

AULA 2

Banco de Dados

Profa. Msc. Sandra Bozolan
Profa. Msc. Vanessa Lessa

Projeto de banco de dados e diagramas ER

- O projeto de banco de dados é, apenas, uma parte e está dividido em seis etapas durante a concepção de um projeto de banco de dados. O modelo ER é mais relevante nas três primeiras etapas.

Segundo Ramakrishnan & Gehrke (2011), essas etapas podem visualizadas a seguir:

Análise de requisitos

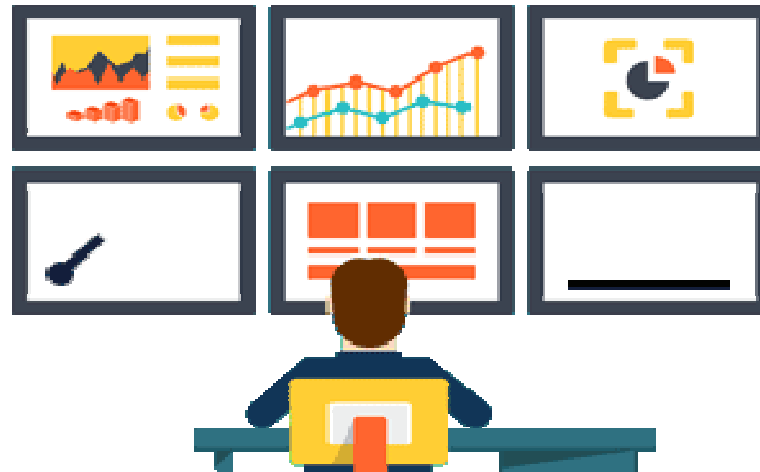
- Fase inicial de um projeto de banco de dados, que permite compreender quais dados devem ser armazenados, que informações os aplicativos devem manipular, quais as operações mais importantes e quais estão sujeitas a requisitos de desempenho. Reuniões informais como os usuários da aplicação, devem ser realizadas visando definir quais alterações serão trabalhadas, a análise de toda a documentação disponível sobre os aplicativos existentes que se deseja substituir ou complementar com o banco de dados, e assim por diante. Muitas metodologias são propostas com o foco de organizar e apresentar as informações coletadas, quase sempre se usa as ferramentas automatizadas.

Fonte:
<https://supremapropaganda.com.br/agencia-de-marketing-digital-publicidade-e-propaganda-em-itajai-sc/>



Projeto Conceitual de Banco de Dados

- Nesta etapa, as coletas de dados da fase de análise de requisitos serão utilizadas para fornecer uma descrição de dados a serem armazenados, levando em consideração as informações mais importantes. O modelo Entidade Relacionamento (ER) disponibiliza diversos modelos de dados semânticos ou de alto nível, utilizados no projeto de banco de dados. Tem como objetivo criar uma descrição simples dos dados que melhor corresponda à visão ou à ideia, que os usuários e desenvolvedores têm em relação aos dados (e às pessoas e aos processos a serem representados nos dados).



Fonte:
<https://onsafety.com.br/entenda-como-funciona-gestao-a-vista/>

Projeto Lógico de Banco de Dados

- A partir da conversão do projeto conceitual de banco de dados, um SGBD será escolhido para implementar as informações de um esquema ER para um esquema de banco de dados relacional.

As três etapas restantes do projeto de banco de dados:

- A partir do esquema lógico, novas funções devem receber atenção;
- Desempenho e protótipos dos esquemas físico;
- Aspectos de segurança, que assegurem que os usuários sejam capazes de acessar os dados de que eles precisam, mas não os dados que desejamos ocultar deles.

```
Categoria (@codCat: inteiro, nomeCat: literal[30])  
Produto (@codProd: inteiro, descProd: literal[50],  
        quantProd: real, precoProd: real, codCatProd: inteiro)
```

Refinamento do esquema

- Nesta etapa, o projeto de banco de dados consiste em analisar a coleção de relações em nosso esquema de banco de dados relacional, para identificar os problemas em potencial e refiná-los. Contrastando às etapas de análise de requisitos e o projeto conceitual, os quais são essenciais ao esquema de refinamento.

Categoria (@codCat: inteiro, nomeCat: literal[30])

Produto (@codProd: inteiro, descProd: literal[50],
quantProd: real, precoProd: real, codCatProd: inteiro)

Projeto físico de banco de dados

- Nesta etapa, todas as alterações devem, ainda, suportar e refinar maior projeto de banco de dados, para assegurar que este satisfaça os critérios de desempenho desejados. Criação de índices em algumas tabelas e agrupamento de tabelas serão inicializados nesta fase, e pode envolver um reprojeito substancial de partes do esquema de banco de dados obtido das etapas anteriores de projeto.

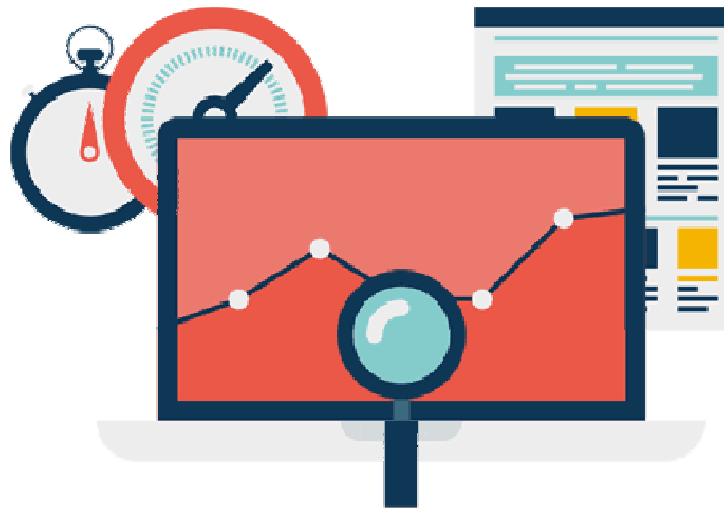
```
--criação da tabela TB_PRODUTO
CREATE TABLE TB_PRODUTO (
  codigo INTEGER PRIMARY KEY,
  quantidade REAL,
  preco REAL,
  descricao VARCHAR(30),
  codigocat INTEGER
);

--criação da tabela TB_CATEGORIA
CREATE TABLE TB_CATEGORIA (
  codigocat INTEGER PRIMARY KEY,
  descricao VARCHAR(30)
);

ALTER TABLE TB_PRODUTO ADD FOREIGN KEY(codigocat) REFERENCES
TB_CATEGORIA (codigocat);
```

Projeto de aplicativos e segurança

- A partir desta etapa novas ferramentas e metodologias podem ser implementadas ao SGBD, e deve-se identificar as entidades (por exemplo: usuários, grupos de usuários, departamentos) e os processos envolvidos no aplicativo. Descrever o papel de cada entidade em cada processo que é refletido em alguma tarefa de aplicativo, como parte de um fluxo dessa tarefa, identificando as partes do banco de dados que devem ser acessíveis e as partes que não devem ser acessíveis, criando as regras de segurança.



Fonte:
<https://www.pinterest.com/pin/856880266584216530/>

Entidades

- Uma entidade é um objeto do mundo real, que pode ser descrita por qualquer pessoa, seja ela concreta ou abstrata. Exemplos incluem: um carro popular, a concessionária, o gerente da loja, as filiais que vendem o mesmo modelo do veículo. Normalmente, é útil identificar uma coleção de entidades semelhantes.
- Uma coleção é chamada de conjunto de entidades. Um conjunto de entidade não precisa ser disjunto: o conjunto de funcionários da concessionária e o conjunto de funcionários da manutenção de veículos, ambos podem conter o funcionário João José da Silva (que pode trabalhar em ambos os departamentos).



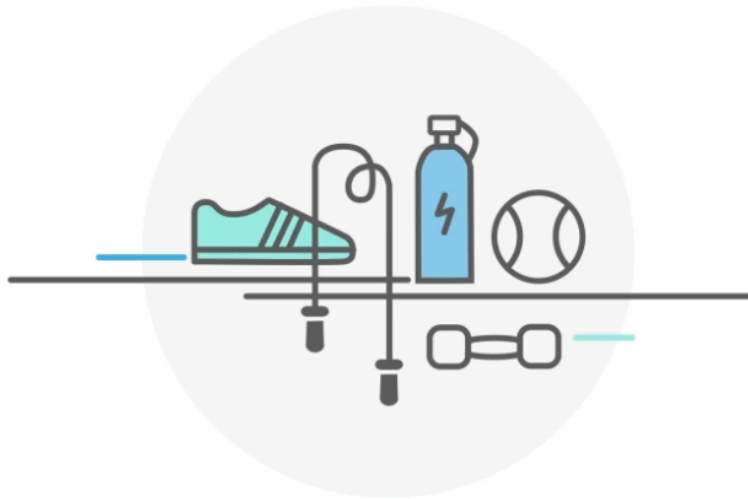
Fonte:
<https://dribbble.com/shots/5958438-Agora-Place-Animated-Illustration>



Fonte: <https://dribbble.com/shots/2216022-Team-Collaboration-Illustration>

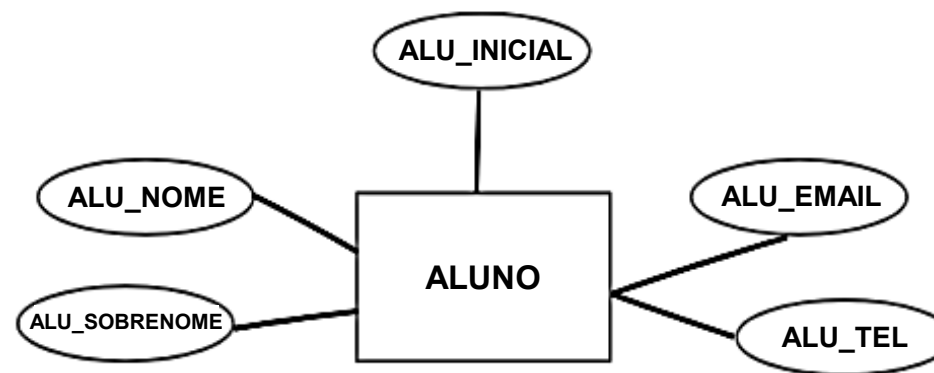
Atributos

- Um atributo é um valor de dado assumido pelos objetos de uma classe.
- Exemplo de atributos da entidade/Classe Aluno: nome, idade e peso são exemplos de atributos. Cor, peso e modelo são possíveis atributos de entidade/Classe Carro. Cada atributo tem um valor para cada instância de objeto.



Atributos

- Na notação de Chen original, os atributos são representados por elipses e conectados ao retângulo da entidade por uma reta.
- Cada elipse contém o nome dos atributos que representa.
- Na notação pé de galinha, os atributos são escritos na caixa de atributos, abaixo do retângulo da entidade.



ALUNO	
	ALU_SOBRENOME
	ALU_NOME
	ALU_INICIAL
	ALU_EMAIL
	ALU_TEL

Fonte: BOB, 2011, p. 116.

Atributos

- **Atributo necessário:** aquele que deve apresentar valor.
- **Atributo opcional:** aquele que não exige um valor.
- Domínio: conjunto de valores possíveis de determinado atributo.
- Os atributos podem compartilhar um domínio.
- Identificadores: um ou mais atributos que identifiquem, de modo exclusivo, cada instância de entidade.
- Identificadores **compostos**: composto de um único atributo.

Atributos

- **Identificador composto** pode ser subdividido.
- **Atributo simples** não pode ser subdividido.
- **Atributo** pode ter, apenas, um valor.
- **Atributos multivalorados** possuem muitos valores.
- Exemplo: telefone; uma pessoa poderá ter mais de um número de telefone.

Relacionamentos

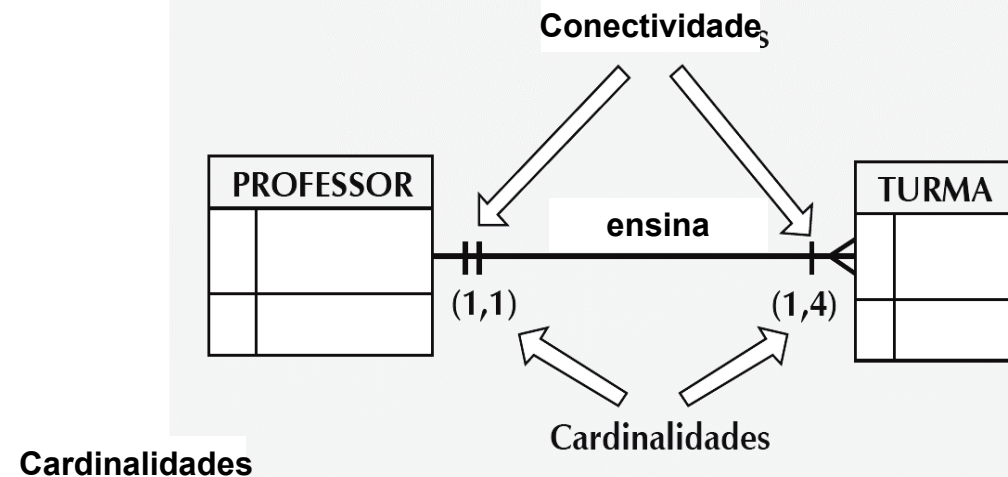
- Associação entre duas ou mais entidades.
- **Participantes:** as entidades que participam de um relacionamento.
- Os relacionamentos entre as entidades sempre operam em ambas as direções.
- Um relacionamento pode ser classificado como 1:M.
 - A classificação torna-se difícil de estabelecer quando, apenas, um lado do relacionamento é conhecido.

Conectividade e Cardinalidade

- O termo “**conectividade**” é utilizado para descrever a classificação dos relacionamentos.
- A **cardinalidade** expressa o número mínimo e máximo de ocorrências de entidades associadas a uma única ocorrência da entidade relacionada.
- São estabelecidas por afirmações muito concisas, conhecidas como regras de negócio.

FIGURA
4.7

Conectividade e cardinalidade
em um DER



Força de relacionamento

Relacionamento fraco (não identificado):

- Ocorre quando a PK da entidade relacionada não contém um componente da PK da entidade pai.

Relacionamento forte (de identificação):

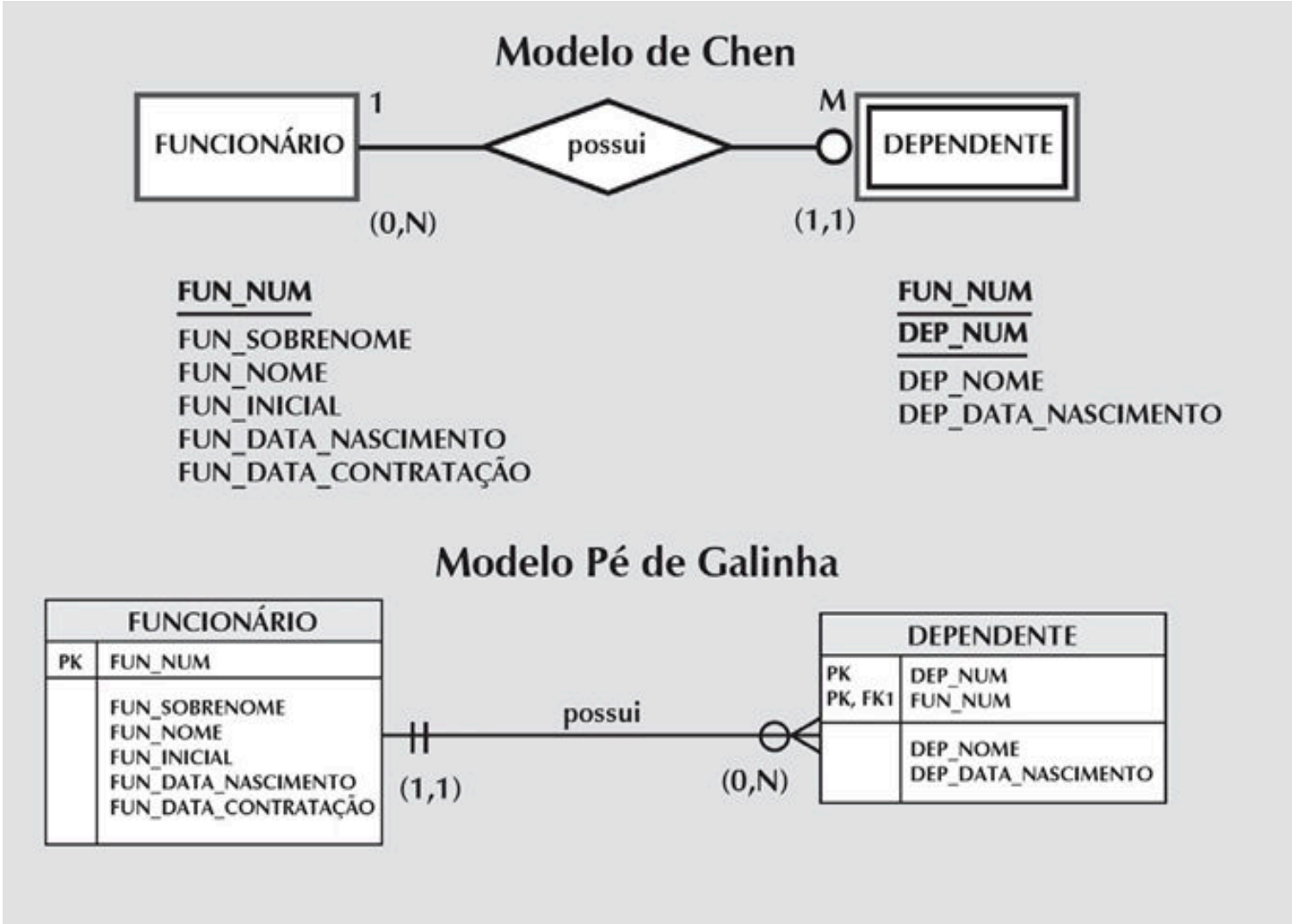
- Ocorre quando a PK da entidade relacionada contém um componente de PK da entidade pai.

Entidade fraca

Entidade fraca é aquela que atende a duas condições:

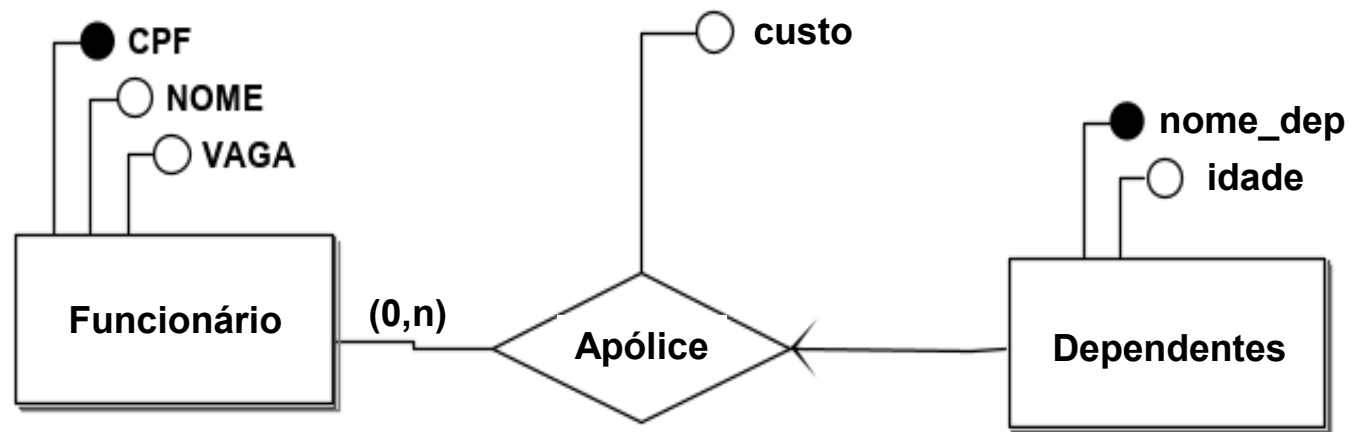
- É dependente de existência;
- A entidade possui uma chave primária que é parcial ou, totalmente, derivada da entidade pai do relacionamento;
- O projetista do banco de dados, normalmente, determina se uma entidade pode ser descrita como fraca, com base nas regras de negócio.

Entidade fraca em um DER



Entidade fraca – Dependente

- O conjunto de entidades fracas Dependentes e o seu relacionamento com funcionários são ilustrados a seguir. A participação total de Dependentes em Apólice é indicada pela ligação entre eles com uma linha grossa.
- A seta de Dependentes a Apólice indica que cada entidade de Dependentes aparece em, no máximo, um (exatamente um, na realidade, em razão à restrição de participação) relacionamento Apólice.
- Para ressaltar o fato de que Dependentes é uma entidade fraca e a Apólice é seu relacionamento identificador. O atributo *nome_dep* é uma chave parcial de Dependentes.



Entidade fraca e opcional em um DER

FIGURA 4.13 TURMA opcional para DISCIPLINA

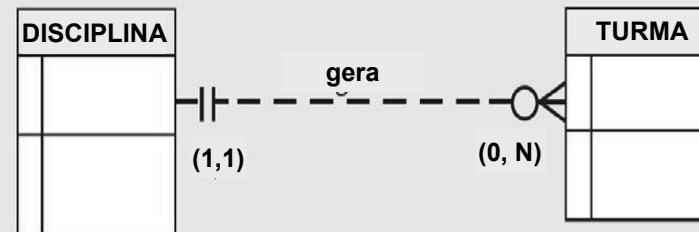
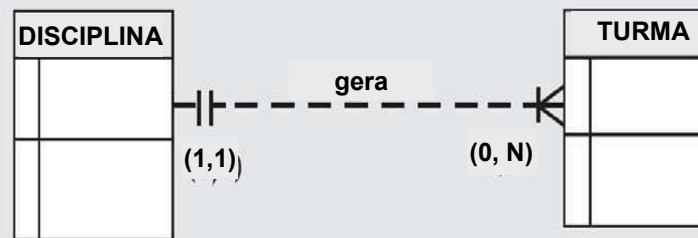
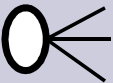





FIGURA 4.14 DISCIPLINA e TURMA em relacionamento obrigatório

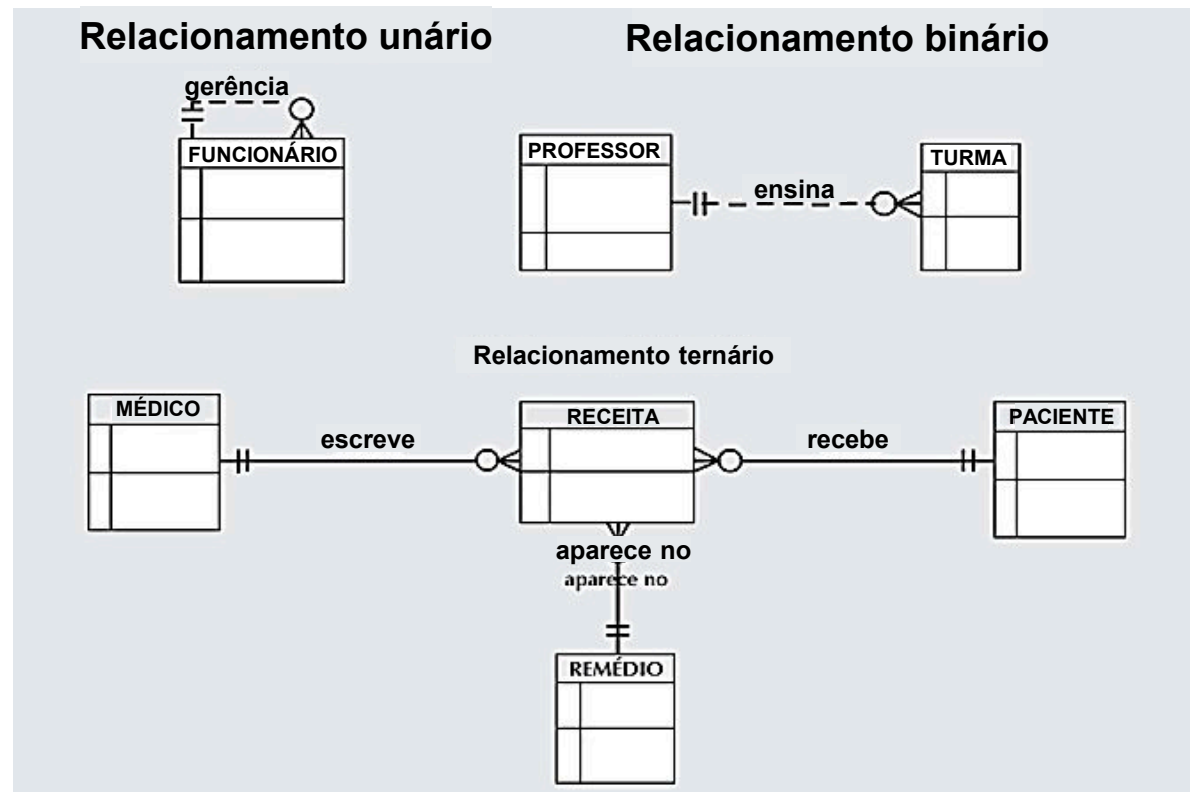


Simbologias

Notação Pé de galinha	Cardinalidade	Comentários
	(0,N)	Zero ou muitos. O lado “muitos” é opcional.
	(1,N)	Um ou muitos. O lado “muitos” é obrigatório.
	(1,1)	Um e somente um. O lado “1” é obrigatório.
	(0,1)	Zero ou um. O lado “1” é opcional.

Três tipos de graus de relacionamento

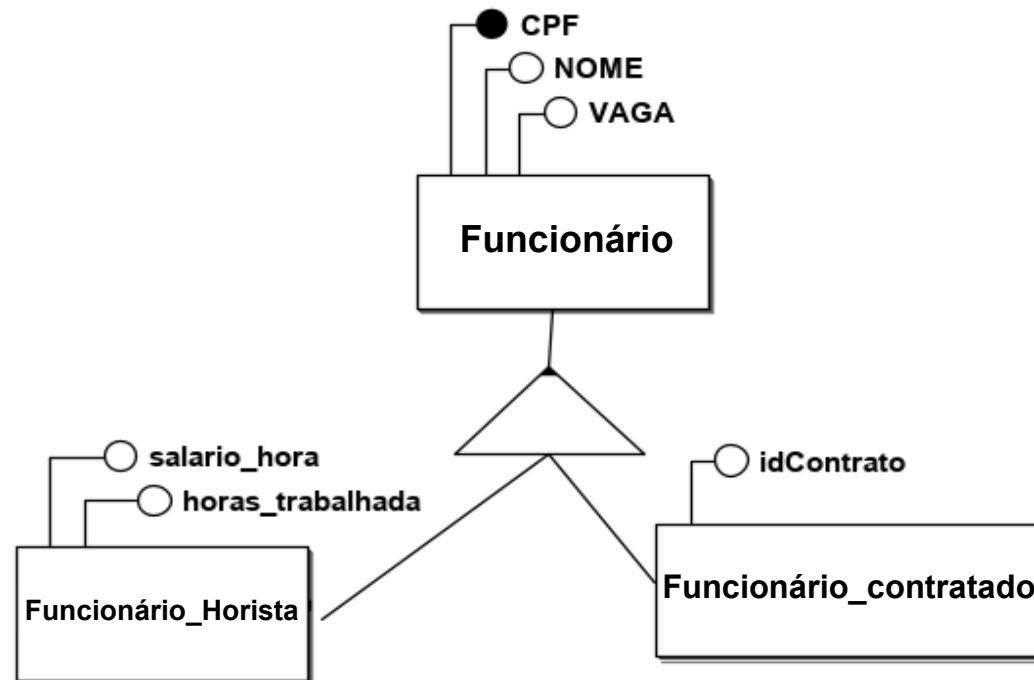
- Indica o número de entidades ou participantes associados a um relacionamento.
- Relacionamento unário: ocorre quando uma associação é mantida em uma única entidade.
- Relacionamento binário: se dá quando duas entidades estão associadas.
- Relacionamento ternário: quando três entidades estão associadas.



Hierarquias de classes

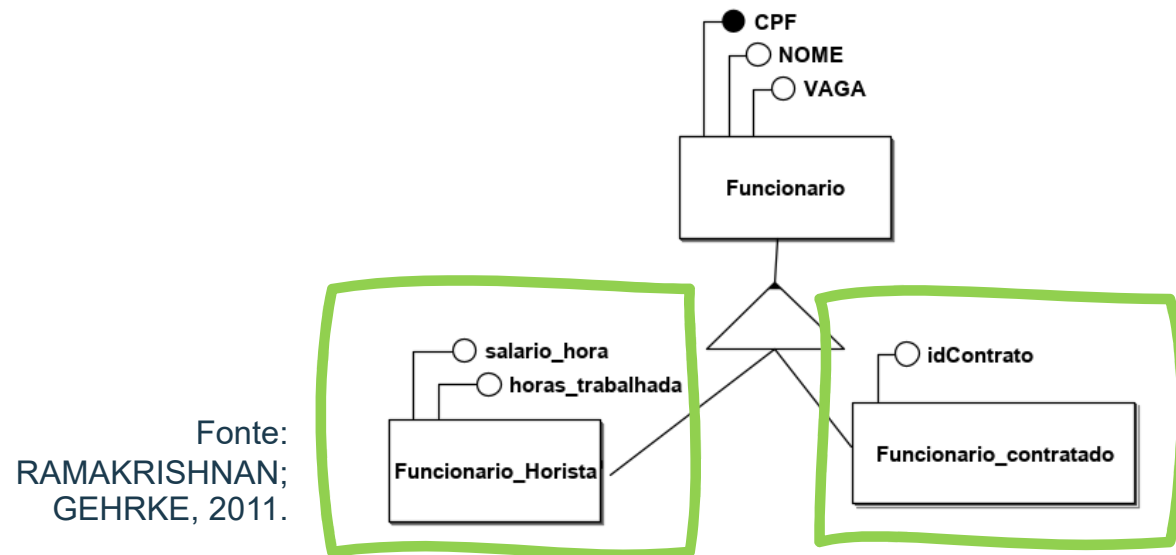
- Classificar as entidades de um conjunto de entidades em subclasses é uma atividade muito comum. Por exemplo, poderíamos considerar um conjunto de entidades Funcionário_Horistas e um conjunto de entidades Funcionário_Contratados para distinguir a forma como os funcionários são pagos. Poderíamos ter atributos horas_trabalhadas e salário_hora definidos para Funcionário_Horistas e um atributo id-contrato definido para Funcionário_Contratados.

Fonte: RAMAKRISHNAN; GEHRKE, 2011.



Hierarquias de classes – Especialização

- A superclasse de Funcionários é dividida em duas classes especializadas. A especialização é o processo de identificar subconjuntos de um conjunto de entidades (a superclasse) que compartilham algumas características distinguíveis. Tipicamente, a superclasse é definida primeiro, as subclasses são definidas em seguida, e os atributos específicos da subclasse e os conjuntos de relacionamentos são adicionados depois.
- Funcionário_Horistas e Funcionário_Contratados são generalizados em Funcionários.



Hierarquias de classes

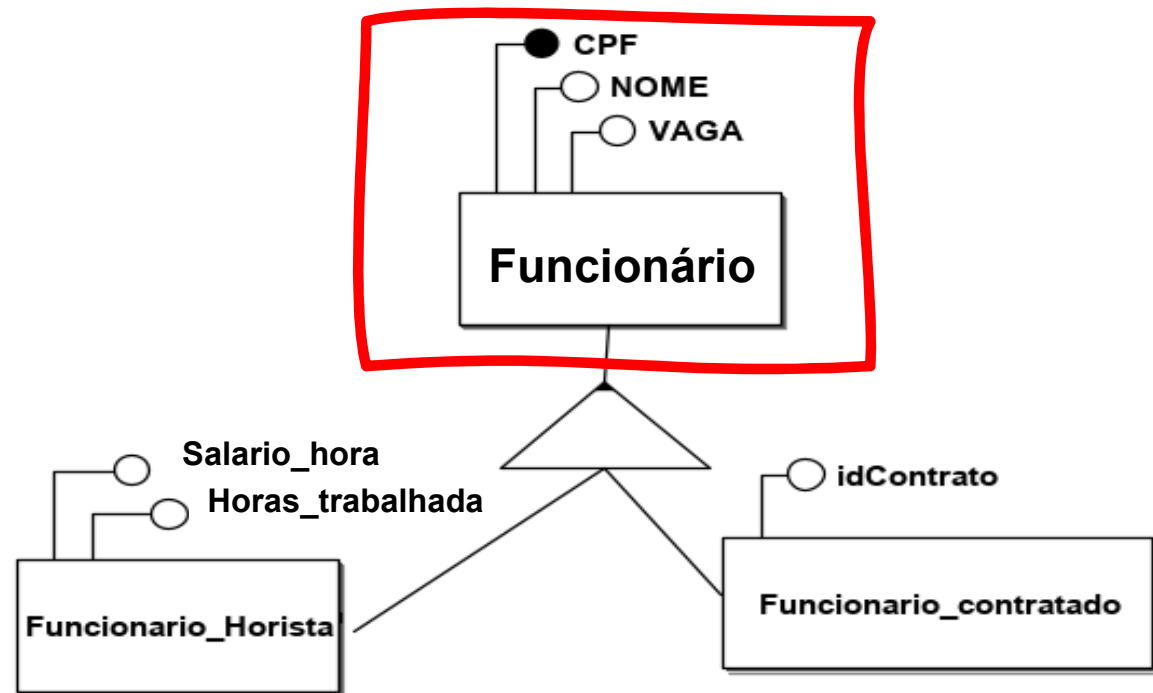
- Toda entidade deve possuir uma semântica em que um desses conjuntos seja, também, uma entidade de Funcionários e, como tal, deva possuir todos os atributos definidos para os Funcionários.

As subclasses podem ser identificadas em duas subclasses (por especialização ou generalização):

- Acrescentando a descrição dos atributos que façam sentido, apenas, às entidades em uma subclasse. Por exemplo, salário_hora não faz sentido para uma entidade Funcionário_Contratados, cujo pagamento é determinado por um contrato individual.

Hierarquias de classes – Generalização

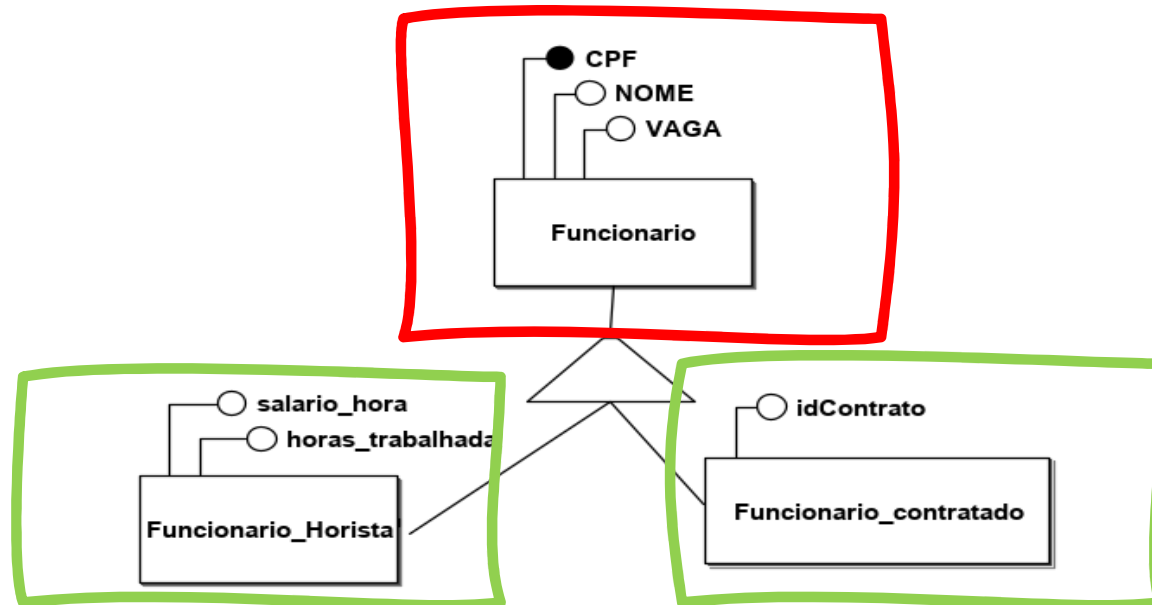
- Generalização consiste em identificar quais são as características comuns de uma coleção de conjuntos de entidades e criar um novo conjunto de entidades que contenha as entidades possuindo essas características comuns. É recorrente que as subclasses são definidas primeiro, a superclasse é definida em seguida e quaisquer conjuntos de relacionamentos, que envolvam a superclasse, são definidos depois.



Hierarquias de classes – Especialização/Generalização

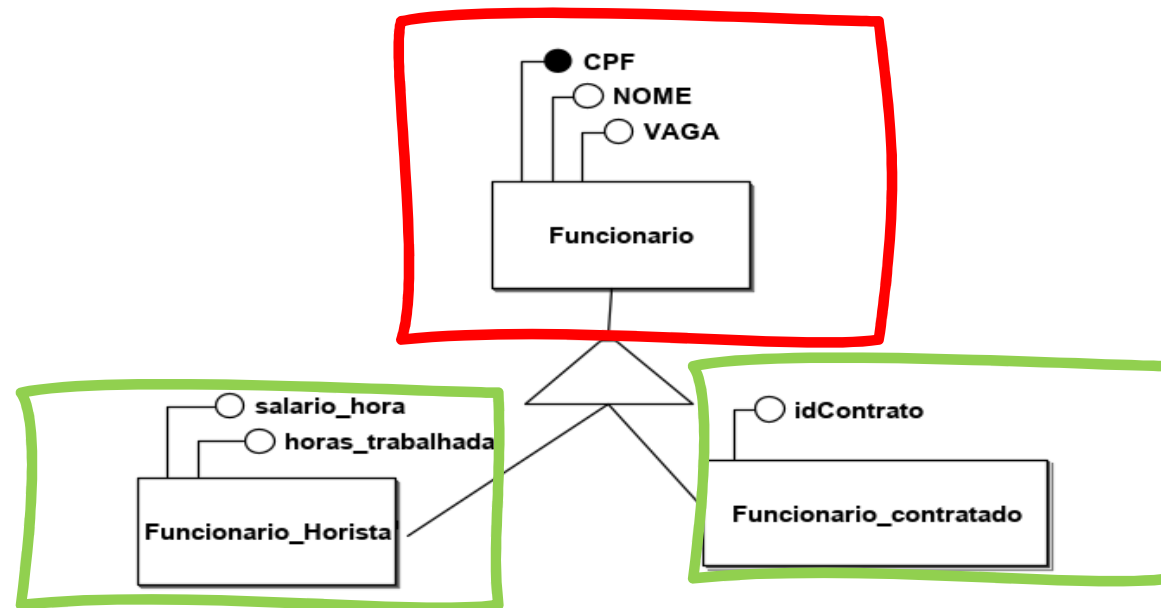
As subclasses podem ser identificadas em duas subclasses (por especialização ou generalização):

1. Acrescentando a descrição dos atributos que façam sentido, apenas, às entidades em uma subclasse. Por exemplo, `salário_hora` não faz sentido para uma entidade `Funcionário_Contratados`, cujo pagamento é determinado por um contrato individual.



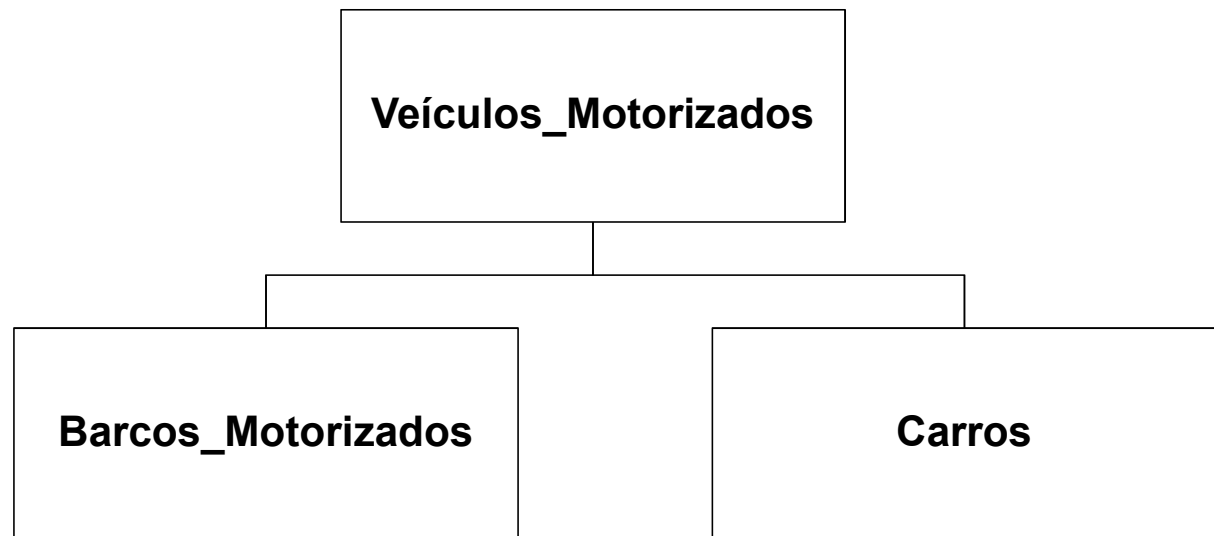
Hierarquias de classes – Especialização/Generalização

2. A identificação do conjunto de entidades que participam em algum relacionamento. Por exemplo, poderíamos definir o relacionamento Gerência de forma que os conjuntos de entidades participantes sejam Funcionário_Sêniores e Departamentos, para assegurar que, apenas, os funcionários sêniores possam ser gerentes.



Hierarquias de classes – Especialização/Generalização

- Um outro exemplo, Barcos_Motorizados e Carros podem ter atributos descritivos diferentes, como a tonelagem e o número de portas, mas como as entidades de Veículos_Motorizados, eles devem ter licenciamento. A informação de licenciamento pode ser capturada por um relacionamento Licenciado_para entre Veículos_Motorizados e um conjunto de entidades chamado Proprietários.

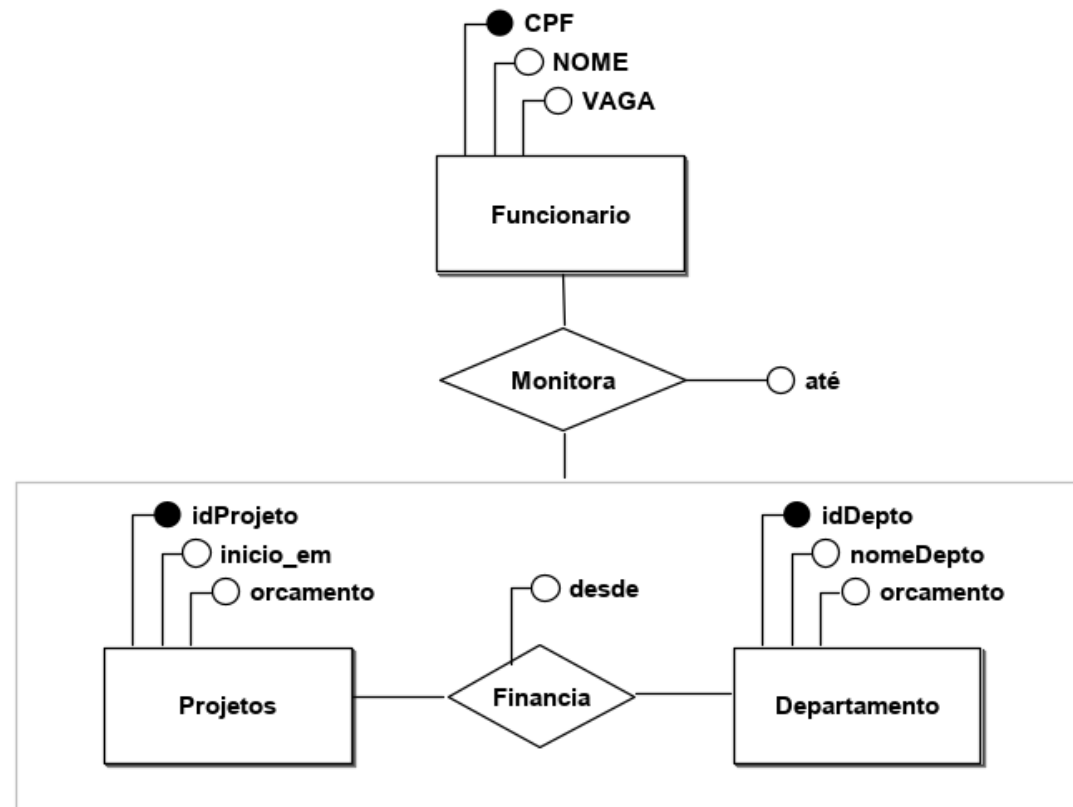


Agregação

- É um tipo especial de associação onde se tenta demonstrar que as informações de um objeto, precisam ser complementadas pelas informações de um objeto de outra classe.
- Esse tipo de associação tenta demonstrar uma relação entre o todo ou a parte entre os objetos associados.
- O conjunto de relacionamentos Financia captura essa informação. Um departamento que financia um projeto pode designar os funcionários para monitorar o financiamento.

Agregação

- Intuitivamente, Monitora deve ser um conjunto de relacionamentos que associa um relacionamento Financia (em vez de uma entidade Projetos ou Departamentos) a uma entidade Funcionários. Entretanto, definimos os relacionamentos para associar duas ou mais entidades.



Referências

- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011.
- RAMAKRISHNAN, R., GEHRKE, J. *Sistemas de Gerenciamento de Banco de Dados*. Porto Alegre: Bookman, 2011.

ATÉ A PRÓXIMA!