Implementar Banco de Dados para Web

UC8

Dados X Informação

Dados: São fatos em uma forma primária, que podem ser armazenados em algum meio.

Ex.: CPF, Nome, Data

Informação: São os fatos organizados de maneira a produzir um significado -> Dados colocados em contexto.

EX.: Lista de clientes com seus números de CPF, ordenados.

Metadados

- Definimos como sendo "Dados sobre os dados".
- Permitem efetuar a representação e identificação dos dados, garantindo sua consistência e persistência.
- Os Metadados são mantidos no Dicinário de Dados (ou. Em um catálogo de Dados).
- Seria a **representação dos dados** no Banco de Dados

Banco de Dados

O banco de dados nada mais é que um **repositório para armazenar informações (dados) de qualquer natureza.** Ele retrata aspectos do mundo real — ou seja, o conceito visto anteriormente em regras de negócios — em que qualquer mudança que se faça no negócio é diretamente replicada no banco de dados.

Um BD é uma coleção organizada de dados.

É composto por diversos objetos, tais como: tabelas, esquemas, visões, consultas, relatório, procedimentos, triggers (gatilhos) entre outros.

Os bancos de dados podem ser **categorizados** de várias maneiras, sendo que uma das principais delas aponta para a sua arquitetura, a qual pode ser centralizada, descentralizada, distribuída e replicada.

Centralizada	Arquitetura em um servidor em que as aplicações ou os clientes podem acessar o banco de dados. Logo, a responsabilidade pela capacidade de armamento e resposta será assumida. Para tal modelo, é necessário alto poder de processamento do servidor e um ótimo desempenho do SGBD.
Descentralizada	Há mais de um servidor para o banco de dados, possibilitando que eles sejam descentralizados. Tem-se, então, garantia de autonomia local e auxílio na comunicação, que, por exemplo, pode ser melhor distribuída entre as aplicações que utilizam tabelas específicas.
Distribuída	Arquitetura em que os dados estão compartilhados em muitos computadores ou servidores, com atualizações ou sincronismo para se certificar da integridade desses dados nos locais em que se encontram.
Replicada	Arquitetura em que o banco de dados é copiado para muitos computadores ou servidores, como em uma metodologia de espelhamento. Todos os bancos são idênticos e, conforme há alterações no primeiro banco, os demais são alterados em cascata. Aliás, esse tipo de arquitetura garante a segurança das informações. No caso de haver falhas em algum *host*, outro poderá assumir seu lugar, visto que os dados são idênticos.

Aplicações

- Sistemas bancários
- Reservas em hotéis
- Controle de estoques em supermercados
- Catálogo de livros em bibliotecas
- E-commerce
- Receita Federal
- YouTube

Sistema de Gerenciamento de Banco de Dados

SGBD

Definição

- Uma coleção de softwares que permite aos usuários criarem e manterem um ou mais bancos de dados.
- São usados para definição, construção, manipulação e compartilhamento dos bancos de dados entre aplicações e usuários.
- Permitem manter BD ao longo do tempo.
- Além disso, no que diz respeito aos dados armazenados, um SGBD traz muitas funcionalidades, como a segurança contra acessos maliciosos ou não permitidos, a proteção contra falhas sistêmicas e de hardware, assim como a distribuição dos dados entre várias aplicações e diversos usuários.

Exemplos de SGBDs

- Oracle Database
- PostgreSQL
- **MySQL**
- MS SQLServer
- IBM DB2
- SAP Sybase
- MongoDB
- Teradata
- SQLite

SGDB e seus quatro elementos

- **Dados**

São todas as informações armazenadas em um banco de dados.

- **Software**

Entre os usuários e o banco de dados físico (dados armazenados), há uma camada de software (programas de aplicação), sendo este o SGBD.

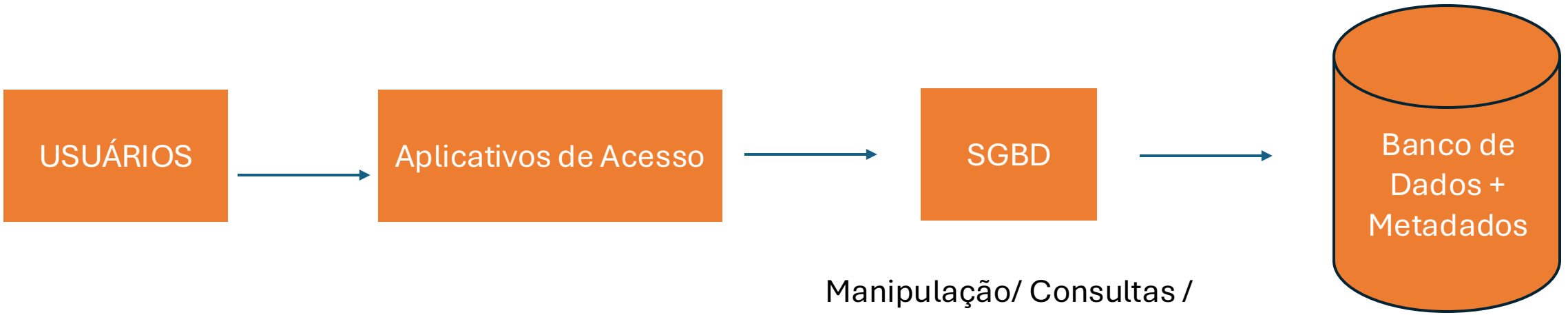
SGDB e seus quatro elementos

- **Hardware**

Refere-se aos volumes de memória secundária (tambores, discos etc.) em que se encontra o banco de dados junto com os dispositivos a ele vinculados, como canais, unidades de controles, entre outros.

- **Usuários**

Existem três tipos principais, estando no topo o **programador / desenvolvedor** (usuário que escreve os programas de aplicação que utilizam o banco de dados); seguido do **usuário final** (tem acesso a um terminal, podendo usar uma linguagem de consulta ou fazer uma chamada para uma aplicação, realizando todas as funções de recuperação, criação, eliminação ou alteração); e, para finalizar, há o usuário que é o **administrador do banco de dados**, também conhecido por ***DBA***.


USUÁRIOS
Aplicativos de Acesso
SGBD
Banco de Dados + Metadados
Manipulação/ Consultas / Definição de Dados

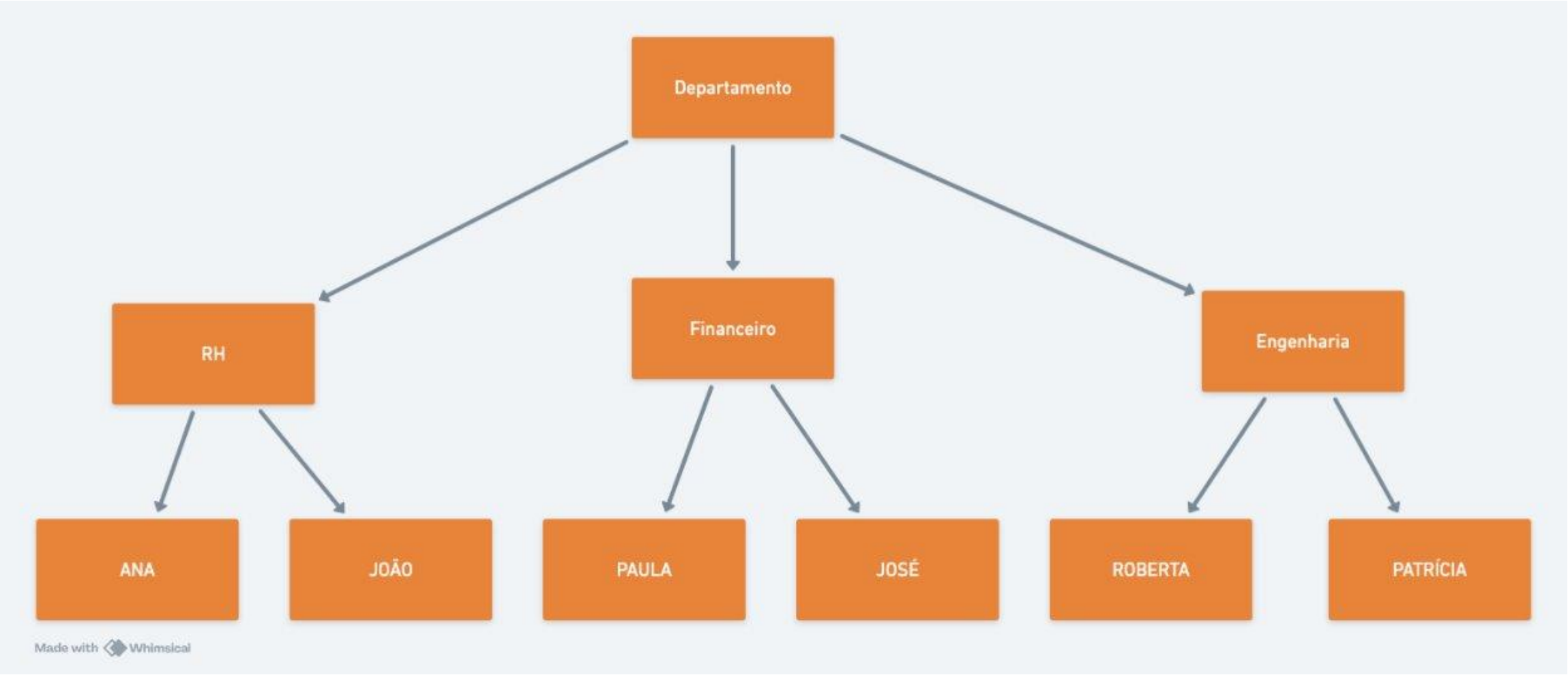
Características e Funcionalidades

- Controle de Redundância (evitar a duplicidade dos dados)
- Múltipas Visões dos Dados
- Controle de Concorrência (dois usuários acessando ao mesmo tempo)
- Backup e Restauração
- Autenticação e Autorização de acesso
- Restrições de Integridade (o que é necessário para cadastrar os dados)

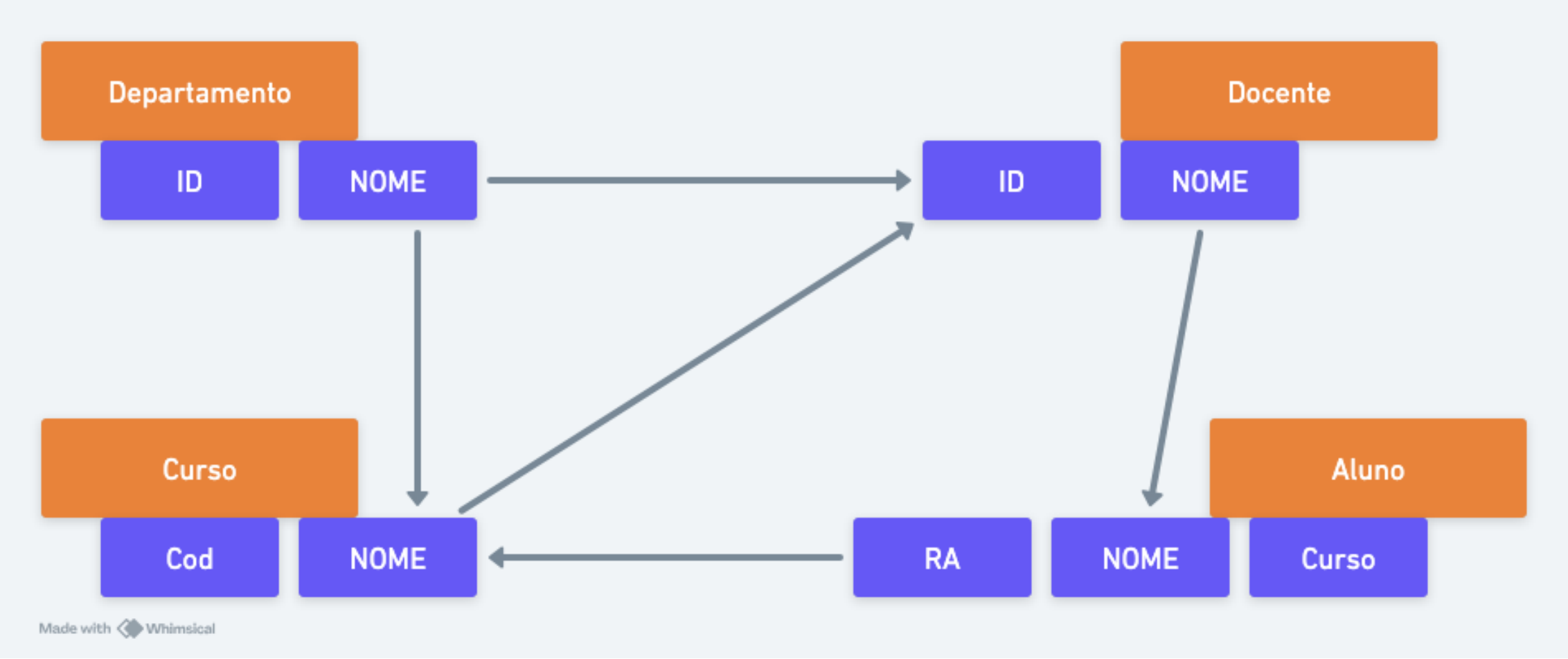
História

Modelos de Banco de Dados

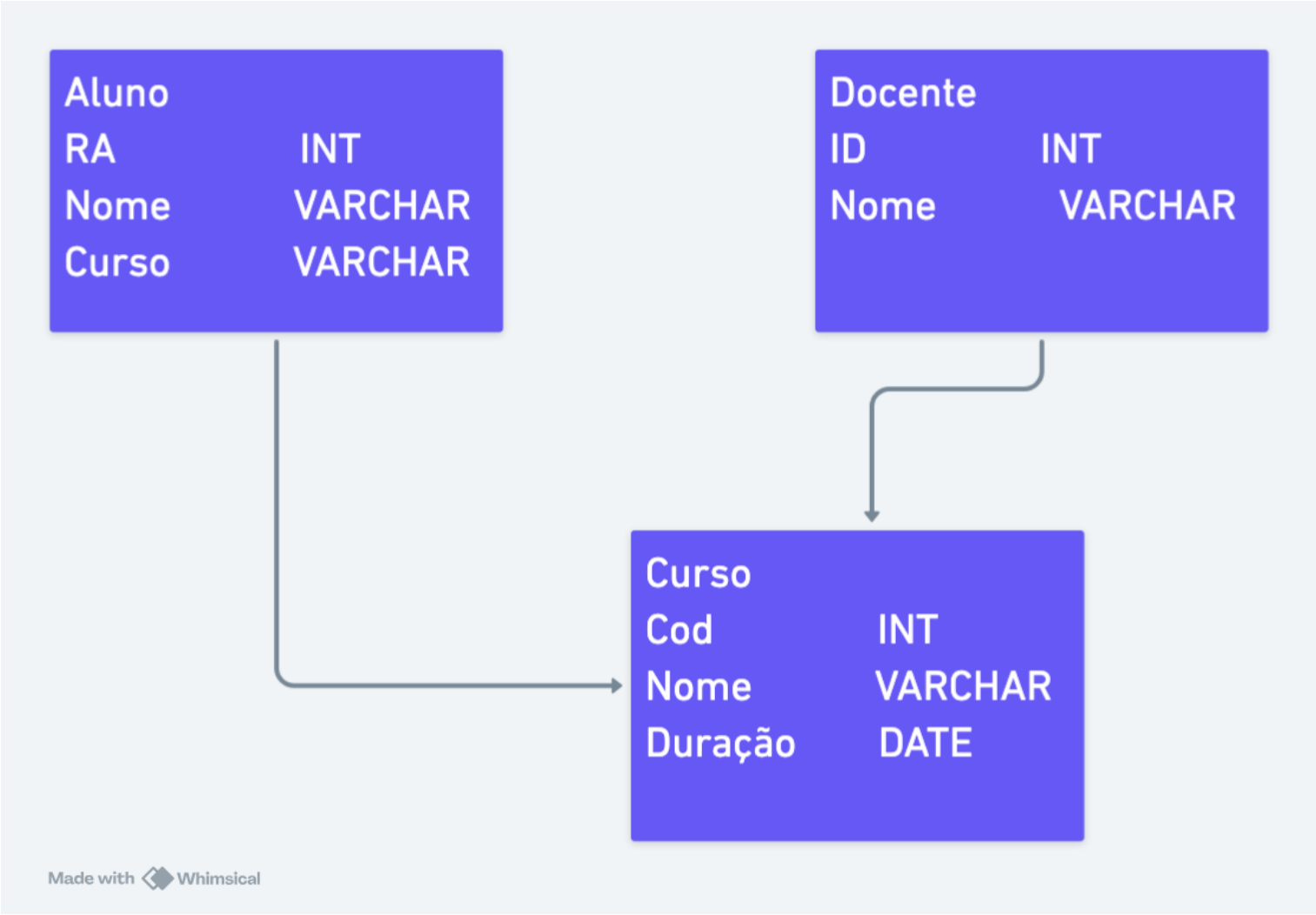
Papel



Hierárquico


Departamento
ID
NOME
Docente
ID
NOME
Curso
Cod
NOME
RA
NOME
Aluno
Curso
Made with Whimsical

Rede



Relacional

Os dados são organizados em coleções de tabelas bidimencionais.

As tabelas são chamadas de relações, organizadas em linhas e colunas.

Componentes:

Tabela: estrutura básica de armazenamento

Tupla: linha ou registros

Coluna: unidade que arma zena um tipo específico de dado(valor)

Relacionamento: associaç ão entre as entidades.

Modelo conceitual

O modelo conceitual utiliza a análise de requisitos por base, sendo um grupo de hipóteses que leva em conta o mundo real, indicando as regras de negócio do sistema. Desse modo, consiste no detalhamento de alto nível, voltado a entender e descrever todos os detalhes de uma organização. Esse detalhamento das informações de negócio, por sua vez, é guardado no banco de dados.

Objetivo

Implementação de um sistema com informações coerentes referentes aos elementos que o compõem, como seus relacionamentos, suas propriedades e seus objetos. De modo geral, tais elementos precisam estar bem estruturados para que o usuário entenda o seu propósito.

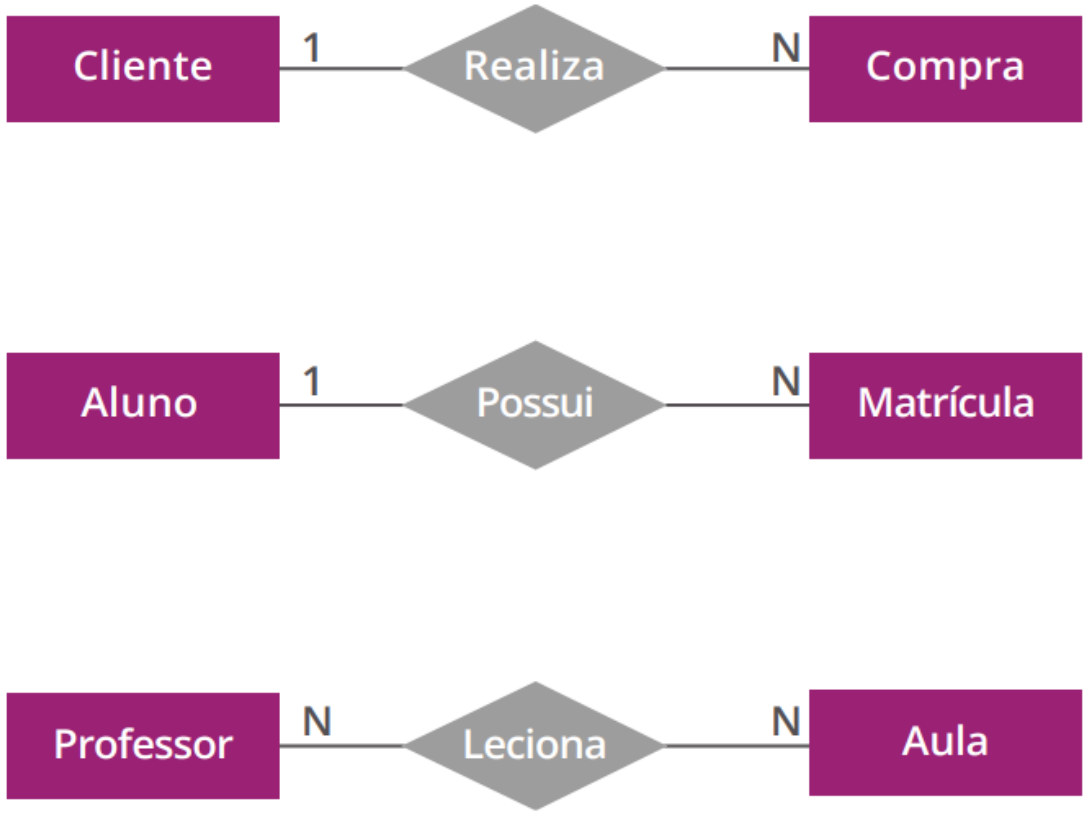
É importante frisar que esse modelo não foca em questões como a abordagem do banco de dados que será escolhida, nem em como será acessado ou quais estruturas físicas serão desenvolvidas pelo SGBD.

Ex:

Cadastro de Produtos em uma Loja

Dados necessários: Nome do produto, categoria de produto (limpeza, higiene, etc), código do fornecedor, tipo de embalagem, tamanho, quantidade.

Representação Gráfica

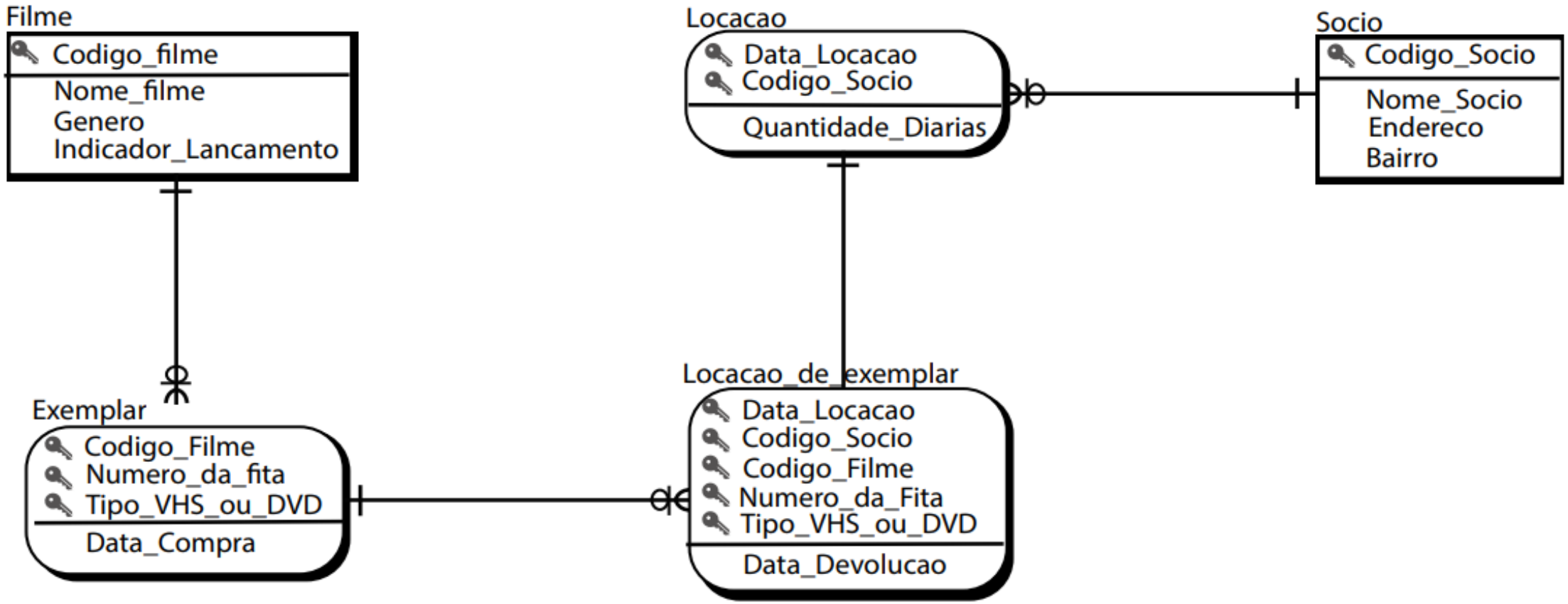


Modelo lógico

Ele detalha a forma lógica das estruturas, como **relações**, **entidades**, **tipo de dados**, **hierarquia**, entre outras particularidades.

Estes elementos existem no banco de dados conforme a possibilidade de sua abordagem, sem levar em conta os detalhes de determinado SGBD.

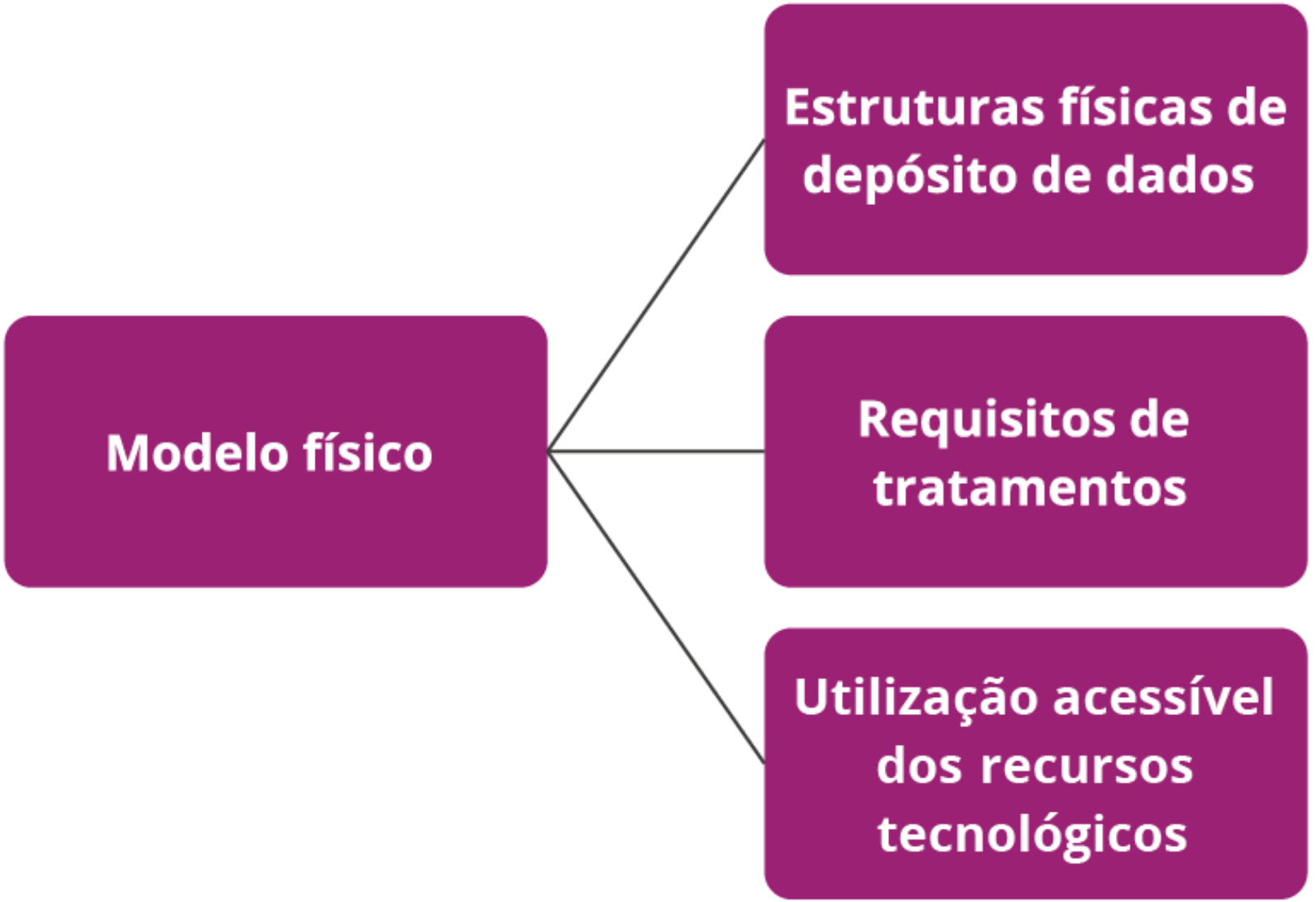
Representação Gráfica



Modelo físico

O modelo físico só será criado quando tivermos o modelo lógico, com o detalhamento das estruturas físicas de depósito dos dados, como tipos e tamanhos dos campos. Geralmente, esse modelo é projetado seguindo os requisitos de tratamentos, considerando a utilização acessível em relação aos meios de recursos tecnológicos disponíveis para o projeto.

O modelo físico consiste na fase final, ou seja, aquela que antecede a criação de banco de dados, considerado como um aperfeiçoamento do modelo lógico.



Modelo Entidade-Relacionamento

MER

Trata-se de um modelo conceitual usado para descrever objetos envolvidos no domínio de um sistema a ser construído, incluindo seus atributos e relacionamentos.

O MER permite representar de forma abstrata e estruturada que irá construir o banco de dados.

Composto pelos seguintes objetos:

- Entidades
- Atributos
- Relacionamentos

Diagrama Entidade-Relacionamento

DER

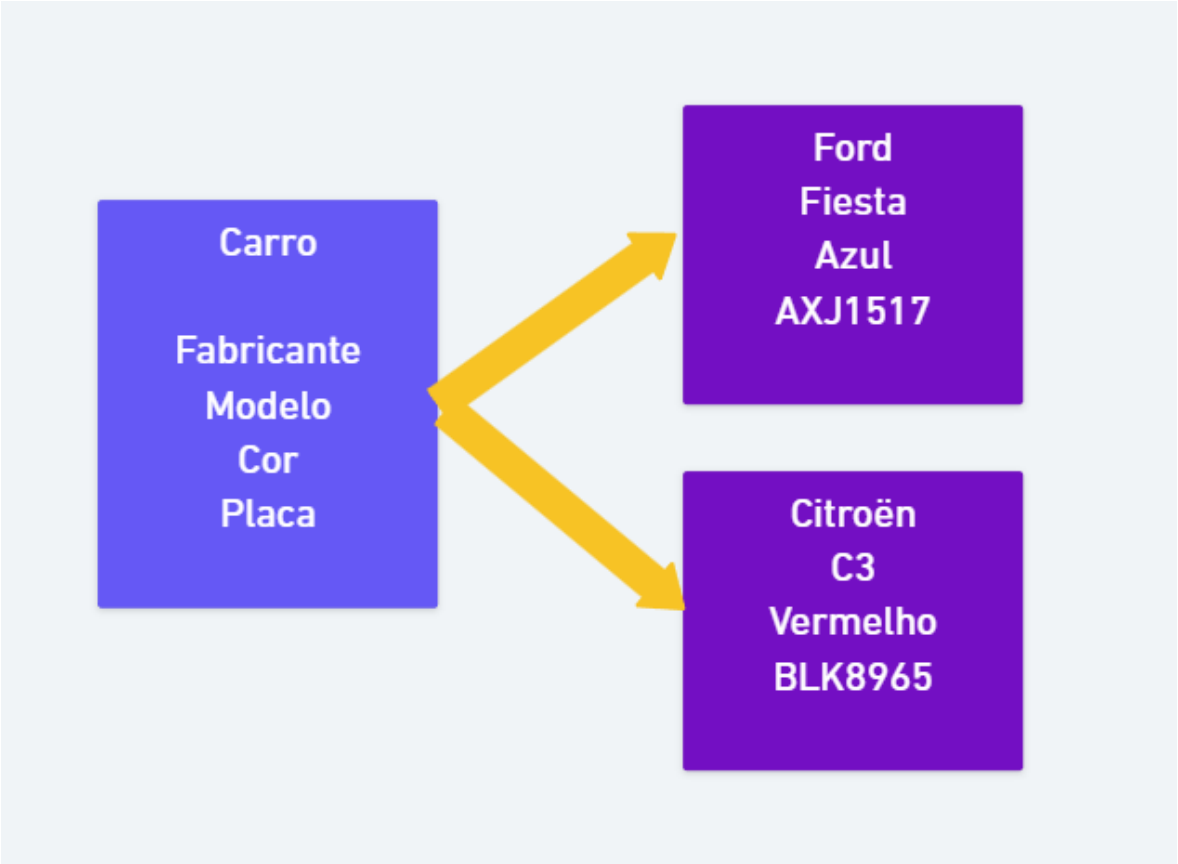
Representação gráfica associada ao MER.

- Retângulos - representam as entidades
- Elipses – representam atributos
- Losangos – representam relacionamentos
- Linhas – ligam atributos a entidades e entidades a relacionamentos.

Instância de Entidade

Uma entidade em si é uma descrição de estrutura e formato das ocorrências da entidade, como um "receita", ou "planta". Uma instância de entidade é uma ocorrência específica de uma entidade.

Ou seja cada objeto de uma entidade é denominado de **Instância de Entidade.**



Entidades - algumas regras de nomeação

- Nomes de Entidades:
  - Devem começar com uma letra;
  - Usar palavra no singular;
  - Não podem ter espaços ou alguns caracteres especiais;
  - Alguns caracteres como "$", "#" e "_" são permitidos em alguns bancos de dados;
  - Os nomes de colunas devem ser únicos dentro de uma tabela;
  - Os nomes de entidades/tabelas devem ser únicos dentro do esquema.

Atributos

Descrevem características da entidade, como por exemplo: fabricante, modelo, cor, placa....

Os atributos possuem um tipo de dados (domínio) nome e valor específico.

Tipos de atributos:

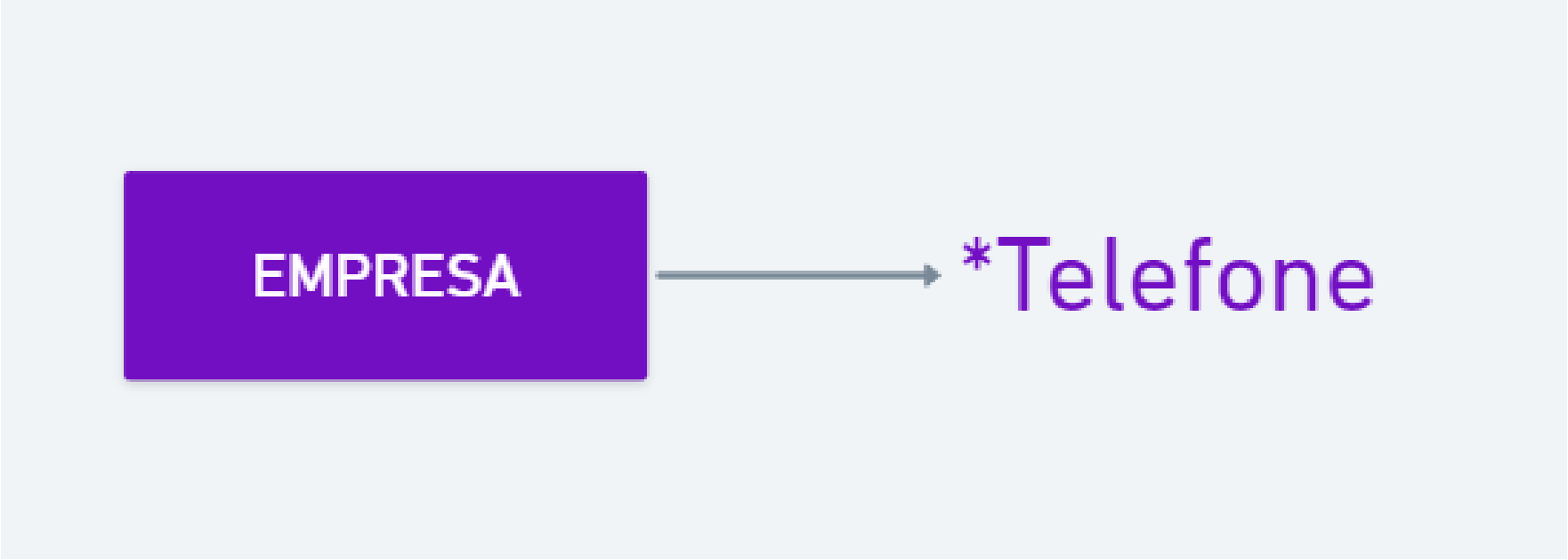
- Simples
- Composto
- Multivalorado
- Determinante
- Identificador

Atributo Composto

Formado por itens menores, pode ser subdividido em outros atributos .

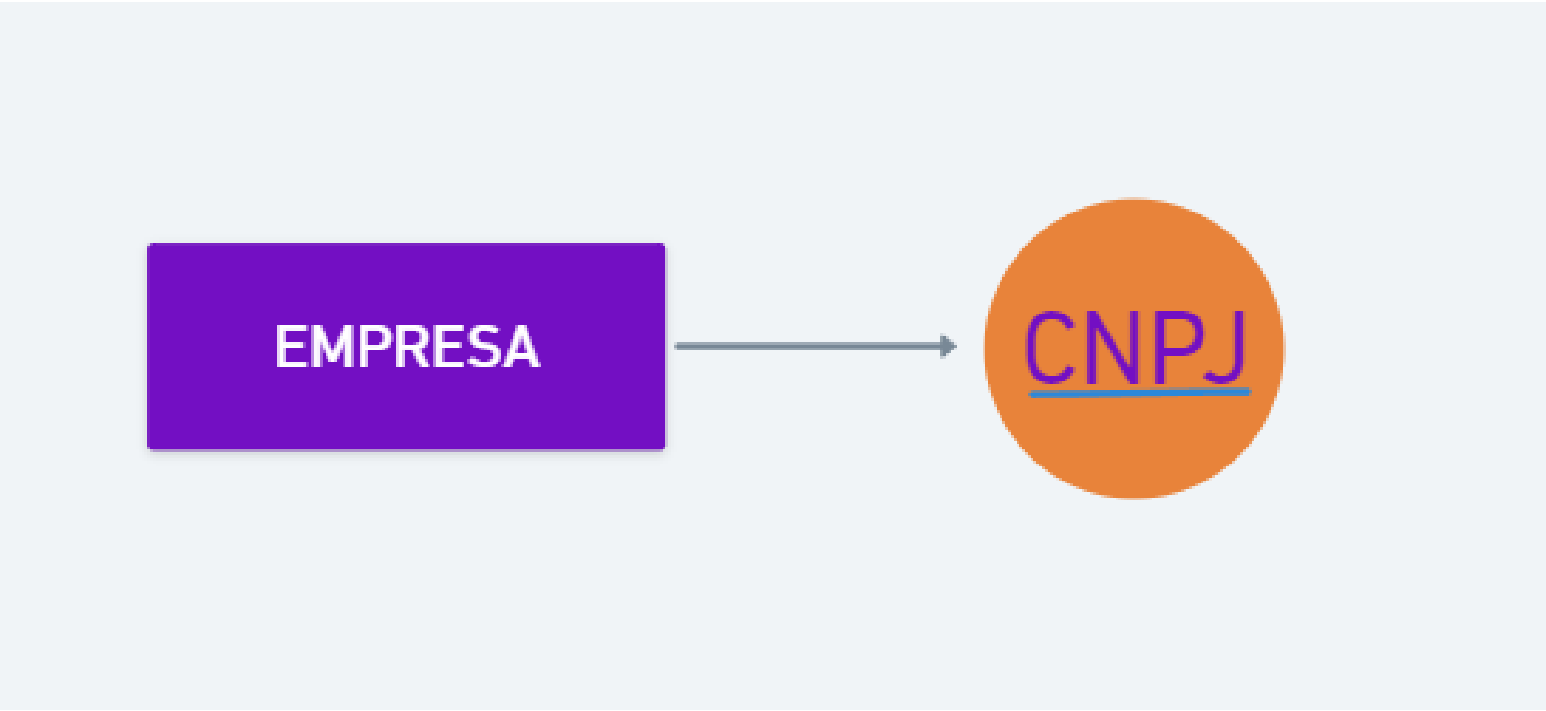
Atributo Multivalorado

Pode conter mais de um valor para o mesmo registro (informação)



Atributo Determinante

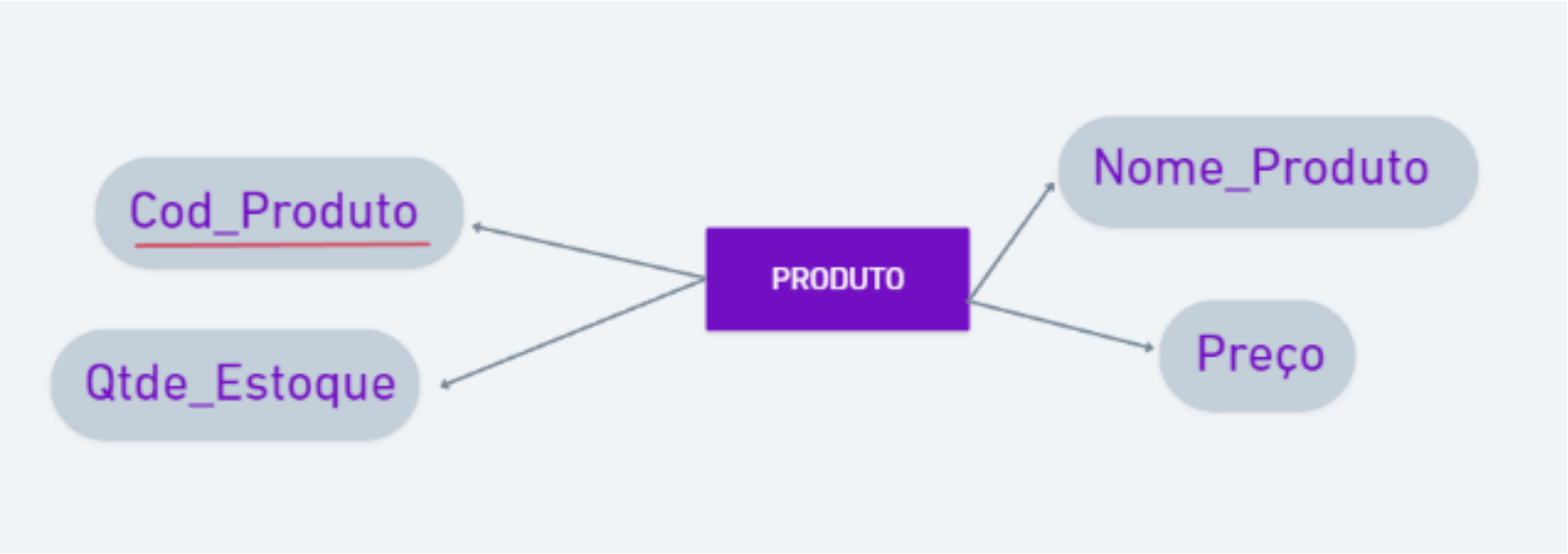
Define de forma única as instâncias de uma entidade.
Não podem existir duas instâncias com o
mesmo valor nesse atributo.



Atributo Identificadores (Chaves)

Uma chave identifica uma instância especifica na classe de entidade.

Produto(Cod_produto, Nome_produto, Preço, Qtde_Estoque)



	Produto
PK	Cod_Produto
	Nome__Produto
	Preço
	Qtde_Estoque

Chave Primária

Identifica de forma **exclusiva** os registros em uma tabela, não podendo ter repetição de valores nem tampouco valor nulo.

Primary Key / PK

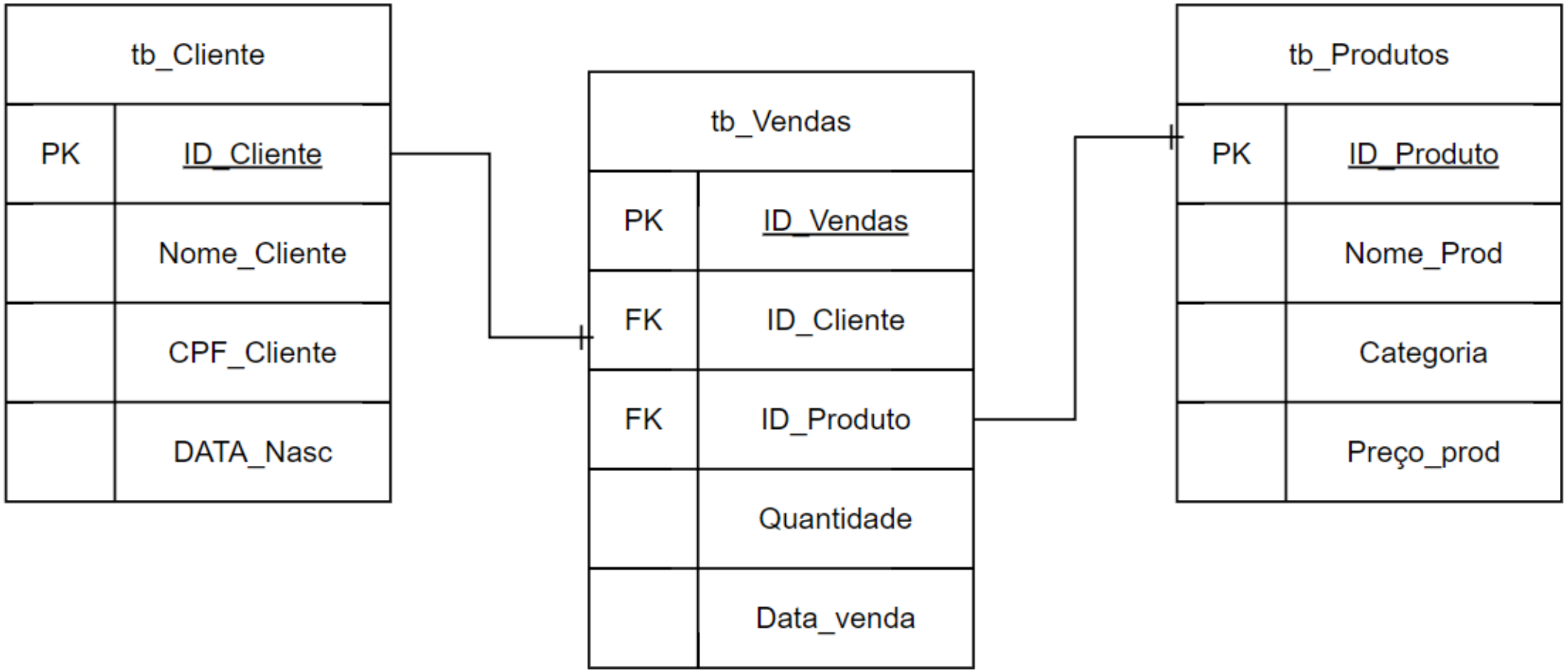
Chave Estrangeira

Coluna de uma tabela que estabelece
um Relacionamento com a Chave Primária (PK)
de outra tabela.

A partir da chave estrangeira (**Foreign Key / FK**) que sabemos com qual registro em outra tabela um registro está relacionado.

Criação de PK e FK

- Não é possível haver valores duplicados em uma PK;
- No geral, não é possível alterar valor de uma PK;
- FK são baseadas em valores de dados, classificadas como ponteiros lógicos;
- Um valor de uma FK deve corresponder a um valor existente em uma PK associada (ou valor de única). Cao contrário deve ser nulo (NULL);
- Uma FK deve referenciar a uma PK ou uma coluna de chave única.


tb_Cliente
PK
ID_Cliente
Nome_Cliente
CPF_Cliente
DATA_Nasc
tb_Vendas
PK
ID_Vendas
FK
ID_Cliente
FK
ID_Produto
Quantidade
Data_venda
tb_Produtos
PK
ID_Produto
Nome_Prod
Categoria
Preço_prod

Domínio

Definir tipo de dados e especifica valores válidos em um campo.

tb_Cliente		
Campo	Tipo de Dado	Tamanho
ID_cliente	Número (INT)	4 caracteres
Nome_cliente	Caractere (VARCHAR)	40 caracteres
CPF_cliente	Caractere (VARCHAR)	11 caracteres
Data_Nasc	Data e Hora (DateTime)	8 caracteres

Exercícios

Exercício: Construção de Modelo Conceitual para Banco de Dados de um Sistema de Reserva de Hotel

- Cenário:

Um hotel deseja desenvolver um sistema de banco de dados para gerenciar suas operações de reserva. O sistema deve ser capaz de armazenar informações sobre quartos, categorias de quartos, clientes, reservas e detalhes das reservas.

- **Quartos**: Cada quarto possui um número, capacidade, preço por noite e pertence a uma categoria.
- **Categorias de Quartos**: Cada categoria tem um nome (por exemplo, standard, deluxe, suite) e uma descrição.
- **Clientes**: Cada cliente possui um nome, endereço, telefone, email e data de registro.
- **Reservas**: Cada reserva é feita por um cliente, tem uma data de início, uma data de término e pode incluir múltiplos quartos.
- **Detalhes das Reservas**: Contém informações sobre a quantidade de cada tipo de quarto em uma reserva específica e o preço aplicado.

Exercício: Construção de Modelo Conceitual para Banco de Dados de uma Loja Virtual

Cenário:

Uma loja virtual deseja desenvolver um sistema de banco de dados para gerenciar suas operações. O sistema deve ser capaz de armazenar informações sobre produtos, categorias, clientes, pedidos e detalhes dos pedidos.

Produtos: Cada produto possui um nome, SKU (identificador único), descrição, preço, quantidade em estoque e pertence a uma categoria.

Categorias: Cada categoria tem um nome e uma descrição.

Clientes: Cada cliente possui um nome, endereço, telefone, email e data de registro.

Pedidos: Cada pedido é feito por um cliente, tem uma data e pode incluir múltiplos produtos.

Detalhes dos Pedidos: Contém informações sobre a quantidade de cada produto em um pedido específico e o preço aplicado.

Tarefa:

Desenvolva o modelo conceitual para o banco de dados descrito acima. O modelo deve incluir entidades, atributos e relacionamentos.

Cardinalidades

Máxima: trata-se do número máximo de instâncias de entidade que podem participar em um relacionamento. Pode ser 1 ou N(muitos).

Mínima: Número mínimo de instâncias de entidade que devem obrigatoriamente participar em um relacionamento; Zero é participação opcional e um é obrigatória.

Dicionário de Dados

DICIONÁRIO DE DADOS

Tabela	Veiculo			
Descrição	Armazenará as informações dos veículos			
Observações	Essa tabela possui uma chave estrangeira da tabela Marca			
Campos				
Nome	**Descrição**	**Tipo de dado**	**Tamanho**	**Restrições de domínio (PK, FK, Not Null, Check, Default, Identity)**
Codigo	Código de identificação da tabela	Int		PK / Identity
Placa	Placa do ônibus.	Varchar	20	Unique / Not Null
Anoveiculo	Ano de fabricação do ônibus.	Int		Not Null
Anocompra	Ano de compra do veículo	Int		Not Null
Codmarca	Chave estrangeira referenciando o código da tabela Marca	Int		FK

Dicionário de dados e modelo de entidade e relacionamento

Para o processo de análise de sistemas, um dos pontos fortes é o MER – Modelo de Entidade e Relacionamento, onde são definidas as entidades que irão compor o sistema e como elas irão relacionar-se.

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

Normalização

Anomalias de Atualização

Problemas que ocorrem em banco de dados mal planejados e não-normalizados, geralmente ocorrendo por excesso de dados armazenados em uma mesma tabela.

São classificadas em:

- Inserção
- Exclusão
- Modificação

Anomalia de Inclusão

Não deve ser possível adicionar um dado a não ser que outro dados esteja disponível.

Por exemplo, não deve ser permitido cadastrar um novo livro sem que um autor já esteja cadastrado.

Anomalia de Exclusão

Ao excluirmos um registro, dados referentes em outra tabela são excluídos.

Ex:

Se excluir um autor, os livros desse autor devem ser excluídos também.

Anomalia de Modificação

Ao alterar um dado em uma tabela, dados em outras tabelas precisam ser alterados.

Ex:

Se o código de um autor for modificado, esse código deve ser modificado na tabela de autores e na tabela de livros, para manter o relacionamento entre os livros e seus autores.

Eliminar as anomalias

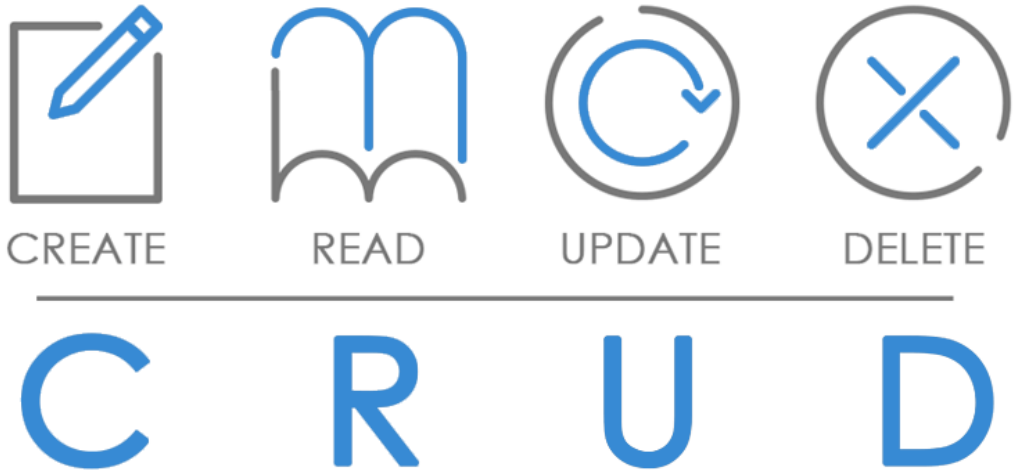
Projetar os esquemas de relações (tabelas) no banco de dados de modo que nenhuma anomalia de inserção, exclusão ou modificação esteja presente nas relações.

Para isso usamos o processo de Normalização

Normalização

Processo de análise de uma relação para assegurar que seja bem formada.

Ou seja, em uma relação podemos inserir, excluir ou modificar registros sem criar anomalias.



Objetivo

Analisar esquemas de relação(tabelas) com base em
suas dependências funcionais e chaves primárias para:

1. Minimizar redundância
2. Minimizar anomalias de inserção, exclusão e modificação

As relações são compostas em esquemas de relação menores que atendem
aos testes de forma normal.

Primeira Forma Normal

Somente possui valores atômicos (não tem atributos multivalorados - indivisíveis)

Não há grupos de atributos repetidos (há apenas um dado por coluna nas linhas)

Existe uma chave primária

Primeira forma normal (1NF)

A primeira forma normal diz que atributos de uma entidade que tem características de armazenamento de vários valores, devem gerar uma nova entidade (ou ser extraído para outra entidade já existente) de dados relacionada a entidade origem.

Separar o Grupo Repetitivo

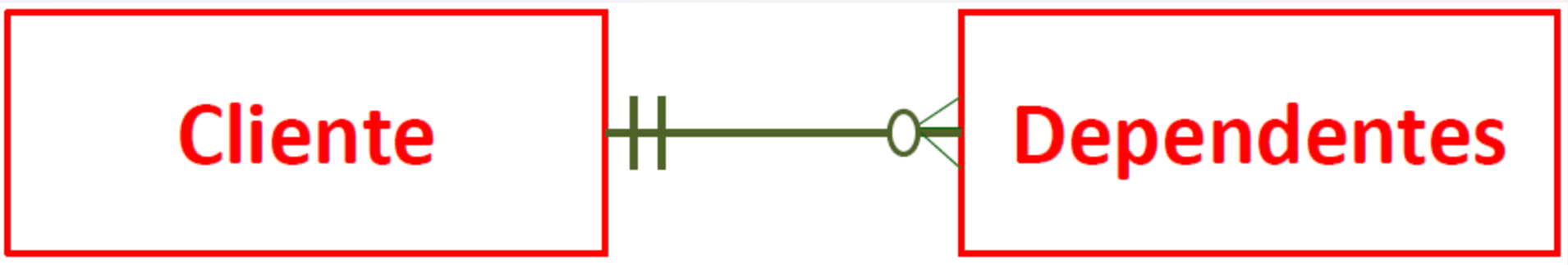
Exemplo: Entidade Cliente

Exemplo: Entidade Cliente

- O cliente pode ter mais de um nome? Não
- O cliente pode ter mais de um CPF? Não
- .....
- O cliente pode ter mais de um dependente? **SIM**

Então **DEPENDENTE** é uma **Entidade**.

Exemplo: Entidades Cliente e Dependentes



tbl_clinte
Cod_Cliente
Nome_cliente
*Tel_cliente
Endereço_cliente

tbl_cliente			
Cod_Cliente	**Nome_cliente**	***Tel_cliente**	**Endereço_cliente**
159	José	99654-2147 2865-3212	Rua do contorno, 321 ap.12 - Itaquera
753	Marcos	3256-7821	Av. Cruzeiro do sul, 3000 - Santana
951	Ana	5513-9812 97561-9875	Rua Tito, 54 - Vila Romana - Lapa

Na coluna telefone temos muitos telefones para um único cliente. E endereço também tem dados compostos.

tbl_cliente			
Cod_Cliente	**Nome_cliente**	**Rua**	**Bairro**
159	José	Rua do contorno, 321	Itaquera
753	Marcos	Av. Cruzeiro do sul, 3000	Santana
951	Ana	Rua Tito, 54 - Vila Romana	Lapa

tbl_telefone	
Cod_Cliente	Tel_cliente
159	99654-2147
159	2865-3212
753	3256-7821
951	5513-9812
951	97561-9875

Segunda Forma Normal

Está na 1FN

Todos os atributos não-chave são funcionalmente dependente de todas as partes da chave primária.

Não existem dependências parciais.

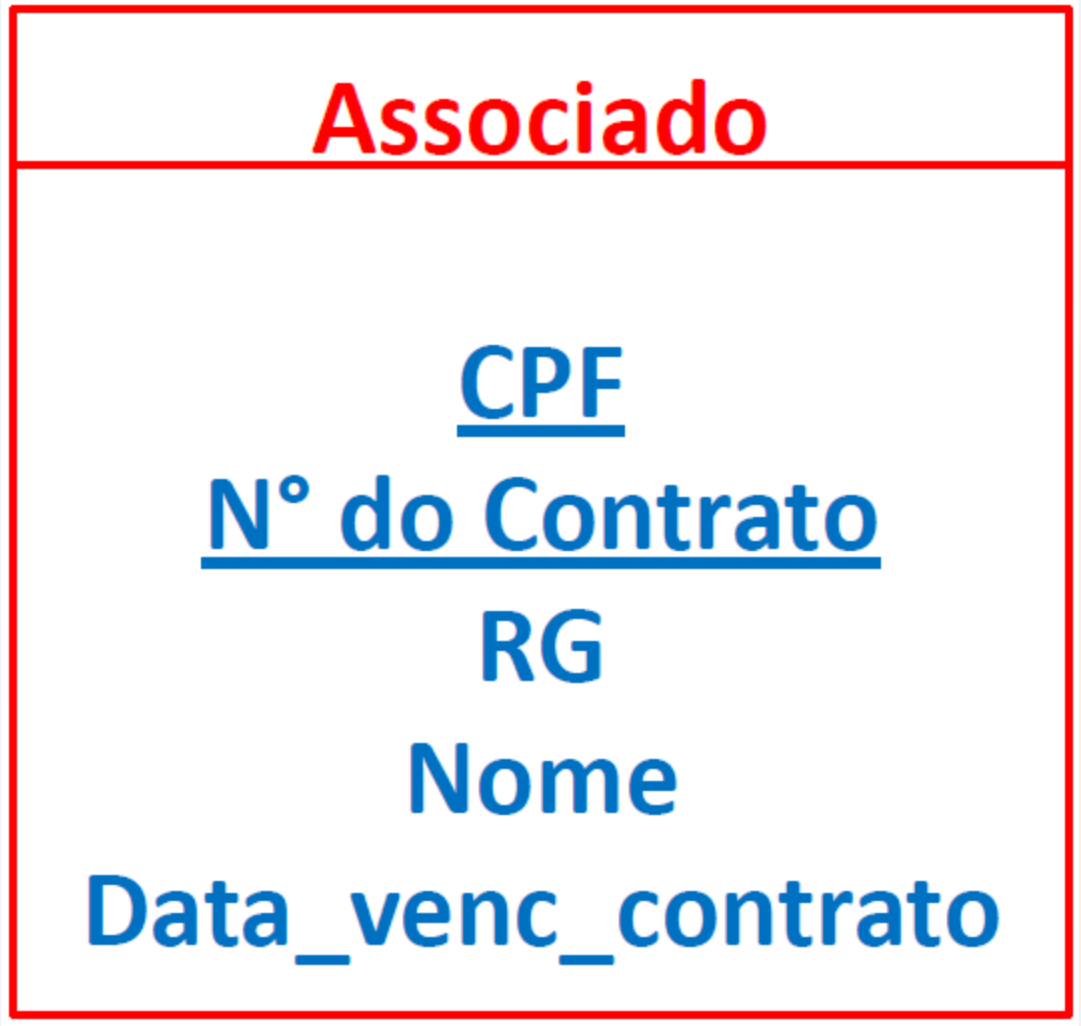
Caso contrário, deve-se gerar uma nova tabela com os dados.

Segunda forma normal (2NF)

A segunda forma normal fala sobre a dependência relativa de dados. Em termos claros podemos dizer que todo atributo de uma entidade que não depender exclusivamente da chave primária, deve gerar uma nova entidade.

Separar o Dependência Funcional Parcial

Exemplo: Entidade Associado

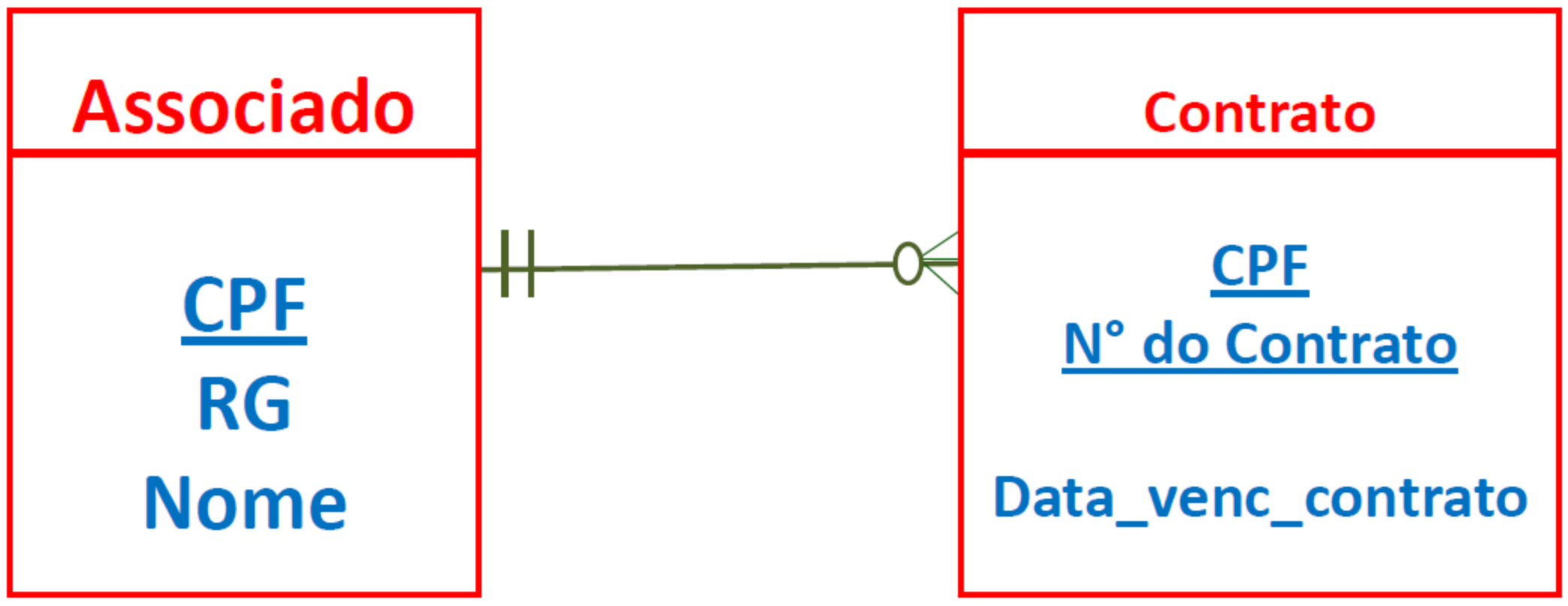


Exemplo: Entidade Associado

- Trocando o atributo nome o cliente será descaracterizado? Sim
- Trocando o atributo os CPF e RG o cliente será descaracterizado? Sim
- ……
- Trocando a Data_venc_contrato o cliente é descaracteriza? **Não**

Então **Contrato** é uma **Entidade**

Exemplo: Entidades Associado e Contrato



tbl_peça
Cod_peça
Cod_fornec
Local_fornecedor
Qtde_estoque
Tel_fornecedor
Qtde_caixas

tbl_peça					
Cod_peça	Cod_fornec	Local_fornecedor	Qtde_estoque	Tel_fornecedor	Qtde_caixas
9	121	São Paulo	512	2365-6532	52
12	122	Manaus	263	4465-8632	27
147	121	São Paulo	196	2365-6532	20
1472	123	Porto Alegre	89	2956-8653	9
3214	122	Manaus	296	4465-8632	30

PK

Os atributos devem ser totalmente dependente da chave primária

tbl_peça			
Cod_peça	Cod_fornec	Qtde_estoque	Qtde_caixas
9	121	512	52
12	122	263	27
147	121	196	20
1472	123	89	9
3214	122	296	30

PK	FK
	PK

tbl_Fornecedor		
Cod_fornec	Local_fornecedor	Tel_fornecedor
121	São Paulo	2365-6532
122	Manaus	4465-8632
121	São Paulo	2365-6532
123	Porto Alegre	2956-8653
122	Manaus	4465-8632

PK

Terceira Forma Normal

Está na 2FN

Não existirem dependência **transitivas** (uma dependência funcional entre dois ou mais atributos que não sejam chave).

Terceira forma normal (3NF)

A terceira forma normal utiliza o principio de transitividade, e diz que todo atributo que dependem não transitivamente da chave primária gera uma entidade.

Separar a Dependência Transitiva

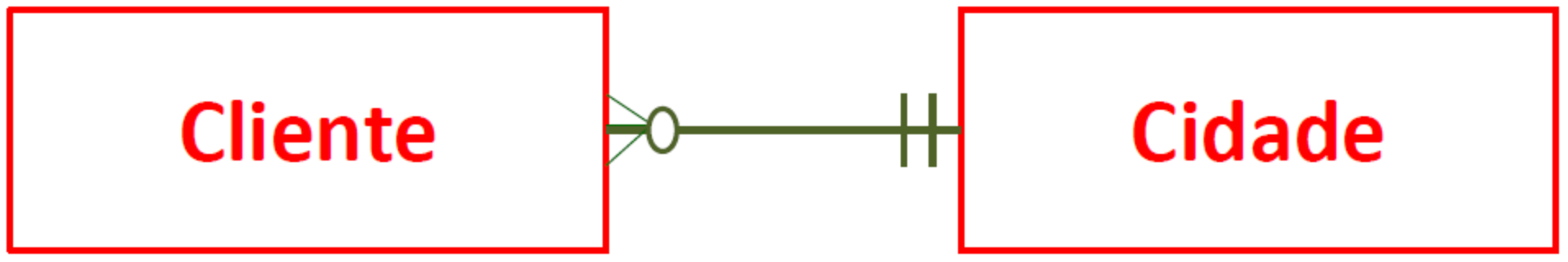
Exemplo: Entidade Cliente

Exemplo: Entidade Cliente

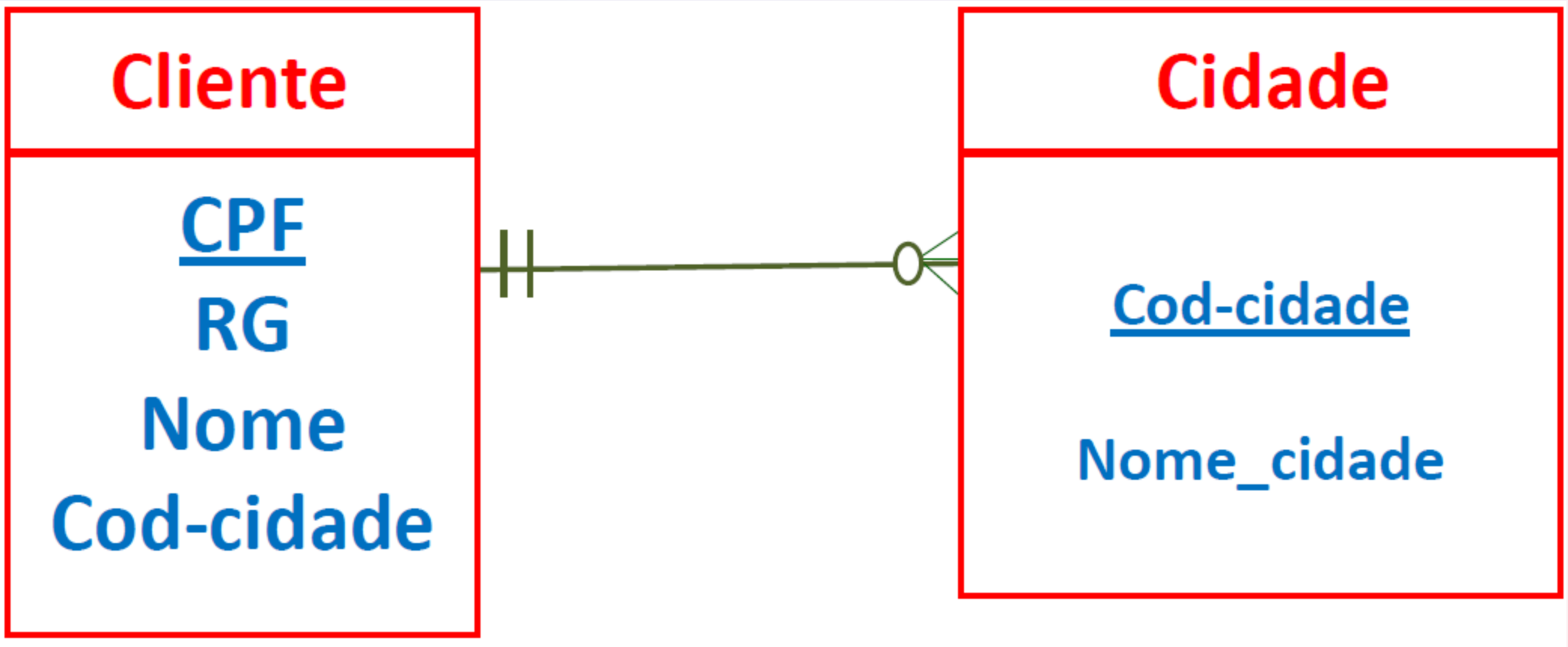
- O nome da cidade depende do CPF? **Não**

Então **Cidade** é uma **Entidade**

Exemplo: Entidades Cliente e Cidade



Exemplo: Entidades Cliente e Cidade



tbl_venda
nota_fiscal
cod_vendedor
nome_vendedor
cod_produto
qtde_vendida

tbl_venda				
nota_fiscal	cod_vendedor	nome_vendedor	cod_produto	qtde_vendida
15326	2	Maria	132	10
15327	6	Ana	153	12
15328	2	Maria	143	11
15329	9	João	132	9
15330	7	Renato	153	12

PK

tbll_venda			
nota_fiscal	cod_vendedor	cod_produto	qtde_vendida
15326	2	132	10
15327	6	153	12
15328	2	143	11
15329	9	132	9
15330	7	153	12

PK	FK

tbl_vendedor	
cod_vendedor	nome_vendedor
2	Maria
6	Ana
7	Renato
9	João

PK	

Resumindo

- Tabela Não-Normalizada
- Remover atributos multivalorados e compostos (1FN)
- Remover Dependências Parciais (2FN)
- Remover Dependências Transitivas (3FN)