Conceitos de banco de dados

Objetivos de aprendizagem:

* Identificar a importância da utilização do banco de dados no contexto organizacional.
* Reconhecer as principais ferramentas de administração de banco de dados.
* Relacionar os principais ambientes de banco de dados.

# 1. Introdução

Neste capítulo, você vai estudar a necessidade do uso de banco de dados no lugar de outra forma de guardar as informações, como, por exemplo, planilhas eletrônicas ou documentos de textos.  
Adicionalmente, você também conseguirá vislumbrar a diferença entre os sistemas de bancos de dados existentes atualmente, suas aplicações mais comumente utilizadas nas corporações em geral e, também, os aspectos positivos e negativos de cada opção de uso dos sistemas abordados.

# 2. Necessidade de uso dos bancos de dados

Imagine que você trabalha em uma pequena loja que tem como principal fonte de receita a comercialização de capinhas de celulares. A cada entrada de mercadoria no estoque ou baixa ao realizar uma venda, é registrada a respectiva ação em um caderno de registros. Tome por base que essa lojinha venda uma capinha de celular por dia: ao longo de um mês, haverá 30 registros de venda e de 2 a 3 registros de entrada de mercadoria. Bem fácil controlar na caderneta, certo?

Mas então o dono da lojinha recebeu uma herança e resolveu comprar um espaço no shopping mais movimentado da cidade para expandir os negócios. De 30 capinhas ao mês, agora a venda aumentou para três mil capinhas. O seu trabalho de registro na caderneta começou a ficar complexo e a tomar cada vez mais tempo do seu dia. Para resolver o problema, um sobrinho do dono da loja fez uma planilha eletrônica linda, cheia de fórmulas e que resolve mais rapidamente seu problema de cadastro.

Como um excelente homem de negócios, seu chefe resolveu abrir 20 filiais novas. Temos, então, 60 mil registros de venda de capinhas mais os registros de entrada de estoque por mês. Se colocarmos que a quantidade de linhas que uma planilha eletrônica suporta atualmente é cerca de um milhão, você iria enchê-la em menos de 18 meses (ou um ano e meio). Além disso, essa planilha ocuparia muito espaço no computador e você levaria muito tempo para recuperar as informações inseridas ali em cada momento de venda.

Então você se cansa de trabalhar nessa loja, entra em uma empresa de telefonia multinacional e se pega imaginando quantas planilhas eles possuem para controlar todas as ligações que seus clientes fazem a cada minuto. Segundo a ANATEL (BRASIL, 2018), a quantidade de linhas telefônicas fixas no Brasil já passou a marca de 40 milhões e meio. Coloque nessa equação a quantidade de linhas telefônicas móveis (celulares) e temos o caos das planilhas eletrônicas.

A solução para qualquer um dos casos apresentados, desde a lojinha que vendia 30 capinhas de celular por mês até a grande companhia telefônica com milhões de novos registros que precisam ser processados a cada segundo, é uma só: banco de dados.

Da mesma forma, a atividade pertinente a qualquer área de uma organização requer um poder grande de armazenamento de informações, ordenação, busca e edição. Pense em um sistema on-line popular qualquer, como, por exemplo, o Facebook. Inúmeros bancos de dados são utilizados para prover a plataforma da rede social, cada qual com seu objetivo: um banco de dados somente para o cadastro dos usuários, outro banco de dados para as postagens dos usuários, um terceiro banco de dados para armazenar as fotos dos usuários, mais outro banco de dados para o feed de notícias, um quinto banco de dados para o chat da plataforma, etc.

Cada sistema tem sua própria divisão de banco de dados, mas você pode ter certeza de que é necessário um ou mais banco de dados para suportar um blog, por mais simplista que possa parecer — até mesmo para grandes plataformas, como Facebook, LinkedIn, Instagram, WhatsApp, Twitter, sistemas bancários e demais sistemas.

Quando você efetua a compra de um produto qualquer para o qual é emitida uma nota fiscal eletrônica, o fluxo de funcionamento daquele ponto de venda (PDV) com a ação do colaborador do estabelecimento ao registrar seu CPF na nota.

Sendo assim, com a utilização de banco de dados, tanto o cadastro de novos registros, como a venda de uma capinha de celular do exemplo anterior, quanto uma nova ligação por parte do cliente da grande companhia telefônica resulta em um novo registro no próprio banco de dados — portanto, uma nova linha da “planilha”. Como os bancos de dados, em sua maioria, estão conectados aos sistemas, esse registro é feito de forma automática e é garantida a integridade da informação.

# 3. Como administrar um banco de dados

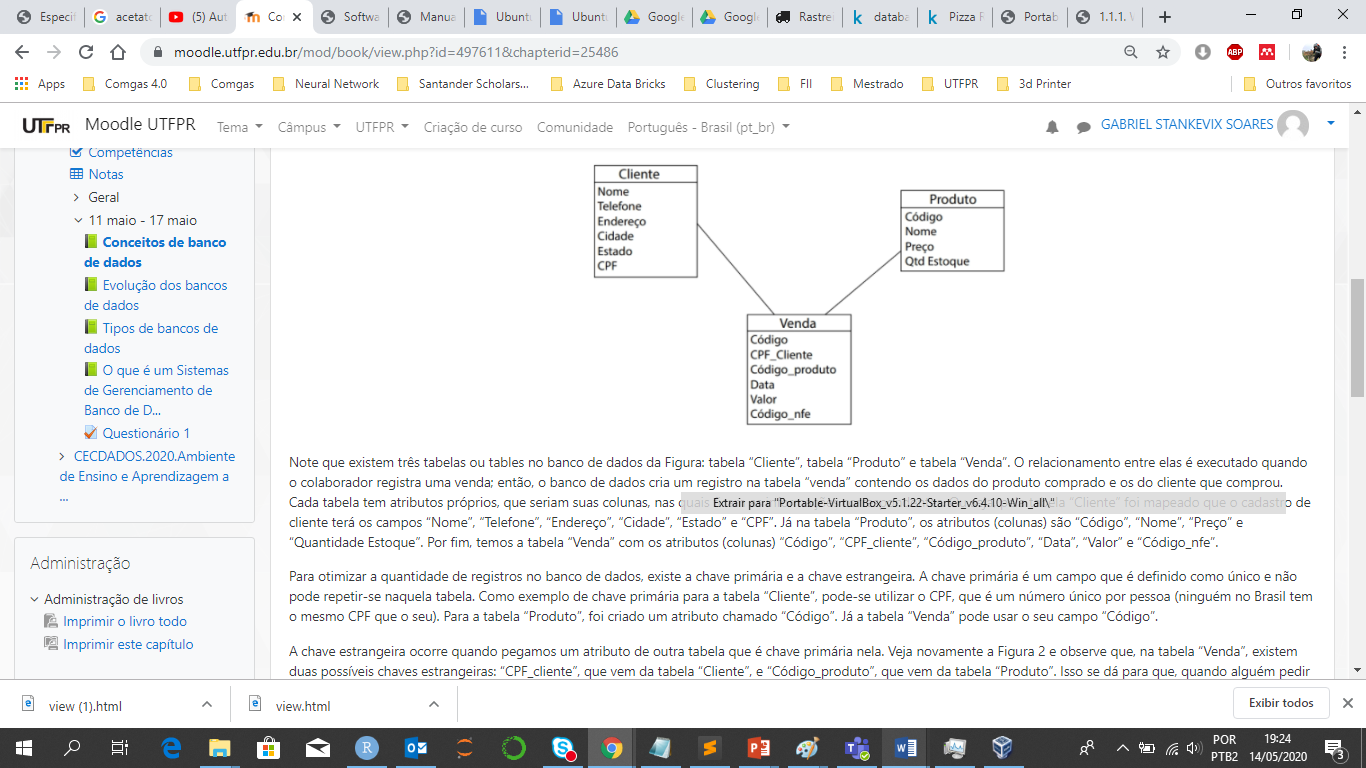
A questão da administração de um banco de dados é ampla e complexa, sendo debatida por diversos autores. Em resumo, você deve ter conhecimento de que tipo de informação está alocada nas tabelas do seu banco de dados e, principalmente, como manipular as informações que ali estão inseridas da forma que necessita.

Um sistema de banco de dados é formado pelo software de banco de dados, o repositório de tabelas denominado database e as próprias tabelas em si ou tables. Dessa forma, um banco de dados para um sistema de agenda telefônica teria o serviço do banco de dados executado, uma database e, dentro desta, estariam as tabelas para gravar as informações de nome, endereço, contato telefônico, e-mail, aniversário, foto, etc.

Portanto, para você conseguir realizar uma boa administração do banco de dados, é necessário conhecer o negócio ou o projeto ao qual ele servirá. De nada adianta ter tabelas que não são necessárias, pois isso impacta na performance do serviço. No exemplo da agenda telefônica, se não existisse o campo “contato telefônico”, o sistema ficaria inútil, pois uma agenda tele-fônica precisa, obrigatoriamente, conter o campo para cadastro do telefone do contato.

Como passo fundamental na administração correta de um banco de dados, é essencial que você tenha a documentação base do sistema que utilizará do banco de dados sob sua administração. Esse documento contemplará a modelagem do banco de dados, os relacionamentos entre tabelas, como será gravada cada informação, quais campos são obrigatórios e quais são opcionais, a necessidade de indexação para melhorar a velocidade de consulta, etc.

A seguir, na Figura, você verá um exemplo simples de uma modelagem de banco de dados, na forma de entidade-relacionamento, para compreensão. Essa forma de ilustração do banco de dados é a mais comumente utilizada em documentações e projetos. Nela, cada retângulo representa uma tabela, cujo título é o seu nome. Para cada tabela, há uma relação de atributos (linhas abaixo do título) e linhas ligando essas tabelas, que representam os relacionamentos entre elas. Por ora, você precisa atentar para o nome e cada atributo das tabelas a seguir, pois iremos estudar esse e outros modelos de banco de dados mais adiante.

 Note que existem três tabelas ou tables no banco de dados da Figura: tabela “Cliente”, tabela “Produto” e tabela “Venda”. O relacionamento entre elas é executado quando o colaborador registra uma venda; então, o banco de dados cria um registro na tabela “venda” contendo os dados do produto comprado e os do cliente que comprou. Cada tabela tem atributos próprios, que seriam suas colunas, nas quais ficam as informações correspondentes. Ou seja, pela tabela “Cliente” foi mapeado que o cadastro de cliente terá os campos “Nome”, “Telefone”, “Endereço”, “Cidade”, “Estado” e “CPF”. Já na tabela “Produto”, os atributos (colunas) são “Código”, “Nome”, “Preço” e “Quantidade Estoque”. Por fim, temos a tabela “Venda” com os atributos (colunas) “Código”, “CPF\_cliente”, “Código\_produto”, “Data”, “Valor” e “Código\_nfe”.

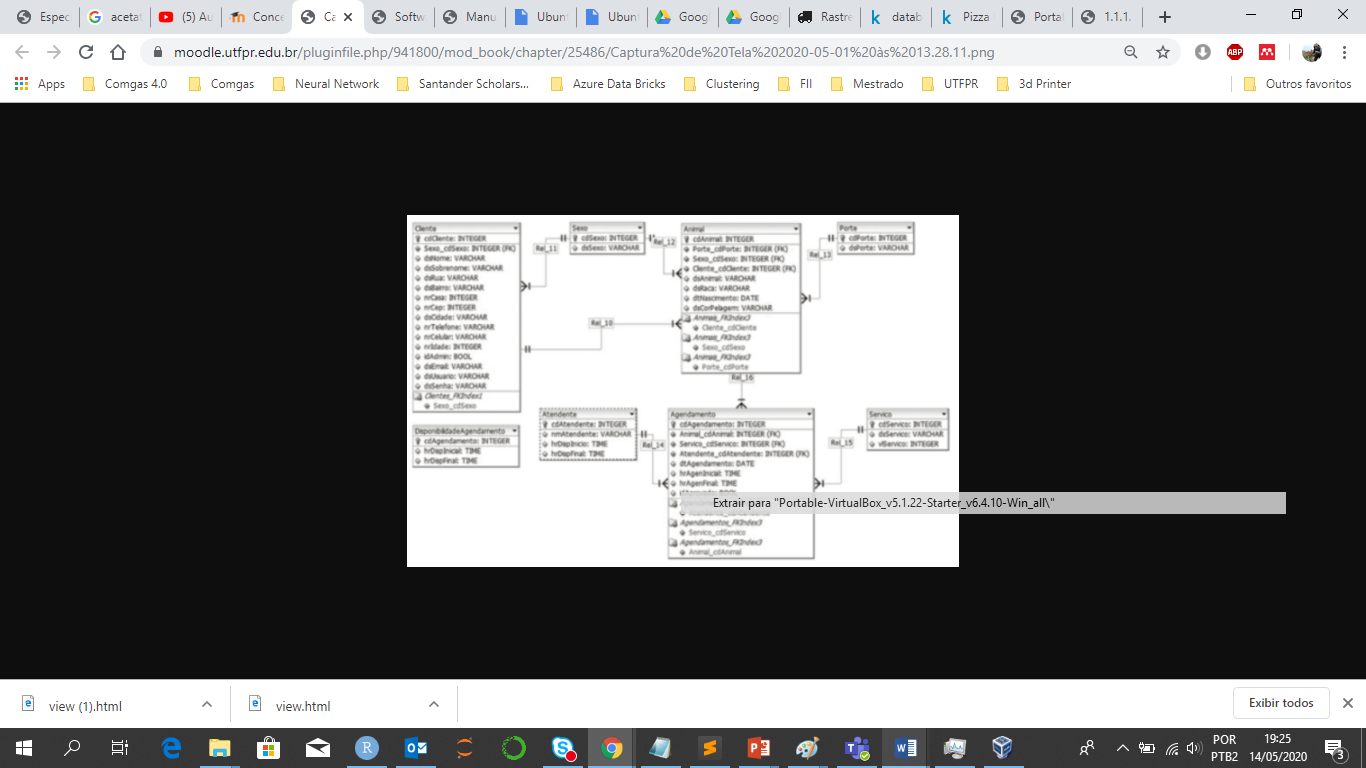
Para otimizar a quantidade de registros no banco de dados, existe a chave primária e a chave estrangeira. A chave primária é um campo que é definido como único e não pode repetir-se naquela tabela. Como exemplo de chave primária para a tabela “Cliente”, pode-se utilizar o CPF, que é um número único por pessoa (ninguém no Brasil tem o mesmo CPF que o seu). Para a tabela “Produto”, foi criado um atributo chamado “Código”. Já a tabela “Venda” pode usar o seu campo “Código”.

A chave estrangeira ocorre quando pegamos um atributo de outra tabela que é chave primária nela. Veja novamente a Figura 2 e observe que, na tabela “Venda”, existem duas possíveis chaves estrangeiras: “CPF\_cliente”, que vem da tabela “Cliente”, e “Código\_produto”, que vem da tabela “Produto”. Isso se dá para que, quando alguém pedir relatório de vendas, ao consultar a tabela “Venda”, o sistema consiga buscar, por meio do campo “CPF\_cliente”, os dados do cliente na tabela “Cliente” e os dados do produto por meio do campo “Código\_produto” na tabela “Produto”.

A dúvida que surge é a de por que realizar esses relacionamentos e não simplesmente criar mais atributos na tabela “Venda” para comportar os dados do cliente e dos produtos nessa própria tabela. A resposta é bem simples: em bancos de dados, assim como em qualquer sistema tecnológico, os dados têm complexidade e, quanto mais dados, maior a complexidade para processá--los. Ao registrar apenas o código do produto e o CPF do cliente e, com isso, realizar a busca nas tabelas correspondentes de “Cliente” e “Produto”, você está otimizando recursos computacionais e mantendo o banco de dados totalmente escalável.

Imagine que podem existir milhões de vendas diárias, de maneira que você pouparia muito processamento ao identificar as vendas apenas com os códigos das demais tabelas e, caso necessário, poderia recuperar a informação dessas tabelas em um relatório para o gerente da loja, por exemplo. Outra vantagem é que, ao fazer uma atualização cadastral de um cliente (telefone, por exemplo), você teria que sair varrendo a tabela de “Venda” também, caso ela tivesse os campos replicados da tabela “Cliente”.

Geralmente, a modelagem de banco de dados ocorre em times, como forma de ser o mais acertado possível, visto que um retrabalho em cima da modelagem é muito mais oneroso do que se já estiver correto no início do projeto. Quando um sistema se torna grande, essa modelagem apresentará centenas de tabelas e milhares de atributos e relacionamentos entre elas. A seguir, na Figura, observe o exemplo de um sistema de clínica veterinária com maior número de tabelas, atributos e relacionamentos:



Observe que, na Figura, aparecem novas informações após o nome do campo, como varchar, integer, date, time, bool, etc. Essas informações se referem ao tipo de dado que a coluna relacionada poderá receber. Então, seguindo esse raciocínio, a tabela “Animal”, no atributo “dtNascimento”, que é do tipo DATE, só poderá receber o valor de data válido. Qualquer coisa diferente disso será descartada pelo próprio banco de dados, que não permitirá a inserção. Outro exemplo: na tabela “Cliente”, o atributo “nrCEP” é do tipo INTEGER, de modo que só aceitará caracteres do tipo inteiro e, novamente, caso a entrada seja diferente de 0 até 9, o próprio banco de dados proíbe a entrada.

Pode-se concluir, então, uma relação de tipos de atributos mais comumente utilizados nos bancos de dados atualmente:

* INT ou INTEGER: recebe apenas números inteiros. Exemplo: 45002.
* VARCHAR: recebe caracteres alfanuméricos. Exemplo: João da Silva.
* DATE: recebe apenas datas válidas. Exemplo: 01/01/2018.
* TIME: recebe apenas valores expressos em hora. Exemplo: 16:54.
* BOOL ou BOOLEAN: recebe apenas o número 0 ou o número 1. Serve para indicar se algo é verdadeiro ou falso, positivo ou negativo, etc. Exemplo: 0 (o sistema sabe que, se esse campo for 0, o cliente está verdadeiro para compra a prazo).

# 4. Exemplos de sistemas de banco de dados e seus usos na prática

Com toda essa introdução sobre os sistemas de banco de dados, vamos abordar, agora, quais são os [tipos de bancos de dados](https://moodle.utfpr.edu.br/mod/book/view.php?id=497619) disponíveis e suas características e usos pelos diversos tipos de aplicações e corporações.

Nos dias atuais, quatro sistemas de banco de dados dominam o cenário geral e um quinto está em franco crescimento. São eles:

* MySQL;
* MariaDB;
* Microsoft SQLServer;
* PostgreSQL;
* Oracle;
* MongoDB (franca expansão).

## 4.1. Linguagem SQL

Para realizar a inserção, remoção, edição ou qualquer outra atividade com os dados que irão ou estão no banco de dados, é necessária uma interação com o mesmo. Dessa forma, a linguagem SQL (Structure Query Language ou Linguagem de Consulta Estruturada) é a que intermedeia essa interação, criando a ponte necessária entre o que precisa ser feito e o banco de dados efetivamente. Assim, para que algo aconteça, é necessário que a linguagem SQL esteja com a sintaxe correta e os valores desejados também corretos.

Aplicada para gerenciar os bancos de dados do tipo MariaDB (e MySQL), Microsoft SQLServer, PostgreSQL e Oracle, é a linguagem comumente usada por database administrators diariamente em suas jornadas de trabalho. Perceba que o MongoDB, quinto item da lista anterior e marcado como em franca expansão, não utiliza a linguagem SQL para criar suas consultas e manipular os dados necessários. Isso se dá pela forma com que ele organiza suas informações e também como disponibiliza as mesmas de volta ao sistema quando este solicita uma consulta, por exemplo. Assim, ele consegue desempenhar uma velocidade incrível de resposta a consultas se comparado aos bancos de dados mais tradicionais que utilizam a linguagem de consulta estruturada (SQL). Portanto, o MongoDB é um banco de dados do tipo não relacional.

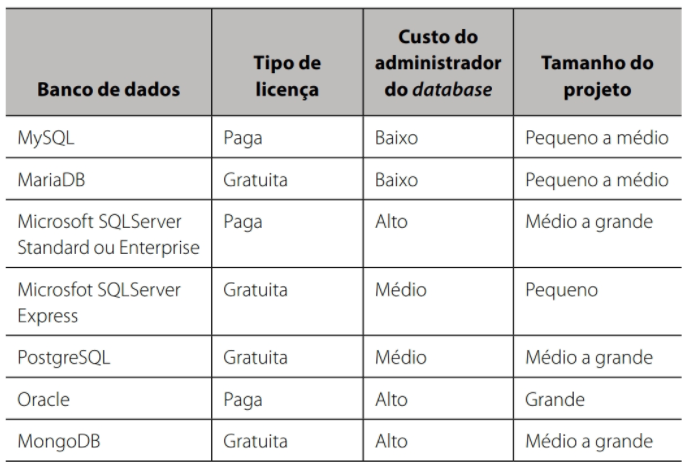
Segundo Elmasri e Navathe (2007), a linguagem SQL nasceu unicamente para ser a linguagem de consulta do sistema System R da IBM. A evolução desta empresa originou a linguagem SQL tão difundida atualmente.

Uma resposta para por que a linguagem SQL é tão utilizada e difundida em todo o mundo pode ser compreendida como:

"A linguagem possui muitos recursos, tanto para manipulação de dados (DML, Data Manipulation Language) como para a definição de dados (DDL, Data Definition Language). Por conta disso, SQL é a linguagem mais utilizada em sistemas de bancos de dados" (KOMATSU, 2011, p. 20).

## 4.2. Tipos de bancos de dados

A grande maioria dos projetos envolvendo bancos de dados toma por base a questão custo x benefício, ainda mais nos tempos atuais, em que as corporações necessitam de diminuição constante dos valores despendidos em qualquer área. Assim, de início, dividimos a lista dos bancos de dados em duas grandes frentes: os bancos de dados que precisam de uma licença paga e os bancos de dados que não necessitam de uma licença de uso paga.



Com base nas informações apresentadas no Quadro, podemos refletir sobre a necessidade do projeto que precisa ser desenvolvido diante da realidade de custos sucintamente apresentados nessa mesma tabela informativa. Quanto maior o projeto, mais robusto precisa ser o banco de dados e, por consequência, mais cara será a cobrança por parte de quem o administra.