

## **Aula 10**

*SERPRO (Analista - Especialização:  
Tecnologia) Bizu Estratégico - 2023  
(Pós-Edital)*

Autor:

**Elizabeth Menezes de Pinho Alves,  
Leonardo Mathias, Paulo Júnior,  
Aline Calado Fernandes**

23 de Maio de 2023

## BIZU ESTRATÉGICO DE BANCO DE DADOS SERPRO

Olá, prezado aluno. Tudo certo?

Neste material, traremos uma seleção de *bizus* da disciplina de **Banco de Dados** para o concurso do **SERPRO**.

O objetivo é proporcionar uma revisão rápida e de alta qualidade aos alunos por meio de tópicos que possuem as maiores chances de incidência em prova.

Todos os *bizus* destinam-se a alunos que já estejam na fase bem final de revisão (que já estudaram bastante o conteúdo teórico da disciplina e, nos últimos dias, precisam revisar por algum material bem curto e objetivo).

*Leonardo Mathias*



*@profleomathias*



## ANÁLISE ESTATÍSTICA

Pessoal, segue abaixo uma análise estatística dos assuntos mais exigidos pela banca **CEBRASPE** no âmbito da disciplina de **Banco de Dados**, além de cadernos com **questões** focadas na banca:

Banco de Dados		Incidência
Aula	Assuntos	Analista
0	1 Banco de dados. (Introdução)	7,50%
1	1.2 Modelagem de banco de dados: físico, lógico e conceitual. (Modelo Conceitual)	1,25%
2	1.2 Modelagem de banco de dados: físico, lógico e conceitual. (Modelo Relacional)	32,50%
3	1.3 Álgebra relacional, SQL/ANSI e linguagens procedurais embarcadas.	22,50%
4	1.1 Arquitetura de banco de dados: relacional ( Oracle)	5,00%
5	1.1 Arquitetura de banco de dados: relacional ( SqlServer)	0,00%
6	1.1 Arquitetura de banco de dados: relacional (PostgreSQL)	7,50%
7	2 Ciência de dados. 2.1 Big Data.	16,25%
8	2.3 Processamento de Dados. 2.3.1 Conceitos de processamento massivo e paralelo. 2.3.2 Processamento em lote (batch). 2.3.3 Processamento em tempo real (real time). 2.3.4 Processamento MapReduce.	3,75%
9	1.2 Aprendizado de máquina. (Aprendizado Supervisionado)	3,75%
10	1.2 Aprendizado de máquina. (Aprendizado NÃO Supervisionado)	
11	1.4 Visão computacional. 1.5 Deep learning.	0,00%
12	1.3 Processamento de linguagem natural (PLN).	0,00%
13	Linguagem R	0,00%
14	1.1 Análise de dados (Pandas, NumPy).	0,00%



Banco de Dados – SERPRO		
Assunto	Bizu	Caderno de Questões
Banco de Dados	1	<a href="http://questo.es/ilw90x">http://questo.es/ilw90x</a>
Modelagem de Banco de Dados: Físico, Lógico e Conceitual (Modelo Relacional)	2	<a href="http://questo.es/kmz49n">http://questo.es/kmz49n</a>
Álgebra Relacional, SQL/ANSI e linguagens procedurais embarcadas	3	<a href="http://questo.es/1acnpx">http://questo.es/1acnpx</a>
Ciência de Dados. Big Data.	4	<a href="http://questo.es/0h9395">http://questo.es/0h9395</a>
Arquitetura de Banco de Dados: Relacional (PostgreSQL)	5	<a href="http://questo.es/esvq71">http://questo.es/esvq71</a>



## Banco de Dados

### 1. BANCOS DE DADOS

- **Dados:** fatos brutos, que não foram organizados, processados, relacionados, avaliados ou interpretados.
- **Informação:** é um dado acrescido de contexto, relevância e propósito.
- **Conhecimento:** é uma informação contextual, relevante e acionável.

#### Tipos de Dados:

- **Não estruturados:** não possuem uma estrutura definida. Podem ser listados como exemplo documentos, textos, imagens e vídeos.
- **Estruturados:** são aqueles quem possuem a mesma estrutura de representação rígida e previamente projetada, ou sejam, existe um esquema que descreve as características dos dados que serão armazenados .
- **Semiestruturados:** apresentam uma organização bastante heterogênea , por exemplo o XML.

ESCLARECENDO!



Há dois tipos de redundância de dados:

**Redundância controlada de dados:** Acontece quando o software tem conhecimento da múltipla representação da informação e garante a sincronização entre as diversas representações.

**Redundância não controlada:** Acontece quando a responsabilidade pela manutenção da sincronia entre as diversas representações de uma informação está com o usuário e não com o software.



Sistema de Gerenciamento de Banco de Dados (SGBD) para atuar como um guardião do banco de dados, que substituiu a abordagem de arquivos. Vejamos uma figura para esclarecer esses contextos:

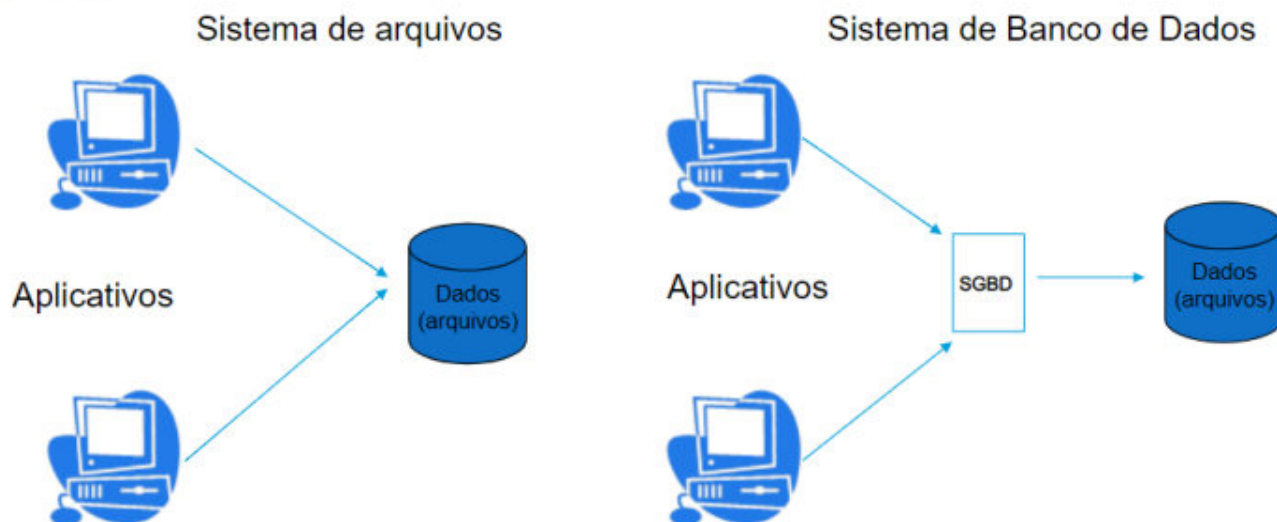


Figura 1 – Sistemas de arquivos e sistema de banco de dados

**Banco** tem diversas definições possíveis no dicionário da língua portuguesa. Dentre elas a que melhor se encaixa no nosso contexto é um conjunto organizado e categorizado de objetos, por exemplo, podemos ter um banco de fotografias ou um banco de leite.

**Dados** são  fatos conhecidos  que podem ser registrados e possuem um significado implícito. Esse conceito, porém, é um pouco amplo e abstrato para nosso intuito. Quando reduzimos o escopo à tecnologia da informação, temos um conceito mais adequado para dado. Ele é a representação física de um evento no tempo e espaço que não agrega fundamento ou significado para quem o sente ou recebe. É, basicamente, um registro!





O banco de dados, por si só, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma **coleção de arquivos de dados computadorizados**. – C J Date

Banco de dados é um conjunto de dados integrados que tem por objetivo **atender a uma comunidade de usuários** - Carlos Heuser.

Banco de dados é um conjunto de dados **estruturados** que são confiáveis, coerentes e compartilhados por usuários que têm necessidades de informações diferentes. - Silberchatz

Acho que você já entendeu o conceito de banco de dados! Na lista acima, você conheceu todas as definições que podem aparecer na sua prova. :)

## 2. Modelo Relacional

- O modelo relacional retrata os dados como sendo armazenados em **tabelas retangulares bidimensionais**, chamadas de **relações**.

Nome da relação <b>ALUNO</b>						
Atributos						
Nome	Cpf	Telefone_residencial	Endereco	Telefone_comercial	Idade	Media
Bruno Braga	305.610.243-51	(17)3783-1616	Rua das Paineiras, 2918	NULL	19	3,21
Carlos Kim	381.620.124-45	(17)3785-4409	Rua das Goiabeiras, 125	NULL	18	2,89
Daniel Davidson	422.111.232-70	NULL	Avenida da Paz, 3452	(17)4749-1253	25	3,53
Roberta Passos	489.220.110-08	(17)3476-9821	Rua da Consolação, 265	(17)3749-6492	28	3,93
Barbara Benson	533.690.123-80	(17)3239-8461	Rua Jardim, 7384	NULL	19	3,25

Cada linha de uma tabela é conhecida como uma **tupla**, ou uma coleção de valores relacionados. Cada coluna é vista como um **atributo**, cujo valor pertence um determinado conjunto de valores possíveis: o **domínio**.



## Álgebra Relacional

- É uma coleção de operações de alto nível sobre relações ou conjuntos cujo resultado é uma nova relação ou conjunto.

OPERAÇÃO	SÍMBOLO	COMUTATIVA	UNÁRIA/BINÁRIA	FINALIDADE
SELEÇÃO	$\sigma(T_1)$	SIM	UNÁRIA	Seleciona todas as linhas que satisfazem a condição de seleção de uma Tabela $T_1$ .
PROJEÇÃO	$\Pi(T_1)$	NÃO	UNÁRIA	Produce uma nova tabela com apenas algumas das colunas de uma tabela $T_1$ e remove linhas duplicadas.
PRODUTO CARTESIANO	$T_1 \times T_2$	SIM	BINÁRIA	Produce uma nova tabela com todas as combinações possíveis de linhas de duas tabelas $T_1$ e $T_2$ .
JUNÇÃO	$T_1 \bowtie T_2$	SIM	BINÁRIA	Produce uma nova tabela com todas as combinações possíveis de linhas de duas tabelas $T_1$ e $T_2$ que satisfazem uma condição de seleção.
UNIÃO	$T_1 \cup T_2$	SIM	BINÁRIA	Produce uma nova tabela que inclui todas as linhas das Tabela $T_1$ e $T_2$ , eliminando as duplicatas – as tabelas devem ser união-compatíveis.
INTERSECÇÃO	$T_1 \cap T_2$	SIM	BINÁRIA	Produce uma tabela que inclui todas as linhas em <u>comum</u> das Tabela $T_1$ e $T_2$ – as tabelas devem ser união-compatíveis.
DIFERENÇA	$T_1 - T_2$	NÃO	BINÁRIA	Produce uma tabela que inclui todas as linhas de uma Tabela $T_1$ que não estão na Tabela $T_2$ – as tabelas devem ser união compatíveis.

## TABELA COM O RESUMO DAS OPERAÇÕES DA ÁLGEBRA RELACIONAL.

OPERAÇÃO	FINALIDADE	NOTAÇÃO
SELEÇÃO	Seleciona todas as tuplas que satisfazem a condição de seleção de uma relação $R$ .	$\sigma_{\langle \text{condição seleção} \rangle}(R)$
PROJEÇÃO	Produce uma nova relação com apenas alguns dos atributos de $R$ , e remove tuplas duplicadas.	$\pi_{\langle \text{lista atributos} \rangle}(R)$
JUNÇÃO THETA	Produce todas as combinações de tuplas de $R_1$ e $R_2$ que satisfazem a condição de junção.	$R_1 \bowtie_{\langle \text{condição junção} \rangle} R_2$
EQUIJUNÇÃO	Produce todas as combinações de tuplas de $R_1$ e $R_2$ que satisfazem uma condição de junção apenas com comparações de igualdade.	$R_1 \bowtie_{\langle \text{condição junção} \rangle} R_2$ OR $R_1 \bowtie_{(\text{atributos junção 1} >), (\text{atributos junção 2} >)} R_2$
JUNÇÃO NATURAL	O mesmo que EQUIJOIN, exceto que atributos de junção de $R_2$ não são incluídos na relação resultante; se os atributos de junção tiverem os mesmos nomes, eles nem sequer precisam ser especificados.	$R_1 \star_{\langle \text{condição junção} \rangle} R_2$ OR $R_1 \star_{(\text{atributos junção 1} >), (\text{atributos junção 2} >)} R_2$ OR $R_1 \star R_2$
UNIÃO	Produce uma relação que inclui todas as tuplas em $R_1$ ou $R_2$ ou tanto $R_1$ quanto $R_2$ ; $R_1$ e $R_2$ precisam ser compatíveis na união.	$R_1 \cup R_2$
INTERSECÇÃO	Produce uma relação que inclui todas as tuplas em $R_1$ e $R_2$ ; $R_1$ e $R_2$ precisam ser compatíveis na união.	$R_1 \cap R_2$
DIFERENÇA	Produce uma relação que inclui todas as tuplas em $R_1$ que não estão em $R_2$ ; $R_1$ e $R_2$ precisam ser compatíveis na união.	$R_1 - R_2$
PRODUTO CARTESIANO	Produce uma relação que tem os atributos de $R_1$ e $R_2$ e inclui como tuplas todas as possíveis combinações de tuplas de $R_1$ e $R_2$ .	$R_1 \times R_2$
DIVISÃO	Produce uma relação $R(X)$ que inclui todas as tuplas $t(X)$ em $R_1(Z)$ que aparecem em $R_1$ em combinação com toda tupla de $R_2(Y)$ , onde $Z = X \cup Y$ .	$R_1(Z) \div R_2(Y)$





## Views

- É um subconjunto de um banco de dados. Permite que se visualize apenas parte dos dados de uma tabela. A view **não** necessariamente existe em forma física, sendo muitas vezes uma tabela **virtual**.

## Índices

- É um mecanismo utilizado para melhorar a velocidade de acesso aos dados. O excesso de índices pode prejudicar o desempenho.

## Chaves

- Superchave: conjunto de um ou mais atributos que permite identificar uma tupla de forma exclusiva.
- Chave primária: é uma superchave mínima escolhida para identificar uma linha da tabela.
- Chave candidata: Superchaves de tamanho mínimo, candidatas a serem possíveis chaves primárias de uma tabela.
- Chave estrangeira: chave primária de um outra relação.



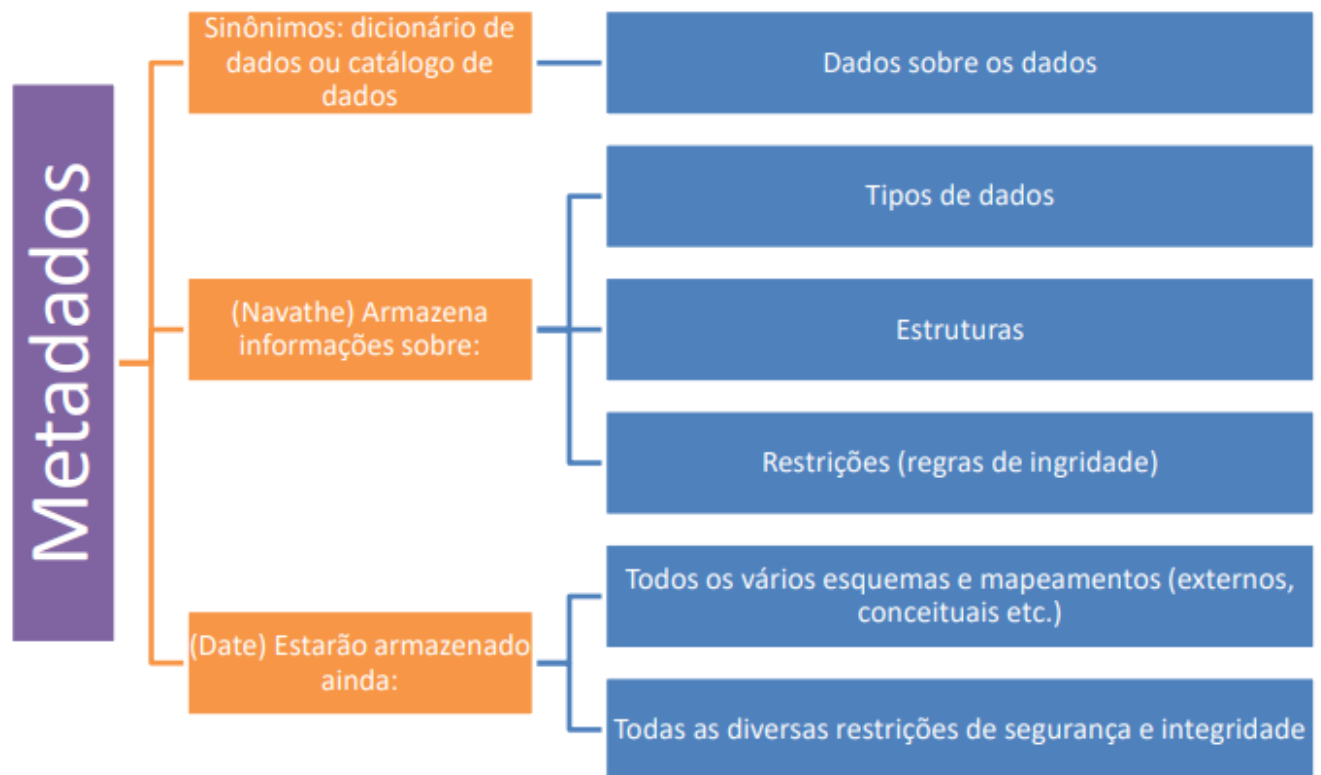
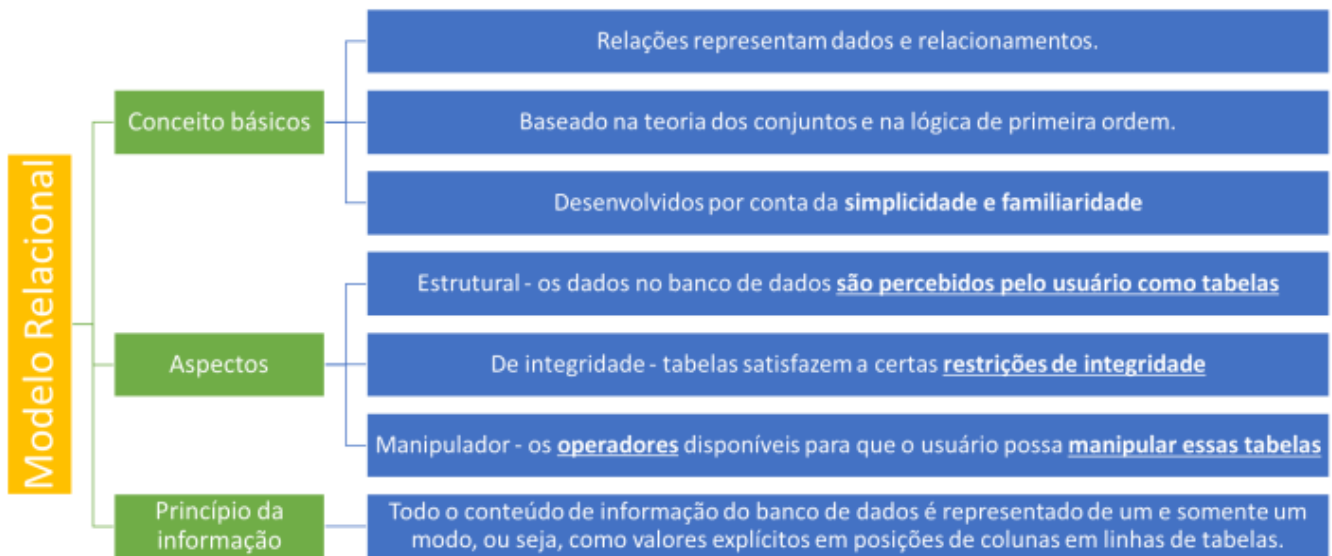


Figura 3 - Metadados, catálogo de dados ou dicionários de dados



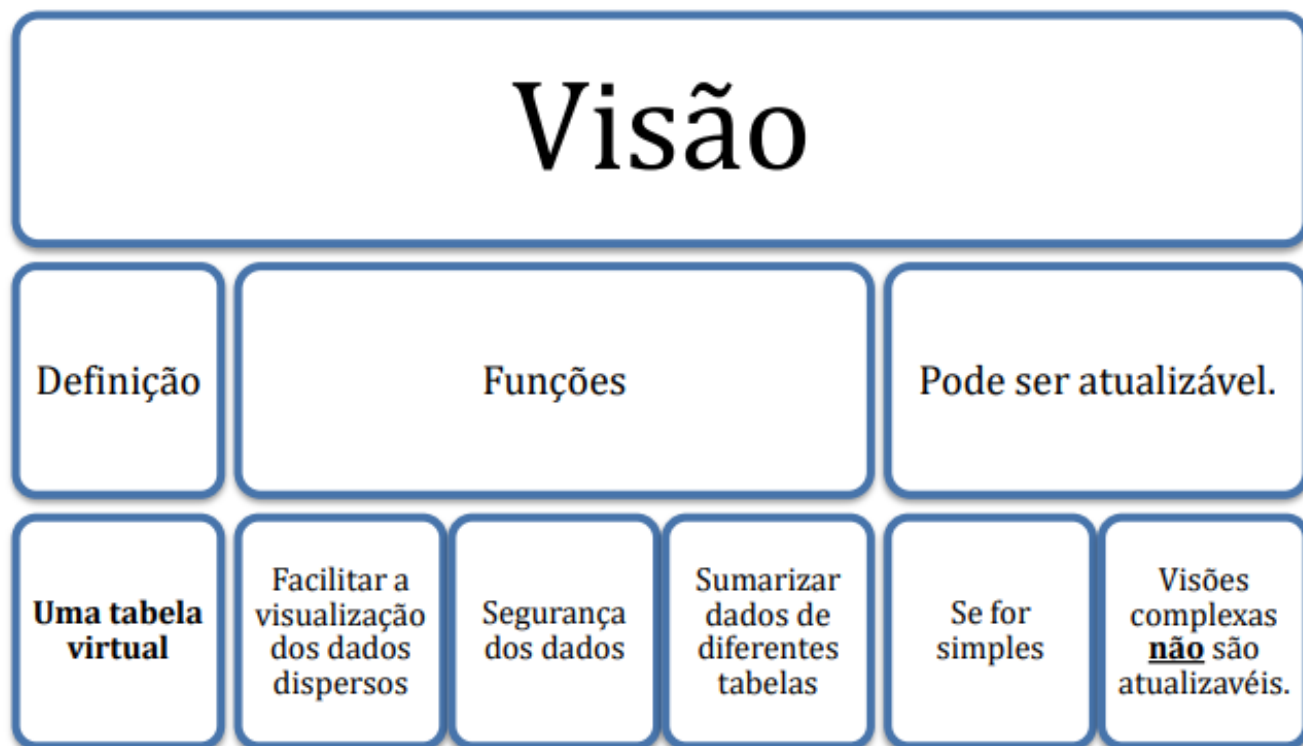


Figura 1 - Conceitos de Visão.

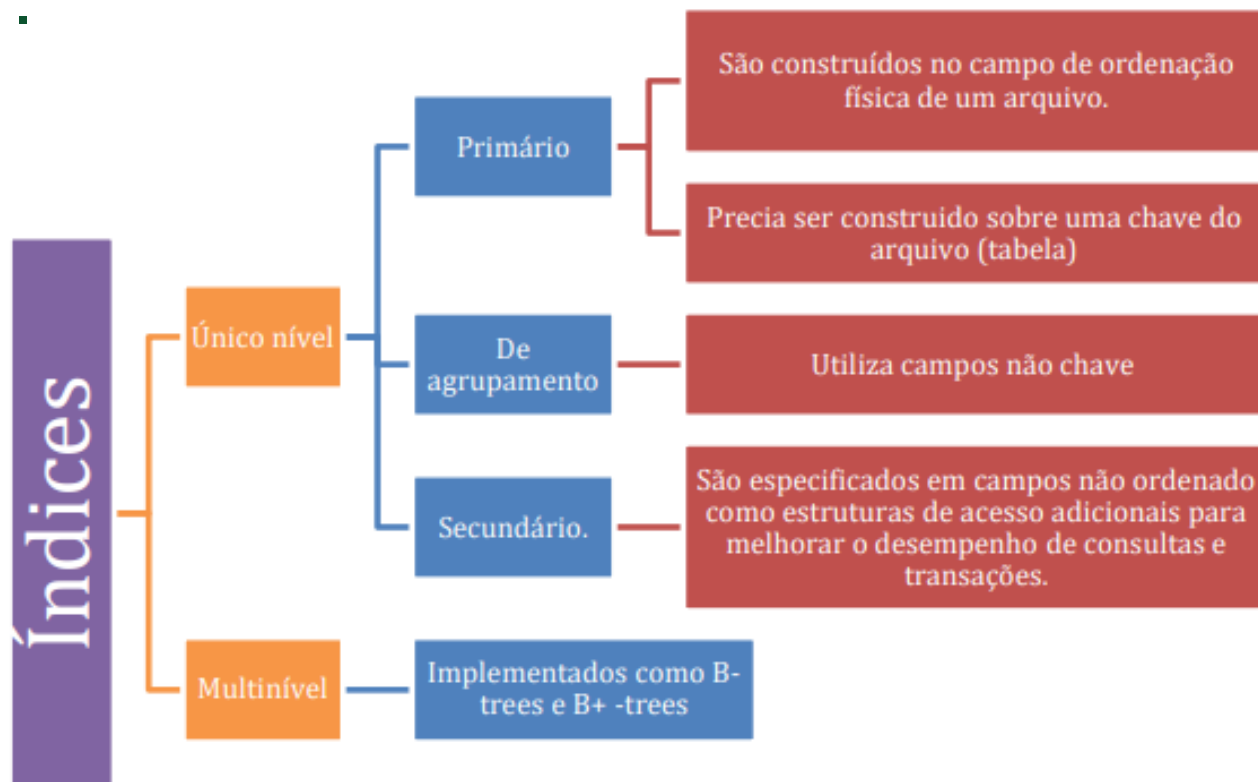


Figura 2 - Conceitos de Índices



## Operadores de conjuntos tradicionais

- **União , Interseção , Diferença e Produto cartesiano**
- Todos eles um pouco modificados para levar em conta o fato de que seus **operandos** são, especificamente, **relações** e não conjuntos arbitrários.

## Operadores relacionais especiais

- **De restrição (também conhecido como seleção), projeção, junção e divisão.**

### Primeira forma normal

- Uma relação R existe na primeira forma normal (1FN) se, e somente se, todos os **domínios subjacentes** contiverem apenas **valores atômicos**.
- No modelo relacional, um domínio é atômico se os elementos do domínio são considerados **unidades indivisíveis**. Um esquema de relação R está na primeira forma normal se **todos os atributos de R são atômicos**.
- A primeira forma normal afirma que o **domínio** de um atributo deve **incluir apenas valores atômicos** (**simples e indivisíveis**) e que o valor de qualquer atributo em uma tupla deve ser um único valor do domínio desse atributo.
- A primeira forma normal **evita** as chamadas relações aninhadas, essas relações contêm vários atributos em uma única coluna e **não** são permitidas no modelo relacional.



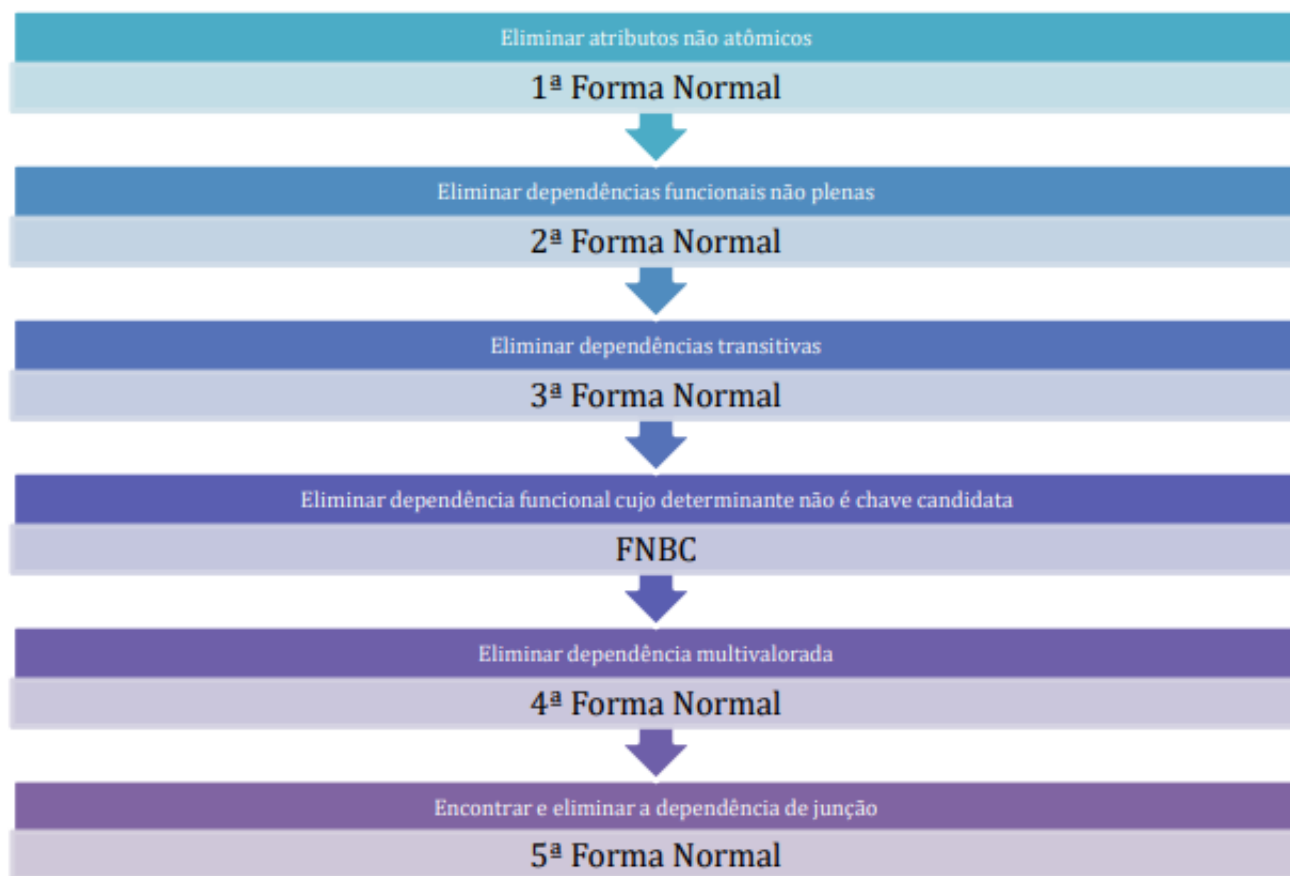


Figura 1 - Resumo das formas normais

### 3. Álgebra Relacional, SQL/ANSI e linguagens procedurais embarcadas

#### Pontos Importantes

Primeiramente, SQL é uma linguagem de programação reconhecida internacionalmente e usada para definição e manutenção de bancos de dados relacionais. A **principal característica da linguagem** é ser **declarativa**<sup>1</sup>, ou seja, os detalhes de implementação dos comandos são deixados para os SGBDs relacionais. Não esqueça disso! Já caiu várias vezes em provas anteriores. No SQL você **declara o que você quer e o SGBD vai achar os dados para você no banco, caso existam é claro**.





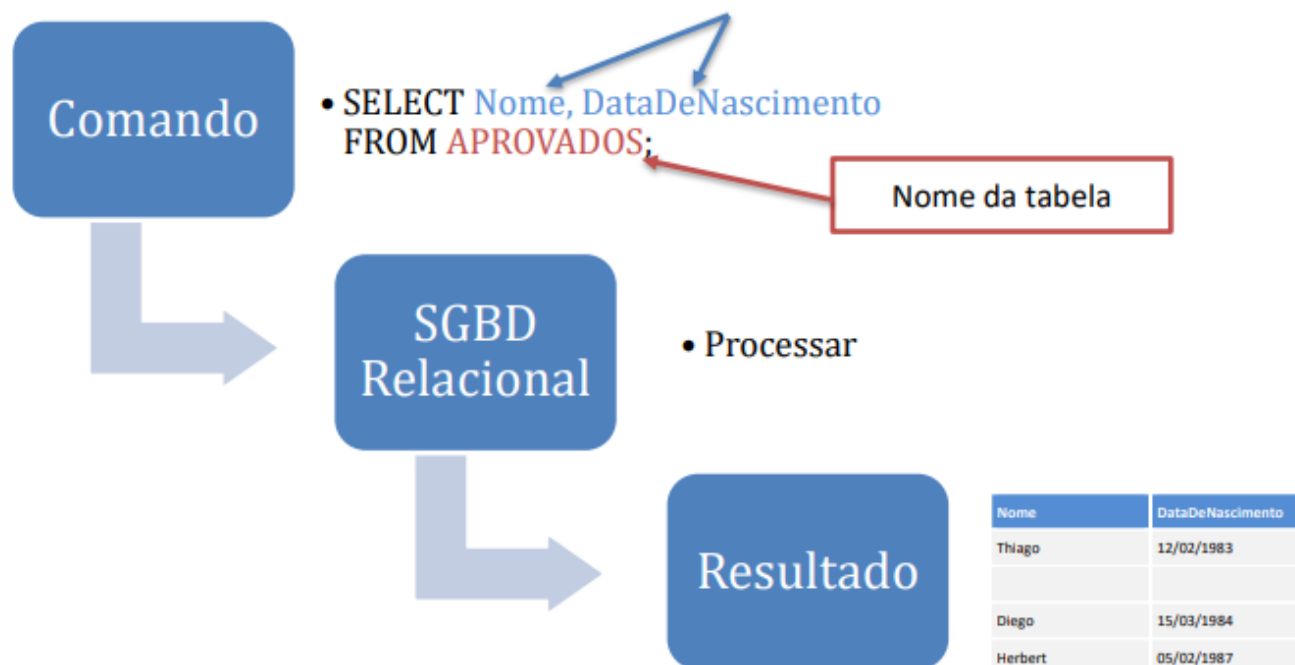


Figura 1 - Um exemplo de consulta

- **DDL** – Data Definition Language – A linguagem de definição de dados contém comandos que criam, modificam e excluem objetos de banco de dados. São exemplos de comando: CREATE, ALTER, DROP e TRUNCATE.
- **DML** – Data Manipulation Language – A linguagem de manipulação de dados fornece instruções para trabalhar com os dados armazenados como SELECT, INSERT, UPDATE, DELETE e MERGE.
  - **DQL** – Data Query Language – A linguagem de consulta de dados é um subconjunto da DML que possui apenas a instrução de SELECT.
- **DTL** – Data Transaction Language – Linguagem de transação de dados inclui comandos de COMMIT, ROLLBACK e SAVEPOINT
- **DCL** – Data Control Language – A linguagem de controle de dados contém os comandos relacionados com as permissões de controle de acesso. Garante os privilégios aos usuários para acessar os objetos do banco. Os mais conhecidos comandos são o GRANT e o REVOKE.



**PL/SQL** - Encontrado em SGBDs Oracle. PL/SQL, que é a sigla para Procedural Language/SQL, contém muitas semelhanças com a linguagem de programação geral.

**Transact-SQL** - Usado pelo Microsoft SQL Server e Sybase Adaptive Server. Como Microsoft e Sybase se afastaram, a plataforma comum que compartilhavam no início na década de 1990 foi dividida, suas implementações de agora também divergem, produzindo dois dialetos distintos de Transact-SQL.

**SQL PL** - extensão processual do IBM DB2 para SQL, introduzido na versão 7.0 do SGBD, fornece construções necessárias para a execução de controle de fluxo em torno de consultas SQL tradicionais e operações.

**PL/pgSQL** - O nome do dialeto SQL e das extensões implementadas no PostgreSQL. A sigla significa Procedural Language/PostgreSQL.

**MySQL** – O MySQL introduziu uma linguagem procedural em seu banco de dados na versão 5, mas não há nenhum nome oficial para ela. Ela segue o padrão definido pela ANSI (SQL/PSM).

PL/SQL	SQL
Os blocos de códigos podem ser usados para escrever programas inteiros.	Os comandos são usados para executar operações de DML e DDL.
Por ser <b>procedural</b> , devemos descrever <b>COMO</b> as coisas devem ser feitas.	Por ser <b>declarativa</b> , devemos definir apenas <b>O QUE</b> deve ser feito.
São executados em blocos inteiros	São executados em declarações únicas. ( <i>statements</i> ).
Usados para criar aplicações.	Usados para manutenção dos dados.
Uma extensão de SQL, é permitido o uso de código SQL dentro do código PL/SQL	Não pode conter código PL/SQL.

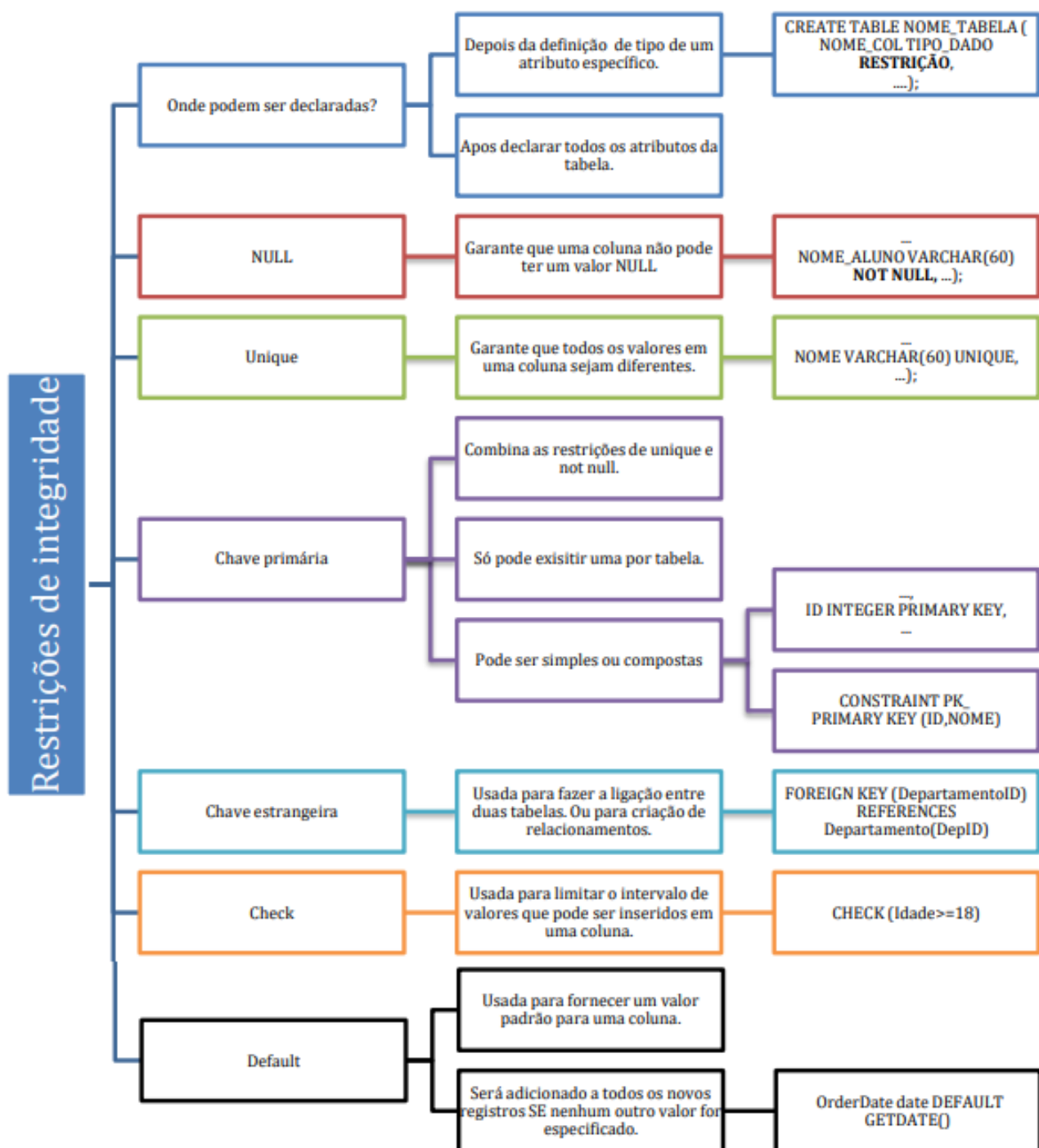
Resumindo os principais comandos usadas para definição de objetos em SQL, conhecidos como DDL (Data Definition Language) são:

COMANDO	DESCRIÇÃO
CREATE	Comando utilizado para criar bancos de dados, tabelas, índices, entre outros.
DROP	Comando utilizado para deletar uma tabela do banco de dados.
TRUNCATE	Comando utilizado para apagar os dados de uma tabela do banco de dados.
ALTER	Comando utilizado para adicionar, deletar ou modificar colunas do banco de dados.
RENAME <sup>1</sup>	Comando utilizado para renomear uma tabela do banco de dados.



- **Restrições em tabelas** – Definida dentro de uma definição de tabela. A restrição pode ser definida como parte da definição de coluna ou como um elemento ao final da definição. Restrições definidas no nível de tabela podem ser aplicadas a uma ou mais colunas.
- **Assertions (Afirmações)** - Um tipo de restrição que é definida dentro de uma definição afirmação (separado da definição da tabela). Uma afirmação pode ser relacionada com uma ou mais tabelas.
- **Restrições de domínio** - Um tipo de restrição que é definida dentro de uma definição de domínio (separado da definição da tabela). A restrição de domínio **está associada com qualquer coluna** que é definida baseada no domínio específico. Aqui você pode criar um domínio para atribuí-lo a uma coluna de uma tabela.





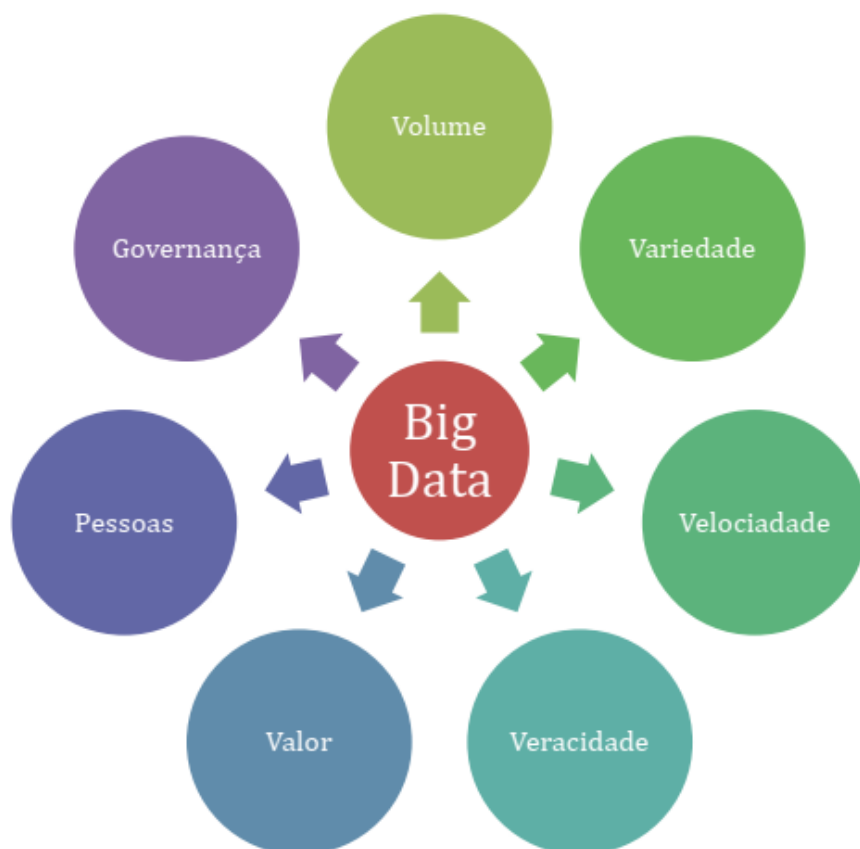
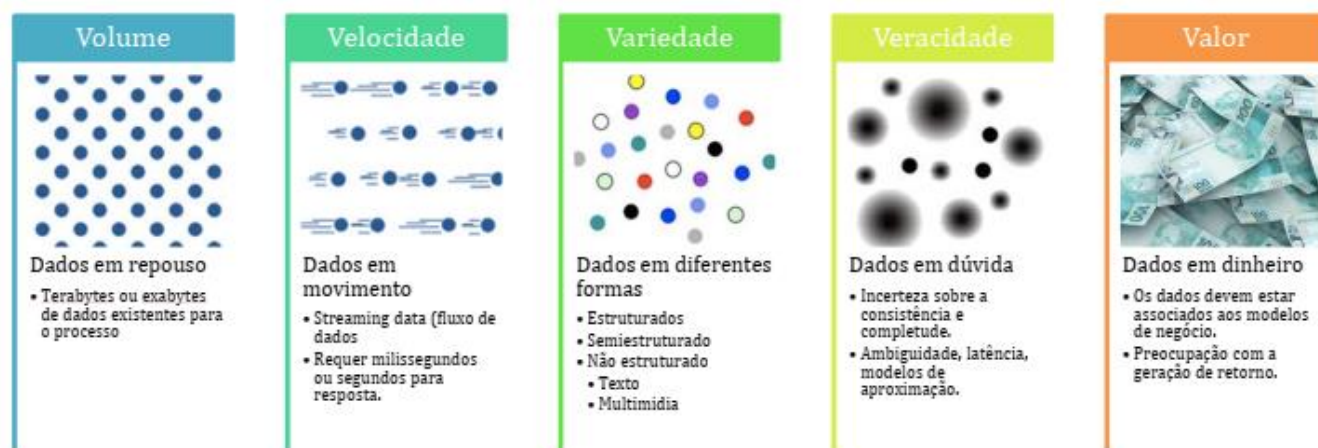
CLÁSULAS	DESCRIÇÃO
<b>SELECT</b>	Comando utilizado para selecionar dados de uma tabela.
<b>FROM</b>	Comando utilizado para indicar de onde os dados devem ser selecionados.
<b>WHERE</b>	Comando utilizado para filtrar os dados.
<b>GROUP BY</b>	Comando utilizado para agregar um conjunto de dados.
<b>HAVING</b>	Comando utilizado para filtrar dados agregados.
<b>ORDER BY</b>	Comando utilizado para ordenar os dados recuperados.



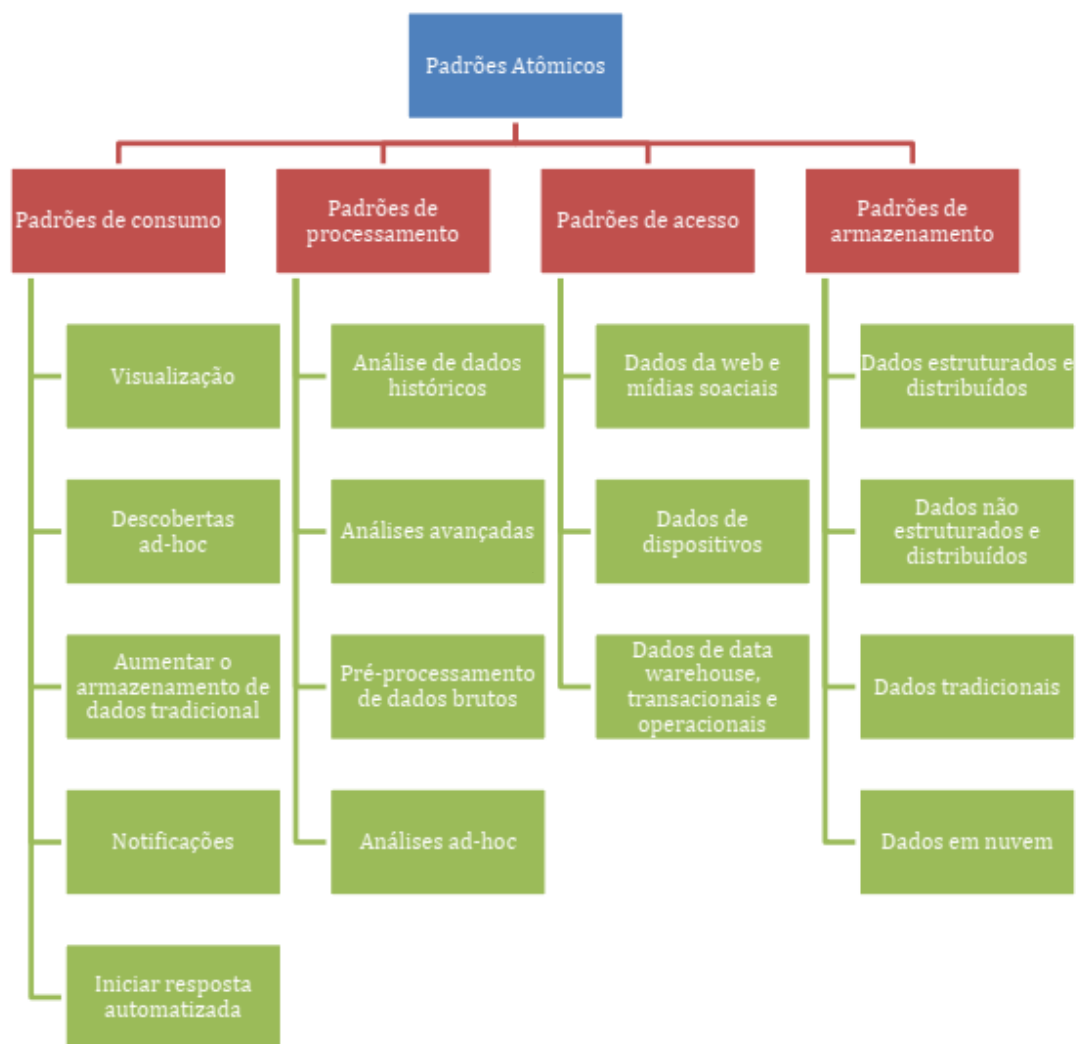


## 4. Ciência de Dados. Big Data.

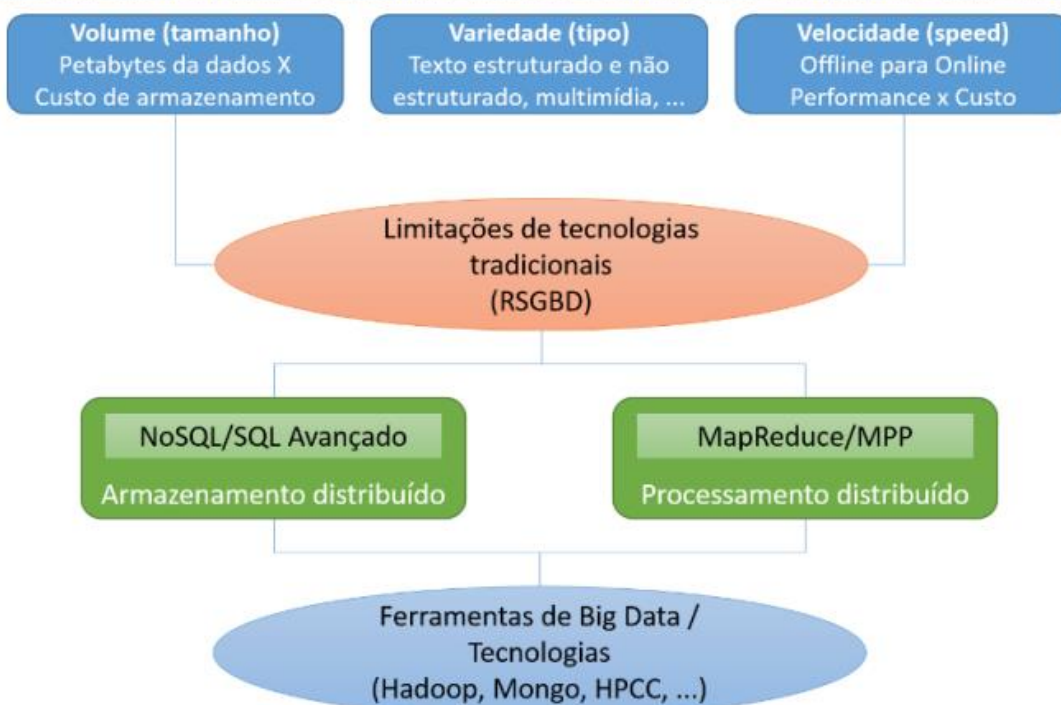
**Big Data** pode ser entendido como a **captura, gerenciamento e análise de dados que vão além de dados estruturados** típicos, que podem ser consultados por sistemas de gerenciamento de banco de dados relacional — frequentemente em arquivos não estruturados, vídeo digital, imagens, dados de sensores, arquivos de log e, na verdade, qualquer dado não contido nos registros com campos pesquisáveis distintos.







#### Conceitos de Big Data





#### 4. Arquitetura de Banco de Dados: Relacional (PostgreSQL)

O PostgreSQL é um descendente de código aberto do programa original desenvolvido em Berkeley. Ele **suporta** grande parte dos comandos da **linguagem SQL/ANSI** padrão e oferece muitas características modernas, entre elas: consultas complexas, chaves estrangeiras, *triggers*, visões atualizáveis, integridade transacional e controle de concorrência multiversão.

Uma instância do PostgreSQL é chamada de **cluster** porque uma única instância pode **servir e lidar com vários bancos de dados**. Cada banco de dados é um espaço isolado onde usuários e aplicativos podem armazenar dados. Um banco de dados é acessado por usuários autorizados, mas os usuários conectados a um banco de dados não podem cruzar os limites do banco de dados e interagir com os dados contidos em outro banco de dados, a menos que se conectem explicitamente ao último banco de dados também.



O PostgreSQL divide os usuários em duas categorias principais:

- **Usuários normais:** esses usuários são aqueles que podem se conectar e manipular bancos de dados e objetos, **dependendo de seu conjunto de privilégios**.
- **Superusuários:** Esses usuários podem fazer qualquer coisa com qualquer objeto de banco de dados.

Esta é uma rápida recapitulação dos principais termos usados no PostgreSQL:

- **Cluster:** refere-se a todo o serviço PostgreSQL.
- **Postmaster:** este é o primeiro processo que o cluster executa, e este processo é responsável por manter o controle das atividades de todo o cluster. O postmaster bifurca-se em um processo de back-end toda vez que uma nova conexão é estabelecida.
- **Banco de dados:** o banco de dados é um contêiner de dados isolado ao qual os usuários (ou aplicativos) podem se conectar. Um cluster pode lidar com vários bancos de dados. Um banco de dados pode ser feito por diferentes objetos, incluindo esquemas (*namespaces*), tabelas, gatilhos e outros objetos.
- **PGDATA:** é o nome do diretório que, no armazenamento persistente, é totalmente dedicado ao PostgreSQL e seus dados. O PostgreSQL armazena os dados nesse diretório.
- **WALs:** contém o log de intenção de alterações do banco de dados, usado para recuperar dados de um travamento crítico.

**/bin** - programas de linha de comando do PostgreSQL, como **psql**.

**/data** – arquivos de configuração e tabelas compartilhadas por todos os bancos de dados. Por exemplo, uma tabela **pg\_shadow** compartilhada por todos os bancos de dados.

**/data/base** - um subdiretório é criado para cada banco de dados. Usando os comandos **du** e **ls**, os administradores podem exibir a quantidade de espaço em disco usado por cada banco de dados, tabela ou índice.

**/doc** - documentação do PostgreSQL.

**/include** – inclui os arquivos usados por várias linguagens de programação.

**/lib** - bibliotecas usadas por várias linguagens de programação. Este subdiretório também contém os arquivos usados durante a inicialização, e exemplos e *templates* de arquivos de configuração, que podem ser copiados para **/data** e modificados.



- **checkpoint** é um processo responsável por executar os pontos de verificação, que são pontos no tempo em que o banco de dados garante que todos os dados sejam realmente armazenados de forma persistente no disco.
- **background writer** é responsável por ajudar a enviar os dados da memória para o armazenamento permanente.
- **walwriter** é responsável por gravar os Logs Write-Ahead (WALs), os logs que são necessários para garantir a confiabilidade dos dados, mesmo no caso de uma falha do banco de dados.
- **stats collector** é um processo que monitora a quantidade de dados que o PostgreSQL está manipulando, armazenando-os para uma análise e decisões posteriores, como decidir quais índices usar para satisfazer uma consulta.
- **logical replication launcher** é um processo responsável por lidar com a replicação lógica.

Autenticação	Descrição
<b>trust</b>	Permite a conexão incondicionalmente. Este método permite a qualquer pessoa se conectar ao servidor de banco de dados PostgreSQL e se autenticar com o usuário que desejarem, sem a necessidade de senha ou qualquer outra autenticação.
<b>reject</b>	Rejeita qualquer conexão incondicionalmente. Isso é útil para "filtragem" de certos hospedeiros de um grupo, por exemplo, uma linha de reject poderia bloquear um host específico para conexão, enquanto uma linha mais tarde permite que os hosts restantes possam se conectar.
<b>scram-sha-256</b>	Executa a autenticação SCRAM-SHA-256 para verificar a senha do usuário.
<b>md5</b>	Exige que o cliente forneça uma senha MD5 para autenticação.





<b>password</b>	Exige que o cliente forneça uma senha não criptografada para autenticação. Uma vez que a senha é enviada em texto simples através da rede, não deve ser usado em redes não confiáveis.
<b>gss</b>	Usa GSSAPI para autenticar o usuário. Este método só está disponível para conexões TCP/IP.
<b>sspi</b>	Usa SSPI para autenticar o usuário. O método só está disponível no Windows.
<b>ident</b>	Obtém o nome do usuário do sistema operacional do cliente entrando em contato com o servidor e verifica se ele corresponde ao nome do usuário do banco de dados. A autenticação ident só pode ser utilizada em conexões TCP/IP. Quando especificado para conexões locais, peer authentication será utilizado em seu lugar.
<b>peer</b>	Obtém o nome de usuário do sistema operacional do cliente e verifica se ele corresponde ao nome do usuário solicitado do banco de dados. Isto só está disponível para conexões locais.
<b>ldap</b>	Autentica o cliente usando um servidor LDAP.
<b>radius</b>	Autentica usando um servidor RADIUS.
<b>cert</b>	Autentica usando certificados de cliente SSL.
<b>PAM</b>	Autentica utilizando o serviço Pluggable Authentication Modules (PAM) fornecido pelo sistema operacional.

O PostgreSQL 12 gerencia os seguintes tipos de particionamento de tabela:

- **Range partitioning** - quando a tabela é dividida em "intervalos". Os intervalos não devem se sobrepor e a faixa é definida por meio do uso de um campo ou conjunto de campos. Aqui estamos falando de valores numéricos.
- **List partitioning** - a tabela será particionada usando uma lista de valores. Essa lista tende a ser um conjunto discreto, por exemplo, cidades ou time de futebol.
- **Hash partitioning** - a tabela será particionada usando um valor de hash que será usado como o valor para dividir os dados em tabelas diferentes.





O PostgreSQL possui um mecanismo de transação muito rico e compatível com os padrões que permite aos usuários definir exatamente as propriedades da transação, incluindo transações aninhadas.

O PostgreSQL depende muito de transações para manter os dados consistentes em conexões simultâneas e atividades paralelas e, graças aos **registros Write-Ahead (WALs)**, o PostgreSQL faz o possível para manter os dados seguros e confiáveis. Além disso, o PostgreSQL implementa o **Multi-Version Concurrency Control (MVCC)**, uma forma de manter a alta simultaneidade entre as transações.

Como não temos muitas transações simultâneas de conexões diferentes não temos como visualizar o que acontece com essas variáveis em um grande banco de dados corporativo. Mas vou apresentar abaixo uma tabela que descreve as colunas do sistema que influenciam nas transações.

Coluna	Descrição
xmin	A identidade (ID da transação) da transação de inserção para esta versão de linha. Aqui você começa a entender o MVCC. Toda vez que uma modificação é feita, uma nova linha é inserida. A versão anterior da linha vai ser excluída do banco de dados em um momento posterior por um processo conhecido como VACUUM.
xmax	Armazena o ID da transação ("xid") da transação que excluiu a tupla. Lembre-se de que UPDATE também exclui uma tupla no PostgreSQL. Então você passa a ter um intervalo entre o xmin e xmax no qual a tupla é válida. O xmax também pode armazenar a transação que está bloqueando a tupla.
cmin	O identificador de comando (começando em zero) dentro da transação de inserção.
cmax	O identificador de comando dentro da transação de exclusão ou zero.

Vamos ficando por aqui.

Esperamos que tenha gostado do nosso Bizu!

Bons estudos!

*Leonardo Mathias*



*@profleomathias*



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.