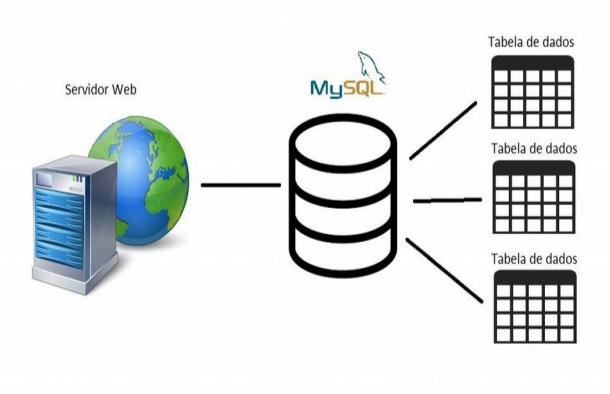
Curso de Operador de Tecnologia da Informação e Comunicações

Banco de dados

Modelagem de dados

Objetivo de Aprendizado

Entender os principais conceitos relacionados a banco de dados na área de modelagem dos dados. Será apresentado o modelo conceitual.



No projeto de um banco de dados, algumas regras devem ser seguidas para que se possa obter sucesso no projeto final. Um banco de dados mal projetado compromete todo o sistema.

Modelos de dados

Modelos de dados constituem artefatos centrais de arquitetura de dados. De maneira simples, podem ser definidos como representações gráficas dos requisitos de informação relevantes a uma determinada área de assunto: o negócio de uma empresa como um todo, uma fatia do negócio, o escopo de um sistema de informação, etc.

Representando o negócio sob a perspectiva dos dados, são utilizados em todas as fases de um projeto de TI, complementando a perspectiva “funcional” dos modelos de processos.

Artefatos de dados tendem a ser mais estáveis do que artefatos de processos, mas também evoluem, sofrendo alterações à medida que os requisitos de negócio mudam.

Os componentes básicos de um modelo de dados são **entidades**, **atributos** e **relacionamentos**. É importante frisar que um modelo de dados não se resume aos objetos gráficos que compõem o diagrama do modelo, mas também à documentação adjacente.

Idealmente, o processo de modelagem de dados deve ser implementado através de uma abordagem de detalhamento sucessivo, com pelo menos três camadas ou níveis de abstração, partindo-se do nível mais abrangente para o mais detalhado (Figura 1).

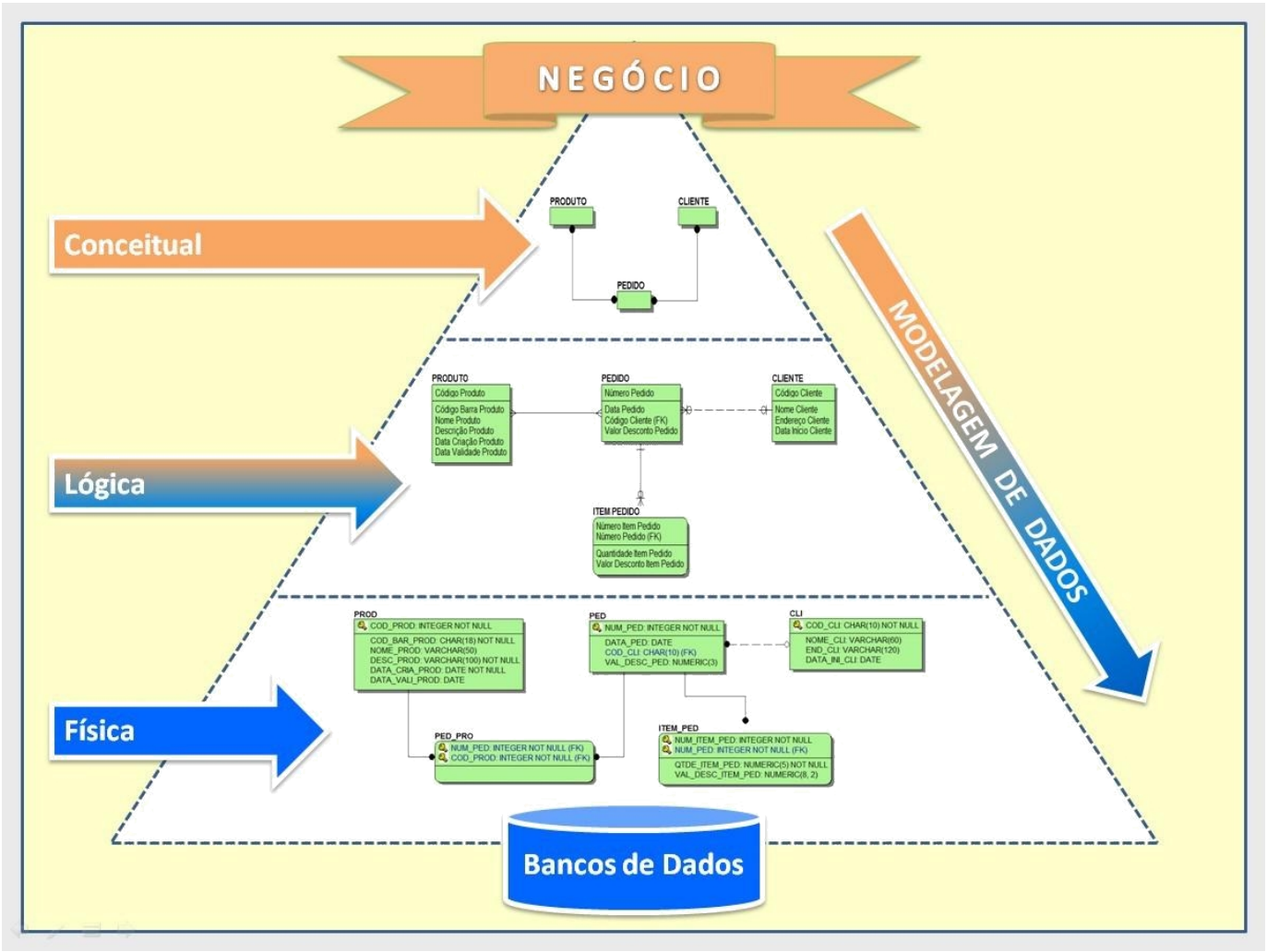


Figura 1 - As três camadas da modelagem de dados: conceitual, lógica e física

Modelo conceitual

DEFININDO MER (MODELO ENTIDADE RELACIONAMENTO) E DER (DIAGRAMA ENTIDADE-RELACIONAMENTO)

Quando se inicia o desenvolvimento de um novo sistema, ou mesmo de uma nova funcionalidade para um sistema existente, um dos primeiros passos a ser executado é o estudo e levantamento dos requisitos necessários para a construção do produto final. Durante essa análise, identifica-se as principais partes e objetos envolvidos, suas possíveis ações e responsabilidades, suas características e como elas interagem entre si.

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento propriamente dito, fornecendo informações sobre os aspectos relacionados ao domínio do projeto em questão.

Modelo Entidade Relacionamento

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 classifica-los de três formas:

- **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.
- **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.
- **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.

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.

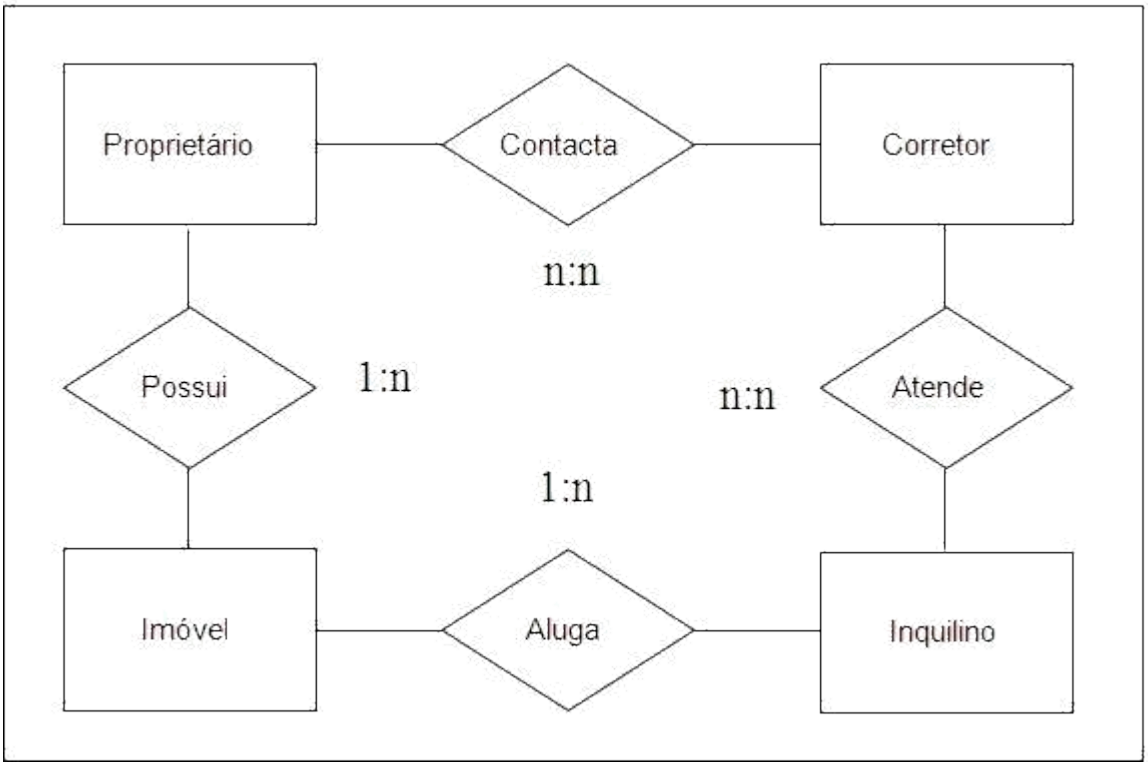


Figura 2. Diagrama de Entidade Relacionamento (variação)

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.

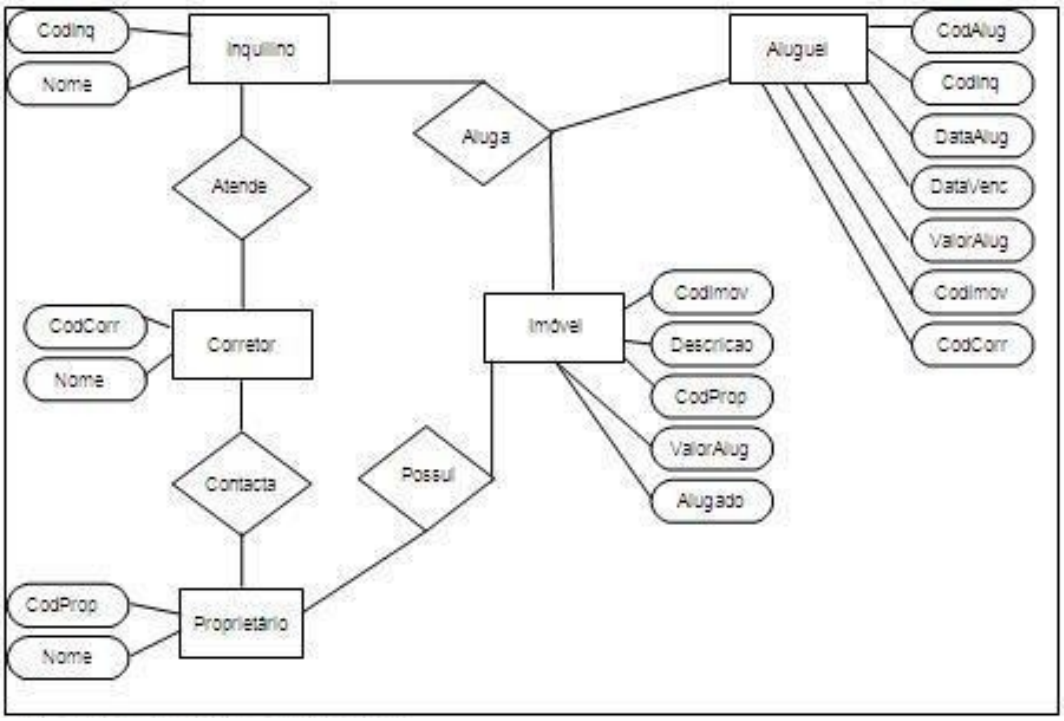


Figura 4. Atributos apresentados como elipses

Em uma notação mais atual, comumente utilizada na **UML**, os atributos aparecem listados dentro do próprio **retângulo da entidade**, enquanto o nome da entidade aparece no topo na forma de título. Na Figura 5 temos um exemplo.

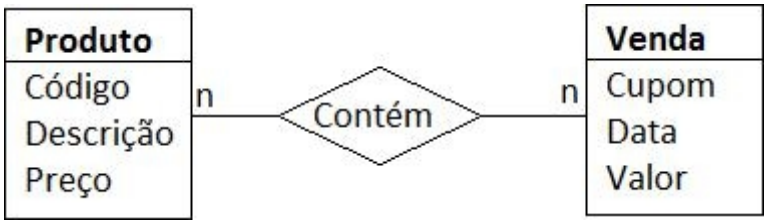


Figura 5. Diagrama com atributos nas entidades

Ferramentas CASE

Do inglês Computer-Aided Software Engineering, as chamadas ferramentas CASE são aquelas baseadas em computadores (softwares) utilizadas na Engenharia de Software para auxílio nas atividades desde análise de requisitos até ,modelagem de dados.

No contexto desse artigo, as ferramentas CASE permitem a criação de diagramas de forma simples em um ambiente de fácil utilização e com recursos para incluir as principais regras de composição dos diagramas. Exemplos comuns desse tipo de ferramenta são: Star UML, Astah e ERwin Data Modeler. Na Figura 6 vemos um exemplo de diagrama sendo construído no Astah.

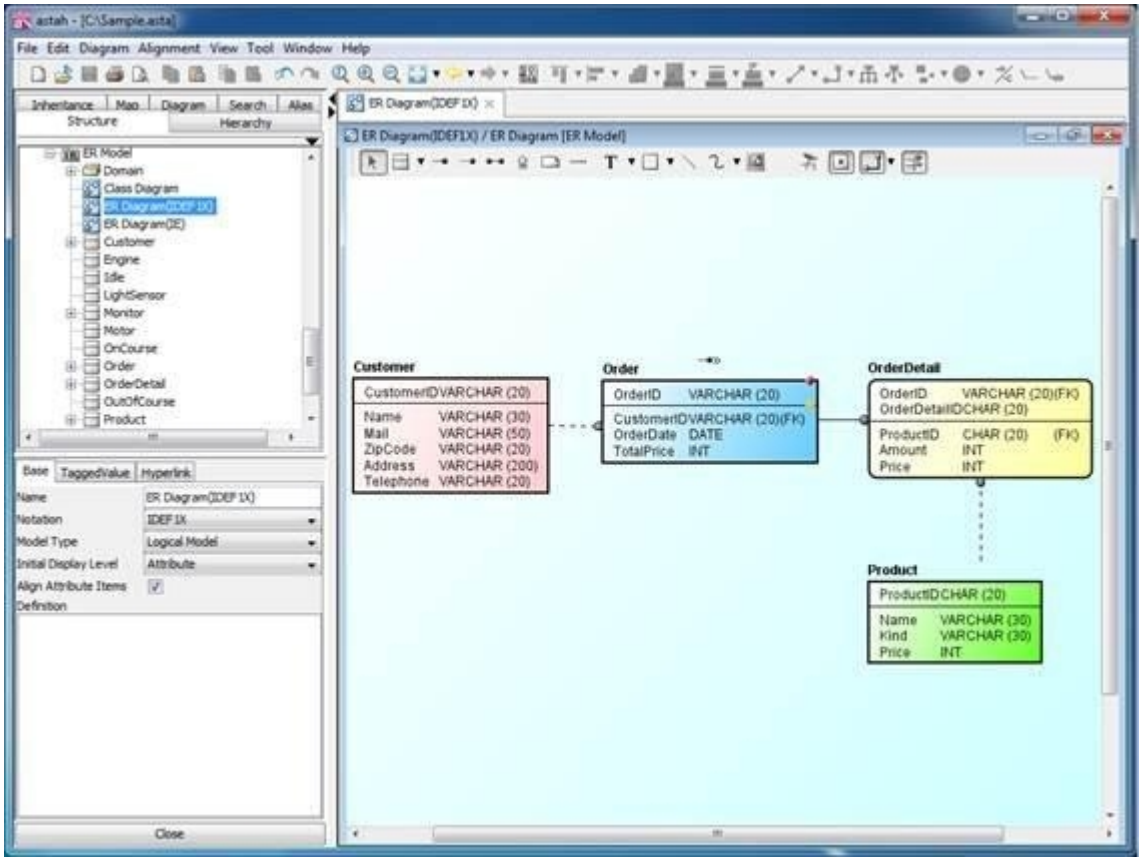


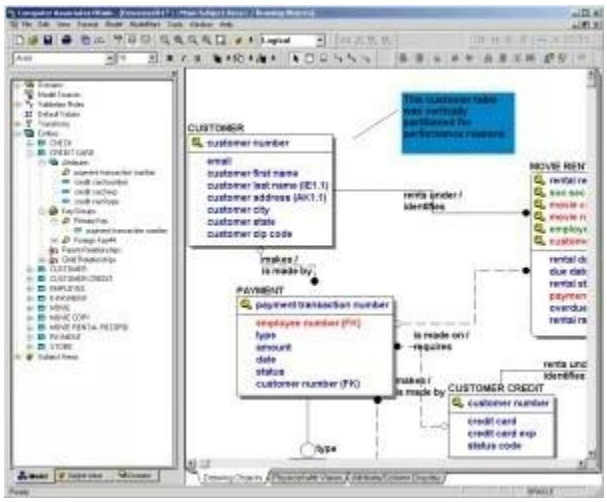
Figura 6. Diagrama no Astah Community

Além dessas ferramentas específicas, alguns IDEs (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) como o Visual Studio e ferramentas de gerenciamento de bancos 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).

Exemplos de ferramentas case para modelagem de dados.

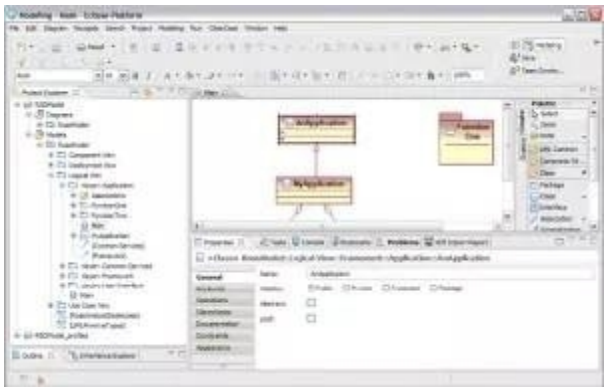
CA Erwin

http://erwin.com/worldwide/portuguese-brazil



IBM’s Relational Rose

http://www-142.ibm.com/software/products/br/pt/enterprise/

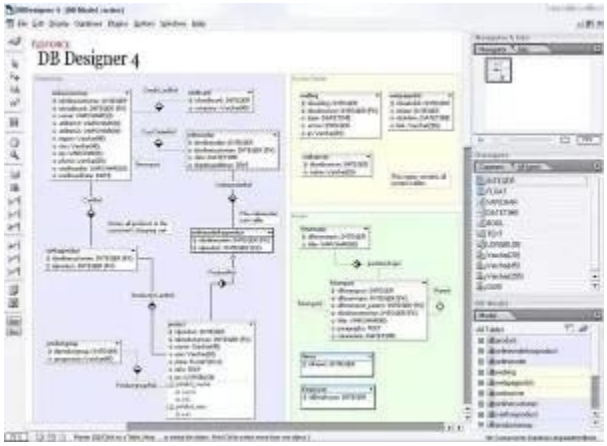


Oracle’s Designer

http://www.oracle.com/technetwork/developer-tools/developer-suite/downloads/index.html

DBDesigner

http://www.fabforce.net/dbdesigner4/



ELEMENTOS ESTRUTURAIS DE UM MODELO CONCEITUAL DE DADOS

A finalidade do modelo conceitual de dados é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Para isto, torna-se necessário o entendimento e a correta aplicação dos mecanismos de abstração, utilizados na modelagem conceitual de dados.

1 - ENTIDADES

Formam um conjunto de “coisas” com conceitos comuns às quais desejamos armazenar os dados.

Entidades podem ser pessoas, lugares, organizações, objetos físicos e tangíveis.

As entidades são representadas através de um retângulo com o nome da entidade escrito em seu centro. Conforme figura a seguir, as entidades são classificadas em dois tipos: Entidades Fortes e Entidades Fracas.

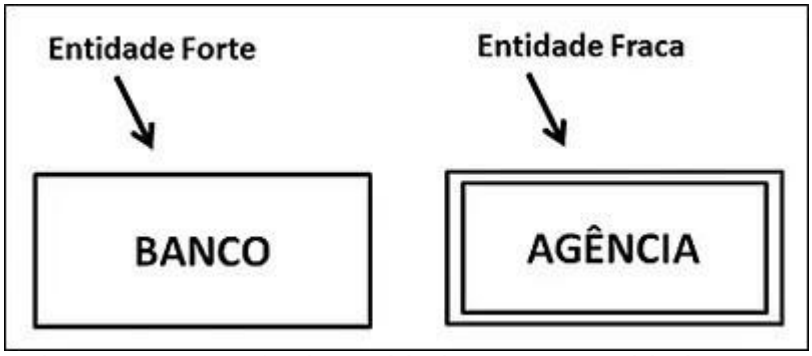


Figura 1 - Notação de Entidade Forte x Entidade Fraca

As entidades fortes possuem um alto grau de independência de existência de identificação. Geralmente, outras entidades podem depender dela para serem identificadas. Podemos tomar como exemplo a entidade “BANCO”, onde a existência da mesma não depende de nenhuma outra entidade para ser identificada.

As entidades fracas possuem dependência de existência e/ou identificação. São sempre ligadas a outras tabelas através de relacionamentos. Podemos tomar como exemplo a entidade “AGENCIA”, onde a existência e identificação da mesma estão vinculadas a outra entidade forte, no caso o “BANCO”.

2 - RELACIONAMENTOS

Relacionamentos são associações entre entidades com um significado específico dentro do mundo real. Os objetos do mundo real não ocorrem de forma isolada, eles se associam ou se vinculam.

A figura de um relacionamento é representada através de um losango, tal como as entidades os relacionamentos são classificados em fortes ou fracos. Tal como as entidades, os relacionamentos também possuem nome e devem expressar o real significado dentro do contexto modelado. A figura a seguir mostra como os relacionamentos são representados em um modelo conceitual de dados.

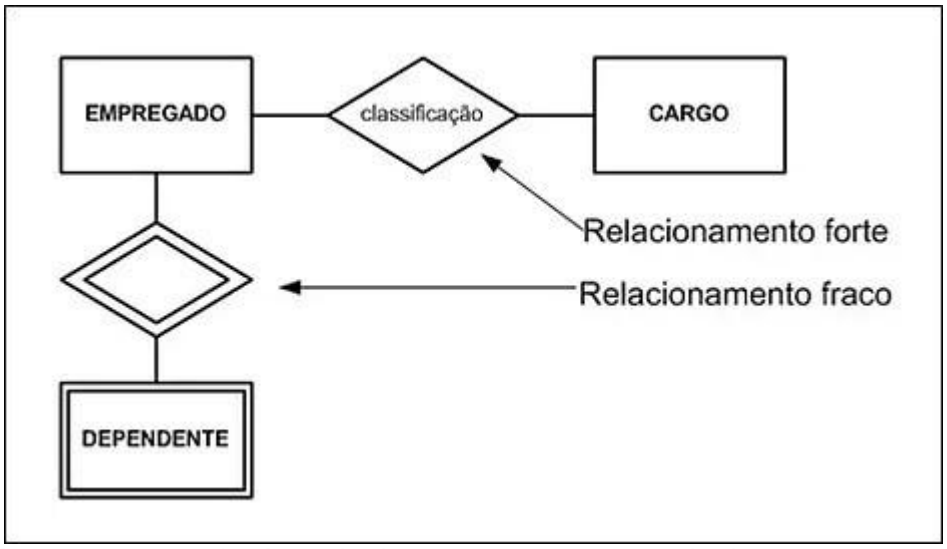


Figura 2 - Notação de Relacionamento Forte e Relacionamento Fraco

Na figura acima DEPENDENTE é uma entidade fraca em relação ao EMPREGADO, sempre que esta relação existir de forma fraca, o relacionamento também será fraco, por esta razão o losango desta relação está representado com uma linha dupla. Já na relação entre EMPREGADO e CARGO não há dependência de existência ou identificação, pois um CARGO não depende de um EMPREGADO para existir e ser identificado e vice-versa.

Quando tratamos de relacionamentos, devemos ter em mente três conceitos importantes que influenciam diretamente na modelagem e entendimento de um modelo conceitual. Os conceitos são o grau, cardinalidade e tipo do relacionamento.

Grau dos Relacionamentos

O grau de um relacionamento corresponde ao número de entidades envolvidas na mesma relação. O grau de um relacionamento pode ser:

- Binário: Onde duas entidades participam de um relacionamento. Este é o grau utilizado na maioria dos relacionamentos.
- Ternário: Onde três entidades participam de um relacionamento. Muito se discute sobre o uso e aplicabilidade de relacionamentos com grau maior que dois (ternários e n-ários) em modelos de dados. Alguns autores sugerem inclusive que esses relacionamentos não sejam adotados.
- N-ário: Onde quatro ou mais entidades participam de um relacionamento.

A figura a seguir mostra exemplos comuns de graus de relacionamentos.

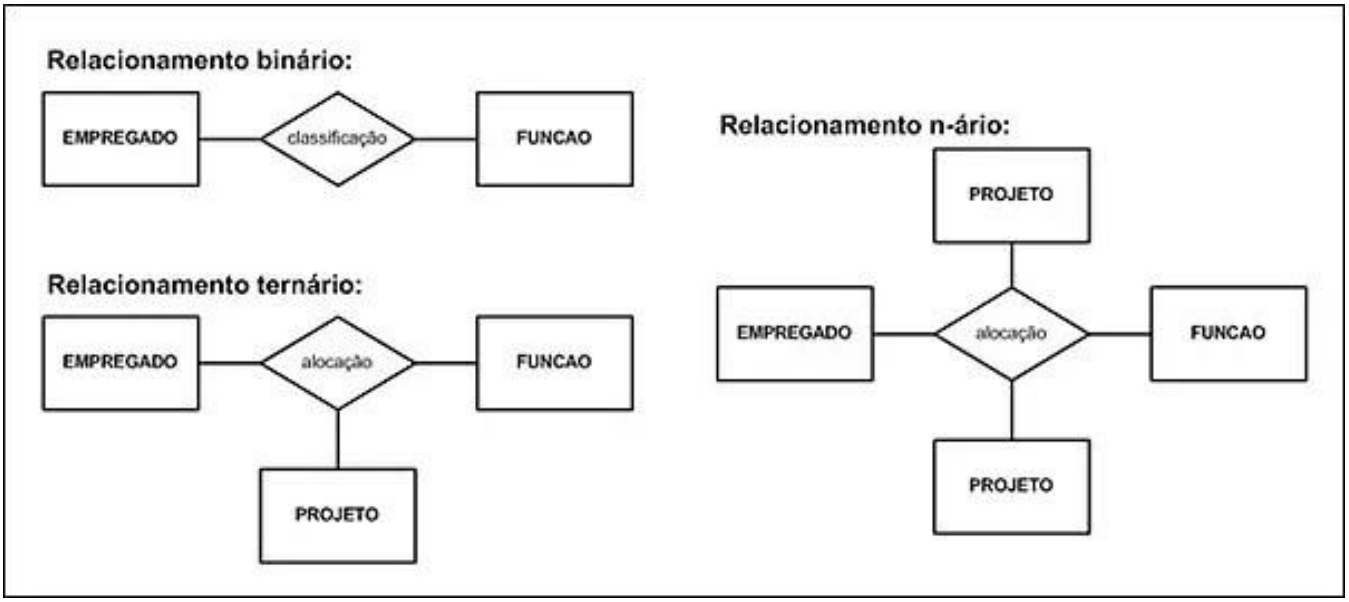


Figura 3 - Grau dos Relacionamentos em Modelos Conceituais

Cardinalidade

A cardinalidade representa a quantidade de vezes que um elemento de um conjunto de entidades pode, em um determinado instante, estar associado em um dado relacionamento, a outros elementos de outras entidades.

A cardinalidade de uma relação é definida em cada um dos sentidos do relacionamento por um conjunto (x,y) onde x representa a cardinalidade mínima e y representa a cardinalidade máxima.

A cardinalidade mínima é responsável por orientar a obrigatoriedade (opcionalidade) do relacionamento. Já a cardinalidade máxima é responsável por definir a quantidade máxima de vezes que um elemento pode estar associado no relacionamento.

Até agora, por fins didáticos, a cardinalidade não estava sendo representada nos exemplos deste artigo, porém vale a pena destacar que seu uso é obrigatório, pois as cardinalidades refletem regras que obrigatoriamente devem ser representadas em um modelo conceitual de dados.

A figura a seguir mostra um exemplo de uso das cardinalidades em um relacionamento dentro de um modelo conceitual de dados.

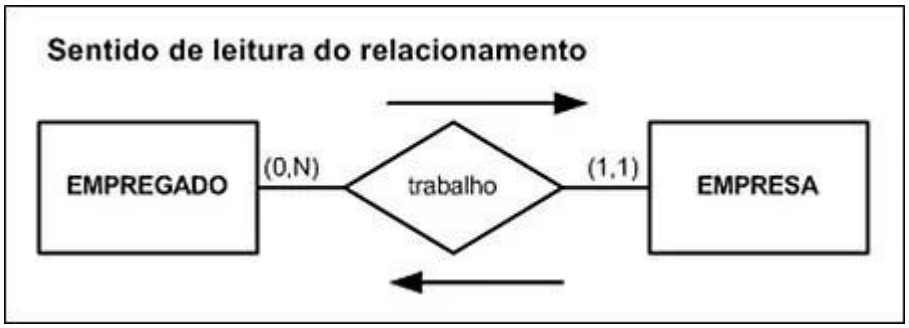


Figura 4 – Exemplo do uso de Cardinalidade nos Relacionamentos

No exemplo acima, um EMPREGADO trabalha em uma e somente uma EMPRESA e, em uma EMPRESA trabalham nenhum ou vários EMPREGADOS. Ou seja, dentro do contexto que foi modelado, é impossível existir um EMPREGADO sem uma EMPRESA associada, porém é totalmente viável criar uma EMPRESA e não associar inicialmente algum EMPREGADO.

Tipos de Relacionamentos

O simples fato de associar duas entidades através de um relacionamento com suas cardinalidades às vezes não são suficientes para representar todas as regras de negócio existentes dentro dessas relações.

Para isto, podemos usar mecanismos de representação um pouco mais detalhados. Sob esta ótica, podemos ainda classificar os relacionamentos em três tipos:

- Relacionamentos independentes;
- Relacionamentos Contingentes;
- Relacionamentos mutuamente exclusivos.

Relacionamentos Independentes:

Tipo de relacionamento presente na maioria das relações. Não há necessidade de interpretação simultânea de outro relacionamento. Ou seja, é independente, não depende de ninguém para existir ou influenciar o seu comportamento.

Relacionamentos Contingentes:

Estabelecem associações simultâneas entre os elementos envolvidos. Ou seja, mais de um relacionamento deve ocorrer em um mesmo instante.

Sua representação é recomendada, pois envolvem regras de negócio específicas que se não mapeadas neste momento, fatalmente serão esquecidas mais adiante no decorrer do projeto.

A figura a seguir mostra um exemplo de relacionamento contingente.

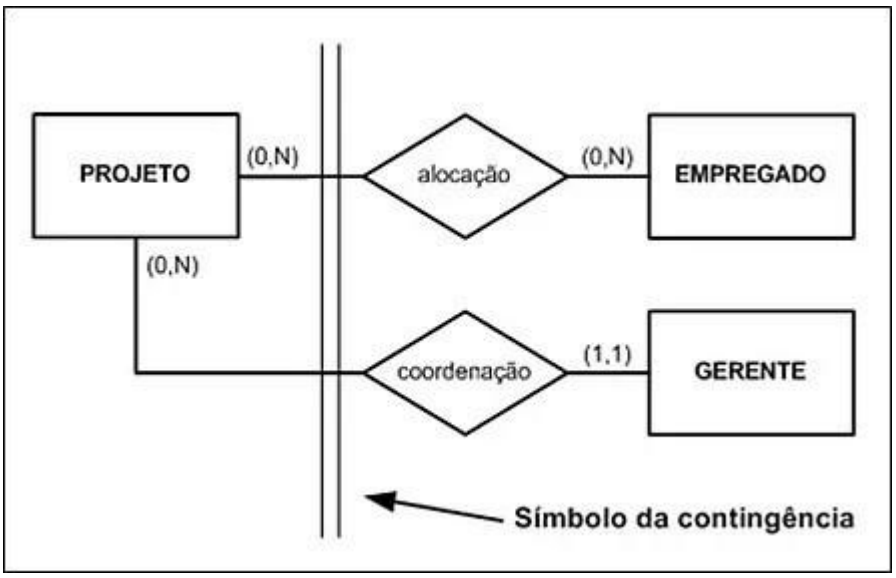


Figura 5 - Exemplo de Relacionamento Contingente

No exemplo acima é impossível alocar empregados em um projeto sem um gerente definido e também não é possível definir um gerente para um projeto sem existir empregados alocados no projeto. Ou seja, os dois relacionamentos devem ocorrer no mesmo instante.

Relacionamentos Mutuamente Exclusivos:

Estabelecem associações onde, se um relacionamento ocorre, os outros não deverão ocorrer em relação a um determinado objeto.

Sua representação é recomendada, pois envolvem regras de negócio específicas que se não mapeadas neste momento, fatalmente serão esquecidas mais adiante no decorrer do projeto.

A figura a seguir mostra um exemplo de relacionamento mutuamente exclusivo.

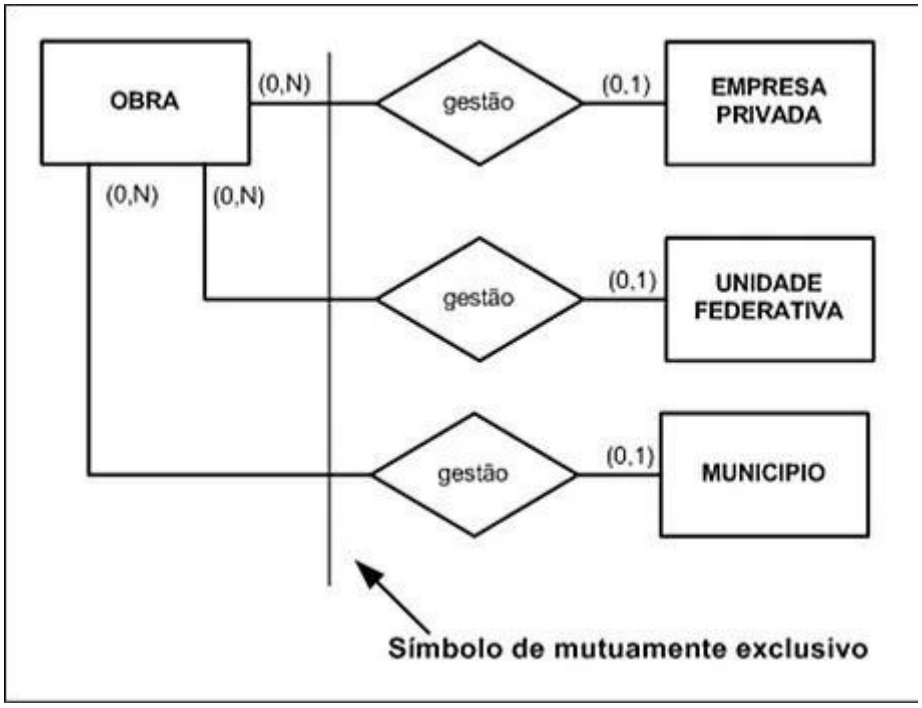


Figura 6 – Exemplo de relacionamento mutuamente exclusivo

O exemplo acima reflete um exemplo onde uma obra é gerida por uma Empresa Privada ou por uma Unidade Federativa ou por um Município. Ou seja, três tipos de entidades podem gerir uma obra, porém somente uma é a entidade gestora.

3 - ATRIBUTOS

Os atributos são informações que caracterizam as entidades e os relacionamentos. Um atributo pode: identificar, descrever, qualificar, quantificar ou registrar o estado/situação/ocorrência de uma entidade.

No modelo conceitual são representados através de “pirulitos” colocados juntos as entidades.

Os atributos podem ser classificados em quatro tipos:

Atributo identificador: Representado através de uma bola cheia na extremidade do atributo. Atributos identificadores identificam ou compõe a identificação única de uma ocorrência em uma entidade. Vale ressaltar que uma entidade e/ou relacionamento pode possuir mais de um atributo identificador, desde que os mesmos em conjunto componham a identificação única.

Atributo não identificado: Representado através de uma bola vazia na extremidade do atributo. Corresponde a maioria das ocorrências de uma entidade. Além disso, atributos não identificados podem ser opcionais, ou seja, em algumas instâncias de entidade, alguns atributos poderão conter valores nulos.

Atributos multivalorados: Representado através de uma flor ou asterisco na extremidade do atributo.

Atributos multivalorados são utilizados para representar mais de uma ocorrência de valor de um atributo dentro de uma mesma instância de uma entidade. Exemplo: Geralmente, uma pessoa possui mais de um número de telefone. Como o objetivo do modelo conceitual é capturar a essência do negócio sem levar em conta aspectos de implementação, este tipo de abordagem é utilizado para representar todas essas instâncias em um único atributo, porém deve-se ter em mente que este tipo de abordagem não deve ser utilizado a partir da modelagem lógica de dados, onde entrarão em cena os conceitos de normalização.

Atributos compostos: Representados através de uma oval com vários nós na extremidade do atributo.

Atributos compostos são utilizados para representar mais de um tipo de informação (qualificação) em um atributo. Tal como o atributo multivalorado, seu uso é recomendado somente no modelo conceitual de dados.

A figura a seguir, mostra os tipos de atributos utilizados em um modelo conceitual de dados.

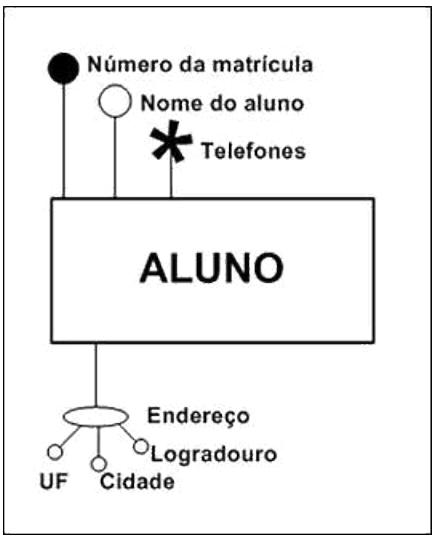


Figura 7 - Notações utilizadas em atributos

O exemplo acima representa atributos comuns aos alunos de qualquer instituição: o número da matrícula é um atributo identificador, o nome do aluno é um atributo não identificador. Já o atributo telefones é um atributo multi-valorado, onde representa os diversos telefones que um aluno possui. Endereço é considerado um atributo composto, pois é formado pela composição da UF, Cidade e Logradouro.

MECANISMOS AVANÇADOS DE ABSTRAÇÃO UTILIZADOS EM UM MODELO CONCEITUAL DE DADOS

Os mecanismos avançados de abstração são excelentes recursos para melhorar o entendimento e representação dos modelos de dados. Alguns mecanismos como o auto relacionamento já são bastante utilizados pelos técnicos que produzem os modelos de dados. As seções a seguir irão detalhar e mostrar exemplos de como esses mecanismos são utilizados.

1 -REPETIÇÃO

Existem situações em que uma mesma instância de uma entidade pode selecionar com outra mesma instância de outra entidade várias vezes.

A repetição permite representar as regras de negócio que expressam a quantidade de instâncias de relacionamentos que podem ser estabelecidos entre os mesmos elementos das entidades participantes de um relacionamento.

Quando trabalhamos com o mecanismo de Repetição, é obrigatória a existência de um atributo identificador no relacionamento onde ocorre a repetição.

A repetição é indicada no modelo através de um número que indica quantas vezes uma mesma instância de uma entidade pode se relacionar com outra instância mais de uma vez em outra entidade. Se o número de vezes for indefinido utiliza-se a letra “N”.

A figura a seguir mostra um exemplo de uso do mecanismo da Repetição.

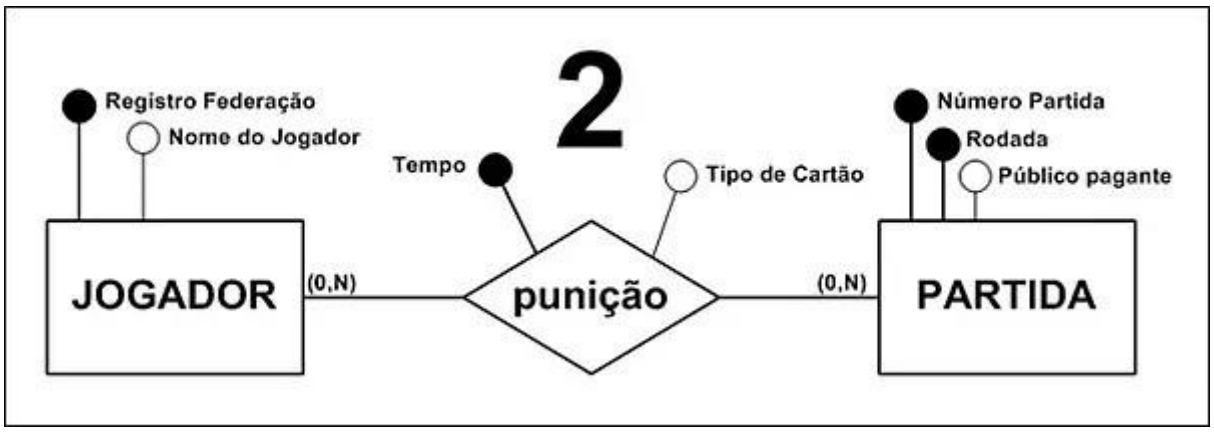


Figura 8 - Exemplo do uso da Repetição em um MCD

No exemplo acima temos a representação de um relacionamento onde são registrados os cartões (amarelo ou vermelho) recebidos pelos jogadores de futebol em uma partida. Conforme regra vigente, um jogador não pode receber 2 cartões amarelos em uma mesma partida. Se isto acontecer, o segundo cartão amarelo é convertido em vermelho e ele é automaticamente expulso da partida. O número 2 colocado em cima do relacionamento “punição” representa esta regra.

2 - AUTO RELACIONAMENTO

O auto relacionamento, também conhecido como relacionamento recursivo representa a associação entre elementos pertencentes à mesma entidade.

Em um auto relacionamento temos sempre dois papéis formados pelos elementos de uma entidade. A representação desses papéis é obrigatória e fundamental para o entendimento do modelo. De forma geral, prefiro utilizar um substantivo para nomear o relacionamento e verbos ou expressões verbais para nomear os papéis.

Geralmente, utilizamos um auto relacionamento quando:

1- Desejamos representar estruturas hierárquicas dentro da mesma entidade. Este tipo de representação sempre utilizará uma cardinalidade (0,1) x (0,N). A figura a seguir mostra um exemplo de utilização de um auto relacionamento com essas características.

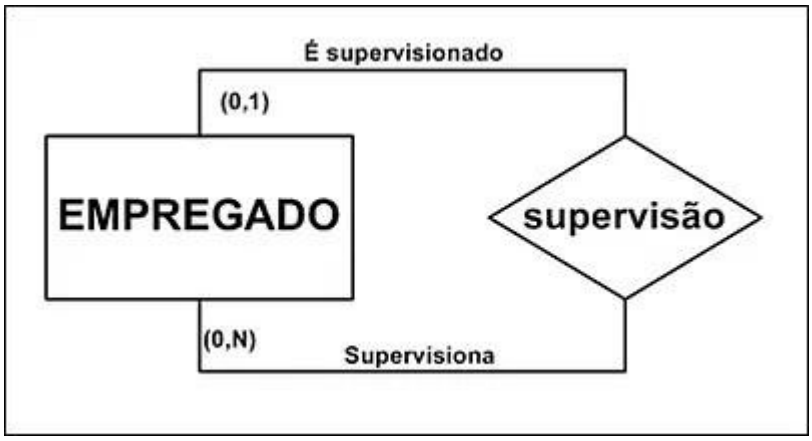


Figura 9 - Exemplo do uso de um auto relacionamento para representar uma hierarquia

Conforme exemplo da figura acima, conseguimos montar uma estrutura de rede hierárquica de empregados dentro de uma empresa.

2 - Desejamos representar estruturas similares a composições com a mesma entidade. Este tipo de representação sempre utilizará uma cardinalidade (0,N) x (0,N). A figura a seguir mostra um exemplo de utilização de um auto relacionamento com essas características.

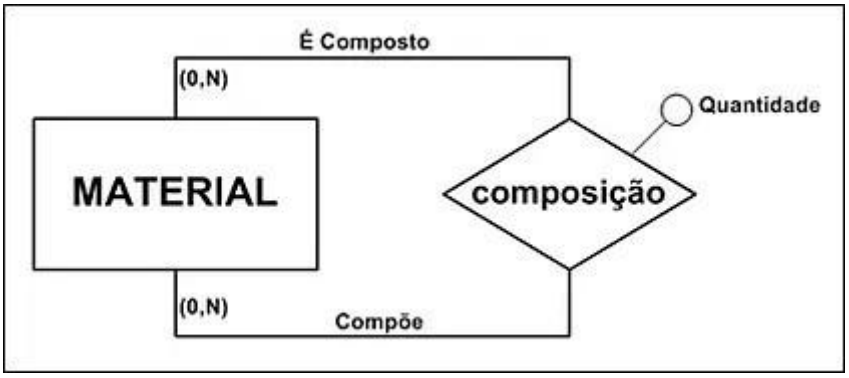


Figura 10 - Exemplo de um auto relacionamento para representar uma composição

Conforme exemplo da figura acima, conseguimos montar uma estrutura que forma uma composição de materiais. O relacionamento composição será responsável por armazenar essas informações. Para ilustrar esta relação com exemplos reais, podemos imaginar que a entidade MATERIAL armazena todos os itens materiais de um carro. O relacionamento “composição” será responsável por montar a composição desses materiais. No caso de um carro, existirá várias instâncias de materiais dentro da entidade MATERIAL, inclusivo o material “motor”. O “motor” por sua vez é formado por vários materiais de menor porte e diversas quantidades. Por esta razão foi colocado o atributo quantidade dentro do relacionamento.

3 - GENERALIZAÇÃO E ESPECIALIZAÇÃO

Existem situações onde precisamos representar entidades comuns com um maior ou menor grau de propriedades em cada uma, sempre mantendo uma visão hierárquica entre essas entidades. Dependendo da situação, podemos utilizar a Generalização ou a Especialização.

A consiste em criar um conceito superior para as entidades existentes, mantendo uma relação de hierarquia de entidade entre a nova entidade (entidade pai) e as entidades já existentes (entidades filhas). A figura abaixo demonstra um exemplo de uso do mecanismo de generalização.

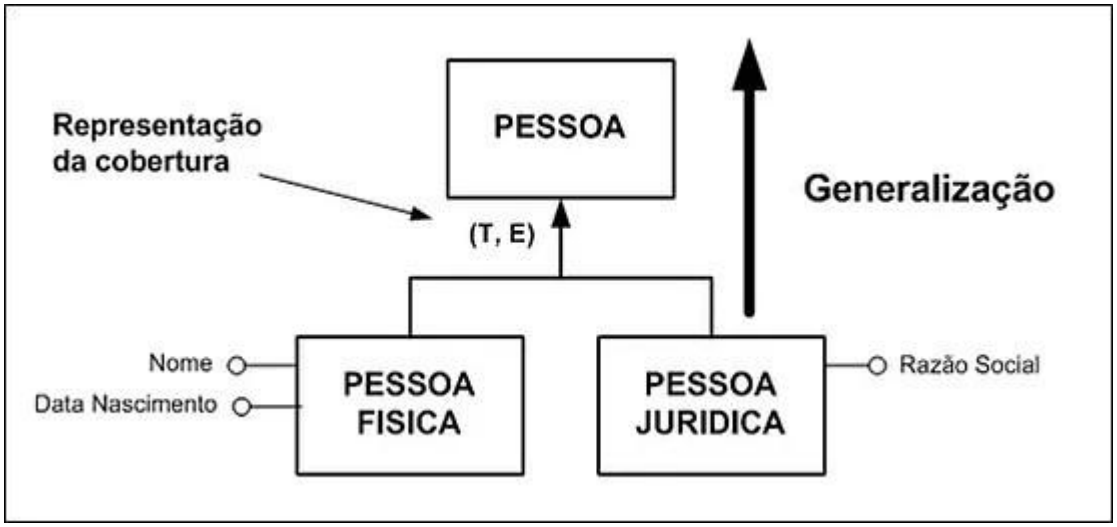


Figura 11 - Uso da Generalização em um Modelo Conceitual

Já a Especialização consiste em criar novos conceitos (entidades filhas) a uma entidade já existente, mantendo uma relação de hierarquia das novas entidades com a entidade pai (já existente). A figura a seguir demonstra um exemplo de uso do mecanismo de especialização.

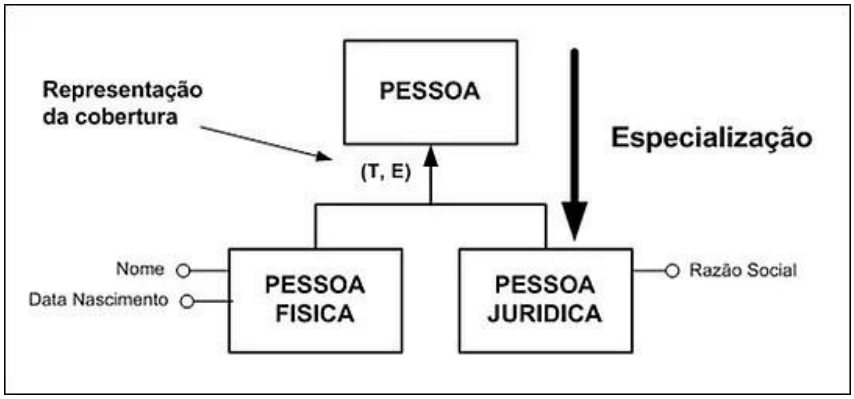


Figura 12 - Uso da Especialização em um MCD

Quando trabalhamos com mecanismos de Generalização / Especialização utilizamos regras de negócio que representam condições envolvendo especialização. A essas condições damos o nome de cobertura.

A cobertura é representada do lado da seta que indica a especialização / generalização por um par de valores (X,Y) onde X representa o conteúdo e Y representa a cobertura.

O valor de conteúdo pode ser representado pelas letras T e P onde:

- **T = Conteúdo Total** (Toda instância de um elemento “E” deve pertencer também a uma instância em uma entidade filha especializada).
- **P = Conteúdo Parcia**l (Pode existir uma instância do elemento “E” que não pertença às entidades especializadas)
- O valor de cobertura pode ser representado pelas letras E e S onde:
- **E = Cobertura Exclusiva** (Toda instância do elemento “E” pode existir no máximo em uma instância nas entidades especializadas).
- **S = Cobertura Sobreposição** (Toda instância do elemento “E” pode existir em várias instâncias das entidades especializadas).

Exemplo de Relacionamento Especialização

Integridade

Realizada por meio de restrições, que são condições obrigatórias impostas pelo modelo, como exemplo integridade de domínio ou referencial.

A integridade de domínio Implementa restrições nas informações armazenadas, quanto mais limitados os dados que podem ser inseridos em um campo, menor será a probabilidade de entrada de dados errados no banco de dados. Também especifica quais dados são absolutamente necessários para que o banco de dados funcione apropriadamente. Podendo ser:

- **Restrições de check**: Permite controlar os dados inseridos em certa coluna, de qualquer tabela, avaliando uma expressão. Ex: maior que, menor que, diferente de.
- **Nulidade**: controla se existe obrigatoriamente o valor para aquela coluna. O valor nulo deve ser evitado, pois implica em desperdício de espaço. Deve utilizar nulo quando o valor existe, mas é desconhecido; o valor é conhecido, mas está ausente.
- **Unicidade**: Toda tabela deve ter definido um atributo ou conjunto de atributos cujo valor ou combinação deve ser distinto em qualquer ocorrência da tabela.
- **Unique**: Determina que todos os valores, de uma determinada coluna, precisam ser exclusivos (diferentes). Gera integridade.
- **Default**: Estabelece um valor padrão para determinada coluna.

Atributo de uma tabela que referencia à outra tabela, a chave primária da entidade pai que migra para a entidade filha através de um relacionamento.

A integridade referencial garante que linhas relacionadas em um par de tabelas continuem relacionadas mesmo depois de terem sido feitas alterações na tabela, desta forma, uma linha em uma tabela que se refere a outra tabela deve referenciar uma linha existente naquela tabela.

Representação de Integridade entre Tabelas

Chave primária compreende a identificação única de uma ocorrência em uma entidade, um identificador das linhas da tabela, no caso de mais de uma chave em uma tabela, é escolhida uma chave primária, desta forma, nenhum valor de chave primária pode ser nulo. Uma chave primária não tem nenhuma ligação com o conceito de ordenação e com o acesso à tabela. Para questões de acesso às informações a recomendação é a utilização de índices.

Documentação

Definição formal dos elementos (dicionário de dados), evitando assim, ambiguidade: falta de clareza, falta de precisão, incerteza, dúvida. Cada um dos elementos identificados e

representados deverá ser definido claramente para que, associando-se seu nome, sua representação e sua definição, sejamos capazes de ter o completo entendimento do conceito que estes procuram transmitir. A dicionarização deve trazer a conhecimento público toda e qualquer informação útil para o processo de compreensão e unificação de conceitos.

Normalização

É um processo formal, passo a passo, que examina os atributos de uma entidade, com objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de linhas específicas, tem como objetivos a preservação da integridade dos dados, gerar estabilidade para o modelo, eliminar redundância. Dados bem definidos, íntegros no seu significado, consistentes, confiáveis, seguros e compartilhados fazem com que cada novo sistema defina apenas os dados que são do seu escopo e compartilhe os demais dados com outros sistemas presentes na organização.

- **Primeira Forma Normal**: O objetivo é retirar os atributos ou grupos repetitivos. Representação de informações que se repetem para a mesma unidade, retratando ocorrências de um mesmo fato dentro de uma única entidade, vinculado a sua chave, onde para cada chave há a ocorrência de uma e somente uma informação de cada atributo. Desta forma, cada campo de uma tabela precisa conter somente um único tipo de dado, e cada parcela de dado deve ser armazenada em somente um lugar. Essa exigência é conhecida como atomicidade de dados.
- **Segunda Forma Normal**: O objetivo é separar as dependências parciais. É preciso que as tabelas estejam na primeira forma normal e que cada uma contenha dados sobre uma e somente uma entidade, onde as colunas que dependem parcialmente da PK, devem formar uma nova tabela, algumas entidades, para serem identificadas e individualizadas, necessitam conter em sua chave mais de um atributo, formando, portanto, uma chave concatenada, verificar se a mesma possui chave concatenada e, se for o caso, constatar se todos os tributos não chaves não apresentam dependência parcial com a referida chave. Isto é, quando os atributos não-chaves dependem parcialmente de chave concatenada.
- **Terceira Forma Normal:** O objetivo é eliminar dependências transitivas. Quando alguns atributos não são dependentes diretos da chave da entidade, mas sim por transitividade, através de outros residentes na mesma entidade referenciada. Isto é dependência indireta de um atributo com a chave da entidade, através de outro atributo não-chave, do qual é diretamente dependente. É preciso que as tabelas estejam na segunda forma normal e que todos os campos não-chaves dependam diretamente da chave primária, ou seja, não pode ter colunas determinadas por outras colunas. Os campos calculados devem ser eliminados, desta forma é verificado se algumas tabelas precisam ser divididas em partes, pois todas as tabelas devem conter informações sobre somente uma coisa.
- **Forma normal de Boyce-Codd (FNBC):** Uma coisa interessante é que ela foi proposta como uma forma mais simples que 3FN, porém mais rígida. Se uma relação está na FNBC, também está na 3FN. Sua definição diz o seguinte: uma relação está na FNBC se todo determinante é chave candidata. Uma tabela está na FNBC se e somente se estiver na 3FN e

todo atributo não chave depender funcionalmente diretamente da chave primária, ou seja, não há dependências entre atributos não chave. Porém nem toda tabela que está na 3FN é uma tabela BCNF

4 - Agregação

A agregação é um mecanismo de abstração onde criamos um novo conceito a partir dos componentes de uma relação. Usamos a agregação quando sentimos a necessidade de associar um relacionamento ao outro. A figura abaixo mostra um exemplo de uso de uma agregação.

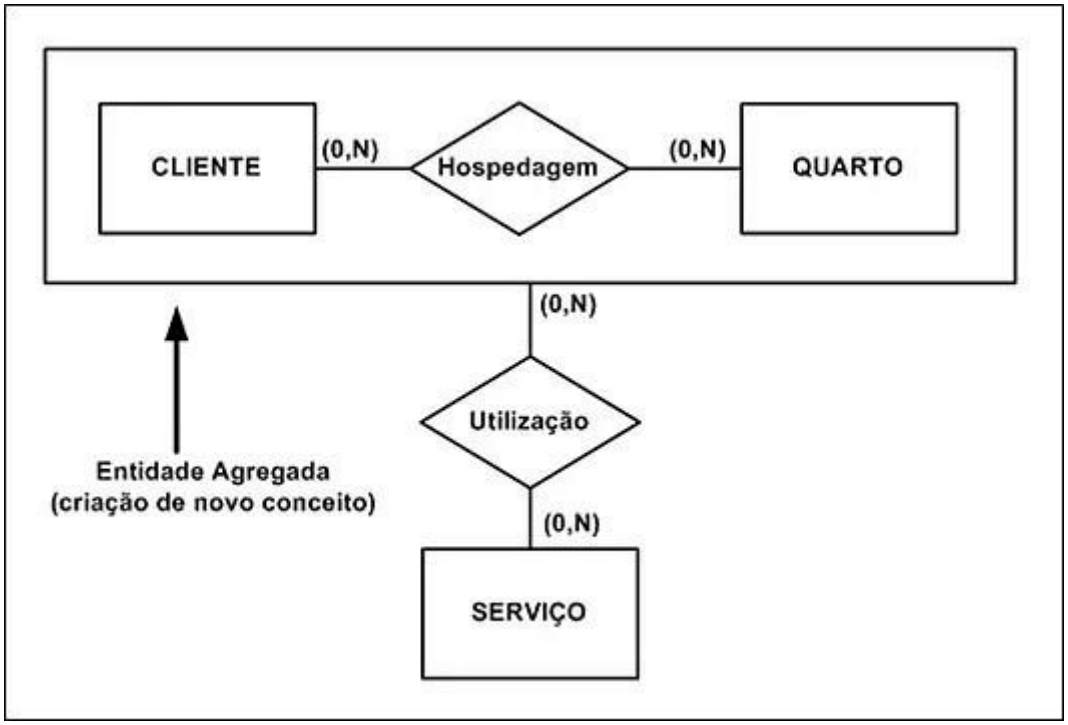


Figura 13 - Exemplo do uso da agregação em um MCD

No exemplo acima, foi necessário criar um conceito mais abrangente para poder representar o consumo de um cliente hospedado em um quarto. Vale ressaltar que, quando utilizamos a agregação, as relações são dependentes. Ou seja, a associação da entidade agregada com a outra entidade só ocorre após a existência do fato (relação) da entidade agregada. No caso, o cliente só poderá consumir serviços, após estar hospedado em um quarto.

Quais os próximos passos?

Conforme comentado no início deste estudo, a finalidade do modelo conceitual de dados é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Este trabalho pode ser concluído aqui, caso a demanda não esteja associada a um desenvolvimento de aplicação ou seguir adiante com o mapeamento do modelo conceitual de dados para o modelo lógico de dados.

Um modelo lógico de dados é considerado um modelo de dados implementável. Ao contrário do modelo conceitual que tem como principal objetivo representar o contexto do negócio, o modelo lógico de dados já procura estabelecer soluções de implementação em bancos de dados, claro que respeitando os requisitos de informação e regras de negócio representadas no modelo conceitual.

De forma geral, recomenda-se que o modelo lógico de dados seja criado a partir do mapeamento do modelo conceitual de dados e, a partir daí, começa-se um trabalho de refinamento para projetar o melhor modelo implementável. Um modelo lógico de dados também pode ser construído a partir do zero, porém toda a questão da participação do cliente/usuário no início da modelagem e a captura do ambiente de negócios poderão ser comprometidas, tornando a solução de modelagem vulnerável a erros de conceito e adequação ao negócio.

EXEMPLO PRÁTICO

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).

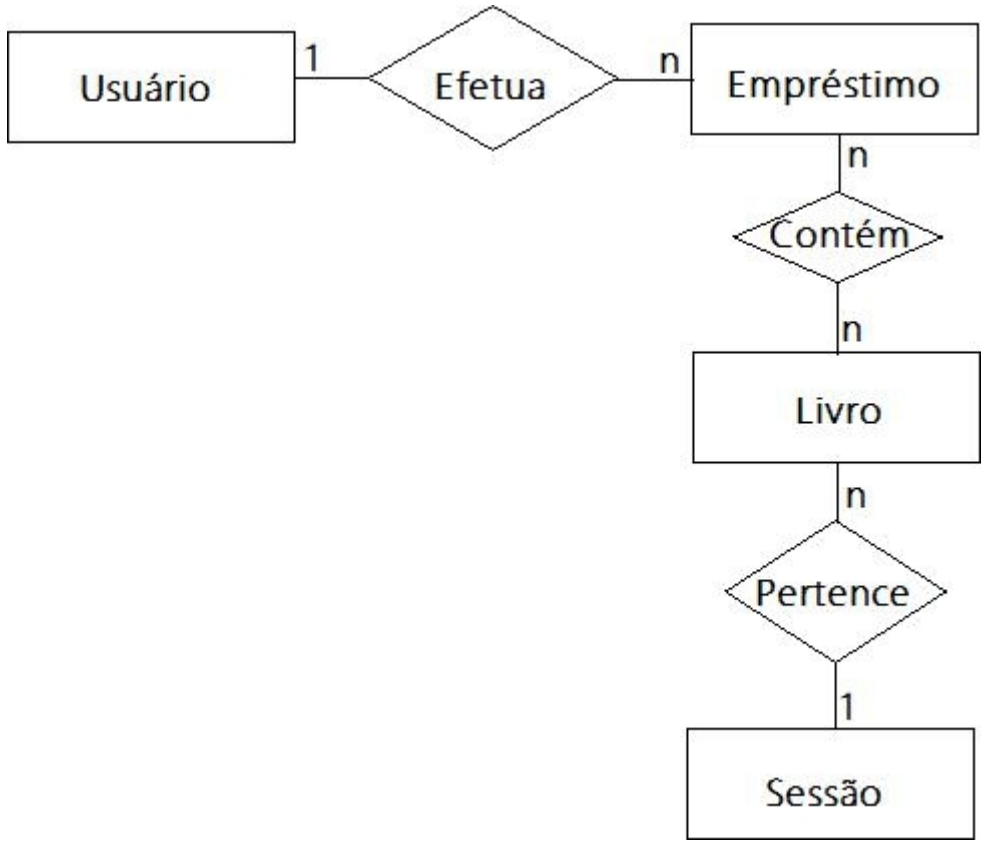


Figura 7. Primeiro DER de um sistema para biblioteca

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,

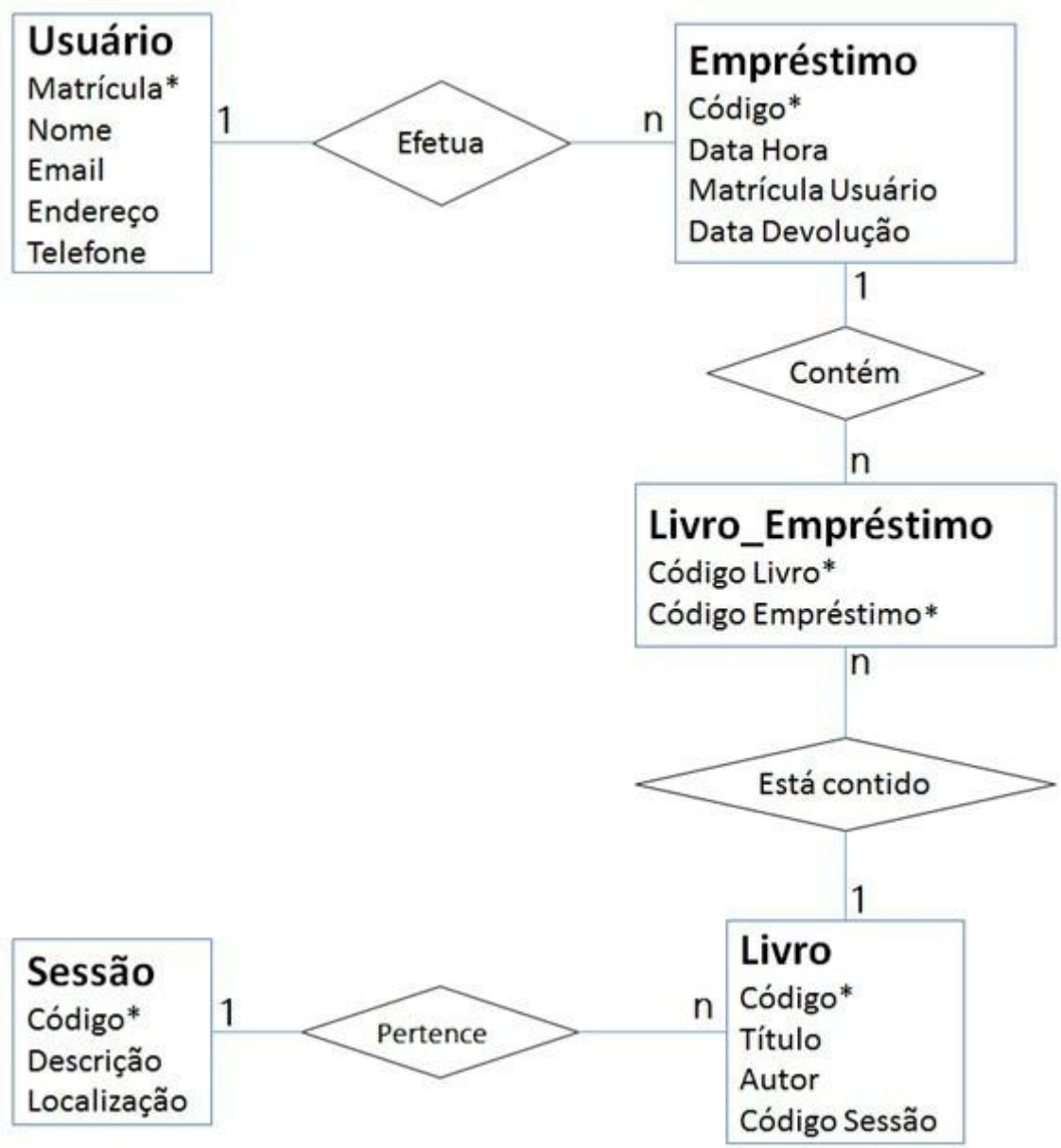


Figura 8. DER mais completo do sistema para bibliotecas

Neste ponto cabe fazer algumas observações importantes:

Especificamos os atributos de cada entidade e marcamos algumas elas 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.

DICIONÁRIO DE DADOS

No processo de análise de sistemas um dos pontos fortes é o MER – Modelo de Entidade e Relacionamento, onde são definidas as entidades que irão compor o sistema e como elas irão relacionar-se.

Junto com o modelo de entidade e relacionamento, é necessário que se mantenha um documento com a explicação de todos os objetos nele criados. Este documento, que pode ser chamado de d**icionário de dados**, permite que os analistas obtenham informações sobre todos os objetos do modelo de forma textual, contendo explicações por vezes difíceis de incluir no diagrama. É válido lembrar que o objetivo do documento é ser claro e consistente.

O modelo que iremos propor é apenas uma sugestão e em outras referências ao assunto é comum encontrarmos formas diferentes de criar e manter o **dicionário de dados**.

Entidade: Cliente				
Atributo	Classe	Domínio	Tamanho	Descrição
Codigo_cliente	Determinante	Numérico		
Nome	Simples	Texto	50	
Telefone	Multivalorado	Texto	50	Valores sem as máscaras de entrada
Cidade	Simples	Texto	50	
data_nascimento	Simples	Data		Formato dd/mm/aaaa

Analisando a tabela acima teremos:

•**Entidade:** é o nome da entidade que foi definida no MER. A entidade é uma pessoa, objeto ou lugar que será considerada como objeto pelo qual temos interesse em guardar informações a seu respeito.

•**Atributo:** Os atributos são as características da entidade Cliente que desejamos guardar.

•**Classe:** as classes podem ser: simples, composto, multivalorado e determinante. Simples indica um atributo normal. Composto indica que ele poderá ser dividido em outros atributos, como por exemplo, o endereço. Multivalorado é quando o valor do atributo poderá não ser único e determinante é um atributo que será usado como chave, como CPF, Código do cliente, etc.

•**Domínio:** podem ser numérico, texto, data e boleano. Podemos chamar também de tipo do valor que o atributo irá receber. A definição desses tipos deve seguir um processo lógico, exemplo: nome é texto, salário é numérico, data de nascimento é data e assim por diante.

•**Tamanho:** define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho é definido apenas para atributos de domínio texto.

•**Descrição:** é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista ou programador do sistema.

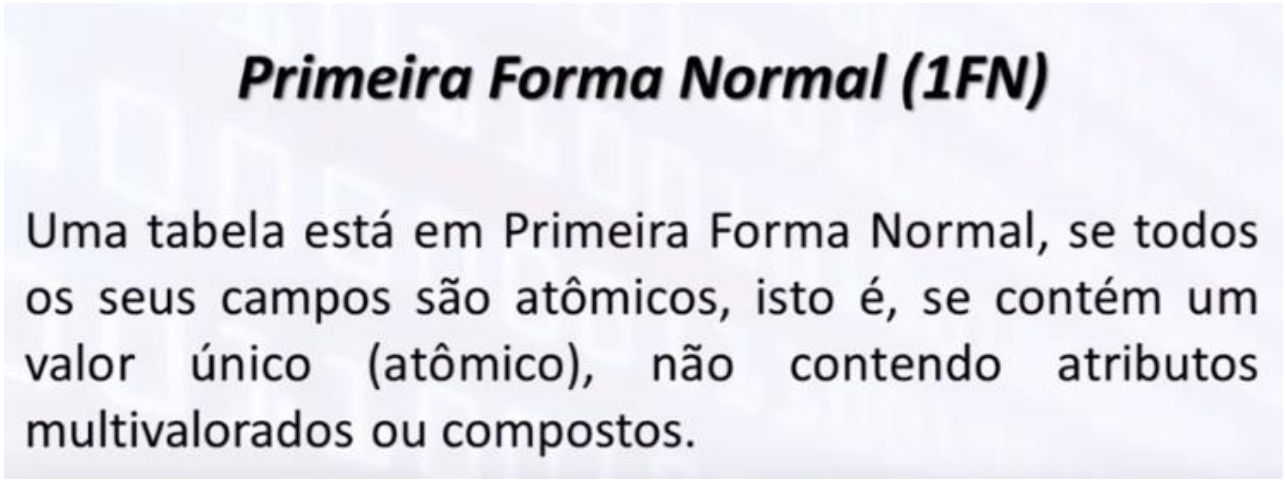
O exemplo acima é referente a uma entidade, então ele deverá ser replicado para as demais entidades do seu modelo.

Você poderá usar o Microsoft Excel para criar as tabelas do **dicionário de dados.** Não é nenhuma regra, mas ele é muito bom para este tipo de trabalho. Desta forma seu **dicionário de dados** poderá ficar concentrado em um único arquivo.

EXEMPLO PRÁTICO DE NORMALIZAÇÃO

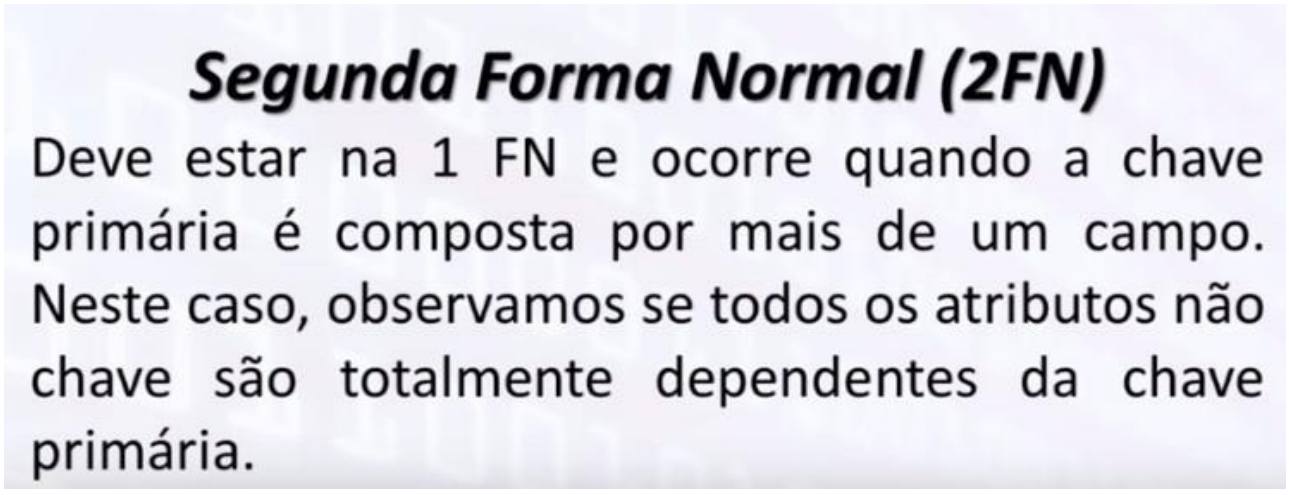
"A Normalização tem como objectivo avaliar a qualidade do Desenho de Tabelas e transformá- lo (em caso de necessidade) num Desenho (Conjunto de Tabelas) equivalente, menos redundante e mais estável."

Primeiro Formulário Normal



OBS: Uma relação está na 1ª Forma Normal (1NF) se e só se cada linha contém exatamente um valor para cada atributo. Neste caso diz-se que são campos atômicos. Isso quer dizer que não podemos ter campos multivalorados ou compostos

Segundo Formulário Normal

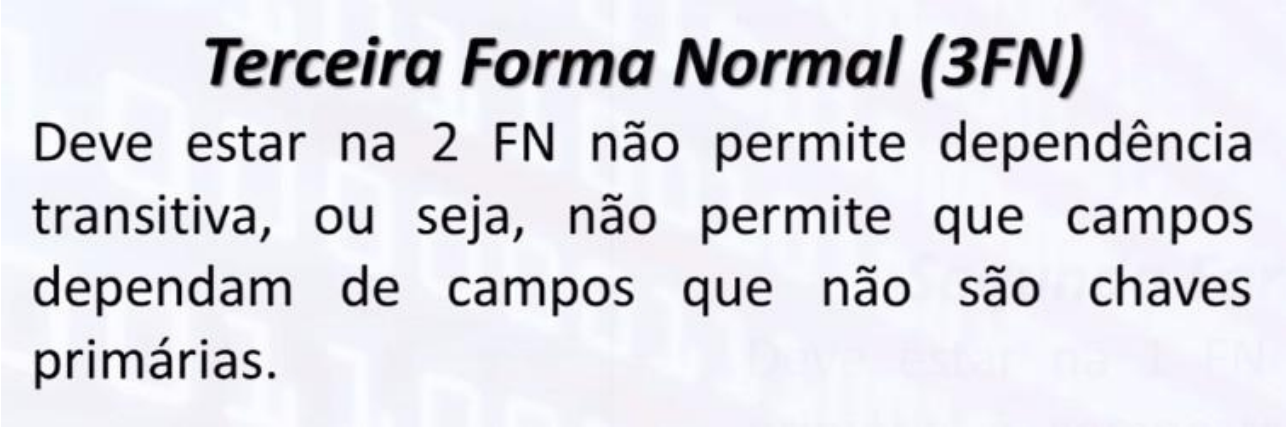


OBS: Não devem existir dependências funcionais ou parciais. Os procedimentos:

a) Identificar os atributos que não são funcionalmente dependentes de toda a chave primária;

b) Remover da entidade todos esses atributos identificados e criar uma nova entidade com eles.

Terceira Forma Normal



Forma normal de Boyce Cood

Uma tabela está na FNBC se e somente se estiver na 3FN e todo atributo não chave depender funcionalmente diretamente da chave primária, ou seja, não há dependências entre atributos não chave. Porém nem toda tabela que está na 3FN é uma tabela BCNF

Pontos a considerar:

- Quando uma chave possui mais de uma chave candidata, podem ocorrer anomalias;
- As chaves candidatas não possuem dependências parciais;

1) EXEMPLO PARA A 1FN

*CodCliente	Nome	Telefone	Endereco
001	José	9563-6352 9847-2501	Rua Seis, 85, Morumbi, CEP 0123-012
002	Maria	3265-8596	Rua Onze, 60, Moema, CEP 0321-123
003	Jânio	8545-8956 9598-6801	Praça Ramos, 100, Liberdade, CEP 1234-234

Exemplo de Tabela Desnormalizada

Passo 1: Dividir as tabelas para aplicar as regras da 1FN, para campos multivalorados (mais de um valor)

*CodTelefone	CodCliente#	Telefone
01	001	9563-6352
02	001	9847-2501
03	002	3265-8596
04	003	8545-8956
05	003	9598-6301
06	001	9999-8888

Passo 2: Separar em campos quando for um campo composto (várias partes dentro de um único campo).

*CodCli	Nome	Rua	N°	Bairro	CEP
001	José	Rua Seis	85	Morumbi	0123-012
002	Maria	Rua Onze	64	Moema	0321-123
003	Jânio	Rua Ramos	100	Liberdade	1234-234

2) EXEMPLO PARA 2FN

Uma suposta tabela de venda de produtos que teria uma chave composta pelo vendedor e o produto.

*NumVendedor	*CodProduto	NomeProduto	NomeVendedor	ValorUnit	Quantidade
137	1934	Imp. Laser	José	1500,00	6
137	1956	Imp. Deskjet	José	350,00	8
186	1923	Imp. Matricial	Marcos	190,00	2
361	1908	Imp. Mobile	Marcia	980,00	3
361	1934	Imp. Laser	Marcia	1500,00	6

Passo 1: Dividir os dados do produto e do vendedor, que não precisam estar na mesma tabela

*CodProduto	NomeProduto	ValorUnit
1934	Imp. Laser	1500,00
1956	Imp. Deskjet	350,00
1923	Imp. Matricial	190,00
1908	Imp. Mobile	980,00
1934	Imp. Laser	1500,00

*NumVendedor	NomeVendedor
137	José
137	José
186	Marcos
361	Marcia
361	Marcia

Dessa forma a tabela de vendas poderia fica da seguinte forma:

CodProd	CodVendedor	QtdProd

3) EXEMPLO PARA 3FN

Teremos nesse exemplo um campo da tabela que depende de outro campo que não é chave primária. Nesse caso mostrado aqui o campo NomeGerente depende do NumDepto que não é chave primaria nessa tabela.

*NumVendedor	NomeVendedor	Comissao	AnoAdm	NumDepto	NomeGerente
137	José	10	2010	73	Cleber
186	Marcos	15	2009	59	Elizabete
204	Clara	10	2010	73	Cleber
361	Marcia	20	2008	73	Cleber

Passo 1: separar os dados do cliente e deixar o seu relacionado por chave estrangeira com o departamento

*NumVendedor	NomeVendedor	Comissao	AnoAdm	#NumDepto
137	José	10	2010	73
186	Marcos	15	2009	59
204	Clara	10	2010	73

Passo 2: separar os dados do gerente e do departamento numa outra tabela.

*NumDepto	NomeGerente
73	Cleber
59	Elizabete

4) EXEMPLO PARA FNBC (Boyce Cood)

Considere a tabela abaixo, onde nesse exemplo não temos uma chave primária facilmente definida. Nesse caso temos como chave candidata a junção dos campos NrFornecedor e NrProduto. Outra possibilidade de uma chave primaria candidata seria a junção dos campos NomeFornecedor e NrProduto. O nome do produto não pode ser a chave primaria porque se repete várias vezes. A mesma coisa acontece com o campo NrProduto.

NrFornecedor	NomeFornecedor	NrProduto	QdtProd
123	Acme	001	600
134	Forg	001	200
134	Forg	002	251
123	Acme	003	250
123	Acme	001	360

Para normalizar uma tabela até a FNBC devemos decompor a tabela com os seguintes passos a seguir:

- Encontrar uma dependência funcional não-trivial $X \rightarrow Y$ que viole a condição de FNBC. X não deve ser uma superchave;
- Dividir a tabela em 2. Uma com os atributos XY, ou seja, todos os atributos da dependência;
- Outra com os atributos X, juntamente com os atributos restantes da tabela original.

Passo 1: separa a tabela de fornecedor, onde há a garantia de que cada linha é única. Tabela fornecedor

NrFornecedor	NomeFonrecedor
123	Acme
134	Forg

Passo 2: criar a tabela de relação entre fornecedor e produto. Nesse caso a sua chave primária vai ser a junção dos campos NrFornecedor e NrProduto. Cada linha vai ter uma informação única Tabela Forn_Prod

NrFornecedor	NrProduto	QtdProd
123	001	600
134	001	200
134	002	251
123	003	250
123	001	360

Referência

https://www.devmedia.com.br/modelagem-de-dados-conceitual-construindo-pontes-entre-dados-e-negocios/30597

http://www.blrdata.com.br/single-post/2016/03/19/Modelo-Conceitual-de-Dados-Aprenda-a-utilizar-os-principais-mecanismos-de-abstra%C3%A7%C3%A3o

https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332

https://www.luis.blog.br/dicionario-de-dados/

https://www.iseg.ulisboa.pt/aquila/getFile.do?fileId=19012&method=getFile

https://www.youtube.com/watch?v=W F-BKoROE

https://www.youtube.com/watch?v=o6mSiTO-vak

https://www.luis.blog.br/normalizacao-de-dados-e-as-formas-normais/