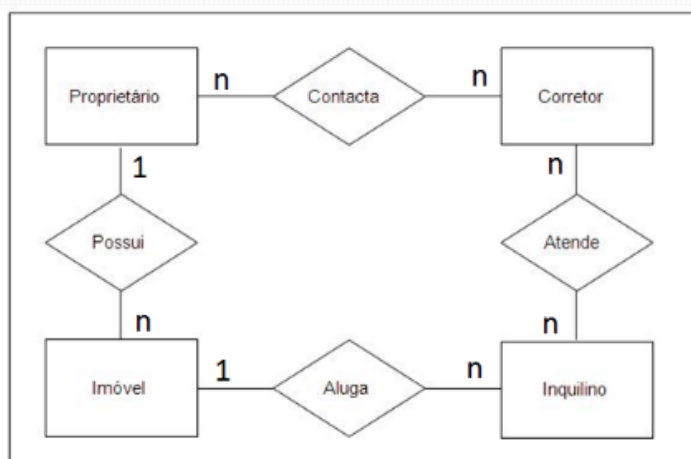


Figura 1. Diagrama Entidade Relacionamento de sistema de imobiliária

No domínio representado pelo diagrama acima temos as seguintes entidades e relacionamentos:

Proprietário contata Corretor (um proprietário pode contatar vários corretores e um corretor pode ser contatado por vários proprietários).

Corretor atende Inquilino (um corretor pode atender vários inquilinos e um inquilino pode ser atendido por vários corretores).

Inquilino aluga Imóvel (um inquilino aluga um imóvel e um imóvel pode ser alugado por vários inquilinos).

Proprietário possui Imóvel (um proprietário possui vários imóveis e um imóvel pertence a apenas um proprietário).

Uma variante da **Figura 1** pode ser vista na **Figura 2**, onde a cardinalidade do relacionamento é exibida junto do losango.

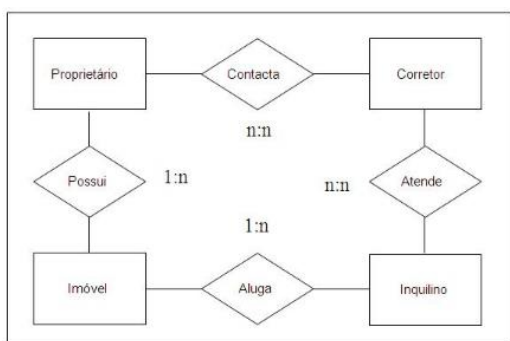


Figura 2. Diagrama de Entidade Relacionamento (variação)

Uma outra variação já mostra a cardinalidade de uma forma mais completa, deixando claro as possibilidades de números de objetos envolvidos em cada relacionamento. Nesse modelo, em cada lado do relacionamento os números aparecem no formato (X, Y) ao invés de um único número como vemos nas figuras anteriores. A **Figura 3** ilustra um exemplo desse tipo.



Figura 3. Diagrama Entidade Relacionamento (variação 2)

Neste diagrama, lemos os relacionamentos da seguinte forma:

1 ou 1 grupo possui 0 ou muitos produtos. Como de um lado temos “1 ou 1”, isso equivale a apenas “1”, pois não temos várias possibilidades. Já do lado do produto, indicamos que um grupo pode possuir nenhum produto, mas também pode possuir vários.

0 ou várias vendas contém 1 ou muitos produtos. Ou seja, um produto pode nunca ser vendido (0 vendas) como também pode ser vendido várias vezes (n vendas). Já uma venda deve conter 1 ou vários produtos, pois uma venda não pode estar vazia (0 produtos).

Os atributos, como já foi dito, podem aparecer no diagrama na forma de elipses ligadas às entidades. Essa foi a notação original proposta, mas como podemos ver na **Figura 4**, ela deixa o diagrama com muitos itens e pode atrapalhar um pouco a organização destes

cos de dados como **SQL Server Management Studio** possuem funcionalidades para criar diagramas facilmente e já gerar o código equivalente (SQL para criação das tabelas, chaves e relacionamentos, por exemplo).

Para fixar tudo que foi visto ao longo deste artigo, vamos agora desenvolver um pequeno exemplo prático em que modelaremos um sistema de bibliotecas, focando especificamente no empréstimo de livros.

Primeiramente precisamos identificar as entidades envolvidas nesse contexto. Sabemos que as entidades físicas existentes são o Usuário da biblioteca e o Livro que será emprestado. Além disso, consideraremos aqui que o livro pertence a uma Sessão, que ajuda na organização das obras do acervo. Em um sistema real pode haver outras informações sobre o livro, mas para esse exemplo a sessão é o bastante. Por fim, temos a entidade lógica Empréstimo, que tanto está relacionada com o usuário, quanto com o livro.

Assim já podemos esboçar nosso primeiro diagrama, simples, contendo as principais entidades e o relacionamento entre elas (**Figura 7**).

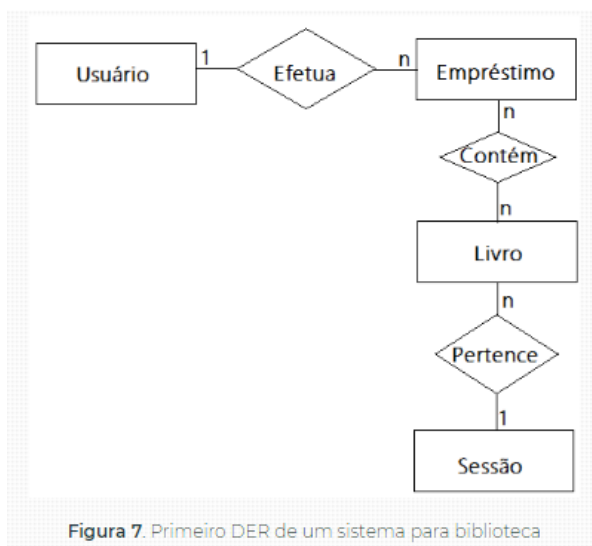


Figura 7. Primeiro DER de um sistema para biblioteca

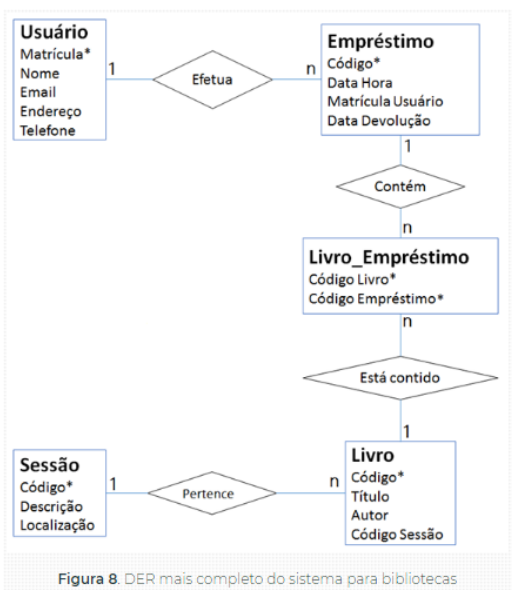
Neste primeiro diagrama podemos identificar alguns dos conceitos vistos:

Entidades fortes: Usuário, Livro e Sessão;

Entidades fracas: Empréstimo;

Relacionamentos: um Usuário efetua vários Empréstimos, vários Empréstimos contêm vários Livros, vários Livros pertencem a uma Sessão.

Agora que visualizamos o domínio no diagrama, podemos adicionar os atributos e outras entidades que se façam necessárias. Assim, passamos à **Figura 8**,



Neste ponto cabe fazer algumas observações importantes:

Especificamos os atributos de cada entidade e marcamos algumas delas com um asterisco, indicando que aquela é a chave primária da tabela, ou seja, um atributo único, que nunca poderá se repetir entre as entidades do mesmo tipo. Note que neste momento ainda não é necessário especificar o tipo de cada atributo (texto, número, data, etc.), isso só será necessário mais adiante, quando já estivermos planejando o banco de dados da aplicação.

Surgiu a entidade associativa Livro_Empréstimo, que representa os livros contidos em um empréstimo (considerando um empréstimo contém vários livros e um livro pode estar contido em vários empréstimos). Esta entidade é composta pelas chaves das duas entidades principais. Se fosse necessário, nesta entidade também poderíamos adicionar informações complementares como quantidade (não se aplica neste caso, mas caberia em um sistema de vendas, por exemplo) e observações sobre o item.

Na entidade associativa, o relacionamento n..n foi dividido em dois relacionamentos do tipo 1..n, agora lidos da seguinte forma: um empréstimo contém vários itens, mas um item só pode estar contido em um único empréstimo (restrito pelas chaves primárias); um livro pode estar contido em vários itens de empréstimo (ser emprestado várias vezes), mas cada item refere-se a um único livro.

O **Modelo Entidade Relacionamento** (e principalmente o diagrama) é uma importante ferramenta durante o desenvolvimento de sistemas, principalmente aqueles mais complexos e difíceis de visualizar sem uma análise mais aprofundada.

A correta modelagem auxilia no correto **desenvolvimento da base de dados** e evita que várias alterações sejam necessárias para corrigir erros de concepção provenientes de falhas durante a análise, ou ainda por problemas de comunicação entre os membros da equipe.

O Modelo Entidade-Relacionamento (modelo relacional) foi definido por Peter Chen em 1976, e teve como base a teoria relacional criada por E.F.Cood (1970). Segundo Chen, a visão de uma dada realidade, baseia-se no relacionamento entre conceitos desta realidade, os quais retratam os fatos que governam esta mesma realidade, e que cada conceito (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade).

O conceito de abstração permite ao analista separar da realidade em estudo, as partes que são realmente relevantes para o desenvolvimento do sistema de informações e excluir da modelagem todos os aspectos que não exercem influência sobre o ambiente a ser modelado.

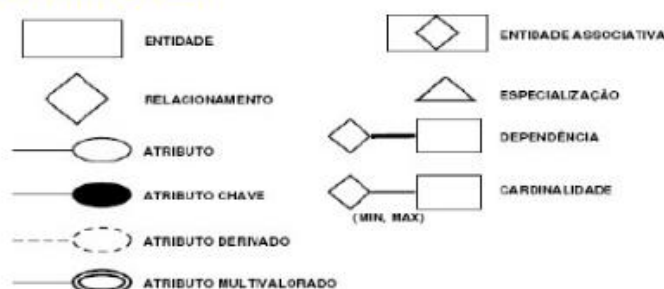
O objetivo da modelagem de dados é possibilitar a apresentação de uma visão única não redundante e resumida dos dados de um problema. Também nos ajuda a entender a estrutura e o significado destes dados.

No desenvolvimento de aplicações em banco de dados, o Modelo Entidade Relacionamento (E-R) é o mais largamente utilizado para a representação e entendimento dos dados que compõem a essência de um problema.

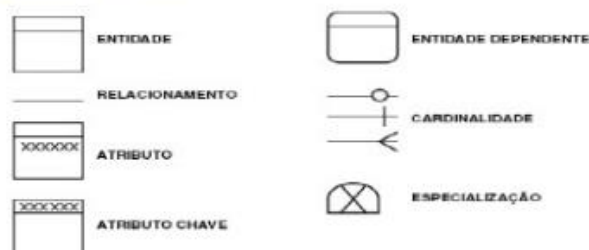
A Modelagem de Dados é amplamente utilizada como meio de conhecer os problemas organizacionais e projetar soluções.

EXEMPLOS DAS NOTAÇÕES MAIS CONHECIDAS NO MODELO ENTIDADE RELACIONAMENTO

Notação Peter Chen



Notação James Martin



MODELO CONCEITUAL

Ao se utilizar Modelagem Conceitual de dados com a técnica de Entidade e Relacionamentos, obteremos resultados e esquemas puramente conceituais sobre a essência de um problema ou negócio para o qual estamos desenvolvendo um projeto, não representando-se procedimentos ou fluxo de dados existentes.

Chen dedicou-se a destacar a importância de reconhecer os objetos que compõem um problema ou negócio, independentemente de preocupar-se com formas de tratamento das informações, procedimentos, programas, etc.

Estes objetos que desejamos conhecer e modelar para um sistema, Chen classificou em dois grupos : Entidades e Relacionamentos.

A Modelagem acontece em 3 fases:

1. Listar os Requisitos de Informação – levantamento de informações junto ao cliente.
2. Construir modelos para cada elemento

3. Escolha da tecnologia a ser utilizada – converter os modelos em sistemas

Ou seja a Modelagem Conceitual de Dados se caracteriza por ser **Independente de tipo de SGBD e Programas a serem desenvolvidos**. Ou seja, nesta fase, não nos preocupamos muito em como será feita a implementação dos programas, serviços web, componentes ou páginas do sistema.

O modelo Entidade Relacionamento propõe que a realidade seja visualizada sob três pontos de vista:

1 – Os objetos que compõem a realidade

2 - os tipos de informação ou características que se deseja conhecer sobre os objetos que compõem a

realidade e

3 - a forma como estes objetos se relacionam entre si.

Desta forma, o Modelo Entidade-Relacionamento é composto por três conceitos: Entidade, Atributo e Relacionamento. Para representação destes conceitos, existe o Diagrama Entidade Relacionamento.

DIAGRAMA ENTIDADE RELACIONAMENTO

- **Técnica mais difundida de modelagem conceitual**

- Abordagem entidade-relacionamento (ER)

É um diagrama que descreve as entidade e relacionamentos entre as entidades em um modelo de dados.

- O Modelo conceitual é representado através de diagrama entidade-relacionamento

Entidade:

Define-se entidade como aquele objeto que existe no mundo real com uma identificação distinta e com um significado próprio.

- Representação de uma entidade.

Propriedades de entidades

- Entidade isoladamente não informa nada
- É necessário atribuir propriedades às entidades
- Propriedades especificadas na forma de: relacionamentos, atributos e generalizações/especializações

ATRIBUTO

É uma característica ou qualidade de uma entidade que tem valor para o negócio, que deve ser observada pelo usuário.

exemplo: Em um cadastro de clientes de uma empresa, podemos encontrar os seguintes atributos de cliente:

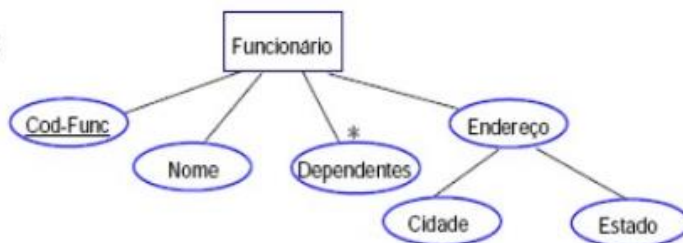
cpf, nome, endereço, cidade, bairro, estado, cep.

A definição dos atributos geralmente acontece durante várias discussões com o cliente. Todo atributo que pode vir a ser necessário em alguma tomada de decisão, no presente ou no futuro, deve ser incluído.

Atributos (campos)

Atributo: Elemento de dado que contém informação que descreve uma entidade

Ex.:



Mas cuidado... evite armazenar dados desnecessários.

Os atributos podem ser classificados de várias formas dependendo do tipo de informação que representam, obrigatoriedade de preenchimento do atributo, e composição do atributo.

Atributo Monovalorado: assume um único valor para cada elemento do conjunto x entidade

Ex.: Nome

Atributo Composto: formado por um ou mais sub-atributos. Quando nos deparamos com uma situação dessas, temos que analisar se deveremos fragmentar estes atributos em sub-atributos.

Ex.: Endereço (composto de rua, número, complemento, bairro, cep)

Atributo Multivalorado: uma única entidade tem diversos valores para este atributo (seu nome é sempre representado no plural)

Ex.: Dependentes

Atributo Determinante: identifica cada entidade de um conjunto-entidade (também conhecido como atributo chave)

Ex.: Cod_Func

Domínio de um Atributo: conjunto de valores permitidos para o atributo.

Ex.: Sexo {M, F}, Estado Civil (C, S, D)

Atributo Obrigatório e Opcional

Alguns atributos de uma entidade são obrigatórios outros são opcionais.

Nome é obrigatório pois toda pessoa deve ter um nome.

Telefone é opcional pois nem toda pessoa possui um telefone.

Para chegarmos a essa conclusão, temos que fazer uma análise minuciosa de campo a campo da(s) entidade(s) que estamos criando.

Então podemos fazer as seguintes definições:

Atributo obrigatório - é aquele que para uma instância de uma entidade ou relacionamento **deve** possuir um valor. (NOT NULL)

Atributo opcional - É aquele que para uma instância da entidade ou relacionamento **pode** possuir um valor. É o que chamamos de valor nulo. (NULL)

Como escolher o nome, tipo de dado e tamanho de um atributo?

Os nomes dos atributos devem sempre ser significativos e descrever claramente o tipo de informação que o atributo representa.

O tipo de dado de um atributo depende da informação que deseja ser armazenada no atributo. O tipo de dado se for armazenar um dado que não fará parte de cálculos, um nome por exemplo, o tipo pode ser **texto**, se for armazenar valores numéricos que farão parte de cálculos o tipo será **numérico (inteiro,simples...)**, se for armazenar datas o tipo será **data** e assim por diante.

O tamanho do atributo define a quantidade de informação que você pode armazenar, assim: para um campo do tipo texto com tamanho 30 você poderá armazenar textos com no máximo 30 caracteres, um campo numérico de tamanho 2 armazena números com até dois dígitos.

Tipos de dados:

Número

Smallint - Valor Inteiro de 15 bits com sinal

integer - Valor Inteiro de 31 bits com sinal

float(p) - formato científico com precisão

decimal(p,q) - formato decimal

Texto

char(n) - Cadeia de tamanho fixo de n caracteres

varchar(n) - cadeia de tamanho variável de n caracteres

long varchar - cadeia de caracteres de tamanho variável

Data/Hora

Date

time

timestamp

Dicas para identificação de Entidades, Relacionamentos e Atributos.

Dado um texto descrevendo o banco de dados a ser projetado:

A presença de um substantivo usualmente indica uma entidade,

A presença de um verbo é uma forte indicação de um relacionamento,

Um adjetivo, que é uma qualidade, é uma forte indicação de um atributo,

Um advérbio temporal, qualificando o verbo, é uma indicação de um atributo do relacionamento.