

# BANCO DE DADOS

Gratuliano Lucena



E-book 1

**FAM**  
ONLINE

# Neste E-Book:

<b>INTRODUÇÃO .....</b>	<b>3</b>
<b>INTRODUÇÃO A BANCOS DE DADOS.....</b>	<b>4</b>
<b>ESTRUTURA E UTILIZAÇÃO DE SGBD .....</b>	<b>5</b>
<b>INTRODUÇÃO A NORMALIZAÇÃO.....</b>	<b>11</b>
<b>INTRODUÇÃO A NOTAÇÃO GRÁFICA – ABORDAGEM O.O .....</b>	<b>14</b>
Autorrelacionamento .....	20
<b>MODELAGEM DE BANCO DE DADOS .....</b>	<b>21</b>
Técnicas de Criação de Índices .....	24
<b>NORMALIZAÇÃO .....</b>	<b>26</b>
1º Forma Normal (1FN) .....	26
2º Forma Normal (2FN) .....	27
3º Forma Normal (3FN) .....	27
<b>INTRODUÇÃO AO DIAGRAMA ENTIDADE RELACIONAMENTO – FERRAMENTA CASE ..</b>	<b>29</b>
<b>CONSIDERAÇÕES FINAIS .....</b>	<b>33</b>
<b>SÍNTESE .....</b>	<b>34</b>

# INTRODUÇÃO

Este módulo é o ponto de partida para entender o que é gerenciador de Bancos de Dados e quais são os principais gerenciadores que operam no mercado de Tecnologia da Informação. Serão explicados os conceitos que envolvem Bancos de dados relacionais.

Além disso, serão explicados os conceitos da engenharia de modelagem de dados, as técnicas de planejar os desenhos de bancos de dados, como é o relacionamento entre tabelas, chamado de normalização, que decompõe os bancos de dados em níveis, por exemplo, 1º, 2º e 3º, em formas normais, também esclarece o que é e como funciona a cardinalidade de dados, também o que é e como funciona a integridade referencial, o que é modelo de entidade relacionamento, o chamado DER, e fecha com os esclarecimentos sobre as técnicas e criação de índices.

Indo além, também é explicado como funciona um software livre para desenhar a modelagem de dados, envolvendo o Diagrama de Entidade e Relacionamento (DER).

# INTRODUÇÃO A BANCOS DE DADOS

O Banco de Dados surgiu da necessidade de guardar informações em meio eletrônico para substituir os arquivos em papéis e por ser mais rápido na recuperação dos dados.

Nos idos dos anos 1960, surgiram os primeiros bancos de dados. A IBM lançou um tipo de banco de dados para os computadores mainframe, chamado de IMS (Information Management System), um banco de dados em disco rígido, com acesso hierárquico para atender a NASA no projeto espacial Apolo.

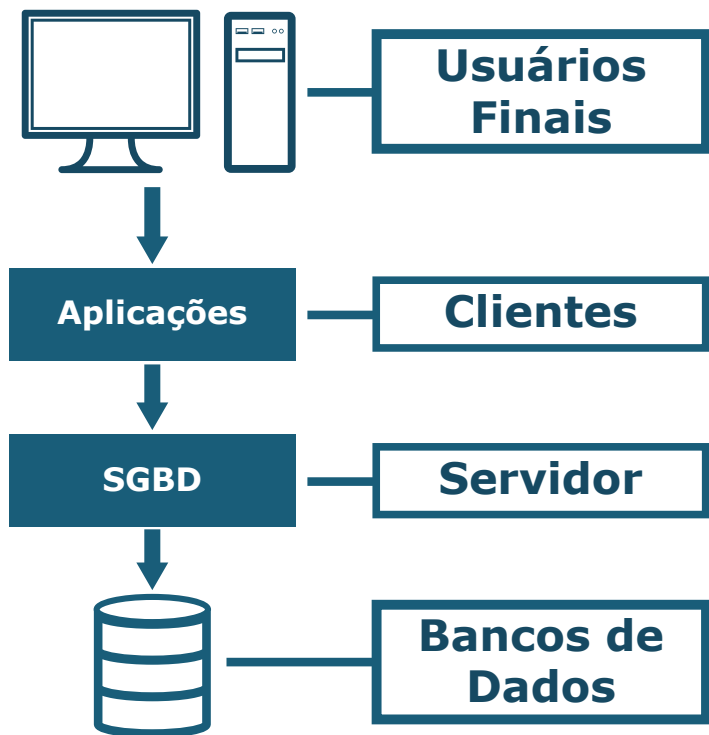
De lá para cá, surgiram vários tipos de armazenamentos de bancos de dados para microcomputadores, unidades de fitas, disquetes, etc.

Um funcionário da IBM, em 1970, o Sr. Edgar Frank Codd, idealizou e lançou o primeiro banco de dados relacional. Todos os bancos de dados relacionais trabalham com o padrão ANSI – SQL.

Existem vários SBDS, operando no mercado de Tecnologia da Informação, os mais destacados são: ORACLE, da ORACLE; DB2, da IBM; Sybase, da SAP; MYSQL, da ORACLE; POSTEGRE, patrocinados por um consórcio de patrocinadores: Redpill Linpro, PGX, EBC, etc.

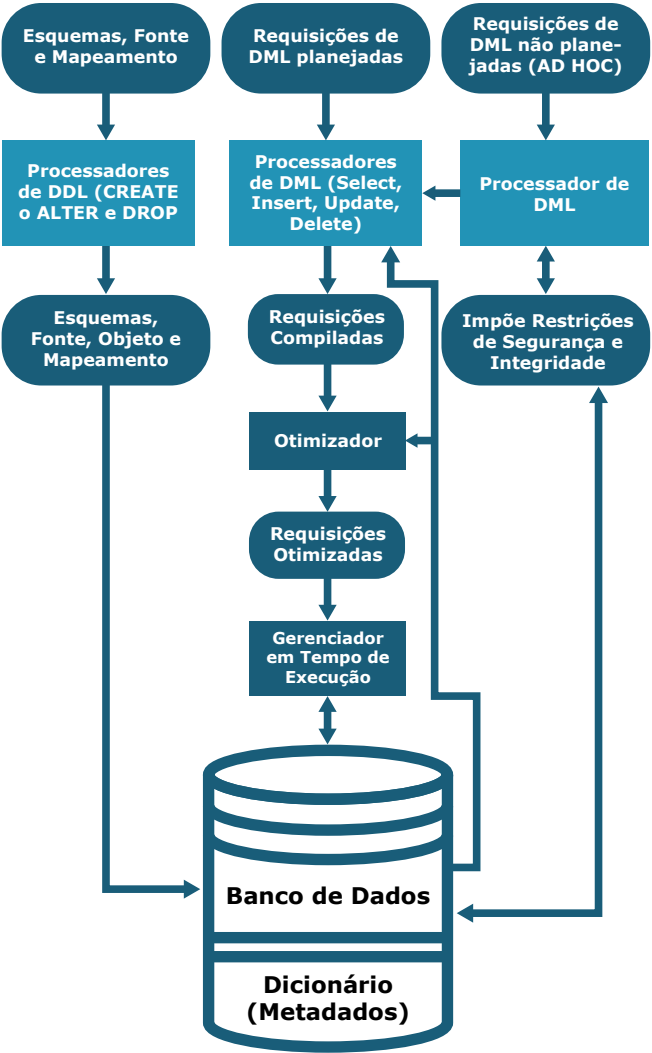
# ESTRUTURA E UTILIZAÇÃO DE SGBD

O Servidor é o próprio SGBD. Compreende definições de dados, manipulação de dados, segurança e integridade de dados, conforme demonstrado na figura 1.



**Figura 1:** Estrutura de interações Usuários x Bancos de Dados. **Fonte:** adaptado de DATE, 2000.

A figura 1 ilustra também que as aplicações estão relacionadas ao *Client* (Clientes) e o servidor (SGBD) exerce o papel de fazer a comunicação entre a aplicação e o banco de dados.



**Figura 2:** Fluxo operacional de um SGBD. **Fonte:** Adaptado de DATE, 2000.

Os bancos de dados possuem um gerenciamento complexo, conforme ilustrado na figura 2. Gerenciam a sintaxe de comandos DDL e DML, gerenciam a priorização de demandas, disponibilizam relatórios para acompanhar os indicadores de desempenho e de situações de erros. O SGBD exerce controle sobre paradas e retomadas de processamento, tem um controle de ponteiros de processamento preciso.

Os SGBDs também geram relatórios com indicadores gráficos em que são fáceis de entender alguma anormalidade ocorrida. Têm interface com planilhas eletrônicas para gerar as tabelas no formato de planilha, processam linguagem de alto nível com o SQL, interagem com todas as linguagens de programação do mercado: Java, VB, Cobol, .NET, Python, PHP, etc. Possuem mecanismos de *backups* automáticos e programados, possibilitam a “importação” e a “extração” de bancos de dados, não importando o volume de dados. Possuem rotinas para reorganização automática dos índices de todas as tabelas, possibilitam integração de banco de dados distribuídos, suportam altos volumes de dados com desempenho estável e confiável. Possibilitam integração com ferramentas case externas para modelagem de dados e possuem também ferramenta case nativa para modelagem de dados.

O Servidor é o próprio SGBD. Compreende definições de dados; manipulação de dados, segurança e integridade de dados. Os Clientes são aplicações executadas sobre o SGBD.

O SGBD concilia e viabiliza o pedido externo (usuário) e interno (SGBD) e a estrutura de armazenamento; executa operações sobre banco de dados armazenados, aceita definições de dados: esquemas externo, conceitual, interno e mapeamentos, e converte para objeto executáveis, inclui processadores ou compilações de DDL – linguagem de definição de dados –, denominados scripts de criação dos bancos de dados, processa ou compila DML (linguagem de manipulação de dados).

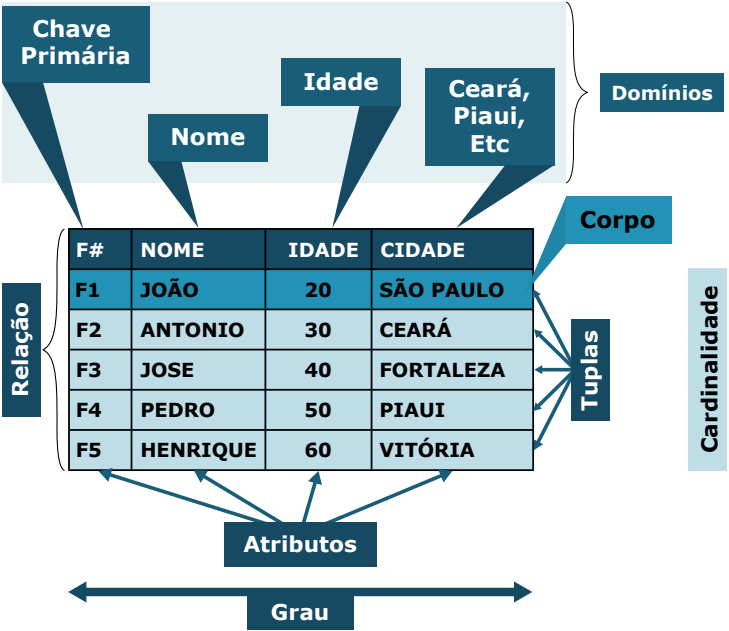
DML planejada é o estudo prévio da solicitação. O programador idealiza a função a ser executada em banco de dados, e simula em ferramentas de editores de Bancos de Dados ou Editores de Linguagens de Programação, e com a execução dos comandos de Bancos de Dados (SQL) verifica se o resultado atende a necessidade do programador.

DML não planejada, ad hoc, pode incorrer em inviabilidade na operação do banco de dados. São provenientes de execução de comandos por tentativa e erro, ocorre de forma direta e interativa, em ferramentas que possibilitam o acesso a banco de dados, por exemplo, editores de Banco de Dados, IDE (workbench) do ORACLE; IDE (workbench) do MYSQL; IDE (workbench) Postgree. Com isso, há um ganho de otimização do SGBD. Além disso, há como prevenir que situações que venham a comprometer a segurança e a integridade dos dados também restringem as regras de incorretas nos acessos para recuperação e concorrência dos dados.



O SGBD tem um dicionário de dados que compreende os nomes dos atributos (metadados), também guarda todas as informações de parâmetros dos bancos criados, nomes dos bancos, nomes das tabelas, nomes de índices, restrições de acessos.

O Modelo Relacional, demonstrado na figura 3, apresenta os elementos que envolvem a estrutura de dados, reflete quais são os componentes que compõem a integridade de dados e a manipulação de dados.



**Figura 3:** Estrutura de Dados de Banco de Dados. Fonte: DATE, 2000.

Cabe explicar os significados dos termos mostrados na figura 3:

**RELAÇÃO** = Corresponde à tabela como um todo; é o conjunto de atributos, tamanho dos atributos, índice da tabela, dados cadastrados, quantidade de linhas.

**TUPLA** = Corresponde a cada linha da tabela (registros), com dados cadastrados (populados).

**ATRIBUTO** = Corresponde às características de nome, tamanho de cada coluna da tabela (campos); os nomes servem de referência para os aplicativos obterem ou atualizarem os dados respectivos aos atributos.

**CARDINALIDADE** = Corresponde à quantidade de tuplas, ou seja, ao número de linhas cadastradas na tabela.

**PREDICADO** = Corresponde ao relacionamento de afinidade entre os atributos.

**CORPO** = Corresponde ao cabeçalho da tabela, que identifica cada atributo da tabela.

**DEADLOCK** é uma situação na qual duas ou mais transações disputam a atualização da mesma tupla, uma espera a outra e aguarda a liberação de desbloqueio.

Interromper um DEADLOCK envolve a escolha de uma transação ROLLBACK, liberando o bloqueio, permitindo assim o prosseguimento da transação que esteja aguardando.

**Acesse o Podcast 1 em Módulos**

# INTRODUÇÃO A NORMALIZAÇÃO

A técnica de normalização abrange uma análise dos dados brutos. Conforme mencionado na tabela 1, a primeira ação é separar os dados em conjuntos de dados com os mesmos objetivos e afinidades, e ainda idealizar um conjunto de dados sem redundância, conforme mostrado nas tabelas 2 e 3. Os dados redundantes são CPF CLIENTE e NOME CLIENTE, conforme ilustrado abaixo.

CONJUNTO DE DADOS DO CLIENTE		
CPF CLIENTE	NOME CLIENTE	ENDEREÇO CLIENTE
1	João Antonio	Avenida Paulo VI número 5
1	João Antonio	Avenida 9 de julho número 6

**Tabela 1:** Informações do cliente com dados redundantes  
Informações do cliente com dados redundantes.

No exemplo da tabela 1, você pode verificar que o CPF e o NOME CLIENTE estão duplicados, e na sequência o ENDEREÇO CLIENTE.

Devido a essa duplicidade, torna-se necessário aplicar uma técnica de normalização para eliminar a duplicidade. Conforme ilustrado na tabela 2 e tabela 3, o ideal é que os conjuntos de dados fiquem isentos de duplicidades. A duplicidade no jargão da modelagem de dados é considerada como redundância.

**PRIMEIRO CONJUNTO DE DADOS SOMENTE DO NOME CLIENTE**

CPF CLIENTE	NOME CLIENTE
1	João

**Tabela 2:** Conjunto de informações de Cliente.**SEGUNDO CONJUNTO DE DADOS SOMENTE DO ENDEREÇO DO CLIENTE**

CPF CLIENTE	TIPO ENDEREÇO CLIENTE	ENDEREÇO CLIENTE
1	Residencial	Avenida Paulo VI número 5
1	Comercial	Avenida 9 de julho número 6

**Tabela 3:** Conjunto de informações do Endereço do Cliente.

Com isso, haverá uma vantagem, pois na necessidade de se obter somente o nome do cliente podemos acessar somente uma vez. Com isso, diminui-se a quantidade de acessos, tendo um ganho de desempenho na velocidade do SGBD. Ainda se ganha ao evitar o armazenamento desnecessário, devido à duplicidade do nome do cliente, pois o nome do cliente é o campo em que, normalmente, há uma necessidade de tamanho de 30 a 40 caracteres. Teríamos uma economia de 30 caracteres.

Vamos entender algumas táticas para desenvolver a técnica de normalização, apoiando-se na modelagem orientada a objetos, que é aplicada tanto na própria metodologia orientada a objetos como também na metodologia ágil.

Há alguns tipos de relacionamentos entre conjuntos de dados diferentes. No exemplo demonstrado nas tabelas 2 e 3, temos de analisar qual é o tipo de relação entre os conjuntos de dados do NOME DO CLIENTE e o conjunto de dados do ENDEREÇO DO CLIENTE.

# INTRODUÇÃO A NOTAÇÃO GRÁFICA – ABORDAGEM O.O

É preciso entender algumas regras de associações entre tuplas (classes). Basicamente, temos as seguintes situações:

- Associação - é um relacionamento estrutural que especifica que objetos de um elemento estão conectados a objetos de outro elemento.

Notação gráfica com ou sem navegação:



- Agregação - relacionamento fraco do tipo “é parte de”. É um tipo especial de associação.

Notação gráfica:



- Composição - relacionamento forte do tipo “é parte de”. A composição entre um elemento (o “todo”) e outros elementos (“as partes”) indica que as partes só existem em função do “todo”.

Notação gráfica:



- Dependência - Uma dependência entre classes indica que os objetos de uma classe usam serviços dos objetos de outra classe.

Notação gráfica:



- Generalização (herança) - Relacionamento entre um elemento mais geral (chamado de superclasse ou pai) e um mais específico (chamado de subclasse ou filho).

Notação gráfica:



- Realização - relacionamento entre uma interface e o elemento que a implementa.

Notação gráfica:



Herança ou Dependências de relações entre tuplas (Classes)

Uma dependência indica que alguma mudança em um elemento (o “Nome do Cliente”) podem refletir em outro elemento (o “Endereço do Cliente”), com isso temos restrições de atualizações entre classes, conforme ilustrado na figura 4.

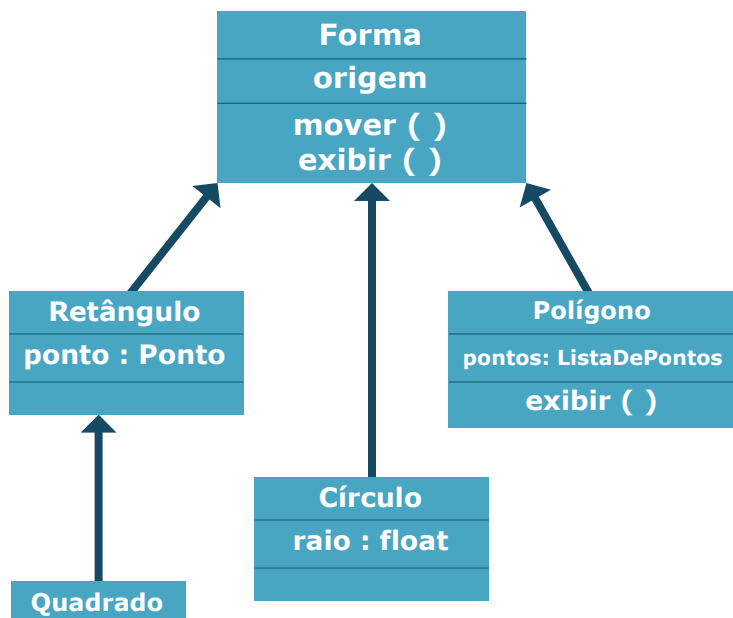
Uma dependência entre classes indica que os objetos de uma classe usam serviços dos objetos de outra classe.



**Figura 4:** Dependência entre classes. **Fonte:** Elaboração própria.

Relacionamento entre um elemento mais geral (chamado de superclasse ou pai) e um mais específico (chamado de subclasse ou filho), conforme ilustrado na figura 5. O gerenciar de banco de dados exerce um controle de restrição, por exemplo, no caso de exclusão. Primeiro há de se excluir os filhos, após isso é que é permitida a exclusão do pai. Esta restrição é chamada de *Constrain*.

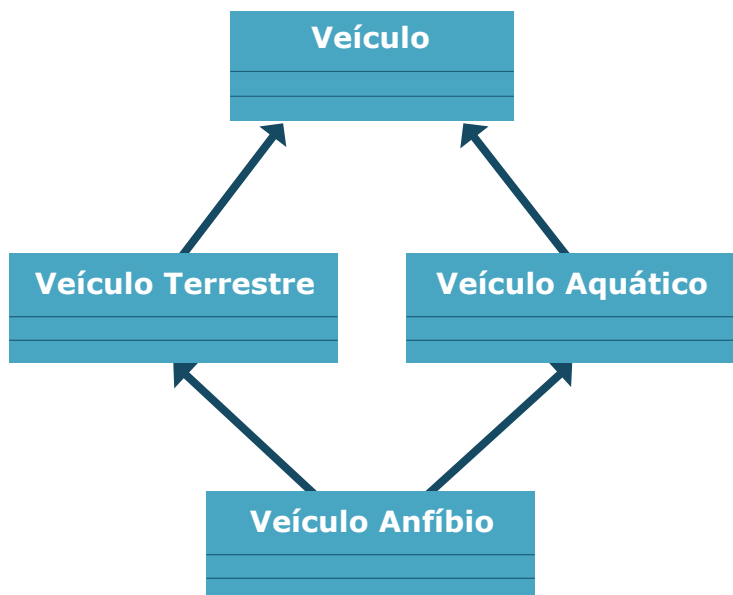




**Figura 5:** Primeiro exemplo de relacionamento de super-classe. **Fonte:** Elaboração própria.

A classe forma pode ter várias representações: um retângulo, um círculo ou um polígono, que herdam as informações da forma; e o quadrado herda as informações da forma e do retângulo. Com isso, por meio da herança, evita-se duplicar informações e funções para as classes que forem necessárias.

Outro exemplo de superclasses, conforme a figura 6: podemos ter veículos terrestre ou aquático, e podemos ter uma classe que reúne as informações e funções dos veículos aquático e terrestre; temos uma classe anfíbio.



**Figura 6:** Segundo exemplo de relacionamento de super-classe. **Fonte:** Elaboração própria.

A Cardinalidade diz respeito a uma técnica de como uma classe é dependente da outra em relação – considerando os conteúdos de cada classe.

Conforme ilustra a figura 7, pode-se explicar em várias representações gráficas, imaginando que tenhamos duas classes para comparar.

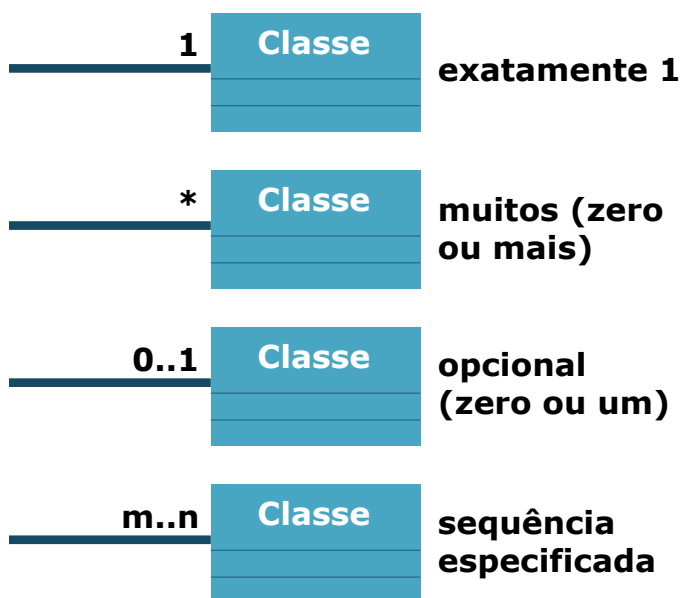
No primeiro caso, temos uma classe que corresponde a somente uma classe associada, com a indicação do número “1”. Exemplo: a classe CPF ocorre somente uma vez com a classe RG.

No segundo caso, a indicação de “\*” significa que temos; de um lado, uma classe com uma ocorrência; e do outro lado, uma classe que tem várias ocorrên-

cias. Exemplo: a classe CPF tem uma ocorrência e na classe endereço podemos ter mais de uma ocorrência: um endereço residencial, um endereço comercial e um endereço de contato.

No terceiro caso, a indicação de “1..0” significa que em uma classe com essa indicação podemos ter uma ocorrência ou ter uma classe sem nenhum dado.

No quarto caso, a indicação de “m..n” significa que podemos ter ocorrência fixa de intervalo, por exemplo 1..5; nesse caso, podemos ter uma linha (registro) e no máximo 5 linhas (registros).



**Figura 7:** Tipos de Cardinalidades. **Fonte:** Adaptado de KRUCHTEN, 2003.

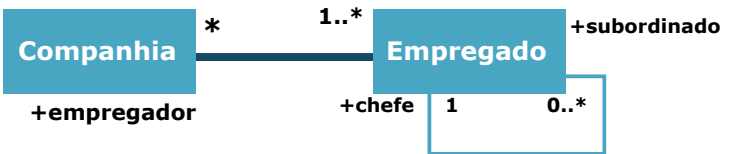
Quando há duas classes envolvidas na associação de forma direta, de uma para a outra (conforme figura 8), traduzindo a representação, significa que um número de Cliente pode ter nenhum pedido “0” ou ter vários números de pedidos “\*”, enquanto que um número de pedido só pode ser de um cliente.



**Figura 8:** Exemplo de relacionamento Cliente x Pedido.  
**Fonte:** Adaptado de KRUCHTEN, 2003.

## Autorrelacionamento

Há situações diferenciadas que são do autorrelacionamento, como mostrado na figura 9, em que o empregado exerce mais de um papel: ora é empregado ora é chefe.



**Figura 9:** Exemplo de relacionamento Companhia x Empregado.  
**Fonte:** Adaptado de KRUCHTEN, 2003.

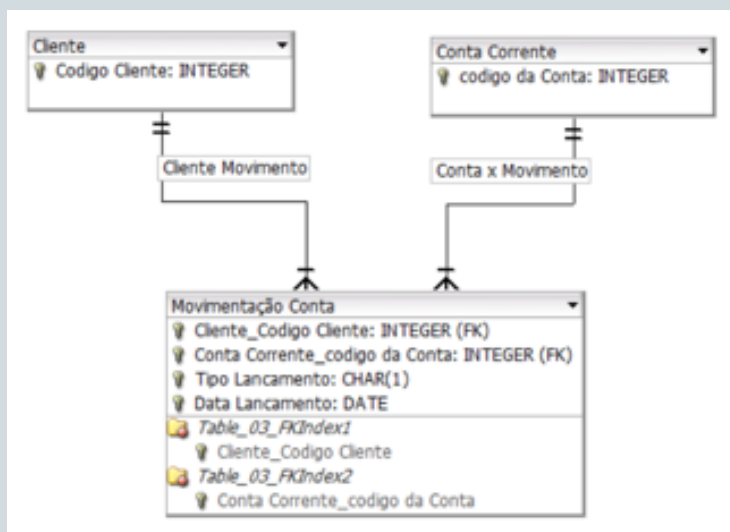
# MODELAGEM DE BANCO DE DADOS

O relacionamento considerado associativo (conforme figura 10) corresponde a criar uma nova classe, que é chamada de associativa. Compõe-se, basicamente, de chaves estrangeiras (*Foreign Key* – FK) mais os dados redundantes de chave própria da classe Movimentação da Conta (tipo de lançamento e data do lançamento), redundância, e ainda de mais um dado que não é chave, é o valor do lançamento. Nesse caso temos que um Cliente tem várias movimentações e que uma Conta Corrente tem várias movimentações.

Toda vez que há chave estrangeira (FK), existe uma restrição. Para excluir qualquer linha (registro) da tabela movimentação Conta, é preciso antes excluir a linha (registro) do Cliente e também, antes, deve-se excluir a linha (registro) da Conta Corrente. Se não for efetuado dessa forma, o gerenciador de Banco de Dados (SGBD) barra a exclusão da tabela Movimentação Conta.

A chave estrangeira possui esse nome porque a chave da Movimentação da Conta é proveniente do Código da classe Cliente (FK). A chave estrangeira Código da conta é proveniente da classe Conta Corrente. Observe que, para atribuir nome às informações do banco de dados, evitamos todo o tipo de acentuação, caracteres especiais.

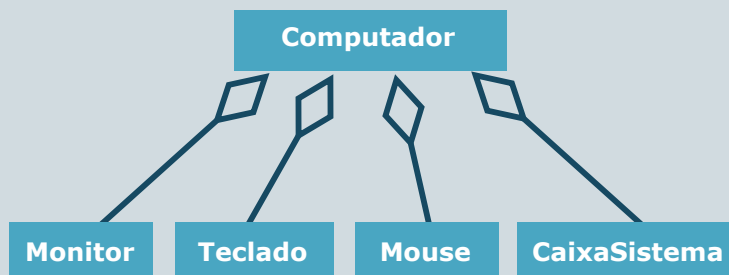
A tabela Movimentação Conta está na 3 FN. Como pode ser observado a seguir, normalmente se considera esta tabela como dados históricos. Considera-se esta tabela com uma cardinalidade de N (\*) para N (\*).



**Figura 10:** <https://db-designer-fork.soft112.com> Exemplo de relacionamento Associativo. **Fonte:** Elaboração própria.

Uma forma especial de associação entre o todo e suas partes, na qual o todo é composto de partes, é a agregação, conforme demonstrado na figura 11.

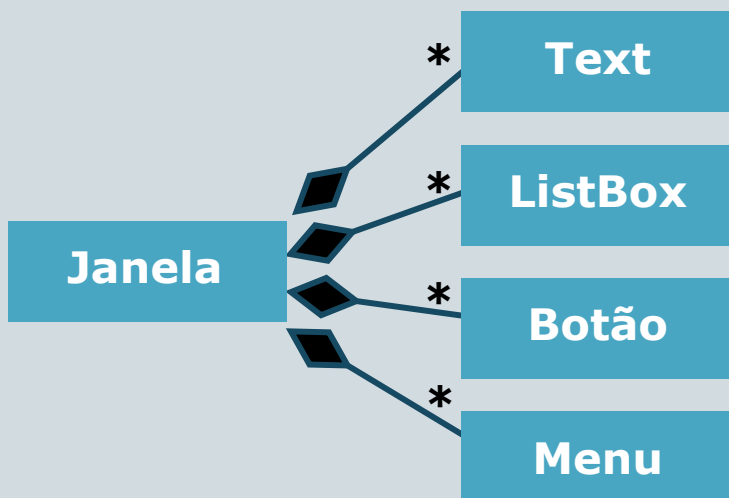
Não impõe que a vida das “Partes” esteja relacionada com a vida do “Todo”, ou seja, se decidirmos excluir algum elemento, isso não interfere no funcionamento do todo, que é o computador.



**Figura 11:** Exemplo de Agregação. **Fonte:** Adaptado de KRUCHTEN, 2003.

A composição, conforme ilustrado na figura 12, é parecida com a agregação, e se diferencia nos seguintes aspectos:

Todos os elementos compõem o todo, e a falta de um só elemento inviabiliza o todo.



**Figura 12:** Exemplo de Agregação. **Fonte:** Adaptado de KRUCHTEN, 2003.

## Técnicas de Criação de Índices

O acesso por índice influi para o SGBD operar com mais rapidez para buscar informações em uma tabela (arquivo).


Para a informação servir de índice, tem de ser única; deve ter somente uma ocorrência em toda a tabela.

 CPF	Nome	Endereço
1	João	Rua X
2	Maria	Rua Y

A informação de índice deve ser do tipo numérica; raramente será do tipo alfanumérica. O índice alfanumérico demora mais para o SGBD converter byte a byte para o binário, que é o formato utilizado pelo SGBD para acessar a tabela. O campo textual normalmente tem tamanho não recomendável.

O índice pode envolver somente uma informação, considerado chave simples; ou pode envolver mais de uma informação, considerado chave composta.

Quando não é possível ter uma informação como chave única na tabela, precisamos elencar mais uma informação para termos outra informação como chave única.

 CPF	Tipo Endereço	Endereço
1	Residencial	Rua X
1	Comercial	Rua Y



Todas as informações de índices devem estar nos primeiros campos da tabela, jamais devem ficar no meio ou no fim da tabela.

As informações para índices não devem ser nulas.

# NORMALIZAÇÃO


O relacionamento entre as classes em que há uma forma de dependência tem uma técnica que faz parte da normalização e ajuda no entendimento nas análises das dependências.


## 1º Forma Normal (1FN)

Cada informação deve ter atributos atômicos; só deve permitir atributos atômicos (simples, indivisíveis).

Essa forma contém anomalias: o telefone da agência é multivalorado, pois podemos ter vários números de telefones. Isso dificulta para incluir, alterar ou excluir um telefone da agência.

**CONTRATO** {id\_contrato, id\_pacote, preco\_pacote, cnpj\_agencia, nome\_agencia, telefone\_agencia, rg\_cliente, nome\_cliente }

**CONTRATO** { id\_contrato, id\_pacote, preco\_pacote, #id\_agencia, rg\_cliente, nome\_cliente }

**AGENCIA** { Id\_agencia, cnpj\_agencia, nome\_agencia, telefone\_agencia }


### Dados sem normalização


Solução 1FN: devemos separar os dados do telefone da agência, criando uma nova relação chamada AGENCIA, criando um dado artificial como chave única = id\_agencia.

## 2º Forma Normal (2FN)

Uma tabela está na 2FN se, e somente se, ela estiver na 1FN e não possuir Dependência Funcional Parcial (DF). Nessa forma, os atributos que não são chaves devem conter informações que se referem à chave inteira, e não somente a parte do registro. Nesta 2FN, o registro CONTRATO foi dividido em duas novas relações, incluindo informações separadas sobre a agência e sobre o cliente.

Nesta 2FN ainda podem ocorrer anomalias de atualização e exclusão, por exemplo, em alterações de dados sobre o pacote.

CONTRATO {  id\_contrato, id\_pacote, preco\_pacote, #id\_agencia, rg\_cliente, }

AGENCIA {  Id\_agencia, cnpj\_agencia, nome\_agencia, telefone\_agencia }


CLIENTE {  rg-cliente, nome\_cliente }


## 3º Forma Normal (3FN)

Uma tabela está na 3FN se, e somente se, ela estiver na 2FN e não possuir DFs indiretas. Tabelas com Dependências Funcionais indiretas devem ser desmembradas em tabelas que não possuem tais dependências funcionais.

Esta 3FN refere-se ao agrupamento de relações requeridas na 2FN e, para resolver a anomalia da 2FN, com relação a pacote, criamos mais uma relação

chamada PACOTE. Quando se chega neste nível, podemos afirmar que não há mais dados redundantes e os índices criados ajudam no bom desempenho de acesso às tabelas. Com isso, o banco de dados se torna flexível para suportar qualquer alteração no “*front end*” (Telas de entradas e saídas de dados). Quando chegamos à terceira forma normal, eliminamos a redundância de atributos. Por exemplo, para todos os contratos teríamos duplicado o nome do cliente. Além disso, temos uma distribuição das informações de forma organizada, com a criação de atributos chaves de forma coerente e, quando precisamos acessar somente dados do cliente, fica mais simples; não há dados desnecessários – da agência, por exemplo.

**CONTRATO** {  id\_contrato, id\_pacote, #id\_agencia, rg\_cliente, }

**AGENCIA** {  Id\_agencia, cnpj\_agencia, nome\_agencia, telefone\_agencia }

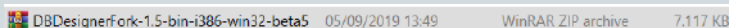
**CLIENTE** {  rg-cliente, nome\_cliente }

**PACOTE** {  id\_pacote, preco\_pacote }

# INTRODUÇÃO AO DIAGRAMA ENTIDADE RELACIONAMENTO – FERRAMENTA CASE

A ferramenta Case serve para desenhar um Banco de Dados lógico. Recomendamos utilizar o software livre da DBDesignerfork, que é uma ferramenta amigável e de facilidade visual para o Windows. É uma ferramenta IDE (Ambiente de Desenvolvimento Integrado); não é necessário instalar, pode ser executado como um software interpretado, pode ser baixado, conforme ilustrado figura 13, no seguinte link: <https://sourceforge.net/projects/dbdesigner-fork/>

O software é baixado no formato zipado:



**Figura 13:** Arquivo zipado do software DBDesignerFork.  
**Fonte:** DBDesignerfork.

Após baixado, extrair para um diretório de sua preferência.

Para executar o software, dar duplo clique no arquivo DBDesignerfork.exe, conforme ilustrado na figura 14:

-			Pasta de arquivos		
Data			Pasta de arquivos	13/03/2010 09...	
Doc			Pasta de arquivos	13/03/2010 09...	
Examples			Pasta de arquivos	13/03/2010 09...	
Gfx			Pasta de arquivos	13/03/2010 09...	
Copying Addition for Win32.txt	1.495	730	Documento de Tex...	19/09/2006 10...	BBAC2164
Copying.txt	15.421	5.725	Documento de Tex...	19/09/2006 10...	5B7D9D6E
DBDesignerfork.exe	3.872.768	1.497.338	Aplicativo	05/04/2010 02...	4043E871
DBDplugin_DataImporter.exe	1.357.824	607.254	Aplicativo	19/09/2006 01...	30A0ED84
DBDplugin_HTMLReport.exe	1.263.104	548.588	Aplicativo	19/09/2006 01...	7D787B6C
DBDplugin_SimpleWebFront.exe	1.570.816	663.600	Aplicativo	19/09/2006 01...	416FFAF4
dbexpms.dll	128.000	54.241	Extensão de aplicat...	30/05/2005 13...	B23D7DEE
dbexpmysql.dll	95.744	45.897	Extensão de aplicat...	30/05/2005 13...	76FB7E22
dbexpora.dll	171.008	72.155	Extensão de aplicat...	30/05/2005 13...	832F29EB
dbxoodbc.dll	266.752	129.251	Extensão de aplicat...	30/05/2005 13...	A5670B62
dbxSQLite.dll	179.712	91.030	Extensão de aplicat...	30/05/2005 13...	83C0578C
libmysql.dll	233.472	130.056	Extensão de aplicat...	30/05/2005 13...	4C61D81A
qtintf70.dll	4.082.688	1.685.584	Extensão de aplicat...	29/12/2005 15...	D18BE450
sqlite.dll	220.672	109.850	Extensão de aplicat...	30/05/2005 13...	7AACCB5B

**Figura 14:** Arquivo baixado da internet do software DBDesignerFork. **Fonte:** DBDesignerfork.

Utilizando-se a ferramenta Case, elaboramos um modelo lógico de banco de dados, conforme ilustrado na figura 15. Esta ferramenta pode ser utilizada por tempo indeterminado, sem a necessidade de licença.

Neste modelo lógico, as chaves estrangeiras (Foreign Keys – FK) foram geradas automaticamente pela ferramenta Case.

A inclusão da entidade CONTRATO possibilitou substituir a relação muitos para muitos (N:N) entre as entidades PACOTE e CLIENTE.

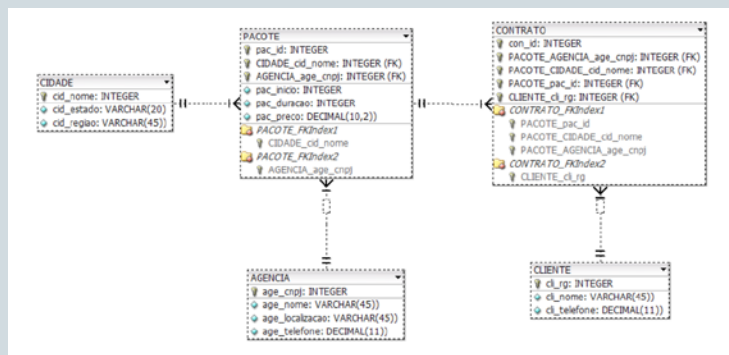
A tabela CIDADE tem uma relação de 1:N com a tabela PACOTE.

A Tabela AGENCIA tem uma relação de 1:N com a tabela PACOTE.

A Tabela PACOTE tem uma relação de 1:N com a tabela CONTRATO.

A tabela CLIENTE tem uma relação de 1:N com a tabela CONTRATO.

A tabela CONTRATO tem duas relações de N's chegando, portanto, pode-se considerar uma relação de N:N.



**Figura 15:** Exemplo de modelagem de dados gerado pela ferramenta Case. **Fonte:** Elaboração própria.

## SAIBA MAIS

É importante conhecer alguns sites de gerenciadores de bancos de dados. É possível encontrar downloads, documentação, tutoriais, entre outros, que facilitarão sua vida para um conhecimento mais aprofundado em bancos de dados:

MYSQL - <https://www.mysql.com>

ORACLE - <https://www.oracle.com/database/technologies/112010-win64soft.html>

POSTGRE - [https://www.elephantsql.com/?gclid=Cj0KCQjwh8jrBRDQARIsAH7BsXeUf-y9SAPYw\\_ers-](https://www.elephantsql.com/?gclid=Cj0KCQjwh8jrBRDQARIsAH7BsXeUf-y9SAPYw_ers-)

TuEF7eVTlwrCYNqgDVzidnI1KMQuhbAxFJ-ViAa-AtIREALw\_wcB

Editor de Banco de Dados (IDE) do MYSQL - <https://dev.mysql.com/downloads/workbench>

Editor de Banco de Dados (IDE) do POSTEGRE - <https://www.postgresql.org/download>



# CONSIDERAÇÕES FINAIS

O sistema gerenciador de Banco de Dados é a uma referência central que alicerça a estrutura da tecnologia da informação, pois é o meio de armazenagem de todo tipo de informações, cadastrais, históricas, científicas, comunicação, cálculos, etc. Um ponto a se destacar é o acesso das informações no momento que se deseja.

Para atender o sistema gerenciador de Banco de Dados (SGBD), contamos com várias opções no mercado: ORACLE, MYSQL, POSTEGRE, SYBASE, DB2.

O que estudamos aqui são os conceitos básicos sobre a estrutura de funcionamento de um Gerenciador de Banco de Dados (SGBD). Também foram explicados os conceitos básicos sobre a modelagem de Banco de Dados; comentamos sobre alguns termos utilizados, por exemplo, cardinalidade e relacionamento.

Apresentamos uma introdução básica da ferramenta Case, onde se desenha a modelagem de dados de forma gráfica, chamada de Diagrama de Entidade e Relacionamento – que também é chamada de Modelo de Entidade e Relacionamento.

# SÍNTESE



## Banco de Dados

Entendemos o que é gerenciador de Bancos de Dados e quais os principais gerenciadores que operam no mercado de Tecnologia da Informação, bem como os conceitos que compreendem o que envolve Bancos de dados relacionais.

- Conceitos da engenharia de modelagem de dados.
- Técnicas de planejar os desenhos de bancos de dados.
- Relacionamento entre tabelas, que compõem os bancos de dados em níveis, em formas normais.
- Cardinalidade de dados.
- Integridade referencial.
- Modelo de entidade relacionamento – o chamado DER.
- Técnicas e criação de índices.
- Software - Diagrama de Entidade e Relacionamento (DER).
- Case – Diagrama/Modelo de Entidade e Relacionamento.

# Referências Bibliográficas & Consultadas

---

ASCENCIO, A. F. G.; ARAÚJO, G. S. **Estruturas de dados**: algoritmos, análise da complexidade e implementações em JAVA e C/C++. São Paulo: Pearson Prentice Hall, 2010. [Biblioteca Virtual]

BARBOZA, F. F. M. *et al.* **Modelagem e desenvolvimento de banco de dados**. Porto Alegre: Sagah, 2018. [Minha Biblioteca]

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. São Paulo: Campus, 2000.

CALSARA, A.; MACHADO, C. A. F.; REINEHR, S. S.; BURNETT, R. C. **Aderência do RUP à norma NBR ISO/IEC 12207**. Dezembro/2002. Disponível em: <https://docplayer.com.br/18795196-Aderencia-do-rup-a-norma-nbr-iso-iec-12207.html>. Acesso em: 03 out. 2019.

DATE, C. J. **Introdução a sistemas de bancos de dados**. 7. ed. Rio Janeiro: Campus, 2000.

DBDesignerfork. **Software livre para modelagem de Dados**. Disponível em: <https://db-designer-fork.soft112.com>. Acesso em: 05 set. 2019

ELMASRI, R.; NAVATHE, S. **Sistemas de banco de dados**. 4. ed. São Paulo: Pearson, 2005. [Biblioteca Virtual]

GANE, C. **Análise estruturada de sistemas**. 7. ed. Rio Janeiro: LTC, 1993.

HAY, D. C. **Princípios de modelagem de dados**. São Paulo: Makron, 1999.

HEUSER, C. A. **Projeto de banco de dados**. 6.ed. Porto Alegre: Bookman, 2009. [Minha Biblioteca]

KRUCHTEN, P. **Introdução ao RUP Rational Unified Process**. 2. ed. Rio de Janeiro: Ciência Moderna, 2003.

MEDEIROS, L. F. **Banco de dados: princípios e prática**. Curitiba: InterSaberes, 2013. [Biblioteca Virtual]

PRESSMAN, R. S. **Engenharia de software**. 5. ed. São Paulo: Makron Books, 2002.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados: implementação em SQL, PL/SQL e Oracle** 11g. São Paulo: Pearson Education do Brasil, 2013. [Biblioteca Virtual]

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados**. 3. ed. Porto Alegre: AMGH, 2011. [Minha Biblioteca]

RESENDE, D. A. **Engenharia de software e sistemas de informação**. 2. ed. Rio Janeiro: Brasport, 2003.

SETZER, V. W. **Banco de dados**. 3. ed. Rio Janeiro: Edgard Blücher, 1989.

SOMMERVILLE, I. **Engenharia de software**. 6. ed. São Paulo: Pearson, 1995.

TELES, V. M. **Extreme Programming**. São Paulo: Novatec, 2004.

WOLFGANG, P. A. T.; KOFFMAN, E. B. **Objetos, abstração, estruturas de dados e projeto usando C++**. Rio de Janeiro: LTC, 2008. [Minha Biblioteca]

**FaM**  
**ONLINE**