

# Técnico em Desenvolvimento de Sistemas

## Banco de Dados

**Prof. Marcel Neves Teixeira**



# Plano de Ensino

## **Disciplina: Banco de Dados**

### **Objetivo da Disciplina e Perfil do estudante**

Visa propiciar a aquisição das competências específicas relativas a criação da estrutura para armazenamento, manipulação e persistência de dados, bem como o desenvolvimento das competências de gestão adequadas a diferentes situações profissionais.



# Plano de Ensino

## Disciplina: Banco de Dados

### Objetivo da Disciplina e Perfil do estudante

**Identificar** conceitos de Banco de Dados.

**Planejar** a modelagem conceitual e lógica de Banco de Dados.

**Aplicar** relacionamentos entre entidades e tabelas, tendo em vista padrões de normalização de dados.

**Criar** projetos de Banco de Dados para sistemas de software.



# Plano de Ensino

## Competências e Habilidades

- ✓ **Planejar** Projeto de Banco de Dados;
- ✓ **Criar** diagramas de Banco de Dados, tendo como base projetos de desenvolvimento de softwares.
- ✓ **Atuar** na fase de planejamento e suporte de Banco de Dados;
- ✓ **Garantir** que a solução de Banco de Dados seja normalizada e que atenda as necessidades do projeto de software.

# INTRODUÇÃO A BANCO DE DADOS



# Introdução a Banco de Dados

## Conceitos

**Dados**  
**X**  
**Informação**  
**X**  
**Conhecimento**





# Introdução a Banco de Dados

## Conceitos

### Dados

Toda e qualquer coisa existente em um ambiente empresarial que pode representar um evento, situação, antes de ser armazenado, ou até mesmo processado.

# Introdução a Banco de Dados

## Conceitos

### Informação

Conjunto de dados que podem ser representados de forma significativa e útil para alguém.

Uma frase, um relatório podem ser considerados como uma informação



# Introdução a Banco de Dados

## Conceitos

### Conhecimento

O conhecimento não pode ser descrito; o que se descreve é a informação.

Também não depende apenas de uma interpretação pessoal, como a informação, pois requer uma vivência do objeto do conhecimento.

# Introdução a Banco de Dados

## Conceitos

### O que é um banco de dados?

Um **banco de dados** é uma ferramenta para coletar e organizar informações. Os **bancos de dados** podem armazenar informações sobre pessoas, produtos, pedidos ou qualquer outra coisa. Muitos **bancos de dados** começam como uma lista em um programa de processamento de texto ou planilha.

# Introdução a Banco de Dados

## Exemplos de Banco de Dados

- ❖ Principais softwares de banco de dados existentes no mercado – **Antigos ou pouco usado hoje!**



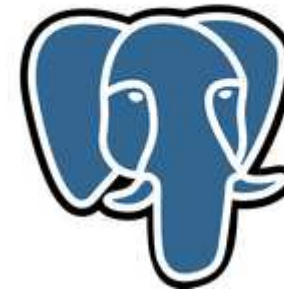
# Introdução a Banco de Dados

## Exemplos de Banco de Dados

- ❖ Principais softwares de banco de dados existentes no mercado – **Atuais, livres ou com custo variado e acessível.**



PostgreSQL



# Introdução a Banco de Dados

## Exemplos de Banco de Dados

- ❖ Principais softwares de banco de dados existentes no mercado – **Atuais e Pagos**



# Introdução a Banco de Dados

## Classificação dos Banco de Dados

- ❖ Com base no exemplos apresentados de banco de dados, podemos classificar os bancos de dados em:

### Relacional (SQL)



### Não Relacional (No-SQL)



# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Relacional

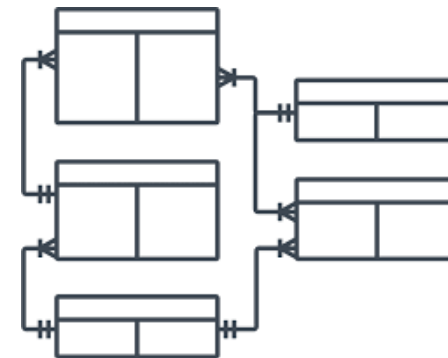
No banco de dados relacional nós teremos os dados sendo guardados em forma de tabelas. Dentro das tabelas os dados são organizados em colunas e em cada coluna contém um tipo de dado (strings, inteiros...).

# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Relacional

Os dados que irão para cada coluna são guardados como suas linhas. Além disso, no modelo relacional, primeiramente toda a estrutura do banco de dados deve ser projetada, caso os esquemas das tabelas não estejam definidos, não será possível inserir dados nelas.





# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Não Relacional

Banco de dados não relacional, ou **NoSQL**, pode ser visto como um padrão de armazenado alternativo ao modelo relacional, podendo oferecer uma maior **escalabilidade**.

E eles surgiram exatamente pela necessidade de se ter uma melhor **performance** e **alta escalabilidade**.

# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Não Relacional

Utilizando **NoSQL**, não se tem a necessidade de fazer toda a estrutura do banco relacional, pois, todas as informações serão agrupadas em um registro, então não precisa que se tenha relacionamentos entre várias tabelas para ser formada a informação, ela estará em sua totalidade no mesmo registro.

# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Não Relacional

Exemplo:

```
{
  "id": 55,
  "País": "Brasil",
  "Região": "América do Sul",
  "População": 201032714,
  "PrincipaisCidades": [
    {
      "NomeCidade": "São Paulo",
      "População": 1182876,
    },
    {
      "NomeCidade": "Rio de Janeiro",
      "População": 6323037,
    }
  ]
}
```

# Introdução a Banco de Dados

## Exemplos de Banco de Dados

### Banco de Dados Relacional ou Não Relacional

A escolha para um banco **relacional** ou **não relacional** estará vinculada a necessidade da aplicação, conforme o tipo de software e padrão que precise.

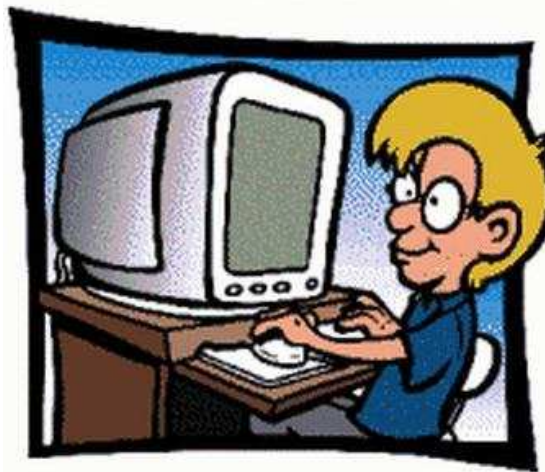
Em tecnologias mais volumosas em processamento de dados o **Não relacional** pode ser mais indicado, porém quando precisamos ter uma melhor organização dos dados o **relacional** é mais interessante.

# Introdução a Banco de Dados

## Quem são os profissionais da área?

**Usuários Finais** – são aqueles que irão utilizar o sistema de banco de dados, ou seja, que será beneficiado pela utilização da tecnologia.

### Usuário



# Introdução a Banco de Dados

## Quem são os profissionais da área?

**Programador** – é a pessoa responsável por desenvolver aplicações que utilizam um sistema de Banco de Dados.

Um programador hoje pode ganhar entre **R\$2.000** e **R\$8.000**

dependendo da empresa e da linguagem de programação

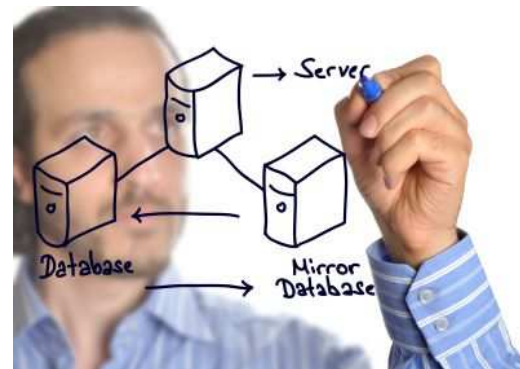


# Introdução a Banco de Dados

## Quem são os profissionais da área?

**Administrador de Banco de Dados (DBA)** – são as pessoas responsáveis por administrar o banco de dados, criar toda a estrutura necessária a atender a necessidade do cliente e cuidar de todas as funcionalidades necessárias para o seu bom funcionamento.

- ❖ Funções;
- ❖ Custo;



# Introdução a Banco de Dados

## Quem são os profissionais da área?

### Cargos e Salários da Evolução de um DBA

#### **DBA Júnior**

Tempo de trabalho: até 2 anos de experiência

Salário: Entre **R\$3.000 e R\$5.000**

#### **DBA Pleno**

Tempo de trabalho: De 2 a 4 anos de experiência

Salário: Entre **R\$5.000 e R\$8.000**

#### **DBA Senior**

Tempo de trabalho: De 4 a 6 anos de experiência

Salário: Entre **R\$8.000 e R\$10.000**



# Introdução a Banco de Dados

## Quem são os profissionais da área?

### Cargos e Salários da Evolução de um DBA



Senior Database Administrator  
(DBA): mensal

everis  
3 salários

R\$ 9.699/mês



Administrador De Banco De Dados  
(DBA) Sênior: mensal

Tivit  
3 salários

R\$ 10.653/mês



Administrador De Banco De Dados  
(DBA) Sênior: mensal

Algar Group  
3 salários

R\$ 13.749/mês



Senior Database Administrator  
(DBA): mensal

Banco do Brasil  
2 salários

Cerca de  
R\$ 8 mil - R\$ 12  
mil



# Introdução a Banco de Dados

## Certificação

### Oracle Certified Associate:

A credencial **Oracle Certified Associate (OCA)**, é normalmente o primeiro passo para alcançar a principal certificação Oracle Certified Professional. A credencial OCA garante que o profissional possui os conhecimentos e habilidades fundamentais, além de, uma sólida base de conhecimento para o suporte aos produtos da Oracle.

**Quantidade de Exames: 2**

**Custo médio de todas as provas: \$250**



# Introdução a Banco de Dados

## Certificação

### Oracle Certified Professional:

A credencial **Oracle Certified Professional (OCP)** é a referência de qualificação profissional e experiência técnica, e comprova os conhecimentos e habilidades necessárias para gerir, desenvolver ou implementar bases de dados corporativas, middleware ou aplicações. Cada vez mais, os gerentes de TI usam a credencial OCP para avaliar a qualificação profissional dos técnicos e candidatos a emprego.

**Quantidade de Exames: 4**

**Custo médio de todas as provas: \$375**



# Introdução a Banco de Dados

## Certificação

### Oracle Certified Master:

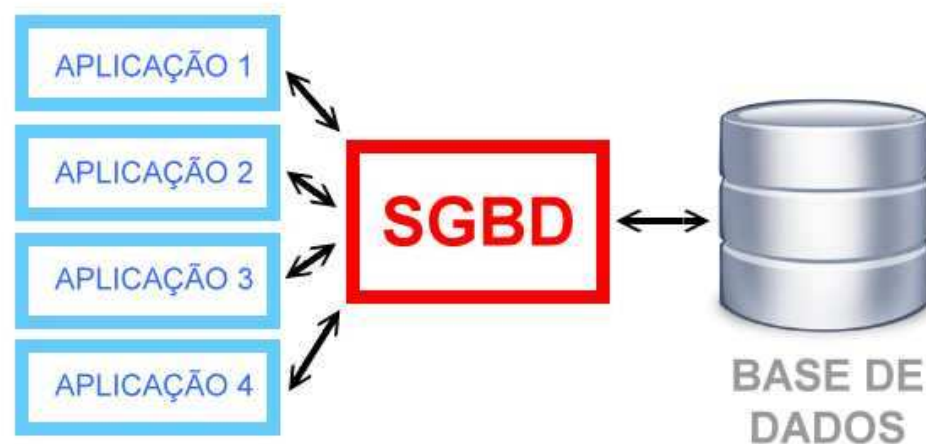
A credencial **Oracle Certified Master (OCM)** credencial reconhece o último nível avançado de habilidades, conhecimentos e capacidade comprovada. Esses profissionais são qualificados para responder às questões mais difíceis e resolver os problemas mais complexos.

**Quantidade de Exames: 5**

**Custo médio de todas as provas: \$2.500**



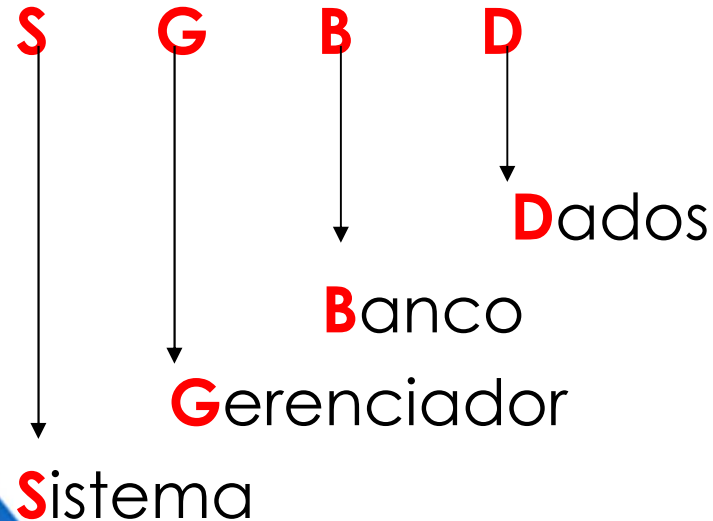
# Introdução ao SGBD e suas Regras e Características



# Introdução a Banco de Dados

## Conceitos ao SGBD

Sistema de BD, cujo objetivo principal é gerenciar o acesso e a correta manutenção dos dados armazenados em um banco de dados.



# Introdução a Banco de Dados

## Conceitos ao SGBD

Tudo que fazemos em um banco de dados passa pelo **SGBD**

O **SGBD** é responsável por salvar os dados no HD, manter em memória os dados mais acessados, ligar dados e metadados, disponibilizar uma interface para programas e usuários externos acessem o banco de dados.

# Introdução a Banco de Dados

## Conceitos ao SGBD

O **SGBD** também é responsável por encriptar dados, controlar o acesso a informações, manter cópias dos dados para recuperação de uma possível falha, garantir transações no banco de dados, enfim, sem o SGBD o banco de dados não funciona!





# Introdução a Banco de Dados



REGRAS

**para ser um  
SGBD !!!**

# Introdução a Banco de Dados

## Conceitos ao SGBD – Regras

Para saber se um Banco de dados realmente possui um SGBD, algumas regras precisam ser encontradas em sua estrutura.

### Auto-Contenção

O banco de dados não é apenas um repositório de dados e sim um ambiente para armazenamento os dados, relacionamentos dos mesmos e gerenciar suas possíveis formas de acesso formas de acesso.

# Introdução a Banco de Dados

## Conceitos ao SGBD – Regras

### Independência dos dados

Essa regra garante que toda a estrutura dos dados permaneça intacta mesmo quando utilizarmos aplicações externas para acessar e controlar os dados, ou seja, nenhuma estrutura dos dados pode ser alterada pela aplicação.

# Introdução a Banco de Dados

## Conceitos ao SGBD – Regras

### Abstração dos dados

Nessa regra em nenhuma situação o usuário final pode ter acesso direto ao banco de dados.

O banco de dados deve permitir que o usuário tenha acesso apenas aos dados que precise.

# Introdução a Banco de Dados

## Conceitos ao SGBD – Regras

### Visões

Cada usuário pode exigir ou mesmo precisar de uma visão diferenciada da base de dados.

# Introdução a Banco de Dados

## Conceitos ao SGBD – Regras

### Transações

Controle de concorrência, ou seja, o acesso ao mesmo dado deve ser de forma controlada e possível de acontecer.

# Introdução a Banco de Dados

**Vantagens**



**para de um  
SGBD !!!**

# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Segurança

#### **Evitar violação de consistência dos dados;**

- Segurança de acesso (usuários e aplicações);
- Segurança contra falhas (recovery);
- Monitoração de transações, categorias de falhas  
manutenção de histórico de atualizações (logs) e  
backups do BD.



# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Concorrência

Evitar conflitos de acesso simultâneo a dados por transações;

# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Controle de redundância

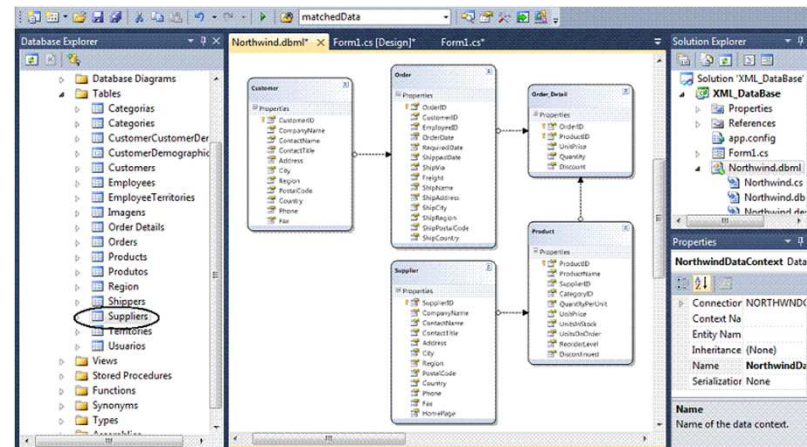
O SGBD deve controlar e não permitir a duplicação de informações, seja por erro do usuário ou até mesmo falhas em seu SGBD.

# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Interfaceamento

Um banco de dados deverá disponibilizar formas de acesso gráfico a sua linguagem natural SQL



# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Métodos de acesso

**DDL (Data Definition Language)** - especificação do esquema do BD (dados e seus tipos de dados, índices, etc), no metodo DDL podemos usar **(Create, Alter, Drop)**



# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Métodos de acesso

**DML (Data Manipulation Language)** - manipulação de dados (**Insert, Update, Delete, Select**) processamento eficaz de consultas considera relacionamentos, predicados de seleção, volume de dados, índices.



# Introdução a Banco de Dados

## Características de um SGBD – Regras

### Métodos de acesso

**DCL (Data Control Language)** responsável pelas permissões de acesso a base de dados.



# Introdução a Banco de Dados

## Vantagens de um SGBD – Regras

- **Rapidez** na manipulação e no acesso a informação
- **Redução** de esforço humano;
- **Redução** da **redundância** e da inconsistência de informações;
- **Redução** de **problemas** da integridade;
- Compartilhamento de dados;

# Introdução a Banco de Dados

## Vantagens de um SGBD

- Aplicação automática de restrições de **segurança**;
- **Controle integrado** de informações distribuídas fisicamente.
- **Precisão** no resultado da informação.
- Maior **Disponibilidade** dos dados.
- **Tempo** de desenvolvimento reduzido do software



# Introdução a Banco de Dados



**Vamos pensar  
um pouco ....**

# Introdução a Banco de Dados

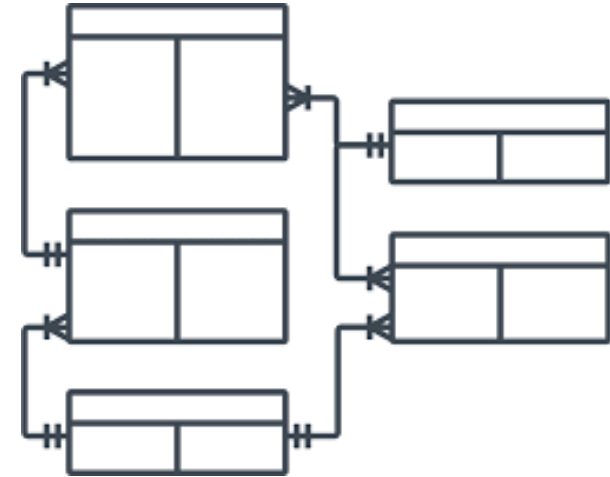
## Conceito

### O que é um G.A.?

Todo e qualquer software de BD que não atenda as regras de um SGBD.



# **Modelo Entidade de Relacionamentos – MER OU Modelo Conceitual**





# Modelo Entidade-Relacionamento

## Modelagem

**Modelagem** é uma forma de obtermos resultados e esquemas puramente conceituais sobre a essência de um sistema, projeto de negócio que se esta desenvolvendo.

# Modelo Entidade-Relacionamento

## Modelo Conceitual

**Modelo Conceitual** é uma representação gráfica do problema levantado junto ao cliente em uma visão técnica, nesse momento ainda não conseguimos visualizar a estrutura real que o banco deverá ter, porém possibilita ter uma ideia de como a estrutura será integrada.

# Modelo Entidade-Relacionamento

## Entidades

**Entidade** define-se como um objeto que existe no mundo real com uma identificação distinta e com um significado próprio.

Uma entidade é como uma representação de uma Classe de dados do negócio, ou seja, um conjunto de informações de mesmas características.

# Modelo Entidade-Relacionamento

A representação de uma **Entidade** no modelo ER se realiza através de um retângulo, com o nome desta entidade em seu interior.



Cliente

Produto

# Modelo Entidade-Relacionamento

## Instâncias

As **instâncias** de uma entidade não são representadas no diagrama de ER, mas são semanticamente interpretadas no mesmo.



# Modelo Entidade-Relacionamento

Uma **instância** nada mais é do que cada linha de uma tabela, vamos imaginar que cada entidade é uma tabela e nela existem linhas com dados distintos, essas linhas caracterizam-se “Instâncias”.

Portanto a instancia são os dados e a informação que a entidade deve representar.

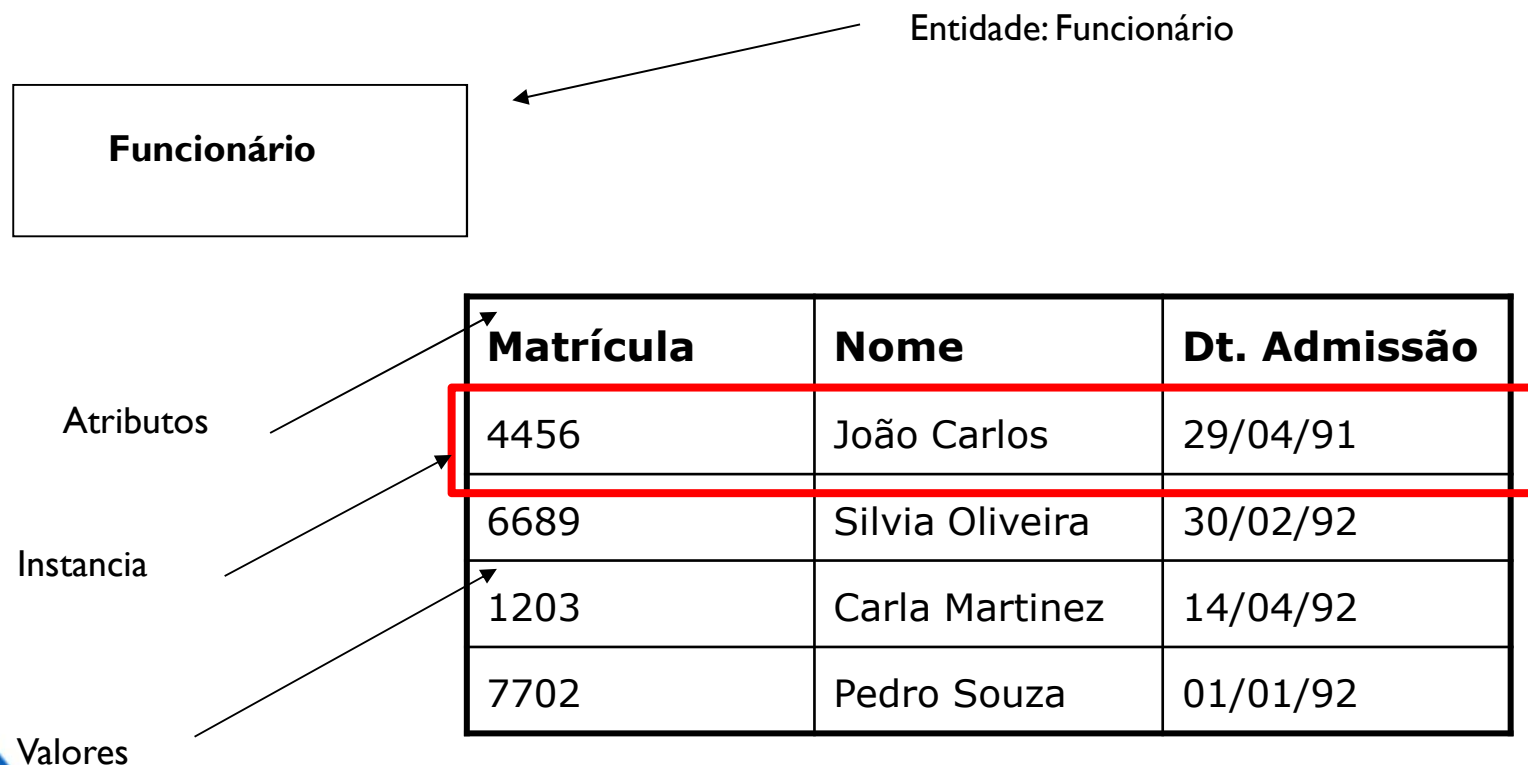
# Modelo Entidade-Relacionamento

## Atributos - Simples

Todo objeto para ser uma entidade possui propriedades que são descritas por **atributos** e **valores**. A junção desses atributos e valores descrevem o conceito de uma instância.

# Modelo Entidade-Relacionamento

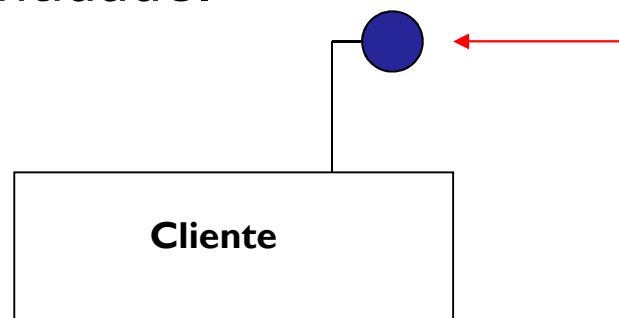
Ex:



# Modelo Entidade-Relacionamento

## Atributos – Identificador ou Determinante

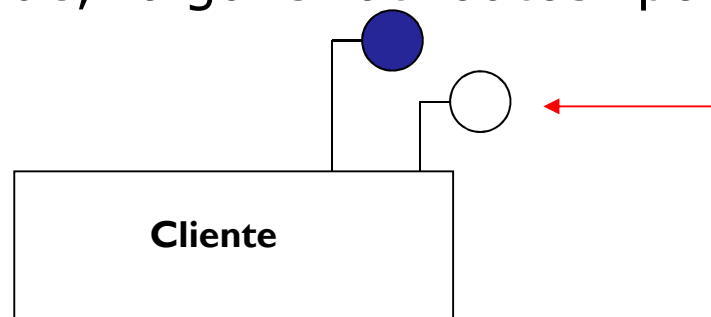
Representado através de uma bola cheia na extremidade do atributo. Atributos identificadores identificam ou compõe a identificação única de uma ocorrência em uma entidade.



# Modelo Entidade-Relacionamento

## Atributos – Não Identificados

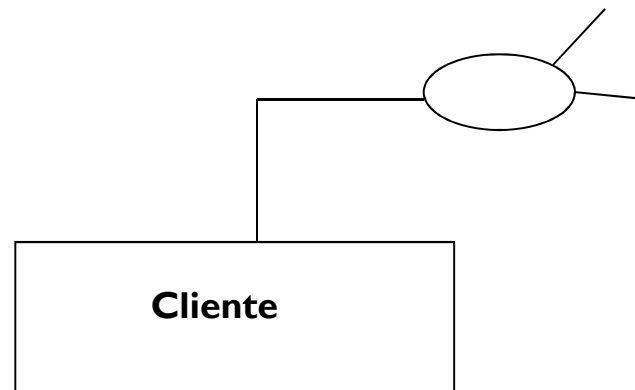
Representado através de uma bola vazia na extremidade do atributo, atributos não identificados podem ser opcionais, ou seja, em algumas instâncias de entidade, alguns atributos poderão conter valores nulos.



# Modelo Entidade-Relacionamento

## Atributos - Composto

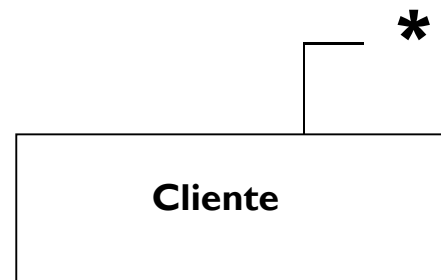
Por exemplo quando temos um atributo chamado “endereço”, seu conteúdo poderá ser dividido em vários outros atributos, como: **Rua, Número, Complemento, Bairro, Cep e Cidade.**



# Modelo Entidade-Relacionamento

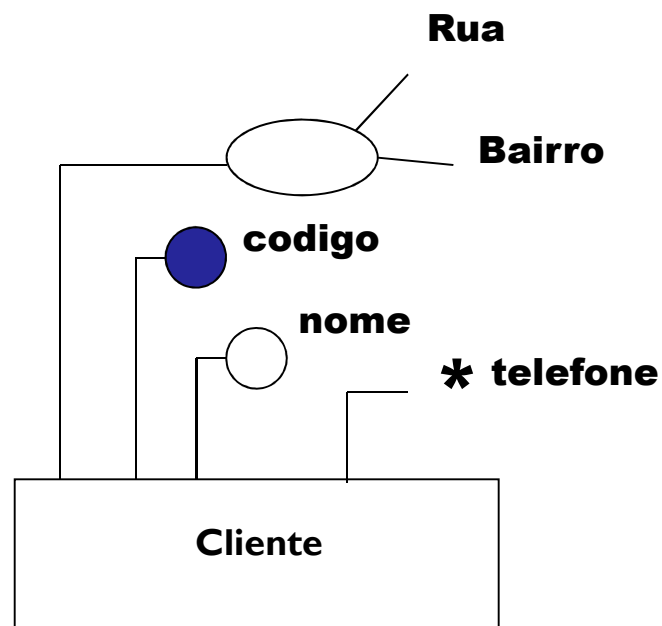
## Atributos - Multivalorados

Por exemplo o atributo **telefone**, poderá ter mais de um número de **telefone**. Esse é indicado colocando-se um asterisco precedendo o nome do atributo.



# Modelo Entidade-Relacionamento

## Exemplo das entidades com seus atributos







# Modelo Entidade-Relacionamento

## Identificando Entidades

Em um primeiro contato com um negócio estamos realizando o processo de levantamento de dados, nessa etapa é o momento de identificarmos as primeiras entidades de um projeto.

# Modelo Entidade-Relacionamento

## Exemplo Prático

“Uma clínica médica necessita controlar consultas médicas realizadas e marcadas pelos médicos a ela vinculados, assim como acompanhar quem são os pacientes atendidos para assim manter um histórico dos mesmos.”

# Modelo Entidade-Relacionamento

## Exemplo Prático

“Um escritório de advocacia necessita criar um sistema para controlar os seus advogados e precisa saber quais são os clientes do escritório. Lembrando que cada advogado atua em determinada área de trabalho, portanto é de grande importância saber quais são os clientes de cada advogado.”

# Modelo Entidade-Relacionamento

## Generalização

Sempre que estamos criando o **modelo conceitual**, temos que tomar o máximo de cuidado, pois muitas vezes uma entidade acaba sendo generalizada e pode estar comprometendo a estrutura do banco de dados.

# Modelo Entidade-Relacionamento

## Generalização

Existem dois modelos de **Generalização** em Banco de Dados:

- Parcial – representado pela letra **P**
- Total – representado pela letra **T**

# Modelo Entidade-Relacionamento

## Generalização Parcial

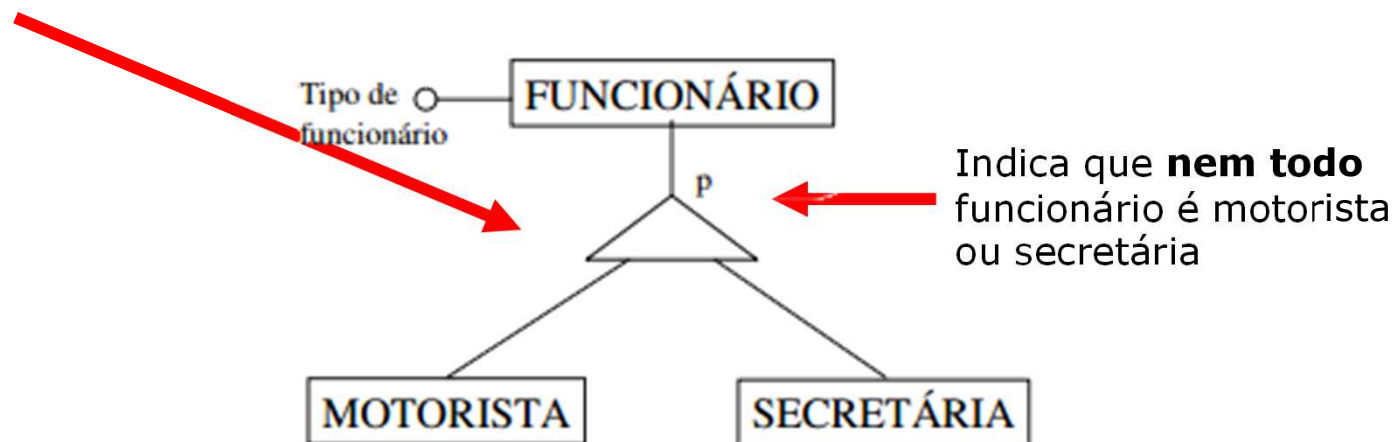
Exemplo:

A entidade “**Funcionário**” imagina-se que temos um sistema onde teremos vários tipos de funcionários que precisam ser controlado como **motoristas** e **secretárias**.

# Modelo Entidade-Relacionamento

## Generalização Parcial

Exemplo:



# Modelo Entidade-Relacionamento

## Generalização Total

Exemplo:

A entidade “**Cliente**” imagina-se que temos um sistema onde precisaremos controlar os clientes **pessoa física** e clientes **pessoa jurídica**.



# Modelo Entidade-Relacionamento

## Generalização Total

Exemplo:



# Modelo Entidade-Relacionamento

## Relacionamento entre entidades

**Relacionamento** é o acontecimento que liga dois objetos.

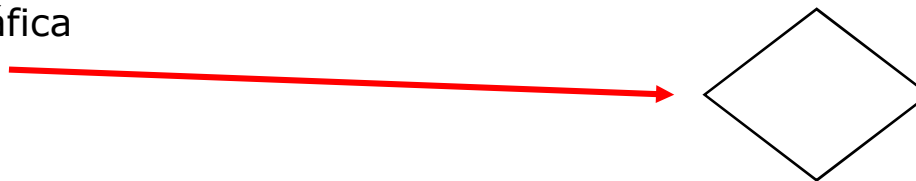
Considerando o conceito de estrutura de banco de dados, podemos dizer que a junção de duas **entidades** ou duas **tabelas** conceitua-se relacionamento.

# Modelo Entidade-Relacionamento

## Grau de Relacionamento ou Cardinalidade

Quando temos um relacionamento entre duas entidades, o número de ocorrências de outra entidade, determina o grau do relacionamento.

Representação Gráfica



# Modelo Entidade-Relacionamento

## Relacionamento de “Um-para-Um”

Neste grau cada elemento de uma entidade relaciona-se com **somente um elemento** de outra entidade.

1º



# Modelo Entidade-Relacionamento

## Relacionamento de "Um-para-Um"

2º



3º



# Modelo Entidade-Relacionamento

## Relacionamento de “Um-para-Muitos”

Este grau de relacionamento é um dos mais comuns no mundo real, sendo o que denominamos de relacionamento básico entre entidades, onde **um item de uma entidade pode se relacionar com vários itens** da outra entidade.

# Modelo Entidade-Relacionamento

## Relacionamento de “Um-para-Muitos”

Um elemento da entidade A relaciona-se com muitos elementos da entidade B, mas cada elemento da entidade B somente pode estar relacionado a um elemento da entidade A.

1°



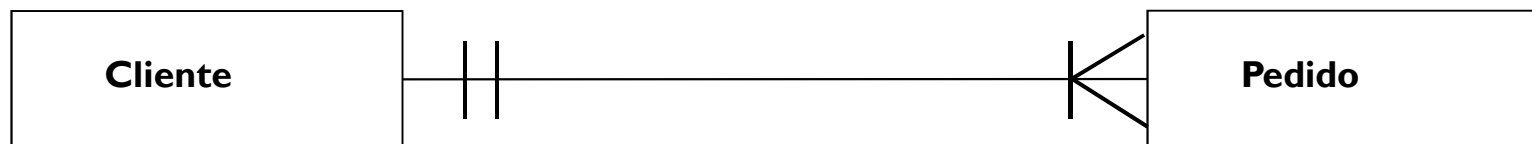
# Modelo Entidade-Relacionamento

## Relacionamento de "Um-para-Muitos"

2º



3º





# Modelo Entidade-Relacionamento

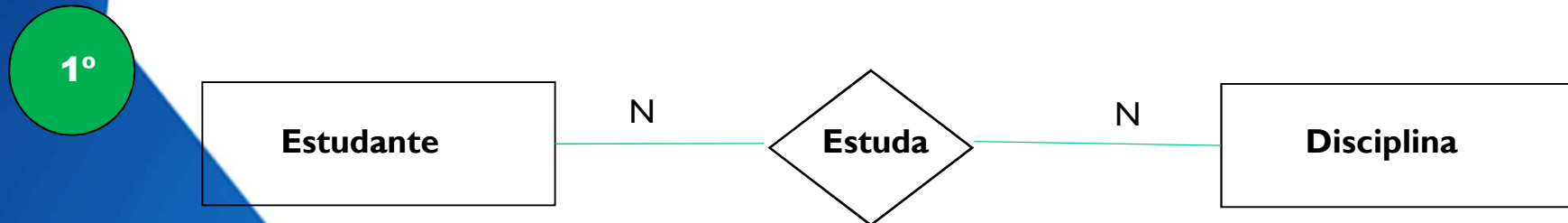
## Relacionamento de “Muitos-para-Muitos”

Um elemento da entidade A relaciona-se com muitos elementos da entidade B, mas cada elemento da entidade B pode estar relacionado a muitos elementos da entidade A.

# Modelo Entidade-Relacionamento

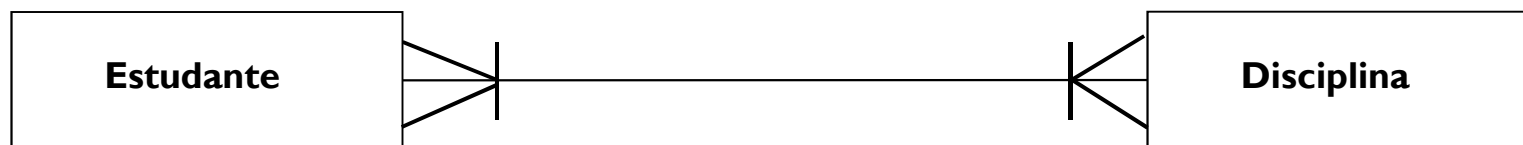
## Relacionamento de "Muitos-para-Muitos"

No exemplo abaixo vemos a associação entre a entidade **Estudante** e a entidade **Disciplina**:



# Modelo Entidade-Relacionamento

## Relacionamento de "Muitos-para-Muitos"



# Relacionamentos - Exercício

**Objetivo: Desenvolver o modelo conceitual para um hospital veterinário.**

A clinica veterinária possui as seguintes características:

- Os clientes podem possuir um ou vários animais que são atendidos.
- Cada animal pode sofrer de uma ou várias enfermidades.
- Cada medico veterinário é especialista em uma espécie de animal, exemplo: (gato, cachorro, aves, etc).
- Os atendimentos são realizados por um único veterinário, porém existem casos em que um animal é atendido por mais de um veterinário.

# Relacionamentos - Exercício

**Objetivo: Desenvolver o modelo conceitual para uma empresa de consultoria.**

- A empresa possui colaboradores que estão distribuídos em departamentos dentro da organização, sendo que cada departamento é responsável por uma área de atuação da empresa.
- Cada trabalho executado pela empresa constitui-se um projeto e a cada projeto estão associados diversos colaboradores trabalhando.
- Um colaborador pode participar de mais de um projeto simultaneamente.
- Todo projeto da empresa sempre terá um colaborador responsável pela sua execução.

# Relacionamentos - Exercício

**Objetivo: Desenvolver o modelo conceitual para uma corretora de valores.**

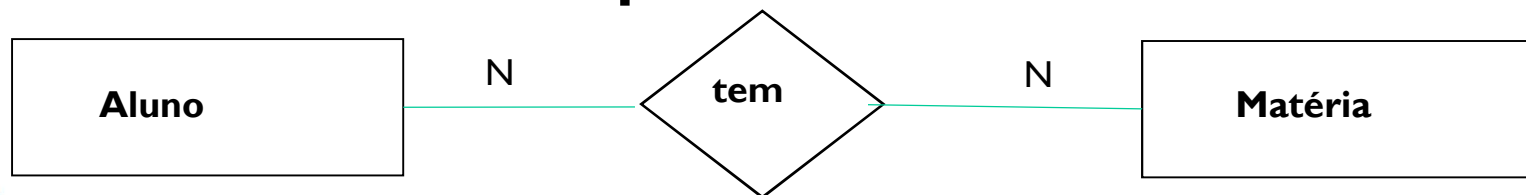
- Uma corretora de valores possui vários operadores de bolsa, também chamados de assessores.
- A corretora possui vários clientes e cada cliente é atendido exclusivamente por um assessor.
- Para a operação da bolsa cada cliente possui uma conta corrente, que pode utilizar para depositar valores e realizar a compra de ações.
- As ações possuem um código de negociação que as identificam exclusivamente, e um cliente pode comprar uma ou várias ações.

# Modelo Entidade-Relacionamento

## Agregação

Existem momentos em que temos uma visão dos dados que nos deixa em dúvida de como representa-los.

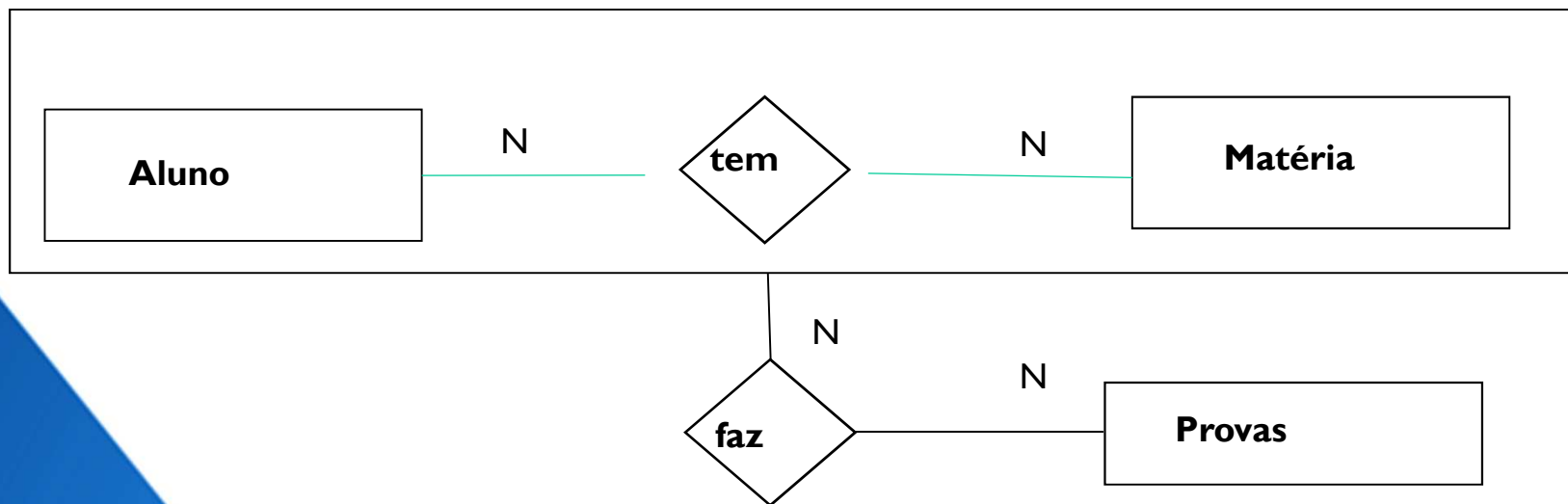
Por exemplo, no modelo abaixo precisamos encaixar a entidade **prova**:



# Modelo Entidade-Relacionamento

## Agregação

Nesse exemplo a entidade **prova** poderia se encaixar tanto na entidade **aluno** como na entidade **matéria**. Assim podemos criar o modelo da seguinte forma:

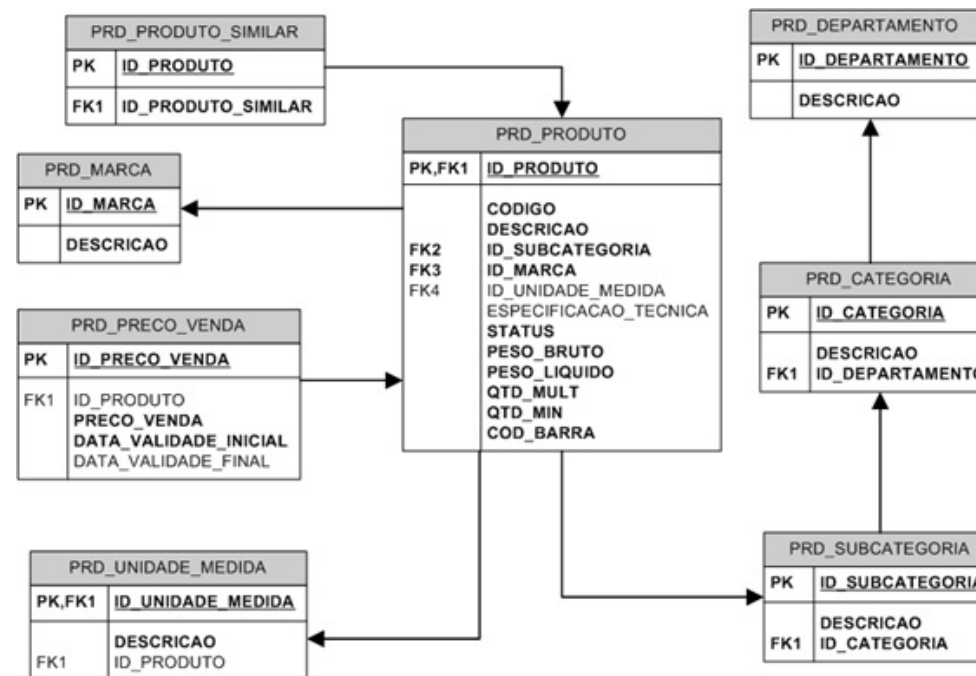




# DER

## DER

### (Diagrama e Entidade de Relacionamento)



# Diagrama Entidade Relacionamento

## Conceito

É a transformação do **(MER) modelo entidade de relacionamento** em representação utilizando a forma de tabelas que também pode ser chamado de **Modelo Lógico ou DER (diagrama Entidade Relacionamento)**.

# Diagrama Entidade Relacionamento

## DataBase



É um repositório onde serão armazenados todos os elementos de um Banco de Dados, o database deve ser relacionando sempre ao assunto do sistema, ou parte do sistema.

Obs: Os databases devem ser referenciados com o prefixo db.

**Ex: dbvideolocadora , dblivraria**

# Diagrama Entidade Relacionamento

## Tabelas

A tabela é um conjunto de **registros** de mesmo tipo. Um banco de dados pode ser formado por **uma** ou **mais** tabelas.

Cada tabela deve ter um **nome** e deve ser **único**, para assim organizar as informações no Banco de Dados.

**Ex: tabela para clientes, fornecedores, produtos**

# Diagrama Entidade Relacionamento

## Tabelas

O nome que atribuímos a uma tabela deve ser padronizado, para assim definirmos o que é aquele objeto, por exemplo:

**Ex: podemos identificar uma tabela com o prefixo “tbl”**

**Clientes = tblclientes**

**Fornecedores = tblfornecedores**

**Produtos = tblprodutos**

# Diagrama Entidade Relacionamento

## Campos

É a menor unidade destinada ao armazenamento de valores (dados).

Cada campo deve conter o seu respectivo tipo de dado.

Ex: Na tabela **tblclientes**

Campo: **nome** – deve guardar o nome do cliente;

**endereco** – deve conter o endereço do cliente;

**bairro** – deve conter o bairro do cliente.

***Note** que não devemos colocar acentos, pontos e nem símbolos em nome de campos*

# Diagrama Entidade Relacionamento

## Campos

Nome	Endereco	Bairro	Cidade	Estado
Jogos Educar	Av. das Nações, 280	Jd. América	Atibaia	SP

Perceba que cada campo deve armazenar o seu respectivo valor (dados).

# Diagrama Entidade Relacionamento

## Campos

Cada campo da sua tabela deve receber o seu respectivo **tipo de dado**.

O tipo de dado permite identificar como o Banco de Dados deve tratar aquele valor, por exemplo:

Ex: O campo nome deve ser criado permitindo guardar "**string**", já o campo data de nascimento deve guardar "**date ou datetime**", um campo valor do produto deve ser "**float**" e o campo quantidade de itens deve receber "**integer**"



# Diagrama Entidade Relacionamento

## Campos e Tipos de dados

**Texto;**  
**Numérico;**  
**Data;**  
**Real;**  
**Auto Incremento;**  
**Booleano.**



**Os tipos de dados podem variar conforme o software do banco de dados utilizado.**

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Texto

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

*Tipos de dados mais utilizados na criação de uma tabela*

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Inteiro

Dados Numéricos Inteiros	
Tipo	Escopo com sinal
Tinyint	-128 a 127
Smallint	-32.768 a 32.767
Mediumint	-8.388.608 a 8.388.607
Int	-2.147.483.648 a 2.147.483.647
Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

*Tipos de dados mais utilizados na criação de uma tabela*

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Booleano

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

*Tipos de dados mais utilizados na criação de uma tabela*

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Real

*Tipos de dados mais utilizados na criação de uma tabela*

Dados Numéricos 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

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Data

*Tipos de dados mais utilizados na criação de uma tabela*

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

# Diagrama Entidade Relacionamento

## Tipos de dados do tipo: Auto Incremento

É um tipo de dados sempre “**integer**” que o próprio Banco de Dados realiza o seu incremento conforme cada registro é inserido na tabela.

# Diagrama Entidade Relacionamento

## Registros

**Registro** é um conjunto de campos que são criados em uma estrutura de tabela que fornecem um significado ou informação.

Nome	Fone	Cidade
João Silva	743-1234	Natal
Maria Penha	(084)9976-1213	Mossoró
Alberto Carlos Jorgeano Lucena	3321-1209	Assu



# Diagrama Entidade Relacionamento

## Hierarquia do BD

Um Banco de dados recebe a seguinte hierarquia:



# Diagrama Entidade Relacionamento

## Chave Primária

**Chave primária** é o atributo cujo valor jamais irá se repetir, a chave primária permite a individualidade de cada instância, otimiza busca de informações no banco de dados, permite gerar relacionamentos com outras entidades, etc.



# Diagrama Entidade Relacionamento

## Chave Primária

**Ex:** Existem dois registros iguais, porém, de pessoas diferentes, através da chave primária podemos identifica-la.

Nome	Cidade
João Silva	Mossoró
Roberto Carlos	Cachoeiro do Itapemirim
Marilia Pereira	Rio de Janeiro
Eduardo Cullen	Forks
Isabela Swan	Forks
João Silva	Mossoró
Harry Potter	Londres
Frodo Baggins	Condado

# Diagrama Entidade Relacionamento

## Chave Primária



**Ex:** Devemos ter um campo que seja único e que possa identificar quem é cada registro.

Código	Nome	Cidade
01	João Silva	Mossoró
02	Roberto Carlos	Cachoeiro do Itapemirim
03	Marilia Pereira	Rio de Janeiro
04	Eduardo Cullen	Forks
05	Isabela Swan	Forks
06	João Silva	Mossoró
07	Harry Potter	Londres
08	Frodo Baggins	Condado

# Diagrama Entidade Relacionamento

## Chave Primária

### Regras para identificarmos a chave primária

- Campo único, ou seja, que não se repita;
- Campos com tamanhos pequenos, pois facilita no BD;
- Campo que nunca pode ser alterado;
- Campo que nunca seja do tipo texto;
- Campos que sejam de preenchimento obrigatório;
- Caso não haja um campo com essas características defina para o próprio BD crie e gerencie esse campo, usando o tipo de dados "Auto-Incremento".



# Diagrama Entidade Relacionamento

## Chave Estrangeira

**Chave estrangeira** permite criar o relacionamento entre tabelas utilizando a chave primária.

Uma **chave primária** transforma-se em **chave estrangeira** em outra tabela para criar a interligação entre as mesmas.

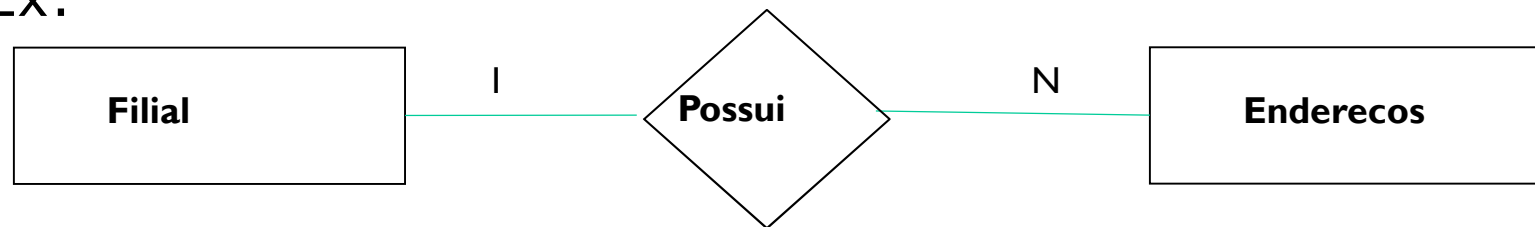


# Diagrama Entidade Relacionamento

## Chave Estrangeira – Cardinalidade 1 x N

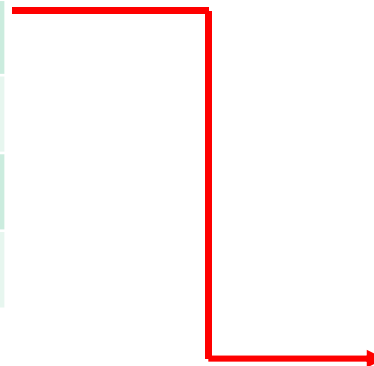


Ex:



TblFilial
Codigo
Nome
CNPJ
Telefone

TblEnderecos
Codigo
Rua
Bairro
Cep
CodFilial

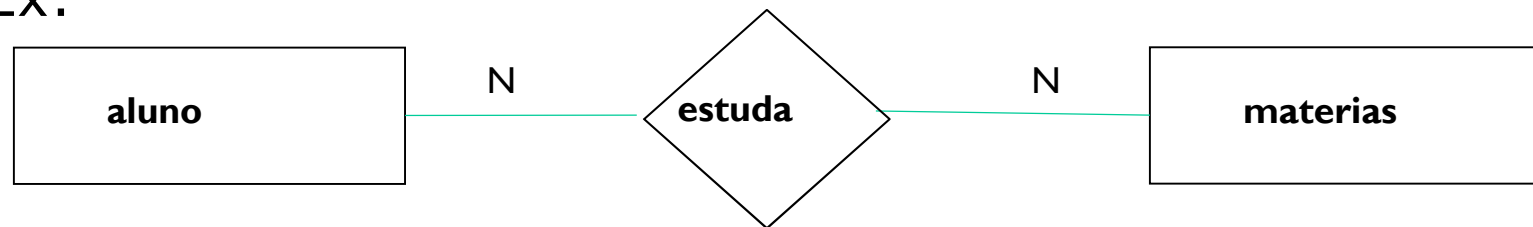


# Diagrama Entidade Relacionamento

## Chave Estrangeira – Cardinalidade N x N



Ex:



Tblaluno
Codigo
Nome
CPF
Telefone

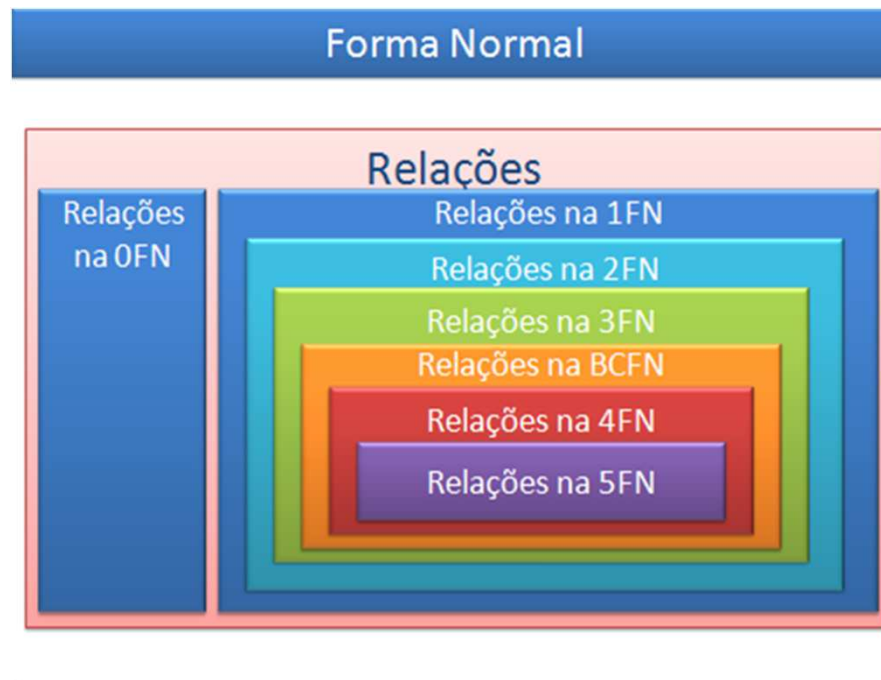
Tblaluno_materia
Codigo
CodAluno
CodMateria

Tblmaterias
Codigo
Nome
Qtde_Horas
Conteudo



# Normalização de Banco de Dados

## As 5 Formas Normais de Banco de Dados.





# Normalização de Banco de Dados

Quando falamos em normalização logo entendemos que nosso banco de dados deve ser estruturado obedecendo procedimentos e regras do **Modelo Relacional de Dados**.

# Normalização de Banco de Dados

A regra de **ouro** que devemos observar no projeto de um banco de dados baseado no **Modelo Relacional de Dados** é a de "não misturar assuntos em uma mesma Tabela".

**Ex:** Por exemplo: na Tabela Clientes devemos colocar somente campos relacionados com o assunto Clientes. Não devemos misturar campos relacionados com outros assuntos, tais como Pedidos, Produtos, etc.

# Normalização de Banco de Dados

Vejamos no exemplo abaixo a **entidade Filmes** quando adicionamos alguns dados nos atributos, quantas informações estão sendo repetidas.

Filmes

Entidade: Filmes							
idFilme	Nome	Gênero	idMidia	Idioma	Tipo	Seção	Preço
656565	Uma janela suspeita	Drama	1001	dub	DVD	25	3,50
656565	Uma janela suspeita	Drama	1002	dub	VHS	25	3,50
656565	Uma janela suspeita	Drama	1003	leg	DVD	25	3,50
656565	Uma janela suspeita	Drama	1004	leg	VHS	25	3,50
323232	Minority Report	Ficção	2550	dub	DVD	32	4,20
323232	Minority Report	Ficção	2550	dub	VHS	32	4,20
323232	Minority Report	Ficção	2550	leg	DVD	32	4,20
323232	Minority Report	Ficção	2550	leg	VHS	32	4,20

# Normalização de Banco de Dados

**Normalmente** após a aplicação das **regras de normalização** de dados, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente previsto.

Apenas repetimos os Números que identifica cada item.

Entidade: Filmes		
idFilme	Nome	Gênero
656565	Uma janela suspeita	Drama
323232	Minority Report	Ficção

Entidade: Mídias					
idMidia	idFilme	Idioma	Tipo	Seção	Preço
1001	656565	dub	DVD	25	3,50
1002	656565	dub	VHS	25	3,50
1003	656565	leg	DVD	25	3,50
1004	656565	leg	VHS	25	3,50
2550	323232	dub	DVD	32	4,20
2550	323232	dub	VHS	32	4,20
2550	323232	leg	DVD	32	4,20
2550	323232	leg	VHS	32	4,20

# Normalização de Banco de Dados

## **Benefícios quando aplicamos a Normalização em BD:**

- Minimização de redundâncias e inconsistências;
- Facilidade de manipulações do banco de dados;
- Ganho de performance no SGBD;
- Facilidade de manutenção do sistema de Informação;
- Entre outros.

# Normalização de Banco de Dados

Atualmente existem 5FNs, porém na prática aplicamos apenas 3, conforme segue abaixo:

- Primeira Forma Normal (1FN);
- Segunda Forma Normal (2FN);
- Terceira Forma Normal (3FN);

# Normalização de Banco de Dados

## Primeira Forma Normal (1FN)

Os procedimentos mais recomendados para aplicar a 1FN são os seguintes:

- a) **Identificar** a chave primária da entidade;
- b) **Identificar** o grupo repetitivo e removê-lo da entidade;
- c) **Criar uma nova entidade com a chave primária** da entidade anterior e o grupo repetitivo.

A chave primária da nova entidade será obtida pela concatenação da chave primária da entidade inicial e a do grupo repetitivo.



# Normalização de Banco de Dados

## Primeira Forma Normal (1FN)

Código_cliente	Nome	Telefone	Rua	Bairro	Cep
C001	José	9563-6352 9847-2501	Rua Seis, 85	Morumbi	12536-965
C002	Maria	3265-8596	Rua Onze, 64	Moema	65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos	Liberdade	68858-633

Sem aplicar a normalização.

Código_cliente	Nome	Rua	Bairro	Cep
C001	José	Rua Seis, 85	Morumbi	12536-965
C002	Maria	Rua Onze, 64	Moema	65985-963
C003	Janio	Praça ramos	Liberdade	68858-633

Aplicando apenas a 1FN.

*Tabela na primeira forma normal*

Código_cliente	Telefone
C001	9563-6352
C001	9847-2501
C002	3265-8596
C003	8545-8956
C003	9598-6301

# Normalização de Banco de Dados

## Segunda Forma Normal (2FN)

Os procedimentos mais recomendados para aplicar a 2FN são os seguintes:

- a) **Identificar** os atributos que não são funcionalmente dependentes de toda a chave primária;
- b) **Remover da entidade** todos esses atributos identificados e criar uma nova entidade com eles.

A chave primária da nova entidade será o atributo do qual os atributos do qual os atributos removidos são funcionalmente dependentes.

# Normalização de Banco de Dados

## Segunda Forma Normal (2FN)

N_pedido	Codigo_produto	Produto	Quant	Valor_unit	Subtotal
1005	1-934	Impressora laser	5	1.500,00	7.500,00
1006	1-956	Impressora desjet	3	350,00	1.050,00
1007	1-923	Impressora matricial	1	190,00	190,00
1008	1-908	Impressora mobile	6	980,00	5.880,00

Sem aplicar a normalização.

Aplicando apenas a 2FN.

Codigo_produto	Produto
1-934	Impressora laser
1-956	Impressora desjet
1-923	Impressora matricial
1-908	Impressora mobile

Tabela na segunda forma normal

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Tabela na 2ª forma normal

# Normalização de Banco de Dados

## Terceira Forma Normal (3FN)

Os procedimentos mais recomendados para aplicar a 3FN são os seguintes:

- a) **Identificar** todos os atributos que são funcionalmente dependentes de outros atributos não chave;
- b) **Removê-los.**

# Normalização de Banco de Dados

## Terceira Forma Normal (3FN)

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Sem aplicar a normalização.

Aplicando apenas a 3FN.

N_pedido	Codigo_produto	Quant	Valor_unit
1005	1-934	5	1.500,00
1006	1-956	3	350,00
1007	1-923	1	190,00
1008	1-908	6	980,00

*Tabela na terceira forma normal*