



BANCO DE DADOS

PROF. DRA. SUELANE GARCIA FONTES

O QUE É BANCO DE DADOS?

- ❑ coleção de dados interrelacionados e persistentes que representa um subconjunto dos fatos presentes em um domínio de aplicação.
- ❑ Conjunto de dados que se relacionam.

Banco de dados = instância de dado + metadados

❑ **Instância de dado**

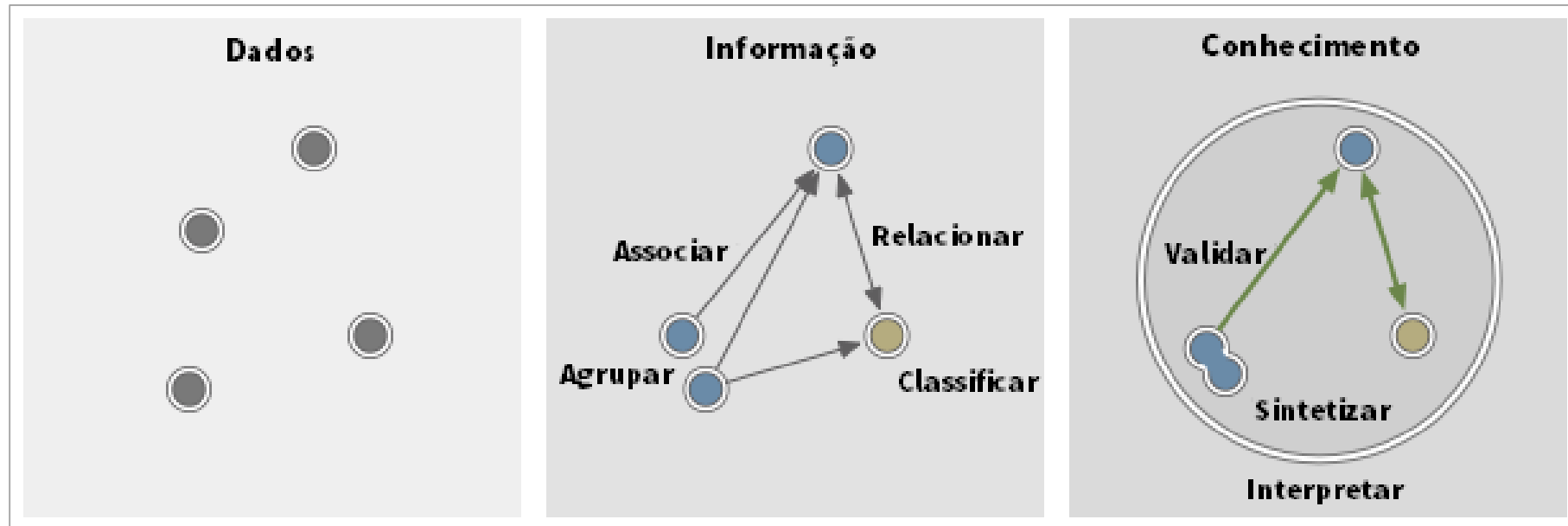
- Dado propriamente

❑ **Metadados**

- Dicionário de dados:
 - ✓ Esquema da base de dados;
 - ✓ Acessado por meio de linguagens de definição de dados.

CONCEITOS

O QUE É BANCO DE DADOS?



Qualquer registro ou indício relacionável a alguma entidade ou evento.

Resultado do processamento, manipulação e organização de dados

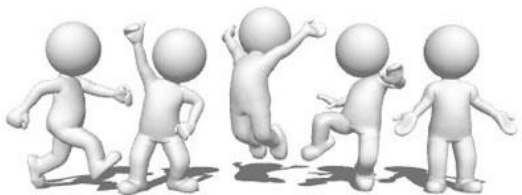
É a capacidade de interpretar e operar sobre um conjunto de Informações

fonte: Adaptado de *Cultivating the Landscape of Innovation in Computational Journalism*, de Nick Diakopoulos

CONCEITOS

O QUE SÃO DADOS PESSOAIS?

Dados pessoais são informação relativa a uma pessoa viva, identificada ou identificável. Também constituem dados pessoais o conjunto de informações distintas que podem levar à identificação de uma determinada pessoa.



O QUE SÃO ANONIMIZADOS?

Não contêm nenhum elemento de identificação



O QUE SÃO DADOS SENSÍVEIS?

Dados pessoais que revelem a origem racial ou étnica, as convicções religiosas, filosóficas ou morais, as opiniões políticas, a filiação a sindicatos ou organizações de caráter religioso, filosófico ou político, dados referentes à saúde ou à vida sexual e dados genéticos ou biométricos.

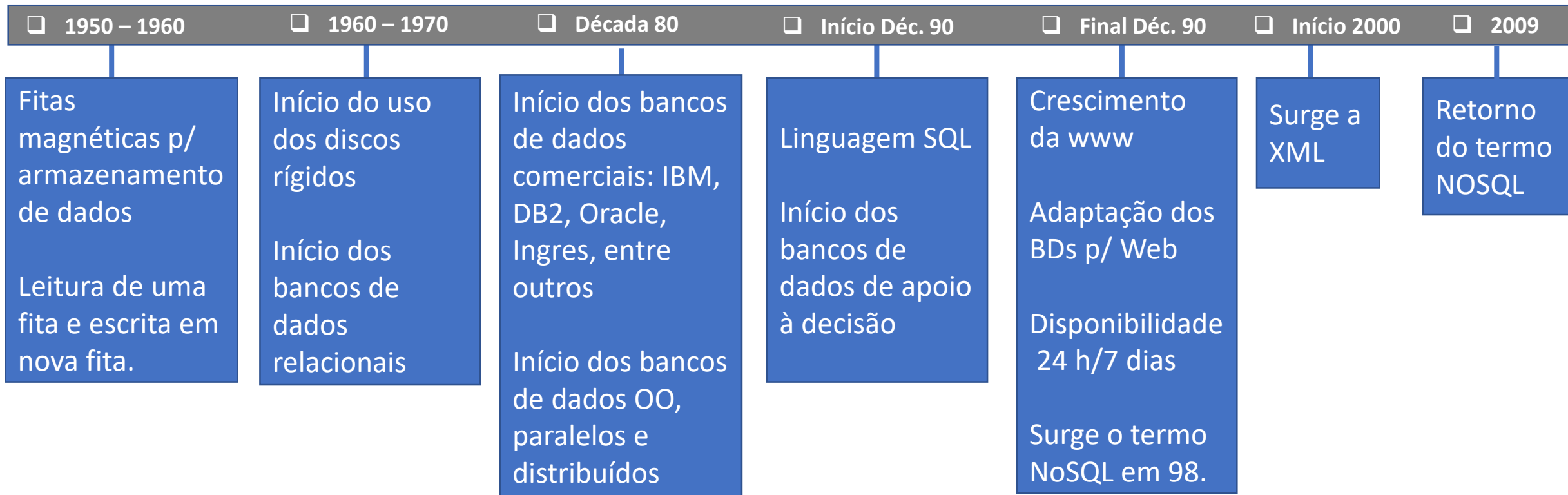


CONCEITOS

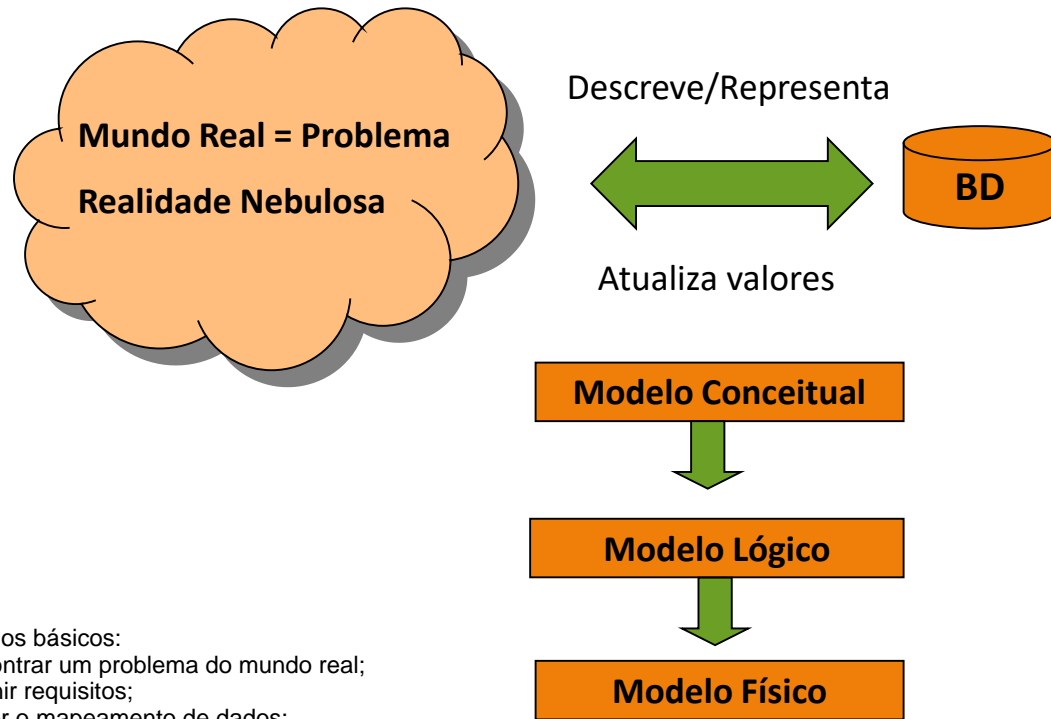


DADOS = VALOR

- Os dados estão em **TODA** parte.
- A quantidade de dados digitais cresce a um ritmo acelerado.
- Artigo da Forbes sugere que até o ano 2020 serão geradas cerca de 1,7 bilhão de novas informações por segundo.
- Organizações orientadas à dados.
- Foco na conversão de dados em valor. Dados brutos para algo relevante.
- Obter vantagem competitiva.
- Novos termos: ***Data Science, Data Mining, Data Analytics e Machine Learning***



HISTÓRIA DO BANCO DE DADOS



Princípios básicos:

1. Encontrar um problema do mundo real;
2. Definir requisitos;
3. Fazer o mapeamento de dados;

SGBD – SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS.

- Software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados.

PostgreSQL



ORACLE



Modelo de Dados - descrição formal da estrutura de um BD.

- É uma descrição dos tipos de informações que estão armazenadas em um banco de dados.

Modelo de Dados Relacional:

- ▣ Modelo Conceitual
- ▣ Modelo Lógico
 - 1. Ilustração (desenho)
 - 2. Conjuntos de tabelas
- ▣ Modelo Físico
 - 3. Script da linguagem SQL

Banco de Dados

Coleção de dados que mantém relações entre si e estão armazenadas em algum dispositivo.

Exemplos:

- ▣ Títulos de uma Biblioteca,
- ▣ Filmes de uma Videolocadora,
- ▣ Empregados de uma Empresa (cargos, salários, etc.),
- ▣ Textos sobre um determinado assunto (BD Textuais),
- ▣ Imagens, Sons e Vídeos (BD Multimídia), etc.

O **gerenciamento** envolve:

- ☐ A definição de estruturas para o armazenamento da informação
- ☐ O fornecimento de mecanismos para manipular as informações
- ☐ Quando vários usuários acessam os dados o SGBD precisa garantir a INTEGRIDADE dos dados.

OBJETIVOS:

- ☐ Isolar os usuários dos detalhes mais internos do banco de dados (abstração de dados).
- ☐ Prover independência de dados às aplicações (estrutura física de armazenamento e a estratégia de acesso).

Vantagens:

- ✓ rapidez na manipulação e no acesso à informação,
- ✓ redução do esforço humano (desenvolvimento e utilização),
- ✓ redução da redundância e da inconsistência de informações,
- ✓ redução de problemas de integridade,
- ✓ compartilhamento de dados,
- ✓ aplicação automática de restrições de segurança,
- ✓ controle integrado de informações distribuídas fisicamente.

Instâncias e Esquemas

- ☐ Os bancos de dados mudam a medida que informações são inseridas ou apagadas.
- ☐ A coleção de **informações armazenadas** é chamada de **INSTÂNCIA** do banco de dados (mudam com frequência).
- ☐ O **projeto geral do banco de dados** é chamado **ESQUEMA** do banco de dados (não mudam com frequência)

SGBD – SISTEMA DE GERENCIAMENTO DE BD

SGBD – CAMADAS

Camada de interface de alto nível que provê consultas, manipulação e definição dos dados

- ✓ Acessível por usuários interativamente (*Query Analyzer*) ou por aplicativos externos;
- ✓ Recebe a solicitação (consulta ou chamada a procedimento) e envia à camada de Processamento de Transações

É a mais próxima do usuário, do que a gente vê;

WorkBeach -> Interface gráfica para utilizar o MySQL - SGBD

Camada de processamento de transações com:

- Tradutor – Linguagem Interna
- Otimizador – qual o melhor caminho? Deve-se usar Índices, Buffers, etc.?
- Verificação da visão do usuário do BD: o que ele está pedindo “existe” para ele?
- Controle de Integridade – As regras semânticas dos dados estão sendo salva guardadas?
- Controle de Concorrência: Atender a todos;
- Controle de segurança: Não permitir acessos indevidos;
- Controle de recuperação de falhas: caso falhe a conexão, ou uma mídia, o BD deve voltar à normalidade.

Ajuda a entender e **interpretar** o que foi solicitado na camada de interface;
Gerenciador responde a perguntas feitas na interface buscando pelo código, por exemplo definindo os índices
Inteligência interna;

Camada de Acesso a Dados

- Usa o sistema de arquivos do Sistema Operacional para armazenar o BD e prover acesso eficiente aos dados físicos.
- Usa Buffers e cuida dos segmentos de *RollBack* e dos registros de Log (*Redo*) – de acordo com o que lhe foi passado pela camada de Processamento de Transações.

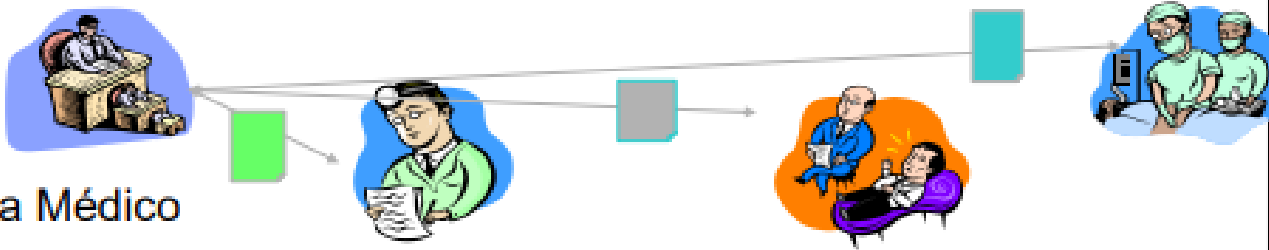
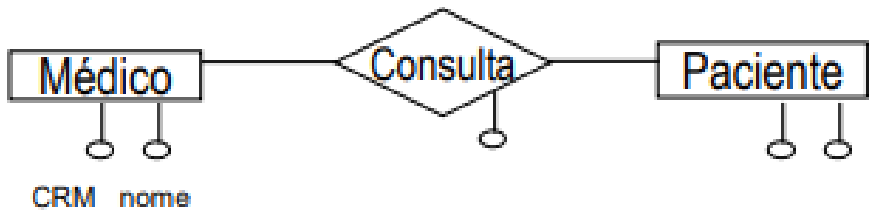
RollBack

Desfaz determinada transação;

Garantindo seguranças em casos de erros;

Transações

- Utilizadas para controlar a integridade dos dados no banco de dados
- Acessos simultâneos de vários usuários
- Falhas no sistema

Mundo Real	 <p>Sistema Médico</p>			
Modelo Conceitual (modelo abstrato dos dados)	<ul style="list-style-type: none"> Independente do modelo de dados Independente do SGBD 			
Modelo Lógico (estrutura dos dados)	<ul style="list-style-type: none"> Dependente do modelo de dados Independente do SGBD <p>Médico (CRM, Nome)</p>	Relacional	Orientado a Objetos	Objeto-relacional
Modelo Físico	<ul style="list-style-type: none"> Dependente do modelo de dados Dependente do SGBD 	<ul style="list-style-type: none"> Organização física dos dados Estruturas de armazenamento de dados Índices de acesso 		

19

Modelo de Dados - descrição formal da estrutura de um BD.

- Um modelo de dados é uma coleção de ferramentas conceituais para a descrição de dados, relacionamentos, semântica de dados e restrições de consistência

- ☐ **Modelos de Dados (conceitual)**

- ✓ Entidade-Relacionamento (ER)
- ✓ Orientado a Objetos (OO)

- ☐ **Modelos de Dados (lógico)**

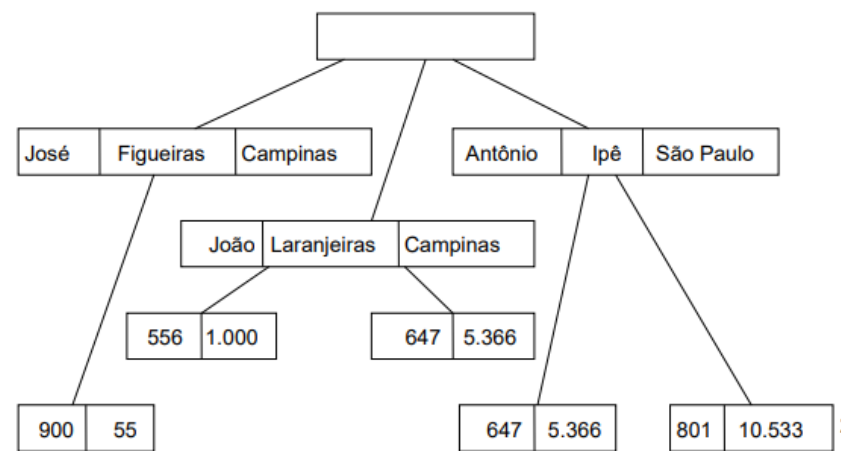
- ✓ Redes
- ✓ Hierárquico
- ✓ Relacional
- ✓ Orientado a Objetos

- ☐ **Modelo de Dados (Físico)**

MODELO DE DADOS

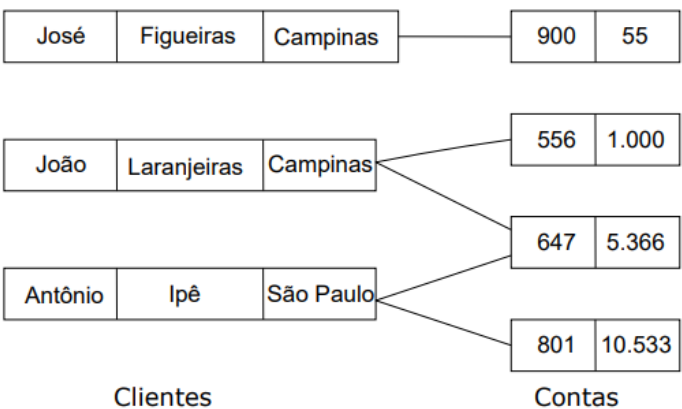
Modelo Hierárquico

- ✓ Os dados e relacionamentos são representados por registros e ligações, respectivamente.
- ✓ Os registros são organizados como coleções arbitrárias de árvores.



Modelo de Redes

Os dados são representados por coleções de registros e os relacionamentos por elos



Modelo Relacional

Relaciona registros a partir de valores do registro.

Tabela Cliente (dados)

cód-cliente	nome	rua	cidade
015	José	Figueiras	Campinas
021	João	Laranjeiras	Campinas
037	Antônio	Ipê	São Paulo

Tabela Conta (dados)

nro-conta	saldo
900	55
556	1.000
647	5.366
801	10.533

cód-cliente	nro-conta
015	900
021	556
021	647
037	647
037	801

Tabela Cliente-Conta
(relacionamento)

MODELO DE DADOS

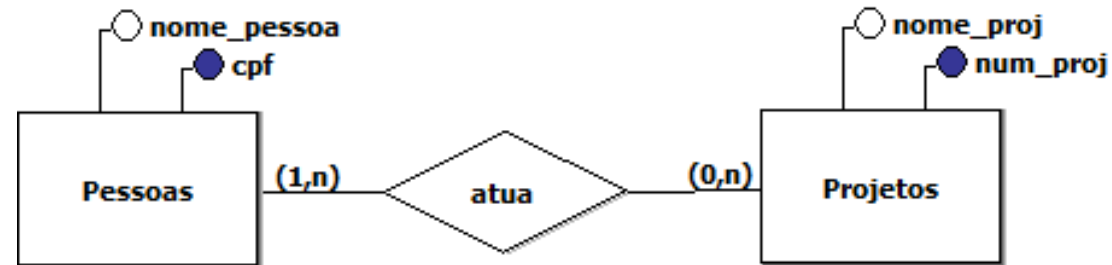
❑ Modelo Conceitual

Descrição do banco de dados de forma independente de implementação em um Sistema de Gerenciamento de Banco de Dados (SGBD). Expressa quais dados aparecerão no BD e não como estes estão armazenados.

MER – Modelo de Entidade e Relacionamento

DER - Diagrama de Entidade e Relacionamento

Exemplo:



Identificar:

- ✓ Entidades
- ✓ Atributos
- ✓ Relacionamentos
- ✓ Chave primária
- ✓ Cardinalidade

MODELO CONCEITUAL






MODELO DE ENTIDADE-RELACIONAMENTO - MER

- O Modelo Entidade-Relacionamento (MER) é um modelo de dados de alto-nível criado com o objetivo de representar a semântica associada aos dados do minimundo.
- O MER é utilizado para na fase de projeto conceitual, onde o esquema conceitual do banco de dados da aplicação é concebido.
- Criado por Peter Chen em 1976. É um modelo independente de aspectos de implementação (modelo de dados conceitual)
- Baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos chamados **entidades** e nos **relacionamentos** entre esses objetos
- A proposta do modelo E-R visa a compreensão da realidade em que se situa o problema.
- **Objetivo:** facilitar o projeto de banco de dados, possibilitando a especificação da estrutura lógica geral do banco de dados
- Representado pelo **Diagrama de Entidade e Relacionamento – DER**








Componentes do Diagrama E-R (Peter Chen)

- **Retângulos:** representam conjuntos de entidades
- **Elipses:** representam atributos
- **Losangos:** representam conjuntos de relacionamentos
- **Linhas:** ligam atributos → entidade e
entidade → relacionamento

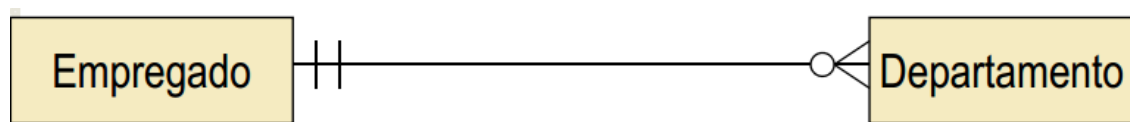
NOTAÇÃO DE PETER CHEN

<u>Símbolo</u>	<u>Significado</u>
	Tipo de Entidade
	Tipo de Entidade Fraca
	Tipo de Relacionamento
	Tipo de Relacionamento Identificado
	Atributo

No BRmodelo utilizamos a notação de Peter Chen

<u>Símbolo</u>	<u>Significado</u>
	Atributo chave
	Atributo Multivalorado
	Atributo Composto
	Atributo Derivativo
	Cardinalidade de razão 1:N para $E_1:E_2$ em R
	Participação total de E_2 em R
	Restrição Estrutural (min,max) sobre a participação de E em R

NOTAÇÃO DE JAMES MARTIN



Exemplo de Modelagem, onde:

◁ = muitos

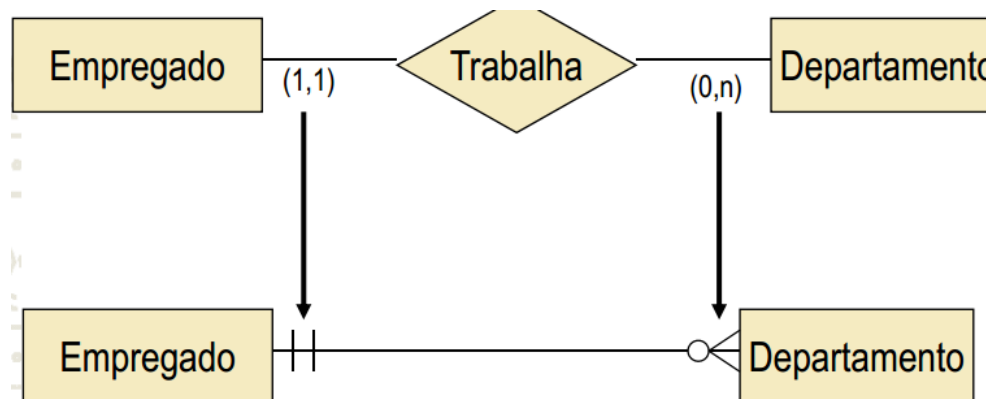
| = um

○ = a ocorrência do relacionamento é opcional;

| = a ocorrência do relacionamento é obrigatória;

Muitos chamam de notação de pé de galinha;

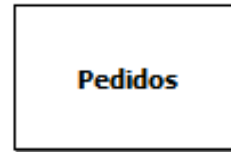
NOTAÇÃO DE PETER CHEN X JAMES MARTIN



Entidade: conjuntos de itens;

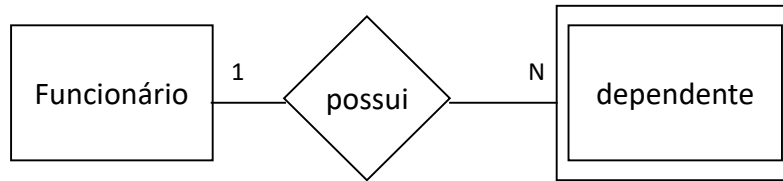
❑ **Entidade Simples:** conjunto de objetos da realidade sobre os quais se deseja manter informações no BD.

Ex.



Apenas um conjunto de itens;

❑ **Entidade Fraca:** tipo de entidade que não pode existir se não estiver relacionada a outra entidade, ou seja, depende da existência de outra entidade.

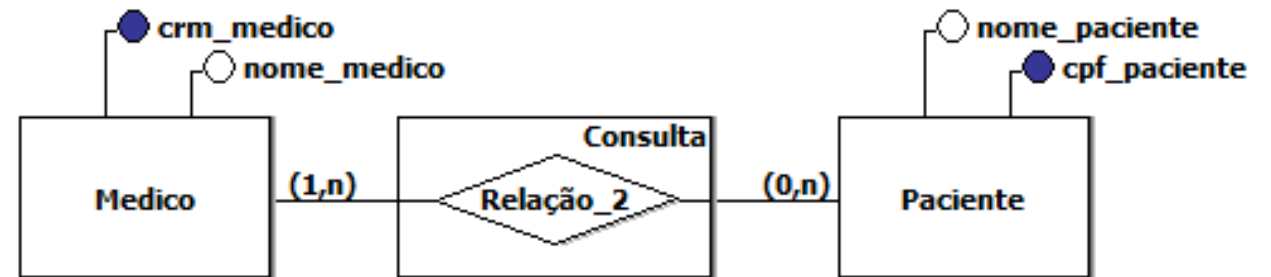


- Entidade que é específica e precisa de uma entidade maior;
- Relação de dependência;

❑ **Entidade Associativa:** Tipo de entidade que associa instâncias de um ou mais tipos de entidades e contém atributos que são peculiares para o relacionamento.

Como identificar?

- Relacionamento do tipo N : M;
- A entidade associativa tem significado independente;
- Tem um ou mais atributos;
- Participa de um ou mais relacionamentos Independentemente das entidades do relacionamento associado.



TIPOS DE ENTIDADE

Atributos: características das entidades;

- ❑ **Atributo (Campos/Colunas)** - dado associado a cada ocorrência de uma entidade. Ex. nome, endereço, tel.
- ❑ **Domínio do atributo** – conjunto de valores que um determinado atributo pode assumir
Ex. tipo de dados – varchar/int/boolean / aceita valor nulo ou não – null/not null
 - Quais são os seus tipo;
 - Ele pode ficar vazio ou não - null
 - Não pode ficar vazio - not null

Tipos de Atributos

- ❑ **Simples ou Monovalorado** - assume um único valor para cada elemento da entidade. Ex. nome
- ❑ **Composto** – conteúdo formado por subAtributos. Ex. Endereco (rua, cep, bairro);
- ❑ **Multivalorado** – uma única entidade tem diversos valores para este atributo (seu nome é sempre representado no plural). Ex. Telefones.
- ❑ **Determinante** – define unicamente as existências de uma entidade (chave primária). Ex. CNPJ.
- ❑ **identificador de entidade (chave primária)** – conjunto de um ou mais atributos e relacionamentos cujos valores servem para distinguir uma ocorrência da entidade das demais ocorrências da mesma entidade.
 - O atributo determinante, identificador de entidade e chave-primária mesma "coisa";
- ❑ **Derivados** aquele cujo valor pode ser derivado de outros atributos ou entidades a ele relacionados.

Ex.: o conjunto de entidades **empregado** possui o atributo quantidade **dependentes** que representa o número de pessoas dependentes do funcionário e, portanto, pode ser obtido através da soma do número das entidades dependentes associadas ao empregado em questão.

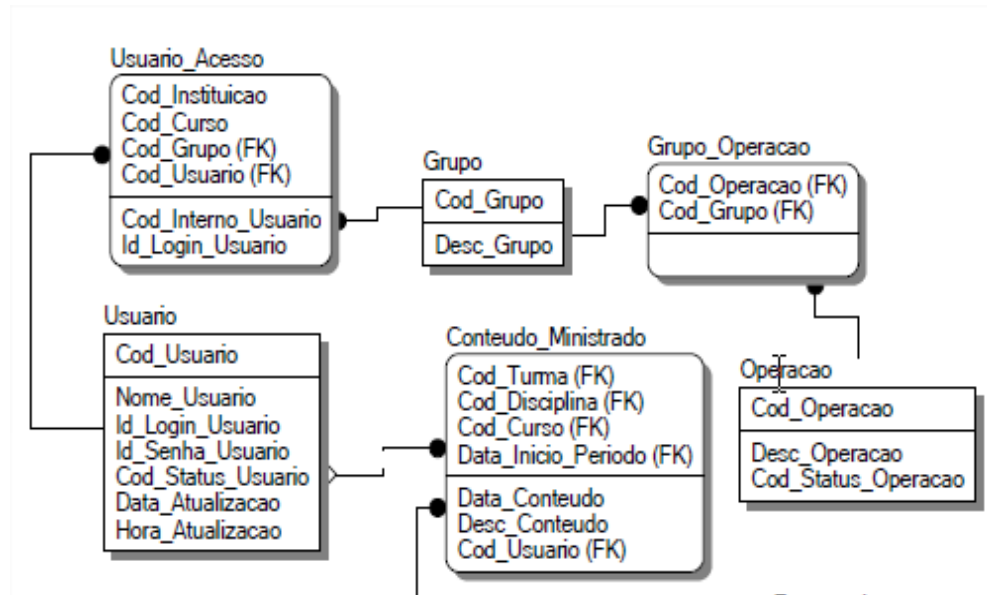
TIPOS DE ATRIBUTOS

❑ Modelo Lógico

Modelo de dados que representa a estrutura de dados de um banco de dados conforme o SGBD a ser utilizado.

- Gestão de acesso ao arquivo;

Exemplo:



Esquema Lógico:

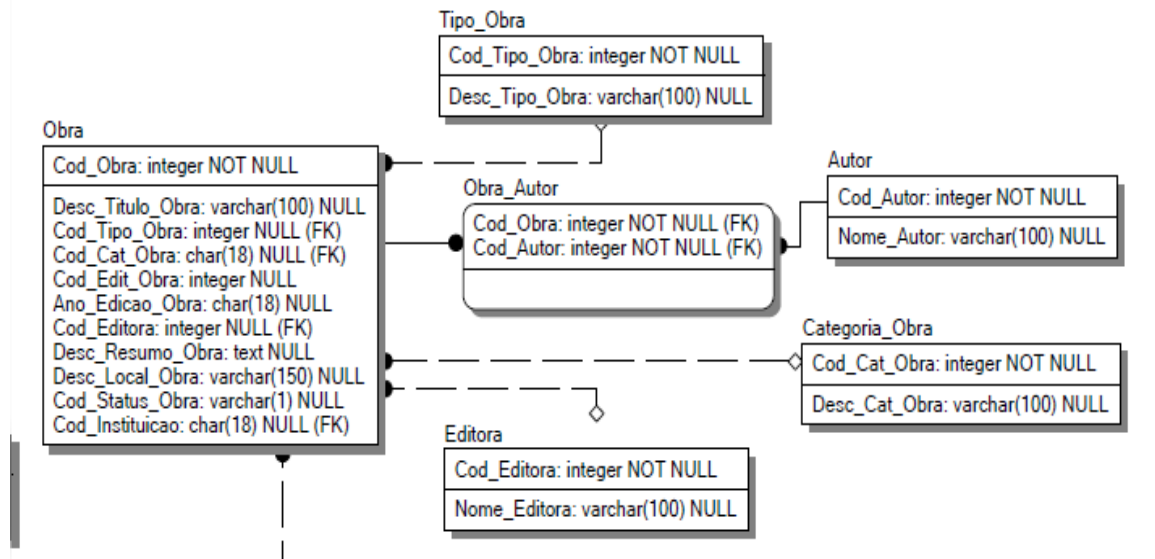
TipoProduto (codTipoProd, descTipoProd)

Produto (codProd, descProd, precoProd, codTipoProd)

codTipoProd referencia TipoProduto

❑ Modelo Físico

Descreve as estruturas físicas de armazenamento de dados baseada no modelo lógico.

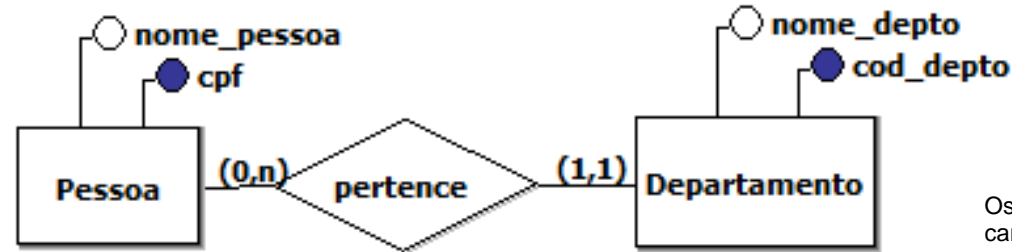


Script SQL:

```
create table editora (  
    cod_editora integer not null primary key,  
    Nome_editora varchar (100) null  
)
```

- ❑ **Relacionamento:** Conjunto de associações entre ocorrências de entidades. É o fato ou acontecimento que liga dois objetos do mundo real (entidades).

Exemplo:



Os tipos de relacionamento estão diretamente ligados a cardinalidade.

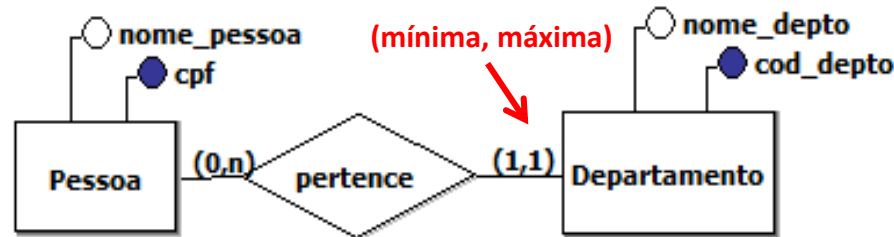
- ❑ **Cardinalidade (Min, Max):** restrição que expressa o número de entidades ao qual outra entidade pode estar associada via um relacionamento.

O quanto eles se relacionam e se eles são obrigados a se relacionar ou não.

Cardinalidade:
Min, Max

Min
0 = opcional
1 = obrigatório

Max
1 = um
N = muitos
(para limitar somente no código, no banco apenas N)

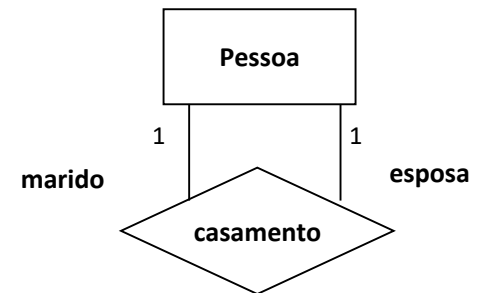
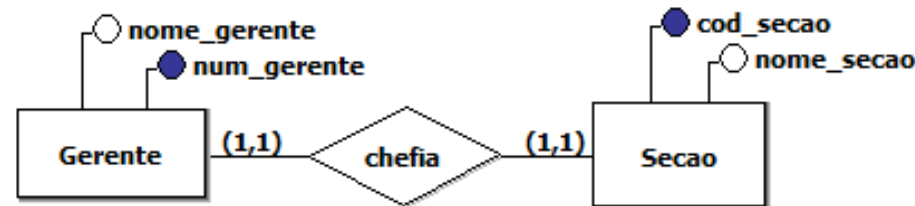
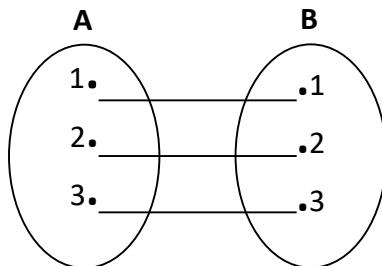


“um empregado **obrigatoriamente está** lotado no máximo em 1 departamento. Um departamento **pode ter** até N empregados lotados nele.”

Tipo de Relacionamento

- ❑ **Relacionamento 1:1**

Exemplo:



TIPOS DE RELACIONAMENTO

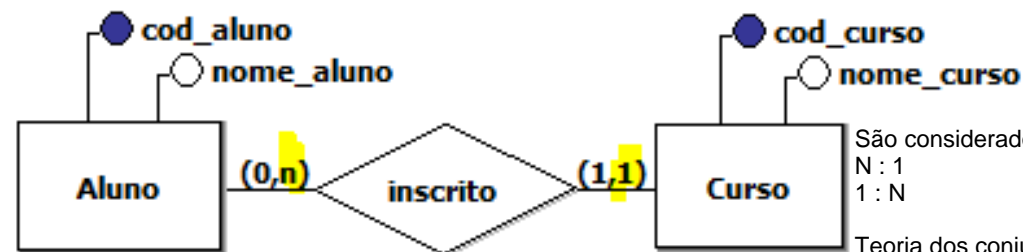
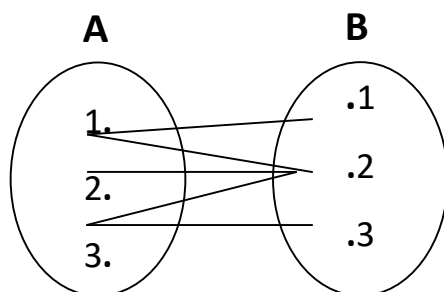
Para saber os tipos de relacionamentos considere as máximas.

Tipos de relacionamento:

1:1
1:N / N:1
N:N

❑ Relacionamento 1:N (Um-para-Muitos)

Exemplo:



São considerados as mesmas coisa
N : 1
1 : N

Teoria dos conjuntos - Mat.

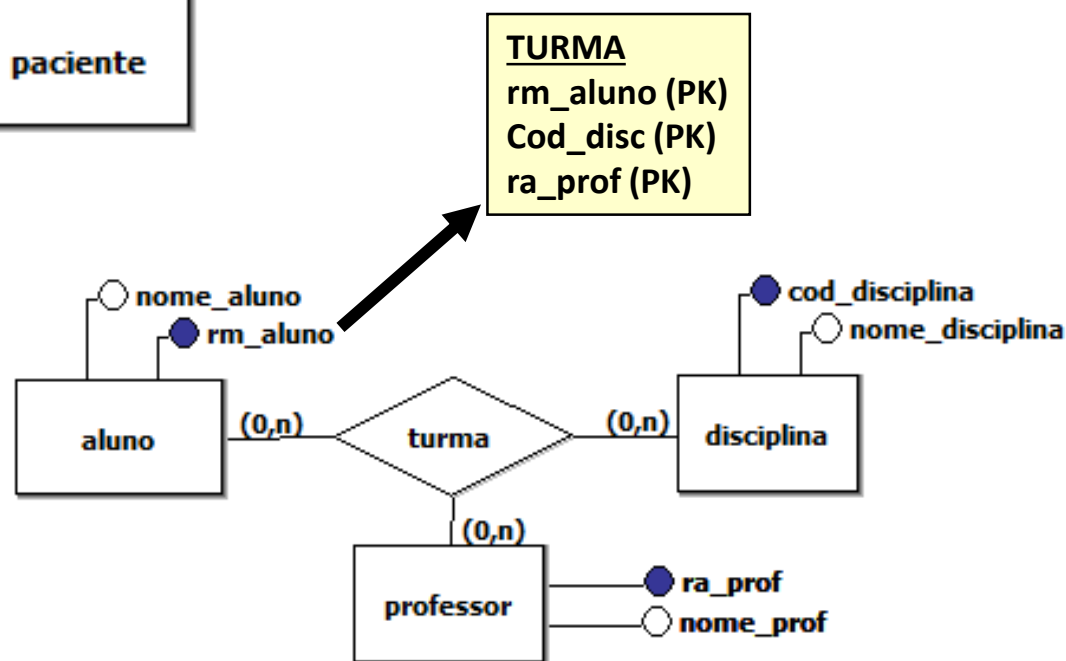
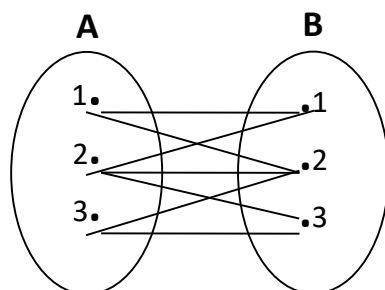
❑ Relacionamento N:N (Muitos-para-Muitos)



Relacionamentos que não existem:

(n, 1) - mínima varia entre 0 e 1.

(1,0) - a máxima precisa ser maior que 0.



TURMA
rm_aluno (PK)
Cod_disc (PK)
ra_prof (PK)

Tabelas:

ALUNO

RM_aluno (PK)

Nome_aluno

DISCIPLINA

Cod_disc (PK)

Nome_disc

PROFESSOR

RA_prof(PK)

Nome_prof

Modelo lógico:
Entidades -> tabelas
Atributos -> campos
Chave primária -> PK
(inglês)

TIPOS DE RELACIONAMENTO

N:N sempre da origem a uma terceira entidade a relação entre eles.

GRAU DO RELACIONAMENTO

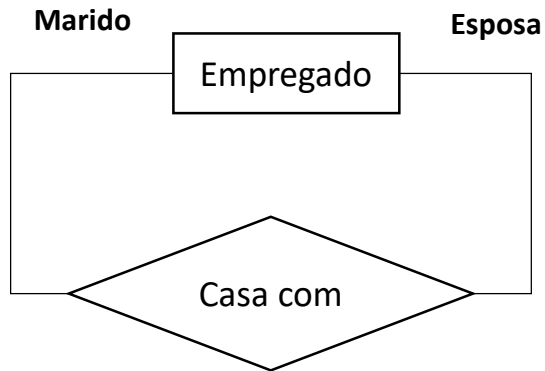
O grau de um relacionamento é o número de entidades participantes:

1. Unários; Posso ter mais de 3 relacionamentos?
Sim!
2. Binários; Quartenário (4 entidades)
Enário (acima de 3 relacionamentos - difícil de acontecer)
3. Ternários.

O grau seria definido de acordo com a quantidade de entidades diretamente ligadas em tal relacionamento;

Losango

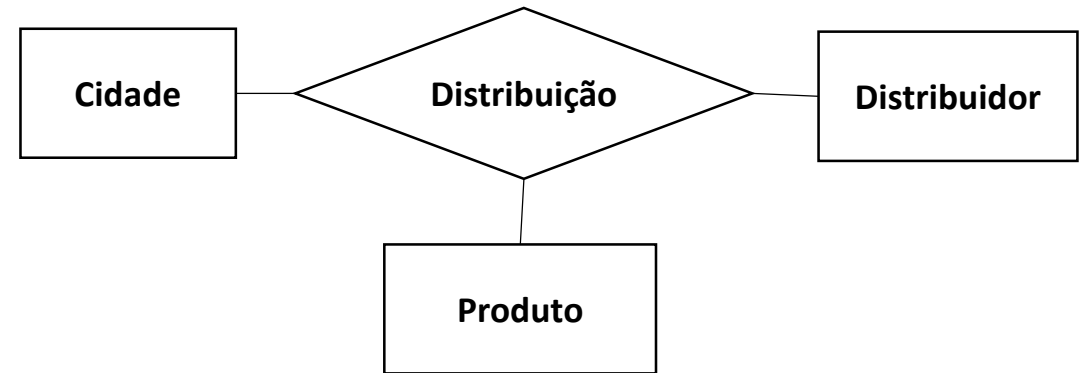
Relacionamento Grau 1



Relacionamento Unário

Auto relacionamento;
Relacionamento entre a mesma entidade, ou seja, a entidade se relaciona com ela mesma.
Empregados se relacionando com empregados

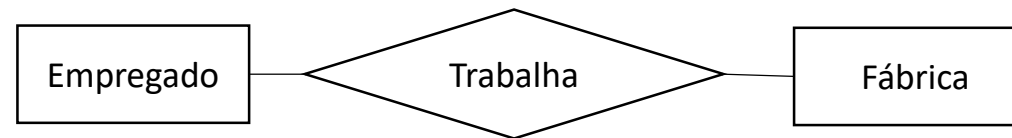
Relacionamento Grau 3



Relacionamento Ternário

Três entidades se relacionando

Relacionamento Grau 2



Relacionamento Binário

Duas entidades se relacionando

❑ Auto-Relacionamento

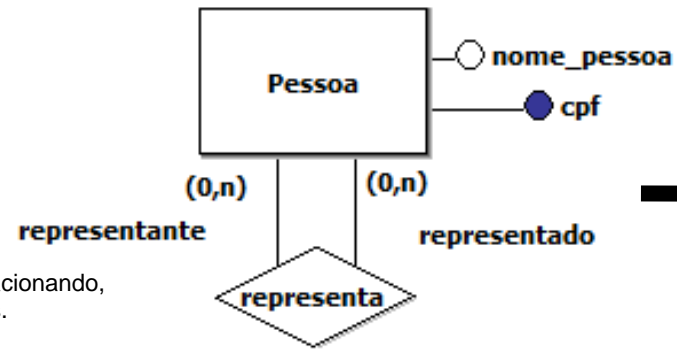
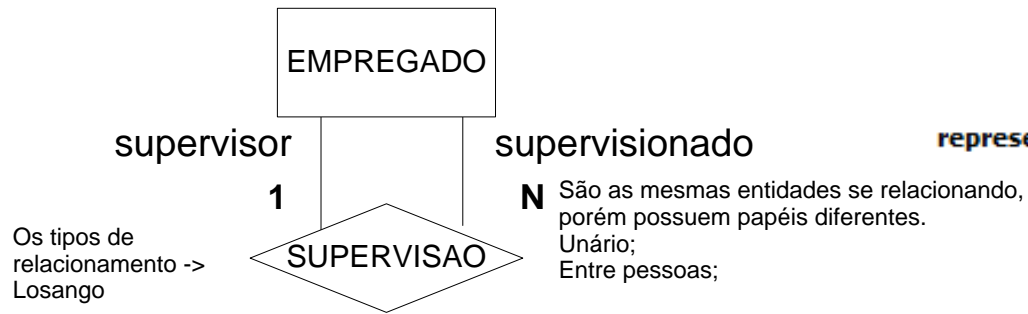
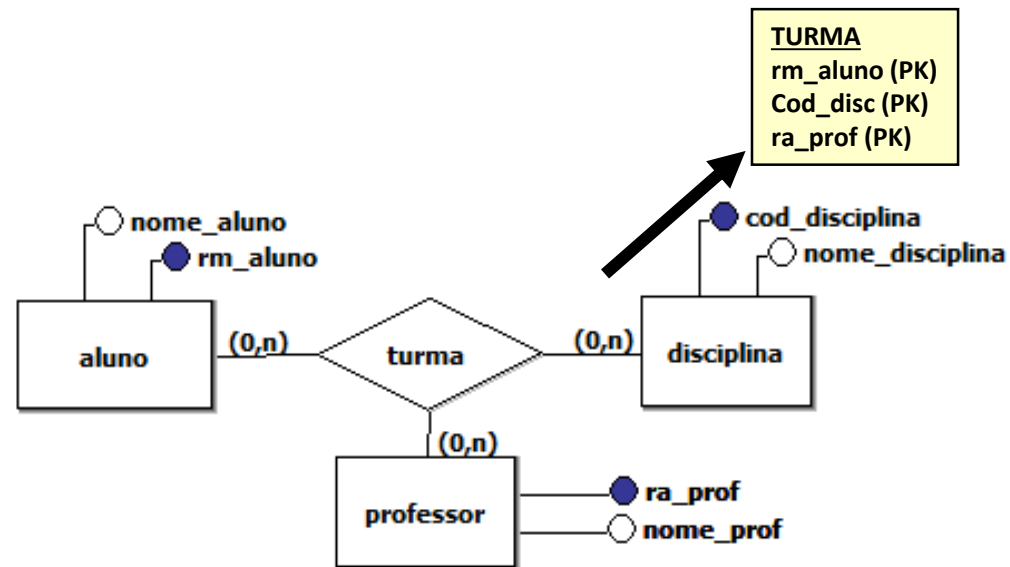
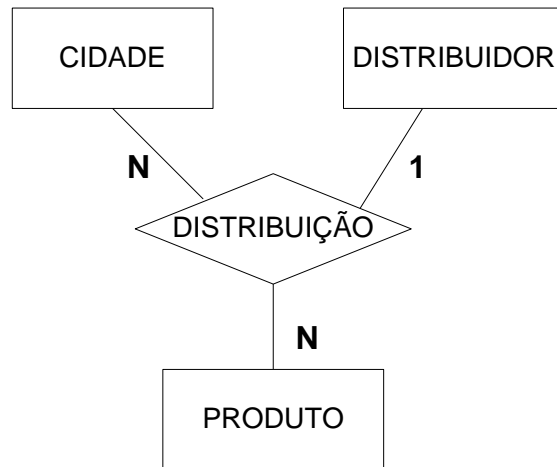


Tabela REPRESENTA
CPF_REPRESENTANTE (PK e FK)
CPF_REPRESENTADO (PK e FK)

Tabela PESSOA
CPF (PK)
NOME_PESSOA

*“um empregado pode ser **supervisionado** por no máximo 1 empregado.
Um empregado pode **supervisionar** no máximo N empregados.”*

❑ Relacionamento Ternário



Tabelas:
ALUNO
RM_aluno (PK)
Nome_aluno

DISCIPLINA
Cod_disc (PK)
Nome_disc

PROFESSOR
RA_prof(PK)
Nome_prof

“um produto em uma cidade é entregue por no máximo 1 distribuidor.”

TIPOS DE RELACIONAMENTO


GENERALIZAÇÃO E ESPECIALIZAÇÃO

Especialização O que se tem de particular (específico);

- Definição de uma entidade que é um subconjunto de uma outra entidade
- Processo de classificar o conjunto de entidades em conjunto de entidades especializados

Generalização O que se tem em comum;

- Definição de uma entidade que é um superconjunto de uma outra entidade.
- Processo de generalizar vários conjuntos de entidades em um só conjunto de entidade

IS A => é um 

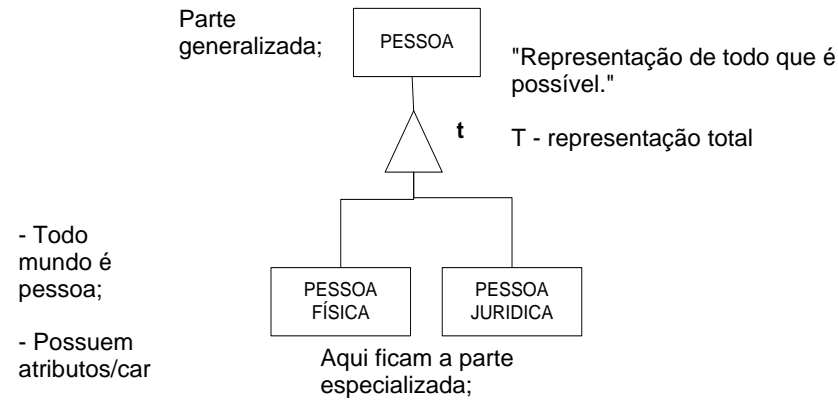
Generalização e Especialização
precisam ser totalmente diferentes
um do outro caso contrário é apenas
uma caracterização;



GENERALIZAÇÃO E ESPECIALIZAÇÃO

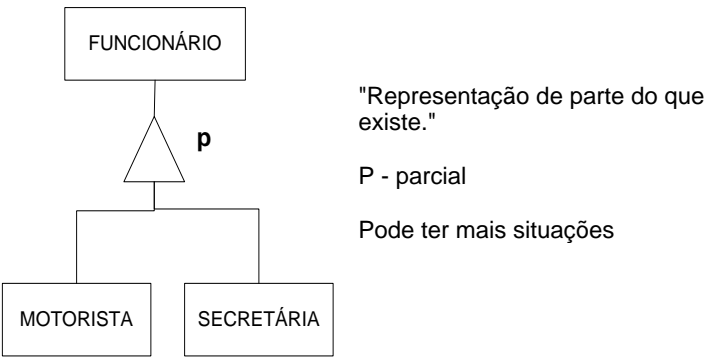
Total: A entidade referida em um certo relacionamento, deve ter todas as ocorrências das entidades especialistas participando daquele relacionamento.

Total Indica que todo cliente é Pessoa Física ou Juridica

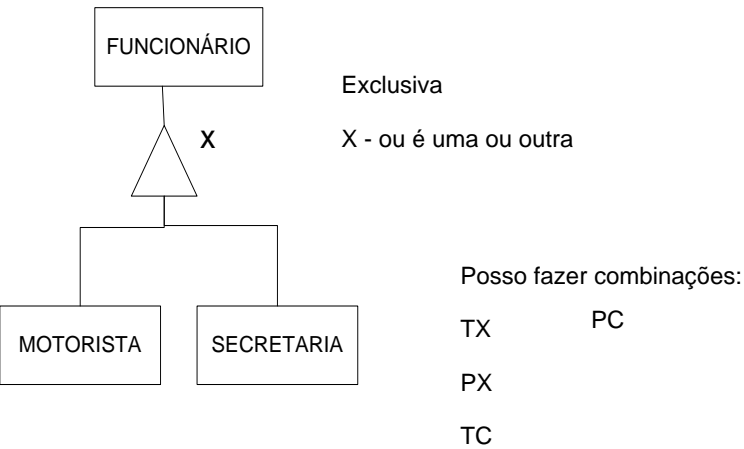


Parcial: A entidade referida em um certo relacionamento, terá algumas ocorrências das entidades especialistas participando daquele relacionamento dependendo de determinadas condições.

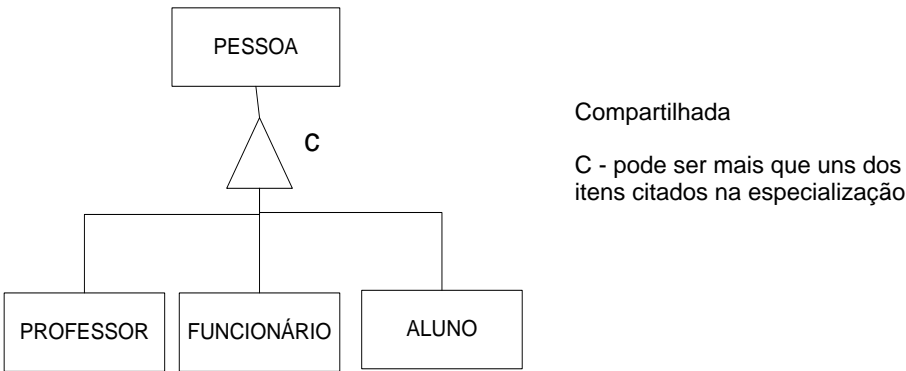
Parcial Indica que nem todo Funcionário é Motorista ou Secretária



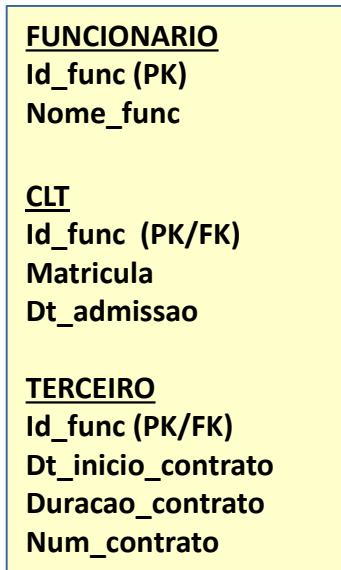
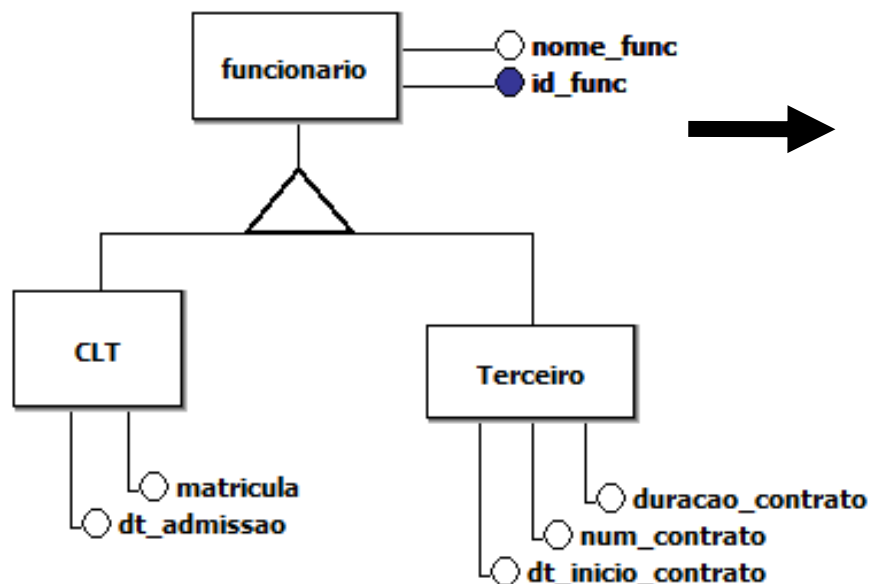
Exclusiva Indica que um Funcionário somente pode ser motorista ou Secretária e não ambos.



Compartilhada Indica que uma pessoa pode ser Professor/Aluno/ Funcionário ao mesmo tempo.



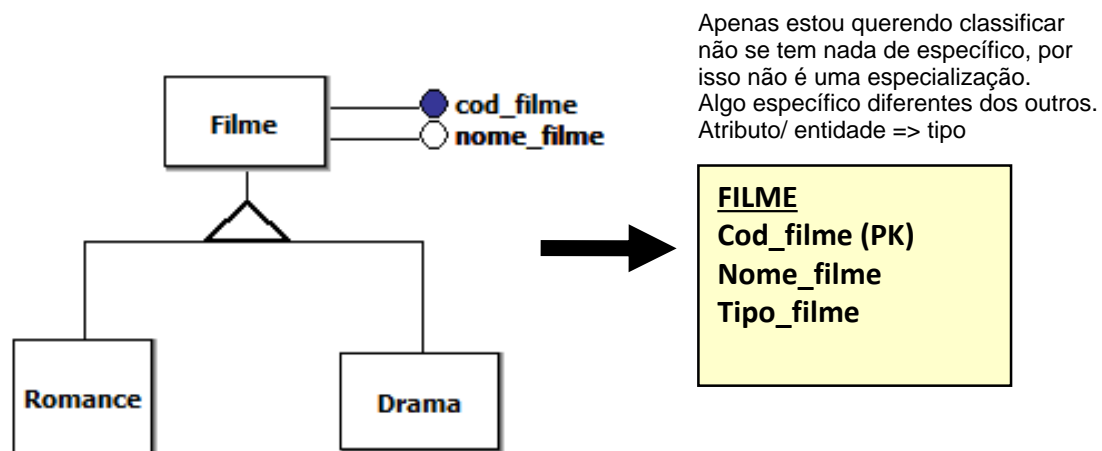
❑ Exemplo – Generalização/Especialização



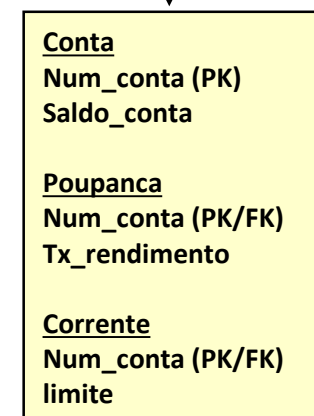
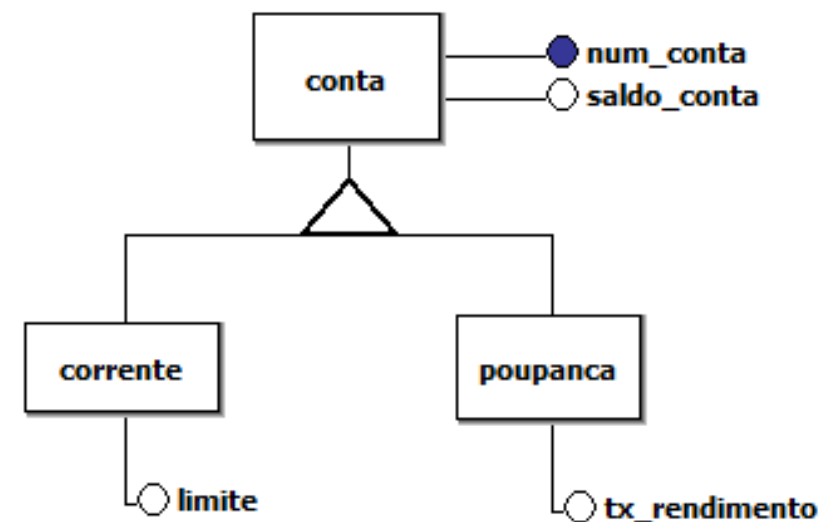
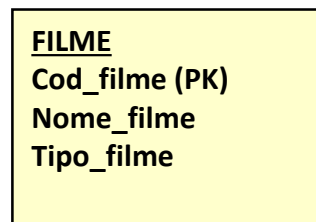
Entidades se transformam em tabelas na tradução

Por isso, teremos a 9 regras

Criar somente uma tabela para a entidade generalizada e migrar todos os atributos e relacionamentos para essa tabela



Apenas estou querendo classificar não se tem nada de específico, por isso não é uma especialização. Algo específico diferentes dos outros. Atributo/ entidade => tipo



Toda situação teremos que colocar uma generalização e especialização?

Não, pois ela não cabe em todas as situações às vezes queremos apenas classificar. Ou seja, não quero guardar dados diferentes.

Generalização e Especialização

Especializada

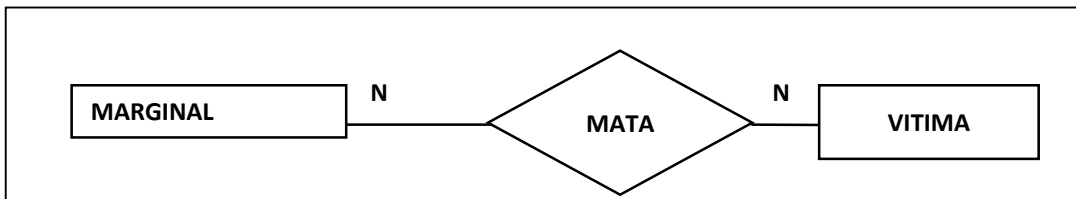
Utilizaremos se realmente tiver atributos muitos diferentes de um para outro.

Criar somente tabelas para as entidades especializadas

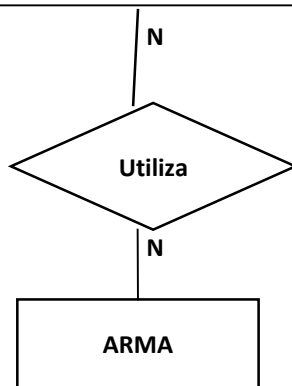
❑ AGREGAÇÃO

Uma limitação do modelo E-R é que não é possível expressar relacionamentos entre relacionamentos. E algumas vezes é necessário relacionar uma entidade com a ocorrência de um relacionamento.

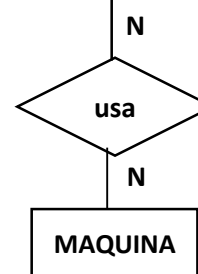
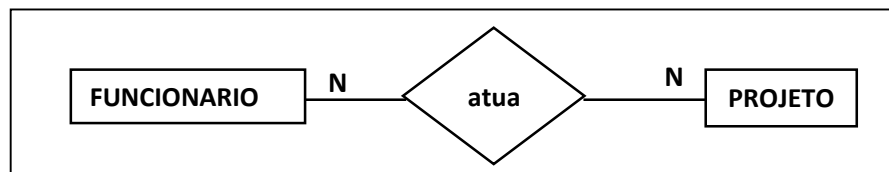
- Sempre vai ser um relacionamento N:N;
- N:N surgimento da terceira entidade
- São se pode ligar losango com losango;



Relacionamos um todo com o losango que se encontra fora;



Um fato somente acontece após a existência do primeiro fato



FUNC MAQ PROJ
RM_FUNC (FK/PK)
COD_PROJ (FK/PK)
NUM_MAQ (FK/PK)

Tabelas:

FUNCIONÁRIO

RM_FUNC(PK)

NOME_FUNC

PROJETO

COD_PROJ (PK)

DESC_PROJ

MAQUINA

NUM_MAQ (PK)

LOCAL

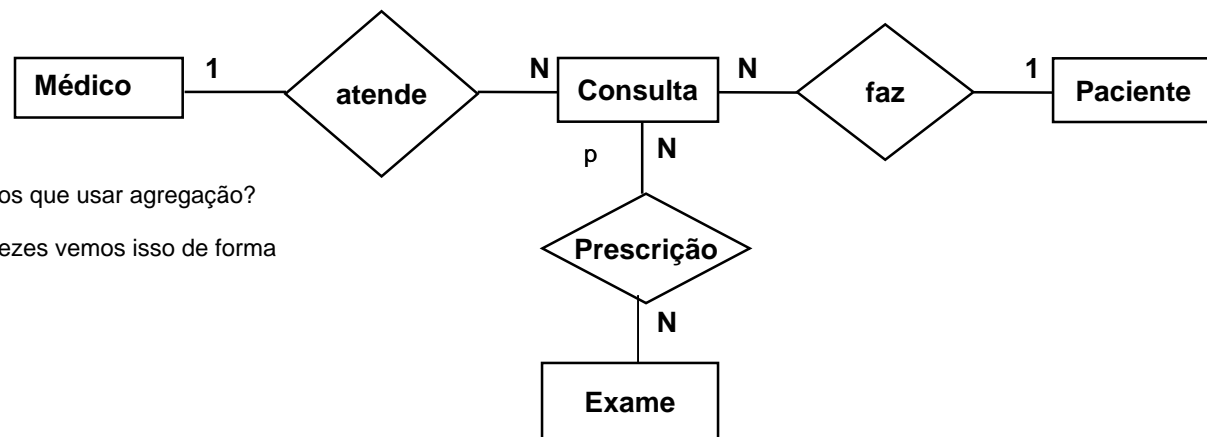
FUNCIONARIO PROJETO

RM_FUNC (FK/PK)

COD_PROD (FK/PK)

AGREGAÇÃO - Semelhança entre modelos

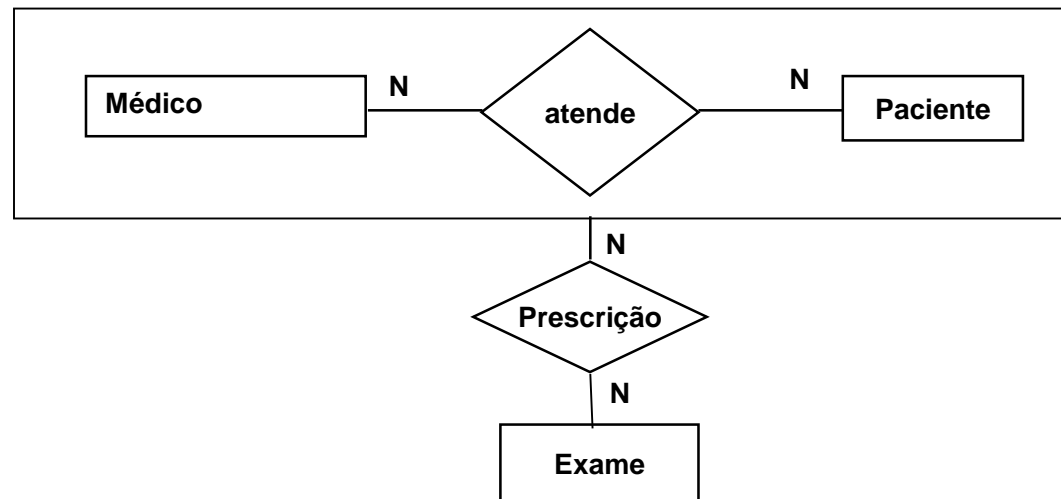
Sem Agregação



Sempre teremos que usar agregação?

Não, pois às vezes vemos isso de forma quebrada.

Com Agregação



❑ **Chave** – Dado que será empregado nas consultas à base de dados.

✓ **Chave Primária** – é uma coluna ou combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela.

Chave composta

Pode ser:

❑ **simples (1 campo)** ou identificar unicamente, não pode se repetir;

❑ **composta (2 ou mais)**

Tem dois ou mais campos.

- Utilizamos pares

- Um campo não identifica unicamente, por isso temos dois ou mais

RG	NOME	ESTADO
1	MARIA	SP
2	JOAO	BA
3	MARCOS	MG

✓ **Chave Estrangeira** – é uma coluna ou combinação de colunas cujos valores aparecem necessariamente na chave primária de uma tabela.

- Relação entre as entidades

- Obrigatoriamente precisa ter como origem a chave simples;

✓ **Chave Alternativa** – demais colunas ou combinações que não são a chave primária

❑ **Índice** – recurso físico que visa otimizar a recuperação de uma informação por um método de acesso. Está relacionado ao desempenho do sistema.

RESTRIÇÕES DE INTEGRIDADE

Um dos objetivos primordiais de um SGBD é a integridade de dados. Dizer que os dados de um banco de dados estão íntegros significa dizer que eles refletem corretamente a realidade representada pelo banco de dados e que são consistentes entre si.

Para tentar garantir a integridade de um banco de dados os SGBD oferecem o mecanismo de restrições de integridade. Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo próprio SGBD.

Podem ser classificadas como:

- Integridade de domínio
- Integridade de vazio
- Integridade de chave
- Integridade referencial

❑ INTEGRIDADE DE DOMÍNIO

Se uma determinada coluna de uma tabela é definida como inteiro, o SGBD somente permitirá a inserção de valores inteiros, não permitindo a entrada de valores alfanuméricos ou reais.

❑ INTEGRIDADE DE VAZIO

Permite ou não de acordo com a definição da coluna a inserção de valores vazios ('deixar em branco').

RESTRIÇÕES DE INTEGRIDADE

❑ INTEGRIDADE DE CHAVE

Uma vez uma ou mais colunas definidas como chave primária o SGBD não permitirá que haja duplicidade no valor delas.

Exemplo: Tabela Empregado onde o campo código do empregado é chave primária. O SGBD não permitirá que sejam inseridos 2 empregados com o mesmo código, havendo então violação de chave primária.

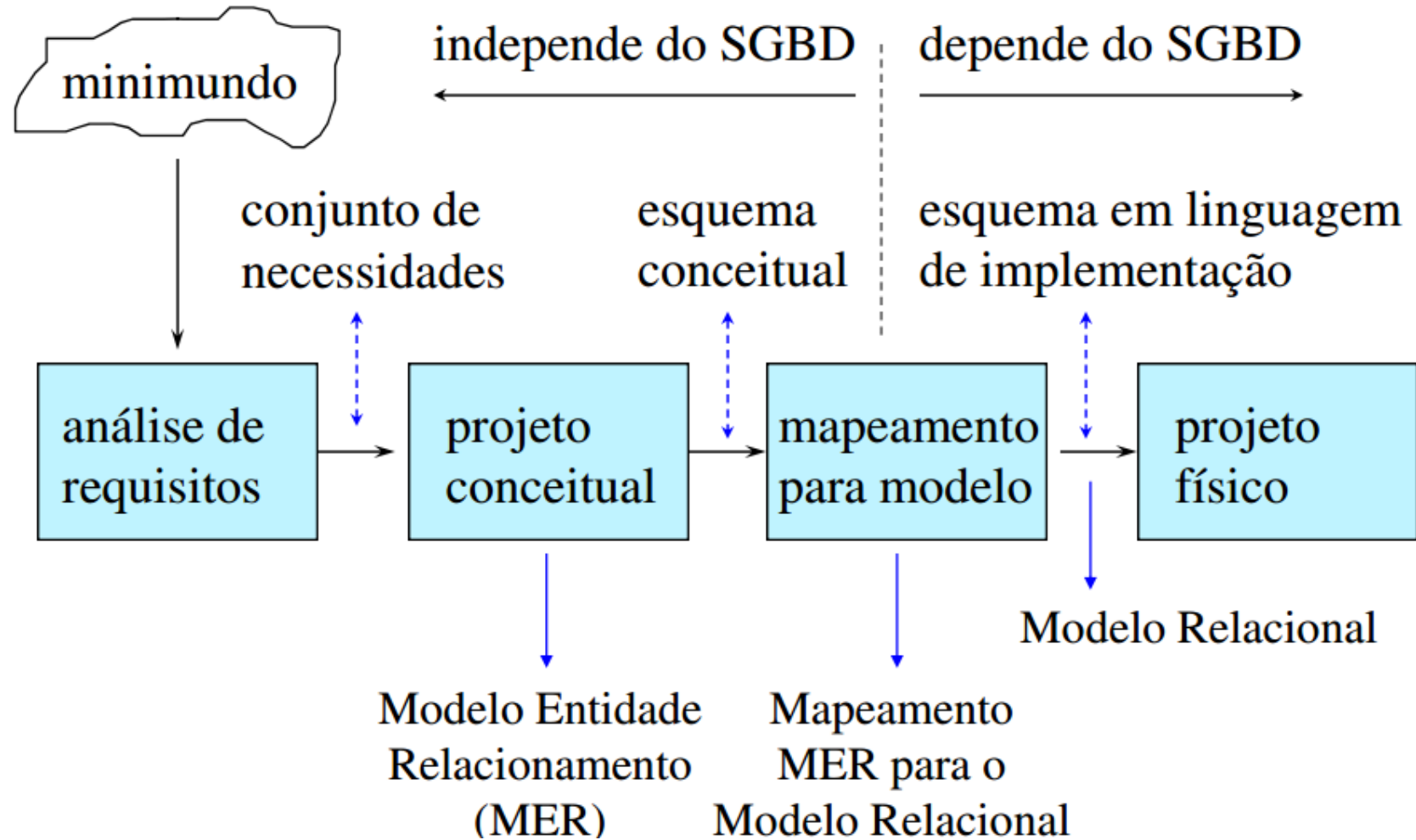
❑ **INTEGRIDADE REFERENCIAL** – mecanismo utilizado para manter a consistência das informações gravadas. Não permite a entrada de valores duplicados nem a existência de uma referência a uma chave inválida em uma entidade.

- Impede que durante a inclusão/exclusão/alteração uma chave estrangeira não tenha correspondente na chave primária da outra entidade.
- Na alteração/exclusão de dados verifica se há registros dependentes nas demais tabelas

❑ MAPEAMENTO ESQUEMA RELACIONAL – MODELO RELACIONAL

Mapeamento

Toda entidade se transforma
em tabela



MAPEAMENTO ESQUEMA RELACIONAL – MODELO RELACIONAL

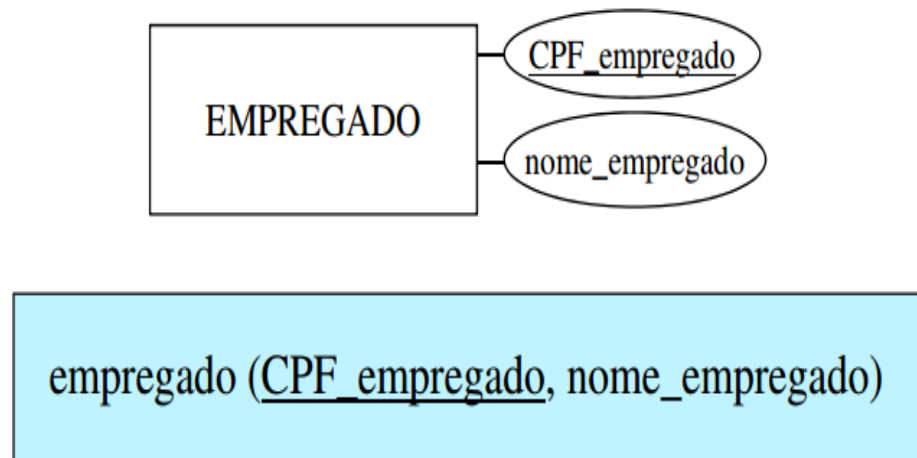
Um esquema ER pode ser transformado para o modelo Relacional por meio de regras de mapeamento.

❑ REGRA 1 – ENTIDADE

No esquema ER cada entidade regular será no modelo relacional representada através da criação de uma tabela (relação).

A tabela deve possuir o mesmo nome da entidade referenciada e deve conter os atributos simples.

TODA ENTIDADE EM TABELA

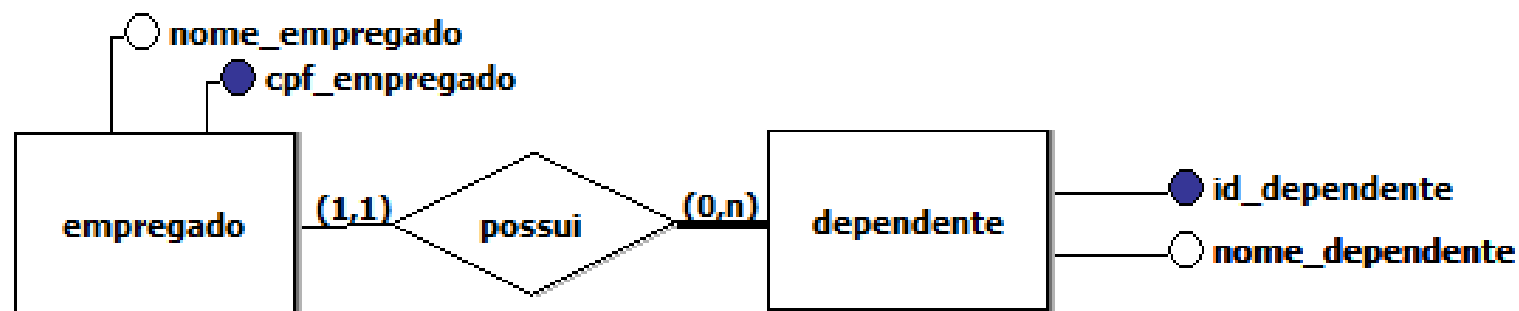


❑ REGRA 2 – ENTIDADE FRACA

No esquema ER cada entidade fraca será no modelo relacional criada uma tabela que incluirá os atributos simples da entidade fraca.

A entidade fraca receberá como **Chave Estrangeira** a Chave primária da tabela proprietária.

A **chave primária** da entidade fraca é a combinação da **chave primária** da entidade proprietária com o **identificador** parcial da entidade fraca.



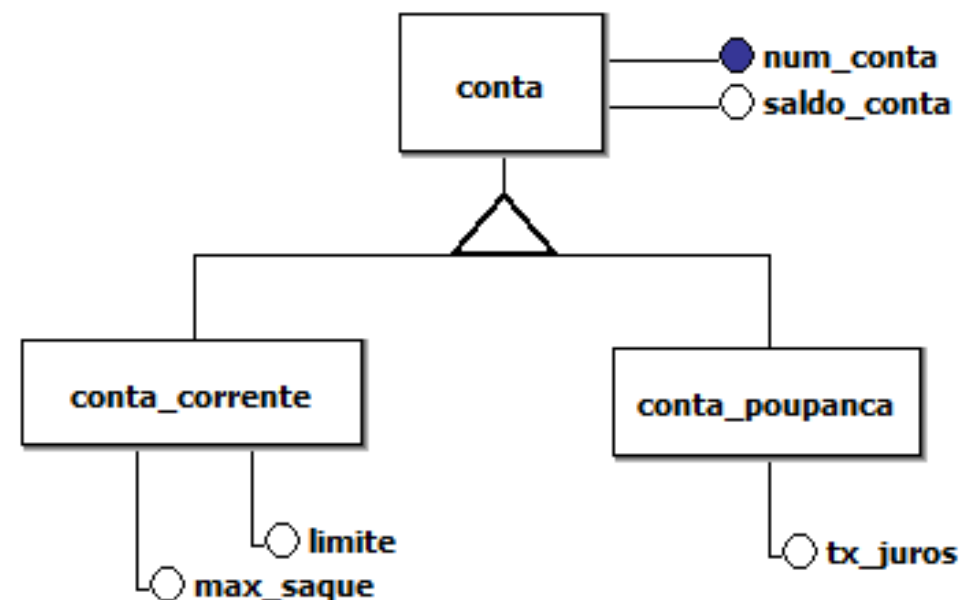
Empregado (cpf_empregado, nome_empregado)

Dependente (cpf_empregado, id_dependente, nome_dependente)

cpf_empregado referencia empregado

❑ REGRA 3 - GENERALIZAÇÃO/ESPECIALIZAÇÃO

Para cada hierarquia de generalização/especialização, incluir na tabela que representa a classe-base um atributo para a categoria da entidade. Criar também uma relação para cada subclasse, tendo como chave primária a chave da classe-base. Incluem-se nessas relações os atributos específicos daquela subclasse.



Conta (num_conta, saldo_conta, tipo_conta)

Conta_Corrente (num_conta, limite, max_saque)

Conta_Poupanca (num_conta, tx_juros)

REGRA 4 - TIPO DE RELACIONAMENTO 1:1

❑ Modelo entidade relacionamento

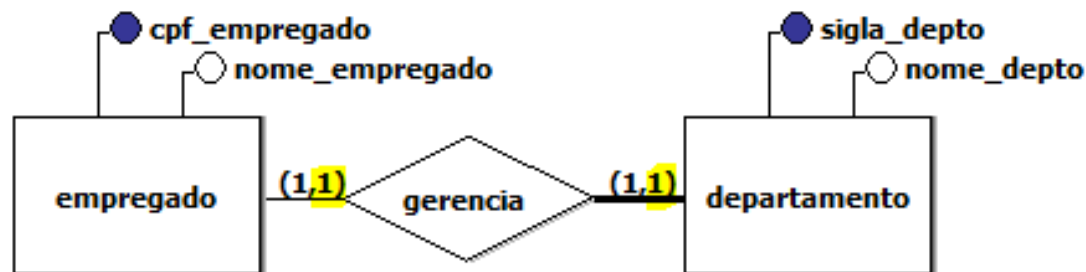
- tipo-relacionamento binário: E1 relacionando-se com E2
- cardinalidade: 1:1

❑ Modelo relacional (3 opções)

- repete-se a chave primária de E1 em E2 e vice versa
- repete-se a chave primária de E1 em E2
- repete-se a chave primária de E2 em E1

Chave estrangeira

- ✓ chave primária de uma relação que é inserida em outra relação utilizada para recuperar informações de outras relações



empregado (CPF_empregado, nome_empregado, sigla_depto)
departamento (sigla_depto, nome_depto, CPF_empregado)

empregado (CPF_empregado, nome_empregado)
departamento (sigla_depto, nome_depto, CPF_empregado)

empregado (CPF_empregado, nome_empregado, sigla_depto)
departamento (sigla_depto, nome_depto)

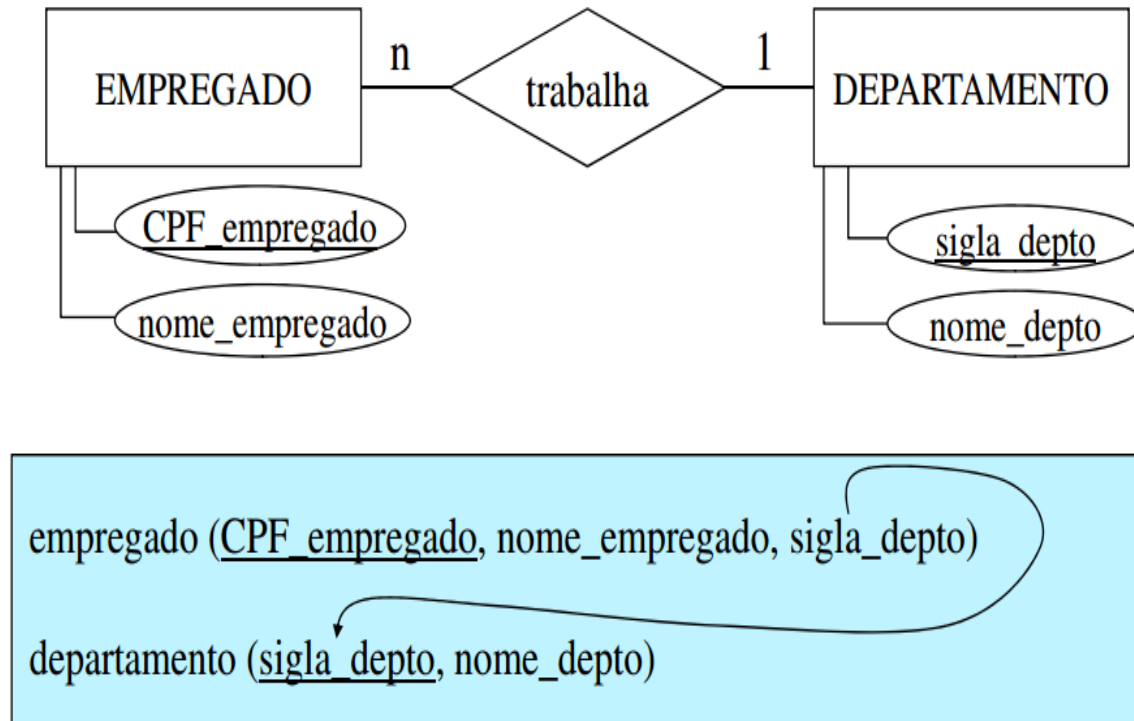
Não pode existir departamento sem gerente
Pode existir empregado que não gerencia o departamento

❑ REGRA 5 - TIPO DE RELACIONAMENTO 1:N

Os relacionamentos 1:N do esquema ER, identificam a tabela S que representa a entidade participante do lado N do relacionamento R.

Inclui-se como chave estrangeira em S a chave primária de T que representa a outra entidade participante de R.

Incluem-se também os atributos simples do relacionamento R como atributos de S.



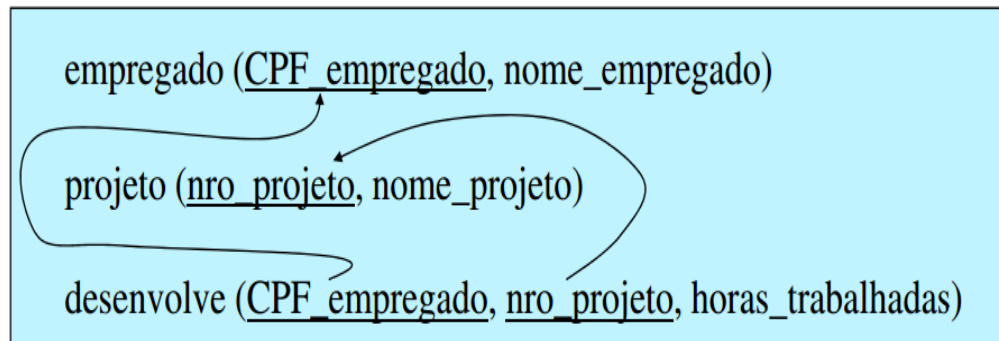
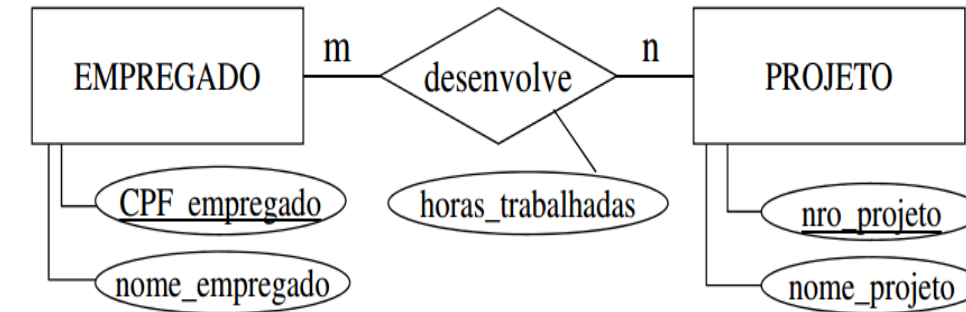
A entidade com N recebe a chave estrangeira.

❑ REGRA 6 - TIPO DE RELACIONAMENTO N:N

Com os relacionamentos N:N do esquema ER, cria-se uma nova tabela S que representa o relacionamento R.

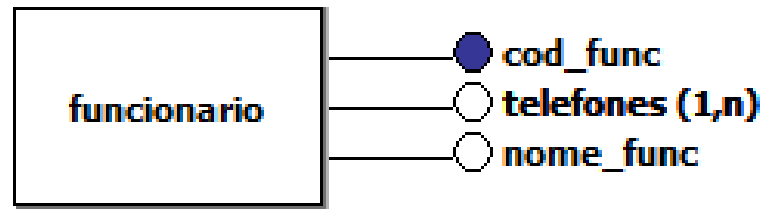
Inclui-se como chaves estrangeiras na nova tabela S, as chaves primárias das tabelas que representam as entidades participantes. Suas combinações formarão a chave primária da nova tabela S.

São incluídos também todos os atributos simples do relacionamento N:N como atributos de S.



❑ REGRA 7 - ATRIBUTO MULTIVALORADO

- ✓ Cada atributo multivalorado A, deve ser criada uma tabela R que inclui um atributo correspondente a A, mais a chave primária K da tabela que representa a entidade ou o relacionamento que tem A como atributo.
- ✓ A chave primária de R é a combinação de A e K.



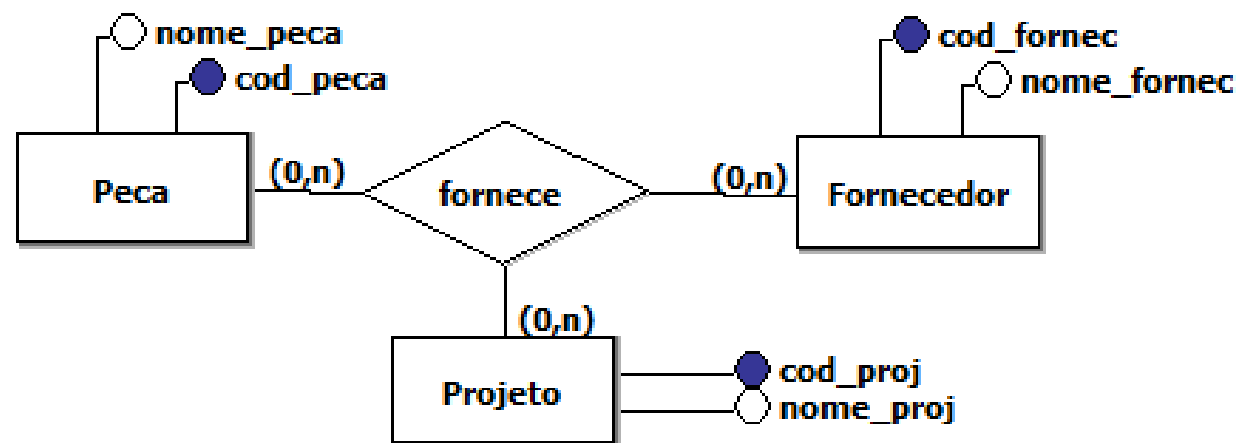
Funcionario (cod_func, nome_func)

Funcionario_Fones (cod_func, num_tel)

Cod_func referencia Funcionario

❑ REGRA 8 – RELACIONAMENTO N-ÁRIO

- ✓ Para cada relacionamento N-ário R, $N > 2$, cria-se uma nova tabela S para representar R.
- ✓ Incluem-se como chaves estrangeiras de S as chaves primárias das tabelas que representam as entidades participantes.
- ✓ Incluem-se também os atributos simples do relacionamento N-ário como atributos de S.
- ✓ A chave primária de S é, usualmente a combinação de todas as chaves estrangeiras que referenciam as relações representantes das entidades participantes.



Fornecedor (cod_fornec, nome_fornec)

Projeto (cod_proj, nome_proj)

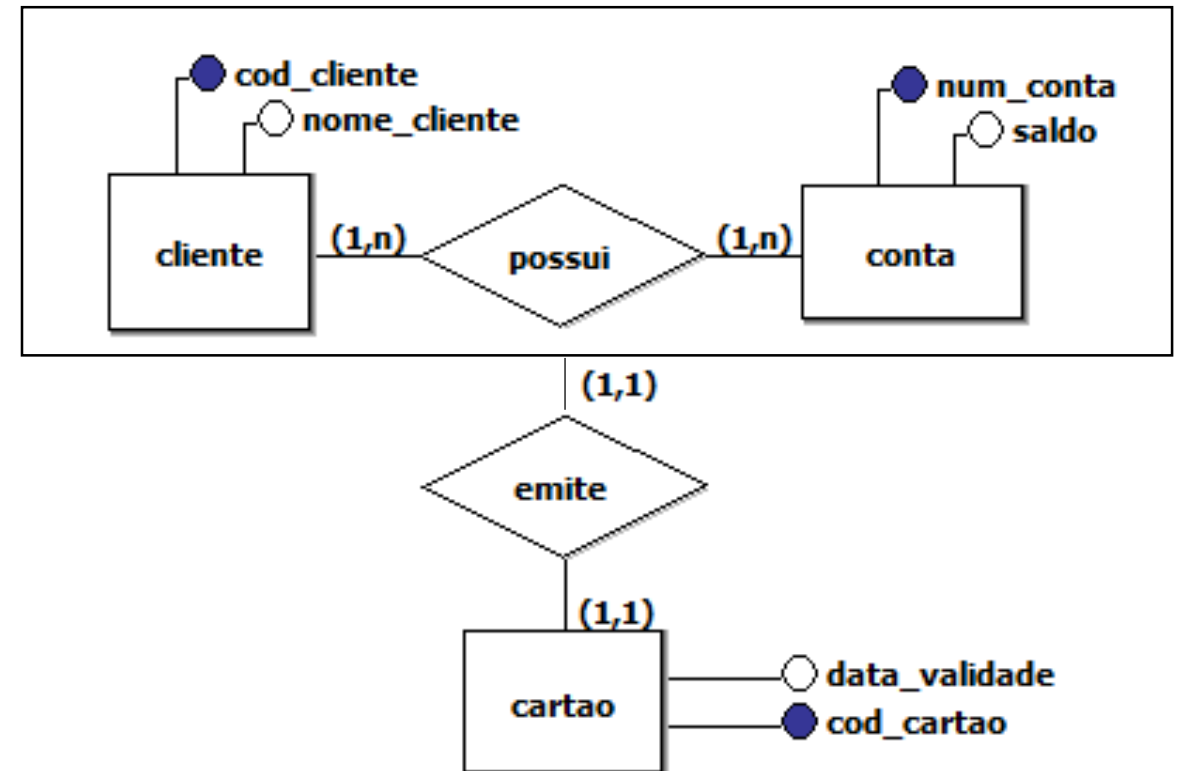
Peca (cod_peca, nome_peca)

Fornece (cod_fornec, cod_peca, cod_proj)

❑ REGRA 9 – RELACIONAMENTOS DE AGREGAÇÃO

Relacionamentos de agregação possuem duas abordagens de mapeamento:

1. **ABORDAGEM** - É mapear as entidades da agregação e o relacionamento. Como o relacionamento é sempre N:N, deverá ser gerada uma nova tabela. Esta tabela deve ser relacionada com a entidade ligada à agregação de acordo com sua cardinalidade.



Cliente (Cod_cliente, nome_cliente)

Conta (Num_conta, saldo)

Conta_Cliente (Cod_Cliente, Num_Conta, Num_Cartao)

Cartao (Num_Cartao, Data_Validade)

❑ REGRA 9 – RELACIONAMENTOS DE AGREGAÇÃO

2. ABORDAGEM - Consiste em desmembrar a agregação ainda no modelo ER gerando uma entidade intermediária resultante do relacionamento N:N da agregação. Esta entidade estará relacionada com as entidades da agregação e com a entidade externa à agregação de acordo com as cardinalidades.

Depois procede-se o mapeamento de acordo com as regras, desta vez sem agregação. Este método é útil, pois em alguns casos a simples colocação de chave estrangeira pode alterar o significado do problema.

Como o relacionamento entre a agregação e cartão é 1:1, então escolhe-se uma das entidades para receber a chave estrangeira.

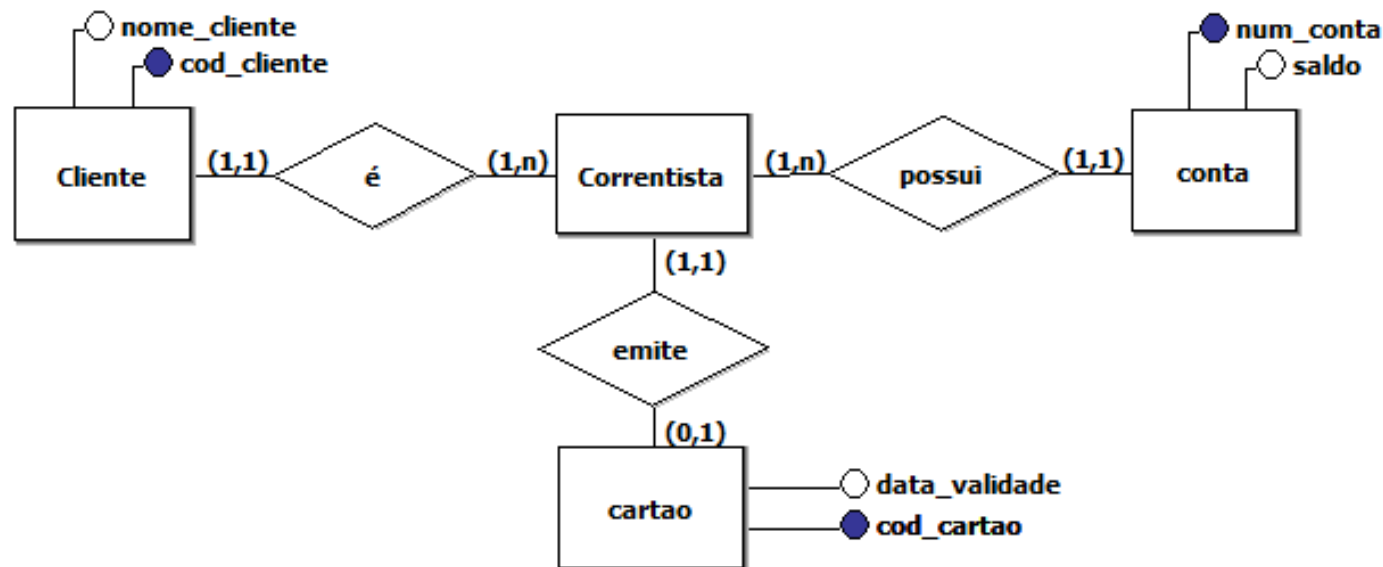
O modelo relacional fica:

Cliente (Cod_cliente, nome_cliente)

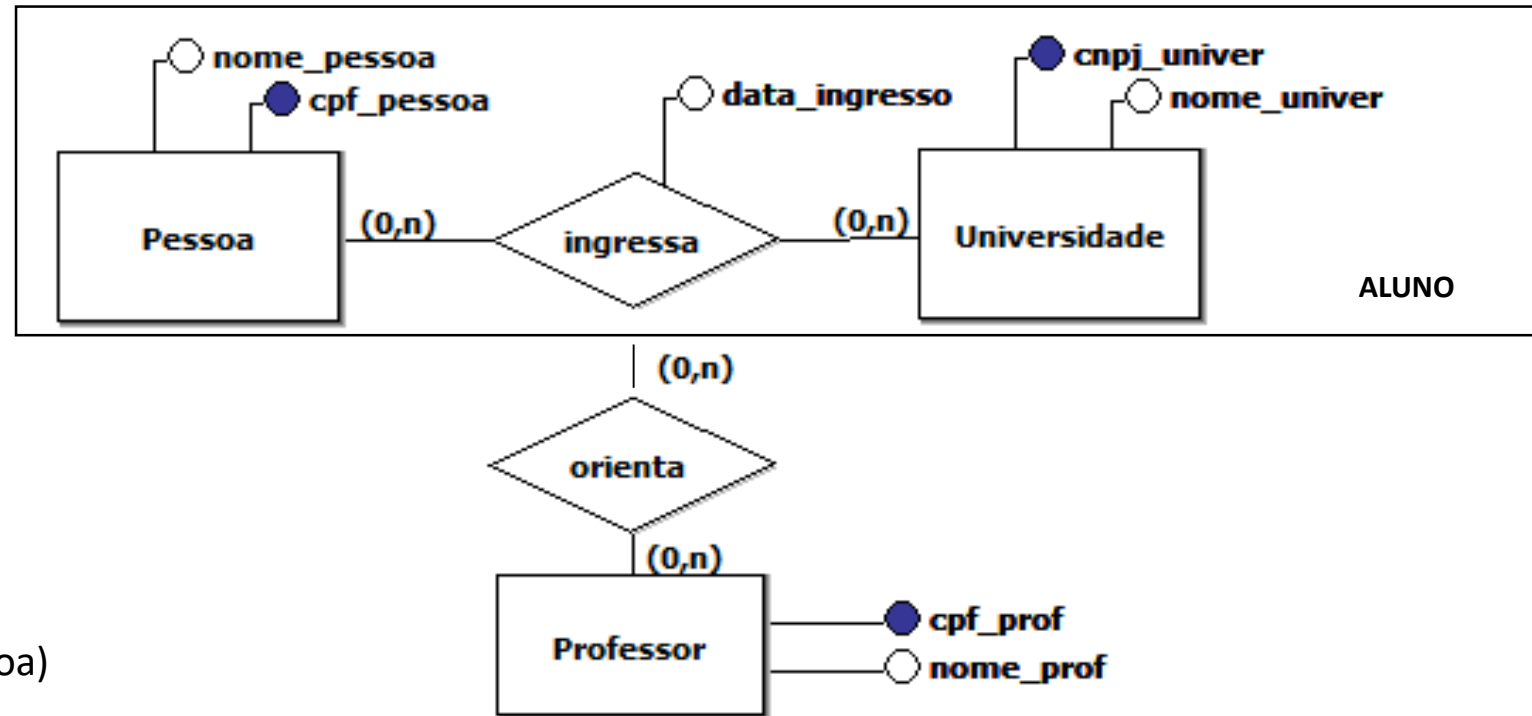
Conta (Num_conta, saldo)

Cartao (Cod_Cartao, Data_Validade)

Conta_Cliente (Num_Conta, Cod_Cliente, Cod_Cartao)



❑ REGRA 9 – RELACIONAMENTOS DE AGREGAÇÃO



❑ Mapeamento:

Pessoa (cpf_pessoa, nome_pessoa)

Universidade (cnpj_univer, nome_univer)

Ingressa (cpf_pessoa, cnpj_univer, data_ingresso)

Professor (cpf_prof, nome_prof)

Orienta (cpf_pessoa, cnpj_univer, cpf_prof)

❑ Dicionário de Dados

Tabela: cargo			
Nome do campo	Descrição	Tipo	Tamanho
id_cargo_	Guarda a identificação do registro	int, identity	-
nome_cargo_	Guarda o nome do cargo	varchar	80
data_cad_cargo_	Guarda a data do cadastro do cargo	varchar	8
hora_cad_cargo_	Guarda a hora do cadastro do cargo	varchar	8
resp_cad_cargo_	Guarda o nome do responsável pelo cadastro	varchar	80

❑ NORMALIZAÇÃO – Objetivo

Gerar um conjunto de esquemas de relação que permita armazenar informações sem redundância desnecessária, ao mesmo tempo permitindo recuperar informações facilmente.

Normalização envolve decomposição e esta decomposição tem de ser reversível de modo que nenhuma informação seja perdida no processo.

A normalização utiliza **projeção** para eliminar dependências funcionais complexas.
A **recomposição** é feita mediante operações de junção (**união**).

Projeto sem normalização pode ter:

- Repetição de informações;
- Incapacidade de representar certas informações

Exemplo

Para incluir um cliente tem que incluir uma conta válida ou null.

Cliente	
PK	<u>idCliente</u>
	nomeCliente descEndereco numConta

idCliente	nomeCliente	descEndereco	numConta
10	Eliana de Lima	R. Das flores, 143	089893893
11	Marcos André da Silva	R. João Martins	848478484
20	Elieser Candido	R. Gaivota	484848848

❑ Tabela Não Normalizada

- ✓ Possui uma tabela aninhada (chamada de grupo repetido ou coluna multivalorada ou coluna não atômica)
- ✓ Forma normal é uma regra que deve ser obedecida por uma tabela para que esta seja considerada “bem projetada”.
- ✓ 1FN – Tabela não contém tabelas aninhadas

Para transformar um esquema da tabela não-normalizada em um esquema na 1FN há duas alternativas:

- ❑ Construir uma única tabela com redundância de dados:

ProjEmp (codProj, Tipo, Desc, codEmp, Nome, cat, sal, dataIni, tempAl)

Nesta tabela, os dados do projeto aparecem repetidos para cada linha da tabela de Empregado.

- ❑ Construir uma tabela para cada tabela aninhada. Cria-se uma tabela para própria tabela que está sendo normalizada e uma tabela para cada tabela aninhada.

Proj (codProj, tipo, desc)

ProjEmp (codProj, codEmp, Nome, cat, sal, dataIni, tempAl)

NORMALIZAÇÃO

- ❑ Principais formas normais existentes:
 - 1FN
 - 2FN
 - 3FN
 - Forma normal de *Boyce/Codd* (FNBC) ou Nova 3FN
 - 4FN
 - 5FN

Estudaremos apenas até a 3FN.

Nota: As três primeiras formas normais atendem à maioria dos casos de normalização.

Uma forma normal engloba todas as anteriores, isto é, para que uma tabela esteja na 2FN, ela obrigatoriamente deve estar na 1FN e assim por diante.

Observe a tabela de **Habilidades_Esportivas**:

RA	Nome	Endereco	Habilidade
1230	Neymar	R. Santos	Futebol
1230	Neymar	R. Santos	Voleibol
1230	Neymar	R. Santos	Basquete
1230	Neymar	R. Santos	Atletismo

Esta tabela está mal projetada!

1. Se Neymar mudar de endereço? (anomalia de atualização)
2. Um novo esporte para Neymar? (anomalia de inclusão)
3. Retirar Neymar do Banco de Dados (anomalia de remoção)

O que pode ser feito?

NORMALIZAÇÃO

Esportista

RA	Nome	Endereco
1230	Neymar	R. Santos

Habilidades

RA	Habilidade
1230	Futebol
1230	Voleibol
1230	Basquete
1230	Atletismo

A repetição da coluna identidade é uma redundância necessária

Melhorando este modelo temos:

Esportista

RA	Nome	Endereco
1230	Neymar	R. Santos

Esportista_Habilidades

RA	codHab
1230	1
1230	2
1230	3
1230	4

Habilidades

codHab	Habilidade
1	Futebol
2	Voleibol
3	Basquete
4	Atletismo

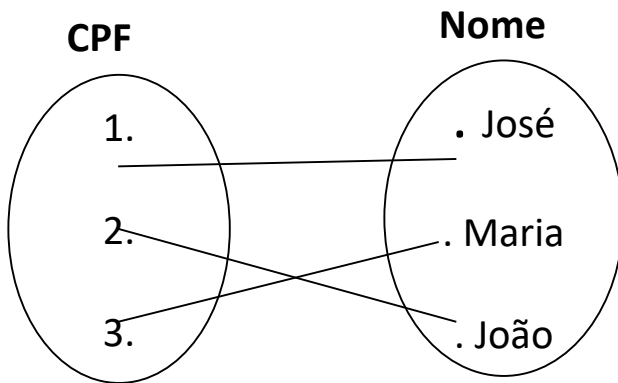
NORMALIZAÇÃO

❑ DEPENDÊNCIA FUNCIONAL

Seja **R** uma relação e sejam **X** e **Y** subconjuntos arbitrários do conjunto de atributos de **R**. Então, temos que **Y** é **funcionalmente dependente** de **X**, em símbolos:

X→Y (leia-se: **X determina funcionalmente Y**, se e somente se cada valor de **X** em **R** tiver associado a ele precisamente um valor **Y** em **R**.)

Exemplo: **A depende funcionalmente B** ou $B \rightarrow A$



Lê-se:

- B funcionalmente determina A
- A é dependente funcional de B
- A é função de B
- $B \neg A$, negação de $B \rightarrow A$
- Uma coluna pode depender funcionalmente de uma combinação de colunas: $(A,B) \rightarrow C$

NORMALIZAÇÃO - DEPENDÊNCIA FUNCIONAL

❑ REGRAS DEPENDÊNCIAS FUNCIONAIS

1. Separação (decomposição)

$A \rightarrow BC$ então $A \rightarrow B$
 $A \rightarrow C$

Ex. $CPF \rightarrow Nome, End$ então $CPF \rightarrow Nome$
 $CPF \rightarrow End$

Leia-se: se com um CPF obtêm-se o nome e o endereço, então com o CPF pode-se encontrar somente o nome ou somente o endereço.

2. Acumulação

$A \rightarrow B$ então $AC \rightarrow B$

Ex. $CPF \rightarrow end$, então $CPF, idade \rightarrow end$

Leia-se: Se com o CPF obtêm-se o endereço de uma pessoa, então com o mesmo CPF e a idade pode-se encontrar o endereço.

NORMALIZAÇÃO - DEPENDÊNCIA FUNCIONAL

❑ REGRAS DEPENDÊNCIAS FUNCIONAIS

3. Transitividade

$A \rightarrow B$ e $B \rightarrow C$ então $A \rightarrow C$

Ex.

$CPF \rightarrow cod_cidade$

$cod_cidade \rightarrow nome_cidade$, então $CPF \rightarrow nome_cidade$.

Leia-se: Se com o CPF obtêm-se o `cod_cidade` de uma pessoa e com o `cod_cidade` obtêm-se o `nome_cidade`. Então, com o CPF encontra-se o `nome_cidade`.

4. Pseudo-Transitividade

$A \rightarrow B$

$BC \rightarrow D$, então $AC \rightarrow D$

Ex.

$CPF \rightarrow cod_func$

$cod_func, mes \rightarrow sal_func$, então $CPF, mes \rightarrow sal_func$

Leia-se: Se com o CPF obtêm-se o `cod_func`, e com o `cod_func` e o `mês` tem-se o salário que ele recebeu, então com o CPF mais o `mês` pode-se obter o salário que ele recebeu no mês.

NORMALIZAÇÃO - DEPENDÊNCIA FUNCIONAL

❑ DECOMPOSIÇÃO SEM PERDA

A decomposição de uma relação diz-se sem perdas quando esta pode ser obtida a partir da junção natural das relações resultantes da decomposição.

A decomposição de uma relação $R=\{X, Y, Z\}$ (em que X , Y e Z são conjuntos de atributos) em $R1 (X,Y)$ e $R2(X,Z)$ é sem perda se $X \rightarrow Y$ ou se $X \rightarrow Z$.

Exemplo:

Seja a relação:

Fornecedor {F#, estado, cidade}

Primary key {F#}

* Decomposição 1

F1 {F#, estado}

F2 {F#, cidade}

Decomposição sem perda

* Decomposição 2

F1 {F#, estado}

F2 {cidade, estado}

Decomposição com perda.

Perde-se a dependência $F\# \rightarrow cidade$

NORMALIZAÇÃO - DEPENDÊNCIA FUNCIONAL

❑ Teorema de HEATH

Seja $R \{A, B, C\}$ uma RelVar, onde A, B, C são conjuntos de atributos. Se R satisfaz à dependência funcional $A \rightarrow B$, então R é igual à junção de suas projeções sobre $\{A, B\}$ e $\{A, C\}$.

Exemplo:

$R \{F\#, \text{status}, \text{cidade}\}$

$R1 \{F\#, \text{Status}\}$

$R2 \{F\#, \text{cidade}\}$

Conclusão: a decomposição da RelVar nas projeções $R1, R2, \dots, Rn$ é sem perda se R for igual a união de $R1, R2, \dots, Rn$.

- ❑ **Dependência Funcional Total (completa)** – ocorre quando uma coluna depende de todos os campos de uma chave primária composta.
- ❑ **Dependência Funcional Parcial** – ocorre quando uma coluna depende apenas de parte de uma chave primária composta
- ❑ **Dependência Funcional Transitiva** – ocorre quando um atributo ou conjunto de atributos A depende de outro atributo B que não pertence à chave primária, mas dependente funcionalmente dela.

Se um atributo X determina Y e se Y determina Z , podemos dizer que X determina Z de forma transitiva, isto é, existe uma dependência funcional transitiva de X para Z .

$\text{cidade} \rightarrow \text{estado}$

$\text{Estado} \rightarrow \text{país}$

$\text{Cidade} \rightarrow \text{país}$ (cidade determina país de forma transitiva)

NORMALIZAÇÃO - DEPENDÊNCIA FUNCIONAL

❑ NÃO NORMALIZADA

CodProj	Tipo	Desc	CodEmp	Nome	Cat	Sal	DataIni	TempAl
Proj01	Novo Desenvolvimento	Sistema	2146	João	A1	4	01/11/91	24
Proj02	Manutenção	Sistema de RH	3145	Silvio	A2	4	02/10/91	24

❑ PRIMEIRA FORMA NORMAL (1FN)

Proj (codProj, tipo, desc)

ProjEmp (codProj, codEmp, nome, cat, sal, dataIni, tempAl)

Proj

codProj	Tipo	Desc
Proj01	Novo Desenvolvimento	Sistema
Proj02	Manutenção	Sistema de RH

- ❑ Na 1FN a tabela não contém tabelas aninhadas.
- ❑ Valores atômicos

ProjEmp

codProj	codEmp	Nome	Cat	Sal	dataIni	TempAl
Proj01	2146	João	A1	4	01/11/91	24
Proj02	3145	Silvio	A2	4	02/10/91	24
Proj01	6126	José	B1	9	03/10/92	18
Proj01	1214	Carlos	A2	4	04/10/92	12
Proj02	8191	Mário	A1	4	04/01/91	12

NORMALIZAÇÃO - PRIMEIRA FORMA NORMAL (1FN)

❑ SEGUNDA FORMA NORMAL (2FN)

- Tabela está na **1FN** e não contêm dependências parciais (que dependem apenas de uma parte da chave primária composta).
- Tabela na 1FN que possui apenas uma coluna como chave primária já está na 2FN. O mesmo ocorre com tabela que contenha apenas colunas chave primária.

Exemplo

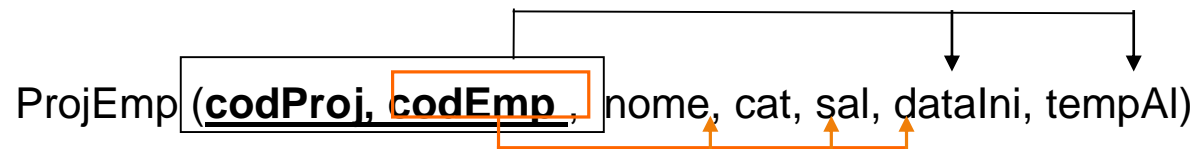
Proj (codProj, tipo, desc) está na 2FN

ProjEmp (codProj, codEmp, nome, cat, sal, dataIni, tempAl) **Pode ter dependências funcionais parciais.**

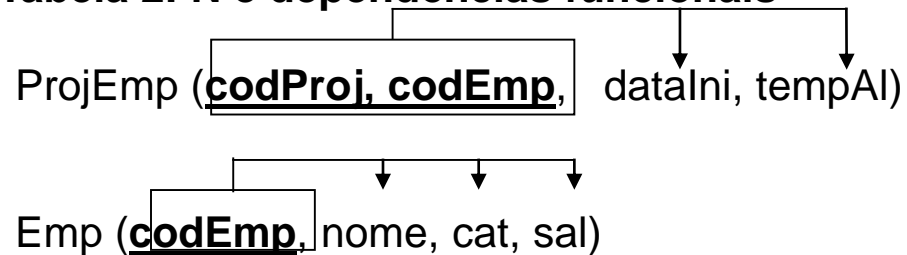
codEmp → nome, cat, sal

codEmp, codProj → dataIni, TempAl

1. **Tabela 1FN e dependências funcionais**



2. **Tabela 2FN e dependências funcionais**



Esquema Lógico

Proj (codProj, tipo, desc)

ProjEmp (codProj, codEmp, dataIni, TempAl)

Emp (codEmp, nome, cat, sal)

Proj

<u>codProj</u>	Tipo	Desc
Proj01	Novo Desenvolvimento	Sistema
Proj02	Manutenção	Sistema de RH

ProjEmp

<u>codProj</u>	<u>codEmp</u>	dataIni	TempAl
Proj01	2146	01/11/91	24
Proj02	3145	02/10/91	24
Proj01	6126	03/10/92	18
Proj01	1214	04/10/92	12
Proj02	8191	04/01/91	12

- ☐ Está na 1FN
- ☐ Não contém dependências parciais

Emp

<u>codEmp</u>	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4

NORMALIZAÇÃO - SEGUNDA FORMA NORMAL (2FN)

❑ TERCEIRA FORMA NORMAL (3FN)

Uma tabela está na 3FN quando está na 2FN e toda coluna não chave depende diretamente da chave primária, ou seja, quando não há **dependências transitivas ou indiretas**.

- ❑ **Dependência Funcional Transitiva ou Indireta** acontece quando uma coluna não chave primária depende funcionalmente de outra coluna ou combinação de colunas não chave primária.

3FN

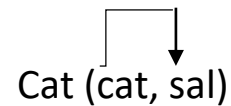
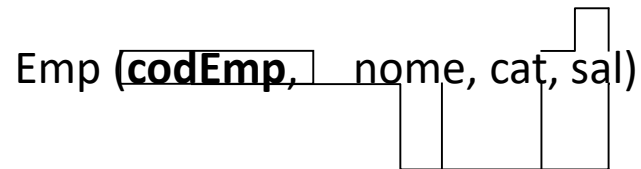
Proj (codProj, tipo, desc)

ProjEmp (codProj, codEmp, dataIni, TempAl)

Emp (codEmp, nome, cat)

Cat(cat,sal)

1. Tabela 2FN



NORMALIZAÇÃO - TERCEIRA FORMA NORMAL (FN)

Para cada coluna não chave fazer a pergunta:

“a coluna depende de alguma outra coluna não chave (dependência transitiva ou indireta)?”

Proj

<u>codProj</u>	Tipo	Desc
Proj01	Novo Desenvolvimento	Sistema
Proj02	Manutenção	Sistema de RH

Emp

<u>codEmp</u>	Nome	Cat
2146	Emanuel Soares	1

ProjEmp

<u>codProj</u>	<u>codEmp</u>	dataIni	TempAI
Proj01	2146	01/11/91	24
Proj02	3145	02/10/91	24
Proj02	8191	04/01/91	12

Cat

Cat	Sal
1	2300,00

NORMALIZAÇÃO - TERCEIRA FORMA NORMAL (FN)

❑ **EXERCÍCIO 1** : Considere a tabela abaixo:

ItemVenda (numNF, codTipoProd, numProd, descProd, dtVenda, codReg, codEmp, qtdItem, precoltem, nomeEmp, descTipoProd)

A tabela apresentada está na 1FN. Apresente a 2FN e a 3FN.

A passagem à 2FN consta da eliminação de dependências funcionais parciais, isto é, de colunas não-chave que dependem apenas de uma parte da chave primária e não da chave primária completa. No exercício, as dependências parciais são:

codTipoProd, numProd → descProd

codTipoProd → descTipoProd

numNF → dtVenda

numNF → codReg

numNF → codEmp

numNF → nomeEmp

ItemVenda (numNF, codTipoProd, numProd, qtdItem, precoltem)

Produto (codTipoProd, numProd, descProd)

TipoProd (codTipoProd, descTipoProd)

Venda (numNF, dtVenda, codReg, codEmp, nomeEmp)

1 FN

<u>numNF</u>	<u>codTipoProd</u>	<u>NumProd</u>	descProd	dtVenda	codReg	codEmp	qtdtem	precoltem	nomeEmp	descTipProd
100	1	1000	Sal	20110101	2010	121	543	2.5	Evandro Silva	Mercearia
200	2	343	Bola	20110215	2011	132	4	4,0	Manoela Souza	Brinquedo

NORMALIZAÇÃO - EXERCICIO 1 (RESOLVIDO)

2 FN

ItemVenda

<u>numNF</u>	<u>codTipoProd</u>	<u>numProd</u>	qtdItem	precoItem
100	1	100	543	2,5
200	2	343	4	4,0

Produto

<u>codTipoProd</u>	<u>numProd</u>	descProd
1	100	SAL
2	200	Bola

TipoProduto

<u>codTipoProd</u>	descTipoProd
1	Mercearia
2	Brinquedo

Venda

<u>numNF</u>	dtVenda	codReg	codEmp	nomeEmp
100	20110101	2010	121	Evandro Silva
200	20110215	2011	132	Manoela Souza

A passagem à 3FN consta da eliminação das dependências funcionais transitivas ou indiretas, isto é, de colunas não-chave que dependem de outras colunas não-chave.

Dependência transitiva: Tabela venda – codEmp → nomeEmp

3FN

ItemVenda (numNF, codTipoProd, numProd, qtdItem, precoItem)

Produto (codTipoProd, numProd, descProd)

TipoProd (codTipoProd, descTipoProd)

Venda (numNF, dtVenda, codReg, codEmp)

Empregado (codEmp, nomeEmp)

NORMALIZAÇÃO - EXERCICIO 1 (RESOLVIDO)

3 FN

ItemVenda

<u>numNF</u>	<u>codTipoProd</u>	<u>numProd</u>	qtdItem	precoItem
100	1	100	543	2,5
200	2	343	4	4,0

Produto

<u>codTipoProd</u>	<u>numProd</u>	descProd
1	100	SAL
2	200	Bola

TipoProd

<u>codtipoProd</u>	descTipoProd
1	Mercearia
2	Brinquedo

Venda

<u>numNF</u>	dtVenda	codReg	codEmp
100	20110101	2010	121
200	20110215	2011	132

Empregado

<u>codEmp</u>	nomeEmp
121	Evandro Silva
132	Manoela Souza

NORMALIZAÇÃO - EXERCICIO 1 (RESOLVIDO)

EXERCÍCIO 2 - No contexto de um sistema de controle acadêmico, considere a seguinte tabela:

Matricula (CodAluno, CodTurma, CodDisciplina, NomeDisciplina, NomeAluno, CodLocalNascAluno, NomeLocalNascAluno)

As colunas possuem o seguinte significado:

- CodAluno: código do aluno matriculado;
- CodTurma: código da turma na qual o aluno está matriculado (código é o identificador da turma)
- CodDisciplina: código que identifica a disciplina da turma;
- NomeDisciplina: nome de uma disciplina da turma;
- NomeAluno: nome do aluno matriculado;
- CodLocalNascAluno: código da localidade em que nasceu o aluno;
- NomeLocalNascAluno: nome da localidade em que nasceu o aluno.

Verifique se a tabela obedece a segunda e a terceira forma normais. Caso não obedeça, faça as transformações necessárias.

A tabela não se encontra na 2FN, pois contém dependências funcionais parciais. A passagem a 2FN resulta nas seguintes tabelas:

(CodAluno, CodTurma)

(CodAluno, NomeAluno, CodLocalNascAluno, NomeLocalNascAluno)

(CodTurma, CodDisciplina, NomeDisciplina)

As duas últimas tabelas não estão na 3FN, já que contém dependências transitivas. Ao eliminá-las, temos o seguinte modelo relacional.

(CodAluno, CodTurma)

(CodAluno, NomeAluno, CodLocalNascAluno)

(CodLocalNascAluno, NomeLocalNascAluno)

(CodTurma, CodDisciplina)

(CodDisciplina, NomeDisciplina)

EXERCÍCIO 3 – A figura 1 apresenta uma lista de todos os artigos submetidos a um congresso. No cabeçalho, aparece o código e o nome do congresso.

Posteriormente, em várias colunas são listados o código do artigo, seu título, seu assunto principal, seu código de autor e os códigos e nomes de vários autores do artigo. Observar que o mesmo código de artigo pode aparecer em diferentes congressos, já que a numeração de artigos inicia em um em cada congresso diferente.

Execute a normalização do documento, mostrando cada uma das formas normais.

NORMALIZAÇÃO - EXERCICIO 3 (RESOLVIDO)

Relação de artigos submetidos ao congresso

Congresso: DB25 – *Advances in Database Systems*

GTs promotores: GT3.1 – *Database Systems*

GT3.3 – *Database Conceptual Modeling*

Código do artigo	Título do artigo	Assunto principal	Código do autor	Nome do autor
1	<i>Semantic Integration in Heterogeneous Databases</i>	<i>Heterogeneous Databases</i>	2	Wen-Suan Li
2	<i>Providing Dynamic Security in a Federated Database</i>	<i>Heterogeneous Databases</i>	4 21	Chris Clifton N.B. Idris
3	<i>Efficient and effective clustering methods</i>	<i>Spatial databases</i>	7 32	W.A. Gray R.F. Churchhouse
4	<i>Automated Performance Tuning</i>	<i>Performance and Optimization</i>	12	Raymond R. Ng
5	<i>Bulk Loading into an OODB</i>	<i>Object oriented databases</i>	14 36	Jiawei Han Kurt. P. Brown
			1	Janet L. Wiener

Congresso: OO03 – *Object Oriented Modeling*

GTs promotores: GT4.6 – *Software Engineering*

Código do artigo	Título do artigo	Assunto principal	Código do autor	Nome do autor
1	<i>Temporal aspects in OO models</i>	<i>Temporal modeling</i>	2	Wen-Suan Li

Figura 1 – Lista de Artigos Submetidos ao Congresso

Não-Normalizada

(codCongr, nomeCongr, (codGT, nomeGT), (numArt, titArt, assPrincArt, (codAutor, nomeAutor)))

1 FN

(codCongr, nomeCongr)

(codCongr, codGT, nomeGT)

(codCongr, numArt, titArt, assPrincArt)

(codCongr, numArt, codAutor, nomeAutor)

2FN

(codCongr, nomeCongr)

(codCongr, codGT)

(codGT, nomeGT)

(codCongr, numArt, titArt, assPrincArt)

(codCongr, numArt, codAutor)

(codAutor, nomeAutor)

3FN = 2FN

EXERCÍCIO 3 – A figura 2 apresenta a definição de um arquivo convencional que armazena dados dos artigos submetidos aos vários congressos.

A definição do artigo está na linguagem COBOL, na qual muitos sistemas legados estão escritos. leitor poderá fazer este exercício mesmo que não conheça COBOL, apenas com base nas explicações a seguir. O arquivo contém um registro para cada artigo submetido ao congresso. Um artigo é identificado pelo código do congresso ao qual foi submetido e pelo seu código.

A descrição em COBOL nos informa que cada registro de artigo contém: o código do congresso, o nome do congresso, o código do artigo, o título do artigo, uma lista com os códigos e nomes de diversos autores, uma lista com código e nome dos temas de que trata o artigo e uma lista com código, nome e status da revisão dos vários especialistas que estão julgando ou julgaram o artigo.

As listas são especificadas em COBOL na forma de cláusulas OCCURS. Os campos NO-DE-AUTORES, NO-DE-TEMAS E NO-DE-REVISORES apenas servem para controlar as respectivas listas de campos.

```
FD  ARQ-ARTIGOS .
01  REG-ARTIGO .
    03  COD-CONGR
    03  NOME-CONGR
    03  NUMERO-ART
    03  TIT-ART
    03  NO-DE-AUTORES
    03  NO-DE-REVISORES
    03  NO-DE-TEMAS
    03  AUTOR OCCURS 1 TO 20 TIMES
        DEPENDING ON NO-DE-AUTORES .
        05  COD-AUTOR
        05  NOME-AUTOR
    03  TEMA OCCURS 1 TO 5 TIMES
        DEPENDING ON NO-DE-TEMAS .
        05  COD-TEMA
        05  NOME-TEMA
    03  REVISOR OCCURS 1 TO 5 TIMES
        DEPENDING ON NO-DE-REVISORES .
        05  COD-REVISOR
        05  NOME-REVISOR
        05  STATUS-REVISAO
```

Figura 2 – Arquivo COBOL que armazena os artigos submetidos

Portanto, não devem ser considerados na normalização, já e são redundantes. O campo de status da revisão pode conter dois valores diferentes e informa se o artigo já recebeu parecer do revisor ou não. Execute a normalização do documento, mostrando cada uma das formas normais.

As listas são especificadas em COBOL na forma de cláusulas OCCURS. Os campos NO-DE-AUTORES, NO-DE-TEMAS E NO-DE-REVISORES apenas servem para controlar as respectivas listas de campos.

Portanto, não devem ser considerados na normalização, já que são redundantes. O campo de status da revisão pode conter dois valores diferentes e informa se o artigo já recebeu parecer do revisor ou não.

Execute a normalização do documento, mostrando cada uma das formas normais.

Não-Normalizada

Obs: Os campos NO-DE-AUTORES, NO-DE-REVISORES e NO-DE-TEMAS contêm informações redundantes, servindo para controlar o número de ocorrências dos grupos repetidos AUTOR, TEMA e REVISOR. Por este motivo estes campos são eliminados já na passagem à forma Não-Normalizada.

(CodCongr, NumArt, NomeCongr, TitArt, (CodAutor, NomeAutor) (CodTema, NomeTema)
(CodRevisor, NomeRevisor, StatusRevisao))

1FN

(CodCongr, NumArt, NomeCongr, TitArt)
(CodCongr, NumArt, CodAutor, NomeAutor)
(CodCongr, NumArt, CodTema, NomeTema)
(CodCongr, NumArt, CodRevisor, NomeRevisor, StatusRevisao)

2FN

(CodCongr, NumArt, TitArt)
(CodCongr, NomeCongr)
(CodCongr, NumArt, CodAutor)
(CodAutor, NomeAutor)
(CodCongr, NumArt, CodTema)
(CodTema, NomeTema)
(CodCongr, NumArt, CodRevisor, StatusRevisao)
(CodRevisor, NomeRevisor)

2FN = 3FN

NORMALIZAÇÃO - EXERCICIO 3 (RESOLVIDO)

REFERÊNCIAS

ALVES, WILLIAM PEREIRA. Banco de Dados Teoria e Desenvolvimento. São Paulo; Érica, 2009.

ELMASRI, RAMEZ & NAVATHE, SHAMKANT – Sistemas de Banco de Dados – LTC – 2002.

GUIMARAES, Célio Cardoso. Fundamentos de Bancos de Dados: modelagem, projeto e linguagem SQL. Campinas, SP. Editora Unicamp, 2003.

HEUSER, CARLOS ALBERTO, Projeto de Banco de Dados, 5ª edição. Série Livros Didáticos da UFRGS; Sagra Luzzatto, 2004.

KORTH, F. & SILBERSCHATZ, A. Sistemas de banco de dados. São Paulo; Makron Books; 3a ed. 1999.

MACHADO, Felipe Nery Rodrigues. Projeto de Banco de Dados: uma visão prática. Editora Érica, 2009.

MACHADO, Felipe Nery Rodrigues. Projeto e Implementação de Banco de Dados. Editora Érica, 2008.

OLIVEIRA, Celso Henrique Pedroso. SQL Curso Prático. Editora Novatec, 2002.

SETZER, VALDEMAR W., SILVA, FLAVIO SOARES CORREA DA. São Paulo. 2005.

SENAC.DN. Modelagem de Dados.6.reimp/Leila Maria Pinheiro Fernandes;Gilda Ache Tavares.Rio de Janeiro:Senac Nacional, 2008.

TAKAHASHI, MANA., AZUMA, SHOKO, TREND-PRO CO, LTD Guia mangá de banco de dados.