

Primeiro Resumo - Banco de Dados I



Aluno - Fabio Henrique Alves Fernandes | 19.1.4128



Professor Doutor - Guilherme Tavares De Assis



Instituição - Universidade Federal de Ouro Preto | Departamento de Computação



Data De Entrega - 14 de outubro de 2021

Introdução sobre Banco de Dados

Conceitos Básicos

Dados

Dados são fatos conhecidos com significados implícitos, que podem ser armazenados

Banco de Dados

Banco de Dados é uma coleção de dados relacionados, que tem algumas propriedades implícitas:

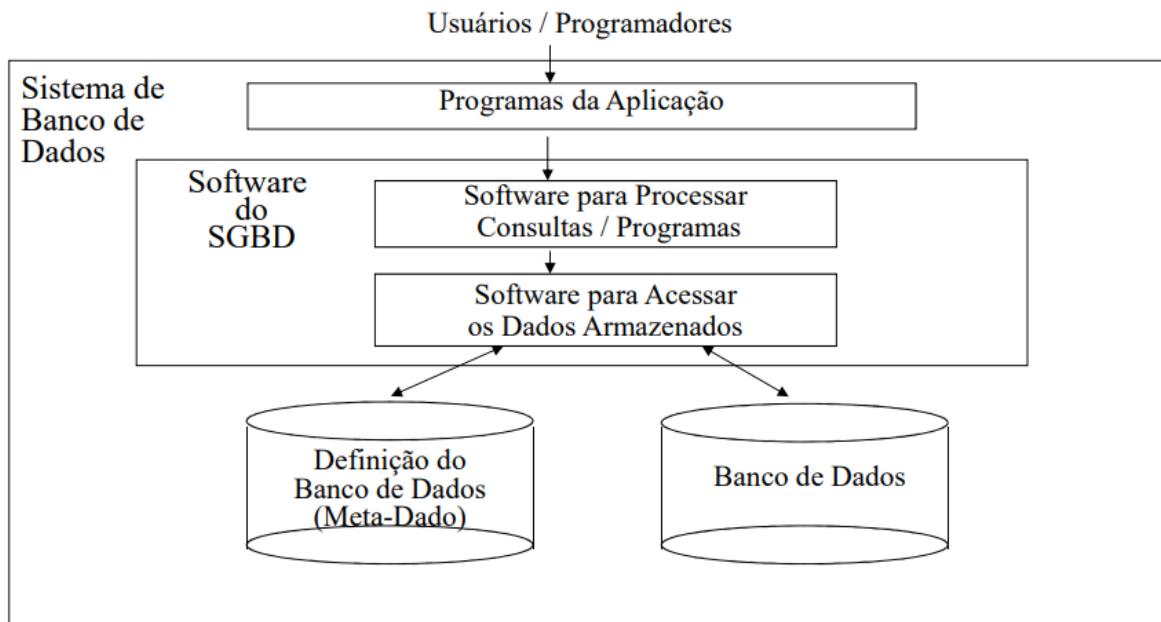
- Representa algum aspecto do mundo real
- Corresponde a uma coleção coerente de dados com significado inerente
- É projetado, implantado e povoados com dados para um propósito específico

Sistema de Gerência de Banco de Dados

Um SGBD é por onde criamos, mantemos e gerenciamos um banco de dados. É um software que, em um propósito geral, facilita a definição, a construção, e a manipulação.

Sistema de Banco de Dados

É um sistema de informação que manipula um banco de dados por meio de um SGBD.



Diferenças entre um Banco de Dados e o Sistema Tradicional de Arquivos

Em relação à independência entre programa e dados, em um sistema de Banco de Dados, os dados são armazenados em um Catálogo, sendo chamados de meta-dados, gerando independência entre programa e dados. Quando há orientação por objetos, as operações sobre os dados são definidas como parte do banco de dados, permitindo também a independência entre programa e dados. Já em um sistema tradicional de arquivos, a estrutura dos dados está embutida nos programas que realizam o acesso a eles, não havendo essa independência entre programas e dados.

Para a definição e manutenção de um banco de dados que será usada por vários usuários, em um SGBD multi-usuário, é permitido que vários usuários acessem o banco de dados ao mesmo tempo, já que o acesso simultâneo é essencial se os dados da aplicação são integrados e mantidos em um único banco de dados, e, é sempre necessário um software de controle de concorrência dentro no SGBD. Já no sistema tradicional, cada aplicação específica tem sua definição e implementação dos arquivos.

Usuários de um SGBD

Projetista

O Projetista é o usuário responsável por identificar os dados a serem armazenados e escolher a estrutura apropriada para representar e armazenar esses dados.

Administrador

O Administrador, como o próprio nome fala, é o usuário responsável por administrar o banco de dados e o SGBD. É ele que autoriza o acesso de usuários, coordenam e monitoram o uso do SGBD e verificam a necessidade de novos recursos de software e hardware para o bom funcionamento do SGBD.

Usuário Final

É o tipo de usuário que acessa o banco de dados para realizar consultas, promove modificações e gera relatórios. Podemos classificá-los como:

- Ingenuos ou parametrizados: o usuário que não tem tanto conhecimento, usando o banco de dados para a realização de consultas e modificações por meio de uma interface simples e própria.
- Casuais: usuários que raramente acessam o banco de dados, mas podem precisar de diferentes informações em cada acesso.
- Especializados: usuários familiarizados com as facilidades de um SGBD, realizando atividades complexas.

SGBD

O que deve fornecer?

Um bom SGBD deve fornecer ferramentas como:

- Controle de redundância: dados redundantes causam problemas como espaço em disco desperdiçado, repetição de uma alteração de lógica e inconsistência de dados;
- Restrição de acesso: em um banco de dados usado por vários usuários, nem todos podem ter acesso irrestrito a todos os dados inseridos;
- Suporte a múltiplas visões dos dados;
- Garantia das restrições de integridade: as restrições podem ser simples, como as relacionadas ao tipo de dado, ou complexas, como as que verificam se um item em um

arquivo está relacionado a itens em outro arquivo;

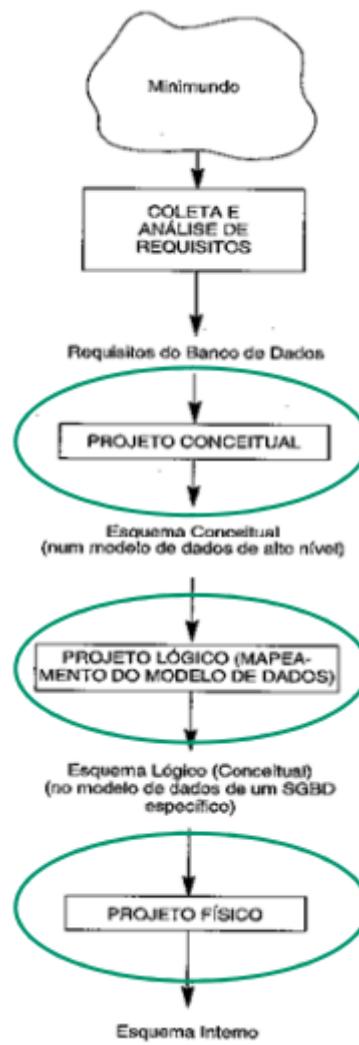
- Backup e recuperação: caso ocorra uma falha, seja de software ou de hardware, um SGBD deve fornecer facilidades para recuperar os dados, de for necessário, e gerar um banco de dados confiante e consistente;
- Múltiplas interfaces de usuários: os usuários do banco de dados especificados acima devem ter interfaces diferentes, para que a experiência seja mais apropriada para o uso;
- Armazenamento persistente para objetos e estruturas de dados de programas: um objeto é persistente se ele continua existindo após o término da execução de um programa e pode ser acessado diretamente por um outro programa;
- Inferência em banco de dados usando regras de dedução: os bancos de dados que possuem essa capacidade são chamados Bancos de Dados Dedutivos.

Quando não usar?

Para se usar um SGBD, existe um aumento de custo em relação ao sistema tradicional de arquivos que se deve a investimento inicial em software, hardware e treinamento; generalidade para definir e processar dados; fornecimento de mecanismos de integridade, segurança, controle de concorrência e recuperação a falhas.

Para usar o sistema de arquivos tradicionais, recomenda-se quando o banco de dados e as aplicações são simples, bem definidas e a expectativa de mudança é pequena; acesso de vários usuários aos dados não é requerido; requisitos de tempo-real de programas são necessários.

Processo metodológico para concepção de um banco de dados



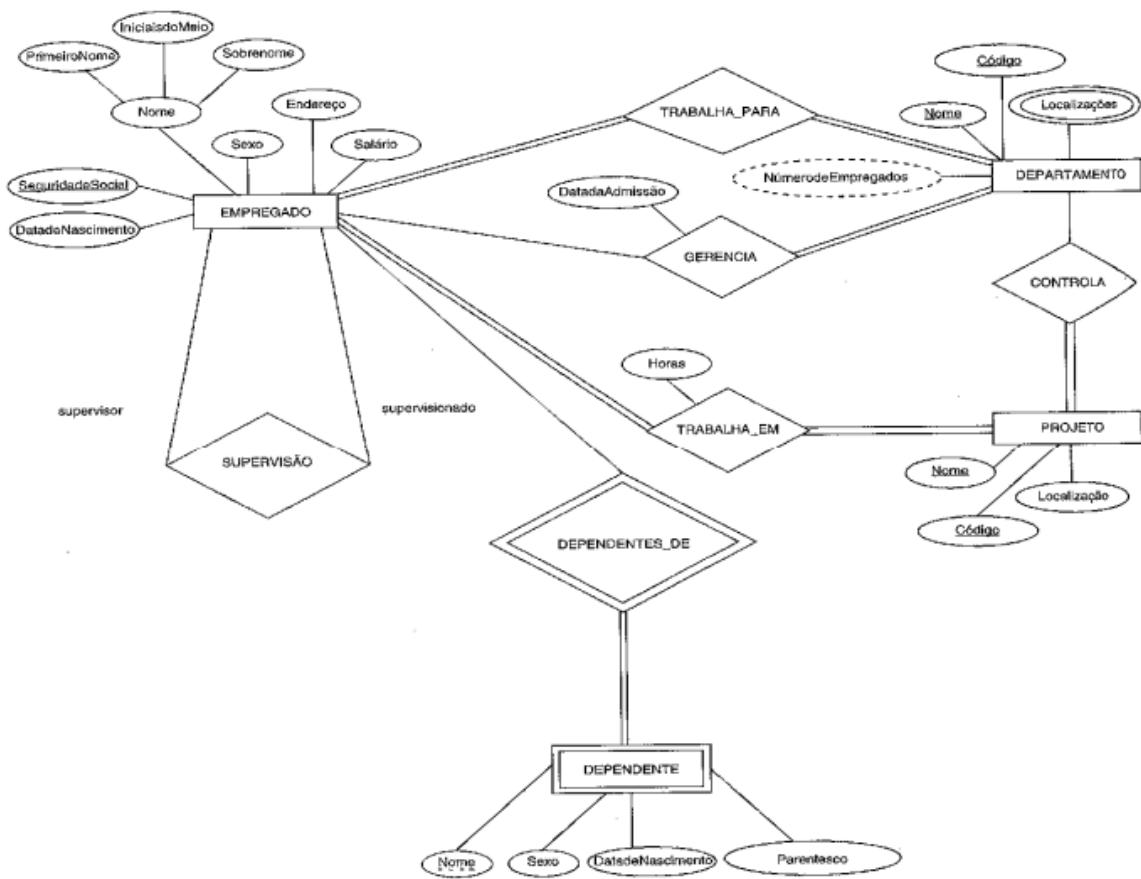
Modelo de Dados

Um modelo dados é um conjunto de conceitos usados para descrever a estrutura de um banco de dados. Pode ser caracterizado de acordo com os tipos de conceito que fornece para descrever a estrutura do banco, podendo ser conceitual, físico ou lógico.

Conceitual

É o modelo que fornece conceitos para descrever os dados como a maioria dos usuários percebe. Esses conceitos são:

- Entidade: os objetos e conceitos do mundo real;
- Relacionamento: a interação entre entidades;
- Atributo: propriedades de uma entidade ou de um relacionamento.



Exemplo de modelo ERE

Lógico

Um modelo que fornece conceitos que podem ser compreendidos pelos usuários, mas não estão muito longe de como os dados serão organizados. Representam os dados usando a estrutura de registro.

Físico

Um modelo que fornece conceitos para descrever os detalhes de como os dados são fisicamente armazenados no computador. Conceitos como:

- Formato dos registros;
- Ordenação dos registros;
- Vias de acesso.

Esquema

É a descrição textual ou gráfica de um banco de dados de acordo com o modelo de dados.
Quando na forma gráfica, é chamado de diagrama de esquema.

Estudante				
Nome	CódigoEstudante	Classe	DepartamentoPrincipal	
Disciplina				
Nome	CódigoDisciplina	Créditos	Departamento	
Turma				
CódigoTurma	CódigoDisciplina	Semestre	Ano	Professor
Histórico				
CódigoEstudante	IdentificadorTurma	Nota		
PréRequisito				
CódigoDisciplina	CódigoPréRequisito			

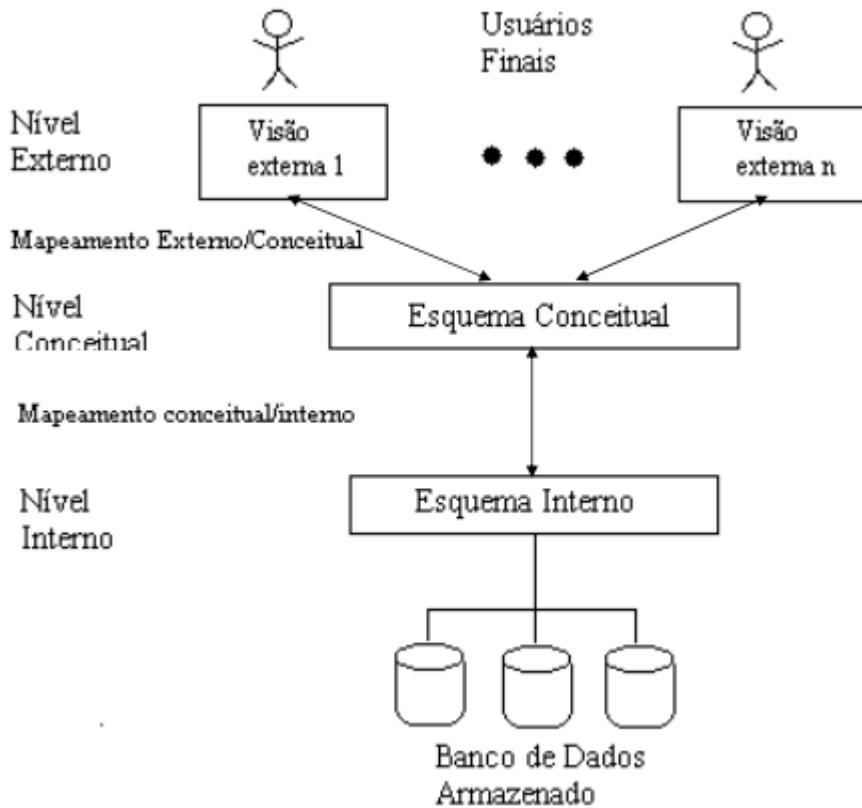
Instância

É o conjunto de dados armazenados em um banco de dados em um determinado instante de tempo, correspondendo ao estado do banco de dados. Os dados no banco de dados podem mudar a todo momento, mudando também a instancia do mesmo.

O SGBD deve garantir que toda instância do banco de dados corresponda a um estado válido. O estado inicial é obtido quando os primeiros dados são inseridos no banco de dados.

Estudante	Nome	CódigoEstudante	Classe	DepartamentoPrincipal	
	João	17	1	DCC	
	José	8	2	DCC	
Disciplina	Nome	CódigoDisciplina	Créditos	Departamento	
	Introdução à Ciência da Computação	DCC1310	4	DCC	
	Estrutura de Dados	DCC3320	4	DCC	
	Matemática Discreta	MAT2410	3	MAT	
	Banco de Dados	DCC3380	3	DCC	
Turma	CódigoTurma	CódigoDisciplina	Semestre	Ano	Professor
	85	MAT2410	II	91	King
	92	DCC1310	II	91	Anderson
	102	DCC3320	I	92	Knuth
	112	MAT2410	II	92	Chang
	119	DCC1310	II	92	Anderson
	135	DCC3380	II	92	Stone
Histórico	CódigoEstudante	CódigoTurma	Nota		
	17	112	B		
	17	119	C		
	8	85	A		
	8	92	A		
	8	102	B		
	8	135	A		
PréRequisito	CódigoDisciplina	CódigoPréRequisito			
	DCC3380	DCC3320			
	DCC3380	MAT2410			
	DCC3320	DCC1310			

Arquitetura de Três Níveis



O objetivo da arquitetura de três níveis é separar as aplicações dos usuários e o banco de dados implementado fisicamente.

- Nível interno: tem um esquema interno que descreve a estrutura física de armazenamento do banco de dados.
- Nível conceitual: tem um esquema conceitual que descreve a estrutura completa do banco de dados para os usuários.
- Nível externo: possui um conjunto de esquemas externos de usuários que descrevem partes do banco de dados que um grupo particular de usuários pode estar interessado.

Os três esquemas são apenas descrições de dados, o único local onde os dados realmente existem é no nível físico. No mapeamento entre os esquemas, o SGBD deve transformar as requisições para o esquema conceitual e as do esquema conceitual em requisições para o esquema interno para serem processadas sobre o banco de dados armazenado.

Independência de Dados

É a capacidade de mudar o esquema em um nível da arquitetura sem ter que mudar o esquema do nível acima.

- Independência de dados lógica: é a capacidade de mudar o esquema conceitual sem ter que mudar algum esquema externo ou programa de aplicação.
- Independência de dados física: é a capacidade de mudar o esquema interno sem ter que mudar o esquema conceitual ou algum esquema externo.

Linguagens de um SGBD

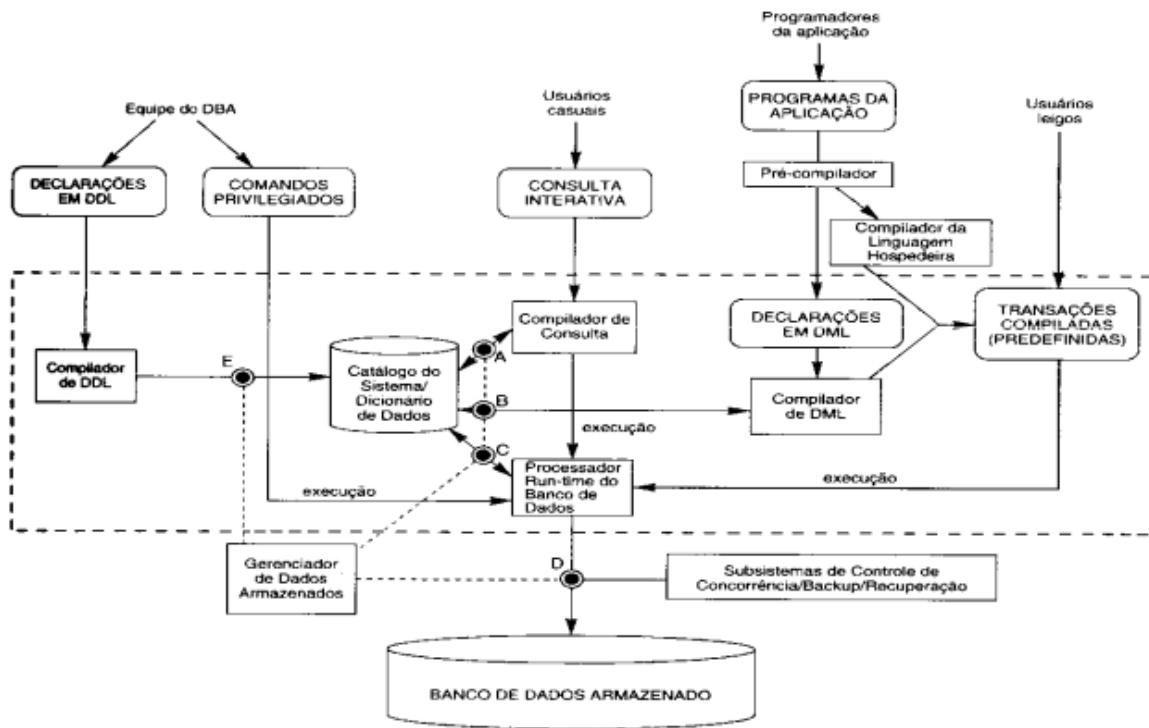
- Linguagem de Definição de Dados (LDD): especifica esquema conceitual;
- Linguagem de Definição de Armazenamento (LDA);
- Linguagem de Definição de Visões (LDV);
- Linguagem de Manipulação de Dados (LMD).

A linguagem de banco de dados relacional SQL representa uma combinação de LDD, LDA, LDV e LMD.

Interfaces de um SGBD

- Interfaces baseadas em menus para browsing;
- Interfaces baseadas em formulários;
- Interfaces gráficas;
- Interfaces de linguagem natural;
- Interfaces para o Administrador de Banco de Dados.

Módulos Componentes de um SGBD



Dentro de um SGBD temos módulos como:

- Gerenciador do banco de dados: controla o acesso aos dados armazenados no disco;
- Compilador de LDD: compila as definições de um esquema, armazenando-as no catálogo do SGBD;
- Processador do banco de dados em tempo de execução: recebe as operações de recuperação e modificação e as executa sobre o banco de dados;
- Compilador de consultas: compila as consultas de alto nível que são fornecidas interativamente;
- Pré-compilador: extrai comandos LMD de um programa de aplicação escritos em uma linguagem de programação hospedeira;
- Compilador da LMD: recebe e compila os comandos LMD extraídos pelo pré-compilador, gerando o código objeto referente.

Utilitários

- Carregador: é utilizado para povoar o banco de dados com arquivos de dados existentes;
- Backup: cria uma cópia de backup do banco de dados;
- Monitor de eficiência: monitora o uso do banco de dados e oferece estatísticas para o administrador do mesmo;

- Reorganizador de arquivos: é utilizado para reorganizar arquivos do banco de dados no intuito de melhorar a eficiência na execução de operações.

Critérios de Clássificação de SGBDs

Modelo de dados lógico

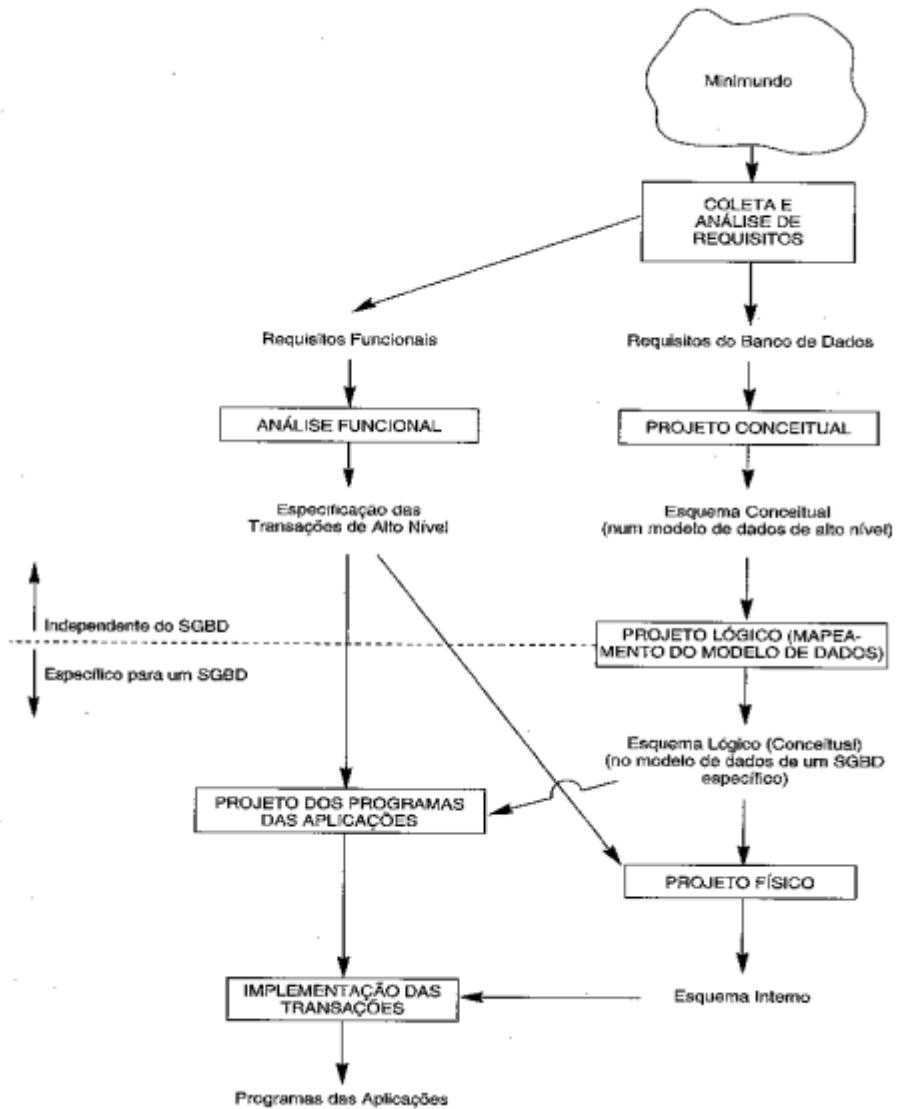
- Relacional;
- Hierárquico;
- Rede;
- Orientado a Objeto;
- Objeto-Relacional.

Número de servidores

- Centralizado: O banco de dados e o SGBD estão localizados no mesmo servidor.
 - Distribuído (SGBDD): o banco de dados e o SGBD estão distribuídos em vários servidores conectados por uma rede, podendo ser de forma homogênea (usando o mesmo SGBD), ou federado (necessitando de uma forma eficaz de conectar-se tudo).
-

Modelagem de Dados

Fases do Projeto de um Banco de Dados



Modelo ER

O Modelo de Entidades Relacionamento é um modelo conceitual para projetar aplicações de banco de dados. Baseia-se na percepção de mundo real como conjuntos de objetos básicos, aqui chamados de entidades e nos relacionamentos entre eles, sendo independente da implementação do banco em si.

Entidade

É um objeto existente no mundo real, distingível dos outros objetos.

Tipos de entidade definem uma coleção de entidades que possuem os mesmos atributos.

Um Conjunto de entidades é uma coleção das instâncias de entidades de um determinado tipo de entidade.

Atributo

É a propriedade de descrição das características de uma entidade. Os atributos podem ser:

- Simples: Indivisíveis;
- Compostos: Divisíveis em subpartes menores;
- Monovalorados: têm um único valor para determinada entidade;
- Multivalorados: têm um conjunto de valores;
- Armazenados: atributos que estão de fato armazenados no banco de dados;
- Derivados: atributos que são obtidos a partir de atributos armazenados, não precisando ser armazenados dentro do banco;
- Atributos complexos: atributos que podem ser compostos e multivalorados ao mesmo tempo;
- Valores nulos: quando uma entidade não pode ter valor aplicado, usamos o valor especial nulo. Também vale quando não se conhece o valor do atributo.

Cada atributo de um tipo de entidade está associado com um conjunto de valores, o qual especifica os valores que podem ser atribuídos àquele atributo para cada instância de entidade.

Chave

É um atributo no qual os valores são únicos para cada entidade. Pode ser formada por mais de um atributo, sendo a combinação dos valores única para cada instância de entidade.

Superchave

É qualquer conjunto de um ou mais atributos nos quais os valores são únicos para cada instância de entidade.

Chave Candidata

É uma superchave que não possui subconjuntos próprios de superchaves.

Chave Primária

É uma chave candidata escolhida como meio principal de identificação de entidades para o tipo de entidade em questão. As demais chaves candidatas são chamadas de chaves alternativas ou secundárias.

Relacionamento

É uma associação entre entidades que representa um fato no mundo real. O tipo de relacionamento sobre entidades define um conjunto de associações entre entidades desse tipo.

O grau de um tipo de relacionamento é o número de tipos de entidades participantes.

Tipos de relacionamento podem ter atributos, assim como os que estão presentes nas entidades relacionadas.

Quando o mesmo tipo de entidade participa mais de uma vez em um tipo de relacionamento desempenhando diferentes papéis temos um tipo de relacionamento recursivo.

Restrição de Cardinalidade

É o que especifica o número de instâncias de um relacionamento que uma instância de entidade pode participar. Para um relacionamento binário entre duas entidades, tem-se as relações de entidade:

1:1: uma entidade A está associada a no máximo uma de B e vice-e-versa;

1:N: uma entidade A está associada a varias outras de B, porém uma de A está associada a somente uma de B;

M:N: uma entidade A está associada a várias de B, assim como uma entidade B está associada a varias de A;

Restrição de participação

Especifica se a existência de uma entidade depende de sua associação a outra entidade por meio de um relacionamento. A participação pode ser:

- Total: todas as intâncias de entidade devem participar
- Parcial: as instâncias nem sempre participam

Restrição de Cardinalidade Mínima e Máxima

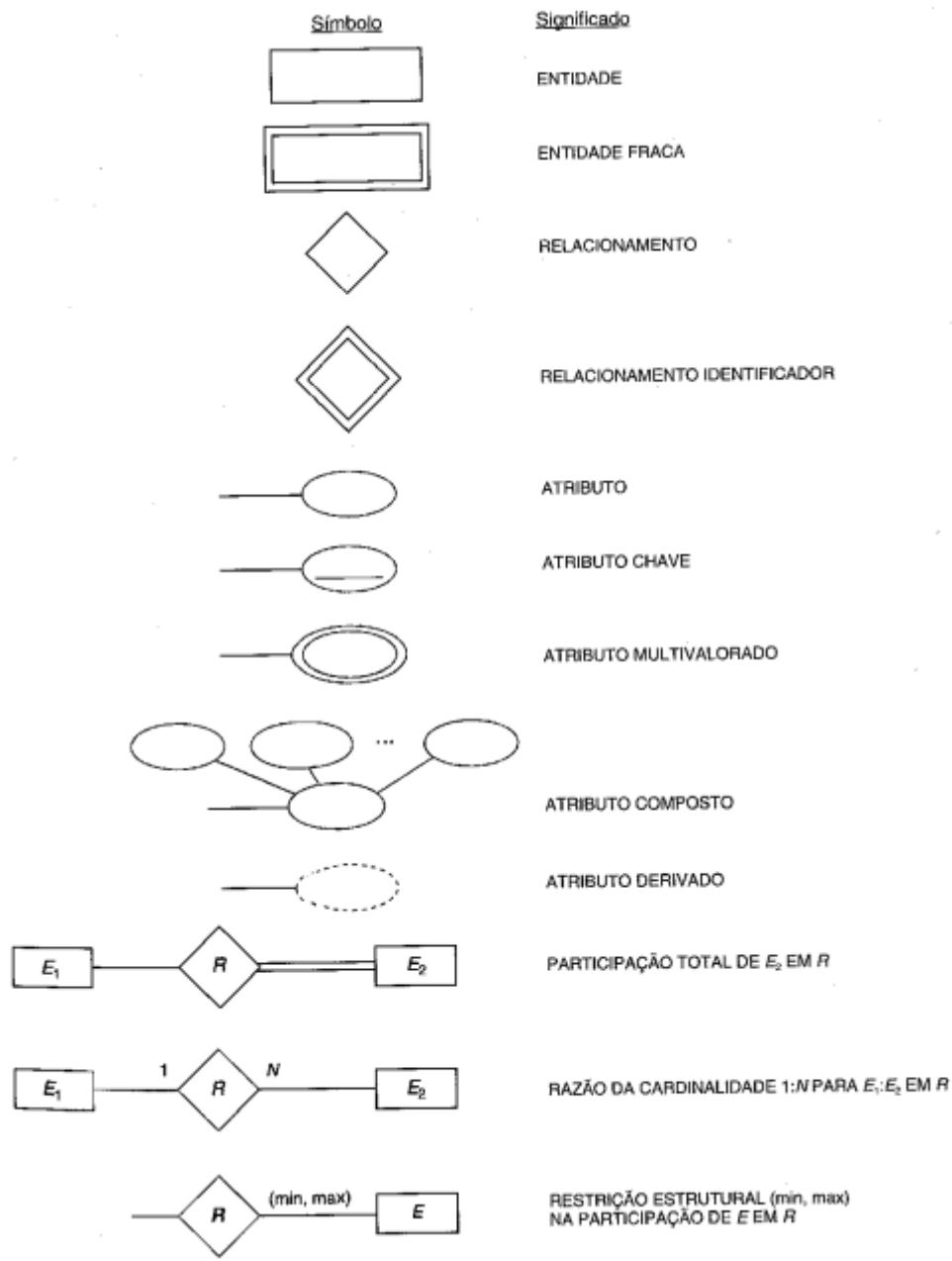
É uma restrição estrutural alternativa em relacionamentos no Diagrama ER. Envolve associar um par de números inteiros a cada participação de um tipo de entidade num tipo de relacionamento (em que $0 \leq \text{min} \leq \text{max}$ e $\text{max} \geq 0$). Em casa instância de entidade, ela deve participar de um mínimo min e um máximo max do relacionamento.

Entidade Fraca

É um tipo de entidade que não tem identificação própria. Está sempre associada a um tipo de entidade forte através de um relacionamento identificador. Sua chave é formada pela

combinação da chave primária da entidade forte com uma chave parcial própria.

Sempre tem uma restrição de participação total em relação ao relacionamento identificador, e é sempre representada no diagrama com contornos duplos, além de poder ser modelada como atributos complexos.



Elementos do Modelo ER

Herança

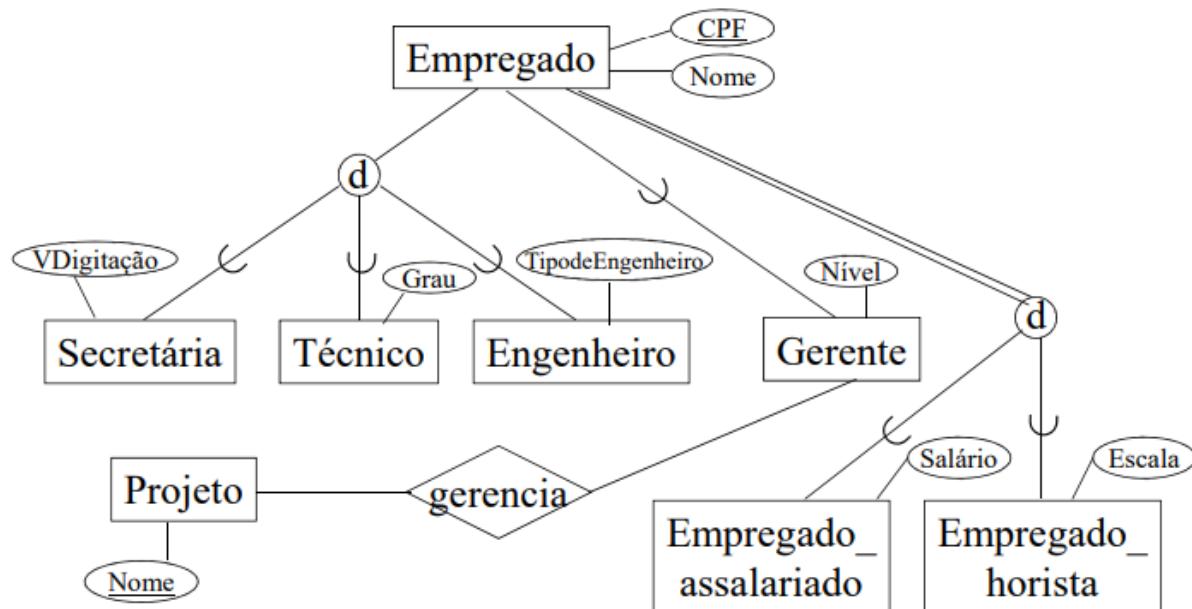
Em muitos casos, um tipo de entidade tem vários subconjuntos de entidades que são significativos para a aplicação. Cada um dos subconjuntos é chamado de subclasse do tipo

de entidade, por exemplo, Empregado, e o tipo de entidade de Empregado é chamado de superclasse. Uma instância de entidade não pode existir no banco de dados somente como membro de uma subclasse; ela deve também ser membro da superclasse em questão.

Não é necessário que toda entidade em uma superclasse seja membro de alguma subclasse. Como uma entidade na subclasse representa a mesma entidade no mundo real da superclasse, ela possui valores de seus atributos específicos bem como valores de seus atributos como um membro da superclasse. Logo, uma entidade que pertence a uma subclasse herda todos os atributos da superclasse. A entidade também herda todos os relacionamentos dos quais a superclasse participa. Também, pode ter seus próprios relacionamentos como subclasse.

Especialização

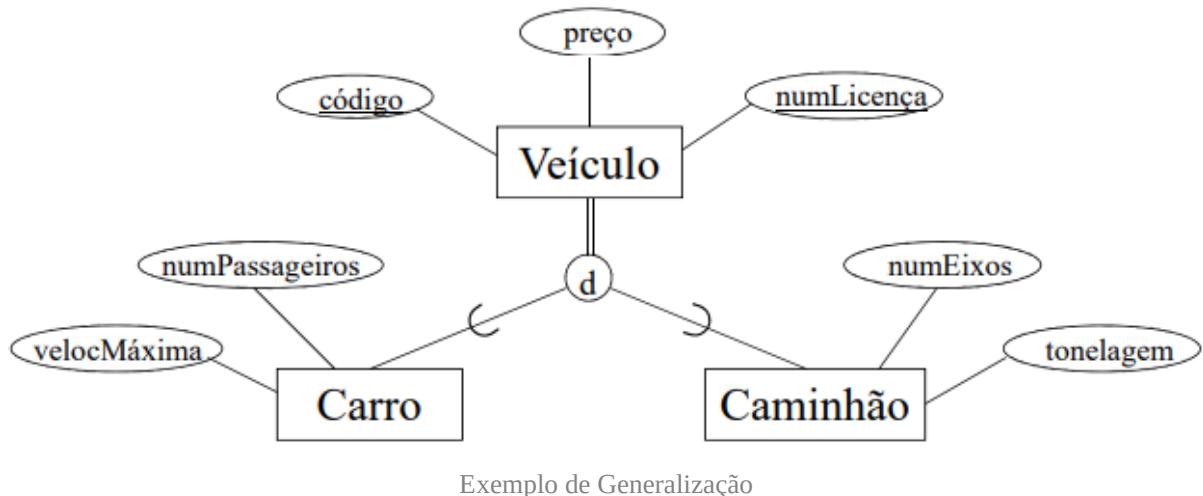
É o processo de definir um conjunto de subclasses a partir de alguma superclasse.



Exemplo de Especialização

Generalização

É o processo inverso da Especialização. As partes semelhantes de duas subclasses se juntam, e as partes diferentes ficam de fora, surgindo uma generalização em uma única superclasse.



Restrições sobre Especialização/Generalização

Restrição de Disjunção

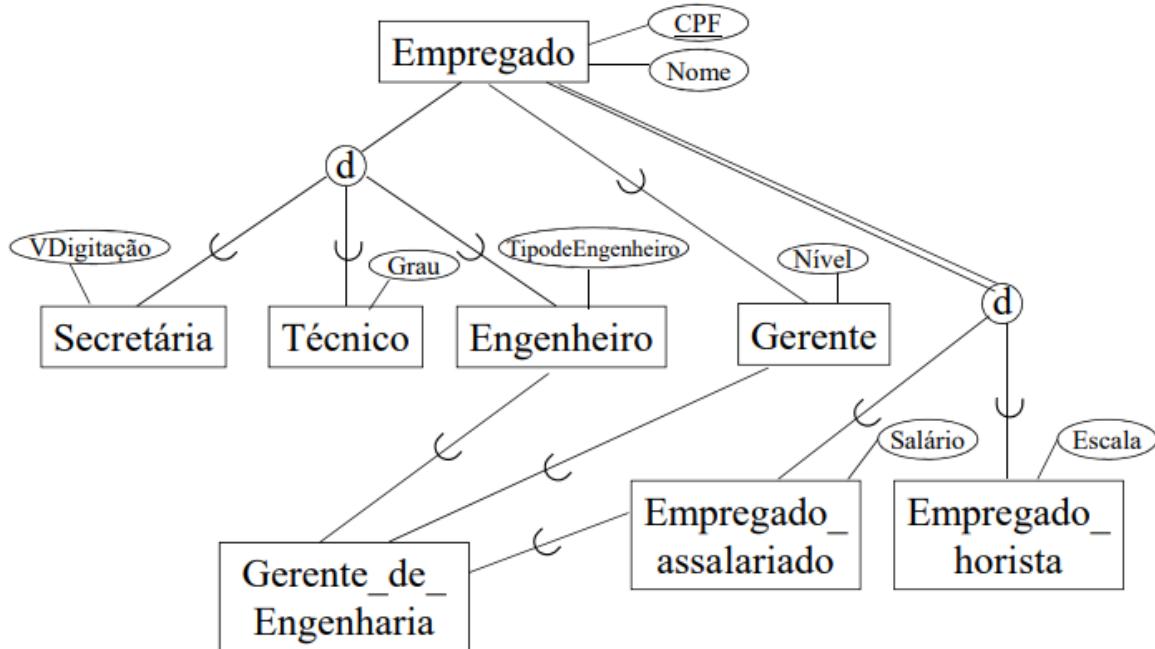
- Disjunção: uma entidade pode ser membro de, no máximo, uma das subclasses.
- Sobreposição: uma entidade pode ser superclasse de mais de uma subclasse.

Restrição de Completude

- Total: toda entidade da superclasse deve ser membro de no máximo uma subclasse.
- Parcial: nem toda entidade da superclasse precisa ser membro de alguma subclasse.

Herança Múltipla

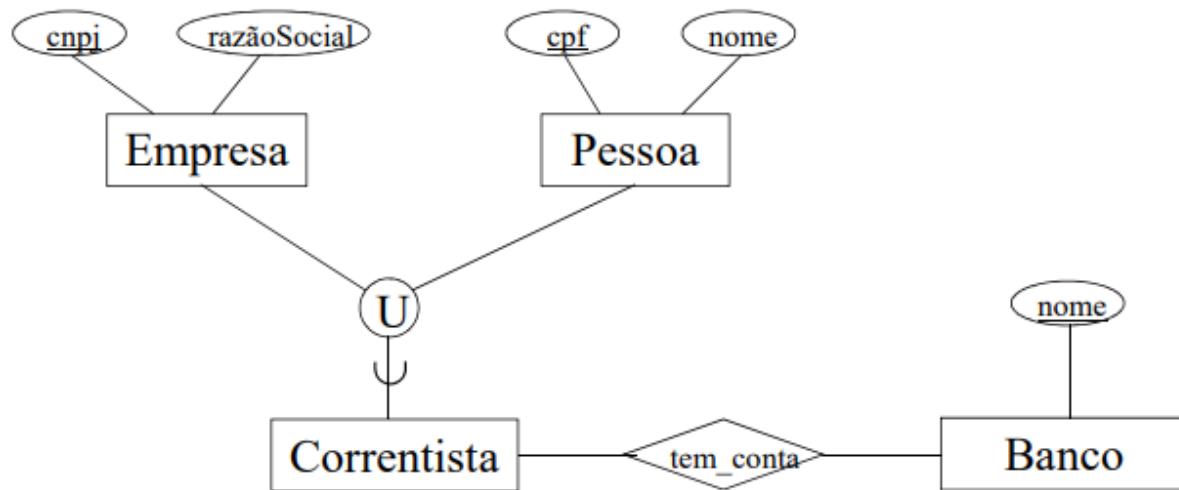
Uma subclasse pode ser descendente de mais de uma superclasse, criando uma herança múltipla.



Exemplo de Herança Múltipla

Tipo União ou Categoria

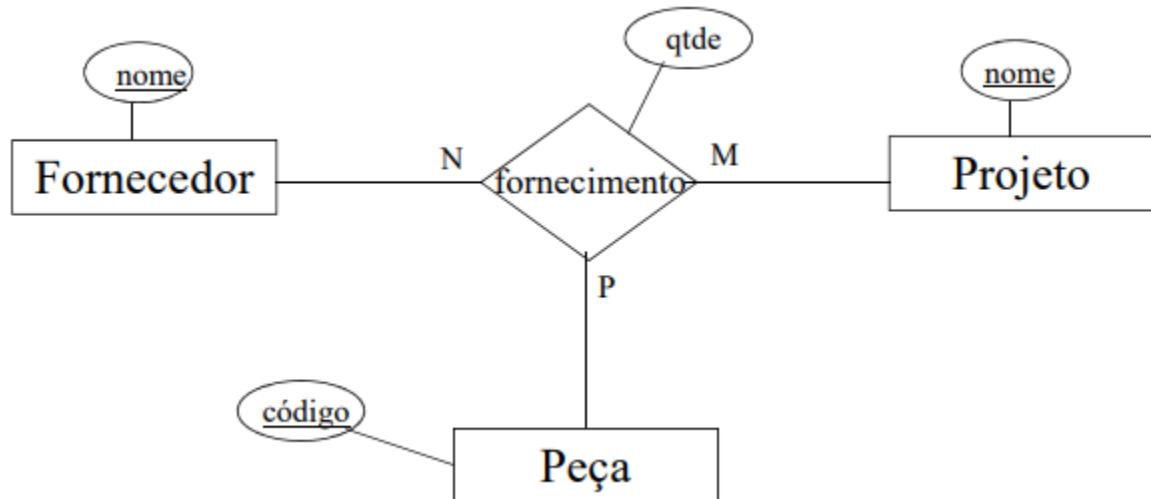
Em alguns momentos é necessário modelar um único relacionamento entre sub ou superclasses com mais de uma superclasse, onde as superclasses representam tipos diferentes de entidades e a subclasse representa uma coleção de objetos que é a união de instâncias das superclasses.



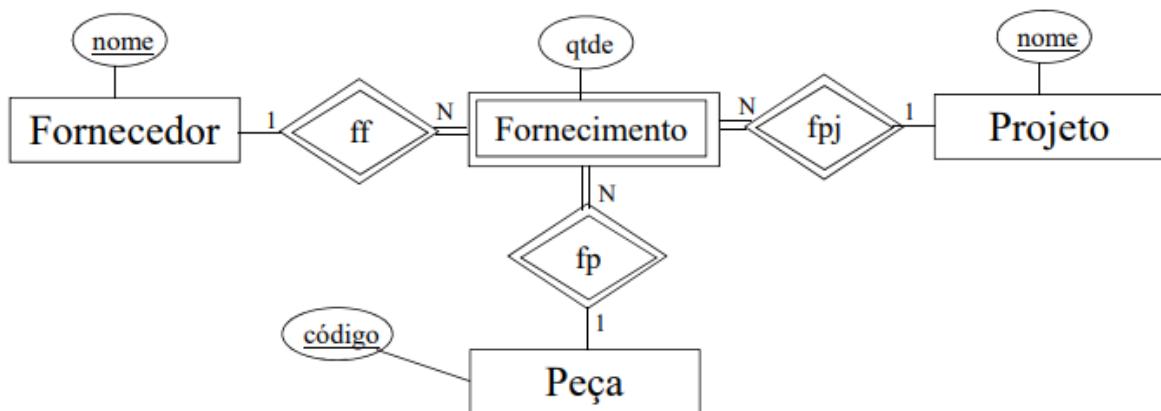
Exemplo de Tipo União ou Categoria

Relacionamento Ternário

É um relacionamento de grau 3, ou seja, tem 3 entidades sendo relacionadas.



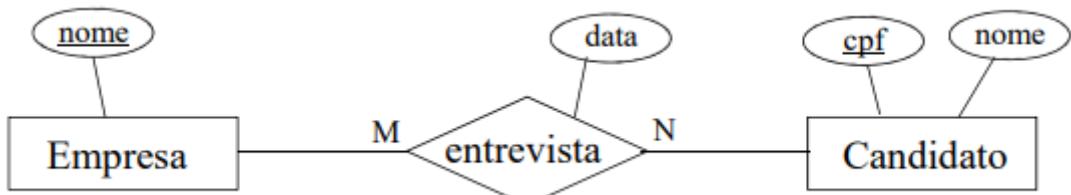
Exemplo Relacionamento Ternário



Exemplo como o do anterior, usando Relacionamento Binário e Entidade Fraca

Agregação

É uma abstração usada para construir objetos compostos a partir de seus componentes. De forma simples, ela é um tipo de relacionamento que deve ser transformado em um tipo de entidade.



Exemplo de Agregação