**Papel do bancos de dados Relacionais (SQL)**

Um banco de dados relacional é um tipo de banco de dados — sim, existem outros tipos de banco, como os não relacionais, mas não nos aprofundaremos neste assunto no momento. Logo, os bancos de dados relacionais **são feitos com base no modelo relacional de organização para representar e organizar dados em tabelas**.

Nesse sentido, podemos ter acesso a diferentes tipos de dados, como os dados financeiros de uma empresa, as informações de pacientes em um hospital, estudantes de uma universidade etc.

Além disso, quando nos referimos a essa estrutura do modelo relacional, nesse tipo de banco de dados **cada linha na tabela representará uma ID única, que chamaremos de chave**. Somado-se a isso, vale ressaltar outras características que fazem parte do modelo de banco de dados relacional como as colunas, as quais contêm atributos de cada dado e podem também ter um valor.

Tudo isso auxilia na construção de relações que interligam a estrutura de dados em um banco. Mas, talvez você esteja se perguntando qual a utilidade de um banco de dados relacional.

Quando temos informações, para podermos entender de forma eficiente, é necessário organizá-las. Dessa forma, um banco de dados relacional **é importante à medida que surge a necessidade de organização dos seus dados**. Além disso, também permite a otimização do trabalho, uma vez que diversas pessoas podem acessar simultaneamente e há uma maior segurança no processo de armazenamento.

Grande parte dos bancos de dados relacionais utilizam a linguagem de programação estruturada especializada para sistemas de banco de dados, o [SQL](https://blog.betrybe.com/sql/).

Para construir um banco de dados temos diferentes tipos de objetivos que fazem parte. Entre eles:

**Tabela:** é responsável por armazenar os dados, como por exemplo o nome e a idade das pessoas estudantes do curso de Ciência da Computação de determinada universidade. A tabela também pode ser chamada de relação, uma vez que um banco de dados relacional pode ter uma ou mais relações.

**Coluna:**é responsável por armazenar apenas um tipo específico de dado, podendo ele ser nulo ou não. É importante lembrar que pode ser uma coluna não-chave, ou seja, o tipo de dado pode se repetir em outras colunas.

**Tupla:** é responsável por representar cada registro de um determinado dado. O registro é identificado por uma chave primária, uma vez que não é possível ter registros duplicados.

**Relacionamento:** atente-se a essa parte, a relação é diferente de relacionamento. Uma vez que a Relação representará a tabela, o relacionamento será a associação entre elas, visto que estão conectadas por chaves primárias e por chaves estrangeiras.

**Chave Primária:** do inglês “primary key” e apelidado carinhosamente como “pk”, a chave primária é um atributo (ou coluna) que identifica um registro de forma única. A exemplo disso, temos o CPF de terminada pessoa estudante.

**Chave Estrangeira:** também chamada de “foreign key” ou “fk”, é um atributo que designará a forma que as tabelas se relacionam entre si. Ou seja, uma fk faz relação a uma pk. A exemplo disso, podemos relacionar a tabela “gostos musicais”, por meio de uma chave estrangeira, que fará o relacionamento com a chave primária na relação de estudantes.

Um SGBDRs é um sistema que **armazena, organiza, recupera, consulta, gerencia e recupera dados de maneira organizada**. O SGBDRs possui uma interface gráfica que auxilia as pessoas usuárias a utilizarem os seus recursos de maneira um pouco mais funcional, uma vez que também ajuda na execução de tarefas de gerenciamento administrativas e no seu desempenho.

Nesse sentido, é essencial ter um banco de dados seguro e que atenda as necessidades do seu problema. Aqui listamos alguns pontos que poderão auxiliá-lo nesse processo de escolha:

* **Necessidades de desempenho e confiabilidade**: já falamos sobre a importância de ter um sistema seguro nesse mesmo tópico, logo, a confiabilidade e proteção dos nossos dados torna-se essencial nesse processo de decisão. Além disso, o desempenho desse sistema precisará suportar grandes volumes de dados para que funcione de maneira rápida e eficaz? Então, esse será um ponto relevante nessa tomada de decisão.
* **Simultaneidade**: se esse banco for utilizado por um grande número de usuários e conectado a diferentes aplicativos é importante saber quanto aquele sistema é capaz de suportar para que não atrapalhe no desempenho do seu produto.
* **Escalabilidade**: quando nos referimos a escalabilidade precisamos entender sobre a base de dados que estamos lidando, caso seja um volume muito grande, existirá, certamente um produto que nos auxilie melhor nessa questão.

**Quais as principais funcionalidades dos SGBDRs?**

Nos SGBDRs, existe uma linguagem própria para a manipulação dessa gama de dados existente. Entre elas, existe o [CRUD](https://blog.betrybe.com/tecnologia/crud-operacoes-basicas/), sigla inglesa para **CREATE, READ, UPDATE E DELETE**. Tal qual é uma funcionalidade básica da linguagem. Agora, vamos criar uma implementação básica para que você possa assimilar como isso funcionaria na prática:

Vamos supor que Maria tem uma mercearia e quer inserir os novos produtos que chegaram para a venda em seu banco de dados.

**Papel do banco de de dados não Relacionais (NoSQL)**

A seguir, apresentamos os principais bancos de dados NoSQL utilizados em todo o mundo.

Redis

O Redis é o banco de dados NoSQL de chave-valor mais utilizado em todo o mundo. Ele vincula um valor a uma chave na sua estrutura, o que facilita o armazenamento e a busca desses dados. Por isso, é muito utilizado pelos desenvolvedores.

Memcached

Este banco de dados não relacional também faz o armazenamento com chave-valor e usa um cache de memória distribuída. Geralmente, é utilizado para criar sites dinâmicos, pois acelera a abertura das páginas e diminui as buscas de dados de fontes externas.

Cassandra

Este banco de dados NoSQL foi desenvolvido no Facebook. Ele usa um banco de dados descentralizado, em que os dados são armazenados em vários datacenters. Ele é otimizado para cluster e fornece baixa latência em suas atualizações.

Hbase

O Hbase é um banco de dados que utiliza conjunto de linhas e colunas para armazenar as informações. Ele é utilizado em diferentes plataformas como o LinkedIn, Facebook e Spotify

Amazon DynamoDB

Este é um banco de dados NoSQL em nuvem, disponibilizado pela [Amazon Web Service](https://blog.saphir.com.br/conheca-os-principais-servicos-da-aws-amazon-web-services/). Ele tem baixa latência, é rápido e flexível, sendo o modelo ideal para aplicações móveis, jogos na web e soluções com internet das coisas.

Ele ainda apresenta alto desempenho e [escalabilidade automática](https://blog.saphir.com.br/o-que-e-escalabilidade-em-cloud-computing/), características imprescindíveis para negócios que precisam crescer com eficiência.

Neo4j

O Neo4j é um banco de dados não-relacional que se baseia em grafos (arestas que se relacionam aos nodes). Ele é uma implementação de código aberto e pode ser útil para casos de mineração de dados e reconhecimento de padrões.

MongoDB

Este também é um banco de dados de código aberto com alta performance. Ele é aceito em diferentes sistemas operacionais e tem como característica ser orientado a documentos.

Sendo assim, ele armazena todas as informações relevantes em um documento e utiliza sistemas avançados de agrupamento e filtragem. Diferentes plataformas e linguagens possuem suporte ao MongoDB, entre elas estão o Java, JavaScript, PHP, Python e Ruby.

Os principais exemplos de empresas que usam o MongoDB são: o site Globo.com, MailBox, MTV e Pearson Education.

Esses são os principais exemplos de bancos de dados NoSQL ou não-relacionais. O uso entre eles pode se diferenciar de acordo com as necessidades de cada negócio. Portanto, o mais indicado é buscar o auxílio de uma empresa especializada em [soluções de armazenamento de bancos de dados](https://saphir.com.br/).

**Diferença entre banco relacional e não relacional**

Já definimos ao longo de todo texto do que se trata um banco de dados relacional, ou seja, ele basicamente é responsável por armazenar e organizar fontes de dados relacionando-as entre si. Mas, agora falaremos do que se trata um banco de dados não relacional.

Nesse sentido, quando nos referimos a um banco de dados não relacional estamos falando **de qualquer banco que não segue o padrão SGBDR, logo, são conhecidos por fazerem parte do sistema NoSQL**. Entre eles, existem os mais populares, como o [MongoDB](https://blog.betrybe.com/dados/mongodb-tudo-sobre/), Cassandra e o Redis.

Para escolher o melhor banco de dados, seja relacional ou não relacional, precisamos definir qual a finalidade e o tipo de dado com o qual estamos trabalhando, uma vez que isso determinará a usabilidade dos dados. Dessa forma, caso você por exemplo, esteja trabalhando com Big Data e fontes de dados do tipo JSON, o ideal será utilizar um banco de dados não relacional, uma vez que ele foi desenvolvido para manipular esse tipo de dado.