**Práticas de Banco de Dados**

Práticas de Banco de Dados

1.00

**PRÁTICAS DE BANCO DE DADOS**

PRÁTICAS DE BANCO DE DADOS

Autor: Me. Paulo Sérgio Pádua de Lacerda

Revisor: Paulo Lacerda

[INICIAR](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html#section_1)

**introdução**

Introdução

Caro(a) aluno(a),

Seja bem-vindo(a) à disciplina de Prática de banco de dados. A proposta principal desta unidade é definir e estimular os conhecimentos fundamentais sobre um Sistema Gerenciador de Banco de Dados (SGBD). Um SGBD tem a capacidade de controlar diversos banco de dado e permitir que operações, via uma interface, possam ser executadas via comandos por um usuário. Capacitar-se nos fundamentos de um SGBD possibilita a melhoria da sua habilidade, tanto de planejar quanto de desenvolver sistemas. Temas como arquitetura, catálogo do sistema, segurança e otimização também são explorados nesta unidade. Em suma, você vai identificar e compreender os princípios de Sistema Gerenciador de Banco de dados. Vamos lá?

**Com são as *Engines* do Banco de Dados?**

Em geral, o sistema gerenciador de banco de dados (SGBD) é um *software* que controla banco de dados, e, por sua vez, é uma estrutura computacional que compartilha e integra dados. Um sistema de banco de dados é composto de um conjunto de metadados, que armazena informações sobre os elementos por meio dos quais dados do usuário são integrados e gerenciados (ROB; CORONEL, 2011).

Um mecanismo de banco de dados é subsistema que o banco de dados utiliza para funcionar ou gerenciar as operações, ou seja, é a estrutura fundamental de operação de um SGBD. Uma *engine*(mecanismo) é a parte do sistema gerenciador de banco de dados que realmente armazena e recupera dados. Na maioria das vezes, o sistema possui uma interface de programação (API) que permite o controle direto do motor, sem a interação com a interface do usuário do SGBD.

Vamos a um exemplo de uso e diferença de mecanismo de banco de dados usado no MySQL. O sistema MySQL é considerado um dos mais populares gerenciadores, muito utilizado para sistema web, *open source* (código aberto) é distribuído e apoiado pela Oracle Corporation (MYSQL 8.0..., 2020).

O  MySQL possui algumas características que determinam a predominância do seu uso por parte de desenvolvedores:

* são bancos de dados relacionais;
* são bancos de dados de fácil uso, rápidos, escalável e confiável;
* opera numa arquitetura cliente-servidor e embarcados;
* possui uma grande variedade de distribuições.

Dentre outras características, possui dois tipos diferentes de mecanismo internos: o InnoDB e o MyISAM. O InnoDB é o mecanismo padrão do MySQL, e, por padrão, tem suporte a operações de transação e a chave estrangeira. O Quadro 1.1 resume os recursos existentes no mecanismo InnoDB.

|  |  |
| --- | --- |
| **Recursos** | **Suporte** |
| Indexação por algoritmo B-Tree | Sim |
| Backup | Sim |
| Suporte a cluster | No |
| Indexação de Clusters | Sim |
| Compressão de dados | Sim |
| Caches | Sim |
| Criptografia dados criptografados | Sim |
| Suporte à chave estrangeira | Sim |
| Índices de pesquisa de texto completo | Sim |
| Suporte geoespacial do tipo de dados | Sim |
| Suporte à indexação geoespacial | Sim |
| ÍndIces de Hash | Não |
| Caches de índice | Sim |
| Travando granularidade | Linha |
| Suporte de replicação (implementado no servidor, em vez de no motor de armazenamento). | Sim |
| Limites de armazenamento | 64TB |
| Indexação T-tree i | Não |
| Transações | Sim |
| Atualizar estatísticas para dicionário de dados | Sim |

Quadro 1.1 - InnoDB recursos  
Fonte: MySQL 8.0... (2020,*on-line*).

Pode-se notar, pelo Quadro 1.1, a capacidade do sistema de armazenamento de dados, cerca de 64T bytes, o uso de criptografia, índices etc.

Já o mecanismo MyISAM não é padrão e originário do antigo mecanismo ISAM, mas também permite diversos recursos. O Quadro 1.2 demonstra alguns recursos do mecanismo MyISAM.

|  |  |
| --- | --- |
| **Recurso** | **Suporte** |
| indexação B-tree | Sim |
| Backup | Sim |
| Suporte de banco de dados cluster | Não |
| Índices agrupados Clustered | Não |
| Dados comprimidos | Sim (Yes) |
| Cache de dados | Não (No) |
| Dados criptografados | Sim |
| Suporte à chave estrangeira | Não |
| Índices de pesquisa de texto completo | Sim |
| Suporte geoespacial do tipo de dados | Sim |
| Suporte à indexação geoespacial | Sim |
| Índices de *hash* | Não |
| Caches de índice | Sim |
| Travando granularidade | Tabela |
| Suporte de replicação (implementado no servidor, em vez de no motor de armazenamento). | Sim |
| Limite | 256TB |
| Indexação T-Tree | Não (No) |
| Transações | Não |
| Atualizar estatísticas para dicionário de dados | Sim |

Quadro  1.2 - MyISAM recursos  
Fonte: MySQL 8.0 (2020, *on-line*).

Observa-se, no Quadro 1.2, que o mecanismo MyISAM permite um armazenamento bem superior ao do InnoDB: cerca de 256TB. Porém, não permite transações e não suporta o uso de chave estrangeiras (*foreign key*).

Outra diferença entre esses dois mecanismos é que em razão de InnoDB oferecer suporte a transações, comandos de *commit* e*rollback*podem ser executados, de maneira que o mecanismo MyISAM, por ser mais simples, é o mais adotado para iniciantes de banco de dados. Ao contrário do InnoDB, o uso de mecanismo MyISAM deve ser declarado na construção da tabela, como no exemplo: CREATE TABLE t (i INT) ENGINE = MYISAM.

**saiba mais**

Saiba mais

A administração de um banco de dados é, muitas vezes, menosprezada, e fica a cargo do DBA (Database Administrator), ou **administrador**de banco de dados, a execução da tarefa. Porém, saber sobre banco de dados é entender como funciona o processo de permissões por meio de comandos de controle DCL (Data Control Language). No link da documentação MySQL, há muitas explicações e exemplos de como usar mais profundamente esses comandos. Acesse:

[ACESSAR](https://oracle-base.com/articles/mysql/mysql-users-and-permisisons)

Além das *engines* de banco de dados, catálogos são fundamentais na estrutura das informações armazenadas e considerados outro recurso importante, sendo esse o tema do nosso próximo tópico de estudo.

**praticar**

Vamos Praticar

Todo banco de dados possui uma*engine*. Essa *engine*é o subsistema que armazena informações sobre todos os elementos de um sistema gerenciador de banco de dados. O MySQL utiliza dois mecanismos: o InnoDB e o MyISAM. Sobre o tema mecanismo de banco de dados e com base no texto anterior, analise as afirmações a seguir:

1. O mecanismo padrão do MySQL é o InnoDB.
2. MyISAM é mais rápido, porém possui menor capacidade.
3. InnoDB possui suporte a transações.
4. MyISAM permite uso de algoritmo T-tree.
5. InnoDB com relação a MyISAM permite relacionamento entre entidades.

A partir das afirmativas apresentadas anteriormente, assinale a alternativa correta:

Parte superior do formulário

**a)**Apenas as afirmativas II, III, IV e V estão corretas.

**b)**Apenas as afirmativas I, II e V estão corretas.

**c)**Apenas as afirmativas III, IV e V estão corretas.

**d)**Apenas as afirmativas I, III e V estão corretas.

**e)**As afirmativas I, II, III, IV e V estão corretas.

Parte inferior do formulário

**Catálogo de Banco de Dados**

Um sistema gerenciador de banco de dados (SGBD) não lida apenas com um banco de dados, mas com um conjunto de banco de dados. Sendo assim, uma das características é o uso de catálogos. Esses catálogos são responsáveis por manter uma descrição das estruturas e restrições dos bancos de dados por meio do uso de metadados, permitindo que o mesmo SGBD possa ser utilizado por diversas aplicações.

As funções básicas de um SGBD estão relacionadas à definição de um banco de dados, o que significa especificar tipos, estruturas e restrições dos dados a serem armazenados. Esse processo é feito por meio do catálogo, ou dicionário de dados, também conhecido como metadados. Também são funções do SGBD a construção de banco de dados, sua manipulação e compartilhamento (VICCI, 2014).

**Modelos de Dados**

Um SGBD é uma ferramenta interessante do ponto de vista do usuário, pois é ele que cria diferentes níveis de abstração de dados. Embora o processo interno de operações de um SGBD relativo a armazenamento e gerenciamento dos dados armazenados seja complexo, esse processo é transparente para o usuário. Não precisa ter conhecimento sobre esse funcionamento interno, nem estar familiarizado com sua complexidade.

Um modelo de dados pode ser definido como um tipo de abstração de dados, em que o usuário não tem muita informação sobre como são armazenados internamente no SGBD. Também se refere a um conjunto de conceitos que descreve tanto a estrutura lógica como física de um banco de dados (ELMASRI; NAVATHE, 2011).

O Quadro 1.3 apresenta um formato de armazenamento interno para um registro, em que se observa uma tabela com informações de como os dados são armazenados internamente no banco de dados.

|  |  |  |
| --- | --- | --- |
| **Nome do item de dados** | **Posicionamento inicial** | **Tamanho em caracteres** |
| id\_turma | 1 | 4 |
| num\_disciplina | 2 | 10 |
| Semestre | 3 | 10 |
| Ano | 4 | 2 |
| Professor | 5 | 30 |

Quadro 1.3 - Quadro de registro  
Fonte: Adaptado de Elmasri e Navathe (2011).

**reflita**

Reflita

Embora novos sistemas gerenciadores de banco de dados estejam surgindo no mercado, os sistemas transicionais existentes ainda são baseados em modelos relacionais com dados estruturados. Os sistemas de controle bancário das instituições financeiras são baseados em SGBD relacionais.

Ao analisar o Quadro 1.3, nota-se que este pode não fazer nenhum sentido na visão de um usuário, pois não tem conhecimento de posicionamento de registros ou mesmo como a unidade de medida bytes é usada para determinar o tamanho de armazenamento.

Ao contrário do quadro anterior, o Quadro 1.4 demonstra um quadro em que a organização dos dados se apresenta em dois eixos: vertical, cujas “colunas” representam os campos, e horizontal, em que as “linhas” representam o registro completo, que, ao final, resume nos dados de informação do usuário.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id\_turma** | **num\_disciplina** | **semestre** | **ano** | **professor** |
| 86 | MAT2410 | Segundo | 7 | Carlos |
| 87 | CC1310 | Segundo | 7 | Pedro |
| 102 | CC3320 | Primeiro | 8 | Ana |
| 112 | MAT1240 | Segundo | 8 | Santos |
| 120 | CC3310 | Segundo | 8 | Igor |

Quadro 1.4 - Quadro de dados  
Fonte: Adaptado de Elmasri e Navathe (2011).

Nota-se pelo Quadro 1.4 que ele se aproxima de um formato com legibilidade para o usuário. Faz sentido a coluna (campos) e as linhas (registros). Numa simples leitura, qualquer usuário com pouco conhecimento em banco de dados é capaz de compreender.

Os modelos de dados associados a um gerenciador de banco de dados podem ser de alto ou baixo nível. O de alto nível corresponde àquele que fornece uma visão dos dados o mais próximo possível da forma como os usuários visualizam e compreendem os dados, o que significa que estes devem ser apresentados o mais próximo possível de sua linguagem de compreensão. Já os modelos de baixo nível correspondem à visão mais detalhada de como os dados realmente são armazenados pelo SGBD, que, em geral, não é tão intuitiva ao usuário.

**Esquemas e Instâncias**

O conceito de esquemas e instâncias se refere à distinção que deve ser feita entre a descrição do banco de dados e dos dados em si que são armazenados e administrados pelo gerenciador de banco de dados. O esquema de um banco de dados se refere a sua descrição e é especificado durante a fase de projeto do banco de dados (ELMASRI; NAVATHE, 2011).

Uma vez definido, esse esquema tem poucas alterações. Cada SGBD possui uma linguagem própria de como definir e apresentar esse esquema, entretanto, podem ser utilizados diagramas para representá-los, facilitando a compreensão.

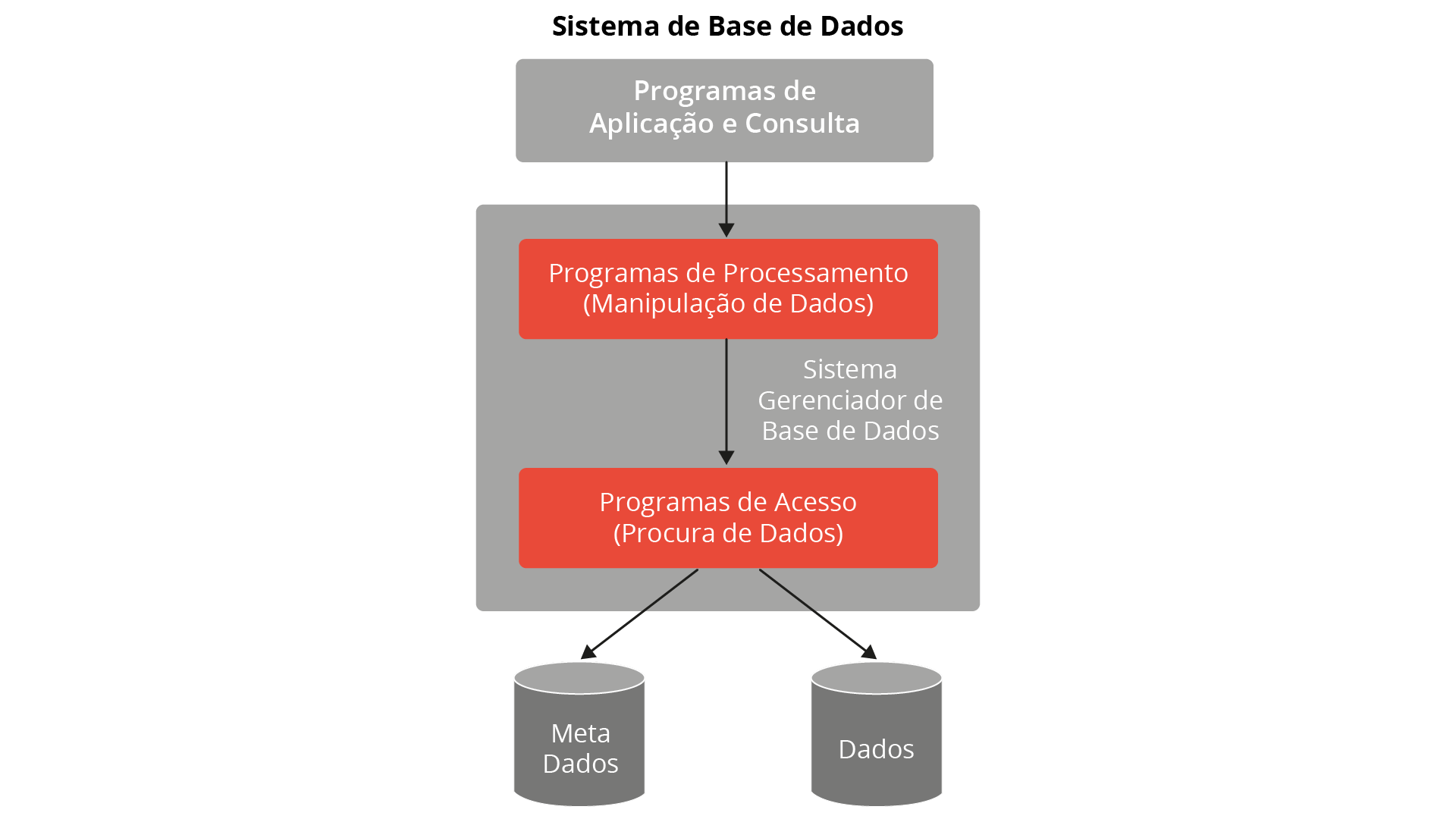
Por outro lado, a instância diz respeito aos dados em si, armazenados e administrados pelo SGBD. As instâncias, ou os dados, sofrem alterações a todo o momento, em função das transações realizadas no banco de dados. Toda instância deve atender às especificações definidas no esquema do banco de dados, e é função do SGBD garantir que essas especificações sejam respeitadas (ELMASRI; NAVATHE, 2011).

A distinção entre esquema e instância é muito importante, uma vez que na elaboração de um projeto de banco de dados o foco está muito mais relacionado à construção do esquema. Uma vez que o esquema está definido, o foco passa a ser nas instâncias, e em um banco de dados recém-criado, encontra-se num estado vazio, precisando ser populado, ou seja, receber dados (novas instâncias).

**Isolamento da Camada de Dados**

Outra característica do SGBD em relação ao uso de catálogo está na capacidade de isolar a camada de dados da aplicação. Em aplicações tradicionais, que não fazem uso de SGBD, e em que os dados são incorporados à aplicação no momento de sua compilação, alterações na estrutura do banco de dados exigem alteração na aplicação, pois se encontra incorporado. Nesse cenário, a estrutura do banco de dados faz parte da estrutura da aplicação e não é possível tratá-la de forma independente (ELMASRI; NAVATHE, 2011).

Aplicações que se utilizam de SGBD possuem camada de dados isolada da aplicação. Isso significa que são independentes e que há maior liberdade em fazer ajustes no banco de dados sem a necessidade de se alterar a aplicação, pois o catálogo possui a função de descrever e administrar os bancos de dados que fazem parte do SGBD (ELMASRI; NAVATHE, 2011). Essa estrutura de isolamento pode ser visualizada na Figura 1.1, na representação da camada de usuário (aplicação) e do sistema gerenciado de banco de dados:

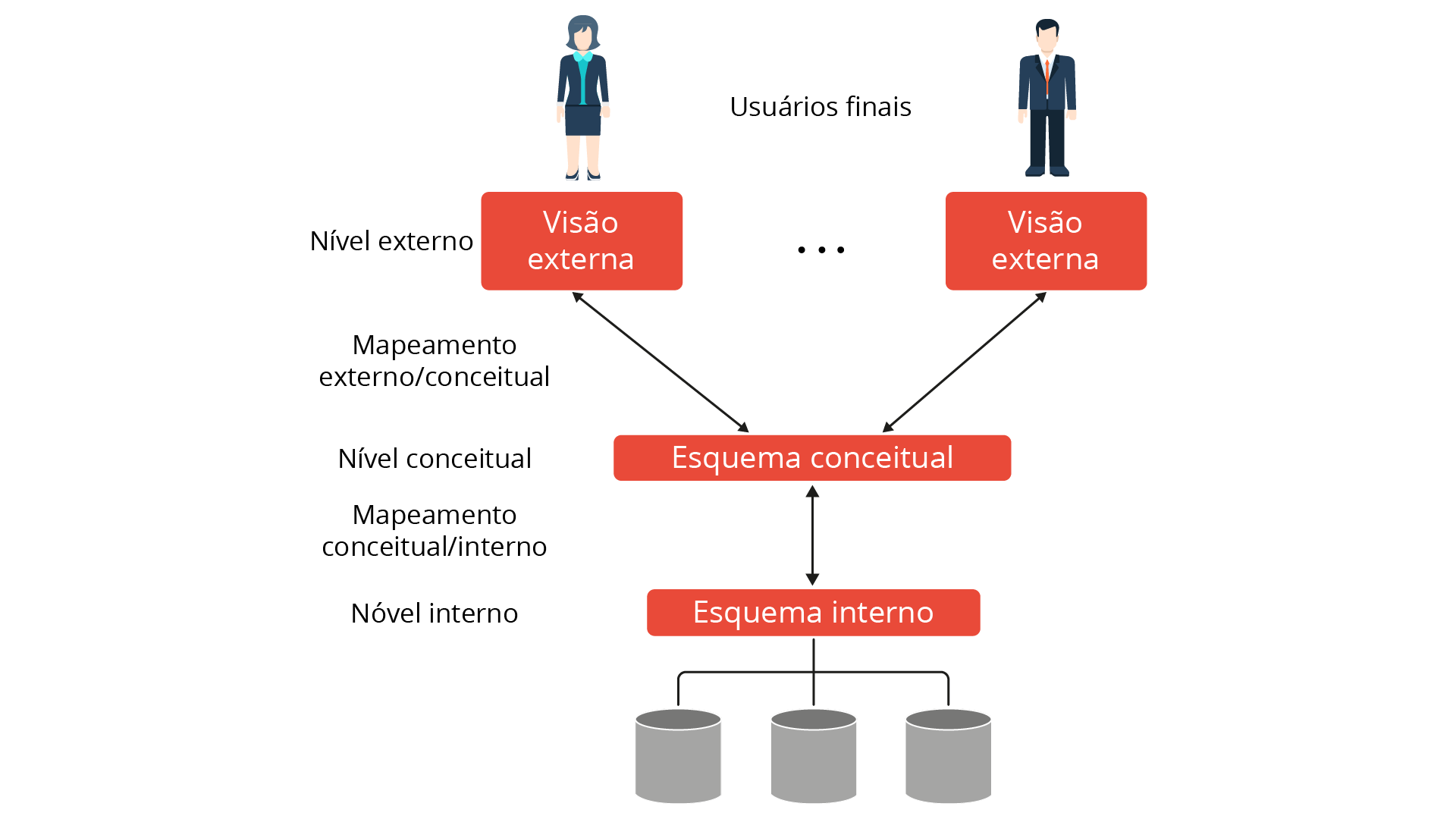
[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.1 - Isolamento de camadas  
Fonte: Elmasri e Navathe (2011, p. 41).*

Em virtude da existência de catálogos, essa característica se torna possível. Assim, as aplicações e os dados operam em camadas distintas. A camada de dados (camada em que o SGBD atua gerenciando os bancos de dados por meio de metadados) opera de forma autônoma, sem nenhuma dependência de aplicações que eventualmente necessitem ter acesso aos seus dados.

Da mesma forma, as aplicações operam em uma camada própria e se conectam ao banco de dados por meio de regras definidas pelo SGBD. As aplicações também possuem maior independência com relação ao seu funcionamento, podendo operar com mais de um SGBD simultaneamente, cada qual em sua camada distinta.

Essa separação de camadas entre aplicação e banco de dados confere maior segurança ao sistema como um todo, bem como aos dados utilizados por ele. Dentro do conceito de camadas existe uma arquitetura comumente utilizada, conhecida como arquitetura de três esquemas. Nessa arquitetura, os esquemas são definidos em três níveis: interno, conceitual e o externo, e são representados na Figura 1.2:

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.2 - Representação de três esquemas  
Fonte: Elmasri e Navathe (2011, p. 22).*

Na arquitetura de três esquemas, conforme apresentado na figura anterior, o nível interno possui um esquema interno, responsável por descrever as características físicas do banco de dados, com informações detalhadas sobre seu armazenamento e acesso. Já o esquema conceitual descreve os aspectos de estrutura do banco de dados em nível de usuário, apresentando detalhes sobre a estrutura de entidade e relacionamentos, ocultando detalhes de estrutura física de armazenamento (ELMASRI; NAVATHE, 2011).

Por fim, há o esquema externo, que descreve a visão do usuário, geralmente implementado a partir de um modelo de projeto representativo baseado em um modelo de alto nível. Além do catálogo e dos elementos que compõem um SGBD, a otimização de consultas é outro recurso importante.

**praticar**

Vamos Praticar

Em sistemas, separar camada é fundamental para que esses módulos operem conectados, porém de forma independente. Um sistema gerenciador de banco de dados possui esse isolamento de camadas e permite separar a aplicação do usuário e operação do banco. Qual o nome da arquitetura usada nesse modelo por um SGBD? Assinale a alternativa correta.

Parte superior do formulário

**a)**Três camadas.

**b)**Três níveis.

**c)**Cliente-servidor.

**d)**Ponto a ponto.

**e)**Model View Controller.

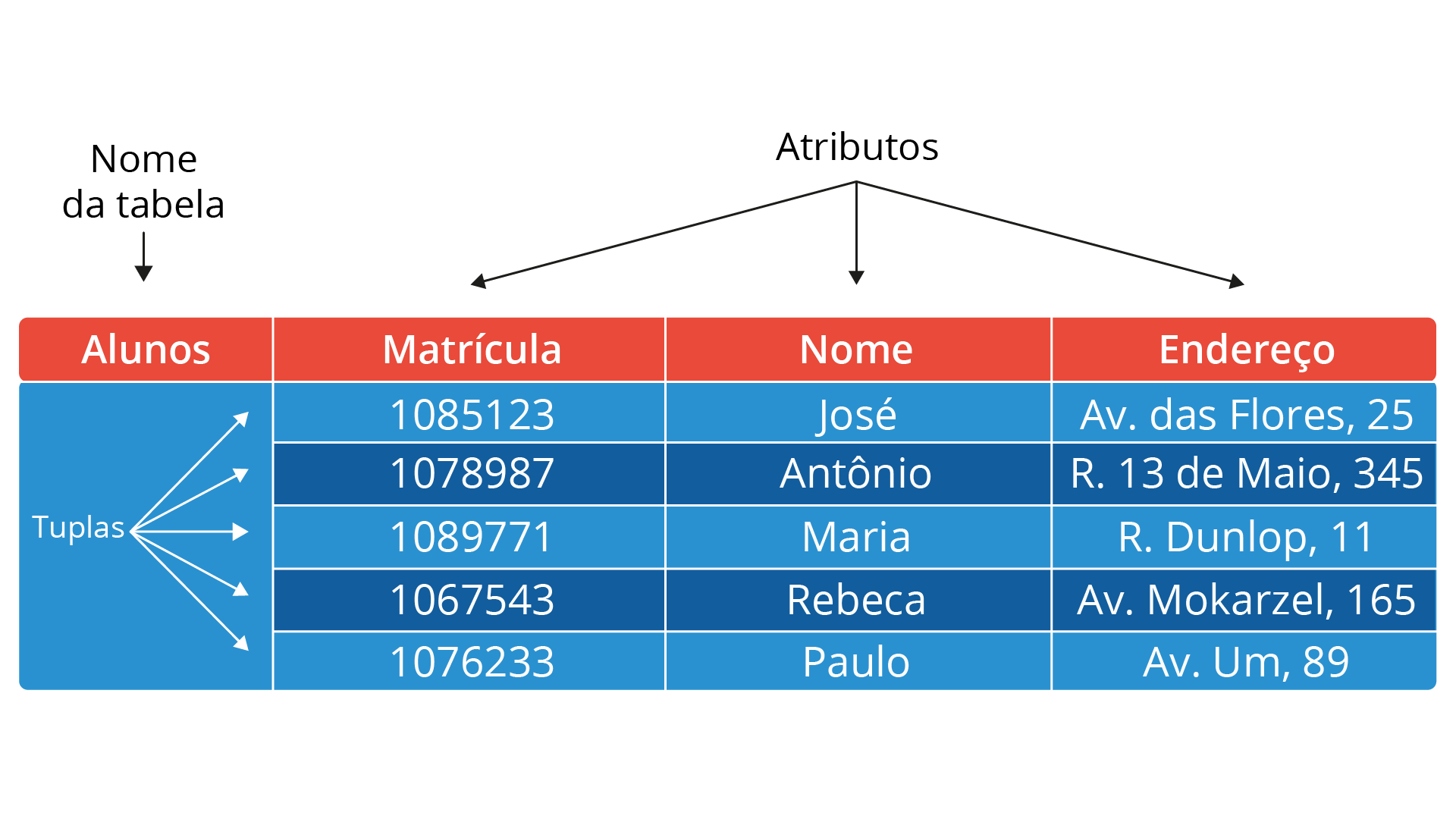
Parte inferior do formulário

**Otimização de Consultas**

O processo de otimização de consultas é um dos aspectos mais importantes de um SGBD. Entretanto, a realização depende da compreensão de alguns conceitos relacionais. Neste tópico, vamos abordar alguns aspectos fundamentais para a realização desse processo. Dentre os conceitos relacionados à otimização de consultas, vamos conhecer o modelo relacional, álgebra relacional e a linguagem SQL. Cada um dos conceitos possui características próprias e é empregado dentro da otimização de consultas de formas específicas. É importante entender sua importância e como estão relacionados dentro do conceito geral.

**Modelo Relacional**

No modelo relacional, o banco de dados é apresentado como uma coleção de relações ou tabela de valores (ELMASRI; NAVATHE, 2011). Na tabela, cada registro possui uma estrutura linear. Para se trabalhar com o modelo relacional, é importante conhecer as terminologias e conceitos empregados: relação se refere à tabela; tupla, representa uma linha ou registro na relação; atributo corresponde a uma coluna ou cabeçalho da tabela. A Figura 1.3 representa uma tabela com dados utilizada no modelo relacional com um conjunto de cinco tuplas, na horizontal. As tuplas também são referidas pelo termo “registro”, enquanto os atributos, por “colunas”.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.3 - Tuplas  
Fonte: Cardoso e Cardoso (2012, p. 40).*

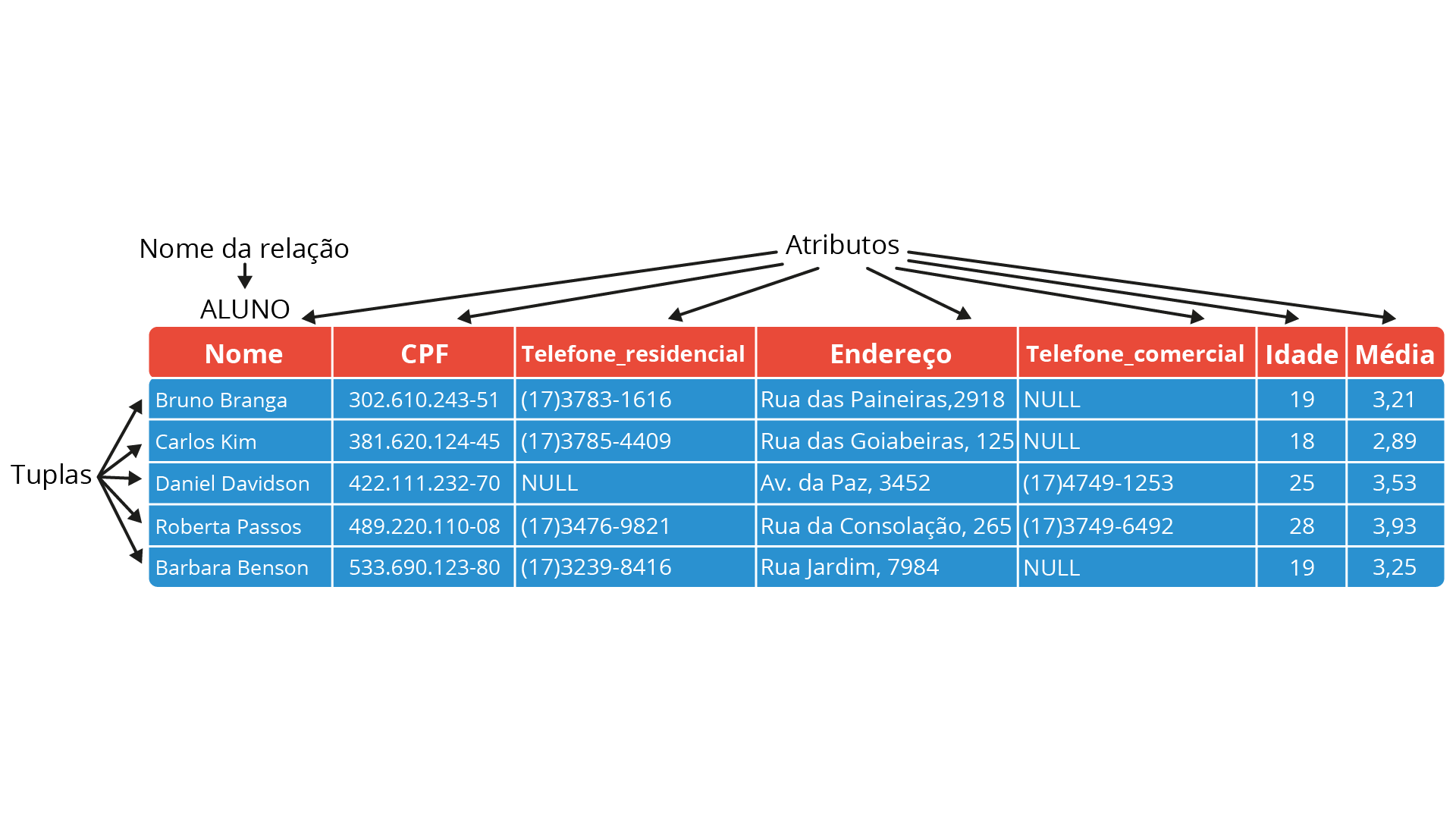
No modelo relacional é utilizado o esquema relacional de banco de dados, que é responsável por descrever uma relação normalmente nomeada no esquema como R. Essa relação possui o grau, que é o número de atributos desse esquema. De forma geral, o esquema relacional pode ser descrito da seguinte forma: R(A1A1, (A2A2, (A3A3,...,(NnNn).

Uma relação de grau seis, que armazena dados sobre alunos em uma escola, poderia ter seu esquema escrito como: ALUNO (Nome, CPF, Telefone\_residencial, Endereco, Idade, Media).

Uma relação r do esquema de relação R é indicada por r(R) e representa o conjunto de n tuplas: r = { (t1t1, (t2t2, (t3t3, ..., (tntn}

Cada n tupla corresponde a uma lista ordenada de n valores: t = { (v1v1, (v2v2, (v3v3, ..., (vnvn}.

Em que cada valor é um elemento do domínio do atributo ou um valor especial NULL. Na Figura 1.4, pode-se observar uma relação ALUNO que corresponde ao esquema ALUNO.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.4 - Relação aluno-atributos  
Fonte: Elmasri e Navathe (2011, p. 4).*

Na Figura 1.4, são apresentados os atributos e tuplas de uma relação ALUNO, com os dados relativos aos registros de alunos. Observe que quando um campo não possui dados, é indicado com o valor NULL. Os esquemas são fundamentais para a descrição das relações que compõem um SGBD.

**Álgebra Relacional**

A álgebra relacional pode ser definida como um conjunto de operações, utilizando linguagem de consulta procedural, tendo como entrada uma ou duas relações (tabelas), produzindo como resultado uma nova relação.

Suas operações estão relacionadas à seleção, projeção, junção, união, diferença, interseção, produto cartesiano, divisão e atribuição. Operações como seleção e projeção são classificadas como primárias ou unitárias, ao passo que operações como junção e produto cartesiano, como binárias (MEDEIROS, 2013). Vamos conhecer um pouco as características dessas operações?

* Seleção

A operação seleção tem como função a seleção tuplas (linhas ou registros), que atendam a uma determinada condição (predicado). É classificada como uma operação unária, já que opera sobre um único conjunto de dados. Sua sintaxe é composta da letra grega maiúscula sigma, denotando seleção, seguida da condição em subscrito e do argumento da relação em parênteses. Exemplo de seleção: σpredicadoσpredicado (tabela)

Assim, para se selecionar a tupla referente ao aluno que tenha o código de matrícula igual a 001, escrevemos: σcodigo=001σcodigo=001 (tabela)

Se o conteúdo da tabela “Aluno” contiver os seguintes registros, um resultado será visualizado (Quadro 1.5).

|  |  |  |  |
| --- | --- | --- | --- |
| código | nome | sexo | nascimento |
| 001 | Jose | M | 01/05/1990 |

Quadro 1.5 - Resultado da seleção  
Fonte: Elaborado pelo autor.

O quadro anterior ilustra um registro com o código (predicado) igual a 001. Isso permite que a consulta (seleção) seja executada com sucesso.

Pode-se adicionar a seleção filtros por meio de operadores lógicos como e (and, ^) , ou (v , or) e o não lógico (not, ⌐). Esses operadores validam a comparação lógica com resultados sempre verdadeiros ou falsos.

Adicionam-se, ao predicado de seleção, símbolos de comparação lógica como igualdade (=), diferente (!=), maior (>), menor( <), maior ou igual (>=) e menor ou igual(<=). Esses símbolos compararam dois objetos num predicado de seleção e, como resultado, tem-se verdadeiro ou falso.

São exemplos de seleção usando esses operadores: σcodigo=001 ^ sexo = f (tabela) e σcodigo=001 ^ ano >= 1990 (tabela).

* Projeção

Trata-se de uma operação unária de relação vertical e tem como função a produção de um subconjunto com apenas os atributos (colunas ou campos) escolhidos na condição especificada. Sua sintaxe é composta da letra grega maiúscula pi (π), denotando projeção, seguida da lista de atributos em subscrito e o argumento da relação em parênteses. Por exemplo: π atributo (tabela). Para selecionar da tabela anterior (“Aluno”) apenas as colunas código e nome, escreve-se π código, aluno (tabela), seja, uma tabela seria representada por dois campos (código, nome).

* Junção

Trata-se de uma operação binária que tem como função a união de duas relações que possuem um atributo em comum (MEDEIROS, 2013). Sua sintaxe é composta do símbolo de junção, seguido da relação e dos atributos de igualdade em subscrito. Por exemplo: tabela1 atributo=atributo tabela2. Sendo assim, quando se deseja saber uma relação dos dados dos alunos que efetuaram pagamento de um determinado curso, escreve-se: pagamento aluno=codigo aluno. Nesse exemplo de junção, o resultado será composto de nome de alunos (não repetidos), e os registros serão somente aqueles que constam na tabela pagamento.

* União

Trata-se de uma operação binária que tem como função criar uma relação a partir de duas outras. Sua sintaxe é composta da relação 1 seguida do símbolo de união e da relação 2. Por exemplo: tabela1 U tabela2. Para saber os dados resultantes da união de alunos de duas escolas, escreve-se: aluno1 U aluno2. A tabela resultante será composta somente dos registros que estejam presentes como conteúdo das tabelas “Aluno1” e “Aluno2”.

A união também pode ser feita a partir do resultado de duas projeções, para retornar o nome dos alunos das duas tabelas. Por exemplo: πnome (aluno1) U πnome (aluno2).

* Intersecção

Trata-se de uma operação binária que tem como função criar uma relação a partir de duas outras relações, estabelecendo uma intersecção das duas, em que na relação gerada serão apresentados apenas os dados pertencentes às duas condições preestabelecidas (MEDEIROS, 2013). A sintaxe é composta da relação 1, seguida do símbolo de intersecção e da relação 2. Por exemplo: tabela1 tabela 2.

* Diferença

Trata-se de uma operação binária que tem como função criar uma relação a partir de duas outras relações. Na relação gerada, serão apresentados apenas os dados que não são comuns às duas relações. Sua sintaxe é composta da relação 1, seguida do símbolo de diferença, e da relação 2. Por exemplo: tabela1 -tabela2. Sendo assim, para saber os dados resultantes da diferença entre as tabelas de “Alunos” das escolas, escreve-se: aluno1 – aluno2.

* Produto cartesiano

Trata-se de uma operação binária que tem como função criar uma relação a partir de duas outras relações. Na relação gerada, serão apresentadas todas as combinações possíveis entre os elementos das relações. Sua sintaxe é composta da relação 1, seguida do símbolo de produto cartesiano, e da relação 2. Por exemplo: tabela 1 x tabela 2. Produto cartesiano trata-se de uma operação dispendiosa, uma vez que o número de registro da primeira tabela é multiplicado pelo número de registros da segunda.

* Renomeação

Trata-se de uma operação unária que tem como função definir o nome de uma tabela em um determinado contexto. Sua sintaxe é composta da letra grega maiúscula (rho), denotando renomeação, seguida do novo nome e do argumento da relação, em parênteses. Por exemplo: p <novo nome> tabela. Sendo assim, para renomear a tabela aluno para estudante, escreve-se: p estudante (aluno).

Todas as operações da álgebra relacional podem ser executadas por meio da Linguagem SQL, assunto do nosso próximo tópico.

**Linguagem SQL**

A linguagem SQL (Structured Query Language) foi desenvolvida pela IBM na década de 1970, tornando-se um padrão para os bancos de dados relacionais e ISO. Por meio dessa linguagem, é possível a definição de metadados e manipulação dos dados em um SGBD, servindo de interface entre o usuário e o SGBD (VICCI, 2014).

Na linguagem SQL, são utilizados os conceitos de tabela, coluna e linha como representação dos conceitos já estudados de relação, atributo e tupla (SANCHES, 2005).

O comando mais utilizado para consultas num banco de dados pela linguagem SQL é o comando SELECT FROM. Podem-se fazer consultas em tabelas, especificando os campos desejados e as condições para a pesquisa, bem como a ordem em que o resultado deve ser apresentado (MEDEIROS, 2013). Esse comando possui a seguinte sintaxe:

SELECT campos FROM tabela WHERE condição.

Sendo assim, para mostrar todos os registros da tabela aluno, escrevemos:

SELECT \* FROM aluno.

O símbolo de (\*) representa todos os campos da tabela. Esse comando select apresenta diversas variações de consultas, pois a string SQL dependerá da regra que deseja ser aplicada ao banco de dados. O Quadro 1.6 apresenta algumas variações do comando select.

|  |  |
| --- | --- |
| **Comando** | **Explicação** |
| SELECT \* FROM aluno WHERE codigo=”001”; | Seleciona somente o registro com código igual a 001. |
| SELECT nome, nascimento FROM aluno; | Seleciona os campos nome e nascimento da tabela aluno |
| SELECT nome, nascimento FROM aluno order by nome; | Seleciona nome, nascimento da tabela aluno ordenado por nome |
| SELECT \* FROM pedido WHERE numero < 3 AND valor > 1500; | Seleciona todos os registros da tabela, pedindo que tenha número menor do que 3 e valor maior do que 1500 |

Quadro 1.6 - Variações do select  
Fonte: Elaborado pelo autor.

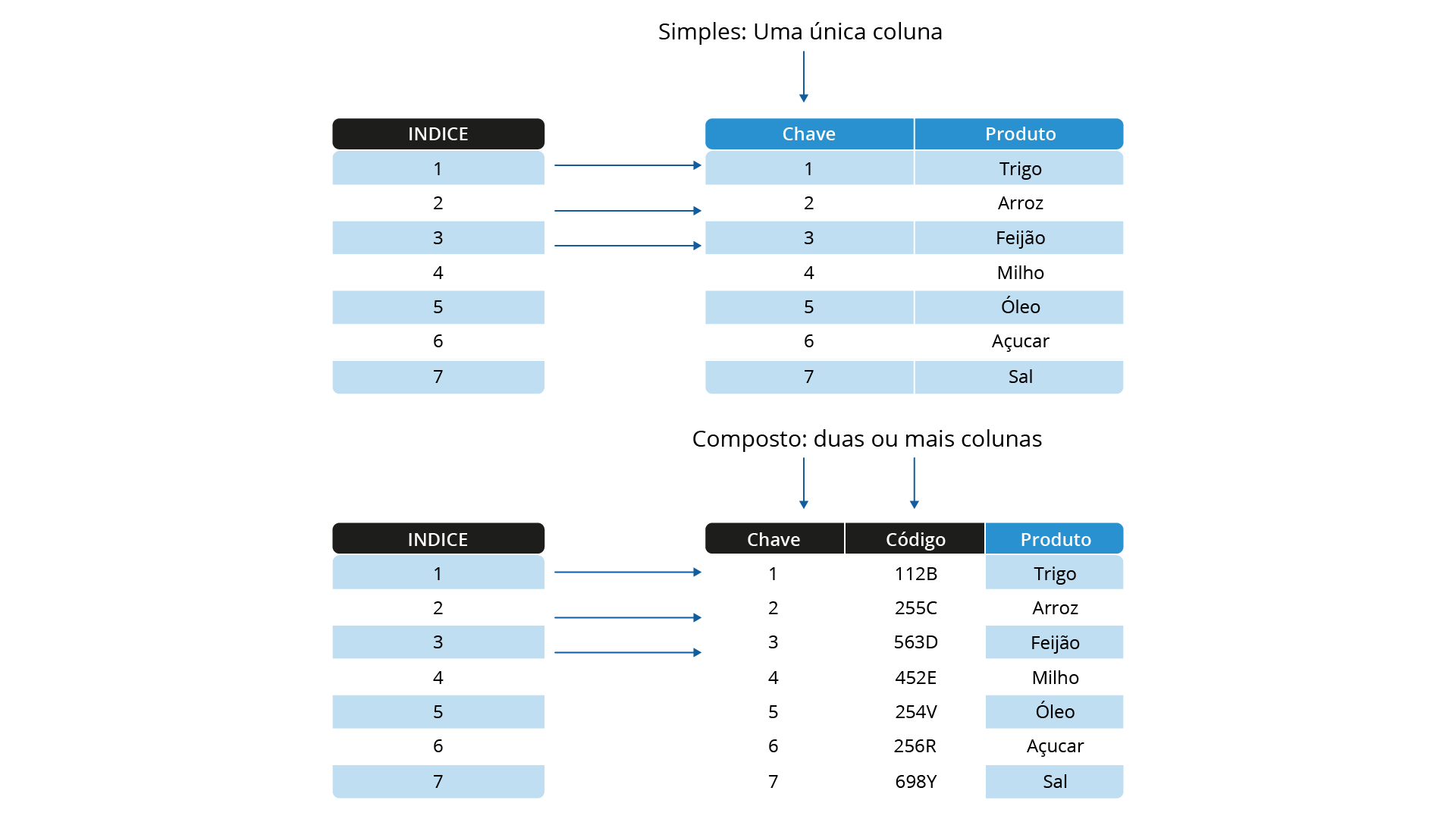
Nota-se que algumas das sintaxes SQL apresentaram a palavra reservada WHERE. Essa palavra é um filtro para o comando SELECT. Comandos lógicos e de comparação lógica podem ser adicionados aos filtros para uma especificação de registo.

**Otimização**

* Índices

Na otimização de uma consulta, o uso de índice é fundamental. Índices melhoram o desempenho da consulta, garantem que determinados dados chaves não sejam duplicados, permitem acesso a dados ordenados sem custo de executar uma ordenação e busca dados específicos com melhor desempenho. O uso de índice pelos gerenciadores de banco de dados se assemelha ao uso de índice num livro. O objetivo é o de otimizar o tempo de acesso aos registros (tuplas, linhas) de uma tabela, por meio de ponteiros criados para os dados armazenados em colunas específicas. Porém, índices não são úteis para inserção, o processo torna-se mais lento, aumenta o consumo de espaço em disco e a necessidade de manutenção interna no banco de dados.

Um índice pode ser composto de um ou mais campos. Quando formado por apenas um único campo é tido como simples, e quando formado pela combinação de dois ou mais campos, é considerado um tipo de índice composto.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.5 - Índices  
Fonte: Elaborada pelo autor.*

Um índice, no banco de dados, é um campo que serve para otimização da tabela. Por meio do uso de índices, a operação de consultar uma tabela para localizar um registro torna-se mais rápida, porque os índices criam ponteiros para dados armazenados em colunas específicas.

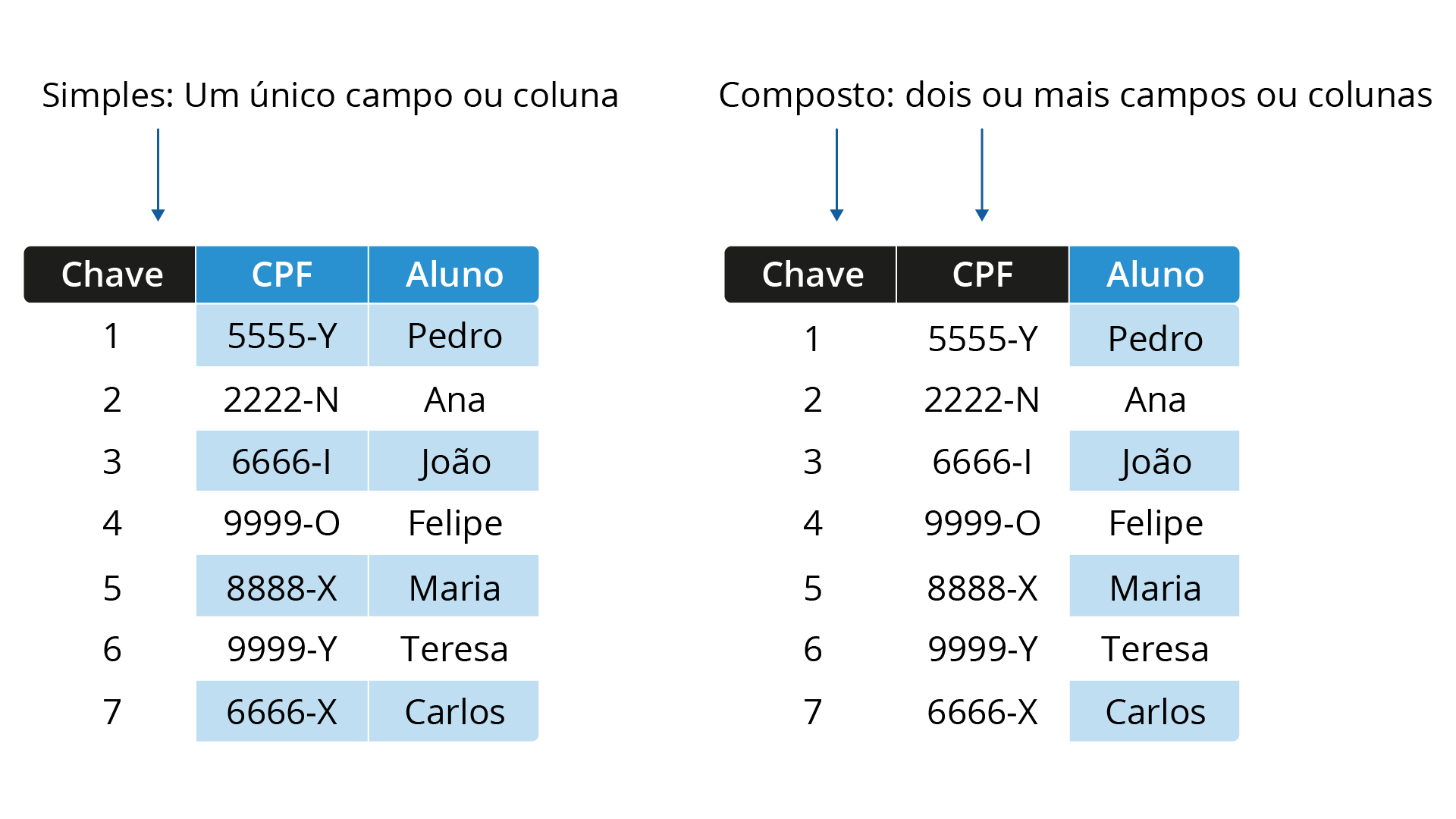
O princípio de índice remissivo num livro – que, por meio da localização de um tópico no índice localiza-se a página onde está a informação – é usado no banco de dados quando um tabela possui índices.

Índice, em banco de dados, pode ser simples ou composto e criado pelo comando CREATE INDEX. Um índice simples faz referência a somente uma única coluna; já um índice composto faz referência a duas ou mais colunas.

Os campos de indexação se referem aos campos utilizados na composição do índice de uma tabela. Os índices são armazenados em uma tabela própria, que contém apenas o valor do campo de indexação e um ponteiro que faz a ligação com o registro correspondente da tabela. Os índices são atualizados automaticamente, toda vez que há alguma alteração no banco de dados.

● **Chaves**

* As chaves são utilizadas para a identificação de um registro e para o estabelecimento de ligações entre linhas de tabelas e podem ser dos tipos primária, alternativa ou estrangeira. Essas ligações entre tabelas são chamadas relacionamento. Esses relacionamentos, por sua vez, garantem uma integridade de dados referenciais. Chaves podem ser consideradas restrições dentro da tabela. Chave primária: trata-se de um identificador único, não duplicado, não nulo, indexado e que distingue uma linha de outras em uma tabela. Exemplos: código, CPF, *e-mail* (Figura 1.6).

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

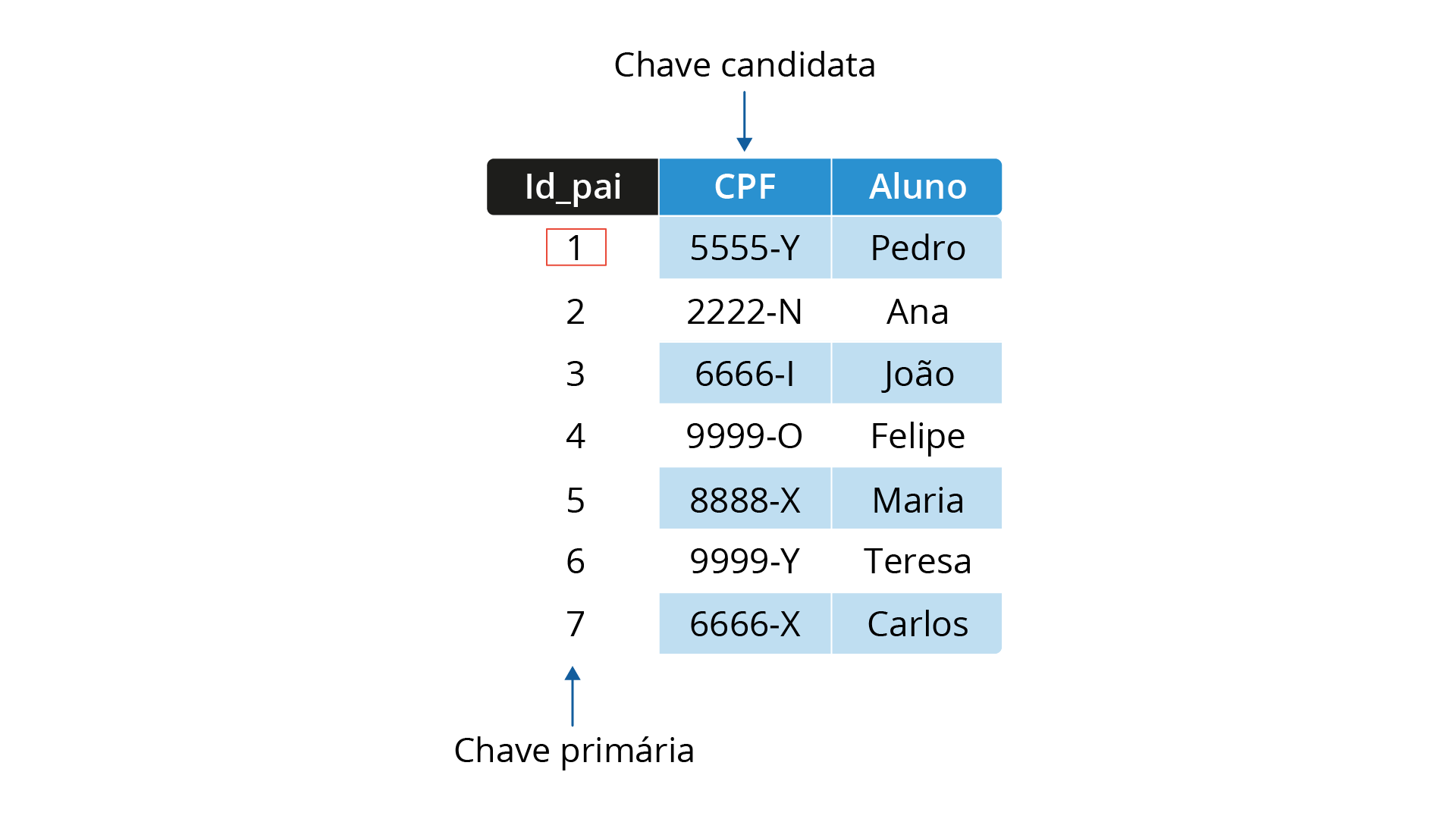
*Figura 1.6 - Chave primária  
Fonte: Elaborada pelo autor.*

A chave primária, numa tabela de banco de dados, serve como um índice interno. Essa chave tem de ser indexada, não nula e não repetida. Uma chave primária é usada nos relacionamentos entre entidades para garantir uma referência de dados ou integridade referencial.

Uma integridade referencial, no banco de dados, é a garantia de que o dado existe. Exemplo: nenhum produto é cadastrado sem antes se cadastrar o fornecedor desse produto. Assim, há uma garantia de que todos os produtos têm origem de um ou mais fornecedores, desde que exista um relacionamento entre a tabela produto e tabela fornecedor.

Uma chave primária pode ser simples, um único campo (coluna), ou composta, diversos campos (colunas). A chave primária é criada pelo termo primary key, relacionado a um campo da tabela, exemplo, primary key (cpf). Na criação da tabela ou na alteração de uma estrutura de tabela.

* Chave alternativa (candidata) (Figura 1.7): trata-se do uso de colunas alternativas que podem ser usadas para a identificação de um registro ou sua distinção em uma tabela. Nesse caso, mais de uma coluna, ou combinações de colunas podem ser utilizadas para distinguir uma linha das demais. Exemplo: suponha uma tabela composta dos campos id, nome e CPF, sendo que id é chave primária. Nesse caso, CPF é uma chave alternativa. Chaves alternativas só são expressas na teoria.

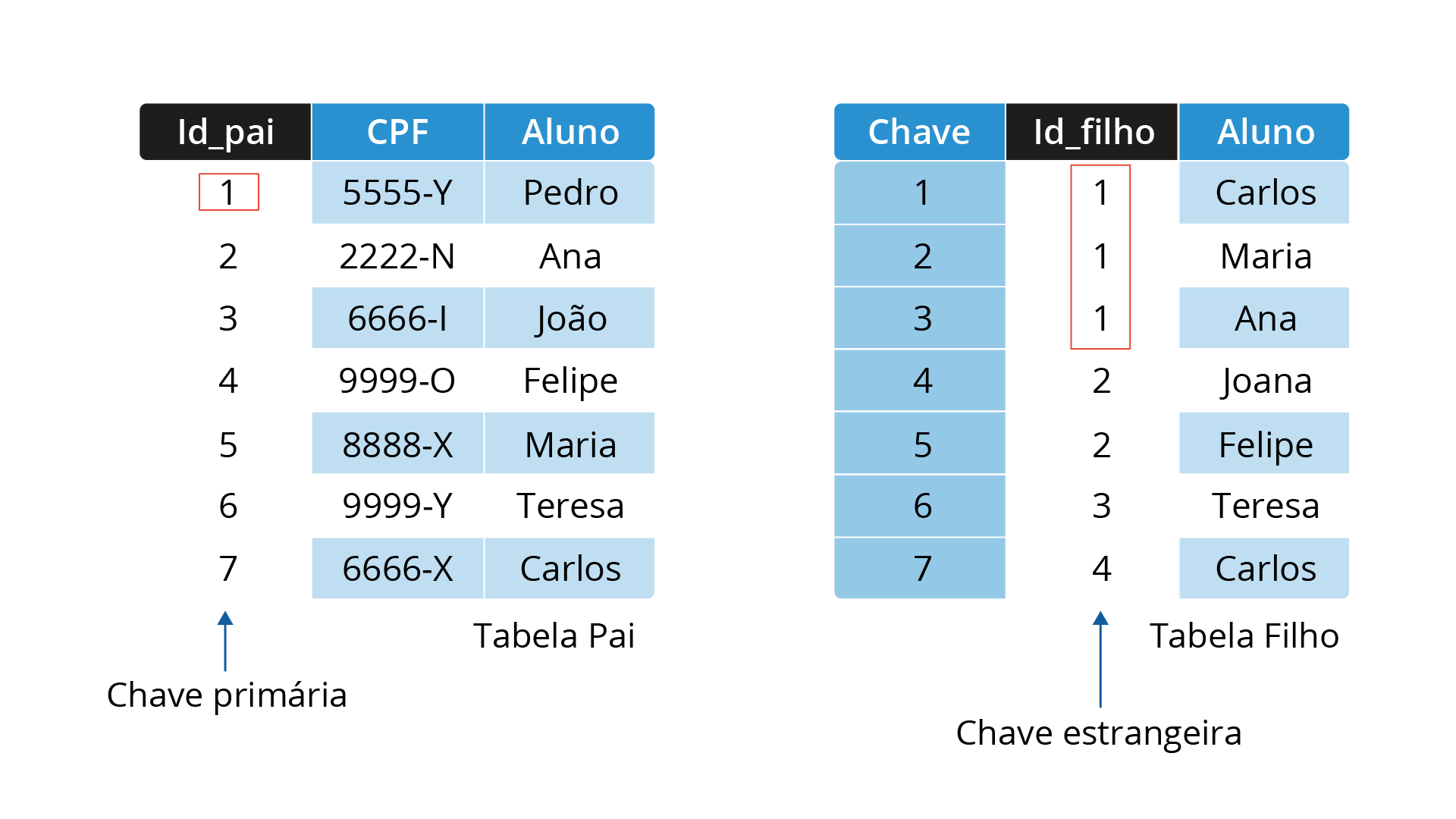
[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.7 - Chave candidata  
Fonte: Elaborada pelo autor.*

Uma chave candidata ou alternativa é a que tem as mesmas características de uma chave primária, ou seja, um campo que pode ser indexado, não nulo e não repetido, porém não está sendo usada como chave primária.

Uma chave candidata tem apenas valor teórico, pois, na prática, não é usada e nem possui qualquer valor como índice, porque não é chave primária. Exemplos de campos que podem ser chave candidata: cpf, *e-mail*, código.

* Chave estrangeira (Figura 1.8): trata-se de uma coluna que estabelece uma relação com as linhas de outra tabela. Ou seja, nessa coluna, é armazenado o valor da chave primária de uma tabela na qual se deseja fazer uma ligação.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

*Figura 1.8 - Chave estrangeira  
Fonte: Elaborada pelo autor.*

A chave estrangeira é um campo que está geralmente associado (relacionado) a uma chave primária, para garantir uma integridade referencial de dados. A chave estrangeira é criada utilizando o termo *foreign key* na criação ou alteração de uma estrutura de tabela. Deve-se passar o campo da tabela que será a chave estrangeira e relacionar com o campo origem da informação (chave primária) da tabela relacionada. Exemplo, FOREIGN KEY (id-filho) REFERENCES TAB\_PAI (id\_pai), em que id\_filho é o campo chave estrangeira da tabela filho e id\_pai o campo chave primária da tabela pai, desde que exista o relacionamento entre tabelas.

Para melhor desempenho, é recomendado que também se crie um INDEX no campo da chave estrangeira. O nome do campo da chave estrangeira pode ser diferente do da chave primária, mas deve ter o mesmo tipo de dados, ou seja, inteiro (FK) e inteiro (PK) ou texto(FK) e texto (PK).

É importante entender que no relacionamento entre chaves primárias e estrangeira, o nome dos campos pode ser diferente, porém o tipo de dado tem de ser o mesmo. Por exemplo: idAluno (inteiro) relacionado num\_aluno(inteiro).

A chave estrangeira é o mecanismo que o SGBD utiliza para a implementação de relacionamentos em um banco de dados que deseja fazer uma ligação num banco de dados relacional.

**Transação**

Trata-se de um conjunto de procedimentos que são executados em um SGBD, cuja integridade está relacionada a quatro propriedades, conhecidas como ACID: Atomicidade, Consistência, Isolamento e Durabilidade. Essas propriedades são descritas por Elmasri e Navathe (2011) da seguinte forma:

* atomicidade: está relacionada ao fato de que todas as ações que compõem uma unidade de trabalho da transação devem ser concluídas com sucesso, para que a transação possa ser efetivada. Se uma eventual falha ocorrer durante o processo, toda a transação deve ser desfeita, o que é conhecido como *rollback*. Por outro lado, quando todas as ações são executadas com sucesso, a transação pode então ser efetivada, o que é chamado*commit*. Esses comandos fazem parte da categoria de comandos de transação de dados (DTL).
* consistência: diz respeito às regras e restrições que são definidas no banco de dados. Todos os relacionamentos por chave estrangeira e validação de campos devem ser obedecidos, para que a transação possa ser efetivada.
* isolamento: as transações devem ocorrer de forma independente e nenhuma transação deve operar no mesmo sistema, pois interfere no funcionamento da transação corrente, nem mesmo visualizar os resultados parciais das operações de uma transação em andamento.
* durabilidade: todas as alterações efetuadas em um banco de dados como resultados de uma transação são permanentes e devem persistir no banco de dados, não sendo passíveis de falha de *hardware*.

As transações estão relacionadas à integridade das informações que são manipuladas em um SGBD. Embora não seja visível para o usuário, internamente, uma série de comandos e mecanismos de controle é executada para a garantia dessa integridade. Entretanto, trata-se de um processo custoso em termos de tempo e processamento, o que demanda maior capacidade de *hardware*.

**Comandos de Manipulação de Dados**

A linguagem SQL é uma poderosa ferramenta para manipulação de banco de dados. As instruções DML (Data Manipulation Language) se referem ao grupo de comandos dentro da linguagem SQL responsáveis pelos processos de recuperação, inclusão, exclusão e alteração de dados em um sistema gerenciador de banco de dados. Essas operações são chamadas, popularmente, CRUD. A letra R faz referência à consulta.

Alguns dos comandos relacionados às transações em banco de dados são: INSERT, UPDATE e DELETE.

* **Inserção de registros**

Para a inserção de registros em uma tabela, utiliza-se o comando INSERT, que tem a seguinte sintaxe: INSERT INTO nome\_tabela (lista-de-campos) VALUES (lista-de-dados).

Para inserir um novo registro numa tabela de nome alunos, escreve-se: INSERT INTO aluno(codigo, nome, sexo, nascimento) VALUES (“004”, ”Roberta”, ”F”, 10/04/1995).

* **Alteração de registros**

Para a alteração de registros em uma tabela, utiliza-se o comando UPDATE, que tem a seguinte sintaxe: UPDATE nome\_tabela SET CAMPO = "novo-valor" WHERE condição.

Para alterar o nome do aluno de código de matrícula número 001 de José para Manoel, na tabela de chamada de alunos, escreve-se: UPDATE aluno SET NOME = “Manoel” WHERE CODIGO=”001”.

* **Exclusão de registros**

Para a exclusão de registros em uma tabela, utiliza-se o comando DELETE, que tem a seguinte sintaxe: DELETE FROM nome\_tabela WHERE condição.

Para excluir o registro do aluno de código de matrícula número 02 na tabela de alunos, escreve-se: DELETE FROM aluno WHERE CODIGO=”002”.

Em todos os comandos INSERT, DELETE E UPDATE, a operação é unilateral, ou seja, escrita ou manipulação diretamente no banco. Essas operações não possuem retorno de informação do banco. O gerenciador somente sinaliza que a operação foi bem-sucedida ou não. Ao contrário de um operação DQL (Data Query Language) ou operação de consultas com*select*. Essa operação sempre retornará um conjunto de informações chamado *recordset* (PUGA; FRANÇA; GOYA, 2014).

Aprendemos sobre as transações em um SGBD, agora, nosso estudo prossegue com outro tópico importante: o controle de concorrência.

**praticar**

Vamos Praticar

Por meio da linguagem SQL, um usuário consegue interagir com o banco de dados. Essa especificação é padrão em todos os sistemas gerenciadores de banco de dados. Um CRUD SQL é uma operação de criação, exclusão, atualização ou consulta, leitura. Sobre comandos DML, assinale a alternativa correta:

Parte superior do formulário

**a)**Filtros no comando *select* podem ser realizados por meio da palavra reservada JOIN, que tem suporte a operadores lógicos e aritméticos.

**b)**Atualizar uma tabela é trocar o nome dela, por exemplo, tabela aluno para alunos por meio do uso do comando UPDATE.

**c)**Quando um usuário cadastra informações via um formulário numa plataforma web, internamente o sistema faz uso do comando o INSERT.

**d)**Comandos de transação, como COMMIT e ROLLBACK, são usados em sistema financeiros para garantir a transação via mecanismo MyISAM no MySQL.

**e)**Uma tabela deve ser criada utilizando o comando SQL CREATE e não precisa declarar o mecanismo MyISAM, pois esse é padrão do MySQL.

Parte inferior do formulário

**Controle de Concorrência**

O controle de concorrência é um mecanismo do SGBD que permite a execução de aplicações concorrentes ou que aplicações concorrentes acessem o banco de dados (ELMASRI; NAVATHE, 2011). É caracterizado quando duas ou mais entidades desejam utilizar o mesmo recurso. O controle de concorrências tem como propósito a minimização de problemas relacionados a atualizações perdidas, dependência sem*commit* e análise inconsistente. Por meio do controle de concorrências, é possível a ocorrência de múltiplas conexões e operações em um sistema gerenciador de banco de dados.

**Gerenciamento de Bloqueios**

O gerenciador de bloqueio é uma função do SGBD que tem como propósito o monitoramento dos bloqueios concedidos às transações. Esse controle é feito por meio de uma tabela de *hashing*, que identifica o objeto de dados pela sua chave (ELMASRI; NAVATHE, 2011).

Uma das técnicas de bloqueio utilizada é a de duas fases, pois esta técnica é  baseada em valores binários (0, 1) associados aos itens de dados. Portanto, quando o item de dados está associado ao valor zero (0), é livre para transações. Caso contrário, o item de dados está associado ao valor um (1), logo, bloqueado para transação As operações para o bloqueio binário são o lock(1) e unlock(0) para desbloqueio, ou seja, representam estado de bloqueado e não bloqueado.

**reflita**

Reflita

Trabalhar no banco de dados com acesso direto pode provocar falhas no sistema e permitir que dados sejam furtados por pessoas sem permissão de acesso. Os dados podem ser usados para influenciar opiniões, definir tendências etc.

A maneira pela qual o bloqueio é efetuado pode ser compartilhado ou exclusivo. O bloqueio compartilhado ocorre quando uma transação recebe esse tipo de bloqueio com a instrução de leitura, permitindo que outras transações possam acessar o mesmo dado apenas para fins de leitura. Porém, quando a instrução é de gravação, não se aplica o bloqueio compartilhado, mas, sim, o bloqueio exclusivo, o que impede que outra transação faça uso do dado enquanto este estiver bloqueado (ELMASRI; NAVATHE, 2011).

O processo de bloqueio e desbloqueio ocorre o tempo todo em um SGBD, pois cada transação necessita manter o bloqueio do item de dado para sua operação. O dado somente deverá ser disponibilizado assim que a transação é concluída. Trata-se de um processo que demanda um controle eficiente e que gera uma carga de processamento no SGBD. Dependendo do SGBD, os bloqueios podem ocorrer em nível de registro (tupla/linha) ou no nível da tabela (relação).

**Problemas Clássicos**

Um dos primeiros problemas que devem ser evitados pelo controle de concorrência é conhecido como atualização perdida. O Quadro 1.7 apresenta o exemplo de atualização perdida na tabela. Isso ocorre quando dois leem os mesmos valores contidos no banco de dados, e, posteriormente, tentam de atualizá-lo com base na informação lida antes que uma das atualizações tenha sido concluída com sucesso.

|  |  |  |
| --- | --- | --- |
| **T1** | **Tempo** | **T2** |
| **Ler(A)** | **1** |  |
|  | **2** | **Ler(A)** |
| **Gravar(A)** | **3** |  |
|  | **4** | **Gravar(A)** |

Quadro 1.7 - Atualização perdida  
Fonte: Elaborado pelo autor.

Note que os dados do campo T2 ficam sem valores na ação de T1, e o mesmo ocorre na ação de T2: o campo T1 fica sem valores.

Outro problema a ser administrado pelo controle de transações é conhecido como leitura suja, pois ocorre quando uma transação atualiza um item do banco de dados, ao mesmo tempo em que outra transação lê esse mesmo item, aparentemente atualizado. Esse problema é ilustrado no quadro 1.8:

|  |  |  |
| --- | --- | --- |
| T1 | Tempo | T2 |
| Ler(A) | 1 |  |
| Gravar (A) | 2 |  |
| Falha | 3 | Ler(A) |
| Retorna (A) | 4 | Gravar(A) |

Quadro 1.8 - Leitura suja  
Fonte: Elaborado pelo autor.

Note que no campo T1 ocorreu uma falha, pois operações de leitura e gravação ocorrem no mesmo item.

**praticar**

Vamos Praticar

Sistema de banco de dados são compartilhados, porém precisam ser íntegros. Hoje, as aplicações compartilham informações em bancos de dados e, por esse motivo, devem preservar os princípios da ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Assinale a alternativa que apresenta um problema causado por falha de ACID.

Parte superior do formulário

**a)**Atualização perdida

**b)**Atualização *on-line*.

**c)**Leitura *on-line*.

**d)**Leitura perdida.

**e)**Atualização suja.

Parte inferior do formulário

**indicações**

**Material Complementar**



**LIVRO**

**Banco de dados**

Sandra Puga, Edson França e Milton Goya

**Editora:**Pearson

**ISBN:**978-85-4301-504-0

**Comentário:**esse livro compreende uma abordagem mais ampla com relação ao um sistema gerenciador de banco de dados, com exemplos práticos na linguagem SQL. A leitura corrobora para uma melhor compreensão sobre armazenamento de dados.



**FILME**

**O quinto poder**

**Ano:**2013

‍**Comentário:**esse filme faz um roteiro da história do WikiLeaks, desde sua criação até a polêmica gerada pelo armazenamento e compartilhamento de informações consideradas secretas por autoridades governamentais. O filme discute, além do aspecto de confidencialidade da informação, os relacionados a rotinas diárias de manipulação dos dados e fragilidade de segurança. Milhares de documentos foram roubados. O problema relacionado à falha, pelo que foi divulgado, era pertinente ao acesso aos dados diretamente no banco de dados, que não previa características de segurança e podia ser facilmente copiado.

[TRAILER](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_1/ebook/index.html)

**conclusão**

**Conclusão**

Nesta unidade, você compreendeu a necessidade do uso dos sistemas gerenciadores de banco de dados (SGBD). Essa ferramenta (*software*) compartilha grandes volumes dados por meio de comandos privados pela linguagem SQL. Na era da Internet, o SGBD se tornou um recurso de tecnologia fundamental para a gestão e organização das informações, com agilidade e precisão.

Porém, há a necessidade de segurança na manipulação de dados por meio de gerenciamentos de operação, para que não ocorram problemas como o de leitura suja. Por fim, você aprendeu os conceitos da linguagem SQL e os mecanismos de banco de dados exemplificados no SGBD MySQL.

**referências**

**Referências Bibliográficas**

CARDOSO, V; CARDOSO, G. **Sistema de banco de dados**: uma abordagem introdutória e aplicada. São Paulo: Saraiva, 2012.

ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados**. 6. ed. São Paulo: Pearson Education do Brasil, 2011.

MEDEIROS, L. F. de. **Banco de dados**: princípios e práticas. Curitiba: Intersaberes, 2013.

MYSQL 8.0 reference manual: including MySQL NDB Cluster 8.0. **My SQL**, 2020. Disponível em <https://dev.mysql.com/doc/refman/8.0/en/>. Acesso em: 11 de jan de 2020.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados**: implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2014.

ROB, P.; CORONEL, C. **Sistemas de banco de dados**: projeto, implementação e gerenciamento. São Paulo: Cengage Learning, 2011. cap.1, p. 6-10.

SANCHES, A. R. Notas de aula. **Fundamentos de armazenamento e manipulação de dados**, São Paulo, 2005. Disponível em: <https://www.ime.usp.br/~andrers/aulas/bd2005-1/aula3.html>. Acesso em: 11 de jan 2020.

VICCI, C. **Banco de dados**. São Paulo: Pearson Education do Brasil, 2014.

© 2020 - LAUREATE - Todos direitos reservados

**Práticas de Banco de Dados**

Práticas de Banco de Dados

1.00

**PRÁTICAS DE BANCO DE DADOS**

PRÁTICAS DE BANCO DE DADOS

Autor: Me. Paulo Sérgio Pádua de Lacerda

Revisor: Paulo Lacerda

[INICIAR](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html#section_1)

**introdução**

Introdução

A proposta principal desta unidade é definir conhecimento sobre procedimentos armazenados. Uma vez que esses procedimentos ficam armazenados no banco de dados, ao usá-los, há a melhoria no desempenho do sistema como um todo, pois são executados do lado do servidor. Esses procedimentos reduzem tráfego na rede e possibilitam a reutilização de código. Capacitar-se nos fundamentos de tópicos avançados aumenta a habilidade de trabalhar com os bancos de dados. Nesta unidade, você ficará familiarizado com tópicos como virtualizações de tabelas, gatilhos e procedimentos armazenados.

**Criando Virtualizações**

Um sistema gerenciador de banco de dados faz uso de virtualizações de tabelas e permite criar visões (views) do modelo de dados. Na verdade, views são tabelas virtuais predefinidas, que resultam numa consulta derivada de outras tabelas, porém não requer armazenamento físico e, sim, virtual (PLEW; STEPHENS, 2000).

Usuários visualizam as views como uma tabela normal, porque uma view tem o aspecto de uma tabela comum e é acessada com uma tabela. Views são consideradas TABELAS VIRTUAIS, entretanto, quando uma visão é criada, uma instrução SQL SELECT é executada no banco de dados.

Embora, a virtualização gere certa limitação em operações de atualizações possíveis nas visões, contudo, em operações de consulta, não há restrições. Numa instrução SELECT, para definição de uma visualização pode conter somente uma simples consulta, ou mais ainda, cálculos complexos, diversas funções explicitamente escritas, com a finalidade de manipular e resumir dados ao usuário.

 Visões são objetos armazenados num banco de dados, entretanto, não ocupam espaço no servidor de dados com relação ao armazenamento. Logo, a diferença principal entre uma View e uma tabela diz respeito aos dados: enquanto a tabela ocupa espaço de armazenamento, pois possui dados físicos, a View não precisa de espaço de armazenamento, porque faz referência aos dados das tabelas utilizadas em sua criação.

    Desse modo, uma visão fica armazenada em memória, ao contrário dos outros objetos de banco de dados. O proprietário do schema (ou o criador da visão) é o dono (owner) das views. Esse proprietário, da mesma maneira que nas tabelas, possui total privilégios sobre as views e pode conceder ou controlar privilégios a outros usuários sobre a virtualização por meio do comando GRANT, com a opção OPTION.

**reflita**

Reflita

Restrições podem ser implementadas nas Views por meio da cláusula WHERE definida na View para colunas ou linhas, com o objetivo de evitar o acesso de usuário as colunas ou linhas de registro.

## Criando Views

Virtualizações são criadas com o comando de instrução DDL (Data Definition Language) CREATE VIEW. Essa View pode ser criada com base em uma única tabela, múltiplas tabelas outras Views. A sintaxe básica de uma View é apresentada no exemplo de código a seguir (Quadro 2.1):

1. CREATE VIEW [nome\_View]

2. AS

3. [instrução SQL]

4. [with cascade] [local] [check option]

Quadro 2.1 - Criando View  
Fonte: Elaborado pelo autor

Na linha um, há o comando DDL de criação da View e o nome atribuído a essa visão. Na linha dois, há a palavra reservada AS (como), a linha três representa a instrução SQL SELECT em si. Views podem ser exploradas de diferentes maneiras. Esse processo dependerá da necessidade da visualização do negócio.

A seguir, exemplos de códigos utilizando Views.

O código do Quadro 2.2 apresenta uma simples consulta a uma tabela intitulada como ALUNO, cujo objetivo é extrair todas as informações da tabela.

1. CREATE VIEW v\_aluno

2. AS

3. SELECT \* FROM aluno;

Quadro 2.2 - Consulta múltiplas  
Fonte: Elaborado pelo autor

Mas podem ser, também, extraídos dados de uma ou mais tabelas e apresentados no Quadro 3. Nesse script, a consulta está sendo realizada em duas tabelas: a tabela de empregado e a tabela de pagamento de empregados.

1. CREATE VIEW v\_emp\_relatorio

2. AS

3. SELECT E.emp\_id, E.nome, E.email, P.cargo, P.data\_contratacao, P.sal

4. FROM

5. empregado AS E, pagamento\_e AS P

6. WHERE

7. E.emp\_id = P.emp\_id;

Quadro 2.3 – Consulta a múltiplas tabelas  
Fonte: Elaborado pelo autor.

Uma virtualização pode ser criada com base em outras visualizações, por meio de uma instrução SQL e a sua sintaxe apresentada no Quadro 2.4:

1. CREATE VIEW v\_emp\_relatorio

2. AS

3. SELECT \* FROM VIEW

Quadro 2.4 – Views de Views  
Fonte: Elaborado pelo autor.

Não crie visualizações com muita profundidade, porque pode afetar a gerenciabilidade das visualizações, pois esse problema é decorrente do cascateamento de Views. Por exemplo, suponha que você tenha uma View2, baseada numa View1 e uma terceira View3 cujas informações foram extraídas da View2. Caso a primeira View seja excluída, as visualizações 2 e 3 não servirão mais, porquanto as dependências relativas a View1 estarão destruídas.

**saiba mais**

Saiba mais

Sinônimos são, meramente, outro nome dado a uma tabela ou uma virtualização. Sinônimos são criados com o intuito de evitar que um usuário possa deter o acesso a uma tabela ou virtualização de outro usuário.

Leia a “Hora 20”, do livro Aprenda SQL em 24 horas, para saber mais sobre sinônimos.

[ACESSAR](https://books.google.com.br/books?id=c9ye91xVM5EC&pg=PA1&dq=aprenda+sql++24+horas&hl=pt-BR&sa=X&ved=0ahUKEwje8M-fuZfnAhWBHbkGHRTNDP0Q6AEIOTAC#v=onepage&q=aprenda%20sql%20%2024%20horas&f=false)

## Apagando Views

O comando SQL DROP apaga um View. Sinônimo de excluir, a palavra ‘DROPAR’ é comumente usada por profissionais da área, ou seja, quando se deseja falar “apague ou exclua uma View”, faça uso do termo “DROPE a View”. A sintaxe do comando é simples e apresentada no Quadro 2.5:

1. DROP View nome\_view

Quadro 2.5 – Exclusão de View  
Fonte: Elaborado pelo autor.

Importante: o comando DROP possui duas opções: RESTRICT e CASCADE. No uso da palavra RESTRICIT, ao apagar uma View (DROP VIEW) e qualquer outra visualização for referenciada à restrição, o comando DROP gera um erro. Ao contrário, no uso do comando DROP VIEW com a opção CASCADE, a instrução é executada com sucesso.

## Invocando Views

As Views criadas e armazenadas num banco de dados podem ser chamadas ou invocadas exatamente como é denominada uma tabela, por meio do comando SELECT. O Quadro 2.6 apresenta a sintaxe básica de chamada de uma View.

1. SELECT \* FROM nome\_View

Quadro 2.6 - Chamando view  
Fonte: Elaborado pelo autor.

A cláusula de filtro como WHERE pode ser usada na chamada de uma View, porém, para melhor desempenho, a sugestão é usar a cláusula nas instruções SELECT que formam a VIEW (PLEW; STEPHENS, 2000).

Além das Views de banco de dados, procedimentos armazenados são objetos do banco de dados fundamentais e considerados outro recurso importante. Este é o tema do nosso próximo tópico de estudo.

**praticar**

Vamos Praticar

Banco de dados também aceita visualização de tabelas Na verdade, na visão do usuário, são lidos como tabelas e executados como tabelas, porém não possuem dados físicos e são armazenadas em memória.

Sobre o tema Views e com base no enunciado, analise as afirmações a seguir:

1. O comando CREATE VIEW cria virtualizações no banco de dados.
2. Um dos benefícios de uso de View é o reuso de comando.
3. Visualizações são permitidas somente para consulta simples ou de múltiplas tabelas.
4. A cláusula WHERE pode ser usada para restringir acesso a Views.

Está correto apenas o que se afirma em:

Parte superior do formulário

**a)**II, III e IV.

**b)**I, II e IV.

**c)**I e IV.

**d)**I e II.

**e)**II e IV.

Parte inferior do formulário

**Procedimentos Armazenados**

Esses procedimentos armazenados são instruções relacionadas de SQL, que pertencem a grupos chamados funções (functions) e subprogramas (procedure) e permite que, no uso, os desenvolvedores de banco de dados tenham mais flexibilidade e facilidade.

## Procedure

Procedimentos armazenados (stored procedures) são blocos de códigos ou comandos SQL que ficam no banco de dados e podem ser executados pelo SGBD, quando necessário. Uma das vantagens do uso de procedimentos armazenados é a padronização de rotinas e automatização de processos. Outra característica interessante é que sua execução pode ser programada diretamente pelo usuário ou acionada por gatilhos.

Os procedimentos armazenados são executados no servidor, o que contribui para uma melhoria da performance de certos processos e operações. É uma característica interessante, porque muitas aplicações que interagem com o banco de dados possuem rotinas de manipulação que se assemelham às realizadas pelos procedimentos armazenados, porém essas rotinas são executadas no lado do cliente, e não no servidor. Diferentemente, os procedimentos armazenados, pelo fato de estarem do lado do servidor, além da performance, também conferem maior segurança na operacionalização de rotinas.

O termo usado pelo padrão SQL para se referir aos procedimentos armazenados é “módulos armazenados persistentes”, porque esses programas ficam armazenados persistentemente pelo SGDB (ELMASRI; NAVATHE, 2011). A criação de programas dentro do banco de dados, por meio de procedimentos armazenados, também permite uma grande flexibilização na execução de processos e administração do banco de dados.

Segundo Elmasri e Navathe (2011), os procedimentos armazenados são necessários nas seguintes circunstâncias:

* quando um programa de banco de dados pode ser utilizado por várias aplicações. Sendo armazenado no servidor, ele poderá ser invocado por todas elas, reduzindo esforço e melhorando a modularidade do sistema;
* programas executados no servidor reduzem a transferência de dados e o custo de comunicação em rede para o tráfego de dados em rede em uma estrutura cliente/servidor (Figura 2.7);
* contribuem para uma melhor eficiência e poder de modelagem de dados complexos, bem como para a manutenção e verificação de restrições difíceis, especificações e criação de gatilhos.

Em geral, os SGBD utilizam o padrão SQL para a criação de procedimentos armazenados. Entretanto, alguns podem estender essas funcionalidades adicionando recursos extras. Por isso, é muito importante estar familiarizado com o padrão SQL e eventuais extensões específicas do SGBD utilizado.

As procedure armazenadas permitem programação procedural, ou seja, permite que você, usuário desenvolvedor de banco de dados, determine o que precisa ser feito, e não como fazê-lo. Por conseguinte, instruções SQL DDL,DML e DQL são permitidas.

Quando uma procedure é criada, realmente é armazenada no banco de dados, mas também todas as funções e sub-rotinas dessa procedure, que fica disponível para ser invocada, quando for necessário. A sintaxe básica de construção de uma procedure é apresentada no Quadro 2.7:

1. CREATE [OR REPLACE] PROCEDURE [schema.]nome\_da\_procedure

2. [(parâmetro1 [modo1] tipodedado1,

3. parâmetro2 [modo2] tipodedado2,

4. ...)]

5. IS|AS

6. Bloco PL/SQL

Quadro 2.7 – Sintaxe de procedure  
Fonte: Elaborado pelo autor.

REPLACE – caso a procedure já tenha sido criada, substitua por essa;

BLOCO PL/SQL – inicia com BEGIN e termina com END;

NOME\_DA\_PROCEDURE – o nome da procedure;

PARÂMETRO – indica o nome da variável que é passada na chamada da procedure ou retornada ou ambos (passada e retornada). O parâmetro MODO determina entrada ou retorno ou ambos;

MODO – parâmetro é de entrada (IN), saída (OUT) ou ambos (IN OUT). Por padrão, sem qualquer especificação de MODO, será, automaticamente, IN;

TIPODEDADO – o tipo de dado. Uso de %TYPE, %ROWTYPE como referências ou qualquer tipo de dado escalar ou composto. Atenção: sem restrição de tamanho de tipo de dados.

IS|AS – a sintaxe do comando aceita tanto IS como AS.

Esses procedimentos podem ter variações entre alguns bancos de dados na forma declarativa da instrução SQL. O Quadro 2.8 apresenta um exemplo de declarações de procedimento.

1. CREATE OR REPLACE PROCEDURE sal\_aumento (p\_id\_emp IN emp.id\_emp%TYPE) IS

2. BEGIN

3. UPDATE

4. pedro.emp

5. SET

6. sal = sal \* 1.20

7. WHERE

8. id\_emp = p\_id\_emp;

9. END sal\_aumento;

10.     /

Quadro 2.8 – Procedure  
Fonte: Elaborado pelo autor.

No exemplo do Quadro 2.8, a finalidade da procedure é promover um aumento salarial ao funcionário Pedro de 20%. Na linha 1, a definição da procedure com definição de parâmetros de modo IN. Logo em seguida, a instrução de atualização do registro com a instrução SQL (UPDATE) e passagem de parâmetro na chamada procedure.

A chamada de uma procedure é feita pelo comando CALL ou EXEC sal\_aumento (12);. Caso queira excluir um procedimento, utilize o comando DROP nome\_PROCEDURE.

**reflita**

Reflita

Embora os Sistemas Gerenciadores de Banco de Dados (SGBD) tenham o mesmo propósito, há diversos SGBDs no mercado e, por isso, os distribuidores desses SGBDs determinam sintaxes diferentes, que implicam algumas diferenças na hora de criar procedimentos.

Portanto, os procedimentos (ou **Stored Procedure**) ficam armazenados no servidor de banco de dados e pré-compiladas à espera de uma execução pelo usuário, utilizando tarefas com comandos DML como INSERT ou UPDATE.

## Functions

As funções também são um tipo de programas armazenados no servidor e possuem características semelhantes aos procedimentos armazenados, porém com a exigência da especificação de um retorno de dado. Funções são chamadas procedures. Sua sintaxe é declarada no Quadro 2.9:

1. CREATE [OR REPLACE] FUNCTION nome\_da\_função

2. [( parameter1 [ mode1] datatype1,

3. parameter2 [ mode2] datatype2,

4. . . .)]

5. RETURN tipo\_de\_dado

6. IS|AS

7. Bloco PL/SQL;

Quadro 2.9 - Functions  
Fonte: Elaborado pelo autor.

A diferença básica entre uma função e um procedimento é que functions, ao contrário das procedures, necessariamente têm de retornar ao menos um valor. Esse valor é retornado através da palavra reservada RETURN.

O Quadro 2.10 apresenta um exemplo de functions com a finalidade de aumentar o salário em 20%.

1. CREATE FUNCTION f(SAL DECIMAL(10,2)) RETURNS DECIMAL

2. BEGIN

3. RETURN SAL \* 1.2;

4. END;

Quadro 2.10 – Exemplo de functions  
Fonte: Elaborado pelo autor.

Um função pode ser chamada pelo comando CALL ou utilizando uma instrução SQL SELECT. Exemplo do quadro 10, call f(1000.00) ou select f(1000.00). A exclusão da function é realizada pelo comando DROP nome\_function.

Tanto procedure quanto functions podem ser alteradas por meio do comando ALTER PROCEDURE | FUNCTION. É apresentada uma sintaxe básica no Quadro 2.11:

1. DROP PROCEDURE IF EXISTS sp\_Country\_UPDATE;

2. CREATE PROCEDURE sp\_Country\_UPDATE

3. ( IN p\_CountryId int,

4. IN p\_CountryName nvarchar(25),

5. IN p\_CountryDescription nvarchar(25),

6. IN p\_IsActive bit,

7. IN p\_IsDeleted bit )

8. UPDATE

9. Country

10.   SET

11.   CountryName = p\_CountryName ,

12.   CountryDescription=p\_CountryDescription,

13.   IsActive= p\_IsActive,

14.   IsDeleted=p\_IsDeleted

15.   WHERE

16.   CountryId = p\_CountryId ;

Quadro 2.11 – Alteração procedure ou functions  
Fonte: Elaborado pelo autor.

Pequenas diferenças existem nos conceitos entre procedure x functions, que são:

* funções não são executadas a partir de uma procedure, mas uma procedure pode executar uma função;
* nas funções, é possível usar em conjunto com as instruções SELECT, WHERE, HAVING, entretanto, nas procedures não é possível;
* nas procedures, é permitido tratar exceções com try/catch, porém nas functions esse tratamento não é permitido;
* pode-se utilizar Transaction Management em um procedure, mas não pode ser usada numa function;
* característica da procedure é retenção no plano de execução. Já function é compilada toda vez que é invocada;
* function pode ser incorporada numa instrução SQL SELECT; já na procedures esse procedimento não é permitido.

Algumas situações em que o uso de função pode ser a solução ideal incluem:

* autenticação de usuários;
* obtenção do perfil dos usuários;
* envio de mensagens aos usuários de um banco de dados;
* produção de dados complexos, que dependem da interação de diversos elementos do banco de dados;
* obtenção de dados contextuais, tais como: data, hora, dia da semana etc.

Além dos procedimentos armazenados no banco de dados (functions e procedures), gatilhos (trigger) estão muito associados a regras de negócio. São também armazenados como objetos do banco de dados e considerados outro recurso importante. Esse é o tema do nosso próximo tópico de estudo.

**praticar**

Vamos Praticar

Procedure e functions são objetos armazenados no banco de dados que otimizam processos de manipulação de dados armazenados, provendo uma flexibilidade e facilidade para os administradores de banco de dados. Assinale a alternativa que descreve corretamente a relação entre functions e procedures.

Parte superior do formulário

**a)**O retorno nas functions é obrigatório, porém na procedures essa obrigatoriedade não existe.

**b)**Procedures podem executar functions, mas funções não executam procedures.

**c)**Tanto functions quanto procedures podem ser usadas com as instruções SELECT e WHERE.

**d)**Gerenciamento de transações pode ser usado com functions e procedure.

**e)**A cláusula HAVING pode ser usada tanto em functions quanto em procedure.

Parte inferior do formulário

**Gatilhos**

Uma das formas de automatizar um sistema gerenciador de banco de dados é usando gatilhos (ou triggers). Os gatilhos podem ser definidos como o uso de regras previamente definidas, que são ativadas por eventos que ocorrem na tabela. Esses eventos acionam operações adicionais – como procedimentos armazenados (stored procedures) – a serem realizadas em algumas outras tabelas e podem estar relacionadas ao envio de mensagens, inclusão, alteração, exclusão de dados, dentre outras (ELMASRI; NAVATHE, 2011).

Os gatilhos também podem ser definidos como blocos de instruções SQL acionados ou disparados automaticamente para a realização de operações DML (PUGA; FRANÇA; GOYA, 2013).

O uso do recurso de gatilhos torna o SGBD um ambiente ativo, cujas transações que são realizadas nas tabelas podem desencadear outras transações ou operações. É um recurso fundamental relacionado ao processo de automação do banco de dados. Permite que tarefas sejam executadas de forma automática, melhorando a eficiência do SGBD e reduzindo o tempo na realização de ações que poderiam ser repetitivas do ponto de vista dos usuários.

Um bom planejamento e definições adequadas no uso de gatilhos podem não apenas dar maior eficiência ao banco de dados como também contribuir para a redução de erros, o que promove, significativamente, qualidade das informações armazenadas e processadas pelo SGBD.

Um gatilho pode ser usado para a verificação da integridade de dados antes de uma instrução DML (INSERT, DELETE, UPDATE), podem reverter (ROLLBACK) transações, modificar dados numa tabela e ler de uma tabela para outro banco de dados (PLEW; STEPHENS, 2000).

Gatilhos, de um modo geral, são ferramentas úteis dentro do banco de dados. Todavia, podem gerar mais sobrecarga de entrada e saída (E/S), e não precisam ser usados quando um procedimento pode ser executado por procedure ou um programa com mesmo resultado (PLEW; STEPHENS, 2000).

## Criando Gatilhos

Os gatilhos são criados por meio da linguagem SQL, e o comando utilizado é o CREATE TRIGGER. A sintaxe desse comando pode ter algumas variações, dependendo do SGBD em uso. Porém, para fins didáticos, vamos adotar o modelo PL/SQL, padrão utilizado pelo SGBD Oracle. Segundo Puga, França e Goya (2013), no padrão PL/SQL, o comando CREATE TRIGGER é apresentado no Quadro 2.12:

1. CREATE [OR REPLACE] TRIGGER [esquema.] nome\_trigger

2. {BEFORE ou AFTER}

3. [evento] ON [esquema.]tabela\_nome

4. [referencing OLD as valor\_anterior ou NEW as valor\_novo]

5. {nível de linha ou nível de instrução}

6. [WHEN (condição)]

7. DECLARE

8. declarações

9. BEGIN

10.     corpo\_trigger

11.     END;

Quadro 2.12 – Sintaxe de trigger  
Fonte: Puga, França e Goya (2013).

Em que:TE

* CREATE [OR REPLACE] TRIGGER: criação/substituição do gatilho;
* esquema: nome do esquema a qual o objeto pertence;
* nome\_trigger: nome/identificador do gatilho;
* BEFORE: indica que a execução do gatilho ocorre antes do evento;
* AFTER: indica que a execução do gatilho ocorre depois do evento;
* evento: determina qual evento DML disparará o gatilho;
* ON tabela\_nome: tabela a qual o gatilho será associado;
* referencing OLD as valor\_anterior ou NEW as valor\_novo: define os identificadores das variáveis;
* nível de linha: disparado uma vez a cada linha afetada pela instrução identificada por FOR EACH ROW;
* nível de instrução: disparado antes ou depois da instrução;
* WHEN (condição): determina que o gatilho somente será disparado quando a condição for satisfeita (apenas para trigger de linha).

Suponhamos que se deseje emitir uma mensagem de aviso **“Não é possível registrar o pagamento fora do prazo”**, toda vez que um pagamento com vencimento atrasado em 30 dias ou mais seja registrado, o gatilho poderia ser escrito da seguinte forma:

CREATE OR REPLACE TRIGGER PAGTO\_FORA\_DO\_PRAZO

BEFORE INSERT ON PGTO

FOR EACH ROW

DECLARE

vencimento FATURA.data\_venc%type;

BEGIN

SELECT data\_venc INTO vencimento

FROM FATURA, PGTO

WHERE FATURA.numero = :new.numero;

IF (SYSDATE - vencimento) >= 30 then

RAISE\_APPLICATION\_ERROR(-20202,"Não é possível registrar o pagamento fora do prazo");

END IF

END;

Quadro 2.13 – Exemplo de gatilho

Fonte: Puga, França e Goya (2013).

Observe que o gatilho PAGTO\_FORA\_DO\_PRAZO será acionado antes da inclusão de um registro na tabela PGTO, uma vez verificado que o vencimento é superior ao período informado na condição (PUGA; FRANÇA; GOYA, 2013).

## Componentes do Gatilho

Um gatilho é formado por três componentes principais: os eventos, a condição e a ação (VICCI, 2014). Vamos conhecê-los?

# GATILHO

Esses componentes que especificam as regras de ocorrência ou comportamento de gatilhos são conhecidos como Evento-Condição-Ação (ECA).

Além desses componentes, algumas questões adicionais estão relacionadas ao uso de gatilhos para a automatização de um SGBD. Um SGBD ativo deve permitir aos seus usuários, por meio de recursos disponíveis, criar, ativar, desativar ou remover regras, quando referir-se a seus nomes.

Regras desativadas não devem ser disparadas pelo evento, mesmo que a condição seja atendida explicitamente. É um recurso importante, que dá liberdade ao usuário para, seletivamente, desativar certas regras por períodos de tempo quando elas não são necessárias. O processo de ativação é o recurso que permitirá ao usuário voltar a ativar uma regra previamente desativada, quando entender que seu funcionamento se tornou novamente necessário.

Já o processo de remoção deve permitir ao usuário excluir uma regra que se tornou desnecessária ao sistema (ELMASRI; NAVATHE, 2011).

**reflita**

Reflita

Os gatilhos podem ser utilizados para automatizar diversas aplicações, como manutenção de coerência, monitoramento de atualizações e envio de mensagem sobre ocorrências no banco de dados.

Fonte: Vicci (2014).

## Exemplo de Aplicações com Gatilho

Uma importante aplicação dos gatilhos está relacionada ao processo de notificação de certas condições que ocorrem conforme regras estabelecidas. Dessa forma, um banco de dados ativo, por exemplo, pode ser utilizado para o monitoramento da temperatura de uma fornalha industrial, fazendo leituras periódicas a partir dos dados gerados por sensores e inseridos no banco de dados. Um gatilho pode ser escrito para analisar esses registros e disparar uma mensagem de alerta ou perigo toda vez que a temperatura exceder certo nível de segurança.

Outra aplicação importante está relacionada às restrições de integridade, por meio da criação de regras que verifica a violação de certas condições de restrição. Esse tipo de aplicação é muito importante e pode estar alinhado às regras de negócio da organização e suas restrições. Um exemplo seria um banco de dados ativo em uma Universidade, que verificaria toda vez que uma nota fosse inserida, alertando o Conselho quando a média ficasse abaixo de determinado valor.

Outro exemplo está relacionado à aplicação referente à manutenção automática de dados derivados. É muito útil quando se deseja que determinados valores sejam gerados com base em outros. Um conjunto de dados usado por uma área estratégica da organização pode ser atualizado a cada venda gerada pela empresa ou para cada pagamento recebido, por exemplo.

De forma simplificada, podemos dizer que o uso de gatilhos é indicado para:

* verificação de integridade de dados;
* validação de dados;
* rastreamento e registro de logs relacionados às atividades do banco de dados;
* arquivamento de registros excluídos.

Apesar das vantagens apresentadas em relação ao seu uso, alguns profissionais preferem restringir o mínimo possível o uso de gatilhos. Dentre os argumentos contrários ao uso de gatilhos, está o de que eles geralmente fazem uso de uma linguagem proprietária, ou seja, utilizam uma versão da linguagem SQL específica do banco de dados. E isso poderia ser um problema para uma eventual migração de banco de dados, uma vez que haveria incompatibilidade de comandos entre os SGBDs.

Outro argumento está relacionado à atualização de versão do SGBD ou upgrade. A prática tem demonstrado que empresas que comercializam SGBDs, muitas vezes ao fazerem atualizações, não levam em conta a estrutura de gatilhos entre uma versão e outra, criando incompatibilidade ou conflitos entre as versões.

No processo de automatização de um banco de dados, outro fator a ser considerado é a sua segurança, assunto a ser tratado no próximo tópico.

**praticar**

Vamos Praticar

Na análise de um negócio, as regras de negócio são requisitos fundamentais, pois determinam muito dos fluxos de processos dentro de um sistema. Trigger ou gatilhos são fundamentais quando se deseja aplicar regras num sistema de banco de dados. Sobre triggers, assinale a alternativa correta:

Parte superior do formulário

**a)**Gatilhos podem ser disparados somente após um comando SQL SELECT.

**b)**Antes, após ou ambos de um evento, um gatilho pode disparar ações.

**c)**Gatilhos não podem ser usados para garantir a integridade de dados.

**d)**Ações como envio de e-mail somente são permitidas após um evento.

**e)**Gatilhos são disparados manualmente após um comando de transação.

Parte inferior do formulário

**Segurança do Banco de Dados**

Na Era da Informação, temos presenciado uma explosão de sistemas informacionais que demandam cada vez mais dados para suas operações. Muitos dados pessoais que antes se limitavam a documentos físicos, guardados em gavetas, malas e armários, hoje se encontram armazenados em meios digitais e em diferentes SGBDs espalhados pelo mundo. Muitas vezes, nem nos damos conta de quantos sistemas possuem nossas informações. Sistemas como redes sociais, cadastros em universidade, biblioteca, governo, empresas etc.

Quando o assunto é segurança de banco de dados, podemos ver a questão por diferentes óticas, que compreendem desde níveis de controle de acesso, que estão muito mais relacionados a políticas de segurança da própria organização, recursos oferecidos pelo SGBD, até à forma como eles podem estar relacionados a aspectos físicos de infraestrutura.

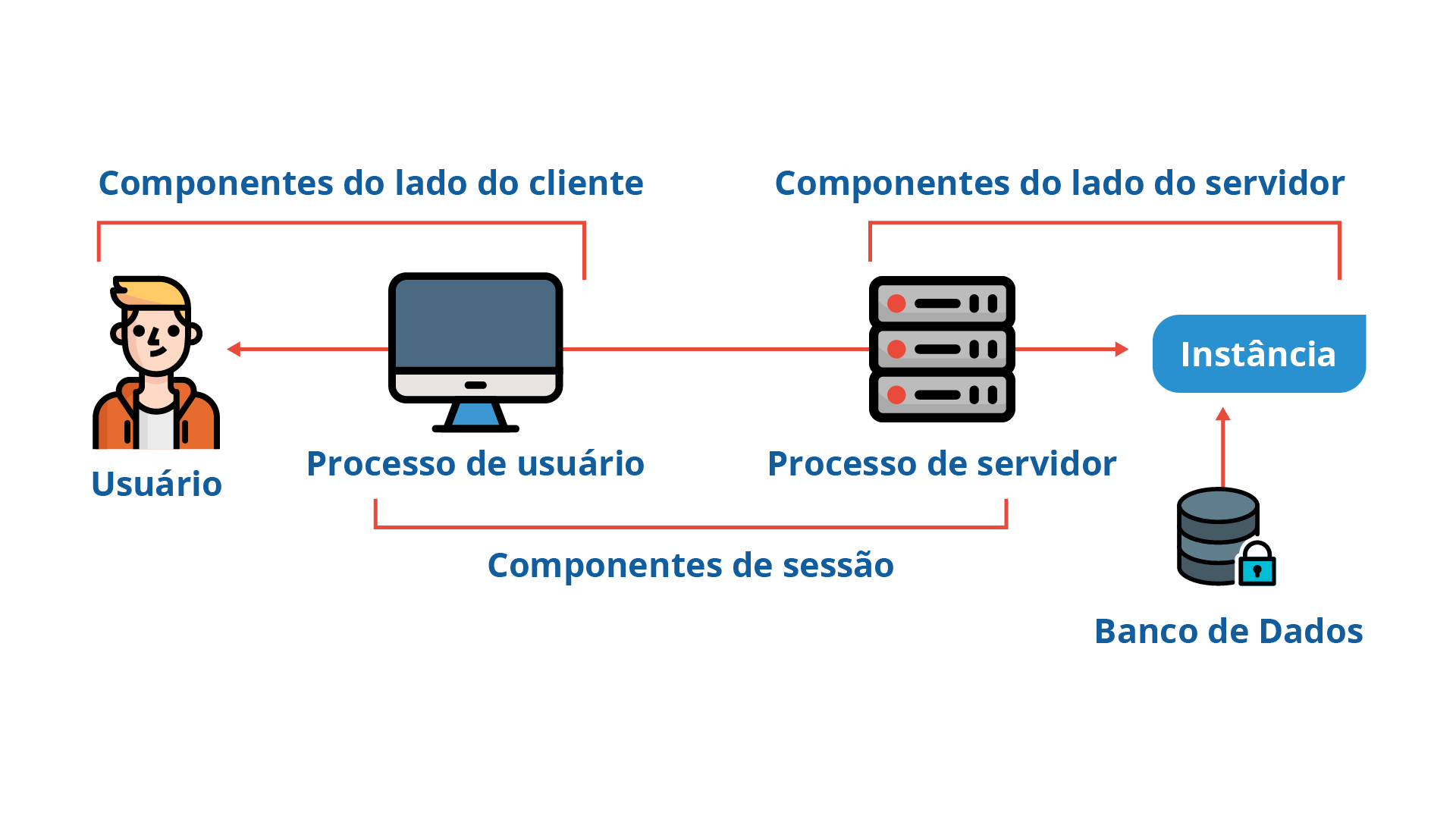
A segurança, dentro de um SGBD, pode ser aplicada aos dados que estejam registrados nas tabelas pertencentes ao banco, o que, aliás, é um ponto vital das políticas de segurança em bancos de dados. As questões de segurança também podem, e devem, ser estendidas às stored procedures, funções e aos pacotes, pois eles são elementos que podem ser ativados diretamente (ou indiretamente) pelos usuários. Quando se dá permissões de acesso a um pacote, tudo que estiver dentro dele será afetado. Caso os códigos não estejam agrupados em pacotes, as considerações de permissão deverão ser realizadas para cada stored procedures ou função. A segurança do banco de dados abrange aspectos externos e internos à empresa. Sendo assim, trata-se de um tema complexo e de extrema importância, o que exige planejamento e organização.

As questões de segurança podem ser de ordens legais e éticas, uma vez que se relacionam ao direito de acesso a determinadas informações. Podem ser informações consideradas particulares, cuja privacidade pode estar coberta por leis nacionais, por exemplo, as diversas leis americanas que controlam a privacidade da informação.

A segurança pode ser de natureza política, uma vez que trata das questões sobre informações que não devem ser públicas, como crédito ou registros médicos pessoais. Também podem ser de nível governamental, institucional ou corporativo. Já os níveis de segurança relacionados às funções de segurança do próprio sistema de software e hardware envolvem políticas de segurança da organização, tais como: portas com autenticação, uso de crachás, restrições físicas para o acesso etc.

De forma semelhante, os níveis de segurança relacionados aos níveis de acesso aos dados do SGBD, inclusive os tipos de acesso vinculados a esses níveis, devem fazer parte de uma política de segurança desenvolvida pela organização.

Há, também, de ser considerado o caso em que o usuário em si não faz (ou faz indiretamente) acesso ao SGBD. Isso ocorre quando o usuário usa uma aplicação e é esta quem acessa o SGBD, como apresentado na Figura 2.1:

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html)

*Figura 2.1 - Relação que mostra como o usuário atua indiretamente com o SGBD, no caso, o acesso ao banco é feito por um aplicativo e não diretamente pelo usuário  
Fonte: Watson e Ramklass (2010, p. 4).*

Nesse caso, as credenciais usadas não são necessariamente as do usuário, mas podem ser uma interna, mantida pelo aplicativo. Pode ocorrer de o usuário ter menos permissão do que o usuário interno usado pelo aplicativo, ou ainda, o contrário. O ideal seria que o sistema dependesse exclusivamente do perfil e dos limites de acesso ao usuário. Todavia, isso introduz um enorme problema no desenvolvimento dos aplicativos.

Com relação à segurança do próprio sistema, devemos levar em consideração a infraestrutura sobre a qual o sistema está sendo executado. É importante porque eventuais falhas podem comprometer as informações em questões de integridade, disponibilidade e confidencialidade. Por isso, as questões de segurança relacionadas à infraestrutura devem prever situações que abordem temas relacionados à redundância e backup.

Os backups são fundamentais para garantir a recuperação de informações no caso de falhas de hardware. Os processos e políticas de backups podem ser automatizados, para garantir maior segurança às informações.

Em relação à segurança do banco de dados, é de responsabilidade do administrador do banco (DBA) gerenciar e conceder privilégios de acesso aos usuários do sistema, de acordo com as políticas da organização. Fazem parte da rotina de trabalho de um DBA: a criação de contas de usuário, com as devidas permissões de acesso ao SGBD, concessão de privilégios a determinados usuários ou grupos de usuários, revogação de privilégio e atribuição de nível de segurança.

As concessões ou revogações de privilégios e sua propagação no banco de dados são feitas por meio dos comandos Data Control Language (DTL): GRANT e REVOKE. Supondo-se a criação das contas C1, C2 e C3 em que o DBA deseja conceder o privilégio para a conta C1 poder criar tabelas no banco de dados, ela poderá ser feita com o seguinte comando: GRANT CREATETAB TO C1.

Supondo-se que se deseje apenas conceder o privilégio para a conta C2 poder inserir ou deletar registros da tabela ALUNO, então, poderia fazer com o seguinte comando: GRANT INSERT, DELETE ON ALUNO TO C2.

Para a concessão de privilégio de usar o comando SELECT, a conta C3 poderia usar o seguinte comando: GRANT SELECT ON ALUNO TO C3.

Para se fazer a revogação desse privilégio poderia utilizar o comando: REVOKE SELECT ON ALUNO FROM C3.

O controle de acesso também pode se dar por meio de papéis, em que é possível conceder níveis de privilégios a grupos de usuários que estão relacionado a um determinado papel. Esse tipo de controle facilita muito o trabalho do DBA, principalmente em sistema com um grande número de contas de usuários. A criação e remoção de papéis se dão por comandos CREATE ROLE e DESTROY ROLE. E a concessão de privilégios aos papéis ocorre da mesma forma como é feito com as contas, com o uso dos comandos GRANT e REVOKE. Como exemplo de criação do papel “gerente”, atribuindo-lhe certos privilégios, poderíamos usar os comandos:

CREATE ROLE gerente

GRANT INSERT, DELETE, SELECT ON ALUNO TO gerente

Com o advento da internet e um grande volume de aplicações indo para a nuvem, novas necessidades têm surgido para a implementação de segurança e controle de acesso, o que tem levado os administradores de banco de dados a pensarem em soluções de integração com sistemas por meio do uso de XML. Assim como políticas de segurança, para evitar tentativas de acesso indevido aos dados por meio de técnicas conhecidas como Injeção de SQL.

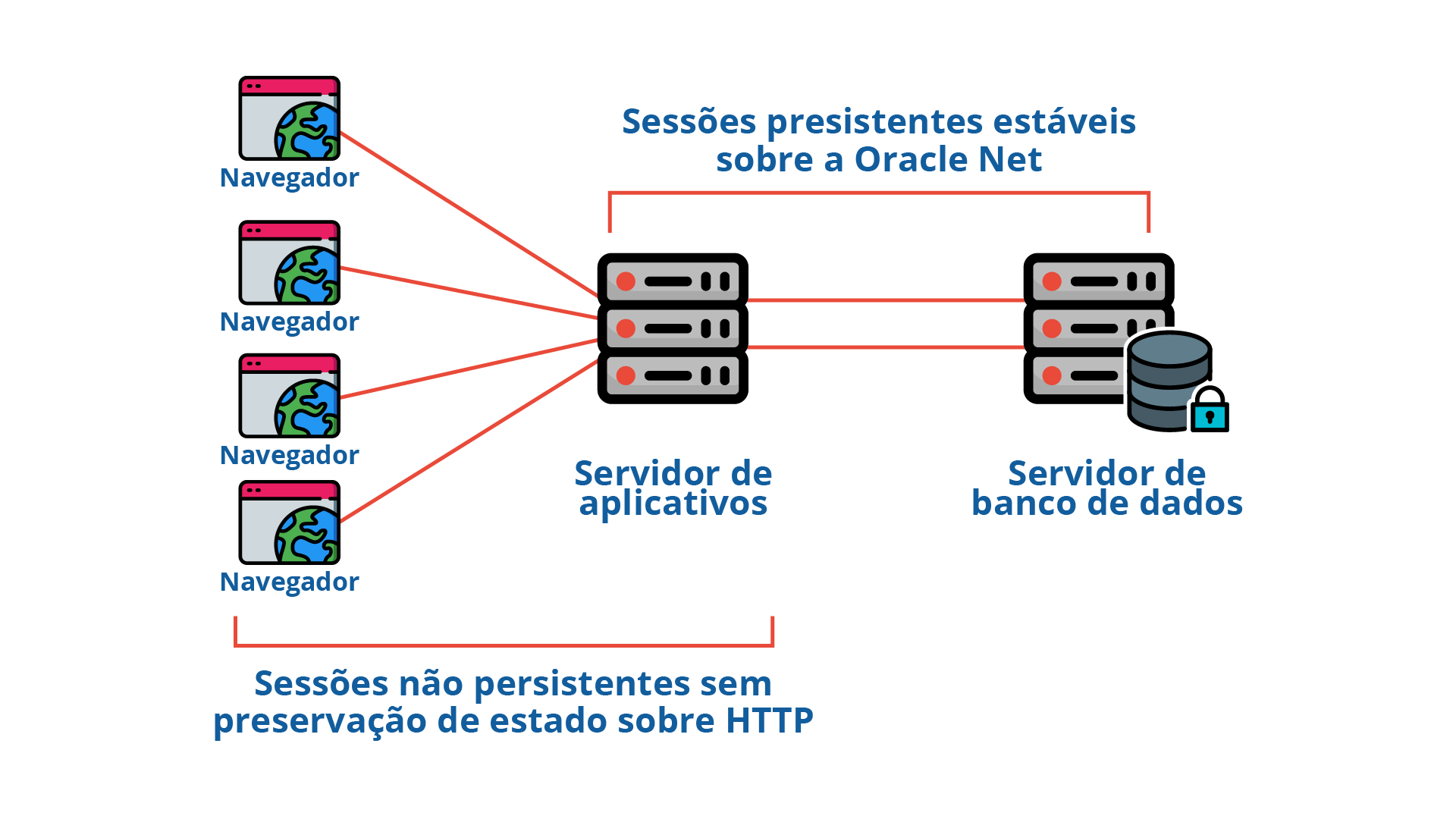
## Injeção de SQL

Os ataques por meio de Injeção SQL (SQL Injection) ocorrem quando usuários mal-intencionados inserem códigos SQL no meio de cadeias de caracteres que são enviadas aos bancos de dados por meio de browsers web.

O DBA precisa estar atento a todos esses tipos de situação para criar os devidos bloqueios e garantir a segurança do banco de dados. Embora seja simples, trata-se de um ataque antigo e muito utilizado. Muitos sites ou sistemas não possuem uma política forte de verificação para a prevenção de ocorrências dessas brechas (W3SCHOOL, 2020). Para efeito de exemplificação de como um código SQL pode ser inserido em uma página ou sistema, vamos considerar o seguinte campo: Código do Usuário: [10 OR 1=1].

Nesse caso, ao invés de ser informado o código solicitado, foi digitado, no campo, a expressão “10 OR 1=1”. Quando esse campo for processado pelo sistema e submetido ao SGBD, receberá um comando como: **SELECT \* FROM usuario WHERE codigo = 10 OR 1=1.**

Observe que o comando SQL é válido e retornará todos os usuários da tabela, uma vez que a condição 1=1 será sempre verdadeira. Essa situação pode ser implementada em um cenário web (Figura 2.2), ambiente usado por diversos navegadores, acessando aplicações que atuam sobre um servidor de banco de dados.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html)

*Figura 2.2 - Cenário típico de uma aplicação web, no qual diversos clientes (navegador) usam um banco de dados de maneira indireta  
Fonte: Watson; Ramklass (2010, p. 5).*

Esse é apenas um pequeno exemplo de como são exploradas as vulnerabilidades de inserção de código SQL, entretanto, é preciso ter em mente que outros meios de inserção podem ser utilizados, inclusive com instruções mais elaboradas e complexas. O mais importante é que medidas de prevenção, como validação prévia dos dados inseridos em um formulário, sejam executadas. Esse cenário de injeção de código SQL é ainda mais preocupante, com o uso intenso que se faz da web e dos bancos de dados.

**praticar**

Vamos Praticar

Segurança é um tópico muito explorado pelas empresas, principalmente na era da Internet, visto que a maioria da empresas já possuem suas aplicações desenvolvidas nas plataformas Web ou mobile. Essas aplicações colam informações de usuário por meio de formulários. Assinale a alternativa que apresenta um problema de segurança causado diretamente por formulários Web ou mobile.

Parte superior do formulário

**a)**SQL Injection.

**b)**Engenharia social.

**c)**Phishing.

**d)**Spam.

**e)**DoS.

Parte inferior do formulário

**indicações**

**Material Complementar**



**LIVRO**

#### SQL e Teoria Relacional

C. J. Date

**Editora:**Novatec

**ISBN:**978-85-7522-433-5

**Comentário:**Esse é um livro que todo administrador de banco deveria ler. Tem uma abordagem muito interessante sobre a linguagem SQL e suas armadilhas e mostra como solucioná-las, colocando em prática a teoria relacional.



**FILME**

#### O homem que mudou o jogo / Moneyball

**Ano:**2011

‍**Comentário:**Esse filme relata a importância do uso de dados armazenados num banco de dados para superar desafios. Por meio da análise desses dados, questões são levantadas no filme, como financeiras, estratégia de negócios, tomada de decisões importantes e gestão de expectativas.

[TRAILER](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_2/ebook/index.html)

**conclusão**

**Conclusão**

Nesta unidade, você compreendeu que tarefas rotineiras e repetitivas geram maior demanda de tempo para sua execução, tornam-se monótonas e dão margem para erros, mas podem ser automatizadas. No nosso contexto de SGBD, o processo de automatização no banco de dados é um recurso muito útil e necessário. Quando planejado e implementado de forma correta dentro do SGBD, propiciará uma significante redução nos erros e, ao mesmo tempo, aumentará a confiança geral no sistema. Além desses aspectos positivos, a automação libera o administrador para cuidar de outras tarefas, pois as atividades de manutenção automatizadas representam uma grande economia de tempo e esforço do profissional.

Neste capítulo, você teve a oportunidade de:

* compreender a importância dos recursos relacionados ao processo de automatização de banco de dados;
* compreender o conceito de procedimentos armazenados e funções;
* apontar os comandos utilizados para a criação de procedimentos armazenados e funções;
* aplicar o conceito de gatilhos e como são implementados em um sistema de banco de dados;
* analisar as políticas de segurança relacionadas aos bancos de dados e o importante papel do DBA na gestão e implementação de políticas de segurança.

**referências**

**Referências Bibliográficas**

ELMASRI, R.; NAVATHE, S. B. **Sistema de Banco de Dados.**6. ed. São Paulo: Pearson Education do Brasil, 2011.

HOW to alter a stored procedure in mysql. **Stack Overlfow**, 2013. Disponível em: <https://stackoverflow.com/questions/13970258/how-to-alter-a-stored-procedure-in-mysql>. Acesso em: 11 jan. 2020.

PLEW, R. R.; STEPHENS, R. K. **Aprenda em 24 horas SQL.** 2. ed. Rio de Janeiro: Campus, 2000.

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

VICCI, C. **Banco de Dados.**São Paulo: Pearson Education do Brasil, 2014.

XML Tutorial. **W3schools**, 2020. Disponível em: <https://www.w3schools.com/xml/default.asp>. Acesso em: 11 jan. 2020.

WATSON, J.; RAMKLASS, R. **OCA – Oracle Database 11g:** fundamentos I SQL. Manual de exame. Rio de Janeiro: Alta Books, 2010.

© 2020 - **LAUREATE**- Todos direitos reservados

**Práticas de Banco de Dados**

Práticas de Banco de Dados

1.00

**PRÁTICAS DE BANCO DE DADOS**

PRÁTICAS DE BANCO DE DADOS

Autor: Me. Paulo Sérgio Pádua de Lacerda

[INICIAR](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html#section_1)

**introdução**

Introdução

Seja bem-vindo(a) à disciplina de Prática de Banco de Dados! A proposta principal desta unidade é definir conhecimentos sobre Big Data. Em um mundo competitivo e digital, empresas buscam por informações que possam garantir liderança de mercado, tomadas de decisões mais assertivas, conhecimento sobre cliente e soluções inovadoras, a informação se torna um bem precioso. Quem detém a informação está à frente do mercado, consegue entender tendências, abrir novos mercados etc. Nesse ponto, tecnologias como Big Data e Mineração de Dados, permitem que empresas possam compreender toda essa ebulição de informações existente, principalmente na internet.

Você ficará familiarizado com tópicos a respeito do conceito de Big Data, entenderá a finalidade da mineração de dados e se familiarizar com exemplos de aplicações usando tais tecnologias. Em suma, você vai identificar e compreender os princípios da tecnologia de Big Data e Mineração de Dados. Vamos lá?

**Big Data**

Nos tempos atuais, com o surgimento de novas configurações e maneiras de fazer uso dos recursos e serviços da tecnologia da informação, as empresas e instituições precisam repensar e transformar o modo como utilizam esses dados e informações para auxiliar e fundamentar seu processo de tomada de decisão e, assim, conduzir seus negócios e alcançar os objetivos e resultados esperados.

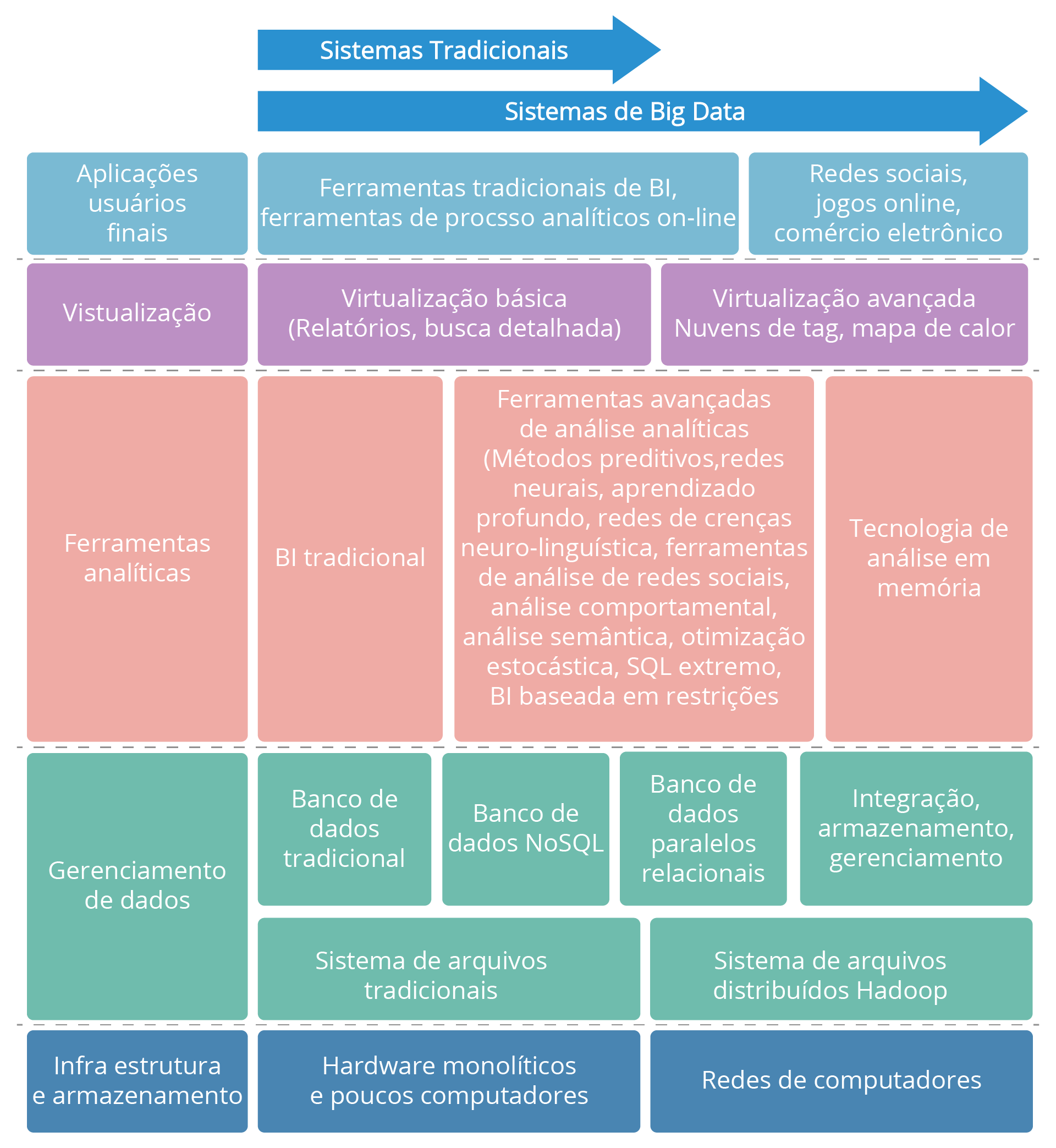
Um dos desafios das empresas na Era da Informação na qual vivemos é saber como lidar com a quantidade de dados disponíveis e como usá-los de maneira útil, ou seja, como transformá-los em informações que serão usadas nos processos de decisões e, assim, possibilitar a vantagem competitiva.

Pode-se dizer que Big Data é um conjunto de soluções de tecnologia que possuem a capacidade de lidar com uma imensa quantidade de dados, de tipos antes não explorados, presentes no dia a dia, sejam dados estruturados, semiestruturados ou não estruturados.

Para Siewert (2013 apud FERNANDES; ABREU, 2014, p. 557),

*[...] Big Data é a captura, gerenciamento e a análise de dados que vão além dos dados tipicamente estruturados, que podem ser consultados e pesquisados através de bancos de dados relacionais. Frequentemente são dados obtidos de arquivos não estruturados como vídeo digital, imagens, dados de sensores, arquivos de logs e de qualquer tipo de dados não contidos em registros típicos com campos que podem ser pesquisados [...].*

A Figura 3.1 elucida diferenças entre um sistema de Big Data e os sistemas tradicionais.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.1 - Tradicional x Big Data  
Fonte: Adaptada de Pasupuleti (2015, p. 33).*

Os sistemas tradicionais (*tradicionais systems*) e sistema de Big Data (*big data systems*), como mostrado na Figura 3.1, possuem tanto características comuns quanto diferentes.

As aplicações usuários finais (*end-user applications*), nos sistemas tradicionais (*traditional systems*), são compostas por ferramentas de BI tradicionais (*Traditional BI suites*) e ferramentas de processo analíticos on-line (*OLAP applications*), mas elas também transitam no mundo do Big Data, somando-se às redes sociais (social media), jogos on-line (*Online gaming*) e sistemas de comércio eletrônico (*e-commerce*).

Com relação à virtualização (*virtualization*), os sistemas tradicionais executam virtualização básica (*basic virtualization*), relatórios (*report*) e busca detalhada (*drilldown*), visto que os sistema de big data executam virtualização avançada (*advanced virtualization*), nuvem de tag (*tag cloud*) e mapa de calor (*heat map*).

Com relação às ferramentas analíticas (*analytics tools*), os sistemas tradicionais usam ferramentas de BI tradicionais, modelos de análise estatística multivariada (*multivariate models*) e regressão (*regression*). Já as ferramentas avançadas de análise analíticas (*advanced analytics*) transitam entre sistemas tradicionais e big data, ou seja, são comuns a ambos os sistemas, podendo ser usados método preditivos (*predictive methods*) como redes neurais (*neural nets*), um de aprendizado de máquinas chamado de aprendizado profundo (*deep learning*), redes de crenças neurolinguísticas (*belief nets Neuro-linguistic programming - NLP*), ferramentas de análise de redes sociais (*Social NW analysis*), análise comportamental (*behavior analytics*), análise semântica (*semantic analytics*), otimização estocástica (*stochastic optimization*), SQL (*extreme SQL*) e BI baseada em restrições (*constraint based BI*). Já as relativas somente ao sistema de big data são a tecnologia de análise em memória (*in-memory analytics*) e processamento de eventos complexos (*Complex Event Processing - CEP*).

A relação com os gerenciados de banco de dados (*Data Management*) está nos sistemas tradicionais e sistemas gerenciadores de banco tradicionais (Traditional Database), sistema de banco NoSQL (*NoSQL database*) e sistema de arquivos convencionais (*Conventional filesystem*). Porém, essa relação com o sistema de big data é diferente, porque fazem uso do sistema distribuído Hadoop (*Hadoop distributed file system*), sistemas paralelos de banco de dados relacionais (*parallel relational DBs*) ou banco de dados de processamento massivo de informações (*massively parallel processing - MPP*), agregado à modelagem de dados (*data modeling*), integração, armazenamento e manipulação de dados (*integration, storage, management*).

Na infraestrutura e armazenamento (*infra and storage*), os sistemas tradicionais usam hardwares monolíticos (*monolithic hardwares*), ou seja, poucos computadores (*few hardwares*) e redes de computadores (*network computers*). Nesse contexto, os sistemas de big data trabalham em redes de arquitetura distribuída (*distributed hardware*), CPUs com múltiplos núcleos de processadores (*MULTICore CPUs*), ambientes virtualizados (*virtualized environments*) e redes de alto desempenho (*high performance networks*).

**Os “Vs” do Big Data**

A tecnologia do Big Data é fundamentada em, basicamente, três pilares: velocidade, volume e variedade, conhecida como 3Vs. Entretanto, alguns autores incluem mais dois Vs: valor e veracidade (TAYLOR-SAKYI, 2016).

As cinco dimensões (pilares) do Big Data podem ser descritas por:

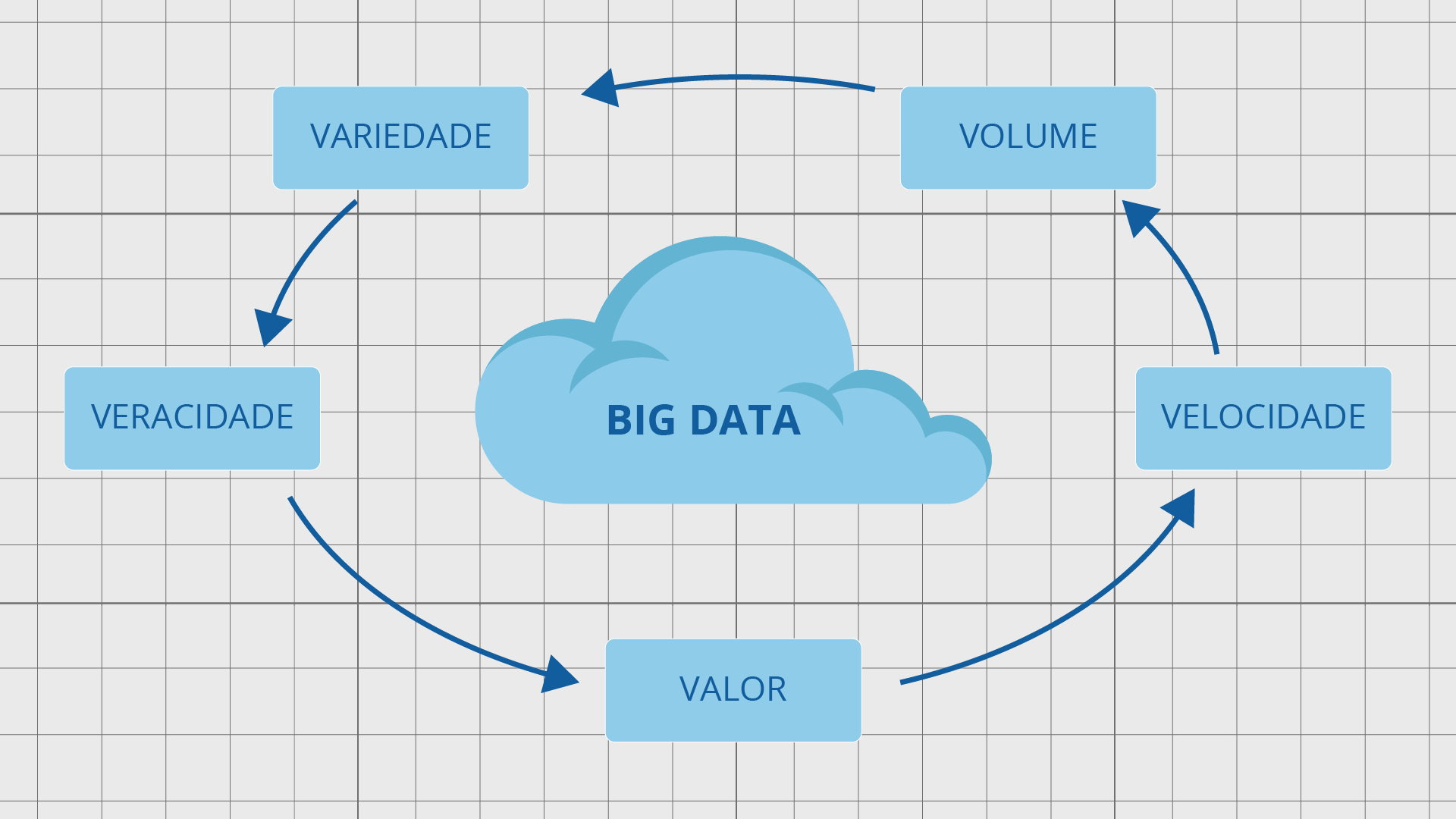
* volume: os dados aumentam exponencialmente, nas casas de Petta, Zetta, Yotta Bytes. Esse aumento de volume foi impulsionado pelas redes sociais e pelo acesso à internet por grande parte da população;
* variedade: essa massa significativa de dados é formada pelos mais variados tipos de arquivos, desde um simples arquivo de texto a complexos arquivos multimídia;
* velocidade: está associada à velocidade que informações são produzidas diariamente na internet, graças ao crescente uso de aparelhos conectados à internet. O usuário passou de somente um leitor a fornecedor de informações. As informações são geradas, principalmente, pelas redes sociais;
* veracidade: confirmação se a informação (dado) corresponde à realidade do mundo real;
* valor: qual o verdadeiro valor do dado para empresa.

**reflita**

Reflita

A agência nacional de segurança americana consegue analisar 1,6% do tráfego global de dados, cerca de 30 milhões (30 peta bytes) todos os dias. Você consegue imaginar o volume de dados a 100%?

A Figura 3.2 apresenta uma explicação macro dos cinco pilares do Big Data.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

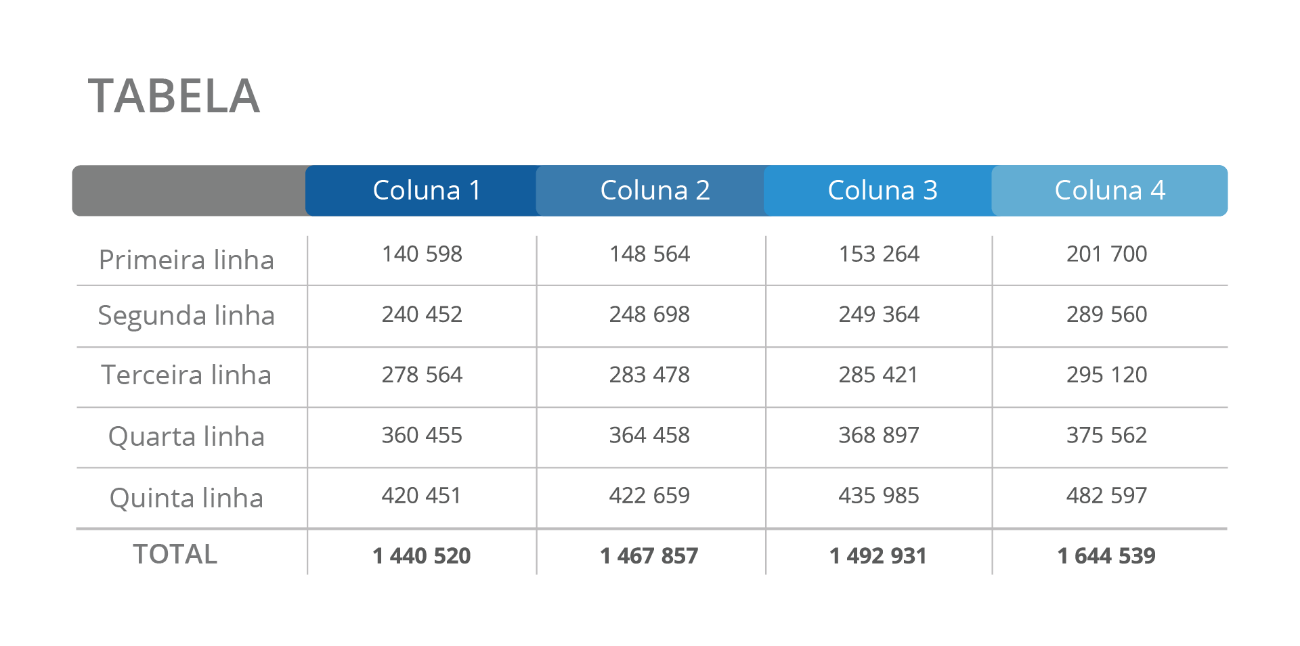
*Figura 3.2 - Cinco pilares  
Fonte: 1tjf  / 123RF.*

A Figura 3.2 centraliza o Big Data em cinco pilares: variedade (*variety*) de dados, velocidade (*velocity*) de produção de dados, volume (*volume*) de dados, valor agregado aos dados ao negócio (value) e veracidade (*veracity*) da informação, ou seja, o quanto a informação é verdadeira com relação a sua aplicação no mundo real.

Esses princípios estão solidificados na ideia de que os dados que transitam no mundo são gerados em alta velocidade, em grande quantidade e nos mais variados tipos de arquivos. Entretanto, à medida que as organizações remodelam seus processos operacionais, necessitam de dados confiáveis, verdadeiros e que tenham valor para seus planejamentos estratégicos.

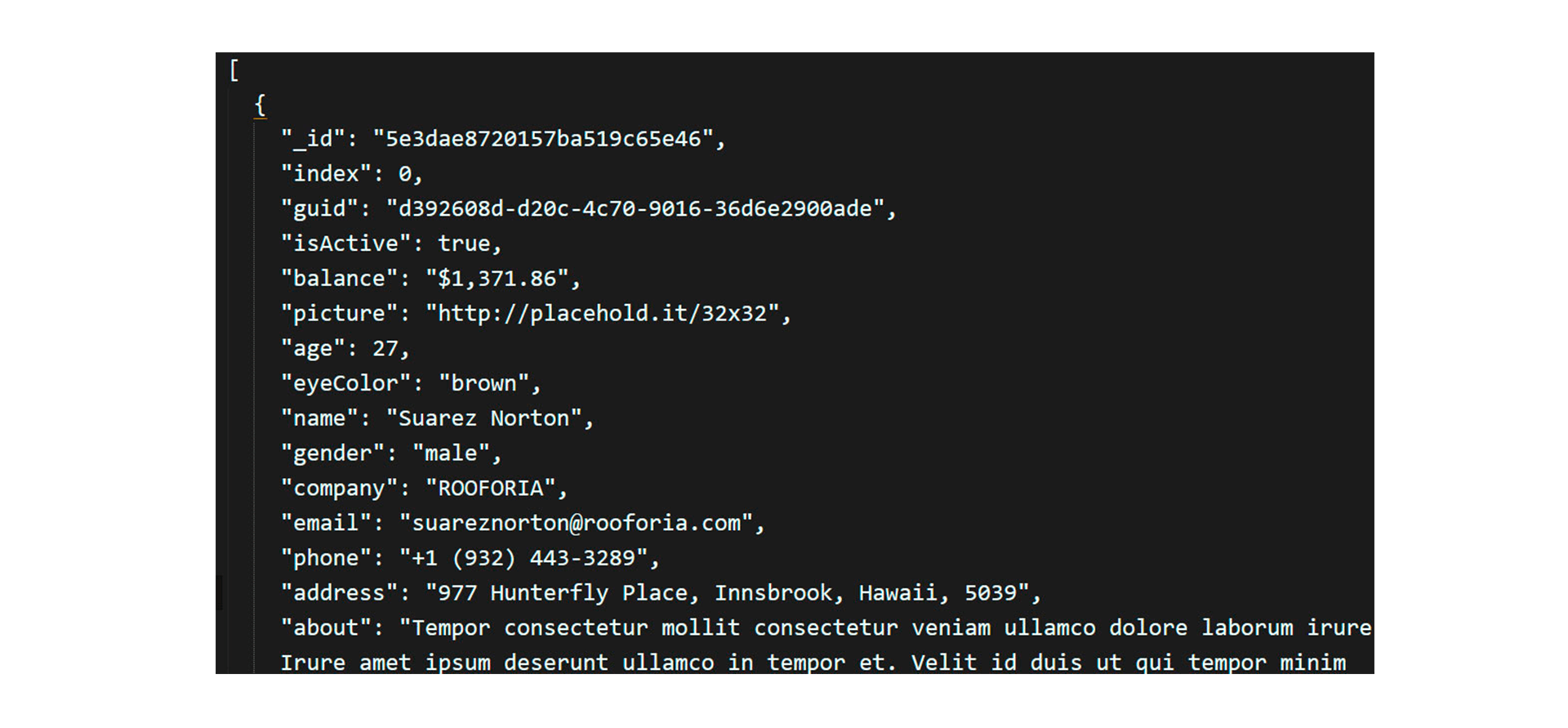
**Tipos de Dados**

Tempos atrás, dados eram geralmente carregados em estruturas tradicionais de banco de dados e possuíam uma estrutura compreensível. Por exemplo, dados de um usuário como nome, CPF, e-mail etc., ou dados da venda de uma certa quantidade de produto, preço da venda, data da venda etc. Esses dados eram chamados de dados estruturais, porque tinham uma estrutura definida e fixa. Um exemplo de uma tabela é demonstrado na Figura 3.3.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.3 - Dados estruturados  
Fonte: Adaptada de aurfupvector / 123RF.*

Porém, com o passar do tempo, novas tecnologias foram sendo usadas em sistemas, como o XML (*Extensible Markup Language*). O XML foi muito usado na transferência de dados entre sistemas não compatíveis, porém, hoje, o formato *Javascript Object Notation* (JOSN), por ser mais leve, veio substituir o formato XML na transferência de arquivos. Embora o XML e JSON sejam estruturados no formato da linguagem (sintaxe), os dados em si não são estruturados, sendo, portanto, denominados semiestruturado. A Figura 3.4 ilustra um arquivo no formato JSON.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.4 - Dados semiestruturados (JSON)  
Fonte: Elaborada pelo autor.*

Já os dados que circulam na internet são denominados não estruturados, pois não apresentam uma estrutura definida, portanto, podem ser de diversos tipos, como um arquivo de texto (em formato Word, pdf), vídeo, som, imagem etc. A Figura 3.5 apresenta um resumo das extensões de dados não estruturados.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.5 - Dados não estruturados  
Fonte: Blankstock  / 123RF.*

Sistemas gerenciadores de banco tradicionais não suportam essa complexidade de dados gerados na internet a cada segundo, mas novas tecnologias surgiram e muitas compõem as tecnologias que sustentam o Big Data.

**Hadoop: a Tecnologia por trás do Big Data**

Hadoop é uma estrutura de código aberto fornecida e mantida pela Apache Software Foundation, criada para armazenar, processar e analisar enormes volumes de dados em um sistema distribuído para clusters, tolerante a falhas, de baixo custo e flexível. Escrito em Java e altamente usado por grandes empresas como Facebook, Google, Twitter, LinkedIn, Yahoo etc., é*Open Soure*(código aberto), o que significa que seu código-fonte está disponível de graça, e qualquer empresa pode editá-lo sob a licença *General Public License*(GPL).

A infraestrutura usada pelo Hadoop é a de ambientes de computação distribuída em clusters de computadores, para criar modelos de programação simples para armazenar e processar grandes dados. Basicamente, o Hadoop é constituído de três componentes: HDFS, MapReduce e Yarn.

O HDFS ou *High Disponibility File System*é um sistema de arquivo de alta disponibilidade, alta escalabilidade, confiável e é o núcleo do Hadoop. Esse sistema de arquivo provê tolerância a falhas.

O MapReduce é um componente fundamental e muito importante do Hadoop, visto que é um modelo de programação paralela e distribuída que consegue trabalhar com grandes volumes de dados. O MapReduce trabalha com dois componentes: Mapper, que mapeia as informações a serem processadas; e o Reduce, que sumariza as informações.

Já o Yarn é o gerenciador de recurso do Hadoop. Em um sistema de cluster com muitos nós, a complexidade de gerenciamento, alocação de recursos e liberação de recursos como memória, CPU e disco são difíceis. Nesse caso, a Yarn é responsável pela gestão desses recursos, ou seja, é responsável pela atribuição dos recursos aos pedidos.

**Desafios do uso de Hadoop**

A inteligência analítica depende de iteratividade e interatividade, porém, no MapReduce, essas tarefas tornam-se mais custosas e complexas. Os nós operam de maneira independente, a comunicação é feita por meio de map-shuffle e sort-reduce, criando diversos arquivos no processo Map-Reduce e tornando o sistema ineficiente para computação analítica.

Trabalhar com Hadoop com eficiência e eficácia não é para programadores ou profissionais iniciantes, visto que a demanda de profissionais qualificados pode ser uma dificuldade.

Os dados são fragmentados entre nós do cluster e podem se tornar um problema de segurança. A segurança do Hadoop pode provir do protocolo de autenticação Kerberos.

Embora haja diversas ferramentas no ecossistema Hadoop, faltam ferramentas para padronização e qualidade de dados. Embora o Big Data seja emergente e muito utilizado pelas empresas, a mineração de dados é um outro assunto fundamental para qualificação de dados e foco de estudos no próximo tópico.

**praticar**

Vamos Praticar

O Big Data é um conceito novo e emergente que está sendo muito utilizado pelas empresas para diversos fins. Esse conceito é baseado em cinco pilares que representam o porquê do uso dessa tecnologia em comparação a um sistema de banco de dados tradicionais.

Assinale a alternativa que descreve corretamente o pilar e sua respectiva descrição.

Parte superior do formulário

**a)**Volume: relativo ao volume de dados gerados.

**b)**Veracidade: relativo a dados originais sem modificação.

**c)**Variedade: relativo à variedade de usuários conectados à internet.

**d)**Valor: relativo ao valor da instituição que usa a tecnologia.

**e)**Velocidade: relativo à velocidade de transmissão de dados.

Parte inferior do formulário

**Mineração de Dados**

Segundo Elmasri e Navathe (2011, p. 698), “o objetivo de um Data Warehouse DW) é dar suporte à tomada de decisão com dados”. A mineração de dados pode ser usada junto com um DW para ajudar certos tipos de decisões, pode ser aplicada a bancos de dados operacionais com transações individuais, para extrair “novos padrões significativos que não podem ser necessariamente encontrados apenas ao consultar ou processar dados ou metadados no Data Warehouse” (ELMASRI; NAVATHE, 2011, p. 698).

Para torná-la mais eficiente, o DW deve ter uma coleção de dados agregada ou resumida. Barbieri (2011) aponta que os conceitos de garimpagem ou mineração de dados estão relacionados à tendência (para aplicações comerciais) de buscar correlações escondidas em altos volumes de dados nem sempre evidentes, principalmente no tratamento cotidiano dos sistemas de informações. Já para Santos e Ramos (2009), a mineração de dados é uma fase do processo de DCBD (Descoberta e Conhecimento em Base de Dados) e consiste na procura de relacionamentos, padrões ou modelos que estão implícitos nos dados armazenados sem grandes bases de dados.

A mineração de dados é um processo pelo qual os dados são armazenados eletronicamente, e a busca por padrões é feita automaticamente com base em regras desenvolvidas sob técnicas de mineração. Esse fato foi explicado há tempos por economistas estatísticos, meteorologistas e engenheiros que procuram por dados, validam e os usam em previsões (WITTEN; FRANK; HALL, 2011).

**Tarefas da Mineração de Dados**

As tarefas associadas à mineração de dados podem ser divididas em dois grupos: descrição e previsão. A **descrição** possibilita a identificação de regras que caracterizam os dados analisados, e a **previsão**faz uso de informações da base de dados para prever o valor de uma outra variável.

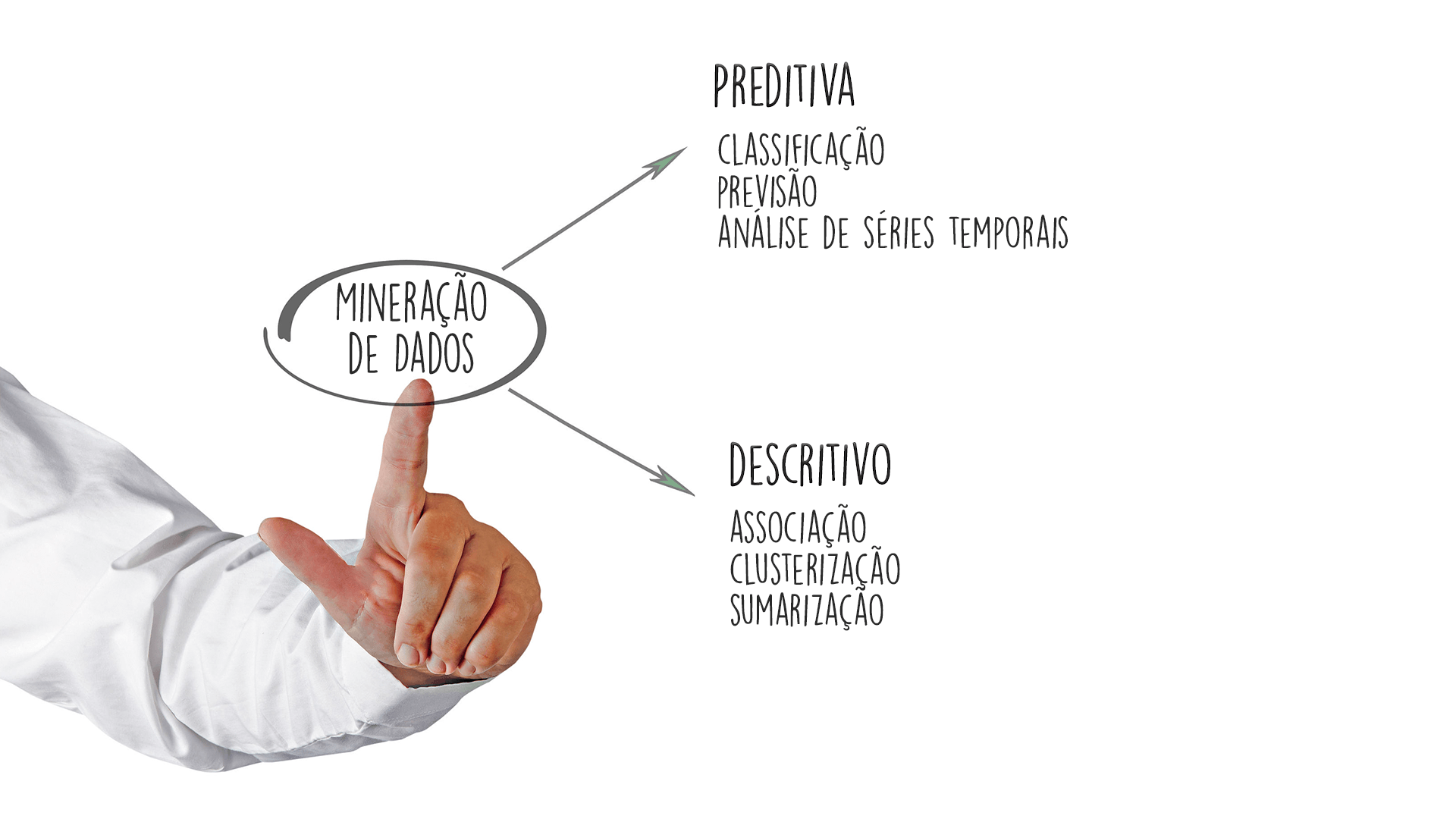
De acordo com Bery e Linoff (2000 apud SANTOS; RAMOS, 2009), existe uma diferença entre descrição e previsão que depende do objetivo da tarefa da mineração de dados, possibilitando um aumento do conhecimento a respeito dos dados e suporte ao processo de tomada de decisão.

Podemos dizer que, quando se trata de previsão, o melhor e mais adequado modelo é o que possibilita a realização de uma previsão mais acertada e elevada, demonstrando superioridade à probabilidade de acertos de outros modelos disponíveis. Porém, o melhor e mais adequado modelo em descrição não necessariamente é o que disponibiliza dados mais precisos, mas o que possibilita um conhecimento mais ampliado dos dados analisados.

Segundo Santos e Ramos (2009), algumas tarefas são associadas à mineração de dados, como: classificação, segmentação, sumarização e modelação de dependências. A seguir, apresentamos as principais características das tarefas da mineração de dados.

* **Classificação:** proporciona um dimensionamento do conjunto de dados dentro do que são denominadas classes predefinidas, realizando a identificação de qual classe cada elemento faz parte. A classificação de dados é um processo dividido em duas fases distintas, sendo a primeira relacionada à identificação de modelo que consiga a integração do grupo de classes que segmentam os dados. Já a segunda fase está relacionada com o processo que classifica, ou seja, esse modelo faz a aplicação do grupo de dados de testes, possibilitando a verificação do desempenho na classificação dos dados desconhecidos. Nos dados apresentados como resultados, é realizada uma análise com o intuito de averiguar o desempenho do modelo escolhido. A precisão desse modelo é estipulada com base na quantidade de registros classificados de forma correta, sendo comparado ao valor real disponibilizado e armazenado no grupo de dados de teste, com o valor previsto pelo modelo. Essa tarefa de classificação é tida como uma tarefa de aprendizagem supervisionada. Mas o que isso quer dizer? Que os atributos e as classes que compõem e nortearão o processo de classificação dos dados já são conhecidos inicialmente. Por exemplo, quando é necessária a utilização da classificação no reconhecimento do grupo do qual o item faz parte, como uma empresa que necessita ter em mãos o padrão de comportamento dos clientes para identificar quais podem abandonar o serviço.
* **Segmentação:**proporciona a identificação de um conjunto de classes ou segmentos nos quais os dados analisados são divididos. É considerada uma tarefa não supervisionada, visto que quem a está utilizando não tem nenhum tipo de influência na definição das classes. Os segmentos ocorrem de agrupamentos encontrados nos dados e que seguem a métrica de similitude. Muitos algoritmos podem ser empregados para identificar segmentos dos dados. Primeiro, consideram que todos os dados fazem parte de um só segmento; em um segundo momento, a divisão acontece entre dois ou mais segmentos, que são divididos novamente e assim sucessivamente, até que um número máximo de segmentos seja atingido, determinado por quem os estão utilizando, ou então até que métricas utilizadas sinalizem que foi obtido o melhor conjunto. Parece confuso? Vamos compreender melhor. A segmentação é semelhante à tarefa de classificação quando o grupo ainda não foi definido, por exemplo, na tentativa de classificar um perfil de cliente com base na localização.
* **Sumarização:** possibilita a descrição de determinados grupos de dados, oferecendo descrições resumidas deles. É possível obter essas descrições a partir da generalização dos dados. Esse tipo de tarefa pode ser usado em uma análise exploratória de dados, com a identificação de grupos de valores ou descrições que permitem a caracterização dos dados analisados. Santos e Ramos (2009) ainda ressaltam que a descrição dos dados realizada a partir da sumarização é um dos objetivos da mineração dos dados por si só, ou também pode estar inserida e integrada em outras tarefas, como para a exploração de dados, possibilitando criar oportunidades para análises futuras. Quer um exemplo para facilitar a compreensão? Separar cliente por idade ou por área de residências possibilita um direcionamento de campanhas de marketing e publicidade para grupos específicos.
* **Modelação de dependências:** o objetivo desse tipo de tarefa está na identificação de um modelo que realize a descrição de dependências significativas entre as variáveis. Algumas situações pontuais da modelação de dependências encontram-se na descrição de associações e de sequências nos dados. A associação possibilita que se realize a identificação de relacionamentos entre os registros armazenados na base de dados, checando a correlação existente entre eles. Esses relacionamentos que surgem dos dados conseguem a apresentação de um nível de confiança importante, permitindo a avaliação do significado das relações encontradas. Como exemplo, podemos citar quando bens de consumo são comprados simultaneamente, ou seja, quando, em um supermercado, é percebido que compradores de salgadinhos têm alta probabilidade de comprarem, também, uma bebida, como um refrigerante.

Essas técnicas podem ser preditivas ou descritivas, sendo usadas na mineração de dados para extrair os tipos de padrões. Para informações sobre eventos passados ou recentes, usa-se a técnica **descritiva**, por outro lado, quando se deseja saber respostas das futuras consultas, usa-se a técnica **preditiva** que, por sua vez, usa dados históricos como princípio essencial para decisões. A Figura 3.6 apresenta um resumo das técnicas.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.6 - Técnicas de mineração  
Fonte: Marigranula / 123RF.*

A Figura 3.6 mostra algumas das técnicas usadas em mineração de dados como análise preditiva, tarefas como associação, preditiva, entre outras técnicas que são usadas por empresas na área de telecomunicações, seguradoras etc.

**reflita**

Reflita

O Brasil é o país do futebol. Será que a mineração de dados ajudaria a encontrar novos Pelés?

**Ferramentas de Mineração de Dados**

Hoje em dia, as ferramentas de mineração de dados comerciais disponíveis no mercado fazem uso de técnicas para a extração de conhecimento. Algumas usam regras de associação, agrupamento, redes neurais, sequenciação, análise estatística, árvore de decisão, entre outras. Também há ferramentas que fazem uso de técnicas mais elaboradas e avançadas, como lógica baseada em caso, algoritmos genéticos, lógica fuzzy, otimização combinatória, entre outras. As ferramentas de mineração de dados fazem uso de técnicas para cumprir o objetivo de extrair conhecimento e informações de volumes imensos de dados

Segundo Elmasri e Navathe (2011), a maioria das ferramentas de mineração de dados utiliza uma interface denominada ODBC (*Open Database Connectivity*), que é um padrão que opera com banco de dados, permitindo acesso aos dados na maior parte dos sistemas de bancos de dados, como os conhecidos Access, dBase, Oracle, SQL Server e Informix. Alguns desses sistemas possuem interfaces disponíveis para programas característicos de bancos de dados. Uma grande parte dessas ferramentas roda em ambientes com sistema operacional Windows e algumas com Unix.

**saiba mais**

Saiba mais

A ferramenta de planilha eletrônica da empresa Microsoft chamada Microsoft Excel pode ser usada para mineração de dados. Você pode compreender mais sobre como usar o MS Excel para minerar dados integrados a um banco de dados relacional SQL.

A seguir, serão apresentadas as ferramentas que fazem uso de interface com o usuário e interface de programação de aplicações (ELMASRI; NAVATHE, 2011).

* **Interface com o usuário:** grande parte das ferramentas conhecidas são executadas em ambientes que possuem interface com o usuário, chamada *Graphic User Interface* (GUI). Dentro dos recursos disponíveis podem estar incluídas técnicas de visualização elaboradas e refinadas de exibição de dados e regras. Logo, pode-se exemplificar a ferramenta MineSet, da SGI, que possui a capacidade de manipular dados de forma interativa. Outra forma de interface são as baseadas em textos. Disponível para ambientes Unix, o Intelligent Miner, da empresa IBM, é um exemplo.
* **Interface de programação de aplicações:**a maior parte dos produtos disponíveis não possibilita a utilização de funções internas, entretanto, alguns fornecem a permissão para que o programador de aplicação realize a reutilização de seu código. As interfaces mais conhecidas são biblioteca C e Dynamic Link Libraries, as conhecidas DLLs. Um ponto importante para destacar é que algumas ferramentas ainda podem incluir linguagens próprias de comando de banco de dados.

Elmasri e Navathe (2011) destacam que existe em torno de 100 ferramentas de mineração de dados disponíveis pelo mundo. Além das americanas, tem a holandesa Data Surveyor e a russa PolyAnaly.

Uma lista de ferramentas de mineração de dados disponíveis atualmente no mercado, com as devidas técnicas utilizadas por cada uma e o ambiente em que rodam está apresentada no Quadro 3.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Empresa** | **Produto** | **Técnica** | **Plataforma** | **Interface\*** |
| AcknoSoft | Kate | Árvores de decisão, raciocínio baseado em caso | Windows UNIX | Microsoft Access |
| Angoss | Knowledge SEEKER | Árvores de decisão, estatística | Windows | ODBC |
| Business Objects | Business Miner | Redes neurais, aprendizado de máquina | Windows | ODBC |
| CrossZ | QueryObject | Análise estatística, algoritmo de otimização | Windows MVS UNIX | ODBC |
| Data Distilleries | Data Surveyor | Abrangente, pode misturar diferentes tipos de mineração de dados | Unix | ODBC compatível com ODMG |
| DBMiner Technology Inc. | DBMiner | Análise OLAP, associações, classificação algoritmos de agrupamento | Windows | Microsoft 7.0 OLAP |
| IBM | Intelligent Miner | Classificação, regras de associação, modelos de previsão | UNIX (AIX) | IBM DB2 |
| Megaputer intelligence | PolyAnalyst | Aquisição de conhecimento simbólico, programação evolucionária | Windows OS/2 | ODBC Oracle DB2 |
| NCR | Management Discovery Tool (MDT) | Regras de associação | Windows | ODBC |
| Purple Insight | MineSet | Árvores de decisão, regras de associação | UNIX (irix) | Oracle Sybase Informix |
| SAS | Enterprise Miner | Árvores de decisão, redes neurais, regressão, agrupamento | UNIX (solaris) Windows Macintosh | ODBC Oracle AS/400 |

Quadro 3.1 - Lista de ferramentas  
Fonte: Adaptado de Elmasri e Navathe (2011, p. 717).

Com a crescente utilização de conceitos e aplicações de *Business Intelligence* (BI), surgiu a necessidade de ferramentas variadas e especializadas em mineração de dados. Barbieri (2011) classifica as ferramentas de mineração de dados da seguinte forma:

* **Ferramentas independentes de aplicação genérica:** são as que disponibilizam um vasto conjunto de recursos e técnicas de visualização, com abordagens mais simplificadas – como é o caso da análise de regressão – e também abordagens mais elaboradas e avançadas – como as redes neurais, regras de associação e análises de agrupamentos.

Essas ferramentas necessitam que uma equipe possua formação na área de Estatística para o momento de preparo dos dados a serem garimpados e extraídos, exigindo um tempo mais longo para a implantação do projeto. Nessa categoria de ferramentas, podemos citar a IBM DB2 Intelligent Miner, da IBM Corp, apresentada em versão dedicada de forma exclusiva à mineração de textos e que consegue a mineração de dados em geral; a Enterprise Miner, do SAS Inc, a Clementine, da SPSS, posteriormente incorporada à IBM, denominada de IBM SPSS Modeler, que possibilita que o projeto de mineração seja exportado como um programa na linguagem C e utilizado em outros sistemas de suporte à decisão da empresa, e ainda a ferramenta Família Knowledge, da Angoss Software Corp.

* **Ferramentas orientadas para algoritmos:**são as que fazem uso de tipos de algoritmos que podem proporcionar um melhor desempenho. Para tanto, é preciso um completo entendimento acerca da problemática a ser resolvida para a mais adequada e correta técnica a ser aplicada.

Ferramentas dessa categoria podem proporcionar um resultado melhor do que das demais. Como exemplos estão Knowledge Seeker, da Angoss, utilizando árvores de decisão e a NeuralWorks, da NeuralWare, para aplicações que necessitam de complexos algoritmos de redes neurais.

* Ferramentas orientadas a aplicações específicas: nessa categoria se enquadram as ferramentas que precisaram passar por adaptações para conseguirem trabalhar com aplicações como CRM, database marketing, entre outras. São fáceis de serem utilizadas, possibilitando que se desenvolvam aplicações por meio da orientação de tutoriais; entretanto, são desenvolvidas para certos tipos de sistemas, não permitindo grandes variações. Podemos destacar o Portrait Customer Analytics, da Portrait Software, que foi comprada pela Pitney Bowes, e as ferramentas da FICO (*Fair Isaac Company*).
* **Ferramentas embutidas em soluções OLAP:**geralmente, são relacionadas a produtos de bancos de dados e aplicações de **Business Intelligence** (BI) ou Processamento Analítico On-line (OLAP).

Proporcionam recursos mais simples, com algoritmos comuns e de fácil aplicação. Como exemplos estão Business Miner, da Business Object, o Scenario, da Cognos, que foi incorporada pela IBM, as ferramentas de data mining dentro do SQL/Server 2008 da Microsoft e a Suíte de data mining da Oracle.

* Ferramentas e serviços de provedores externos: são as mais completas, porque envolvem a prestação dos serviços de mineração de dados. Os serviços citados englobam desde a abordagem de metodologias para mineração de dados até hosting dos serviços de garimpagem de dados, enquadrando-se em um novo conceito. O DW Appliance, fornecido pela IBM, Oracle e HP são exemplos.

Então, há diversas ferramentas de mineração de dados disponíveis no mercado, cada uma com suas particularidades e aplicações. Embora a mineração de dados seja uma tecnologia explorado no mercado, Data Lake é um conceito emergente e também foco de estudo do nosso próximo tópico.

**praticar**

Vamos Praticar

A mineração de dados é uma técnica de encontrar padrões em um repositório de dados ou em uma massa de dados. Essa técnica faz uso de algoritmos que são rotinas predefinidas, que permitem uma busca por um determinado padrão na massa de dados.

Sendo assim, assinale a alternativa que apresenta um algoritmo utilizado em conjunto com a técnica de mineração na técnica de classificação.

Parte superior do formulário

**a)**Algoritmos de redes neurais.

**b)**Árvore binária.

**c)**Redes neurais.

**d)**Algoritmo *first-in-first-out* (fifo).

**e)**Round-robin.

Parte inferior do formulário

**Data Lake**

Data Lake é uma arquitetura emergente que evoluiu para atender à necessidade das organizações de se adaptarem à nova realidade de negócios. Empresas posicionam o objetivo com foco no cliente, são sensíveis aos seus*feedbacks* e ficam atentas a posicionamentos negativos. A fim de recuperar o atraso com o cenário de negócios em mudança, há imenso potencial para a construção de um Lago de Dados para armazenar, processar e analisar enormes quantidades de dados  estruturados e não estruturados (JOHN; MISRA, 2017).

Um Lago de Dados pode ter definições flexíveis. Entretanto, no conceito fundamental, é um repositório de armazenamento e processamento de dados no qual todos os dados em uma organização podem ser colocados para que todos os sistemas internos e externos, parceiros, e os dados dos colaboradores entram nele e os insights surgem.

A seguir, apresentamos algumas características de um Data Lake:

* é um enorme repositório que contém formato bruto de arquivos;
* não é Hadoop;
* as ferramentas usadas no Data Lake são diferentes;
* não é um banco de dados tradicional;
* implementações no Data Lake fazem uso de vários bancos de dados NoSQL e In-Memory;
* um Lago de Dados complementa várias funcionalidades de um Data Warehouse;
* armazena grandes volumes de dados não estruturados e estruturados;
* não é voltado para transação Processamento;
* auxilia na criação de modelos de dados que são flexíveis e podem ser revisados sem redesenho do banco de dados;
* executa o enriquecimento de dados: aumento, classificação e padronização dos dados;
* permite modelos preditivos quase em tempo real e mais precisos que vão além da amostragem e ajudam na geração de modelos multidimensionais;
* um cientista de dados pode realizar quaisquer consultas ad-hoc e construir um modelo avançado em qualquer tempo, iterativamente.
* permite modelar os dados, não só na forma relacional tradicional, mas com o valor real dos dados.

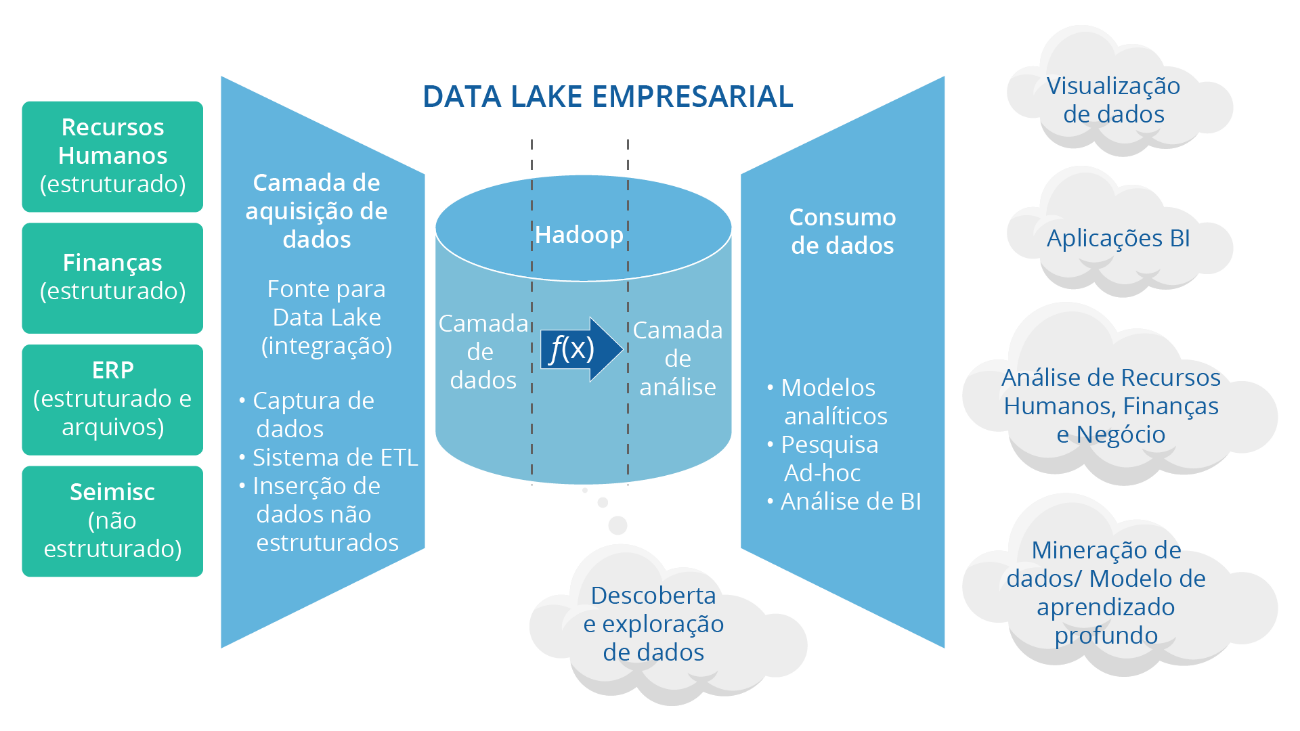
A flexibilidade no armazenamento de todos os dados em um único local ou repositório de Big Data, criando múltiplos pontos de vista, permite que uma arquitetura como a do Data Lake implemente controles para a consistência de dados corporativos.

Controles são aplicados por políticas de governança da informação, logo, fazem uso de *Master Data Management* (MDM), Pesquisa Gerenciamento de dados (RDM) e de outros diversos controles de segurança, colaboração corporativa e controles de acesso implementado.

**Design do Data Lake**

A simplificação de custos e TI são as maiores características do Data Lake. Em um Data Lake, diversos sistemas podem ser fontes de dados. Fontes como sistemas tradicionais relacionais, sistemas de arquivos estáticos, registros na web ou dados de sensores providos da Internet das Coisas podem ou não ser dados estruturados.

Uma camada analítica é construída dentro dos processos do Data Lake, portanto, há necessidade de uma definição, um esquema e uma estrutura para os dados brutos. A partir desse ponto, com base nas diretrizes estabelecidas pela governança de dados, os dados são consumidos pela aprendizagem preditiva modelos, aplicações de *Business Intelligence* (BI) e faixas de ciência de dados. A Figura 3.7 apresenta essa estrutura de um Data Lake empresarial.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.7 - Estrutura de um Data Lake  
Fonte: Adaptada de Gupta e Giri (2018, p. 14).*

A Figura 3.8 ilustra as etapas envolvidas no planejamento de um Data Lake e execução. Note que, na fase de construção, as tarefas são concentradas nos dados em comparação com a fase de planejamento.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

*Figura 3.8 - Etapas de um Data Lake  
Fonte: Adaptada de Gupta e Giri (2018, p. 19).*

Desafios impulsionaram o conceito de um Data Lake em uma empresa e permitiram modificar a forma de tratamento, processamento e armazenamento de dados. No início, os dados das aplicações eram armazenados individualmente, em um processo tradicional, porém, um novo tipo de armazenamento insere todos os dados em somente um único repositório, Data Lake, e permite que esses aplicativos individuais comecem a expor esses dados da organização por meio de acesso de Data Mart ou como uma camada de acesso de dados sobre um Data Warehouse (DW).

Tecnologias como Big Data, Mineração de Dados e Data Lake são novas, resultado de uma evolução natural do mundo de Tecnologia da Informação e Comunicação agregada ao negócio. No próximo tópico, veremos exemplos de como organizações de diferentes áreas utilizam a mineração de dados para alcançarem seus objetivos estratégicos.

**praticar**

Vamos Praticar

O volume de informações que trafega diariamente na internet extrapola a capacidade de gerenciadores de banco tradicionais. Essas informações são dados dos mais variados tipos, como arquivos textuais, imagens, registro de banco de dados, arquivos de transferência de dados etc. Algumas tecnologias emergentes conseguem manipular esses dados.

Com base no texto, assinale a alternativa que apresenta uma tecnologia que agrupa dados brutos em um único repositório.

Parte superior do formulário

**a)**Hadoop.

**b)**Data Lake.

**c)**MapReduce.

**d)**Data mining.

**e)**Yarn.

Parte inferior do formulário

**Na Prática**

Empresas dos mais variados segmentos da economia e de diversos portes fazem uso da mineração de dados para alcançar seus objetivos e metas estratégicas, tais como: crescimento dos lucros, redução dos gastos, obtenção de competitividade, superação da concorrência, expansão dos negócios, entre tantos outros.

As ferramentas de mineração de dados podem ser utilizadas em áreas, como:

* na **saúde**, por meio do auxílio e indicação de diagnósticos mais precisos e mais rápidos;
* na **política**, na identificação do perfil e padrão comportamental dos eleitores;
* nas **vendas**, para a identificação do perfil, padrão de comportamento dos consumidores, bem como suas necessidades e preferências;
* no **setor financeiro**, para a identificação de padrões para o auxílio no gerenciamento de relacionamento com o cliente;
* na **gestão de pessoas**, na identificação e análise de competências e habilidades nos currículos de candidatos;
* no**setor de cobranças**, para detecção de fraudes.

Você consegue ter a dimensão do quanto é importante e valioso para essas organizações terem acesso a informações e, a partir disso, conseguirem alcançar resultados positivos e significativos?

Vamos, a seguir, conhecer alguns casos práticos de utilização de ferramentas de mineração de dados por grandes empresas.

**Grande Loja de Departamento**

Barbieri (2011) relata o caso da utilização de mineração de dados em uma grande rede de lojas de departamentos dos Estados Unidos.

Em 2004, havia a previsão de um grande furacão em direção à região sul do país. A empresa decidiu garimpar os dados de produtos de consumo registrados na área e descobriu que, em outras passagens de furacões, houve um aumento significativo de produtos, não necessariamente relevantes a uma situação de emergência.

Com a realização das análises, a empresa descobriu que, no lugar de lonas, lanternas e pregos, uma torta de morango teve o consumo aumentado em sete vezes. Outra curiosidade constatada foi que a cerveja foi o produto mais vendido no período antes da passagem do furacão.

Com o levantamento desses dados, a empresa decidiu organizar uma operação de logística para deslocamento de estoques desses produtos para a região em questão. A análise realizada com base em históricos anteriores de dados possibilitou a rede de departamentos alcançar enormes lucros. A empresa conseguiu compreender o comportamento de seus clientes em momentos de aflição, quando estão expostos a riscos de tornados e furacões, a partir do uso da mineração de dados em um *Data Warehouse* (DW).

**Empresa do Setor de Saúde no Brasil**

Segundo Barbieri (2011), uma grande empresa de saúde do Brasil decidiu realizar a aplicação de técnicas de mineração de dados em suas ações preventivas. Por exemplo, clientes mulheres foram divididas por áreas de trabalho, e constatou que existia um comportamento padrão relacionado à faixa de idade quando se tornam gestantes. A partir dessa técnica, foi possível calcular com um grau de precisão mais elevado os gastos médios de cada grupo e realizar ajustes necessários nos valores dos serviços oferecidos pela empresa.

Outra aplicação diz respeito à redução dos custos com internações hospitalares. A empresa de saúde, por meio da aplicação de mineração de dados entre seus mais de cinco milhões de beneficiários, coletou características e histórico de saúde, como obesidade, hipertensão, diabetes ou indicadores elevados de colesterol, entre outros índices. Adicionando essas informações a um sistema de *call center*, a empresa conseguiu uma interação maior com clientes, com o intuito de realizar a conscientização sobre essas condições de saúde e implementar ações preventivas que resultaram em redução em torno de 48% nas internações (BARBIERI, 2011).

**Empresa do Setor Financeiro**

Santos e Ramos (2009) descrevem que uma empresa do setor financeiro dos Estados Unidos, fundada em 1888, é especializada em seguros de vida de acordo com as necessidades dos clientes, a classe média dos americanos.

Na década de 1980, a empresa passou a trabalhar com produtos financeiros para clientes pessoa física e empresas. Uma área de grande importância para a empresa é o setor de marketing, que necessita não só manter contato, como também conhecer o perfil e comportamento de seus clientes. As informações da empresa estavam armazenadas em mais de 25 sistemas informatizados diferentes, sem comunicação entre eles.

A empresa decidiu implantar um DW de marketing que comportasse e abrangesse todas as informações sobre os clientes. Esse poderoso banco de dados, alinhado com um conjunto de ferramentas de mineração de dados, serviu de base para uma nova forma de gestão de relacionamento com clientes. A empresa investiu em colaboradores com conhecimento do negócio e com competências técnicas para a melhor utilização possível das informações armazenadas no Data Warehouse.

Um ponto importante para o sucesso da implantação foi o trabalho a partir de algumas campanhas de marketing que geraram resultados imediatos e significativos como: campanha para adoção de pagamento eletrônico como processo de preferência no momento de pagamento dos valores de apólices; campanha para contato com os atuais clientes e oferecimento de produtos que complementassem os serviços já adquiridos, levando à identificação daqueles clientes que poderiam comprar os produtos complementares e como resultado o crescimento nas vendas; campanha de vendas de coberturas complementares com o objetivo de identificar clientes com poucos riscos que tivessem cobertura pela apólice e oferecer coberturas extras; campanha de identificação dos clientes mais lucrativos e aqueles que poderiam adquirir outros produtos; e campanha que visava fornecer informações sobre os clientes e mercados aos agentes de seguros.

O desenvolvimento de todas essas campanhas levou a empresa a resultados positivos e impactantes nas vendas, além do melhor conhecimento sobre seus clientes, necessidades e preferências.

**praticar**

Vamos Praticar

Na prática, o mercado é competitivo e encontrar novos caminhos ou solidificar os antigos passa por tomada de decisões. As empresas adotam sistemas com tecnologias como Big Data e Data Mining para buscar embasamento por melhores tomadas de decisões estratégicas.

Sendo assim, assinale a alternativa que apresenta um cenário (sistema) em que o Big Data e Data podem ser aplicados.

Parte superior do formulário

**a)**Análise de uso de prótese dentária.

**b)**Análise do tempo de registro Web.

**c)**Análise de *dashboard*.

**d)**Análise de perfil de investidor financeiro.

**e)**Análise de número de matrículas.

Parte inferior do formulário

**indicações**

**Material Complementar**



**LIVRO**

**Aprenda mineração de dados: teoria e prática**

Fernando Amaral

**Editora:**Alta Books

**ISBN:**978.857.608.988-9

**Comentário:**Esse é um livro que traz uma proposta diferenciada, com uma linguagem mais didática para o entendimento dos fundamentos da mineração de dados, com exemplos de aplicação por empresas.



**FILME**

**Transcendence – A Revolução**

**Ano:**2014

‍**Comentário:**Esse filme tem ingredientes de Inteligência Artificial (IA), física quântica, nanotecnologia e Big Data. O ator Johnny Depp interpreta Dr. Will Caster, reconhecido e famoso pesquisador na área de IA, que sofre tentativa de morte quando está analisando milhares de dados sobre o cérebro, com o intuito de desenvolver uma máquina com a capacidade de lidar com informações de todos os tipos e fontes (Big Data) com as emoções humanas.

Para conhecer mais sobre o filme, acesse o vídeo a seguir.

[TRAILER](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_3/ebook/index.html)

**conclusão**

**Conclusão**

Nesta unidade, você compreendeu que a tecnologia Big Data é um conceito que se refere a uma imensa quantidade de dados e informações armazenados e representa um grande desafio às empresas contemporâneas, que precisam saber como lidar com esse excesso de dados e transformá-los em informações úteis e relevantes para auxiliar na tomada de decisão.

Outro ponto importante destacado foi a mineração de dados, isto é, o processo de exploração e extração, feito de maneira automática, em grandes quantidades de dados armazenados em bancos de dados e de como identificar padrões de comportamento.

Você também foi apresentado à nova e emergente tecnologia denominada Data Lake, que é definida por um armazenamento de dados brutos usados por empresas para inserir dados em um único repositório para diversos propósitos. Por fim, foram apresentados alguns exemplos práticos de uso das tecnologias da unidade.

**referências**

**Referências Bibliográficas**

AMARAL, F. **Aprenda mineração de dados:** teoria e prática. Rio de Janeiro: Alta Books, 2016.

BARBIERI, C. **BI2** – Business Intelligence: modelagem & qualidade. Rio de Janeiro: Elsevier, 2011.

ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados.** 6. ed. São Paulo: Pearson Addison Wesley, 2011.

FERNANDES, A. A.; ABREU, V. F. A. **Implantando a governança de TI:** da estratégia à gestão dos processos e serviços. 4. ed. Rio de Janeiro: Brasport, 2014.

GUPTA, S.; GIRI, V. **Practical Enterprise Data Lake Insights:** Handle Data-Driven Challenges in an Enterprise Big Data Lake. Apress, 2018.

JOHN, T.; MISRA, P. **Data Lake for Enterprises**. Packt Publishing Ltd, 2017.

PASUPULETI, P.; PURRA, B. S. **Data lake development with big data**. Packt Publishing Ltd, 2015.

SANTOS, M. Y.; RAMOS, I. **Business Intelligence:** tecnologias da Informação na Gestão de Conhecimento. Lisboa: FCA, 2009.

TAYLOR-SAKYI, K. **Big data:**understanding big data. arXiv preprint arXiv:1601.04602, 2016.

WITTEN, I. H.; FRANK, E.; HALL, M. A.**Data Mining:** Practical Machine Learning Tools and Techniques. 3. ed. Santos: Morgan Kaufmann, 2011.

© 2020 - LAUREATE - Todos direitos reservados

**Práticas de Banco de Dados**

Prática de Banco de Dados

**PRÁTICAS DE BANCO DE DADOS**

PRÁTICA DE BANCO DE DADOS

Autor: Me. Paulo Sérgio Pádua de Lacerda

[INICIAR](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html#section_1)

**introdução**

Introdução

Caro(a) aluno(a),

Seja bem-vindo(a) à disciplina de Prática de Banco de Dados! Neste material você ficará familiarizado com tópicos a respeito dos princípios que sustentam o conceito de *Business Intelligence* (BI), entenderá a finalidade e outras ferramentas usadas e demais tecnologias envolvidas na extração e qualificação de dados, além de ferramentas, como *Data Warehouse* (DW), *Dashboard, On-Line Analytical Processing* (OLAP) e ETL. Você verá também outros tipos de  banco de dados, como banco de dados distribuído orientado a objetos e objeto-relacional.

Em suma, você vai identificar e compreender os princípios de Business Intelligence. Vamos lá?

***Business Intelligence* (BI)**

No mundo competitivo, o ambiente de negócio está em constante evolução, aumentando a cada tempo a sua complexidade. Empresas públicas ou privadas são desafiadas a traçarem novos planejamentos estratégicos que acompanhem tal evolução e, assim sendo, busquem por soluções inovadoras em seus modos de operação.

Nessa nova era, tempo é dinheiro/lucro, tomadas de decisões precisam ser ágeis, eficientes, estratégicas e táticas, e, de fato,  uma enorme quantidade de informações se faz necessária para que esses dados e conhecimentos, de forma eficiente e eficaz, sejam transformados em tomadas de decisões com impacto positivo no negócio. Tarefa complexa e que não pode ser feita apenas por pessoas, mas deve ter um suporte computadorizado, portanto, empresas precisam aderir a tecnologias que garantam velocidade e segurança no processamento desses dados.

**reflita**

Reflita

No futebol a presença da tecnologia está muito presente. Por exemplo, hoje consegue-se mapear o quanto um jogador correu durante uma partida, em que parte do campo ele mais atuou, logo pode-se fazer planejamentos individuais a cada jogador de acordo com  suas características de movimentação em campo. Nesse sentido, será que Business Intelligence poderia ser implementado no futebol para auxílio da comissão técnica? Como?

Esse suporte computadorizado vai de processamento de dados, armazenamento de dados, análise de dados, monitoramento de dados a soluções de problemas de negócio, e ferramentas como *Business Intelligence* tornam-se um dos pilares de uma nova gestão. Aplicações como *Key Performance Indicator* (KPI),  *Balanced Scorecard* (BSC) e *Continuous Improvement* (CI) são exemplos de aplicações onde o DW pode ser implementado.

**Um Pouco da História**

As raízes do *Business Intelligence* estão nos anos 70, quando sistemas de informações gerenciais (MIS) geravam relatórios estáticos, bidimensionais e sem análise analítica. Na década de 80, porém, esse conceito evoluiu e passou a ser possível ter suporte a extração dinâmica de relatórios multidimensionais, gerações de previsões, análise de tendências e eram chamados de *Enterprise Information System* (EIS). Já na década de 90, o termo foi usado por Gartner Group,  incorporou outras funcionalidades, como inteligência artificial (IA),  e passa a ser chamado de *Business Intelligence*. Comercialmente, na década de 2000, todos os produtos empresariais relacionados ao tema passaram a incorporar inteligência no negócio.

**Definição**

Há diversas definições para o termo *Business Intelligence*, uma delas diz que BI é um termo que agrega diversas outras tecnologias como ferramentas de base de dados, ferramentas analíticas, aplicativos, metodologias e uma arquitetura, ou seja, uma combinação entre técnicas e tecnologia. Segundo da Silva, Silva e Gomes (2016), o grupo Gartner usou o BI pela primeira vez com o propósito de melhorar o processo de gestão das empresas em termos de obtenção de dados, organização, análise, ter as informações compartilhadas e monitoradas. O termo faz referência à infraestrutura e às ferramentas, mas também às melhores práticas usadas para resultar na eficiência e eficácia no processo de gestão, nas tomadas de decisão e desempenho dentro das organizações por meio de acesso e análise das informações.  Para Turban et al. (2009, p. 27), *Business Intelligence* “[...] é um termo amplo que consegue abranger metodologias, aplicações e ferramentas”. Por representar essa abrangência toda e poder ser atribuído a diferentes significados, o conceito pode levar à confusão.

Turban et al. (2009, p. 47) ainda ressaltam que “[...] o *Business Intelligence* é um conceito que pode ser aplicado em todos os tamanhos de organizações, das pequenas às grandes. Para organizações que não desejam realizar a implementação do BI baseada em *Data Warehouse*, já estão disponíveis ferramentas de análise para funções específicas”.

Essas ferramentas são utilizadas em sistemas como Sistema de Apoio à Decisão (SAD) ou Sistema de Informação Gerencial (SIG) (LAUDON, K.; LAUDON, J., 2009). São sistemas para dar suporte à gestão das empresas através da manipulação de dados oriundos dos diversos departamentos existentes. Assim, diretores e gestores podem executar  planejamentos estratégicos com eficiência e eficácia. Já para Barbieri (2011, p. 95):

*[...] de forma mais ampla, pode ser entendido como a utilização de variadas fontes de informação para definir estratégias de competitividade nos negócios da empresa. Podem ser incluídos nesta definição os conceitos de estruturas de dados, representadas pelos  bancos de dados tradicionais, Data Warehouse, e Data Marts, criados objetivando o tratamento relacional e dimensional de informações, bem como as técnicas de data mining aplicadas sobre elas, buscando correlações e fatos escondidos.*

Para Santos e Ramos (2009, p. 78), o BI deve:

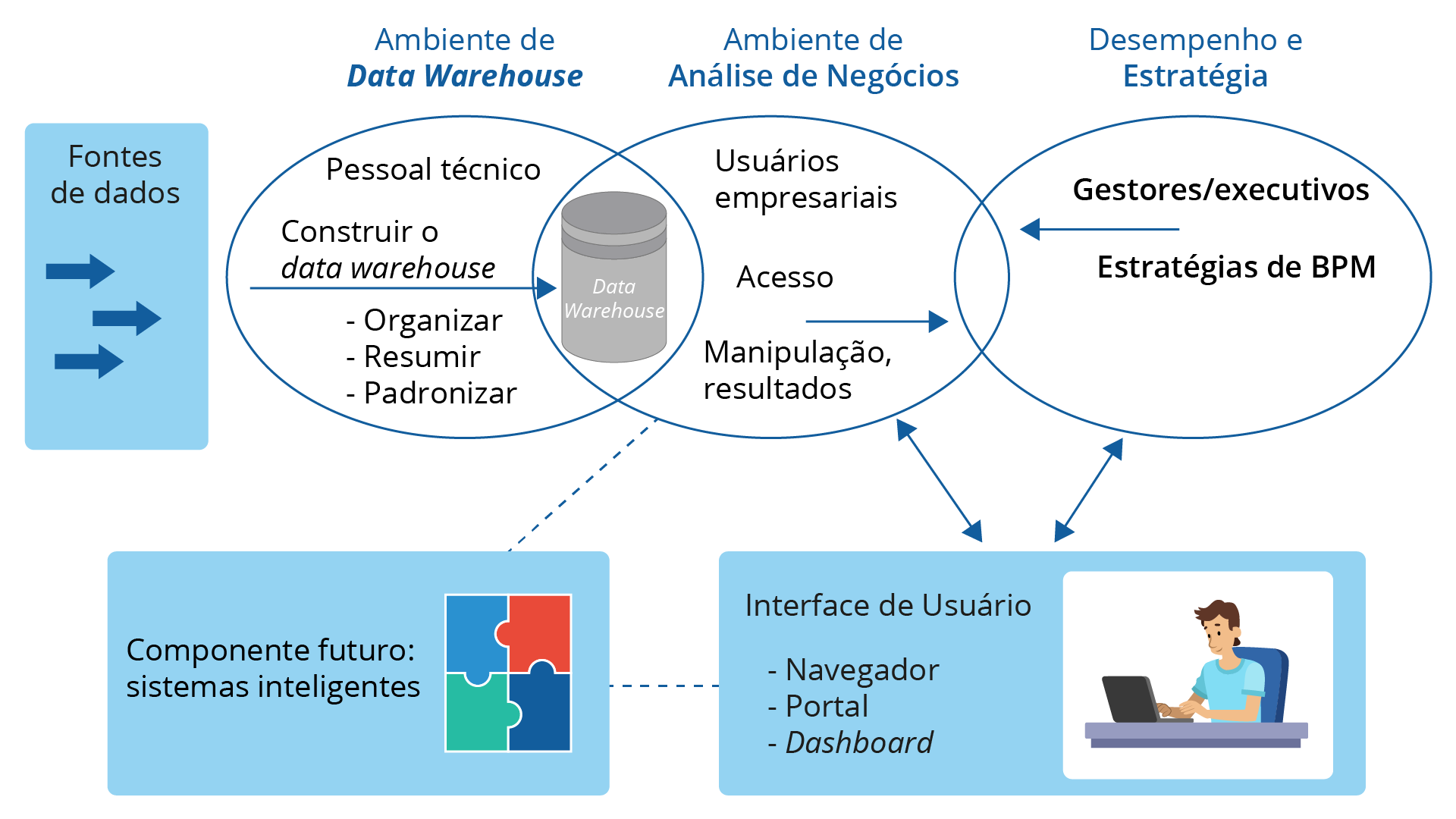
*“[...] combinar a coleta e o armazenamento de dados, bem como realizar a gestão de conhecimento por meio de variadas ferramentas de análise que proporcionam a extração de informação útil e relevante dos dados que estão armazenados. Algumas ações que são atribuídas aos sistemas de BI são:  
• elaboração de previsões apoiadas e fundamentadas em dados históricos, nos comportamentos e performances anteriores e recentes da organização;  
•   criação de conjunturas que consigam evidenciar o impacto da mudança de variáveis;  
•   permissão do acesso ad-hoc aos dados para responder e contestar demandas que não foram pré-estabelecidas;  
• análise de forma detalhada da organização para obtenção de conhecimento aprofundado.*

As soluções BI têm objetivos semelhantes aos sistemas de apoio à decisão, como a preocupação com a transformação de dados em informações relevantes, disponibilizando-as às pessoas certas e no momento certo.

**Arquitetura**

A urgência de crescimento dos negócios e do aprimoramento das decisões empresariais levam as empresas aos planejamentos de como capturar, compreender e agregar valor nos seus dados. Líderes passaram a ter responsabilidade em seus processos empresariais e legitimar os dados baseados em regulamentações ou leis (*Sarbanes-Oxley Act* de 2002), ainda que as organizações precisem agir com inteligência, visto que o processo de vida da informação é mais restrito e gestores precisam ter a informação na hora, no lugar e no momento certo para que ganhos competitivos sejam eficientes e eficazes.

Nesse contexto, a arquitetura de BI passa desde a entrada de dados, por ambientes de *Data Warehouse* e análise de dados até a visualização por meio de gráficos sinalizados de desempenho (*dashboard*). A figura a seguir apresenta uma visão macro de uma arquitetura de BI de alto nível.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.1 - Visão macro de BI  
Fonte: Sharda, Delen e Turban (2019, p. 17).*

Em suma, um sistema de *Business Intelligence* é composto basicamente de quatro componentes essenciais: um ambiente de *Data Warehouse*, um ambiente de análise de negócio, um sistema de mineração de dados (*data mining*) e *Business Process Management* (BPM) para gerenciamento de desempenho e uma interface para visualização e interação com usuário como um *dashboard*(Figura 4.2).

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.2 - Um exemplo de dashboard  
Fonte: Maxim Evseev / 123RF.*

Embora o conceito de *Business Intelligence* seja muito usado pelas empresas, ele integra outras ferramentas como *Data Warehouse*, *Data Mart* e *Dashboard*, que serão foco de estudos no próximo tópico.

**praticar**

Vamos Praticar

O conceito de *Business Intelligence* é um conjunto de várias tecnologias que, integradas, dão suporte computacional a empresas em decisões de negócio. Duas estruturas que fazem parte dos pilares do BI são DW e aplicações. Assinale a alternativa correta que indica uma aplicação em que pode ser implementado o BI:

Parte superior do formulário

**a)**KPI.

**b)**B2C.

**c)**CRM.

**d)**ERP.

**e)**B2B.

Parte inferior do formulário

**Ferramentas**

Fazer uso de *Business Intelligence* não é somente investir em Tecnologia da Informação, apesar de o BI não existir sem a TI. Já vimos que o *Business Intelligence* é um conceito com ferramentas e metodologias que tem o intuito de transformar os dados em informações e as informações em conhecimento para apoio e fundamentação da tomada de decisão. Este tópico apresentará as principais ferramentas de BI.

***Data Warehouse* (DW) e *Data Mart***

Com a imensidão de informações disponíveis e geradas, as empresas precisam enfrentar o desafio de minerar e extrair as mais importantes, relevantes e necessárias. As informações são disponibilizadas às empresas nos mais variados tipos e meios. Podem estar em forma de relatórios em papel, relatórios digitais, e-mails, mensagens em redes sociais, entre muitos outros.

Em muitos casos, as empresas precisam superar dificuldades no momento de análise dessas informações. Além da variedade e de diferentes fontes de informações, outro ponto importante para se destacar está na organização das informações. Empresas, muitas vezes, possuem as mesmas informações em bancos de dados diferentes. Uma maneira de solucionar esse problema está na implantação de um *Data Warehouse*. E você sabe o que é isso?

*Data Warehouse* é um sistema de hardware que armazena dados oriundos de diversos departamentos de uma organização que são guardados ou armazenados num único local com segurança. Para Santos e Ramos (p. 79, 2009), “[...] *Data Warehouse* é um repositório desenvolvido especialmente para o fortalecimento da informação da organização que leve a uma configuração válida e consistente, que permita aos usuários a análise de dados de maneira seletiva e rigorosa.” O *Data Warehouse* armazena em um único local informações originadas de diversificadas fontes.

Segundo Medeiros (2013, p. 29), “[...] a partir do desenvolvimento de um DW, disponibilizando grandes volumes de dados, é possível para a organização em questão a busca de determinadas informações relacionadas a padrões nos dados que se repetem em certos períodos de tempo”.

Essa solução tecnológica tornou-se necessária na integração dos dados das empresas que eram armazenados em computadores diferentes e precisavam ser acessíveis para todos os tomadores de decisão em todos os níveis de decisão da organização. Muitas empresas enfrentam o problema de terem seus dados e informações armazenados em bancos de dados diferentes, a solução é utilizar o *Data Warehouse*. A figura a seguir apresenta uma visão macro do sistema.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.3 - Data Warehouse  
Fonte: Paisan Homhuan / 123RF.*

Nota-se, pela Figura 4.3, que a extração, a transformação e a carga (ETL) podem ser realizadas de diversos bancos de dados (repositório de dados) e inseridas dentro do *Data Warehouse*. Assim, os dados ficam armazenados em um único repositório de dados para serem utilizados posteriormente.

Já  Inmon (1996 apud SANTOS; RAMOS, 2009, p. 79) define:

*[...] o DW é um banco de dados mantido autonomamente em relação às bases de dados operacionais da organização. As informações armazenadas nele são identificadas temporalmente e permitindo o armazenamento de dados de muitos anos. Usuários em geral conseguem inserir questões complexas e avançadas no DW, e assim questões rotineiras das organizações, como a coleta, armazenamento e a manipulação da informação ficam para as bases de dados operacionais [...].*

Existem algumas diferenças entre as bases de dados operacionais e um DW, como as apresentadas no quadro a seguir.

|  |  |
| --- | --- |
| **Bases de Dados Operacionais** | **Data Warehouse** |
| Objetivos operacionais | Registro Histórico |
| Acessos de leitura/escrita | Acessos só de leitura |
| Acesso por transações predefinidas | Acesso por questões *ad-hoc* e relatórios periódicos |
| Acesso a poucos registros de cada vez | Acesso a muitos registros de cada vez |
| Dados atualizados em tempo real | Carregamentos periódicos de mais dados |
| Estrutura otimizada para atualizações | Estrutura otimizada para processamento de questões |

Quadro 4.1 - DW x SO  
Fonte: Santos e Ramos (2009, p. 79).

O *DW*, portanto, armazena variados dados provenientes de diversas fontes de uma empresa. Em várias ocasiões, o *Data Warehouse* é conhecido como um enorme repositório de dados (banco de dados) com uma capacidade de armazenamento de milhões de informações (dados). Entretanto, um *Data Warehouse* é muito mais que um gigantesco repositório, pois as características do DW permitem que essa tecnologia de seja um instrumento muito benéfico para o negócio das organizações.

Segundo Silberschatz, Korth e Sudarshan (2006), as principais características dos *Data Warehouse* são:

Organização:

um fator importante que permite o *Data Warehouse* organizar dados e informações por assunto, usar metadados e usar as informações como base para a tomada de decisão.

Consistência:

a codificação dos dados é a mesma; dentro do *Data Warehouse* dados são padronizados.

Variedade de tempo:

o armazenamento no *Data Warehouse* varia entre cinco e dez anos para que análises históricas possam ser executadas, avaliação de tendências, entre outros.

Não volatibilidade:

os dados armazenados no DW são somente para leitura e, por essa razão, não podem ser atualizados.

Estrutura relacional:

a estrutura deste tipo de banco é normalmente relacional, ou seja, a organização e o armazenamento dos dados são feitos em tabelas.

Arquitetura cliente/servidor:

permite uma maior facilidade de acesso por parte do usuário.

Inmon (1996 apud SANTOS; RAMOS, 2009), por outro lado, sistematiza as principais características do *Data Warehouse* como:

Orientado por assunto:

um repositório que armazena informações por assunto.

Integrado:

constituído a partir de várias bases de dados da organização, portanto, faz uso de técnicas de limpeza e integração dos dados para garantir a consistência dos dados, como por exemplo, estruturas de codificação utilizadas etc.

Catalogado temporalmente:

a finalidade do DW é armazenar as informações por um histórico de tempo.

Para Vicci (2014, p. 184), os DWs:

*[...] possuem características importantes como: permitir uma visão conceitual multidimensional de dados; proporciona dimensionalidade de forma genérica; permite dimensões e níveis de associação de forma ilimitada; permite transações sem restrições entre as dimensões; permite transparência; proporciona acessibilidade; suporta múltiplos usuários; permite flexibilidade de relatórios, entre outras.*

De acordo com Silberschatz, Korth e Sudarshan (2012), há tópicos relevantes  que merecem uma atenção diferenciada no momento da criação de um Data Warehouse, tais como:

Na coleta dos dados:

precisa entender dois fatores: como e quando a coleta de dados será realizada, por exemplo, dados colhidos no período noturno.

Com relação ao modelo:

os dados têm origens diferentes e muito provavelmente tenham esquemas diferentes.

Manipulação dos dados (transformação e limpeza):

dados inconsistentes podem ser corrigidos como erro de digitação de nomes, endereços incompletos etc.

Propagação de atualização:

informações de atualizações sobre origens de dados devem ser propagadas.

Resumo de dados:

armazena-se dados resumidos de uma relação em vez de uma relação inteira. Por exemplo, num sistema de venda de roupa, em vez do armazenamento sobre cada item vendido, armazena-se somente o resumo total de vendas de roupas por nome do produto ou por categoria de vendas.

Trabalhar com *Data Warehouse* fornece benefícios para as organizações: as informações gerenciais ficam todas armazenadas em um único banco de dados, possibilitando maior agilidade no acesso e na leitura; todas as informações gerenciais, de departamentos e áreas diferentes, utilizarão os mesmos conceitos, ou seja, falarão a mesma linguagem; as informações serão manipuladas de forma que não haja erros de integridade, permitindo melhor qualidade; é possível armazenar informações históricas, permitindo a realização de projeções e tendências, entre outros.

Um ponto importante que merece ser abordado é o conceito de *Data Mart*. Um *Data Mart* é um *Data Warehouse* de tamanho reduzido que proporciona o apoio ao processo de tomada de decisão de uma área ou assunto específico, e não à organização como um todo.

Podemos afirmar que um *Data Mart* é um subconjunto de informações do *Data Warehouse* de determinadas áreas ou assuntos da organização.

**Análise de Dados com Ferramentas OLAP**

As ferramentas OLAP, do inglês *On-Line Analytical Processing*, ou seja, Processamento Analítico On-line, têm como objetivo realizar análises das informações armazenadas nos repositórios que fazem parte da integração dos sistemas.

Segundo Elmasri e Navathe (2011), OLAP é um termo usado para descrever a análise de dados complexos do *Data Warehouse*. Nas mãos de habilidosos trabalhadores do conhecimento, as ferramentas OLAP utilizam capacidades de computação distribuída para análises que exigem mais armazenamento e poder de processamento do que se estivessem localizadas de forma econômica e eficientemente em um desktop individual.

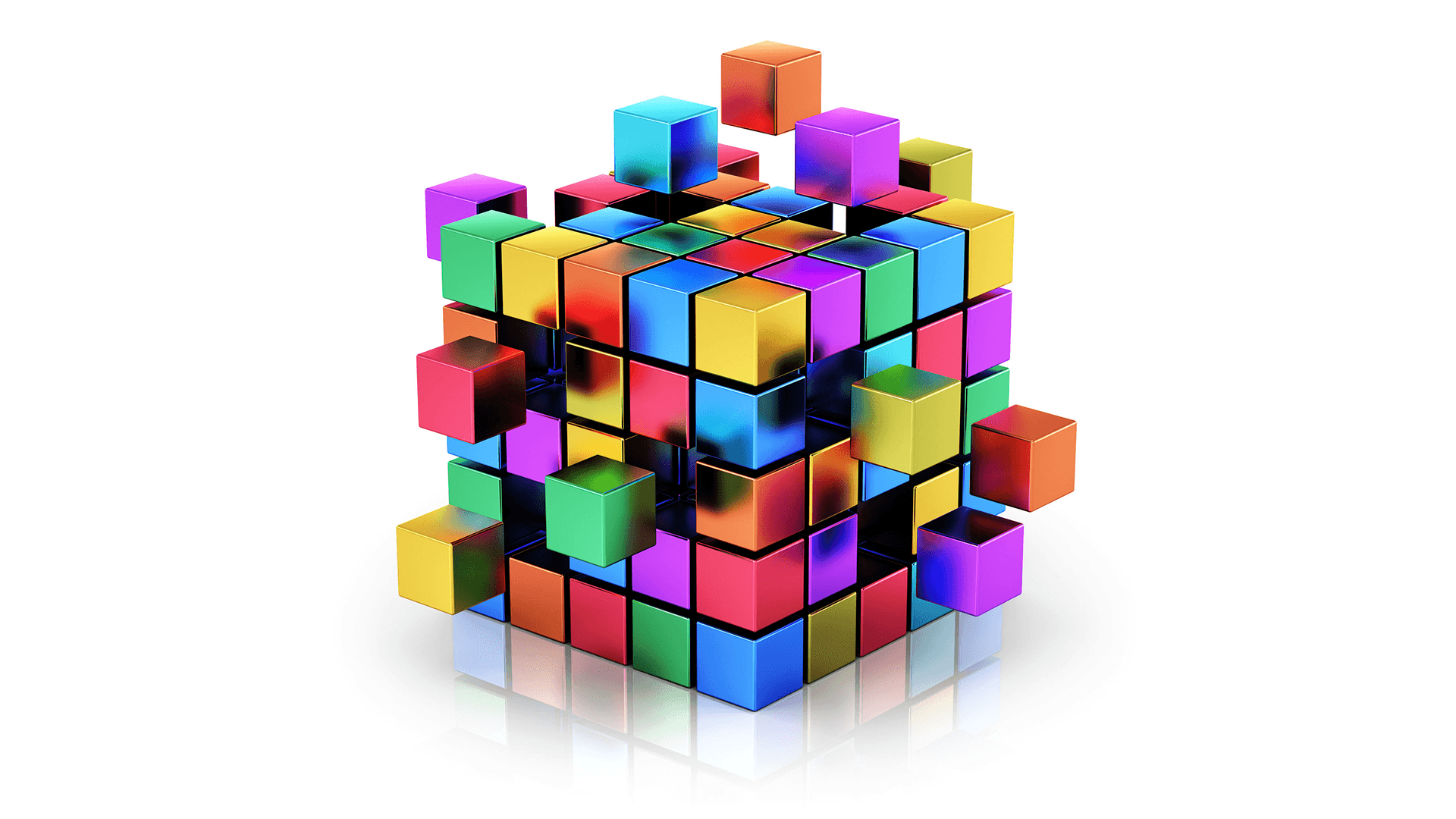
OLAPs são ferramentas tecnológicas com capacidade de proporcionar aos gestores de negócios e tomadores de decisões uma análise e visualização dos dados da organização de maneira mais ágil, facilitada e consistente.

Segundo Turban et al. (2009), OLAPs são sistemas que se referem a uma grande quantidade de atividades normalmente executadas por usuários finais no ambiente on-line. Isso inclui a geração e a resposta de consultas, solicitações de relatórios e gráficos *ad hoc* e sua execução.

Gouveia *et al*. (2011) afirma que é uma solução desenvolvida para a recuperação das informações-chave com o objetivo de proporcionar facilidade e flexibilidade para a análise de dados da organização como um todo, permitindo uma melhor tomada de decisão pela camada gerencial.

Diferentes tecnologias podem ser utilizadas para a exploração de um *Data Warehouse*, porém a mais comum é a OLAP, que, por meio da criação de cubos, proporciona a análise da informação em diferentes perspectivas.

Um cubo multidimensional funciona de forma semelhante a um cubo mágico: movendo as peças, pode-se ter uma visão do cubo, girando novamente, outra visão sobre os mesmos dados no banco de dados.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.4 - Cubo multidimensional  
Fonte: scanrail / 123RF.*

Para Santos e Ramos (2009), as ferramentas OLAP proporcionam a análise, de forma multidimensional, dos dados que estão em repositórios, não importando o local onde estão armazenados. As ferramentas OLAPs são divididas em:

* **ROLAP**(*Relacional OLAP*): trabalham intermediando um banco de dados relacional e as ferramentas da camada de usuário que exercem a função de cliente para realizar a análise dos dados. Fazem uso de gerenciador de base de dados relacionais para armazenamento e gestão dos dados que serão analisados.
* **MOLAP**(*Multidimensional OLAP*): esse tipo de ferramenta faz uso de base de dados multidimensionais e, portanto, dá suporte a diversos tipos de leitura de dados (consultas de dados).
* **HOLAP**(*Hybrid OLAP*): uma ferramenta que combina a escalabilidade da ferramenta ROLAP com a característica de velocidade da ferramenta MOLAP.

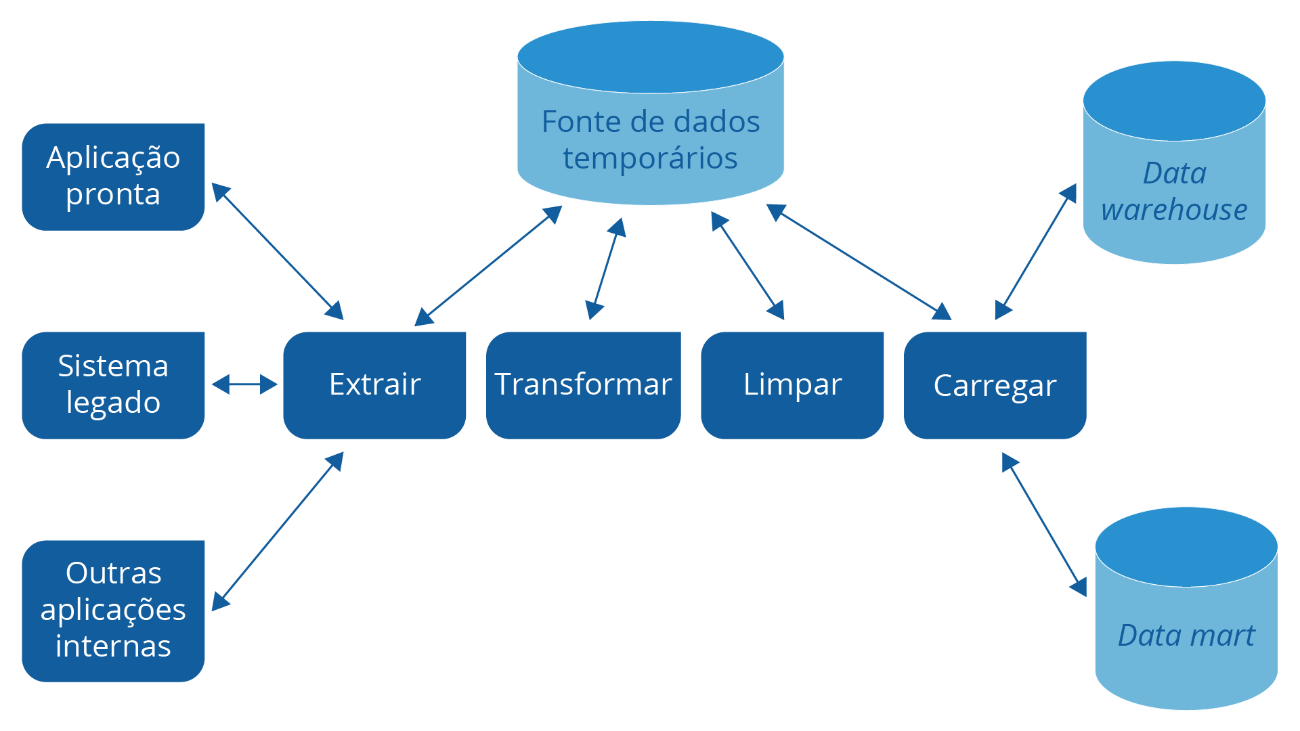
Já de acordo com Turban *et al*. (2009), os processos aplicados por ferramentas OLAP podem abranger diferentes abordagens, como por exemplo:

* **MOLAP**: nessa abordagem são utilizados dados inseridos em estrutura de cubo, de modo que esses dados podem conter diversas vistas de todas as suas dimensões. O banco de dados utilizado é especializado e multidimensional.
* **ROLAP**: nessa abordagem bancos de dados relacionais são utilizados e visões (consultas não pré-processadas) dinâmicas podem ser realizadas.
* **WOPAL**: essa abordagem faz uso de navegador Web para acesso dos dados. Santos e Ramos (2009) ressaltam  as tarefas disponíveis para a manipulação de cubos. São elas:
* ***drill-down***: do mais abrangente para o mais específico, por exemplo, faz a análise do faturamento do ano, depois semestre, mês  e dia;
* ***roll-up***: ao contrário do *drill-down*, faz do mais específico para o mais abrangente, por exemplo, análise do faturamento do dia, depois mês, semestre e, por fim, ano;
* ***slice and dice***: através de execução de corte e redução, restringe informações.  Por meio de condição, o corte permite filtrar ou selecionar informações no cubo. Já a redução (*dice*), através de critérios usados para duas ou mais dimensões, cria limitações nos dados;
* ***pivot***(rotate): a rotação possibilita percorrer todos os lados de visualização dos dados, permitindo representações diferentes.

As ferramentas OLAP são  usadas sozinhas em um projeto de *Business Intelligence* ou então combinadas com ferramentas de mineração e dados.

**Ferramentas *Extraction, Transformation and Loading* (ETL)**

As ferramentas ETL são responsáveis pela extração dos dados das bases operacionais de uma organização, de modo que sejam efetuadas transformações com o intuito de gerar informações valiosas e relevantes e, por fim, armazená-las em um repositório para que o acesso seja facilitado. O processo de ETL é responsável pela carga das tabelas do DW. A figura a seguir apresenta uma macro visão do processo de uma ferramenta ETL.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.5 - Processo ETL, extração, transformação e carga no Data Warehouse  
Fonte: Turban et al. (2009, p. 72).*

A operação “extrair” é a capacidade de acesso e extração de dados de qualquer fonte. A operação “transformar” faz a aplicação de regras de negócio de requisitos baseados nas áreas do DW com o intuito de normalizar os dados que serão armazenados no repositório.

Napoli (2011) afirma que as transformações podem ser complexas quando na extração de dados de um ambiente heterogêneo e podem apresentar alguns problemas, como:

* chaves primárias inconsistentes;
* inconsistências nos dados;
* dados inválidos (ação do  processo de limpeza);
* sinônimos e homônimos (dados repetidos não são fáceis de reconhecer);
* lógica de processo intrínseco.

A operação “limpar” consiste em retirar os que não respeitam as regras de negócio ou que estão incompletos. O uso de bases de estagiamento para manter os dados pós-extração e durante os processos de transformação e limpeza é bastante comum. Por fim, os dados já transformados e limpos são armazenados no repositório *Data Warehouse*. Continuando com nosso estudo, vamos para o próximo tema: *Dashboards*.

***Dashboard***

*Dashboards*são ferramentas que proporcionam uma interface com usuários para que informações sejam demonstradas de forma visual, levando à melhor compreensão dos envolvidos no processo de tomada de decisão.

**saiba mais**

Saiba mais

Há tópicos de comparação de *engine*de banco de dados. Entender as diferenças entre bancos de dados ajuda na tomada de decisão para uso no desenvolvimento de uma aplicação no sistema. O site *DB-Engines.com* oferece algumas comparações entre os principais *engines*de bancos de dados.

[ACESSAR](https://db-engines.com/en/ranking)

Segundo Alexander e Walkenbach (2010), um *Dashboard*é uma espécie de painel para visualização que proporciona e leva os usuários a compreender de forma mais rápida, relevante e facilitada os objetivos ou métricas organizacionais. Possui algumas características importantes, como:

* faz uso de gráficos para as evidências das tendências, possibilitando a realização de comparações que identifiquem exceções;
* disponibiliza somente as informações mais pertinentes e relevantes para o objetivo;
* abrange conclusões relevantes e significativas para o objetivo, dispensando a necessidade que o usuário tenha de analisar.

Por outro lado, para Kaushik (2010), um *Dashboard*ideal deve ser aquele orientado à ação, que consegue a contemplação no seu desenvolvimento dos quatro pontos citados a seguir:

* apresentar métricas separadas e com seu desenvolvimento progressivo ou posição ilustrada de forma gráfica;
* interpretar e contextualizar os dados, realizando a identificação de tendências e apontamentos de potenciais riscos característicos a determinadas situações;
* ações e possíveis próximos passos, implicando na identificação da origem dos problemas quando surgirem, facilitando sua resolução;
* proporcionar impacto na organização e no cliente: o desenvolvedor do *Dashboard*deve levar em conta e realizar estimativas dos impactos dos dados, da análise realizada e da resolução dos possíveis problemas na organização e no cliente.

Embora o *Dashboard*faça parte do conceito de ferramentas que compõem um *Data Warehouse*, com o advento da Internet, surge a área de Data Science, uma área multidisciplinar voltada para a análise de dados e que será o foco de estudos do próximo tópico.

**praticar**

Vamos Praticar

Na análise de dados, a estrutura de um *Data Warehouse* é composta por quatro distintos pilares:  repositório de dados, ambiente de análise de dados, um *data mining* e aplicações de controle e visualização de dados. Assinale a alternativa correta que indica uma ferramenta usada para visualização de dados.

Parte superior do formulário

**a)***Dashboard*.

**b)**OLAP.

**c)**Excel.

**d)**ETL.

**e)***Data mining*.

Parte inferior do formulário

***Data Science***

Em um cenário no qual a internet tem sido a fonte principal da origem dos dados, e esses dados, na sua maioria, são compostos por estruturas de dados classificadas como não estruturadas, os profissionais que trabalham com *Big Data* (desenvolvedores) centralizam o seu trabalho principalmente na dimensão de volume de dados, com estruturas como Hadoop e Spark, capazes de processar conjuntos de dados muito grandes em paralelo.

*Data Science* (ciência dos dados) é uma área multidisciplinar que pode, por exemplo, concentrar-se em dimensões menos pesquisadas, a exemplo de  velocidade e variedade, que são características de aplicações de dados rápidas. Assim a ciência de dados ou ciência orientada por dados combina diferentes áreas de trabalho em estatística e computação para interpretar dados para fins de tomada de decisão (CURTY; CERVANTES, 2016).

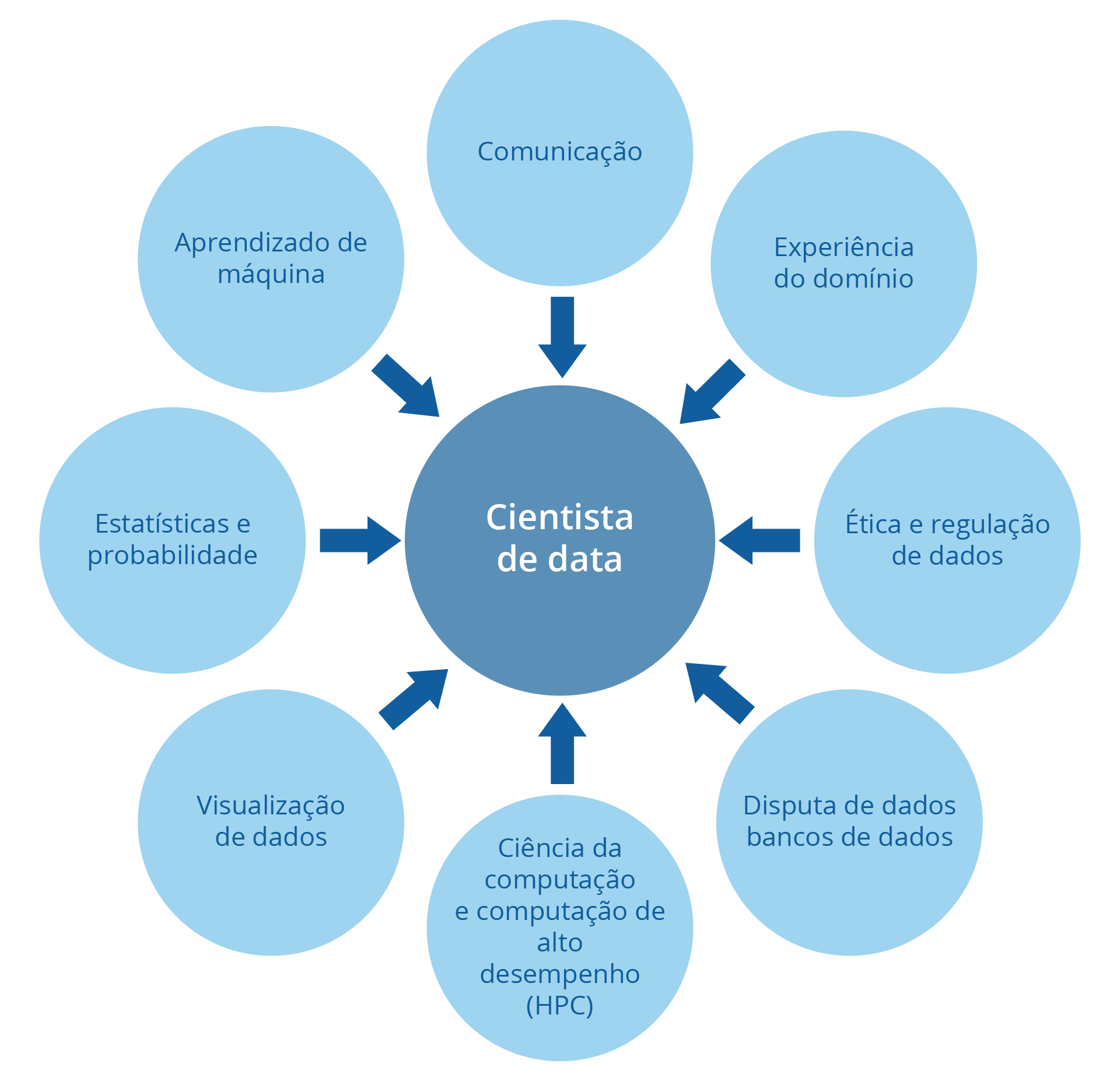
Áreas relacionadas com aprendizado de máquina e mineração de dados deram origem a muitos elementos usados na ciência dos dados.  Essas disciplinas se interconectam com foco na melhoria da tomada de decisões através da análise de dados.

Em *Data Science*, pode-se extrair diferentes tipos de padrões. Por exemplo, extrair  padrões que auxiliem na identificação de grupos de clientes exibindo comportamentos e gostos semelhantes. Esse tipo de operação é chamado de *clustering*(agrupamento de dados)  na ciência de dados. Mas também pode-se querer extrair produtos com um padrão que identifique outros que são comumente comprados em conjunto. Essa técnica é chamada de *association-rule mining*.

**Uma Breve História**

A Ciência de Dados originalmente foi usada em substituição para "Ciência da Computação " em 1960. Uma década e meia após, o termo foi utilizado para definir o levantamento dos métodos de processamento de dados utilizados em diferentes aplicações. Na década de 2000,  a Ciência de Dados foi introduzida como uma disciplina independente. A Harvard Business Review publicou um artigo em 2012 descrevendo o papel do cientista de dados como o "trabalho mais sexy do século 21" (DAVENPORT; PATIL, 2012, *on-line*).

Então, torna-se complexo o domínio de todas as áreas que envolvem Data Science, e, de fato, a maioria dos cientistas de dados geralmente tem conhecimento aprofundado e real experiência em apenas um subconjunto delas. No entanto, é importante entender e estar ciente da contribuição de cada área para um projeto de ciência de dados. A Figura 4.6 apresenta um cenário das disciplinas que envolvem *Data Science*.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.6 -  Disciplinas que envolvem Data Science  
Fonte: Kelleher e Tierney (2018, p. 21).*

**Tarefas**

O objetivo em Data Science é encontrar soluções de problemas do mundo real através de padrões encontrados nos dados, para tal feito, tem a necessidade de classificar os dados em quatro classes gerais de tarefa:

* agrupamento  ou segmentação (*clustering*);
* detecção de anomalia (*outlier*);
* mineração de regras de associação (*association-rule mining*);
* previsão (*prediction*).

Essas tarefas determinam o modo com que os dados serão analisados de acordo com as regras relacionadas ao negócio, como verificação de perfil de compras, correção de dados inconsistentes, classificação de processos, entre outras.  Essas classes são descritas como:

**Agrupamento ou segmentação**: usado para classificar elementos como clientes e produtos e muito utilizado nos negócios, por exemplo, produtos mais comprados em época de promoção.

**Detecção de anomalias ou análise atípica**: envolve a busca e a identificação de casos que não estão em conformidade com os dados típicos em um conjunto de dados.

**Mineração de regras de associação**: uma estratégia padrão em vendas é a venda cruzada, ou sugestão de compra aos clientes de produtos relacionados que eles também podem querer comprar ou produtos complementares.

**Previsão**: análise de previsão de preço em um determinado tempo, por exemplo, previsão de taxa de abandono ou cancelamento.

Embora *Data Science* seja uma área abrangente que englobe áreas como mineração de dados e outras, há outras tecnologias de banco de dados, como banco de dados distribuídos, orientados a objetos e o denominado NoSQL, que precisam ser compreendidas e que serão foco de estudos do próximo tópico.

**praticar**

Vamos Praticar

*Data Science* é uma área multidisciplinar que abrange diversas outras áreas como mineração de dados, *Big Data* etc. O foco dessas tecnologias é encontrar soluções que permitam um embasamento eficiente para a tomada de decisões num negócio.  Assinale a alternativa correta com relação à ciência dos dados.

Parte superior do formulário

**a)**Extração, transformação e carregamento.

**b)**Técnica de agrupamento.

**c)**Volume, variedade e velocidade.

**d)**OLAP.

**e)***Data Mart*.

Parte inferior do formulário

**Outras Tecnologias**

Um banco de dados é um repositório que armazena de forma integrada os dados operacionais de uma organização. Existem tecnologias aplicadas a essa importante ferramenta, como os bancos de dados distribuídos, bancos de dados orientados a objetos e bancos de dados objeto-relacionais.

**Banco de Dados Distribuído**

Os bancos de dados podem ser centralizados em um mesmo lugar ou então estar divididos em diferentes lugares. Tradicionalmente se usa muito banco de dados centralizado, ou seja, instalado em um único computador em que são executados programas e rotinas para acessá-lo. Os usuários podem estar ligados diretamente ao banco ou por meio de uma rede de computadores. O crescimento das redes de computadores impulsionou a utilização de recursos de computação distribuídos. A partir daí surgiram os sistemas gerenciadores de bancos de dados distribuídos  (SGBDD).

De acordo com Elmasri e Navathe (2011), a definição de um banco de dados distribuído está relacionada a um conjunto de diversos bancos de dados inter-relacionados e distribuídos através de uma rede de computadores, e um SGBDD (sistema de gerenciamento de banco de dados distribuídos) relacionado a um sistema de software que faz o gerenciamento de um banco de dados distribuído enquanto transforma essa distribuição transparente ao usuário.

Ainda segundo os autores, existem vantagens importantes dos bancos de dados distribuídos:

maior facilidade e flexibilidade de desenvolvimento da aplicação:

por estarem distribuídos e não centralizados, ou seja, servidores de banco espalhados pela rede de computadores, há uma transparência no controle e na distribuição dos dados, o que permite uma eficiência na criação e na manutenção das aplicações;

confiabilidade:

quando se fala em confiabilidade, quer-se referir à probabilidade de um sistema estar funcionando em determinado momento no tempo;

disponibilidade:

está relacionada à probabilidade do sistema se manter disponível de forma contínua,  durante um período de tempo. É obtida por meio do isolamento de falhas ocorridas em seu site de origem sem que os demais bancos de dados conectados à rede sejam afetados.

No momento em que os dados e o SGBDD são distribuídos por vários sites, quando um apresentar falhas, os demais permanecem operando. Somente os dados e o SGBDD existentes no site que apresenta defeitos ficarão sem acesso, o que melhora a confiabilidade e a disponibilidade. Outra forma de melhorar a disponibilidade ainda mais é com a replicação dos dados e o SGBDD em mais de um site.

[[](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)

*Figura 4.7 - Uma visão de banco de dados distribuído  
Fonte: macrovector / 123RF.*

Já em um ambiente centralizado, a falha ocorrida em um único site deixará o sistema inteiro indisponível a todos os usuários. Um ponto a se ressaltar é que, em um banco de dados distribuído, pode acontecer de alguns dados no site ficarem inacessíveis, porém os usuários ainda conseguirão ter acesso a outras partes ou a outros dados do banco de dados. Agora se os dados no site que apresenta falha forem duplicados em outro site antes da falha ocorrida, não terá problema para o usuário. Assim, há:

maior desempenho:

um SGBD distribuído consegue a fragmentação do banco de dados mantendo os dados mais próximos de onde eles serão mais necessários e mais utilizados. A operação de localizar os dados permite a redução da disputa pela CPU e serviços de E/S, reduzindo os atrasos nos acessos envolvidos nas redes remotas. Em uma situação em que o banco de dados grande foi distribuído por diversos sites, existem bancos de dados de tamanho reduzido em cada site. Isso resulta em consultas e transações locais acessando dados em um único site com desempenho, o que resulta em melhora em função dos bancos de dados locais menores;

expansão mais fácil:

a vantagem de expansão é mais facilitada em ambientes distribuídos e está diretamente ligada à inclusão de mais dados, ao aumento do tamanho do banco de dados ou à inclusão de mais processadores.

Segundo Casanova e Moura (1999), usar SGBDDs é uma contribuição significativa para que seja aumentada a produtividade em desenvolvimento de aplicações, um ponto importante desde muito tempo. Para Lobato e Junior (2017), os bancos de dados distribuídos na era digital têm sua importância devido a características como localização (geograficamente distribuídos), a amplitude dos dados (alcance maior de usuários e informações) e principalmente a velocidade de transformação das redes de comunicação de dados a cada dia mais rápida e sofisticada. A distribuição e as transações nessa arquitetura de banco são transparentes. São exemplos de sistema de banco de dados distribuído o sistema de agências bancárias, o sistema de rede de supermercados etc.

**Banco de Dados Orientado a Objetos**

Você já deve ter escutado algo sobre orientadas a objetos ou então as linguagens de programação orientadas a objetos (LPOO). O termo orientado a objetos surgiu daí. Atualmente a orientação a objetos pode ter aplicação em outras áreas, como é o caso de banco de dados.

Um objeto pode ser caracterizado por duas partes importantes denominadas estado e comportamento. De acordo com Elmasri e Navathe (2011, p. 237), “[...] o objeto pode ter uma estrutura de dados complexa, bem como operações específicas definidas pelo programador. Os objetos em uma LPOO existem apenas na execução do programa, assim, eles são chamados de objetos transientes”.

O banco de dados orientado a objetos faz uso de objetos de maneira que sejam armazenados de forma permanente, portanto, passam a ser objetos persistentes, ou seja, permanecem existentes além da finalização do programa, podendo ser recuperados posteriormente e compartilhados por outros programas.

Elmasri e Navathe (2011) destacam que os principais conceitos usados nos bancos de dados orientados a objetos e no sistema objeto-relacional são:

* **identidade de objeto**: os objetos têm identidades únicas, que possuem interdependência nos seus valores de atributos e nas geradas pelo SGBDOO;
* **construtores de tipos**: estruturas de objetos complexas como vetors ou listas podem ser criadas e aplicadas a conjuntos de construção básica como *array*e *list*;
* **encapsulamento de operações**: com métodos públicos, pode-se acessar atributos e métodos internos ao objeto, garantindo uma independência de construção dos objetos.
* **compatibilidade de linguagem de programação**: objetos persistentes e transientes são tratados de maneira transparente. Os objetos se tornam persistentes ao serem alcançáveis de uma coleção persistente (extensão) ou pela nomeação explícita;
* **hierarquias de tipos e herança**: pode-se herdar características de objetos como seus atributos e métodos de outro objeto. Em alguns modelos, heranças múltiplas, ou seja, heranças de características de dois ou mais objetos, são permitidas;
* **extensões**: objetos de tipos semelhantes podem ser salvos em uma só extensão. Há restrições na hierarquia de tipos na relação conjunto/subconjunto relativas às suas extensões;
* **polimorfismo e sobrecarga de operador**: as operações e os nomes de método podem ser sobrecarregados para que se apliquem a diferentes tipos de objeto em diversas implementações.

Um banco de dados orientado a objetos é um sistema que consiste no armazenamento de objetos, da mesma forma como ocorre nas linguagens de programação orientadas a objetos. Uma diferença importante de ressaltar é que o objeto não deixará de existir com o encerramento do programa.

**Banco de Dados NoSQL**

De fato, o mais volumoso repositório de informação é a internet, assim contém os mais variados tipos de arquivos como arquivos de imagens, sons, vídeo, textos,  mas também das mais variadas extensões .mp4, .mp3, .jpg, .txt, etc. (PIVERT, 2018). Esses arquivos ou dados fazem parte da classificação de dados não estruturados.

Sistemas de banco de dados tradicionais relacionais ou orientados a objetos, mesmo distribuídos em rede, não conseguem trabalhar com essas novas estruturas de arquivos. Desde 1970, tem se operado uma evolução radical e rápida (revolução) dos sistemas de banco de dados durante as últimas décadas, sempre no sentido da adaptação das novas estruturas de arquivos.

De fato, ao longo das últimas décadas, várias pesquisas surgiram inspiradas no modelo de banco de dados relacional, apenas para citar como exemplo,  bancos de dados dedutivos, bancos de dados orientados a objetos e banco de dados distribuídos.  No cenário atual, tendo a internet como fonte de dados, novas ferramentas (tecnologia) emergem para sustentar os novos tipos de dados.

Surgem, então, os novos sistemas de banco de dados, os denominados NoSQL (*Not Only SQL*). Esses sistemas visam duas questões importantes: manipular  com dados estruturados, semiestruturados e não estruturados e gerenciar com eficiência a estrutura do *Big Data* em um distribuído, trabalhando com grandes tabelas de informações (PIVERT, 2018).

Características de um sistema NoSQL:

* **Utilização do processamento paralelo para processamento das informações**: faz uso de *MapReduce*para alcançar um desempenho em suas operações.
* **Distribuição em escala global**:  diversos *datacenters*espalhados pelo mundo estão disponíveis aos usuários.
* **Banco de dados que trabalham no esquema chave/valor**(Key/Value): fazem uso de tabela *hashing*distribuída, objetos são armazenados indexados por chaves, que são o elemento de busca desses objetos.

São exemplos de banco de dados NoSQL: MongoDB, Cassandra, Redis, Hbase, Amazon DynamoDB e Neok4j.

**praticar**

Vamos Praticar

A partir da década de 1970,  os primeiros conceitos de banco de dados foram escritos e baseados em dados estruturados, e mais ainda, década após década, novos tipos de sistemas gerenciadores de banco dados (SGBD) foram surgindo. Assim, na era da internet, há uma necessidade de novos modelos para trabalhar com dados não estruturados. Com base no texto, assinale a alternativa correta que apresenta uma tecnologia de banco de dados que trabalhe  numa estrutura de *Big Data*.

Parte superior do formulário

**a)**Banco de dados relacional.

**b)**Banco de dados NoSQL.

**c)**Banco de dados distribuído.

**d)**Banco de dados orientado a objetos.

**e)**Banco de dados armazenamento RDF.

Parte inferior do formulário

**indicações**

**Material Complementar**



**FILME**

**Minority Report**

**Ano**: 2002

**Comentário**:  O filme traz uma reflexão sobre informações que podem ser usadas tanto para o bem quanto para o mal. A história se passa no ano de 2054, quando um sistema possibilita que crimes sejam previstos com alto grau de precisão, fazendo com que o percentual de homicídios seja reduzido a zero. Tom Cruise vive o detetive John Anderton, responsável pelo combate ao crime, que descobre uma previsão de que ele irá cometer um assassinato, colocando em dúvida seu comportamento e sua ética ou a confiabilidade do sistema (banco de dados).

[TRAILER](https://unifacs.blackboard.com/bbcswebdav/institution/laureate/conteudos/CTI_PRABAD_20/unidade_4/ebook/index.html)



**LIVRO**

**Business Intelligence da Informação ao Conhecimento**

**Editora**: FCA

**Autores**: Maribel Yasmina Santos e Isabel Ramos

**ISBN**: 978-972-722-880-5

**Comentário**: Este livro aborda como sistemas de *Business Intelligence* (BI) fazem uso dos dados existentes nas organizações para gerar informação relevante para o processo de tomada de decisão.

**conclusão**

**Conclusão**

Neste módulo você compreendeu o conceito de Business Intelligence como uma solução capaz de organizar, analisar, gerenciar e monitorar dados e informações que darão o suporte ao processo de tomada de decisão. Ao longo deste material, você teve a oportunidade de: reconhecer as diferentes tecnologias na área de banco de dados; identificar os princípios básicos de Business Intelligence; comparar ferramentas como OLAP, ETL, Dashboard; classificar os tipos de banco de dados; compreender o conceito de Data Science; conhecer as características do Data Warehouse, banco de dados distribuídos, banco de dados orientado a objetos, banco de dados objeto-relacional e NoSQL.

**referências**

**Referências Bibliográficas**

ALEXANDER, M.; WALKENBACK, J. **Excel dashboards and reports**. Indianapolis: Wiley Publishing Inc., 2010.

BARBIERI, C. **Bi2 business intelligence**: modelagem e qualidade. Rio de Janeiro: Elsevier, 2011.

CASANOVA, M. A.; MOURA, A. V. **Princípios de sistemas de gerência de bancos de dados distribuídos**. Rio de Janeiro: PUC-Rio, 1999.

CURTY, R. G.; CERVANTES, B. M. N. Data science: ciência orientada a dados. **Inf. Inf**., Londrina, v. 21, n. 2, p. 1-3, maio/ago. 2016.

DAVENPORT, T. H.;  PATIL, D. J. Data scientist: the sexiest job of the 21st century. **Harvard Business Review**. [S.I.], oct. 2012. Disponível em: <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>. Acesso em:  15 jan. 2020.

DA SILVA, R. A.; SILVA, F. C. A.; GOMES, C. F. S. O uso do business intelligence (BI) em sistema de apoio à tomada de decisão estratégica. **Revista GEINTEC**, Aracaju, v. 6, n. 1, p. 2780-2798, 2016.

ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

GOUVEIA, H. C. et al. Aplicação da ferramenta OLAP em diferentes módulos de um sistema ERP melhorando a tomada de decisão. **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, Franca, v. 1, n. 1, 2011.

KAUSHIK, A. **Web Analytics 2.0**. Indianapolis: Wiley Publishing Inc., 2010.

KELLEHER, J. D.; TIERNEY, B. **Data science**.  Cambridge: The MIT Press, 2018.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de informações gerenciais**. São Paulo: Pearson, 2009.

LOBATO, D. C.; JÚNIOR, S. N. D. Aplicações de bancos de dados distribuídos utilizando a filosofia cliente/servidor. **Revista Traços**, Belém, v. 1, n. 1, 2017.

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

NAPOLI, M. **Aplicação de ontologias para apoiar operações analíticas sobre fontes estruturadas e não estruturadas**. 2011. 133 f. Dissertação (Mestrado em Engenharia e Gestão do Conhecimento) - Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2011.

PIVERT, O. **NoSQL data models**: trends and challenges. London: ISTE, 2018. v.1.

SANTOS, M. Y.; RAMOS, I. **Business intelligence da informação ao conhecimento**. Lisboa: FCA, 2017.

SHARDA, R.; DELEN, D.; TURBAN, E. **Business Intelligence e análise de dados para gestão do negócio**.  4. ed.  Porto Alegre: Bookman, 2019. p. 17.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2006.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2012.

TURBAN, E. et al. **Business intelligence**: um enfoque gerencial para a inteligência do negócio. São Paulo: 2009.

VICCI, C. **Banco de dados**. São Paulo: Pearson, 2014.

© 2020 - LAUREATE - Todos direitos reservados