



# Banco de Dados

## ▼ Navegação

1.
  - 1.1.
  - 1.2.
  - 1.3.
    - 1.3.1.
    - 1.3.2.
  - 1.4.
    - 1.4.1.
    - 1.4.2.
    - 1.4.3.
  - 1.5.
  - 1.6.
  - 1.7.
  - 1.8.
    - 1.8.1.
    - 1.8.2.
    - 1.8.3.
  - 1.9.
  - 1.10.
  - 1.11.

1.12.

1.13.

1.14.

1.15.

1.16.

## 2. Prática

2.1.

2.2.

2.3.

2.4.

# Teoria

## 1. Dados

São fatos conhecidos que podem ser registrado.

São valores que armazenado é usado como matéria prima para **informações**

## 2. Informações

São dados processados e formatados, com o objeto de caracterizar um elemento, fato ou situação.

### **3. Conhecimento**

Integra elementos, informações, experiências de forma adequada ao raciocínio humano.

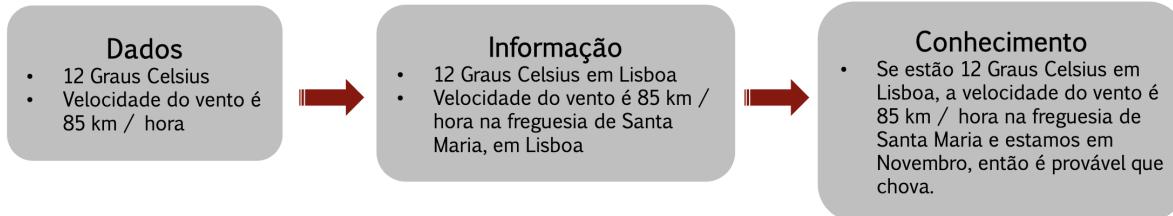
E nos proporciona a tomada de decisão.

## **Conceitos iniciais**

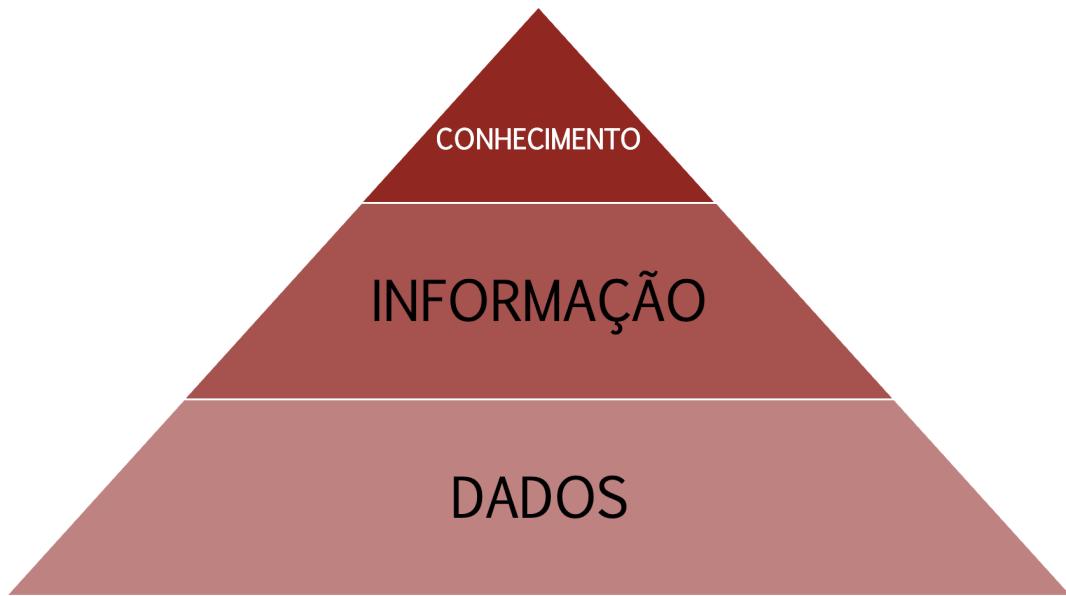
### **1. DadosxInformaçõoxConhecimento**

Um relatório com dados de um paciente, juntos formam as informações, e se analisado pelo médico se torna conhecimento.

Porém se o relatório for lido pelo próprio cliente, ele pode não tirar conclusões corretas e logo não gerando conhecimento.



Vemos que dados é a base de tudo, com os dados conseguimos as informações, e só com a análise das informações conseguimos tirar as conclusões que geram o conhecimento.



## Banco de Dados

Coleção de dados estruturados e relacionais.

Dados organizados com significados definidos e específico.

- Uma organização aleatória de dados não é um banco de dados.
- Um banco de dados é projetado, construindo e populado, com dados que possuem um objetivo específico, normalmente para gerar informações.
- Ele possui um grupo de usuários e aplicações pré-definidos, em que esses usuários estão interessados.

### 1. Sistema de banco de dados

Conjunto de dados inter-relacionados que são manipulados apenas por uma aplicação.

Normalmente referenciados como SGBD - Sistema Gerenciador de Banco de Dados.

## 2. Persistência

Os dados de um banco de dados são ditos persistentes porque não são dados efêmeros ou volátil, como dados IO(input output), como a RAM.

Os dados de um Banco de dados persistem pois uma vez aceito pela SGBD, só serão removidos através de outra chamada explícita solicitante a remoção. E nunca por um meio acidental, como por exemplo o término do fluxo da aplicação.

Um Banco de Dados é uma coleção de **dados persistentes**, utilizada por aplicações de empresas ou corporações

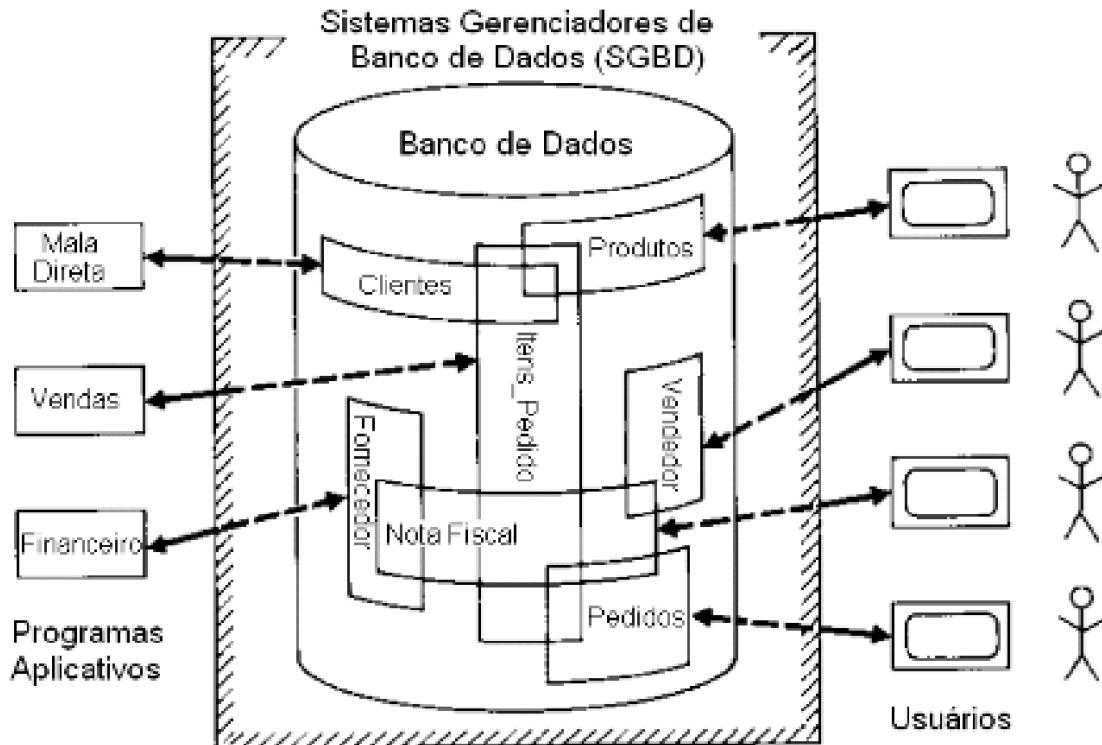
## SGBD - Sistema Gerenciador de Banco de Dados

Em inglês: BDMS

**Representação simplificada:**

- Elimina redundâncias desnecessárias de dados.

- Diferentes aplicativos e usuários podem acessar os mesmos dados, pois o SGBD trata o acesso concorrente.



- Conjunto de dados associados a um conjunto de programas.
- Ambiente apropriado e eficiente para recuperação e armazenamento de informações.
- Gerenciamento de grandes volumes de informações:
  - Definição de **estruturas de armazenamento**
  - Definição de **mecanismos de manipulação**

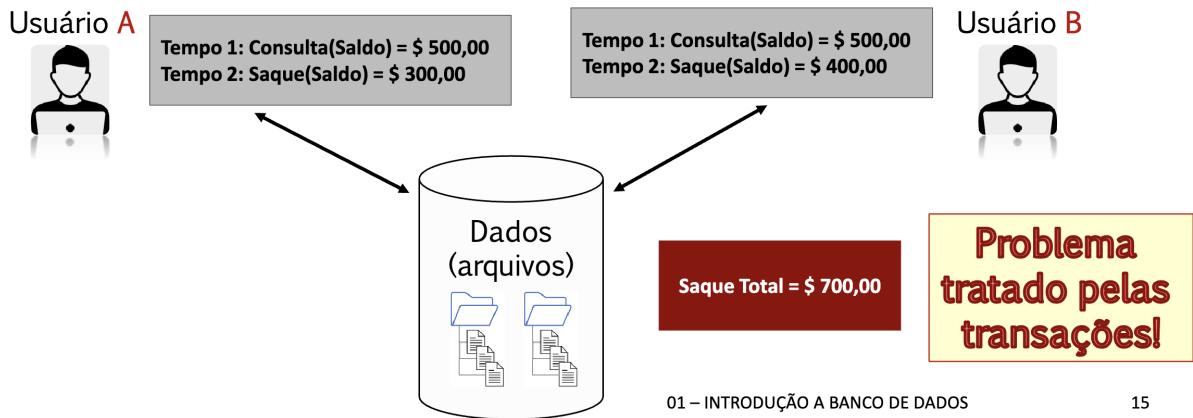
- Trata questões de segurança, como falhas, acesso não autorizado e restrições.

## 2. SGBD vs File System

- › Dessa forma, cada aplicativo / usuário cria e estrutura seus dados de forma **independente**, gerando **problemas**:
  - **Redundância e Isolamento de dados**: programas distintos utilizam os mesmos dados em arquivos e formatos de armazenamento diferentes.
  - **Inconsistência de dados**: não há como garantir que uma modificação de dados seja realizada nos diferentes arquivos em que estão armazenados.
  - **Dificuldade no acesso aos dados**: o acesso aos dados é feito apenas pelos programas de aplicativos que gerenciam esses dados.
  - **Anomalias por acesso concorrente**: os dados podem ficar inconsistentes se manipulados simultaneamente por aplicativos / usuários distintos, sem que haja uma coordenação do acesso.
  - **Problemas de segurança**: não é possível conceder acesso a parte dos dados.
  - **Problemas de integridade dos dados**: regras de negócios (restrições impostas aos dados) devem ser implementadas em sistemas aplicativos, de forma independente.

Ou seja, nosso SGBD cuida de todos esses pontos e outros para nós, sem isso seria só um sistema de acesso a arquivos, o que nos sujeitaria a todos esses problemas.

### Exemplo de acesso e manipulação concorrentes



01 – INTRODUÇÃO A BANCO DE DADOS

15

Note o problema de concorrência, não temos como ter essa tratativa sem um SGBD

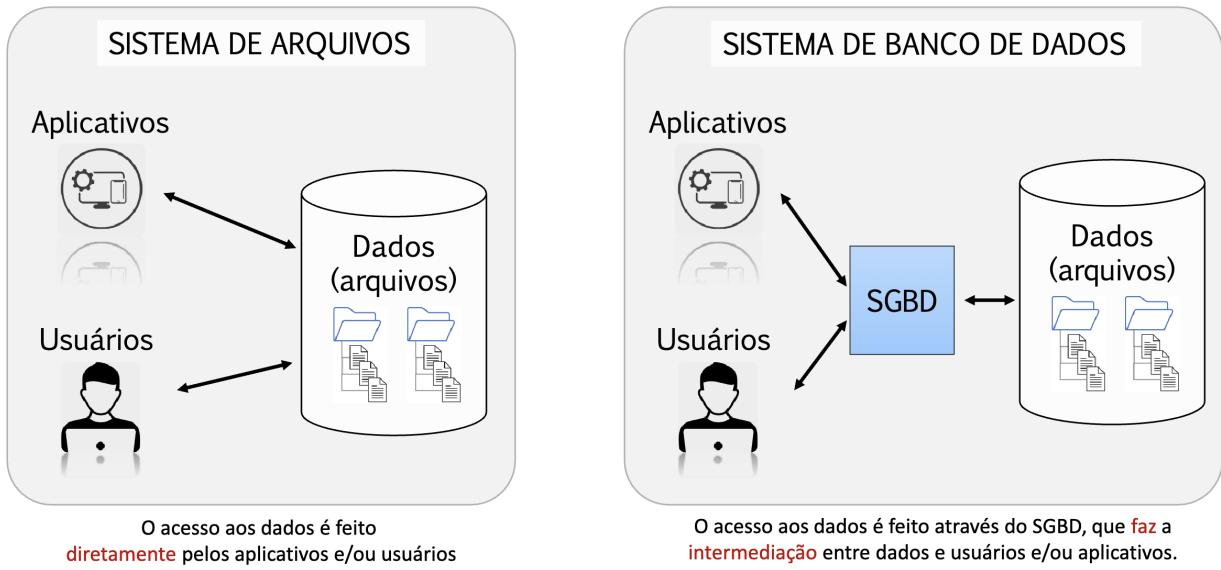
#### › COM um SGBD

- **Compartilhamento de dados:** evita a redundância e isolamento de dados, pois estes passam a ser armazenados e acessados da mesma forma, controlados pelo SGBD.
- **Inconsistência evitada:** o SGBD é o único acesso aos dados (redundância reduzida), permitindo a propagação da alteração feita em um dado duplicado (redundância controlada).
- **Facilidade de acesso aos dados:** o SGBD é o único acesso aos dados.
- **Supporte à Transação:** evita os problemas do acesso concorrente, pois tem **Duporte a Transações** (várias operações de alteração são executadas em sequência, sem falha de nenhuma, para garantir a consistência de dados), que é dita **atômica**(indivisível). Caso alguma das operações da transação não seja realizada, o SGBD garante o retorno dos dados ao seu valor anterior (**rollback**) à transação.
- **Segurança Reforçada:** o SGBD, além de ser o único acesso aos dados, também permite atribuir permissões diferentes (restrições de segurança) para usuários diferentes.
- **A Integridade dos Dados pode ser mantida:** é possível configurar regras de negócios (restrições impostas aos dados) no próprio SGBD.

Notamos que um sistema de arquivos como excell, conseguimos acessar diretamente ele e tratar dados como precisarmos, mas teríamos varios problemas como citamos acima.

Agora note um sistema de banco de dados que possui um SGBD

## SGBD vs. Sistema de Arquivos



## 3. Independência de Dados

### › Sistema Dependente de Dados

- A maneira como os dados estão armazenados fisicamente e a técnica para obter esses dados são determinadas pelos requisitos de uma aplicação.
- O conhecimento da representação física e da técnica de acesso está embutida no código da aplicação.

### › Independência de Dados através de SGBDs

- Garante imunidade das aplicações a alterações na representação dos dados e na técnica de acesso.
- A ideia é armazenar os dados de forma independente dos sistemas de aplicação, uma vez que os dados são patrimônio da empresa / corporação.

### › Independência Física

- É a capacidade de **modificar o esquema físico dos dados** sem que isso gere qualquer necessidade de modificação no programa de aplicação que acessa os dados.
- Modificações físicas são necessárias eventualmente para otimizar o desempenho do SGBD.

### › Independência Lógica

- É a capacidade de **modificar o esquema lógico dos dados** sem que isso gere qualquer necessidade de modificação no programa de aplicação que acessa os dados.
- O esquema lógico envolve a estrutura lógica de dados; alterar a estrutura lógica acontece, por exemplo, quando é preciso acrescentar uma nova informação ao banco (como novas moedas em um sistema bancário).

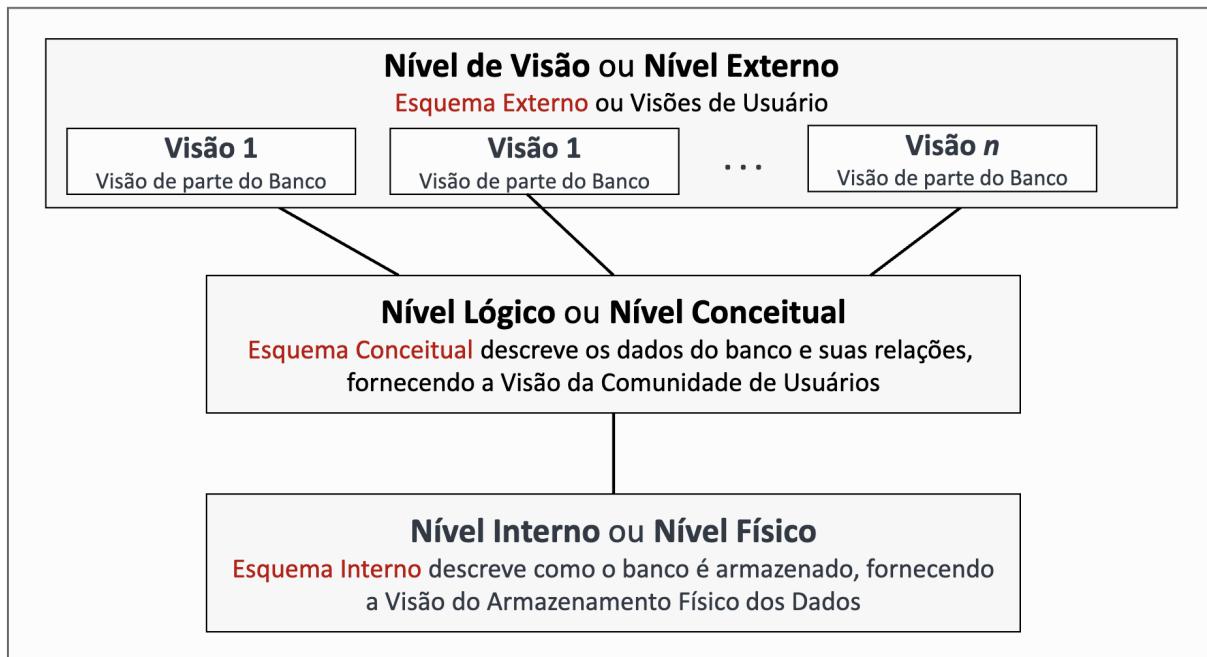
## SGDB

### Benefícios

- Permite uma visão mais abstratas dos dados.
- Reduz as redundâncias de dados.
- Realiza o acesso as "pastas".

- Permite aplicar restrições (constraint), reforçando padrões e regras.
- Melhor segurança.

## Níveis de abstrações



- **Nível de visão ou externo.**

Visão do usuário final/cliente, sem informações precisar, porém dedutivas.

- **Nível lógico ou nível conceitual.**

Esquema conceitual que descreve o banco e seus dados, dando uma visão mais abstratas aos usuários.

- **Nível Interno ou Nível Físico.**

Esquema Interno, descreve como o banco é por dentro, ou seja, literalmente.

## Instância vs Esquema

- Semelhante á variáveis.

### Esquema Lógico

Estrutura geral do banco de dados. O que define os tipos e "declara" todo o espaço e os valores das tabelas.

### Instância

Conteúdo real do banco em determinado momento.

## DBA e DA

### › Funções do DA

- Decidir o conteúdo das informações do BD em um **Esquema Conceitual** (**Nível Lógico** ou **Nível Conceitual** = **visão de negócio**)

### › Funções do DBA

- Decidir a estrutura de armazenamento físico mais adequada para implementar o **Esquema Conceitual** (**Nível Físico** = **visão técnica**)
- Definir concessão de **autorização para acesso a dados**, para controlar que partes do banco de dados os vários usuários podem acessar.
- Implementar as estratégias de **back-up** e **recovery**
- Monitorar e adequar o desempenho

# Modelo de dados

Conjunto de modelos que definem um banco de dados, dando relacionamentos, semânticas e regras.

## Modelo Relacional

Modelo clássico. baseado em registro estruturados.

- **Tuplas** = linhas ou registros
- **Atributos** = colunas
- **Domínios** = conjunto de valores

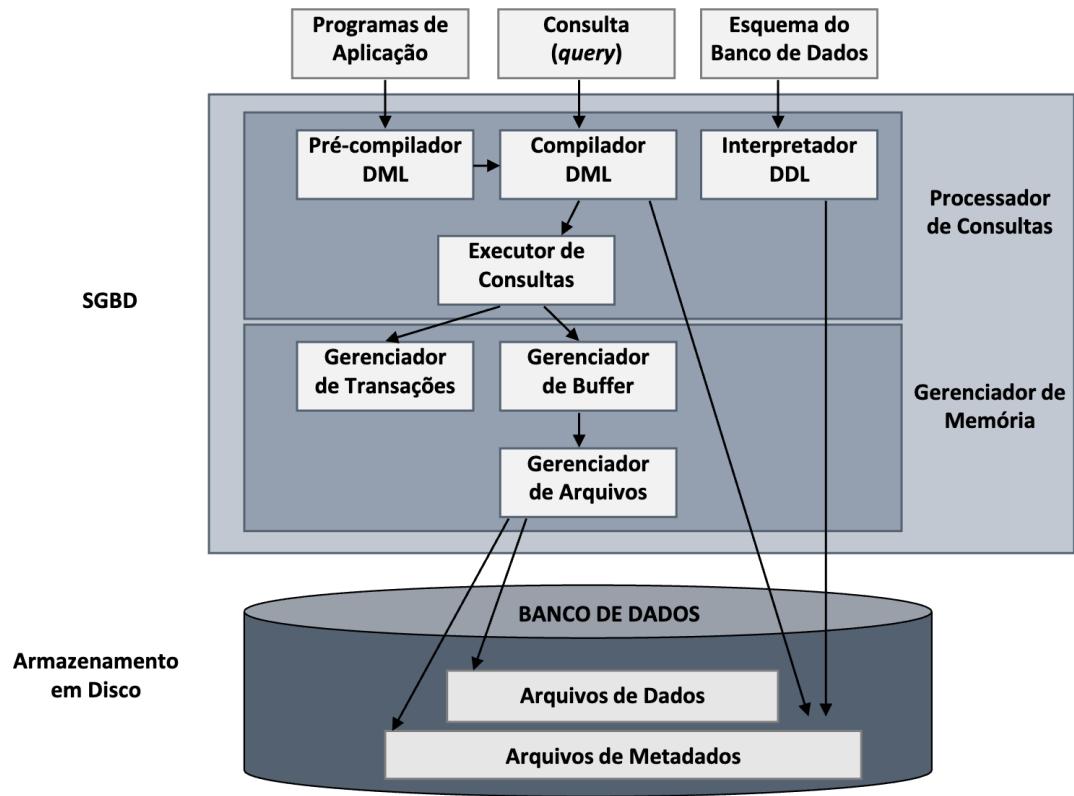
Nome_Cliente	Seguro_Social	Rua_Cliente	Cidade_Ciente	Número_Conta
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-125
Turner	182-173-6091	Putnam	Stamford	A-305

Diagrama explicativo:

- Um **Atributo** é apontado para a coluna "Número\_Conta".
- Uma **Tupla** é apontada para a linha da família Johnson.
- Um **Domínio** é apontado para o valor "182-173-6091" na coluna "Seguro\_Social".

- Domínio é um termo também sendo usado em Matemática Discreta, para representar valores de um conjunto  $f()$ ;

## Estrutura geral



- Fazemos o acesso direto ao SGBD, que internamente distinguirá o tipo de solicitação.
- Irá passar pelo pré-compilador da linguagem que foi realizada a requisição.
- Passará pelo executor.
- Disso ao executar irá entrar no gerenciador de memoria.
- Vários processos complexos internos e então acessa aos arquivos de dados em disco.

## Questões

- Defina SGBD?

Um sistema inteligente de controle de memória e consultas, com diversas funcionalidades úteis de usuários, consultas, integridade, e etc. Isso garante diversas vantagens a uma busca convencional de arquivos.

- **Dados persistentes**

Dados que se mantêm gravados apesar da volatilidade ou não, isso garante que eles foram salvos definitivamente.

- **Integridade**

Garante que nosso dado se mantém correto apesar de adversidades, como 2 pessoas tentando acessar ou modificar um mesmo dado, e +.

- **Compartilhamento**

- **DA**

Data Analyst

- **DBA**

Database Administrator. Responsável tecnicamente para as questões de banco, sendo responsável por otimizar SQLs, estruturar bancos inteiros de formas performática, e +

- **Linguagem de consulta**

SQL, uma linguagem de consulta que faz uso do DML para realizar buscas em uma série de arquivos através do SGDB

- 2) Descreva as principais **vantagens e desvantagens** de um SGBD
  - 3) Quais as principais diferenças entre um **sistema de arquivos** e um **sistema de banco de dados?**
  - 4) Diferencie **esquema** e **instância** de BD.
  - 5) Diferencie **independência lógica** e **física** de dados .
  - 6) Descreva os principais **níveis de abstração** de um sistema de banco de dados.
- 
2. Controle de usuários, otimização, integridade de dados, persistencia, redundancia, controle de acessos e +. Suas desvantagens
  3. O sistema de arquivos não possui redundância, controle de usuários, e por isso se duas pessoas acessarem ao mesmo tempo, pode ocorrer redundâncias. Problemas de segurança, otimização, e controle dos dados.
  4. Esquema é uma declaração de tipos e memórias de dados em um banco. Já instância é um conteúdo inserido no banco de dados.
  5. Independência lógica é uma mudança no esquema físico do banco. Já a lógica é a mudança lógicamente feita em um banco, como uma adição de um novo atributo.

6. Nível do usuário, como o usuário vê os esquemas de dados.  
Nível lógico/conceitual, como os dados estão modelados, de forma abstrata. Nível interno ou físico, é como os dados estão real nas aplicações, em nível físico.

## Modelo de Dados

Conjunto de ferramentas conceituais usadas para descrever os dados, relaciona-los, e verificar semântica e regras.

O modelo de dados reflete o nível conceitual e lógico, garantindo a criação físico com artefatos concretos e modelagem.

- › Existem diversos modelos, como por exemplo:
  - Modelo Entidade-Relacionamento
  - Modelo Orientado a Objetos
  - Modelo Relacional

Os modelos a seguir são **especificações** como o UML, no qual são usados para modelagem de sistemas no geral. O que pode se refletir em modelagem de bancos também.

## Modelo Orientado a Objetos

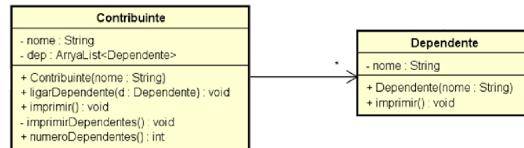
Diagramação de objetos com outros objetos, sendo cada objeto uma entidade/pessoa/tabela/objeto.

### › Modelo Orientado a Objetos

- A base desse modelo é um **conjunto de objetos**, que possui suas **propriedades e métodos**.
- Um conjunto de objetos com as mesmas propriedades e métodos é denominado de **classe**.
- A única forma de acessar as propriedades de um objeto é através de seus **métodos**.

### › Modelo Orientado a Objetos

- Classes, com propriedades e métodos
- Especificação de Funcionalidade juntamente com seus Dados



## Modelo Entidade-Relacionamento

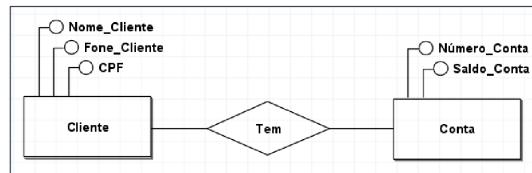
Diagramação com base em entidades, representações únicas da identidade de algo da vida real, que se relacionam com outras (se existir), e possuí seus atributos.

### › Modelo Entidade-Relacionamento (MER)

- Percebe o mundo real como um conjunto de objetos básicos, chamados **entidades**, e o relacionamento entre eles.
- Uma **entidade** é uma "coisa" ou "objeto" do mundo real (p. ex.: pessoa, carro, conta de cliente, ...).
- As entidades são descritas por seus **atributos** (p. ex.: idade, telefone e endereço para a entidade pessoa).
- Um **relacionamento** é uma associação entre entidades (p. ex.: deposita entre as entidades cliente e conta).

### › Modelo Entidade-Relacionamento (MER)

- Entidades, atributos, relacionamentos
- Nível Conceitual
- Projeto Conceitual de Dados



## Modelo Relacional

Diagramação com representação de conjuntos(tabelas talvez), que se relacionam com outros.

### › Modelo Relacional

- Usa um conjunto de tabelas para representar tanto dados como a relação entre eles.
- Cada tabela possui múltiplas **colunas** e cada uma possui um **nome único**.
- Usado para especificar a **estrutura lógica do BD** e sua **implementação**

### › Modelo Relacional

- Tabelas
- Nível Lógico
- Projeto Lógico de Dados

Tabela Cliente		
CPF	Nome_Cliente	Fone_Cliente
12345678901	José da Silva	4555-5588
22222666666	Maria Pires	4666-6600

Tabela Conta		
CPF	Número_Conta	Saldo_Conta
12345678901	1456-88	300,00
22222666666	2589-84	100,00

## Fases de um desenvolvimento

### 1. Criar o modelo conceitual.

Planejar o Banco de dados pensando em entidades (**retângulos**), e criar os seus relacionamentos (**linhas que ligam entidades**).

### 2. Criar o modelo lógico.

Considerar limitações ou regras do SGDB que irá utilizar.

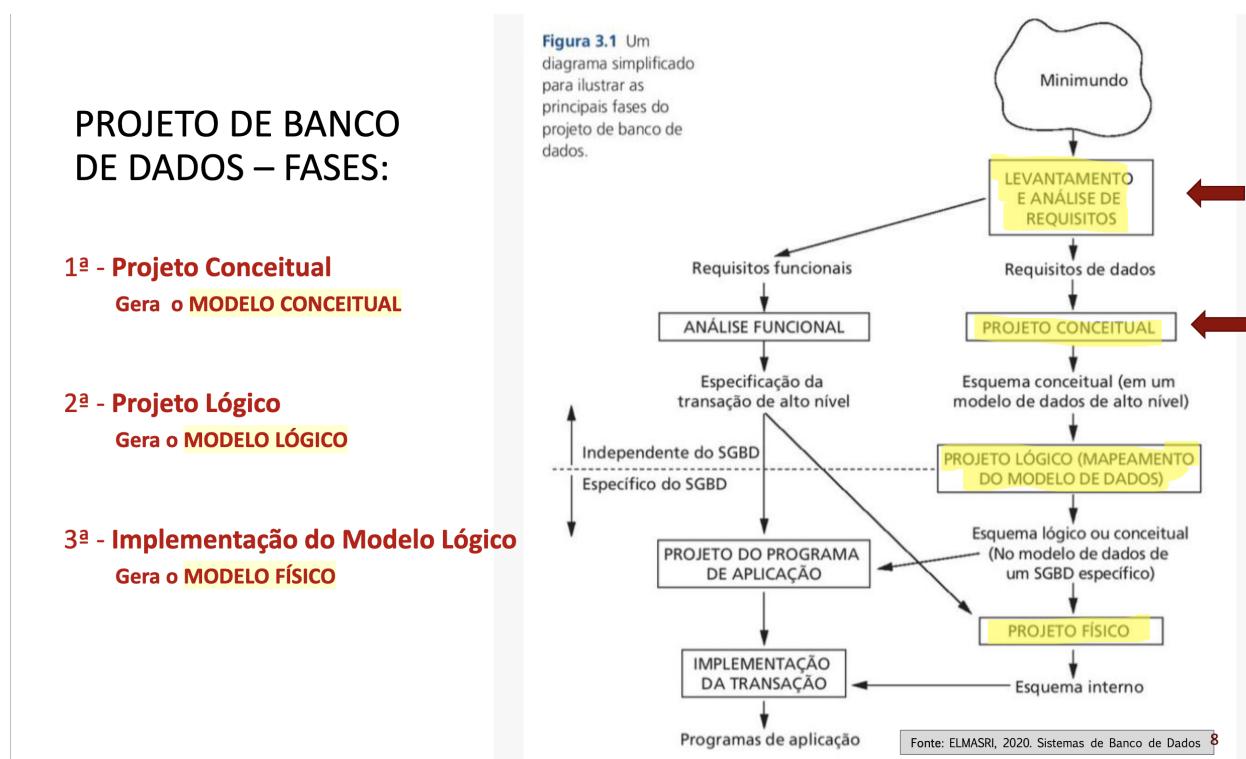
Com isso iremos criar modelos mais amplos e flexíveis, que representaram nossas entidades em tabelas e seus

relacionamentos. É a estrutura que o físico e o banco de dados irá usar para modelagem, e é lógico por que ele fica escondido.

### 3. Criar o modelo físico.

Implantar o modelo lógico, incluindo uma análise dos recursos até o momento, e adicionando características se necessário.

#### Exemplo:



Fonte: ELMASRI, 2020. Sistemas de Banco de Dados 8

## Levantamento de requisitos

### Esquema conceitual

Indica as necessidades da empresa.

- Também indica as **necessidades funcionais** da empresa.
- Na **especificação das necessidades funcionais**, os usuários descrevem os tipos de operações (ou transações) que serão realizadas nos dados.
  - > Exemplo de Operações: modificar ou atualizar dados, pesquisar e recuperar dados específicos e excluir dados.
- Nessa fase do **projeto conceitual**, o projetista pode revisar o esquema para garantir que ele atende às necessidades funcionais.

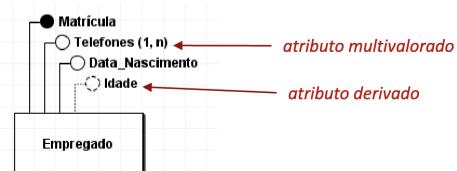
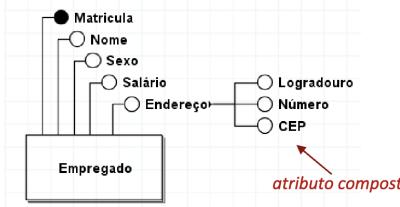
## Estudo de Caso

- Uma empresa precisa de um software para **gerenciar seus funcionários**.
- Durante a **fase de entrevistas (levantamento de requisitos)**
  - > Você irá perguntar o que a empresa deseja armazenar.
  - > Uma possível resposta da empresa será: informações sobre **empregados**, seus **dependentes**, o **departamento** de cada empregado etc.
  - > Portanto, podemos identificar algumas entidades no nosso modelo ER como
    - Empregado,
    - Departamento e
    - Dependente.
  - > Depois de **identificar essas entidades**, é necessário **modelá-las** no diagrama entidade relacionamento.

Modelo entidade relacionamento - MER, é a forma que o projetista apresenta a modelagem do banco ao cliente.

## Tipos de atributos

- › **Atributo chave**
  - Representa unicamente uma instância da entidade
  
- › **Atributo composto**
  - Seu conteúdo é formado por vários itens menores
  
- › **Atributo derivado**
  - Seu conteúdo é formado por mais de um valor.
  - Exemplo: **idade** → atributo derivado do atributo **data-nascimento**.
  
- › **Atributo multivalorado**
  - Seu conteúdo é formado por mais de um valor.
  - Exemplo: **Telefone** → um empregado poderá ter mais de um número de telefone



## Tipos de chaves

- › **Superchave**
  - É um conjunto de um ou mais **atributos** que nos permite identificar de maneira única uma entidade.
  - Exemplo: o atributo **CPF\_cliente** identifica unicamente uma ocorrência na entidade **Cliente**. Por outro lado, o atributo **nome\_cliente** não é uma **superchave**, pois algumas pessoas podem ter o mesmo nome.
  -
  
- › **Chave candidata**
  - São **superchaves** para as quais nenhum subconjunto de **atributos** pode ser **superchave**.
  - Exemplo: embora os atributos **CPF\_cliente** e **nome\_cliente**, juntos, possam fornecer uma **superchave**, sua combinação não forma uma chave candidata, uma vez que **CPF\_cliente**, sozinho, é uma **chave candidata**.
  -
  
- › **Chave primária** **PK – Primary Key**
  - É a **chave candidata** escolhida pelo projetista do BD para a identificação única de ocorrências em uma entidade.
  - Ou seja, não existem duas ocorrências distintas, da mesma **entidade**, com os mesmos valores na **chave primária**.
  - É ao mesmo tempo **superchave** e **chave candidata**.

- › **Chave estrangeira** **FK – Foreign Key**
  - Um **entidade** pode incluir, entre seus **atributos**, a **chave primária** de outra entidade.
  - Essa chave é chamada de **chave estrangeira**.

- **Superchave**

É o conceito de um **atributo**(identificados) principal, no qual representa toda a tabela.

- **Chave Candidata**

Essa chave é usada para identificar todos atributos (identificadores) únicos da tabela, que são candidatos a se tornarem superchaves, porém podem ou não ser uma. Ou seja, é um atributo único, obrigatório para a tupla existir, e necessário para a integridade da tabela.

- **Chave Primária**

É a escolha da chave candidate, definindo-a como a superchave da tabela. Então será a definição, do identificador principal do conjunto de dados

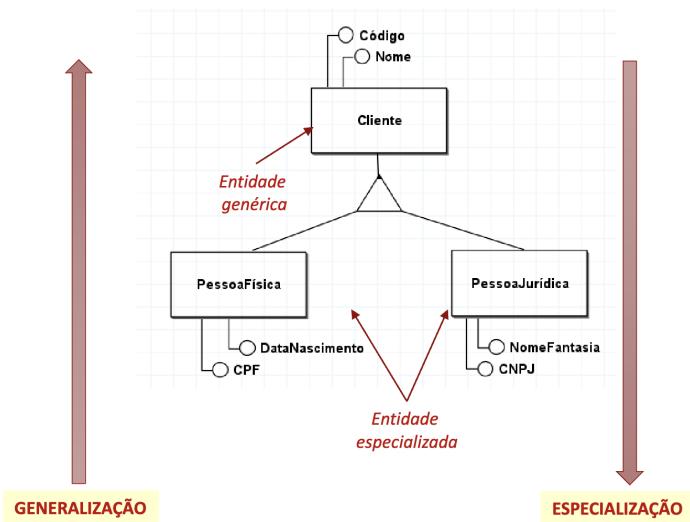
- **Chave Estrangeira**

Uma chave primária de outra entidade.

## Generalização/Especialização

### > Conceito

- Possibilita atribuir propriedades particulares a um subconjunto das ocorrências **especializadas** de uma **entidade genérica**.
- Exemplo: atributos das **entidades especializadas** modeladas
  - > **Pessoa Física:**
    - CPF
    - Sexo
    - Nome
    - ID (PK)
  - > **Pessoa Jurídica:**
    - CNPJ
    - Tipo
    - Nome
    - ID (PK)



## Relacionamentos

Entidades/Objetos, possuem relacionamento direto com outros, e devemos mapear isso primeiramente da seguinte forma:

#### > Levantamento de Requisitos

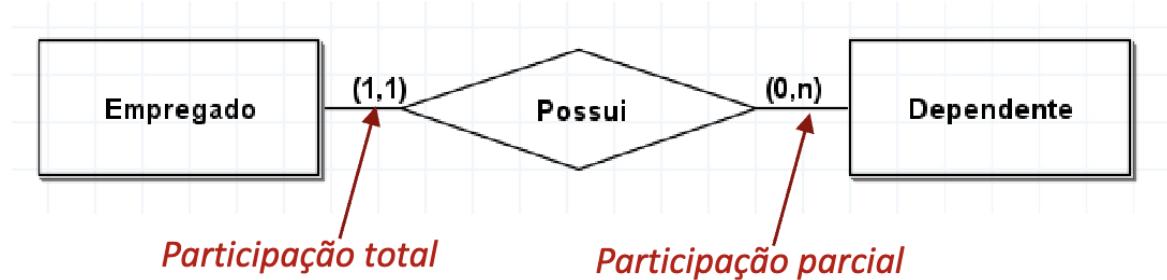
- Para identificar os relacionamentos no exemplo acima, você poderia perguntar:
  - > Como um empregado está relacionado com um departamento?
  - > Um empregado pode trabalhar em apenas um departamento?
  - > Um departamento possui mais de um empregado?
- Dependendo do tipo de resposta fornecida, um relacionamento poderá ser definido de três formas:
  - > um-para-um
  - > um-para-muitos
  - > muitos-para-muitos

## Cardinalidade

Expressa o número de entidades que a outra entidade se relaciona. Podendo ser os seguintes:

**(0, n)**

Pode se relacionar com nenhum ou muitos da entidade B, não sendo obrigatório o relacionamento.

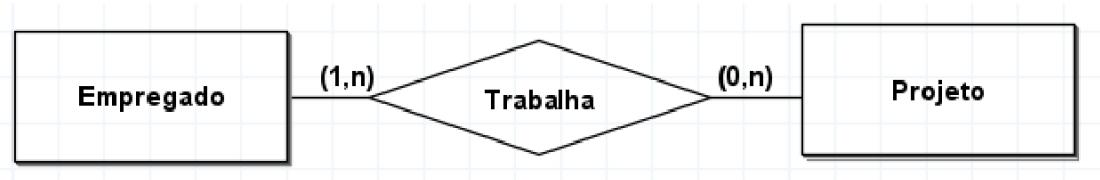


**(0, 1)**

Pode ou não se relacionar outra uma outra entidade.

**(1, n)**

Pode se relacionar com UMA ou MAIS com a entidade relacionada, observação que não pode não conter nenhum relacionamento nesse caso.



Um empregado pode trabalhar em 0 ou muitos projetos, porém um projeto é trabalhado por um ou mais empregados.

**(1, 1)**

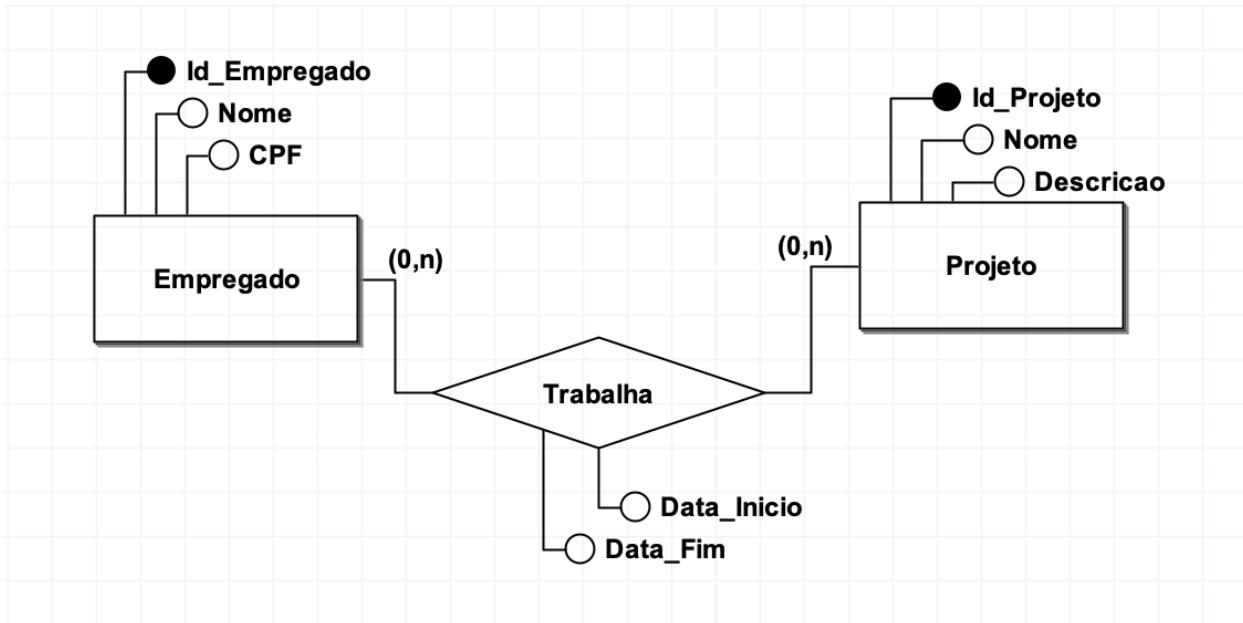
Uma entidade se relaciona obrigatoriamente com outra, sendo usada normalmente em entidades fracas.

**(n, n)**

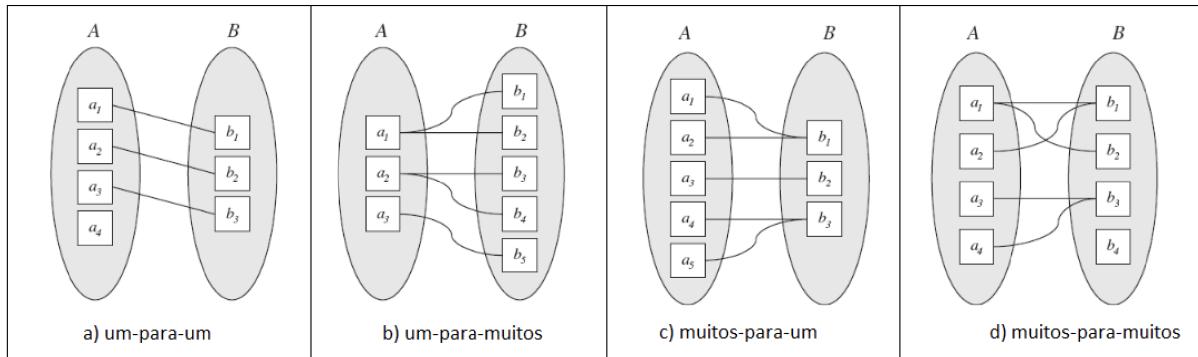
Muitos para muitos, esse tipo de relacionamento onde um empregado trabalha em vários projetos, e o projeto possui vários empregados trabalhando.

Esse tipo de relacionamento resulta em uma entidade assoativa como resultado do relacionamento. Essa entidade expressa ambos

os relacionamentos.



Cria uma tabela para realizar o relacionamento.



Note a semelhança de conjuntos também estudados em Matemática Discreta.

## Entidade Fraca x Identificadora

**Fracas:** São entidades que dependem de outras para existir, sozinhas não fazem sentido.

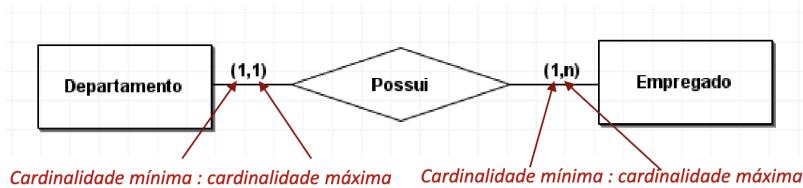
**Identificadora:** Identificam entidades fortes.

## Cardinalidade

Serve para definirmos o numero de ocorrência de uma entidade em outra.

Para definir corretamente a cardinalidade, devemos fazer algumas perguntas e levantar alguns requisitos .

- Por exemplo, dado um relacionamento entre Departamento e Empregado, pode-se fazer as seguintes perguntas:
  - > **Pergunta:** Um **Departamento** possui quantos **Empregados**?
    - *Resp.: No mínimo 1 e no máximo N.*
  - > **Pergunta:** Um **Empregado** está lotado em quantos **Departamentos**?
    - *Resp.: No mínimo em 1 e no máximo em 1.*



**Cardinalidade mínima:** Valor mínimo de ocorrência que pode se ocorrer no relacionamento

**Cardinalidade máxima:** Valor máximo de ocorrência que pode se ocorrer no relacionamento

**Ao levantar esses requisitos com o cliente, considere estes pontos:**

### › Levantamento de Requisitos

- Considere as seguintes questões:
  - › Um empregado pode não ter dependentes?
  - › Um dependente pode ter mais de um empregado?
  - › Determinado empregado pode possuir mais de um dependente?
  - › Pode existir dependente sem algum empregado associado?

## Entidade Associativa

Em relacionamentos muitos para muitos, é necessário uma expressão maior no relacionamento, feito por uma entidade(tabela) a parte, somente para isso.

## › Conceito

- Redefinição de um **relacionamento** que passa a ser tratado como sendo também uma **entidade**

- Exemplo:

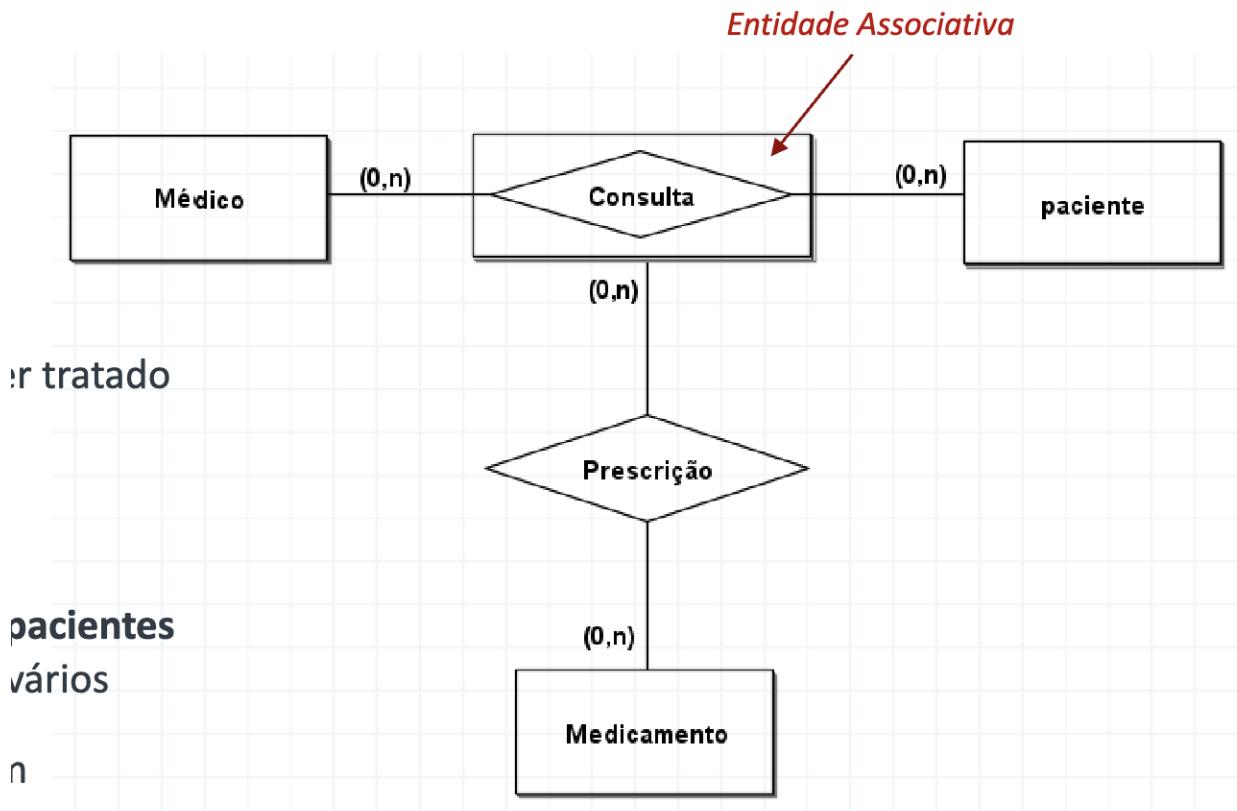
- › **Levantamento de Requisitos**

- Médicos realizam **consultas** com vários **pacientes**
    - Paciente podem realizar **consultas** com vários **médicos**
    - Durante as **consultas**, os **médicos** podem prescrever **medicamentos**
    - Um **medicamento** pode ser prescrito em várias **consultas**

- › **Considerações para o Modelo**

- Cada **consulta** pode gerar uma **prescrição de medicamento** diferente
    - A entidade **Medicamento** precisa se relacionar com a relação entre **Paciente** e **Médico**, que é **Consulta**.

**Resultado:**



ser tratado

**pacientes**  
vários

n

## Resumo anotações BrModelo

-  **Nova Entidade**
-  **Novo Relacionamento**
-  **Novo Auto Relacionamento**
-  **Nova Especialização**
-  **Nova Especialização (exclusiva)**
-  **Nova Especialização (duas entidades)**
-  **Nova União**
-  **Nova União (de duas entidades)**
-  **Nova Entidade Associativa**
-  **Novo Atributo**
-  **Novo Atributo Multivlorado**
-  **Nova Ligação**

**O que é auto relacionamento?**

**O que é especialização?**

**O que é União?**

Símbolo	Significado	Símbolo	Significado
	Entidade		Atributo composto
	Entidade fraca		Atributo derivado
	Relacionamento		Participação total de E2 em R
	Relacionamento de identificação		Razão de cardinalidade 1: N para E1 : E2 em R
	Atributo		Restrição estrutural (min, max) na participação de E em R
	Atributo-chave		
	Atributo multivalorado		

## Fluxo

1. Seguir um fluxo baseado no diagrama acima.
2. Levantar requisitos com o usuário
3. Levantar casos de usos
4. Implementar modelo conceitual.
5. Validar com o cliente o conceitual.
6. Criar modelo lógico implementando o conceito e dinâmicas do SGDB
7. Validar modelo lógico com as regras do banco.
8. Criar modelo físico.

# MySQL

## Tipos de dados do MySQL

Dados Temporais		
Tipo	Formato padrão	Valores permitidos
Date	AAAA-MM-DD	1000-01-01 a 9999-12-31
Datetime	AAAA-MM-DD HH:MI:SS	1000-01-01 00:00:00 a 9999-12-31 23:59:00
Timestamp	AAAA-MM-DD HH:MI:SS	1970-01-01 00:00:00 a 2037-12-31 23:59:00
Year	AAAA	1901 a 2155
Time	HHH:MI:SS	-838:59:59 a 838:59:59

Dados de Texto Não-Binário	
Tipo de texto	Numero máximo de bytes
Tinytext	255
Text	65.535
MediumText	16.777.215
LongText	4.294.967.295
Varchar	65.535
Char	255

Dados de Texto Binário	
Tipo de texto	Numero máximo de bytes
Tinyblob	255
Blob	65.535
Mediumblob	16.777.215
Longblob	4.294.967.295
Varbinary	65.535
Binary	255

Dados Numéricicos Inteiros		
Tipo	Escopo com sinal	Escopo sem sinal
Tinyint	-128 a 127	0 a 255
Smallint	-32.768 a 32.767	0 a 65.535
Mediumint	-8.388.608 a 8.388.607	0 a 16.777.215
Int	-2.147.483.648 a 2.147.483.647	0 a 4.294.967.295
Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	0 a 18.446.744.073.709.551.615

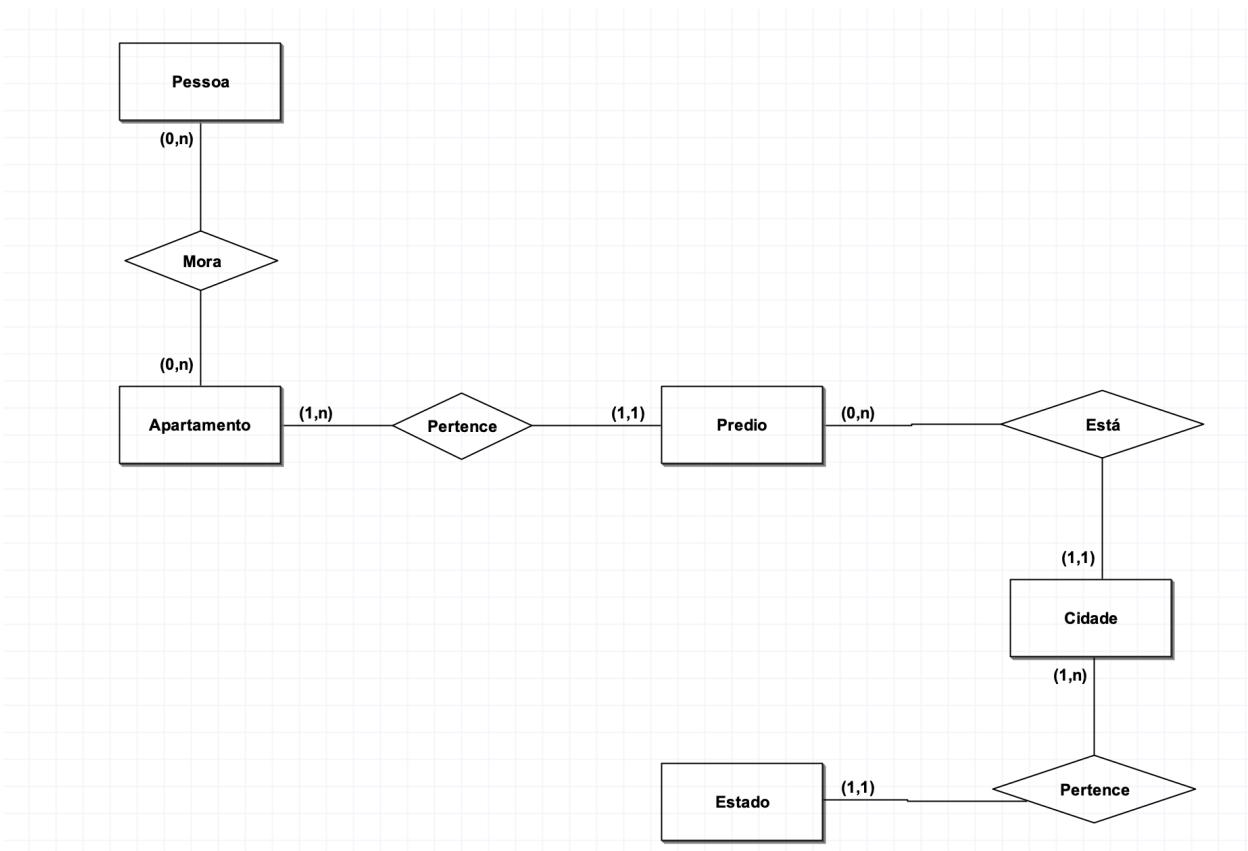
Dados Numéricicos (Bit e Boolean)	
Tipo	Numero máximo de bytes
bit	1
bool ou boolean	1

Dados Numéricicos de Ponto Flutuante e Ponto Fixo	
Tipo	Escopo numérico
Float(p,e)	-3,402823466E+38 a -1,175494351E-38 e de 1.175494351E-38 a 3,402823466E+38
Double(p,e)	-1,7976931348623157E+308 a -2,2250738585072014E-308 e de 2,2250738585072014E-308 a 1.7976931348623157E+308
Decimal(p,e)	-1,7976931348623157E+308 a -2,2250738585072014E-308 e de 2,2250738585072014E-308 a 1.7976931348623157E+308

## Aula-22/08

Select pode consulta ou realizar um processamento interno no banco de dados, fazendo uso de funções.

**Criação de modelos conceituais para treinar cardinalidade:**



- Instalar o MySql
- Seguir Slides
- Realizar Trabalhos
- Realizar projetos.

## Projeto Lógico dos Modelos Computacionais

1. O início de um projeto, deve-se gerar o modelo DER(Diagrama entidade relacionamento), um modelo conceitual independente do banco de dados, que deve ser o primeiro a ser gerado após a coleta de requisitos.
2. O modelo lógico é dependente do SGBD.

3. Por ultimo a implementação do Lógico, sendo uma implementação física.

## **1ª Fase - Projeto Conceitual**

**Gera o MODELO CONCEITUAL**

## **2ª Fase - Projeto Lógico**

**Gera o MODELO LÓGICO**

## **3ª Fase - Implementação**

**Gera o MODELO FÍSICO**

### **Modelo Relacional**

#### **Historia**

- › Criado nos anos 70 pelo matemático **Edgar F. Codd**, começou a ser usado comercialmente nos anos 80.
- › Considerado como o **primeiro modelo de dados para aplicações comerciais**.
- › Existe uma grande **base teórica e matemática** para este modelo, que apoia eficientemente o **projeto de banco de dados relacionais**
- › Permite um **processamento eficiente** das necessidades de informação de seus usuários.

## Visão Informal

- Aspecto Estrutural

Os dados de um banco relacional é representado como uma tabela.  
(Teoria do Conjunto)

- Aspecto de Integridade

As tabelas podem obdecer a restrições.

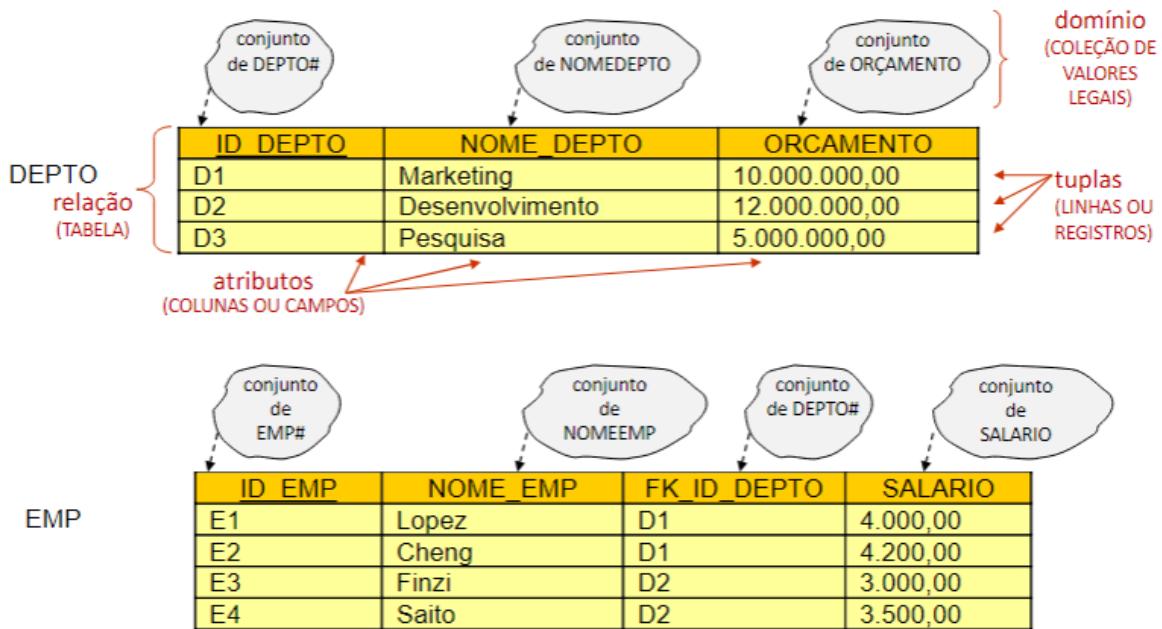
- Aspecto de Manipulação

Os operadores que o usuário utiliza para manipular as tabelas, resultam em outras tabelas de visualização.

## Terminologia

Termo Relacional Formal	Termo Informal Equivalente
<b>Relação</b>	<b>Tabela</b>
<b>Tupla</b>	<b>Linha ou registro</b> (par ordenado, ou trio ordenado, ou etc...)
<b>Atributo</b>	<b>Coluna ou campo</b> (cada coluna faz referência a um <b>Domínio</b> )
<b>Cardinalidade da Relação</b>	<b>Número de linhas ou registros</b> (quantidade de tuplas da relação)
<b>Domínio</b>	<b>Coleção de valores legais</b> (conjuntos de onde provêm os objetos relacionados)
<b>Chave Primária</b>	<b>Identificador único</b>

## Exemplos



## Esquema X Instância

## **Esquema da Relação**

Uma lista de atributos e seu domínio. Ou seja, um conjunto com o identificador sendo o atributo, mostrando e identificando a que esse conjunto de dados se refere, e o domínio que é o atomo de cada valor do conjunto.

Esquema = Conjunto de dados fechado/definido

## **Instância da Relação**

Variáveis de uma Relação, valores que podem ser modificados e possuem estado. Um Snapshot da relação **naquele momento**.

### › **Esquema**

- $A_1, A_2, \dots, A_n$  são **atributos**
- $R = (A_1, A_2, \dots, A_n)$  é o **esquema da relação**
- **Exemplo:**
  - › **Professor = (ID, Nome, CPF)**
  - › **Disciplina = (ID\_disciplina, nome, ementa, creditos)**

## **Restrições e Integridade**

Integridade em banco de dados se refere a precisão dos dados, e dados que não foram adulterados ou que obedeça a certas restrições.

- Exemplo:

- › Um campo **status** pode aceitar apenas valores de **1 a 100**
- › **CPF** é um campo de valor único, não se repete nas linhas da tabela

- São dados que respeitam as restrições pré-estabelecidas, e se comportam como tal.

### **Restrição**

Conhecido também como Constraint.

Regras que podem ser definidas e atributos, relações e + no banco de dados. Essas regras mantém os conjuntos de dados nas conformidades pré-estabelecidas, mantendo assim a integridade e garantindo um processamento adicional onde os dados são guardados de fato.

› NOT NULL

- Especifica que a coluna **não** pode conter valores nulos (**valor requerido**)

› PRIMARY KEY

- Especifica a coluna que identifica unicamente uma linha da tabela

› UNIQUE

- Especifica que os valores na coluna devem ser únicos
- Não são permitidos valores nulos

› CHECK

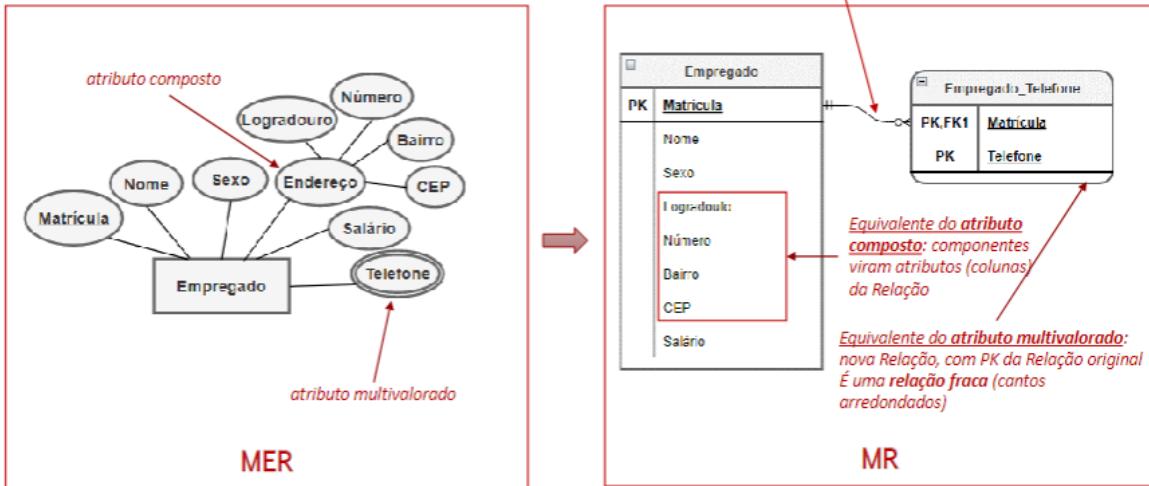
- Especifica regras para os valores da coluna

› Consistência em Banco de Dados

- Garante que todas as **restrições de integridade** de um banco de dados são **satisfitas**
- As **restrições asseguram** que as **mudanças feitas** no banco de dados, **por usuários autorizados**, **mantenham a consistência** de dados

## Mapeamento MER → MR

## > Transformações



- Note que o atributo composto é implementado junto à relação principal, mesclando tudo.
- Já o multivalorado, é como uma referência (0,n), onde o telefone estará em uma tabela a parte referenciando a relação de empregado, podendo assim conter quantos quiser (multivalorado).

**Prática formativa 1**

**Prática formativa 2**