

# Banco de Dados de Alta Performance

1ª  
Edição

# **BANCO DE DADOS**

**Autoria:**  
**Geraldo Barbosa do Amarante**

## EXPEDIENTE

<b>REITOR:</b>	• <b>FICHA TÉCNICA</b>
PROF. CLÁUDIO FERREIRA BASTOS	• <b>AUTORIA:</b>
<b>PRÓ-REITOR ADMINISTRATIVO FINANCEIRO:</b>	• GERALDO BARBOSA DO AMARANTE
PROF. RAFAEL RABELO BASTOS	• <b>SUPERVISÃO DE PRODUÇÃO EAD:</b>
<b>PRÓ-REITOR DE RELAÇÕES INSTITUCIONAIS:</b>	• FRANCISCO CLEUSON DO NASCIMENTO ALVES
PROF. CLÁUDIO RABELO BASTOS	• <b>DESIGN INSTRUCIONAL:</b>
<b>PRÓ-REITOR ACADÊMICO:</b>	• ANTONIO CARLOS VIEIRA
PROF. HERBERT GOMES MARTINS	• <b>PROJETO GRÁFICO E CAPA:</b>
<b>DIREÇÃO EAD:</b>	• FRANCISCO ERBÍNIO ALVES RODRIGUES
PROF. RICARDO ZAMBRANO JÚNIOR	• <b>DIAGRAMAÇÃO E TRATAMENTO DE IMAGENS:</b>
<b>COORDENAÇÃO EAD:</b>	• ISMAEL RAMOS MARTINS
PROFA. LUCIANA RODRIGUES RAMOS	• <b>REVISÃO TEXTUAL:</b>
	• ANTONIO CARLOS VIEIRA

### FICHA CATALOGRÁFICA CATALOGAÇÃO NA PUBLICAÇÃO BIBLIOTECA CENTRO UNIVERSITÁRIO ATENEU

AMARANTE, Geraldo Barbosa do. Banco de dados. Geraldo Barbosa do Amarante. – Fortaleza: Centro Universitário Ateneu, 2020.

172 p.

ISBN: 978-65-88268-36-0

1. Sistemas Gerenciadores de Banco de Dados (SGBD). 2. Linguagem SQL. 3. Técnicas avançadas de dados. 4. Bancos de dados em alta disponibilidade e alta performance. Centro Universitário Ateneu. II. Título.

Todos os direitos reservados. Nenhuma parte desta publicação pode ser reproduzida, total ou parcialmente, por quaisquer métodos ou processos, sejam eles eletrônicos, mecânicos, de cópia fotostática ou outros, sem a autorização escrita do possuidor da propriedade literária. Os pedidos para tal autorização, especificando a extensão do que se deseja reproduzir e o seu objetivo, deverão ser dirigidos à Reitoria.



# SEJA BEM-VINDO!

---

Caro estudante, este é o material da disciplina *Banco de dados*, que pretende ser o guia para o estudo e entendimento da tecnologia de banco de dados, tecnologia esta que é a base para todo e qualquer sistema computadorizado. De nada adiantaria sistemas sofisticados de software se não tivéssemos onde guardar as informações produzidas por eles, de uma forma segura e disponível.

Podemos dizer que o banco de dados é a “alma da empresa”, não importando o seu porte no mundo dos negócios, seja uma empresa gigantesca, como a Amazon ou a Google, ou uma microempresa familiar, a questão da tomada de decisões baseada em dados independe do seu tamanho, é vital para a sua sobrevivência e crescimento no mercado.

A revista *The Economist*, em 2017, trouxe uma frase que transmite bem o quanto é importante gerir os dados da empresa, a frase dizia que “*o recurso mais valioso do mundo não é o petróleo, mas sim os dados*”. Isto significa que é crucial as empresas manterem as informações de qualidade sobre os seus clientes, fornecedores, suas transações contábeis, processos etc. de forma segura e disponível.

Este livro está dividido em quatro unidades. A primeira unidade tem o objetivo de dar uma visão geral dos sistemas gerenciadores de banco de dados (SGBD), fornecendo uma base conceitual. Também discutiremos o modelo relacional que é mais amplamente utilizado na maioria dos sistemas gerenciadores de banco de dados comerciais. Veremos também tópicos relacionados a um projeto de um pequeno banco de dados a ser implementado nas unidades subsequentes. Na segunda unidade, aprenderemos a instalar e configurar o SGBD MySQL, assim como a ferramenta MySQL Workbench para que seja possível desenvolver uma aplicação prática dos conceitos aprendidos utilizando a linguagem SQL. Na terceira unidade, aprofundaremos mais nas características e recursos do SGBD MySQL, para aprendermos a tornar os nossos bancos de dados mais eficientes e com alto desempenho. Na quarta e última unidade, discutiremos assuntos relacionados à distribuição de banco de dados, seu monitoramento e como melhorar o desempenho desta estrutura.

Bons estudos!

Estes ícones aparecerão em sua trilha de aprendizagem e significam:



### **ANOTAÇÕES**

Espaço para anotar suas ideias.



### **MATERIAL COMPLEMENTAR**

Texto ou mídias complementares ao assunto da aula.



### **CONECTE-SE**

Convocar o estudante para interagir no fórum tira-dúvidas.



### **MEMORIZE**

Tópico ou fato importante de lembrar.



### **CURIOSIDADE**

Informação curiosa relacionada ao conteúdo.



### **OBJETIVOS DE APRENDIZAGEM**

Objetivos de estudo do capítulo ou unidade.



### **EXERCÍCIO RESOLVIDO**

Atividade explicativa para guiar o estudante.



### **PRATIQUE**

Exercícios para fixar os conteúdos.



### **FIQUE ATENTO**

Informação complementar ao texto principal.



### **REFERÊNCIAS**

Fontes de pesquisa citadas no texto.



### **LINK WEB**

Indicação de sites.



### **RELEMBRE**

Resumo do conteúdo estudado.



# SUMÁRIO

## 01

### SISTEMAS GERENCIADORES DE BANCO DE DADOS (SGBD)

1. Visão geral sobre sistemas gerenciadores de banco de dados .....	8
2. Tipos de banco de dados .....	14
3. O modelo relacional.....	14
4. Projeto de banco de dados - normalização .....	20
5. As formas normais mais utilizadas .....	23
6. O dicionário de dados.....	32
7. Aplicando a normalização .....	37
8. O diagrama de entidades e relacionamentos.....	42
Referências .....	47

### O SISTEMA GERENCIADOR DE BANCO DE DADOS MYSQL E A LINGUAGEM SQL

1. O sistema gerenciador de banco de dados MySQL.....	50
1.1. Instalação do MySQL .....	51
1.2. Configuração .....	53
2. A linguagem SQL ( <i>structure query language</i> ) .....	59
2.1. Utilizando o MySQL para a criação e manipulação de banco de dados .....	61
2.2. Criação das tabelas do banco de dados .....	65
2.3. Criação das chaves primárias .....	68
2.4. Implementando a propriedade de AUTO INCREMENTO .....	69
2.5. Implementando os relacionamentos entre as tabelas .....	71
2.6. Manipulação dos dados.....	75
3. Tabelas temporárias.....	84
3.1. Vantagens das tabelas temporárias .....	85
3.2. Criação de tabelas temporárias.....	85
3.3. Criação de tabelas temporárias com informações vindas de um comando <i>SELECT</i> .....	86
3.4. Exclusão das tabelas temporárias.....	86
Referências .....	90

## 02

# 03

## TÉCNICAS AVANÇADAS DE DADOS

1. Particionamento.....	94
1.1. Particionamento tipo <i>RANGE</i> .....	95
1.2. Particionamento tipo <i>LIST</i> .....	97
1.3. Particionamento tipo <i>COLUMN</i> .....	98
1.4. Particionamento do tipo <i>HASH</i> .....	102
1.5. Particionamento do tipo <i>KEY</i> .....	103
2. Indexação.....	104
2.1. Classificação dos índices conforme sua estrutura.....	106
2.2. Criação de índices no MySQL.....	106
3. Objetos avançados de banco de dados.....	107
3.1. Visões ( <i>Views</i> ).....	107
3.2. Funções ( <i>Functions</i> ).....	112
3.3. Gatilhos ( <i>Triggers</i> ).....	116
4. Procedimentos armazenados ( <i>stored procedures</i> ).....	121
Referências.....	129

## BANCOS DE DADOS EM ALTA DISPONIBILIDADE E ALTA PERFORMANCE

1. A alta disponibilidade e a alta performance.....	132
2. A replicação.....	133
2.1. Teoria da replicação.....	135
2.2. Replicação no MySQL.....	145
3. <i>Cluster</i> .....	147
4. Monitoramento.....	151
4.1. Descrição dos processos de monitoramento e suporte.....	153
4.2. Ferramentas de monitoramento no MySQL.....	161
5. Técnicas e dicas para alta performance de banco de dados.....	166
Referências.....	171

# 04



Geraldo Barbosa do Amarante

## Unidade

# 01

## SISTEMAS GERENCIADORES DE BANCO DE DADOS (SGBD)

### Apresentação

Esta unidade tem como objetivo apresentar a teoria relevante associada aos conceitos de sistemas gerenciadores de banco de dados relacionais, às técnicas mais utilizadas em projetos de banco de dados e às ferramentas de documentação, como o dicionário de dados e o diagrama de entidades e relacionamentos.

A importância de entender estes conceitos está associada ao desenvolvimento da capacidade de executar um bom planejamento dos bancos de dados, implementando de maneira correta e permitindo um alto desempenho nas tarefas de busca de informações nele armazenadas.





## OBJETIVOS DE APRENDIZAGEM

- *Promover a efetiva compreensão dos fundamentos ligados à tecnologia de bancos de dados relacionais e introduzir o estudante no universo do projeto de banco de dados;*
- *Analisar os conceitos apresentados nesta disciplina, de modo que seja possível aplicar nas unidades subsequentes, quando da utilização de uma ferramenta de banco de dados;*
- *Compreender como os tópicos teóricos abordados nesta unidade serão utilizados em um trabalho prático de construção de um banco de dados.*

## 1. VISÃO GERAL SOBRE SISTEMAS GERENCIADORES DE BANCO DE DADOS

Com a evolução das tecnologias da informação e da comunicação, muitos **serviços** estão sendo oferecidos utilizando a rede mundial de computadores, ou como popularmente conhecida, a **internet**, através da utilização dos computadores ou dos diversos dispositivos móveis. Hoje, temos à nossa disposição ferramentas de software como as disponibilizadas pelas empresas Google e Amazon, as redes sociais, entre muitos outros, além de um grande número de aplicativos para celulares que facilitam a vidas das pessoas, permitindo, por exemplo, a criação dos mais variados serviços de entrega em que, através de aplicativos, os pedidos são feitos e o pagamento já é automaticamente efetuado por cartões de crédito.

“...um grande número de aplicativos para celulares que facilitam a vidas das pessoas...”

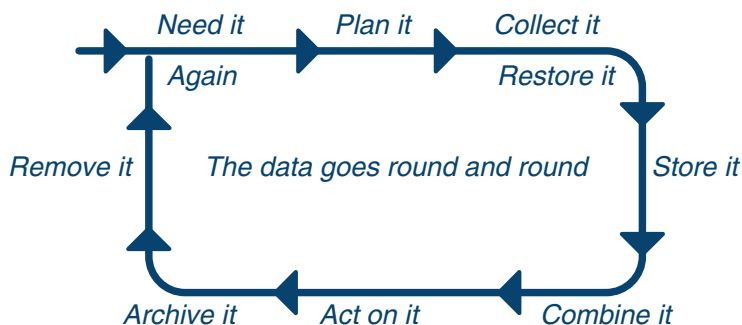
Além de toda esta tecnologia que coloca à nossa disposição dispositivos eletrônicos modernos que nos permite o acesso à rede mundial (internet), temos um componente não palpável, mas fundamental para a vida moderna, que são os **dados**. De nada adiantaria toda a tecnologia se

não fosse possível guardá-los, modificá-los e recuperá-los e com o devido processamento transformá-los em **informações** úteis, que por sua vez, sendo interpretados, irão gerar valiosos **conhecimentos** que serão a base para as tomadas de decisões estratégicas, garantindo a sobrevivência do negócio. Deste conhecimento devidamente analisado virá a sabedoria sobre o negócio da empresa e sobre o ambiente no qual ela está inserida, permitindo previsões futuras, através da utilização das técnicas da **análise preditiva** e da utilização das aplicações de **inteligência artificial**, sempre levando em conta a massa de dados que se tem à disposição.

Para que sejam úteis, os dados devem ser guardados, mantendo algumas exigências básicas, dentre elas a **confiabilidade**, que diz respeito à manutenção da privacidade dos dados, não permitindo que estas informações sejam roubadas ou modificadas. Outra exigência é a **integridade**, que diz respeito a manter os dados protegidos contra qualquer ação que os deixem imprecisos, corrompidos ou danificados, mantendo, assim, a confiabilidade deles. Pode-se citar ainda a **disponibilidade**, de nada adianta podermos gravar os dados e não termos como permitir a acessibilidade a eles, pelas pessoas ou sistemas autorizados.

Podemos afirmar que os dados têm um **ciclo de vida**, para ilustrar, utilizaremos a figura retirada do livro *Beginning Relational Data Modeling*, cujos autores são Sharon Allen e Evan Terry. Podemos notar que este ciclo tem várias **fases** que acontecem obedecendo a uma sequência, como demonstrado na figura a seguir.

**Figura 01:** Ciclo de vida dos dados.



Fonte: ALLEN; TERRY, 2005.

No quadro a seguir, o ciclo de vida dos dados é explicado.

**Quadro 01:** Descrição do ciclo de vida dos dados.

<p><b>"NEED IT"</b> Surge a necessidade dos dados.</p>	<p>Nesta fase, a necessidade surge quando alguém precisa tomar uma decisão ou resolver um problema e sente a necessidade de ter informações que forneçam uma base de conhecimento que a auxiliem na tomada de decisão ou resolução do problema. Como exemplo, temos uma empresa que comercializa produtos alimentícios e necessita de algumas informações que servirão de base para as suas campanhas de marketing. Então, algumas perguntas precisam ser respondidas, entre elas, as listadas abaixo:</p> <ul style="list-style-type: none"><li>• Dos produtos que comercializo, quais são aqueles responsáveis por 80 % do meu faturamento?</li><li>• Dos produtos que comercializo, quais são aqueles cuja venda mensal não representa 20% do meu faturamento?</li></ul>
<p><b>"PLAN IT"</b> Planejar como e onde conseguir os dados.</p>	<p>Após as necessidades dos dados devidamente identificadas, passaremos para a fase de pesquisarmos onde conseguir os dados necessários. Então, precisamos verificar se temos sistemas na empresa que já nos forneçam os dados, ou se teremos de buscá-los em outras fontes. Aqui, entra a análise dos elementos de dados, com o objetivo de identificar algumas características peculiares, tais como: a frequência com que ocorrem, seu tamanho, as regras de negócios que eles devem obedecer, seus métodos de captura e armazenamento, a segurança, confiabilidade e também o controle da qualidade destes dados, que é de fundamental importância, haja vista que não poderemos tomar decisões corretas a partir de dados sem qualidade.</p>
<p><b>"COLLECT IT"</b> Fazer a coleta dos dados.</p>	<p>Após termos identificados os dados que serão necessários, teremos que coletá-los, tratá-los e entendê-los e ter a certeza de que eles terão a qualidade suficiente para serem úteis na nossa tomada de decisão ou na resolução do problema proposto que temos em mãos.</p>
<p><b>"STORE IT"</b> Após feita a coleta dos dados úteis, estes devem ser guardados.</p>	<p>Nesta fase, os dados já foram devidamente coletados, tratados e identificados como dados úteis, portanto, teremos que guardá-los, definindo uma forma de armazenamento, para que possam ser reutilizados no futuro.</p>
<p><b>"COMBINE IT"</b> Combinar os dados com outros elementos de dados, vindos de outras fontes, que irão contribuir com a melhor visão da situação que estamos tentando elucidar.</p>	<p>Nesta fase, os meus dados já foram definidos, já sabemos exatamente que dados e como estes devem ser apresentados. No entanto, podemos precisar ter uma visão mais ampla do contexto que nossos dados nos apresentam. Como exemplo, estamos tentando tomar uma decisão de como devemos conduzir as nossas campanhas de marketing para os produtos da minha empresa. Então, além das informações que temos sobre as vendas, precisaremos combiná-las com as informações das tendências do mercado nacional e até internacional. Desta forma, precisaremos combinar os nossos dados com os dados sobre o cenário nacional e internacional, para ter uma visão ampla do mercado atual, a fim de que a tomada de decisão ou o problema que precisamos resolver estejam incluídos num cenário global, onde as tendências irão afetar em muito as decisões que serão tomadas.</p>

<p><b>“ACT ON IT”</b> Ações estratégicas importantes serão tomadas com base nos dados.</p>	<p>Os elementos de dados estão prontos para serem a base das ações estratégicas para o negócio. Inclusive, podem ser usados para outros fins dentro da empresa, uma vez que já foram analisados, entendidos e definidos que são úteis para a tomada de decisão ou resolução de problemas.</p> <p>Nesta fase, já há a comprovação de que os dados têm uma qualidade refinada e que podem, seguramente, apoiar uma tomada de decisão estratégica que impactará positivamente os negócios da empresa, ou que são fundamentais para a resolução de um problema.</p>
<p><b>“ARCHIVE IT”</b> Arquivar os dados definitivamente, preocupando-se em manter cópias de segurança.</p>	<p>Nesta fase, os dados coletados, tratados, combinados, já representam um conjunto de dados de tamanho considerável e de grande importância. Portanto, sendo necessário guardá-los definitivamente e adicionalmente ter a preocupação com cópias de segurança, que são chamados de “backups”, que permitirão reduzir o volume dos dados em ambiente de produção e também de propiciar uma forma de recuperação dos dados no caso de um desastre.</p>
<p><b>“REMOVE IT”</b> Definir política para a remoção dos dados que não têm mais importância para os negócios.</p>	<p>Os dados serão guardados em dispositivos de armazenamento durante um determinado período. Portanto, políticas devem ser definidas para a manutenção destes dados. Em muitos casos, podemos ter até políticas definidas por legislação, por exemplo, manter os dados guardados durante um período de cinco anos. Ou mesmo políticas definidas no âmbito da empresa, por exemplo, podemos definir que em ambiente de produção fiquem gravados os dados do último ano, mas que a empresa tem que manter todos os dados dos últimos cinco anos.</p>
<p><b>“NEED IT AGAIN”</b> Novas necessidades de dados surgem.</p>	<p>Nos processos diários das empresas, com a dinâmica das tomadas de decisões ou a necessidade de resolução de problemas, necessidades de dados sempre surgem. O trabalho feito de identificação, coleta, tratamento e armazenamento de dados, pode ser útil sempre que surgem estas novas necessidades. Outros dados podem ser necessários, mas os que foram tratados com certeza também serão muito utilizados. Além disto, séries históricas podem ser necessárias e os dados guardados nas mídias de armazenamento também podem ter que fazer parte do novo trabalho de coleta e análise de dados. Daí a importância da manutenção das cópias de segurança, para futuras recuperações de dados.</p>
<p><b>“RESTORE IT”</b> Definir políticas para a ocorrência de necessidades de restauração de dados.</p>	<p>Numa nova fase de coleta de dados, os dados já coletados anteriormente, tratados e guardados, podem ser novamente necessários, para que em combinação com outros dados coletados em outras fontes possam ser a base para novas tomadas de decisões ou resolução de problemas. Portanto, uma política de manutenção de cópias dos dados, conjuntamente com uma política de rápida recuperação de dados, devem ser implementadas para suportar uma nova fase de coleta e análise de dados.</p>

Fonte: Elaborado pelo autor.

Para que seja possível acompanhar o ciclo de vida dos dados, faz-se necessário termos um **software especializado** para este fim. Este software foi sendo desenvolvido ao longo de várias décadas e é conhecido atualmente como **Sistema de Gerenciamento de Banco de Dados** (SGBD). A sigla DBMS (*database management system*) também é bastante utilizada na literatura de banco de dados. Ele se constitui em uma ferramenta poderosa para criar, gerenciar e manter grandes quantidades de dados de forma eficiente, permitindo que persistam por longos períodos de tempo, primando pelo aspecto da segurança, que impede o acesso não autorizado aos dados neles gravados.

Segundo Abraham Silberschatz (2006, p. 24):

Um sistema de gerenciamento de banco de dados (DBMS) é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. A coleção de dados, normalmente chamada banco de dados, contém informações relevantes a uma empresa. O principal objetivo de um DBMS é fornecer uma maneira de recuperar informações de banco de dados que seja tanto *conveniente* quanto *eficiente*.

Nesta definição, podemos identificar dois conceitos importantes: o primeiro deles é o do **sistema gerenciador de bancos de dados** (SGBD ou DBMS), que fornece aos usuários, através de seus programas, a capacidade de incluir, excluir, alterar e consultar os dados existentes nos bancos de dados que o compõem. Para que estas ações sejam possíveis, ele disponibiliza uma linguagem de manipulação de dados chamada **SQL** (*structured query language*) ou linguagem estruturada de consulta. Além da manipulação dos dados, esta linguagem ainda permite a criação, alteração e exclusão das estruturas de dados componentes do banco de dados. Para que o SGBD possa executar sua tarefa de manutenção de grandes quantidades de dados, ele, internamente, possui inúmeras **funcionalidades especializadas**, das quais, resumidamente, podemos citar:

- Funcionalidades responsáveis pela segurança dos dados, que impedem que usuários ou sistemas não autorizados acessem aos bancos de dados;
- Funcionalidades responsáveis por manter a integridade dos bancos de dados, que atuam para manter a consistência dos dados armazenados;

- Funcionalidades responsáveis pelo controle de concorrência, que permitem e controlam o acesso compartilhado ao banco de dados, uma vez que os dados devem estar disponíveis para muitos usuários ou sistemas;
- Funcionalidades para o controle de recuperação, cujo papel é o de restaurar os bancos de dados para um estado de consistência. Basicamente, o controle de recuperação permite que em caso de ocorrência de algum desastre onde haja a perda de dados ou a corrupção dos bancos de dados, impedindo o seu acesso normal pelos usuários ou sistemas, seja causado por uma falha de hardware, de software; ou uma atuação indevida de usuários, este mecanismo disponibiliza ferramentas para a recuperação. Dois termos são bem conhecidos nos jargões de banco de dados: o “*backup*”, que é a ação de gerar cópias de segurança dos bancos de dados, portanto, os SGBDs possuem ferramentas especializadas responsáveis por gerar estas cópias; e o “*restore*”, que é a ação de, tendo em posse da cópia de segurança, permitir a devida restauração do banco de dados para um estado de consistência que possa voltar a ser utilizado, portanto, o SGBD também disponibiliza ferramentas especializadas para este fim;
- Funcionalidade para manter um catálogo acessível ao usuário, que contém descrições dos dados no banco de dados, ou seja, manter “dados sobre os dados”, que são chamados de metadados. Os metadados descrevem as estruturas dos bancos de dados, tamanhos dos dados e tipos, restrições, chaves primárias, chaves estrangeiras, além de manter informações sobre os objetos dos bancos de dados, como suas tabelas, visões, funções, procedimentos armazenados etc., que explicitam as especificidades e dependências técnicas do recurso; inclui também os metadados voltados para apoio à gestão dos direitos relacionados ao recurso.

O outro conceito importante retirado da definição é o conceito de **banco de dados**, que está definido como uma **coleção de dados inter-relacionados**. Podemos acrescentar que um banco de dados deve ser estruturado de forma a permitir **acessos** independentemente dos **programas** que os usuários utilizam, desde que estejam devidamente autorizados, eles devem ser capazes de adicionar novos dados, modificá-los, recuperá-los e

excluí-los. Para isto, os bancos de dados são organizados em uma estrutura que contém campos, registros e arquivos. Por **campo**, entende-se ser a menor unidade de dados que possa compor uma informação, por exemplo um nome, um endereço. Os registros são uma coleção de campos que têm um relacionamento lógico, por exemplo, um **registro** formado pelo campo matrícula, pelo campo nome e pelo campo endereço, formando um registro sobre um funcionário. Os **arquivos**, por sua vez, são coleções de registros também logicamente relacionados entre si, por exemplo, um arquivo com as informações de todos os funcionários de uma empresa.

## 2. TIPOS DE BANCO DE DADOS

Existem basicamente dois tipos de banco de dados: os relacionais e os não relacionais. No caso dos bancos de dados **relacionais**, estes seguem os conceitos do modelo relacional proposto por Edgar Frank Codd, um matemático britânico que, em 1970, publicou um artigo intitulado *Relational Model of Data for Large Shared Data Banks* (Modelo de dados relacional para grandes bancos de dados compartilhados). O tipo de banco de dados **não relacionais** ou **NoSQL** trouxe um paradigma diferente na tecnologia de armazenamento de dados, permitindo velocidade, escalabilidade e flexibilidade para o tratamento dos dados não estruturados. Os bancos de dados NoSQL estão fora do escopo deste livro, mas segundo Sadalage e Fowler (2013, p. 35):

[...] há o fato óbvio de que bancos de dados NoSQL não utilizam SQL. Alguns deles têm linguagens de consulta, e faz sentido que elas sejam semelhantes ao SQL para que sejam mais facilmente aprendidas. [...]

## 3. O MODELO RELACIONAL

Neste modelo, os dados em um banco de dados são representados por conjuntos de relações que são chamados de **tabelas**. Cada tabela terá um nome único que a identificará dentro do seu respectivo banco de dados. Ela será composta por um **conjunto de atributos**, que podem ser chamados de **colunas** ou **campos** formando os registros ou tuplas. As colunas ou campos possuirão nomes que as identificarão e um domínio que definirá o tipo de dados que será armazenado. De uma forma geral, os elementos que compõem um banco de dados relacional são os seguintes:

**Entidade:** pode representar dentro de um banco de dados qualquer coisa do mundo real, seja ela concreta ou abstrata, como exemplo de entidade, teremos os clientes de uma loja, os alunos de uma universidade, os funcionários de uma empresa etc.;

**Atributo:** é uma propriedade da entidade. A entidade é formada por um grupo de atributos. No exemplo dos clientes de uma loja, um dos atributos seria o nome do cliente. No exemplo da universidade, um atributo seria a matrícula do aluno. No caso dos funcionários de uma empresa, um atributo seria a data da admissão etc.;

**Relacionamento:** é uma relação lógica entre duas ou mais entidades. Representa uma regra do negócio, que mostra como uma determinada entidade se relaciona com as demais existentes no banco de dados, como os dados se associam entre eles e qual o seu grau de dependência com as demais entidades do banco de dados. Como exemplo, podemos ter uma entidade que representa os funcionários de uma empresa e uma entidade que representa os dependentes deste funcionário, portanto, estas entidades se relacionam e representam uma regra do negócio, existirão funcionários sem dependentes, funcionários com dependentes menores que receberão auxílio creche etc.

Quando falamos em **relacionamentos** entre as entidades, devemos notar que um novo conceito aparece, este conceito é a **cardinalidade**, que se refere ao relacionamento entre duas entidades, os tipos básicos de cardinalidade são:

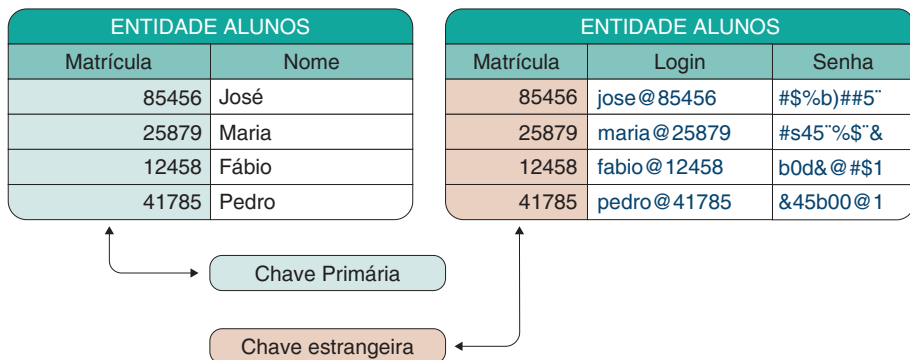
- **Relacionamento 1 para 1**

Neste tipo de relacionamento, no qual a cardinalidade é de “1 para 1”, significa dizer que um único registro na primeira entidade está relacionado a um único registro na segunda entidade. A **chave primária** da primeira entidade irá compor a segunda entidade, mas não como chave primária, este atributo transforma-se em uma **chave estrangeira**. Numa situação em que aparece este tipo de relacionamento, devemos verificar se não há a possibilidade do atributo da segunda entidade ser convertido em um atributo da primeira entidade.



Vamos analisar a figura a seguir:

**Figura 02:** Chaves primárias e chaves estrangeiras.



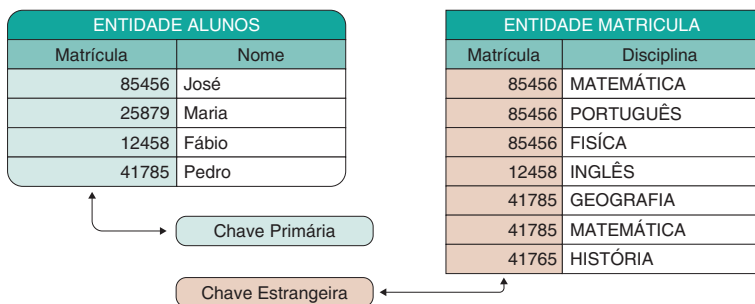
Fonte: Elaborada pelo autor.

Na primeira entidade, a matrícula é a chave primária, e na entidade segurança é chave estrangeira. Notamos que na segunda entidade há somente um registro por matrícula, então, dizemos que a cardinalidade do relacionamento é de “1 para 1”. Dependendo das nuances do projeto, os atributos login e senha da entidade segurança poderiam ser atributos da entidade alunos, portanto, a entidade segurança deixaria de existir.

- **Relacionamento 1 para N**

A cardinalidade, neste caso, é de “1 para N”, isto é, um registro na primeira entidade faz referência a vários registros na segunda entidade. O atributo correspondente à **chave primária** da primeira entidade passa a ser um atributo da segunda entidade, e será chamada de **chave estrangeira**, pois representa um atributo que vem de outra entidade. Na primeira entidade, o atributo de chave primária identifica unicamente o registro e, portanto, não pode ser repetido, já na segunda entidade, este atributo poderá se repetir diversas vezes, como mostra a figura. A primeira entidade, a entidade alunos, possui o atributo matrícula, que representará unicamente um aluno. Este atributo irá para a entidade matrícula e pode ser repetido várias vezes, pois um aluno poderá ser matriculado em diversas disciplinas.

**Figura 03:** Relacionamento entre entidades.



Fonte: Elaborada pelo autor.

### • Relacionamento N para M

Os relacionamentos entre entidades que possuem a cardinalidade “M para N”, ou seja, vários registros da primeira entidade se relacionam com vários registros da segunda entidade, exigem que se faça a criação de uma entidade intermediária, de forma que possamos reduzir este tipo de relacionamento para o tipo de cardinalidade “1 para N”. A **chave primária** da primeira entidade vai se transformar em atributo na entidade intermediária (**chave estrangeira**), e a chave primária da segunda entidade também vai se transformar em atributo na entidade intermediária (**chave estrangeira**).

Para exemplificar um relacionamento entre entidades do tipo “N para M”, tomemos a entidade publicações da figura.

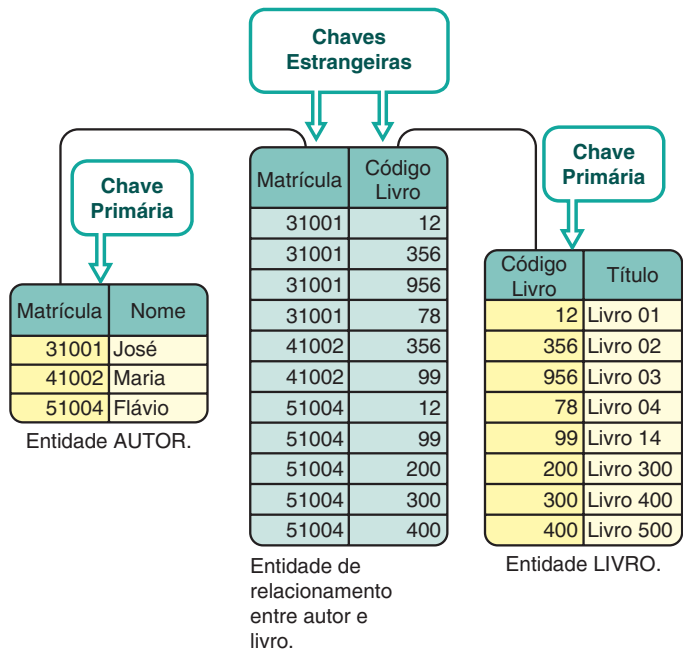
**Figura 04:** Entidade publicações.

Matrícula	Nome	Código Livro	Título
31001	José	12	Livro 01
31001	José	356	Livro 02
31001	José	956	Livro 03
31001	José	78	Livro 04
41002	Maria	356	Livro 02
41002	Maria	99	Livro 14
51004	Flávio	12	Livro 01
51004	Flávio	99	Livro 14
51004	Flávio	200	Livro 300
51004	Flávio	300	Livro 400
51004	Flávio	400	Livro 500

Fonte: Elaborada pelo autor.

Ela possui informações dos autores e seus respectivos livros publicados. Podemos notar que os autores possuem mais de um livro publicado e, por sua vez, os livros podem ter mais de um autor.

Figura 05: Relacionamento entre entidades.



Fonte: Elaborada pelo autor.

De acordo com as regras de cardinalidade do modelo relacional, teremos que criar uma **entidade separada** para os autores e também para os livros. Observamos, neste caso, que o atributo matrícula na entidade autor agora identifica unicamente um registro. Da mesma forma, o atributo código do livro na entidade livro identifica unicamente um registro. Portanto, estes dois atributos serão as **chaves primárias** de suas respectivas entidades.

Agora, criaremos uma entidade para relacionar a entidade autor com a entidade livro. Esta entidade vai conter o atributo matrícula vindo da entidade autor e o atributo código do livro vindo da entidade livro. Estes dois atributos serão as chaves estrangeiras, pois são referenciadas pelas chaves primárias de cada uma das entidades livro e autor.

Estes relacionamentos foram transformados em **dois relacionamentos** com a cardinalidade “**1:N**”, o que significa que um registro na entidade autor, identificado unicamente pelo atributo matrícula, que é a chave primária, está relacionado a **vários registros** na entidade que relaciona autor e livro pelo atributo matrícula, que representa a chave estrangeira. Da mesma forma, um registro na entidade livro, identificado unicamente pelo atributo código do livro, que é a chave primária, relaciona-se com vários registros na entidade de relacionamento autor e livro, pelo atributo código do livro, que é a chave estrangeira.

**Domínio:** é um conjunto de valores possíveis de um atributo. Como exemplo, vamos pegar o atributo data de admissão do funcionário, que fará parte da entidade que representará todos os funcionários de uma determinada empresa. Este atributo terá que guardar informações sobre datas, então, o seu domínio será um conjunto de datas.

“ Este atributo terá que  
guardar informações sobre datas,  
então, o seu domínio será um  
conjunto de datas. ”

**Chave:** pode ser formada por um ou mais atributos, que identificarão uma linha ou tupla de forma única, isto é, se a chave for formada por um único atributo (chave simples), o valor deste atributo não poderá se repetir. Como exemplo, temos a matrícula 31.855 de um funcionário, que é única dentro de uma empresa, somente um funcionário será identificado por ela. Podemos também ter a ocorrência da chave formada por mais de um atributo (**chave composta**), por exemplo, as regras da empresa permitem que as matrículas dos funcionários se repitam, desde que eles trabalhem em filiais da empresa localizadas em unidades da federação diferentes. Então, no Ceará, poderemos ter a matrícula 31.855, e poderemos ter a mesma no estado do Amazonas. O que irá garantir a unicidade desta informação dos funcionários será a composição do número da matrícula do funcionário com a identificação da unidade da federação onde está localizada a filial da empresa. Desta forma, para fazer a identificação de um funcionário em particular, precisaremos de uma chave composta formada pelas informações da matrícula e também do identificador da unidade da federação.

As chaves podem ser do tipo primária ou estrangeira:

- **Chave primária:** é a que identifica unicamente um registro de dados em uma relação;
- **Chave estrangeira:** é um atributo que permite relacionar duas ou mais entidades.

#### Observação:

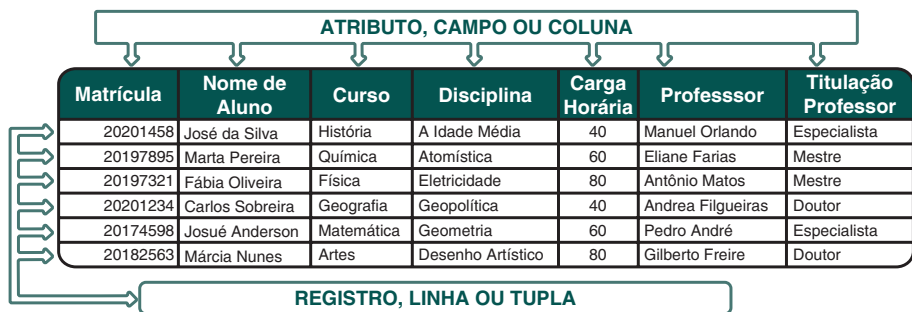
- **Chave candidata:** pode ocorrer que uma entidade tenha mais de um atributo que identifique unicamente um registro desta entidade, neste caso, teremos várias chaves candidatas a serem chave primária da entidade. Como cada entidade só pode ter uma chave primária, somente uma delas será selecionada;
- **Superchave:** quando ocorrer que um conjunto de atributos possam identificar de forma única um registro da entidade, dizemos que este conjunto é uma superchave. Quando a superchave não pode ser reduzida a outras superchaves, ela se transforma em uma chave candidata.

## 4. PROJETO DE BANCO DE DADOS - NORMALIZAÇÃO

No contexto de bancos de dados relacionais, as informações são organizadas em **tabelas** que podem ser vistas como uma planilha, formada por colunas e linhas. Na literatura de banco de dados, poderemos encontrar as colunas referenciadas como **campos** ou **atributos** e as linhas como **registros** ou **tuplas**.

Na figura abaixo, temos informações sobre alunos de uma instituição de ensino, seus cursos, as disciplinas nas quais estão matriculados, os respectivos professores com suas titulações, colocadas em uma tabela do tipo utilizada pelo software Excel, que é bastante útil para guardar este tipo de informação.

**Figura 06:** Componentes de uma entidade.



Fonte: Elaborada pelo autor.

Antes de discutirmos a utilidade desta planilha para o gerenciamento dos dados, vamos reproduzir um novo conceito, o conceito de **normalização**. Segundo Felipe Machado e Maurício Abreu (2004, p. 155), a normalização é definida como a seguir:

O conceito de normalização foi introduzido por E. F. Codd em 1970 (primeira forma normal). Esta técnica é um processo matemático formal, que tem seus fundamentos na teoria dos conjuntos.

Através deste processo pode-se, gradativamente, substituir um conjunto de entidades e relacionamentos por um outro, o qual se apresenta "purificado" em relação às anomalias de atualização (inclusão, alteração e exclusão) as quais podem causar certos problemas, tais como: grupos repetitivos (atributos multivalorados) de dados, dependências parciais em relação a uma chave concatenada, redundâncias de dados, desnecessárias perdas acidentais de informação, dificuldade na representação de fatos da realidade observada e dependências transitivas entre atributos.

Os autores descrevem em sua definição do conceito da técnica de normalização sobre as **anomalias de atualização**, que podem causar problemas. Para entender este conceito vamos trabalhar em um exemplo prático.

Sobre a nossa planilha da figura anterior, vamos verificar algumas características mostradas no quadro a seguir:

**Quadro 02: Anomalias de atualização.**

Matrícula	Nome de Aluno	Curso	Disciplina	Carga Horária	Professor	Titulação Professor
20201458	José da Silva	História	A Idade Média	40	Manuel Orlando	Especialista
20197895	Marta Pereira	Química	Atomística	60	Eliane Farias	Mestre
20197321	Fábia Oliveira	Física	Eleticidade	80	Antônio Matos	Mestre
20201234	Carlos Sobreira	Geografia	Geopolítica	40	Andrea Filgueiras	Doutor
20174598	Josué Anderson	Matemática	Geometria	60	Pedro André	Especialista
20182563	Márcia Nunes	Artes	Desenho Artístico	80	Gilberto Freire	Doutor
OPERAÇÃO DE INSERÇÃO DE REGISTROS		Tomando como base a tabela apresentada acima, verificamos que sempre que haja a necessidade de inserir um novo registro, como exemplo, mais um aluno será matriculado no curso de História que irá cursar a disciplina “A Idade Média”, teremos que repetir as informações de curso, disciplina, carga horária, nome do professor e a sua titulação. Isto, estamos descrevendo o trabalho de inserir um aluno, imagine a entrada de 100 registros de alunos.				
OPERAÇÃO DE ATUALIZAÇÃO DE REGISTROS		Vamos supor que a disciplina de “Geometria” mudou de nome e passou a se chamar “Geometria Plana”, imaginemos uma instituição de ensino como a UniATENEU com milhares de alunos matriculados em uma disciplina, teremos que percorrer cada um dos alunos matriculados na disciplina “Geometria” e fazer a modificação para cada registro. Outra situação seria a mudança do professor da disciplina, teríamos um trabalho semelhante, modificando para cada aluno o nome do novo professor da disciplina. Por mais que a ferramenta que estivermos utilizando tenha recursos para minimizar o trabalho, como o software Excel, mesmo assim ainda será um trabalho considerável e a manutenção da consistência dos dados será totalmente dependente da atenção de quem está fazendo.				
OPERAÇÃO DE EXCLUSÃO DE REGISTROS		No caso da necessidade da exclusão de registros na nossa planilha, observemos que ao fazê-lo, perderemos as informações do curso, da disciplina e do professor. Se por um acaso eu tiver apenas um aluno matriculado em uma determinada disciplina, se for necessário excluí-lo, perderemos todas as informações referentes ao curso, disciplina e professor.				

Fonte: Elaborado pelo autor.

Após esta explanação, devemos observar que a normalização do banco de dados é uma tarefa de extrema importância. Portanto, a preocupação com uma boa **modelagem dos dados** na fase do projeto do banco evitará que tenhamos os problemas apontados por Felipe Machado e Maurício Abreu (2004, p. 155), quando da sua definição de normalização, que transcrevemos a seguir:

[...] podem causar certos problemas, tais como: grupos repetitivos (atributos multivalorados) de dados, dependências parciais em relação a uma chave concatenada, redundâncias de dados, desnecessárias perdas acidentais de informação, dificuldade na representação de fatos da realidade observada e dependências transitivas entre atributos.

Voltando um pouco à definição de Sistemas Gerenciadores de Banco de Dados, dada por Abraham Silberschatz (2006, p. 24), que ele define como “uma coleção de dados inter-relacionados”, podemos compor esta definição de Silberschatz com a definição de normalização de Felipe Machado e Maurício Abreu e deduzir que esta **coleção de dados** deve ser modelada corretamente com o intuito de aproveitar todo o potencial que o SGBD pode oferecer na **recuperação** dos dados para que seja tanto “conveniente quanto eficiente”. A definição de Silberschatz ainda cita o “conjunto de programas para acessar esses dados”, que para funcionar eficientemente, tornando o nosso **banco de dados de alto desempenho**, necessita que a “coleção de dados inter-relacionados” esteja atendendo às modernas técnicas de modelagem de dados.

## 5. AS FORMAS NORMAIS MAIS UTILIZADAS

As formas normais mais utilizadas são as descritas abaixo, mas existem outras. Para o nosso banco de dados que estaremos projetando, as **três formas normais** serão suficientes. São elas:

- Primeira forma normal (1NF);
- Segunda forma normal (2NF);
- Terceira forma normal (3NF).

Para as definições das formas normais, utilizaremos as do livro *Projeto e modelagem de banco de dados*, de 2014, cujos autores são Toby Teorey, Sam Lightstone, Tom Nadeau e H. V. Jagadish, traduzido por Daniel Vieira:

Uma tabela estará na primeira forma normal (1FN) se, e somente se, todas as colunas tiverem apenas valores atômicos, ou seja, se cada coluna só puder ter um valor para cada linha na tabela.

Uma tabela está na segunda forma normal (2FN) se, e somente se, ela estiver na 1FN e os atributos não chaves forem totalmente dependentes da chave primária. Um atributo será totalmente dependente da chave primária se estiver no lado direito de uma DF que tem no lado esquerdo a própria chave primária ou algo que possa ser derivado da chave primária usando a transitividade das DFs.



Uma tabela está na terceira forma normal (3FN) se, e somente se, para cada dependência funcional não trivial  $X \rightarrow A$ , onde  $X$  e  $A$  são atributos simples ou compostos, uma das duas condições precisa ser mantida: ou o atributo  $X$  é uma superchave, ou o atributo  $A$  é membro de uma chave candidata. Se o atributo  $A$  é membro de uma chave candidata,  $A$  é chamado de atributo primo. (TEOREY *et al.*, 2014, p. 111-117)

Nas definições das formas normais acima, o termo “**DF**” significa **dependência funcional**, que é uma propriedade em que um atributo determina unicamente o valor de um outro atributo. A notação  $X \rightarrow A$  significa que o atributo **X** determina o atributo **A**. Como exemplo, a matrícula do aluno identifica os demais atributos do aluno. De posse da informação de matrícula, teremos acesso às demais informações sobre o aluno, tais como o seu nome, curso, seu endereço, telefone, notas etc.

Uma definição adicional é a **dependência funcional transitiva**, ela ocorre quando um atributo não depende diretamente da chave primária, mas depende de um outro atributo da entidade. Como exemplo, a disciplina não depende da matrícula do aluno, que é a chave primária da entidade. Poderemos ter diversos alunos matriculados em diversas disciplinas, a matrícula do aluno não garante a identificação da disciplina. Se desejo obter informações sobre as disciplinas, deverei ter uma chave primária para identificar unicamente cada disciplina e que agrupe os demais atributos referentes a cada disciplina, tais como carga horária, horários, salas etc.

“ Uma definição adicional  
é a dependência funcional  
transitiva...”

Para um melhor entendimento da **normalização**, iremos desenvolver um exemplo prático, tomando como base a figura abaixo (Quadro 03: Planilha de controle de vendas), que representa informações sobre as vendas das filiais de uma determinada loja de departamentos. O objetivo será utilizar a planilha para modelar um banco de dados que possa ser implementado em um sistema gerenciador de banco de dados relacional, no nosso caso, utilizaremos o **MySQL**.

**Quadro 03:** Planilha de controle de vendas.

Código da Venda	Nome do Cliente	Endereço do Cliente	CPF do Cliente	Data da Venda	Produto	Vendedor	Quantidade	Preço Unitário (R\$)	Valor Total da Venda (R\$)	Loja	Departamento
1256	José da Silva	Rua Júlio Alcides, 323, Maraponga, - Fortaleza-Ceará	125.569.478-49	11/11/2015	Impressora laser	Manuel Orlando	1	1.200,00	1.200,00	Maraponga	Informática
325	Marta Pereira	Avenida Pereira Filgueiras, 567, Centro, São Paulo -São Paulo	159.234.852-56	20/05/2020	Pneus	Eliane Farias	4	236,00	944,00	Centro	Automotivo
2356	Fábia Oliveira	Travessa Santos Dumont, 1256, Águas Belas, Niterói-Rio de Janeiro	169.425.888-69	14/05/2014	Geladeira	Antônio Matos	1	2.546,00	2.546,00	Niterói	Eletrodomésticos
12	Carlos Sobreira	Vila São José , 234 , Acaí, Santarém-Pará	178.523.962-89	25/02/2013	CPU Dell	Andrea Filgueiras	1	3.521,00	3.521,00	Santarém	Informática
896	Josué Anderson	Rua Floriano Peixoto, 876, Centro, Salvador-Baia	128.658.963-78	15/08/2003	Sapato masculino	Pedro André	10	158,00	1.580,00	Pelourinho	Roupas e Calçados
3654	Márcia Nunes	Rua Gomes de Matos, 6789, Montese, Fortaleza-Ceará	137.965.225-52	01/01/2020	Violão	Gilberto Freire	2	1.547,00	3.094,00	Montese	Instrumentos Musicais

Fonte: Elaborada pelo autor.

Para normalizar o quadro anterior, deveremos aplicar as regras práticas abaixo:

## I. A primeira forma normal

- Identificar as **colunas** que contenham **dados não atômicos**, ou seja, aquelas colunas que contém dados que podem ser separados em novas colunas. Como exemplo, a coluna “Endereço do Cliente”, possui o tipo de logradouro (Rua, Avenida, Travessa) que pode compor uma coluna chamada “Tipo de Logradouro”; possui na mesma coluna o número da residência do cliente que poderá compor uma nova coluna chamada “Número”; traz também a Cidade e o Estado do cliente que poderiam ser desmembradas em outras duas colunas, uma referente à “Cidade” e outra ao “Estado”;
- Identificar qual a coluna ou colunas combinadas que possam identificar unicamente uma linha da **tabela**. Esta coluna ou colunas combinadas irá compor a **chave primária**. Neste momento da normalização isto não é possível, porque existem colunas que não são totalmente dependentes de uma única coluna que fosse selecionada como chave primária. Como exemplo, na tabela, temos informações do produto, cliente, vendedor, loja e departamento que não podem ser identificados por uma única chave primária. Eles terão que ser desmembrados em novas entidades que terão suas respectivas chaves primárias;
- Identificar os grupos de **colunas** que formarão outras entidades e separar estes grupos de colunas;
- Devem ser criadas **novas tabelas** com os grupos identificados no item anterior e gerar as chaves primárias.

## II. A segunda forma normal

- A tabela já está na primeira forma normal;
- Identificar todas as colunas que agrupadas serão dependentes funcionalmente de sua própria **chave primária**. Como exemplo, as colunas de cliente, loja, produto, departamento e vendedor terão cada uma suas próprias chaves primárias;

- Estes atributos identificados devem ser removidos da tabela e serem colocados em novas tabelas que agruparão as colunas que a eles pertencem. Como exemplo, a coluna “preço unitário” pertence ao produto, não tendo um relacionamento direto com o cliente ou a loja;
- A **chave primária** da nova entidade será o atributo do qual os atributos removidos são funcionalmente dependentes, por exemplo, no caso do produto, seu nome e valor unitário; no caso do cliente, seu nome e seu endereço etc.

### III. A terceira forma normal

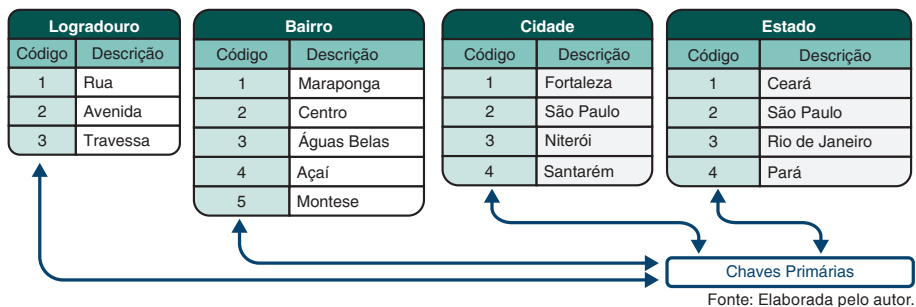
- A tabela já está na segunda forma normal;
- Neste momento da normalização, serão tratadas as dependências funcionais transitivas. Todas elas devem ser removidas, significando, como exemplo, que o cliente terá a sua própria chave primária, assim como o vendedor, o produto, a loja e os departamentos.

No final da aplicação da **técnica da normalização** na nossa tabela inicial, teremos que ter eliminado as possibilidades de termos as seguintes **anomalias**:

- Não deveremos ter informações redundantes no nosso banco de dados;
- Não devem existir dependências parciais em relação às chaves primárias selecionadas;
- Nas operações de alterações de dados no banco de dados (inclusão, alteração, exclusão) jamais devem ocorrer perdas acidentais de informação;
- Todas as dependências funcionais transitivas entre os atributos da entidade em relação às chaves primárias devem ter sido eliminadas.

Iremos iniciar a normalização verificando quais as colunas que possuem **informações não atômicas**. Podemos observar que a coluna “Endereço do Cliente” na tabela possui informações que devem ser separadas, tais como as informações de logradouro, bairro, cidade e estado.

**Figura 07:** Chaves primárias.



Antes de continuarmos, faremos aqui um comentário sobre a razão pela qual é tão importante fazermos as devidas normalizações. Uma das premissas fundamentais para um **banco de dados de alto desempenho**, é que ele seja **rápido** quando for necessário que ele responda a **consultas** feitas pelos usuários. Imagine um banco de dados de uma empresa de atuação nacional, que tem milhões de clientes cadastrados, e vamos supor que as informações estivessem cadastradas no formato da nossa planilha sem a aplicação da técnica da normalização. Caso necessitássemos fazer uma consulta simples, por exemplo, quais os clientes da empresa que residem no bairro da Maraponga, em Fortaleza, por mais que o sistema gerenciador de banco de dados me ofereça recursos para formatar os dados e encontrar o que desejo, com toda a certeza, as consultas teriam um tempo de resposta inviável. Quando estudarmos a linguagem *Structured Query Language* (SQL), poderemos ter uma visão mais prática sobre este assunto.

Com a ação que executamos, criando **novas entidades**, conseguimos gerar outras tabelas, cuja finalidade principal é tornar as informações do nosso banco de dados atômicas, dando para cada uma destas tabelas uma **chave primária**. Criamos em cada uma das entidades (logradouro, bairro, cidade, estado) uma tabela contendo uma coluna chamada código, que vai identificar unicamente cada linha das nossas novas tabelas. Geralmente, para a criação destes códigos se utiliza nos sistemas gerenciadores de banco de dados um **mecanismo automático**, que garante esta unicidade. Veremos este mecanismo quando estudarmos o sistema gerenciador de banco de dados MySQL.

Agora, passaremos a verificar as **dependências parciais**. Existirão conjuntos de atributos pertencentes a determinadas entidades, que podem ser identificados por um único atributo, que já conhecemos (chave primária). Verificando nossas planilhas, chegaremos à conclusão mostrada no quadro a seguir:

**Quadro 04:** Entidades e tabelas.

ENTIDADES	ATRIBUTOS	TABELA	NOME DA COLUMNA
CLIENTE	Código do Cliente Nome do Cliente Endereço CPF Código do Estado Código da Cidade Código do Bairro Código do Logradouro	tbcliente	Cliente_Codigo Cliente_Nome Cliente_Endereco Cliente_Cpf Cliente_Estado Cliente_Cidade Cliente_Bairro Cliente_Logradouro
PRODUTO	Código do Produto Nome do Produto Preço Unitário	tbproduto	Produto_Codigo Produto_Nome Produto_Precio
LOJA	Código da Loja Nome da Loja	tbloja	Loja_Codigo Loja_Nome
VENDEDOR	Matrícula do Vendedor Nome do Vendedor	tbvendedor	Vendedor_Matricula Vendedor_Nome
DEPARTAMENTO	Código do Departamento Nome do Departamento	tbdepartamento	Departamento_Codigo Departamento_Nome
VENDA	Código da Venda Código do Cliente Código do Produto Código da Loja Código do Departamento Matrícula do Vendedor Data da Venda Quantidade	tbvenda	Venda_Codigo Cliente_Codigo Produto_Codigo Loja_Codigo Departamento_Codigo Vendedor_Matricula Venda_Data Venda_Quantidade
ESTADO	Código do Estado Nome do Estado Sigla	tbestado	Estado_Codigo Estado_Nome Estado_Sigla
CIDADE	Código da Cidade Nome da Cidade	tbcidade	Cidade_Codigo Cidade_Nome
BAIRRO	Código do Bairro Nome do Bairro	tbbairro	Bairro_Codigo Bairro_Nome
LOGRADOURO	Código do Logradouro Nome do Logradouro	tblogradouro	Logradouro_Codigo Logradouro_Nome

Fonte: Elaborado pelo autor.

Antes de prosseguir com a **modelagem** do nosso banco de dados, gostaria de relembrar o conceito de **domínio**, que já foi citado anteriormente, como sendo representado por um conjunto de valores possíveis de um atributo. Neste mapeamento que fizemos das entidades e seus atributos, poderemos notar que os atributos terão diferentes tipos de dados. Como exemplo, todas as chaves primárias são um número inteiro, os nomes do cliente, vendedor são conjuntos de caracteres, a data da compra é um tipo data etc. O quadro a seguir resume os tipos de dados:

**Quadro 05:** Tipos de dados e atributos.

TIPOS DE DADOS	ATRIBUTOS
NÚMERO INTEIRO	Códigos do cliente, do produto, da loja, do departamento, da venda, matrícula do vendedor e quantidade da venda.
CONJUNTO DE CARACTERES	Nomes do departamento, do vendedor, da loja, do produto, endereço, CPF.
DATA	Data da Compra.
NÚMERO DECIMAL	Preço unitário do produto.

Fonte: Elaborado pelo autor.

Verificando estes tipos de dados, notamos que o CPF foi definido como sendo do tipo conjunto de caracteres e não numérico. Isto ocorre porque existem CPFs iniciados pelo número zero, levando em conta este detalhe, se o definirmos como inteiro, o zero da esquerda será perdido, havendo problemas quando se for validar o CPF, pois o algoritmo leva em conta todos os números.

“...as chaves primárias são um número inteiro, os nomes do cliente, vendedor são conjuntos de caracteres, a data da compra é um tipo data etc.”

Para uma melhor ilustração desta questão dos domínios, ou tipos de dados, abaixo está uma tabela com alguns tipos de dados do sistema gerenciador de banco de dados **MySQL**:

**Quadro 06:** Tipos de dados do MySQL.

TIPO DE DADO	DESCRIÇÃO
DECIMAL(M, D)	Ponto decimal com M dígitos no total (precisão) e D casas decimais (escala); o padrão é 10,0; M vai até 65 e D até 30.
INTEGER, INT	Número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -2147483648 até 2147483647. Sem sinal, a margem de valores é de 0 até 429.496.295.
FLOAT(M, D)	Ponto flutuante com precisão M e escala D; o padrão é 10,2; D vai até 24.
CHAR(M)	<i>String</i> que ocupa tamanho fixo entre 0 e 255 caracteres.
BOOL BOOLEAN	Valores binários 0 / 1; Na verdade, é um alias para o tipo TINYINT(1).
VARCHAR(M)	<i>String</i> de tamanho variável, até 65535 caracteres.
BLOB MEDIUMBLOB TINYBLOB	Campo com tamanho máximo de 65535 caracteres binários; <i>Binary Large Objects</i> são usados para armazenar grandes quantidades de dados, como imagens.
MEDIUMTEXT	Permite armazenar até 16.777.215 caracteres.
LONGTEXT	Permite armazenar até 4.294.967.295 caracteres.
DATE	Uma data de 01/01/1000 a 31/12/9999, no formato YYYY-MM-DD.
DATETIME	Uma combinação de data e hora de 01/01/1000 00:00:00 a 31/12/9999 23:59:59, no formato YYYY-MM-DD HH:MM:SS.
TIME	Hora apenas, no formato HH:MM:SS YEAR(M) – Ano nos formatos de 2 ou 4 dígitos; Se forem 2 (YEAR(2)), ano vai de 1970 a 2069; para 4 (YEAR(4)), vai de 1901 a 2155. O padrão é 4.

Fonte: <https://bit.ly/3lrrU5P>.



## FIQUE ATENTO

*Alguns termos utilizados no quadro:*

- **String** - significa uma sequência de caracteres;
- **Alias** - é um termo que significa “apelido”; no quadro, ele descreve que o tipo de dados **BOOLEAN** é um apelido para o tipo de dados **TINYINT(1)**.
- **YYYY-MM-DD** - significa ano, mês e dia.



## 6. O DICIONÁRIO DE DADOS

Verificando os quadros anteriores, podemos notar que seus conteúdos formam um documento com informações sobre os dados que irão compor o nosso banco de dados. Este documento na literatura de banco de dados é conhecido como um **dicionário de dados**.

O dicionário de dados será um dos produtos da primeira fase do projeto de um banco de dados, que é fase de **análise de requisitos**, na qual se faz o levantamento de todas as necessidades de armazenamento de dados, as regras do negócio e toda a informação importante que irá compor o documento final dicionário de dados.

No livro *Projeto e modelagem de banco de dados*, de 2014, cujos autores são Toby Teorey, Sam Lightstone, Tom Nadeau e H. V. Jagadish, traduzido por Daniel Vieira, poderemos ler o trecho abaixo, que esclarece bem esta fase do projeto de banco de dados:

[...] O projetista de banco de dados precisa entrevistar a população de usuários finais e determinar exatamente a finalidade de uso do banco de dados e o que ele precisa conter. Os objetivos básicos da análise de requisitos são os descritos a seguir:

- Delinear os requisitos de dados da empresa em termos dos elementos de dados básicos.
- Descrever a informação sobre os elementos de dados e os relacionamentos entre eles necessários para modelar esses requisitos de dados.
- Determinar os tipos de transações que devem ser executadas no banco de dados e a interação entre as transações e os elementos de dados.
- Definir quaisquer restrições de desempenho, integridade, segurança ou administrativas que tenham de ser impostas sobre o banco de dados resultante.
- Especificar quaisquer restrições de projeto e de implementação, tais como tecnologias, hardware e software, linguagens de programação, políticas, padrões ou interfaces externas específicos.

- Documentar por completo todos os itens anteriores em uma especificação de requisitos detalhada. Os elementos de dados também podem ser definidos em um sistema de dicionário de dados, normalmente fornecido como parte integral do sistema de gerenciamento de banco de dados. (TEOREY *et al.*, 2014, p. 57-58)

As informações constantes em um dicionário de dados são chamadas de **metadados**, ou seja, “são os dados sobre os dados”. A citação retirada do site *Safer net* dá uma definição sucinta, mas esclarecedora:

O prefixo “Meta” vem do grego e significa “além de”. Assim, Metadados são informações que acrescem aos dados e que têm como objetivo informar-nos sobre eles para tornar mais fácil a sua organização. Um item de um metadado pode informar do que se trata aquele dado numa linguagem inteligível para um computador. Os metadados têm a função de facilitar o entendimento dos relacionamentos e evidenciar a utilidade das informações dos dados. (SAFER NET, s. d.)

Como exemplo, poderemos ter em uma determinada tabela o dado “15”, este dado, nesta forma, não tem muita utilidade, pois o simples fato de tê-lo armazenado não significa nada. No entanto, se eu acrescentar no meu dicionário de dados que ele significa **15%** e que é a taxa de juros que uma determinada instituição financeira cobra em suas operações de empréstimo ao ano, já teremos então um **metadado de negócio**. Outra informação que preciso adicionar no meu dicionário de dados é que este número é do tipo inteiro e que estará na coluna *taxaJuros* da tabela *OperacoesFinanceiras*. Neste caso, estou me referindo a **metadados técnicos**, que fazem referência a como os dados serão armazenados no nosso banco de dados.

Um bom dicionário de dados deve conter no mínimo as seguintes **informações**:

- Nomes de todas as tabelas e seus respectivos esquemas, data na qual foram criadas e alteradas;
- Nomes das colunas das tabelas com seus tipos de dados e tamanho;
- Informações sobre o banco de dados, nomes, datas de criação e localização dos arquivos físicos;

- Informações sobre as chaves primárias das tabelas e chaves estrangeiras;
- Informações sobre todas as restrições de integridade;
- Detalhamento de todos os objetos que compõem o banco de dados, como as visões, as funções, procedimentos armazenados, índices e gatilhos.

Para iniciar o nosso dicionário de dados, iremos aproveitar as informações dos quadros anteriores e gerar o dicionário de dados mostrado no quadro a seguir. Com o desenvolvimento do nosso banco de dados, outros objetos irão aparecer e serão incluídos no nosso dicionário.

**Quadro 07:** Dicionário de dados - TABELAS.

DICIONÁRIO DE DADOS - TABELAS					
ENTIDADE	TABELA	NOME DO CAMPO	TIPO DE DADOS	TAMANHO	DESCRIÇÃO
CLIENTE	TbCliente	Cliente_Codigo	Integer	Pré-definido	Número que identifica unicamente um cliente dentro da tabela.
		Cliente_Nome	Varchar	100	Cadeia de caracteres que informa o nome do cliente.
PRODUTO	TbProduto	Produto_Codigo	Integer	Pré-definido	Número que identifica unicamente um produto dentro da tabela.
		Produto_Nome	Varchar	100	Cadeia de caracteres que informa o nome do produto.
LOJA	TbLoja	Loja_Codigo	Integer	Pré-definido	Número que identifica unicamente uma loja dentro da tabela.
		Loja_Nome	Varchar	100	Cadeia de caracteres que informa o nome da loja.
VENDEDOR	TbVendedor	Vendedor_Matricula	Inteiro	Pré-definido	Número que identifica unicamente um vendedor dentro da tabela.
		Vendedor_Nome	Varchar	100	Cadeia de caracteres que informa o nome do vendedor.
DEPARTAMENTO	TbDepartamento	Departamento_Codigo	Integer	Pré-definido	Número que identifica unicamente um departamento dentro da tabela.
		Departamento_Nome	Varchar	100	Cadeia de caracteres que informa o nome do departamento.

VENDA	TbVenda	Venda_Codigo	Integer	Pré-definido	Número que identifica unicamente uma operação de venda dentro da tabela.
		Venda_Data	Datetime	Pré-definido	Data e hora na qual ocorreu a venda.
		Venda_Quantidade	Integer	Pré-definido	Quantidade de produtos comprados.
		Venda_Cliente	Integer	Pré-definido	Código do cliente que fez a compra.
		Venda_Produto	Integer	Pré-definido	Código do produto vendido.
		Venda_Loja	Integer	Pré-definido	Código da loja que fez a venda.
		Venda_Departamento	Integer	Pré-definido	Código do departamento do qual foi vendido o produto.
		Venda_Vendedor	Integer	Pré-definido	Matrícula do vendedor que fez a venda.
ESTADO	TbEsta	Estado_Codigo	Integer	Pré-definido	Número que identifica unicamente uma unidade da federação na tabela.
		Estado_Nome	Varchar	100	Nome da unidade da federação.
		Estado_Sigla	Varchar	2	Sigla da unidade da federação.
CIDADE		Cidade_Codigo	Integer	Pré-definido	Número que identifica unicamente uma cidade dentro da tabela.
		Cidade_Nome	Varchar	100	Cadeia de caracteres que informa o nome da cidade.
BAIRRO	TbBairro	Bairro_Codigo	Integer	Pré-definido	Número que identifica unicamente um bairro dentro da tabela.
		Bairro_Nome	Varchar	100	Cadeia de caracteres que informa o nome do bairro.
LOGRADOURO	TbLogradouro	Logradouro_Codigo	Integer	Pré-definido	Número que identifica unicamente um logradouro dentro da tabela.
		Logradouro_Nome	Varchar	100	Cadeia de caracteres que informa o nome do logradouro.

Fonte: Elaborado pelo autor.

**Quadro 08:** Dicionário de dados – CHAVES PRIMÁRIAS.

DICIONÁRIO DE DADOS - CHAVES PRIMÁRIAS				
ENTIDADE	TABELA	NOME DO CAMPO	TIPO DE DADOS	NOME DA CHAVE PRIMÁRIA
CLIENTE	tbcliente	Cliente_Codigo	Integer	pk_Cliente_Codigo
PRODUTO	tbproduto	Produto_Codigo	Integer	pk_Produto_Codigo
LOJA	tbloja	Loja_Codigo	Integer	pk_Loja_codigo
VENDEDOR	tbvendedor	Vendedor_Matricula	Inteiro	pk_Vendedor_Codigo
DEPARTAMENTO	tbdepartamento	Departamento_Codigo	Integer	pk_Departamento_codigo
VENDA	tbvenda	Venda_Codigo	Integer	pk_Venda_Codigo
ESTADO	tbestado	Estado_Codigo	Integer	pk_Estado_Codigo
CIDADE	tbcidade	Cidade_Codigo	Integer	pk_Cidade_Codigo
BAIRRO	tbairro	Bairro_Codigo	Integer	pk_Bairro_Codigo
LOGRADOURO	tblogradouro	Logradouro_Codigo	Integer	pk_Logradouro_codigo

Fonte: Elaborado pelo autor.

**Quadro 09:** Dicionário de dados – CHAVES ESTRANGEIRAS.

DICIONÁRIO DE DADOS - CHAVES ESTRANGEIRAS					
CHAVE ESTRANGEIRA	Lado N do Relacionamento			Lado 1 do Relacionamento	
	ENTIDADE	TABELA	COLUNA	TABELA	CHAVE PRIMÁRIA
fk_cliente_estado	CLIENTE	tbcliente	Estado_Codigo	tbestado	Estado_Codigo
fk_cliente_cidade			Cidade_Codigo	tbcidade	Cidade_Codigo
fk_cliente_bairro			Bairro_Codigo	tbbairro	Bairro_Codigo
fk_cliente_logradouro			Logradouro_Codigo	tblogradouro	Logradouro_Codigo
fk_venda_cliente	VENDA	tbvenda	Cliente_Codigo	tbcliente	Cliente_Codigo
fk_venda_loja			Loja_Codigo	tbloja	Loja_codigo
fk_venda_vendedor			Vendedor_Matricula	tbvendedor	Vendedor_Matricula
fk_venda_produto			Produto_Codigo	tbproduto	Produto_Codigo
fk_venda_departamento			Departamento_Codigo	tbdepartamento	Departamento_Codigo

Fonte: Elaborado pelo autor.



## FIQUE ATENTO

No dicionário de dados, na parte que se refere às tabelas, temos na coluna **tamanho** dados com o tamanho pré-definido. Observando o Quadro 07, vemos que alguns tipos de dados já trazem o seu tamanho definido como o inteiro, a data, o boolean. Já outros tipos, precisamos informar suas dimensões, como o tipo varchar, char, decimal e float.

## 7. APLICANDO A NORMALIZAÇÃO

Após todas estas considerações teóricas que foram feitas sobre **normalização**, e aplicando-as em nosso exemplo, ou seja, normalizando a nossa tabela de vendas, teremos como resultado as tabelas a seguir:

**Figura 08:** Técnica de normalização: novas entidades.

ENTIDADE LOGRADOURO	
Logradouro_Codigo	Logradouro_Nome
1	Rua
2	Avenida
3	Travessa

ENTIDADE BAIRRO	
Bairro_Codigo	Bairro_Nome
1	Maraponga
2	Centro
3	Águas Belas
4	Açaí
4	Montese

ENTIDADE CIDADE	
Cidade_Codigo	Cidade_Nome
1	Fortaleza
2	São Paulo
3	Niterói
4	Santarém

ENTIDADE ESTADO	
Estado_Codigo	Estado_Nome
1	Ceará
2	São Paulo
3	Rio de Janeiro
4	Pará

ENTIDADE LOGRADOURO			
Cliente_Codigo	Cliente_Nome	Cliente_Endereço	Cliente_Cpf
1	José da Silva	Rua Júlio Alcides, 323	125.569.478-49
2	Marta Pereira	Avenida Preira Filgueiras, 567	159.234.852-56
3	Fábia Oliveira	Travessa Santos Dumont, 1256	169.425.888-69
4	Carlos Sobreira	Vila São José, 234	178.523.962-89
5	Josué Anderson	Rua Floriano Peixoto, 876	128.658.963-78
6	Márcia Nunes	Rua Gomes de Matos, 6789	137.965.225-52

ENTIDADE LOJA	
Codigo_Loja	Loja_Nome
1	Maraponga
2	Centro
3	Niterói
4	Santarém
5	Pelourinho
6	Montese

ENTIDADE DEPARTAMENTO	
Departamento_Codigo	Departamento_Nome
1	Informática
2	Automotivo
3	Eletrodomésticos
4	Informática
5	Roupas e Calçados
6	Instrumentos Musicais

ENTIDADE PRODUTO		
Produto_codigo	Produto_Nome	Produto_Preço
1	Impressora laser	1.200,00
2	Pneus	236,00
3	Geladeira	2.546,00
4	Cpu dell	3.521,00
5	Sapato masculino	158,00
6	Violão	1.547,00

ENTIDADE VENDEDOR	
Produto_codigo	Produto_Nome
1	Manuel Orlando
2	Eliane Farias
3	Antônio Matos
4	Andrea Filgueiras
5	Pedro André
6	Gilberto Freire

ENTIDADE VENDA		
Venda_Codigo	Venda_Data	Venda_Quantidade
1	11/11/2015	1
2	20/05/2020	4
3	14/05/2014	1
4	25/02/2013	1
5	15/08/2003	10
6	01/01/2020	2

Fonte: Elaborada pelo autor.

Todas as **dependências funcionais** foram resolvidas, para isto foram criadas tabelas que agregam seus conjuntos de atributos relacionados e criamos uma coluna que será nossa chave primária, e como já foi definido que identificará unicamente cada registro em cada uma das tabelas.

Existe uma coluna na nossa tabela original de controle de vendas, que não foi incluída em nenhuma das tabelas acima, é a coluna “Valor Total da Venda”. Ela é uma coluna calculada, resultante do produto entre as colunas “Preço Unitário” e “Quantidade”. Aqui, estamos projetando um **banco de dados de alto desempenho**, isto significa que além de nos preocuparmos com eficiência, segurança etc., também devemos nos preocupar com a economia de recursos. O **armazenamento de dados** em discos deve ser feito de forma que somente os dados estritamente necessários estejam lá, contribuindo para a velocidade nas operações de busca de dados. Gravar uma coluna calculada é um consumo desnecessário de recursos de armazenamento, porque o cálculo pode ser feito dentro do comando de busca, o que aprenderemos quando estivermos discutindo a **linguagem estruturada de pesquisa** (*Strutured Query Language* – SQL) no banco de dados MySQL.

Identificadas as tabelas, criadas as chaves primárias para cada uma delas, iremos estudar como elas se interligam e implementar o **mecanismo de relacionamento** que já foi discutido.

A nossa planilha original tinha como objetivo de guardar dados referentes a vendas feitas por uma rede de lojas, com diversas filiais presentes em mais de um estado. Então, iniciaremos estudo pela ENTIDADE VENDA, que possui, no presente momento, apenas os atributos:

- Venda\_Codigo, que é a chave primária da entidade;
- Venda\_Data, que representa a data da venda;
- Venda\_Quantidade, que representa a quantidade de produtos vendidos.

Verificamos, portanto, que estes **atributos** não são suficientes para compor completamente as informações sobre uma determinada venda. Se necessitarmos gerar um relatório, por exemplo, não conseguiremos fazê-lo com a configuração atual da ENTIDADE VENDA. Então, iremos complementar



as informações, mas agora, utilizando o relacionamento entre as entidades. Na figura a seguir, está indicado estes relacionamentos entre as diversas tabelas que foram compostas a partir da normalização da tabela original.

Figura 09: ENTIDADE VENDA.

ENTIDADE DE VENDA							
Venda_Codigo	Venda_Data	Venda_Quantidade	Cliente_Codigo	Vendedor_Matricula	Produto_Codigo	Loja_Codigo	Departamento_Codigo
1256	11/11/2015	1	1	1	1	1	1
325	20/05/2020	4	2	2	2	2	2
2356	14/05/2014	1	3	3	3	3	3
12	25/02/2013	1	4	4	4	4	4
896	15/08/2003	10	5	5	5	5	5
3654	01/01/2020	2	6	6	6	6	6

Fonte: Elaborado pelo autor.

Analisando as informações que são necessárias para identificar uma VENDA, chegamos à conclusão que precisamos de informações que estão contidas nas entidades do cliente, do vendedor, do produto, da loja e do departamento. Para resolver isto, complementamos a entidade venda, com as chaves primárias de cada uma das entidades que identificamos ser necessário trazer as informações. Estes atributos agora passam a se chamar **chaves estrangeiras** correspondentes a cada uma das entidades relacionadas. Observe que a chave primária da entidade venda é o atributo “Venda\_Codigo”, que identificará unicamente um registro de venda.

Observemos que apesar de no nosso exemplo da ENTIDADE VENDA ter as chaves estrangeiras sem repetição, ocorre apenas porque o exemplo é muito simples, mas todas elas poderiam ter **valores repetidos** pelos seguintes motivos:

- UM CLIENTE poderia fazer VÁRIAS compras, então, na entidade venda este cliente teria vários registros. Como exemplo, um cliente poderia comprar pneus, uma geladeira, um violão etc.;
- UM VENDEDOR poderia fazer VÁRIAS vendas, portanto, gerando vários registros na nossa entidade venda. Como exemplo, um vendedor poderia vender vários pneus, várias geladeiras, vários violões etc.;
- UM PRODUTO poderia ser vendido VÁRIAS vezes, enquanto ele tivesse estoque. Exemplo, um vendedor ou vários vendedores poderiam vender pneus para diversos clientes;

- UMA LOJA poderia fazer VÁRIAS vendas, portanto, gerando inúmeros registros na entidade venda. Exemplo, a loja da Maraponga poderia fazer centenas de vendas, assim como a loja do Montese etc.;
- UM DEPARTAMENTO poderia ter VÁRIOS de seus produtos vendidos, gerando vários registros na entidade venda. Como exemplo, o departamento de eletrodomésticos poderia vender inúmeros, assim como o departamento de informática também.

Nestes exemplos acima, podemos notar os termos UM e VÁRIOS, o que nos faz relembrar um assunto já discutido anteriormente, os **relacionamentos entre entidades**, especificamente o relacionamento “1 para N”, e é exatamente este tipo de relacionamento que se aplica entre a ENTIDADE VENDA, que será sempre o lado “N” do relacionamento, e as demais entidades, CLIENTE, VENDEDOR, PRODUTO, LOJA, DEPARTAMENTO, que representarão o lado “1” do relacionamento.

É importante observar que nas entidades do lado “1” do relacionamento, os **registros** não se repetem, exatamente porque um dos objetivos da normalização é evitar a ocorrência de informações redundantes. Um cliente terá seus dados de cadastro incluídos no banco de dados somente uma única vez. Similarmente, os registros referentes aos produtos, aos vendedores, às lojas, aos departamentos etc., que em sistemas informatizados formam a parte de **cadastro**. Como exemplo, no cadastro de clientes, poderemos ter mais de um “Manuel Orlando”, desde que não seja a mesma pessoa, tenham ao menos a informação de CPF diferentes. Nestes cadastros, a ocorrência de mais de um registro idêntico constitui-se em um erro de cadastramento.

Agora que, seguindo as regras da normalização, organizamos todas as entidades, vamos representar os seus **relacionamentos**, conforme a figura mostrada a seguir. Sobre a figura, poderemos fazer as seguintes considerações: no caso do endereço, por conta de uma das regras de normalização, tivemos que tornar as informações de endereço atômicas, ou seja, na nossa planilha inicial tínhamos o endereço propriamente dito junto com as informações de logradouro, bairro, cidade e estado, o que resultou em uma entidade para cada uma destes atributos, com as respectivas chaves primárias.

Para gerar o relacionamento entre estas entidades (logradouro, bairro, cidade e estado) e a entidade cliente, criamos na entidade cliente atributos correspondente às chaves primárias de cada uma destas entidades. As chaves primárias das entidades logradouro, bairro, cidade e estados se transformaram em atributos na entidade cliente, com o nome de chaves estrangeiras.

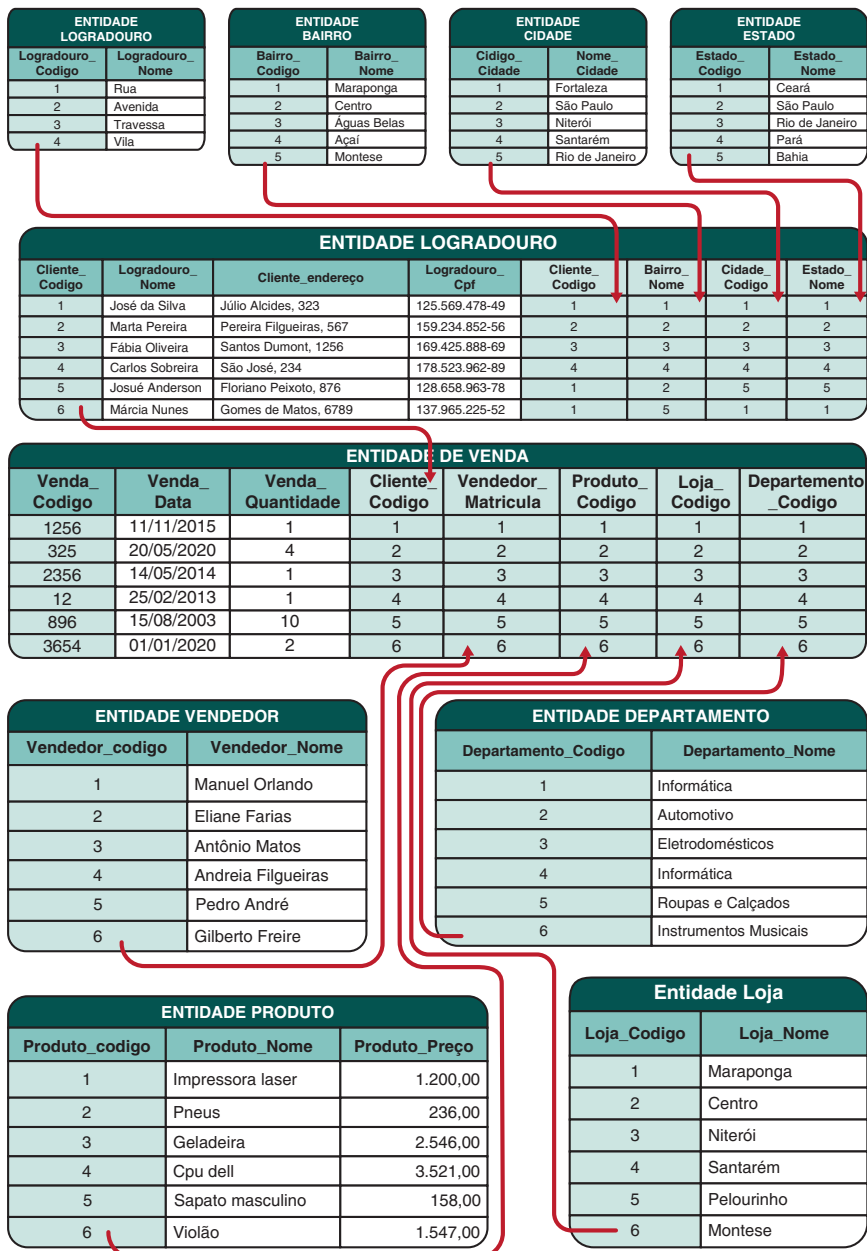
Em função das **regras de normalização** para a resolução das **dependências funcionais**, criamos as entidades cliente, vendedor, produto, loja e departamento. Para relacionar estas entidades com a entidade venda, criamos na entidade venda um atributo correspondente a cada um deles, trazendo as suas chaves primárias. Estes atributos passam a se chamar chaves estrangeiras, semelhante ao que foi explanado na entidade clientes.

Observamos agora que não teremos mais as **anomalias de alteração**, porque agora, por exemplo, se precisarmos mudar um nome de um produto, não precisaremos alterar nada nos registros de venda que utilizam este produto, uma vez que no registro da venda contém somente o código do produto. Se precisarmos alterar um endereço de um cliente, também em nada irá modificar o registro de venda correspondente. Caso precisemos excluir uma venda, não perderemos dado algum referente aos produtos, lojas, clientes, produtos ou departamentos.

## 8. O DIAGRAMA DE ENTIDADES E RELACIONAMENTOS

Notamos, então, que temos na nossa figura **entidades e relacionamentos** que representam o relacionamento “**1 para N**”. A nossa figura não está adotando nenhuma notação utilizada nas literaturas de banco de dados, mas utilizaremos os dados da figura para adotar uma ferramenta muito utilizada, chamada **Diagrama de entidades e relacionamentos**, que é um tipo de diagrama utilizado no projeto de banco de dados. Estes diagramas possuem símbolos e conectores que proporcionam uma visão de informações importantes, tais como as principais entidades dentro do escopo do sistema e as inter-relações entre essas entidades, ou seja, os relacionamentos entre elas.

**Figura 10: Entidades e relacionamentos.**

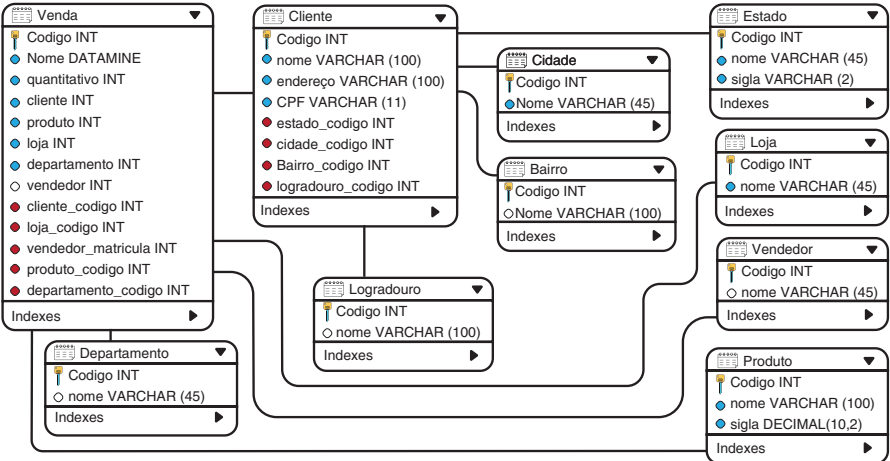


Fonte: Elaborada pelo autor.

Para construirmos o nosso diagrama de entidades e relacionamentos, utilizaremos o sistema gerenciador de banco de dados **MySQL** (SGBD MySQL), portanto, iremos necessitar da instalação do sistema gerenciador de banco de dados propriamente dito, que representa o lado do servidor, e de uma ferramenta para interagir com o SGBD, para podermos enviar comandos ao SGBD, que neste caso, constitui o lado do cliente (modelo cliente/servidor). A utilização desta ferramenta irá permitir que possamos construir os nossos bancos de dados, assim como gerar as tabelas e os respectivos diagramas. Ambos os softwares citados possuem versões que podem ser utilizadas gratuitamente, disponibilizadas pela **Oracle** no endereço: <https://dev.mysql.com/downloads/mysql/>. Para o desenvolvimento dos exemplos que serão discutidos neste livro, utilizamos o *MySQL Community Server 8.0.22* e o *MySQL WorkBench 8.0 Community*.

A seguir, representaremos o diagrama de entidade e relacionamentos correspondente à nossa figura anteriormente estudada (Figura 10), gerada pela ferramenta de construção de modelos do **MySQL Workbench**, a qual estudaremos o seu funcionamento em uma outra unidade. Posteriormente, iremos desenvolver passo a passo a criação do banco, das tabelas, dos diagramas e desenvolveremos os conceitos correspondentes referentes à linguagem SQL, tanto no que diz respeito à construção dos objetos, no caso o próprio banco de dados, suas tabelas, chaves primárias, chaves estrangeiras e outros componentes, como também referentes à manipulação dos dados, nas operações de inclusão, alteração e exclusão.

**Figura 11:** Diagrama de entidades e relacionamento no MySQL.



Fonte: Elaborada pelo autor.

De posse deste diagrama de entidades e relacionamentos, passaremos a discutir assuntos relacionados à **linguagem SQL** (*Structured Query Language*), que é uma linguagem de programação utilizada para permitir que enviemos comandos ao Sistema Gerenciador de Banco de Dados, no nosso caso utilizaremos o **MySQL**. Ela é uma linguagem padrão para os SGBDs relacionais, possui instruções que são usadas para executar tarefas como criar, alterar e excluir bancos de dados e seus objetos, incluir, alterar e excluir dados. Bancos de dados comerciais como o Oracle, Postgresql, MySQL, Sybase, Microsoft SQL Server, Access, Ingres etc.

Esta linguagem é amplamente utilizada atualmente, e será o assunto da próxima unidade.



## PRATIQUE

1) *O que é um Sistema Gerenciador de Banco de Dados?*

---

---

---

---

2) *O que é um banco de dados?*

---

---

---

---

3) *Defina dado, informação, conhecimento e sabedoria.*

---

---

---

---

4) *Defina entidade, atributo e relacionamento.*

---

---

---

5) Explique as cardinalidades do relacionamento de “1 para 1” e “1 para N”.

---

---

---

---

---

6) O que é uma chave primária e uma chave estrangeira?

---

---

---

---

---

7) Defina o termo normalização.

---

---

---

---

---

8) Enumere e explique as três formas normais mais utilizadas no projeto de banco de dados.

---

---

---

---

---

9) Quais os significados dos termos “string” e “alias”?

---

---

---

---

---

10) O que são metadados?

---

---

---

---

---



## RELEMBRE

*Nesta unidade, foram estudados assuntos referentes aos conceitos de Sistemas Gerenciadores de Banco de Dados (SGBD), ressaltando a importância da gerência e manutenção dos dados para as empresas.*

*Discutimos o ciclo de vida dos dados, como surge a necessidade de identificá-los, e falamos sobre a análise de requisitos que faz o levantamento dos dados para o projeto do banco de dados, sobre o modelo relacional que atualmente é o mais utilizado como base para os SGBDs comerciais.*

*Estudamos ainda assuntos relacionados ao projeto do banco de dados, discutindo sua estrutura e componentes, além das técnicas de normalização, e as formas normais mais utilizadas. Discutimos também os tipos de dados do SGBD MySQL, o dicionário de dados, além do diagrama de entidades e relacionamentos.*



## REFERÊNCIAS

GILLENZO, Mark I. **A fundamental of database management system**. Fogelman College of Business and Economics. University of Memphis, 1990.

MACHADO, Felipe; ABREU, Maurício. **Projeto de banco de dados: uma visão prática**. 11. ed. São Paulo: Érica, 2004.

SADALAGE, Pramod J.; FOWLER, Martin. **NoSQL Essencial: um guia conciso para o mundo emergente da persistência poliglota**. 1. ed. Rio de Janeiro: Novatec, 2013.

SAFER NET. **O que são os metadados?** Disponível em: <https://new.safernet.org.br/content/o-que-s%C3%A3o-os-metadados#>. Acesso em: 01 dez. 2020.

SILBERSCHATZ, Abraham. **Sistema de banco de dados**. Rio de Janeiro, 2006.

TEOREY, Tobey J. **Projeto e modelagem de banco de dados**. Rio de Janeiro. Elsevier, 2014.





## ANOTAÇÕES

48 | BANCO DE DADOS

[www.UniATENEU.edu.br](http://www.UniATENEU.edu.br)



**Núcleo de Educação a Distância**

Rua Coletor Antônio Gadelha, Nº 621  
Messejana, Fortaleza – CE  
CEP: 60871-170, Brasil  
Telefone (85) 3033.5199