



Fundamento em Engenharia de Dados

Bootcamp: Engenheiro de Dados

Fernanda Farinelli

2021

Fundamentos

Bootcamp: Engenharia de Dados

Prof^a. Fernanda Farinelli

© Copyright do Instituto de Gestão e Tecnologia da Informação.

Todos os direitos reservados.

Sumário

Capítulo 1. Conceitos fundamentais de Big Data:	5
Dados, tipos de dados e fontes de dados	5
Big data	8
Web semântica	9
Dados abertos	10
Linked data e Linked open data	10
Ontologias	10
Pipeline de dados do Big Data e de Engenharia de Dados	13
O papel da Engenharia de Dados	15
Capítulo 2. Armazenamento de Dados	17
Fundamentos de banco de dados	17
Sistemas de Arquivos distribuídos	18
Armazenamento em nuvens	19
Capítulo 3. Modelagem de Dados	23
Modelo conceitual	24
Modelo entidade-relacionamento	25
Modelo lógico	27
Modelo relacional	28
Modelo de dados físico	29
Sistemas gerenciadores de bancos de dados relacionais	30
Modelo de dados relacional físico	31
Capítulo 4. Linguagem SQL	34
Capítulo 5. Data warehouse e modelagem dimensional	36
Modelo dimensional de dados	37

Extração, Transformação e Carga (ETC)	40
Online Analytical Processing.....	41
<i>Data lake</i> , data swamp e data pond.....	48
Capítulo 6. Bancos de Dados NOSQL	50
Sistemas gerenciadores de bancos de dados NoSQL.....	50
Teorema CAP	50
Propriedades BASE	52
Categorias de Bancos de Dados NoSQL	53
Capítulo 7. Fundamentos de Análise de dados	58
Principais tipos de análise de dados	58
Análise descritiva	58
Análise diagnóstica.....	59
Análise preditiva	59
Análise prescritiva.....	60
Análise Exploratória de Dados.....	60
População e amostra.....	61
Variável.....	61
Medidas	62
Web mining	63
Text mining	64
Capítulo 8. Coleta e preparação de dados	65
Coleta de dados.....	65
Preparação de dados.....	66
APIs de coleta de dados	67
Web Crawler e web scraping	68
Referências.....	69

Capítulo 1. Conceitos fundamentais de Big Data:

A evolução das tecnologias de informação e comunicação, assim como o surgimento da internet, mudou o dia a dia das pessoas trazendo as atividades humanas para o mundo virtual. A *World Wide Web*, a maior rede de informação global, em ritmo evolucionário, passou por fases que ficaram conhecidas como Web 1.0, Web 2.0 e Web 3.0 (SHIVALINGAIAH, NAIK, 2008). Vivemos cercados por uma grande quantidade de dados que apoiam as nossas decisões, tal disponibilidade oferece oportunidades para a obtenção de conhecimento, ou seja, ao submeter os dados a processos de análise, obtém-se informação e conhecimento útil nos processos decisórios das organizações (HEATH, BIZER, 2011).

Dados, tipos de dados e fontes de dados

Nas últimas décadas, os dados assumiram um papel vital na estratégia das empresas, tornando-se um dos grandes ativos existentes no patrimônio das organizações (DAMA, 2009, p. 1). Mas o que são dados? Veja no Quadro 1 o que são dados, informação e conhecimento.

Quadro 1 – Dados, informação e conhecimento.

Dado	Informação	Conhecimento
Simple observações sobre o estado do mundo.	Dados dotados de relevância e propósito.	Informação valiosa da mente humana.
Facilmente estruturado. Facilmente obtido por máquinas.	Requer unidade de análise. Exige consenso em relação ao significado.	Inclui reflexão, síntese e contexto. Difícil estruturação.

Frequentemente quantificado. Facilmente transferível.	Exige necessariamente a mediação humana.	Difícil captura em máquinas. Frequentemente tácito. Difícil transferência.
---	---	---

Fonte: Retirado de DAVENPORT (1998, p. 18).

Tal definição enfatiza o papel dos dados em representar fatos sobre o mundo, ou seja, *dados são fatos capturados, armazenados e expressos como texto, números, gráficos, imagens, sons ou vídeos* (DAMA, 2009, 2017). Nossos dados não se resumem a ações, podem ser objetos, localizações, quantidades, textos, imagens e áudios, ou qualquer coisa que possa ser digitalizada e armazenado em algum banco de dados (DAMA, 2017).

As atividades e ações virtuais e os diversos sistemas de informação produzem, coletam e analisam dados. Os dados podem vir de diferentes fontes de dados, ou seja, uma fonte é o local de onde o dado é coletado ou adquirido. Podem ser arquivos, banco de dados, portal de notícias ou até mesmo um *feed* de notícias. As fontes de dados podem ser qualquer dispositivo ou estrutura que forneça dados, localizada ou não no mesmo computador ou dispositivo que o programa de coleta.

Os dados podem assumir diferentes formatos ou tipos conforme sua origem,

a

Figura 1 apresenta um quadro resumizando comparativamente estes tipos de dados.

Figura 1 – Comparativo dos tipos de dados.



Os dados armazenados nos bancos de dados relacionais são considerados dados estruturados, como dados semiestruturados podemos citar os dados armazenados em arquivos XML e como dados não estruturados temos os dados originários de mídias sociais, imagens, vídeos e áudios. A maior parte dos dados que são coletados atualmente são dados não estruturados. Acredita-se que 95% dos dados gerados hoje são em formato não-estruturado (MAYER-SCHÖNBERGER,CUKIER, 2013:47).

Big data

Este volume de dados e suas diversas fontes e formatos levou ao fenômeno que ficou conhecido como *Big Data*. Termo cunhado em meados dos anos 90 por Michael Cox e David Ellsworth, cientistas da Nasa, que discutiram sobre os desafios da visualização de grandes volumes de dados, aos limites computacionais tradicionais de captura, processamento, análise e armazenamento (COX,ELLSWORTH, 1997). O termo *Big Data* é usado para descrever este grande conjunto de dados que desafia os métodos e ferramentas tradicionais para manipulação de dados considerando um tempo razoável de resposta. *Big Data* é

caracterizado pela tríade volume, variedade e velocidade (LANEY, 2001). Ainda temos duas importantes características, veracidade e valor (TAURION, 2013).

O **volume** diz respeito à quantidade de dados que são produzidos e coletados pelas organizações. As organizações coletam dados de diversas fontes, implicando na **variedade** dos tipos (estruturados, semiestruturados e não estruturados) e formatos dos dados coletados. A **velocidade** diz respeito tanto ao quão rápido os dados estão sendo produzidos e quão rápido os dados devem ser tratados para atender a demanda da organização. As decisões são tomadas em tempo real. Temos ainda a **veracidade** ou confiabilidade dos dados, ou seja, eles devem expressar a realidade e ser consistentes. Enfim, o **valor**, ou a utilidade dos dados ao negócio, como agregam valor (TAURION, 2013).

Web semântica

A Web Semântica é uma extensão da web que estrutura o significado de seu conteúdo de forma clara e bem definida, permitindo aos computadores interagir entre eles trocando informações. Sua principal motivação é ter uma web de dados, no qual tais dados sejam significativos tanto para os humanos quanto para as máquinas (BERNERS-LEE; HENDLER; LASSILA, 2001).

Na Web Semântica são os vocabulários que definem os conceitos e relacionamentos usados para descrever e representar uma área de interesse. Muitas vezes uma ontologia pode ser empregada quando se tem uma coleção de termos mais complexa e formal. O papel dos vocabulários, portanto, é o de auxiliar na integração dos dados, tratando os casos de ambiguidade de termos usados em diferentes bases de dados por exemplo.

Dados abertos

O conceito de dados abertos remete a ideia de conteúdo aberto, ou seja, disponível para todos. Dados abertos são dados que podem ser livremente publicados na web, seguindo alguns padrões predefinidos, e a partir de sua publicação podem ser reutilizados e redistribuídos por qualquer pessoa ou aplicativo, sujeitos, no máximo, à exigência de atribuição da fonte e compartilhamento pelas mesmas regras (ISOTANI,BITTENCOURT, 2015; OKI, 2019).

Linked data e Linked open data

Fundamenta-se na ideia de interligar dados na web ao invés de documentos. *Linked data* (LD), ou dados interligados, é uma forma de publicar dados na web de forma estruturada, de modo que uma pessoa ou máquina possa explorar estes dados. Relacionado à web semântica, propõe um conjunto de princípios, padrões e protocolos para serem adotados para publicar dados na web e para interligar os dados. As ligações permitem aos usuários da web navegar entre diferentes fontes. Além disso, as ferramentas de busca ficam aptas a indexar a web e fornece recursos de pesquisa mais sofisticados sobre o conteúdo rastreado (BERNERS-LEE, 2006; BIZER,HEATH,BERNERS-LEE, 2009; HEATH,BIZER, 2011).

Adicionalmente, temo o conceito de dados abertos interligados ou *Linked Open Data*, que remete a ideia de conteúdo aberto ou disponível para todos, mas com interconexões entre os dados, ou seja, são dados interligados que se encontram disponíveis livremente na web (BERNERS-LEE, 2006).

Ontologias

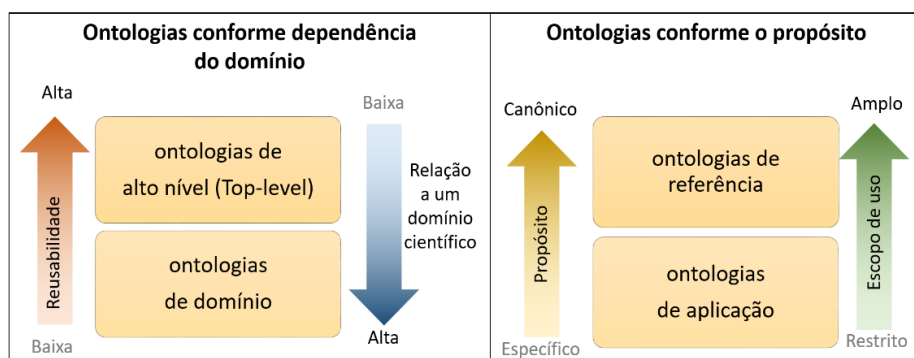
Ontologia é um termo polissêmico e objeto de pesquisa em diversas áreas, como: Filosofia, Ciência da Computação e Ciência da Informação. A palavra ontologia é derivada do grego, onde “*Onto*” exprime a noção do ser e “*Logia*” é algo dito ou a

maneira de dizer. Ela pode ser entendida com disciplina filosófica ou como artefato representacional. Como disciplina da Filosofia, a Ontologia estuda a natureza da existência das coisas. Como artefato representacional, a ontologia representa conhecimento acerca de vários domínios de conhecimento, através da formalização das relações entre termos e conceitos (ALMEIDA, 2013).

As ontologias se classificam conforme descrito abaixo e ilustrado na Figura 2 (FARINELLI, 2017).

- Ontologias de alto nível: descrevem conceitos amplos independentes de um domínio particular. Ex.: relacionadas a espaço, tempo, eventos etc.
- Ontologias de referência: descrevem conceitos relacionados a atividade ou tarefas genéricas, independentes de domínio. Ex.: diagnóstico.
- Ontologias de domínio: descrevem conceitos relacionados a domínios específicos, como direito, computação etc. É a categoria mais comum.
- Ontologias de aplicação: descrevem conceitos dependentes de um domínio e tarefa específicos.

Figura 2 – Classificação das ontologias.



Fonte: Traduzido de FARINELLI (2017).

Uma ontologia pode ser muito complexa, com milhares de conceitos ou muito simples, descrevendo apenas um ou dois conceitos. A especificação de uma ontologia inclui os seguintes elementos (FARINELLI, 2017):

- Entidade: é algo que você deseja representar em um domínio particular. Qualquer coisa que exista, existiu ou irá existir. Ex.: eventos, processos e objetos.
- Instância ou indivíduos: representam uma unidade de objetos específicos de uma entidade, ou seja, indivíduos de um determinado universal.
- Atributos: propriedades relevantes da entidade/classe ou instância que ajudam a descrevê-la.
- Relacionamento: descreve o tipo de interação entre duas classes, duas instâncias ou uma classe e uma instância.
- Cardinalidade: uma medida do número de ocorrências de uma entidade associada a um número de ocorrências em outra.
- Axioma: uma proposição lógica que é considerado verdadeiro. Restringem a interpretação e o uso das classes envolvidas na ontologia.

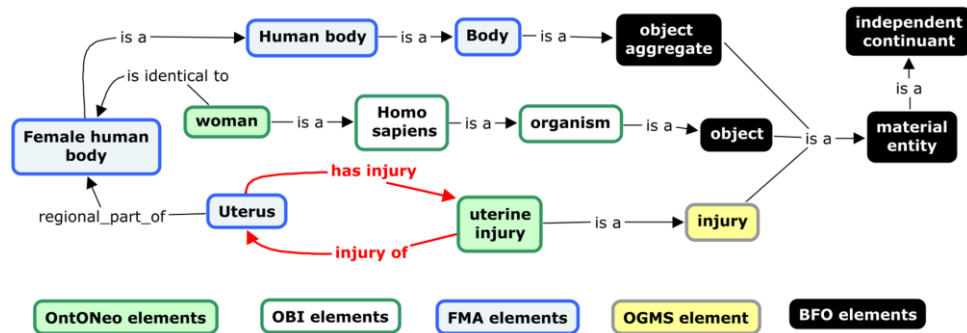
As ontologias descrevem entidades sobre a perspectiva dos universais e particulares. Os particulares ou indivíduos são ocorrências únicas de algo existente na realidade, por exemplo, cada um de nós é uma única ocorrência ou indivíduo de um "homo sapiens". Os universais ou tipos são entidades reais que generalizam os particulares existentes no mundo, por exemplo, "homo sapiens" é uma entidade geral ou universal referente aos particulares que cada um de nós é.

Um exemplo de ontologia é mostrado na .

Figura 3. Nesta ontologia é descrita uma pequena fração do domínio médico obstétrico, especificamente para descrever a relação de lesão uterina com o corpo

humano feminino. É possível se identificar as entidades ou classes representadas por elipses e as relações pelas setas.

Figura 3 – Parte da ontologia Ontoneo.



Fonte: FARINELLI (2017).

Pipeline de dados do Big Data e de Engenharia de Dados

Dados tradicionais e Big Data demandam processos de coleta, armazenamento, processamento, análise e visualização, porém a diferença se volta para as três características, volume, variedade e velocidade (TAURION, 2013). Ao longo das últimas três décadas, surgiram várias abordagens para mineração de dados, dentre elas cita-se as metodologias KDD, CRISP-DM e SEMMA (AZEVEDO, SANTOS, 2008; SHAFIQUE, QAISER, 2014). O pipeline de dados é o conjunto de processos e atividades que uma organização executa para extrair informações dos seus dados, capazes de subsidiar seu processo de tomada de decisão. Um exemplo de pipeline fundamentado nos processos de mineração de dados KDD, SEMMA e CRISP-DM, além da cadeia de valor proposta por CURRY (2016), aqui nomeado de cadeia de valor do *Big Data*, é apresentada na

Figura 4. Nesta cadeia de valor, o fluxo de informações é descrito como uma série de etapas necessárias para gerar valor e informações úteis dos dados.

Figura 4 – Cadeia de Valor de Big Data.



A grande responsabilidade de uma equipe de engenharia de dados é construir todo o pipeline de dados (data flow), principalmente nos processos de coleta, modelagem, pré-processamento ou preparação e armazenamento de dados. Cada um destes processos será tratado nos capítulos seguintes.

O papel da Engenharia de Dados

A Engenharia de Dados é a área responsável por planejar, criar, manter e evoluir toda a estrutura de dados de uma organização (NASCIMENTO, 2017). É neste cenário que o profissional de engenharia de dados ganha importância no mundo de Big Data. Este profissional deve:

- Entender a origem e natureza dos dados.
- Lidar o planejamento e desenvolvimento do esquema de dados.
- Definir estruturas confiáveis para suportar todo o fluxo de dados e implementar a coleta, preparação e armazenamento.
- Propor e implementar a estrutura de armazenamento e soluções de *Data-Warehouse*.

- Entender a necessidade de integração de dados.
- Propor e implementar a estrutura de integração e rotinas de ETL.

Estes profissionais planejam e criam a infraestrutura necessária para receber o fluxo de dados desde sua aquisição, transformação, armazenamento, até o processamento e visualização de dados. São os responsáveis por prover a arquitetura que deve acolher o ciclo de vida dos dados. A estrutura projetada pelo engenheiro de dados geralmente envolve as operações listadas (PARUCHURI, 2017):

- Aquisição ou Ingestão (Data Ingestion): envolve atividades e estrutura para coleta dos dados de diversos formatos e origens.
- Processamento: envolve infraestrutura capaz de suportar o processamento dos dados, levando em conta o volume e a variedade, para obter os resultados finais desejados em tempo hábil.
- Armazenamento: envolve infraestrutura capaz de armazenar os dados coletados e os resultados dos dados processados ou analisados, visando inclusive a recuperação rápida destes dados.
- Acesso: envolve uma estrutura voltada para segurança cujo o usuário esteja habilitado a acessar os resultados finais das análises.

A estrutura projetada dever ser projetada para ser escalável, para suportar o volume de dados e crescimento da demanda por armazenamento, além da necessidade de suportar o armazenamento e processamento dos dados dos diversos formatos, assim como garantir a confiabilidade, integridade e segurança dos dados. Esta estrutura, o projeto de arquitetura de dados, foca na transformação dos dados em um formato útil para análise. A engenharia de dados, assim com a ciência de dados, é um campo amplo abrangendo uma infinidade de habilidades, plataformas e ferramentas. No entanto, um único engenheiro de dados não precisa conhecer o espectro completo, em geral, ele trabalha em conjunto com outros engenheiros e analistas (PARUCHURI, 2017).

Capítulo 2. Armazenamento de Dados

Os sistemas de informação não são nada sem os dados e estes precisam estar armazenados em algum repositório. O conceito de armazenamento de dados é muitas vezes relacionado à persistência de dados, ou seja, o dado deve ser persistido ou armazenado em local não volátil, de forma que eles possam ser recuperados posteriormente para consulta e análise. Isso significa armazenar estes dados em um local que possa garantir a integridade dos dados por um período de tempo indeterminado, até que eles sejam atualizados ou descartados propositalmente.

Fundamentos de banco de dados

Este conjunto de dados armazenados são conhecidos como Banco de Dados. Um banco de dados é “uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. Um sistema de banco de dados como o conjunto de quatro componentes básicos: dados, hardware, software e usuários (ELMASRI,NAVATHE, 2005; SILBERSCHATZ,KORTH,SUDARSHAN, 2012).

Para suportar as necessidades dos sistemas de bancos de dados, foram criados os Sistema Gerenciadores de Banco de Dados (SGBD ou em inglês Data Base Management System - DBMS). SGBDs são sistemas ou softwares utilizados para gerir os bancos de dados, permitindo: i) criar, modificar e eliminar bases de dados; ii) realizar as operações básicas com os dados (inserir, alterar, excluir e consultar); iii) garantir a segurança de acesso aos dados; iv) garantir a integridade de dados, controle de concorrência e possibilidades de recuperação e tolerância a falhas (SILBERSCHATZ,KORTH,SUDARSHAN, 2012).

Os objetivos de um sistema de banco de dados são o de isolar o usuário dos detalhes internos do banco de dados (promover a abstração de dados) e promover a independência dos dados em relação às aplicações, ou seja, tornar independente da aplicação a estratégia de acesso e a forma de armazenamento. Existem diversos

paradigmas tecnológicos de SGBDs, como por exemplo os SGBDs em rede, hierárquicos, relacionais, NoSQL. Vamos tratar dos SGBDs relacionais e NoSQL nos capítulos seguintes.

Sistemas de Arquivos distribuídos

Quando uma base de dados atinge a capacidade máxima de espaço provida por uma única máquina física, torna-se necessário distribuir esta responsabilidade com um determinado número de computadores. Sistemas que gerenciam o armazenamento de arquivos em uma ou mais máquinas interligadas em rede são denominados sistemas de arquivos distribuídos. Assim, os Sistemas de Arquivos Distribuídos (SADs) são sistemas de arquivos que permite aos programas armazenarem e acessarem dados ou arquivos, espalhados (distribuídos) em uma rede de máquinas como se fossem locais, possibilitando que os usuários acessem arquivos a partir de qualquer computador em uma rede. Questões como desempenho e segurança no acesso aos arquivos armazenados remotamente devem ser comparáveis aos arquivos registrados em discos locais (COULOURIS *et al.*, 2013).

Os SADs são responsáveis pela organização, armazenamento, recuperação, atribuição de nomes, compartilhamento e proteção de arquivos. Os dados consistem em uma sequência de elementos ou arquivos (bytes), acessíveis pelas operações de leitura e escrita de qualquer parte desta sequência. Para WHITE (2012) seu funcionamento depende dos protocolos de rede utilizados, portanto todos os problemas relacionados a própria rede tornam-se inerentes a este tipo de abordagem, adicionando maior complexidade aos sistemas de arquivos distribuídos, como por exemplo em relação aos sistemas de arquivos convencionais.

Os Sistemas de Arquivos Distribuídos proporcionam: (i) acesso remoto aos arquivos armazenados em um servidor; (ii) acesso aos dispositivos de entrada e saída de outras máquinas; (iii) controle de versão e restauração de cópias de segurança; (iv) transparência de acesso e localização; (v) transparência de concorrência; (vi)

provêm: confiabilidade, redundância, disponibilidade, escalabilidade para armazenar os dados em forma de arquivos.

Como exemplos de SADs podemos citar: GFS (Global File System), GFS (Google File System), AFS (Andrew File System), CODA (Constant Data Availability), PVFS (Parallel Virtual File System); GlusterFS, SUN Network Filesystem, Lustre e HDFS (Hadoop Distributed File System).

Armazenamento em nuvens

Cloud computing ou computação na nuvem, é o termo usado para caracterizar os serviços de TI sob demanda com pagamento baseado no uso. O conceito em torno de computação em nuvem foca em permitir a seus usuários acessar recursos computacionais (por exemplo, servidores, armazenamento, redes, serviços e aplicações) de maneira prática e sob demanda, rapidamente e que podem ser liberados para o usuário sem qualquer envolvimento gerencial (SOUSA, MOREIRA, MACHADO, 2009; TAURION, 2009).

A computação nas nuvens parte do princípio de que a computação não é um produto, mas um serviço. Quer dizer, uma empresa não precisa possuir uma licença de software, um servidor ou uma plataforma de desenvolvimento. Basta ter acesso às funcionalidades e à infraestrutura desses softwares e hardwares de uma outra empresa que fornece estes serviços. Além disso, não apenas recursos de computação e armazenamento são entregues sob demanda, mas toda a pilha de computação pode ser aproveitada na nuvem. A escalabilidade é uma característica fundamental na computação em nuvem. As aplicações desenvolvidas para uma nuvem precisam ser escaláveis, de forma que os recursos utilizados possam ser ampliados ou reduzidos de acordo com a demanda. Podemos imaginar a computação em nuvens como uma enorme rede de nós que precisa ser escalável. Para os usuários a escalabilidade deve ser transparente, não necessitando eles saberem

onde estão armazenados os dados e de que forma eles serão acessados (SOUSA *et al.*, 2010; SOUSA,MOREIRA,MACHADO, 2009).

As nuvens podem ser caracterizadas em diversos modelos para a sua implantação (pública, privada, comunitária e híbrida) e diferentes modelos de serviços (IaaS - *Infrastructure as a Service*, PaaS - *Platform as a Service* e SaaS - *Software as a Service*, DBaaS – *Database as a Service*). Estes modelos de serviço oferecem diferentes tipos de serviços e possuem diferentes níveis de controle, flexibilidade e gerenciamento. Em linhas gerais, o que diferencia os tipos de serviços entre si é o tipo de cliente final ao qual cada um se destina, ou seja, quais tipos de tecnologia o cliente está contratando.

IaaS - Infraestrutura como Serviço: o modelo Infraestrutura como serviço é um formato de computação em nuvem que provê todos os recursos de infraestrutura de um *Data Center* local como serviço, disponível via Internet. Nesse modelo a empresa contrata uma capacidade de hardware que corresponde a memória, armazenamento e processamento. Oferece recursos como servidores, rede, armazenamento, firewalls e outras tecnologias de computação para construir um ambiente de aplicação sob demanda, que podem incluir sistemas operacionais e aplicativos. Em geral, quem administra o ambiente de infraestrutura na nuvem é quem oferece o serviço e não quem contrata, mas o controle sobre os sistemas operacionais, distribuição do armazenamento e aplicativos implantados fica a cargo do contratante. Alguns exemplos de provedores de IaaS são: Amazon Elastic Compute Cloud¹ (EC2), Citrix, Eucalyptus² e o Rackspace Cloud.

PaaS - Plataforma como Serviço: o modelo Plataforma como Serviço é um formato de computação em nuvem que provê a seus clientes um ambiente completo composto por todos os recursos necessários para o desenvolvimento de software em uma ou mais linguagens de programação, tais como compiladores, depuradores,

¹ <https://aws.amazon.com/pt/ec2/>

² <https://www.eucalyptus.cloud/>

bibliotecas e um sistema operacional. Deve-se ter em mente que o ambiente de desenvolvimento pode ter limitações quanto às linguagens de programação, gerenciadores de banco de dados, sistema operacional etc., conforme o fornecedor do serviço. Isto quer dizer que o ambiente ou plataforma não é genérico, mas sim uma plataforma completa para uma determinada finalidade. No modelo PaaS, o contratante não administra ou controla a infraestrutura subjacente, não precisa se preocupar com a configuração de infraestrutura necessária para que esta plataforma esteja em funcionamento, configurações como sistemas operacionais e servidores de aplicação; mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações de aplicações hospedadas nesta infraestrutura. São exemplos de provedores PaaS: o Microsoft Azure³, Aneka, Force.com, Google AppEngine, entre outros.

SaaS - Software como um Serviço: o modelo de Software como um Serviço, também conhecido como serviços de aplicativos em nuvem, proporciona sistemas de software com propósitos específicos, que são disponíveis para os usuários por meio da Internet e acessíveis a partir de vários dispositivos do usuário através de uma interface e um navegador Web. SaaS é a opção mais comumente utilizada pelas empresas. Esse modelo utiliza a Internet para entregar funcionalidades em forma de serviços aos usuários. Nesse modelo o usuário enxerga apenas o software que precisa usar e não tem conhecimento de onde realmente estão localizados os recursos empregados, nem quais linguagens de programação foram usadas no desenvolvimento do serviço, nem o sistema operacional e o hardware sobre o qual a aplicação executa. No SaaS, o usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistema operacional, armazenamento ou mesmo as características individuais da aplicação, exceto configurações específicas. Como exemplos de SaaS podemos destacar: Hotmail, Dropbox, SQL Azure, Facebook, Skype, Twitter, Gmail e o Google Docs.

³ <https://azure.microsoft.com/pt-br/>

DBaaS - Database como um Serviço: o modelo de Banco de Dados como um Serviço refere-se ao serviço em nuvem, onde é oferecido SGBDs em forma de uma plataforma flexível escalável, sob demanda, no modelo autosserviço de gerenciamento fácil. Permite aos usuários configurar, operar e dimensionar seus bancos de dados, sem ter que saber nem se preocupar com a infraestrutura para o banco de dados específico. Alguns serviços DBaaS, dentre outros, podem ser obtidos por Amazon RDB e MongoDB Atlas.

Capítulo 3. Modelagem de Dados

Modelagem é uma representação simples, normalmente gráfica, de estrutura de dados reais mais complexas, sendo um modelo de uma abstração de um objeto ou evento real de maior complexidade. Sua função é auxiliar na compreensão das complexidades do ambiente real. O ato de modelar é a atividade de criar representações que expressam objetos ou eventos do mundo real (CHEN, 1976, 1990; COUGO, 1997; HEUSER, 2008).

Um modelo é uma representação abstrata e simplificada de um sistema real, com a qual se pode explicar ou testar o seu comportamento, em seu todo ou em partes. Por exemplo, uma planta baixa, um manequim etc.

A modelagem de dados é o passo inicial do projeto na criação de um banco de dados, pois é nele que criamos um modelo de dados específico para um determinado domínio. O modelo de dados é uma representação formal dos dados que forma a estrutura de um banco de dados, descrevem os tipos de informações que estão armazenadas em um banco de dados. Estes modelos são ferramentas que permitem demonstrar como serão construídas as estruturas de dados que darão suporte aos processos de negócio, como esses dados estarão organizados e quais os relacionamentos que pretendemos estabelecer entre eles. Dada a complexidade dos sistemas do mundo real e os diferentes níveis de envolvidos com o projeto de banco de dados, o modelo de dados deve expressar a visão de mundo para estes diferentes níveis de usuário. Desta forma, é necessário construir uma abstração dos objetos e fenômenos do mundo real, de modo a obter uma forma de representação mais conveniente para cada envolvido. Neste sentido, consideramos três níveis de abstração (CHEN, 1990; COUGO, 1997; DAMA, 2017; ELMASRI, NAVATHE, 2005; HEUSER, 2008; SILBERSCHATZ, KORTH, SUDARSHAN, 2012):

- **Nível físico:** o nível mais baixo de abstração descreve como os dados estão realmente armazenados. No nível físico, complexas estruturas de dados de baixo nível são descritas em detalhes;

- **Nível conceitual ou lógico:** o próximo nível de abstração descreve quais dados estão armazenados de fato no banco de dados e as relações que existem entre eles. Aqui o banco de dados inteiro é descrito em termos de um pequeno número de estruturas relativamente simples. Embora a implementação de estruturas simples no nível conceitual possa envolver complexas estruturas de nível físico, o usuário do nível conceitual não precisa preocupar-se com isso. O nível conceitual de abstração é usado por administradores de banco de dados, que podem decidir quais informações devem ser mantidas no BD.
- **Nível de visão do usuário:** o mais alto nível de abstração descreve apenas parte do banco de dados que são direcionadas para entendimento dos usuários finais. O nível de abstração das visões de dados é definido para simplificar esta interação com o sistema, que pode fornecer muitas visões para o mesmo banco de dados.

Dados estes níveis de abstração, os modelos de dados também possuem diferentes níveis de detalhes aderentes aos diferentes níveis de abstração de representação de dados.

Modelo conceitual

Representa ou descreve a realidade do ambiente do problema, constituindo-se em uma visão global de negócio dos principais dados e relacionamentos, independente das limitações e especificações de implementação (tecnológicas ou paradigma de desenvolvimento). É uma descrição em alto nível, mas que tem a preocupação de capturar e retratar toda a realidade de uma organização, por isso é o modelo mais adequado para o envolvimento do usuário que não precisa ter conhecimentos técnicos.

Modelo entidade-relacionamento

O modelo entidade-relacionamento (MER) é um modelo de nível conceitual originalmente definido por Peter Chen em 1976, baseado na teoria relacional e teoria dos conjuntos, e passando a ser referência no processo de modelagem conceitual de dados. A construção deste modelo tem a finalidade de mostrar ao cliente os principais aspectos do banco de dados, assim como permitir uma interação mínima do usuário final com a tecnologia de banco de dados, e não se preocupando em representar como estes dados estarão realmente armazenados. Dessa forma, é possível a compreensão desse usuário de modo a garantir correção e respeito às regras de negócio por ele impostas. Este modelo possui basicamente três componentes principais: as entidades, os relacionamentos e os atributos (CHEN, 1990; COUGO, 1997; ELMASRI, NAVATHE, 2005; HEUSER, 2008; SILBERSCHATZ, KORTH, SUDARSHAN, 2012).

Entidade é um conjunto de objetos do mundo real de mesma natureza, com as mesmas características ou atributos, abrigados sob um nome genérico, sobre as quais há necessidade de manter informações no banco de dados. Estes objetos podem ser concretos (existem fisicamente), como por exemplo pessoas e carros, ou abstratos (existência conceitual), por exemplo uma compra ou um curso. As entidades se classificam como fortes ou fracas. As **entidades fortes** são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Por exemplo, em um sistema de vendas a entidade produto independe de quaisquer outras para existir. As **entidades fracas** são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido.

Os **atributos** são propriedades ou características que descrevem uma entidade. Por exemplo, um carro pode ser caracterizado por uma cor e uma marca, uma pessoa pode ser caracterizada pelo nome e data de nascimento. Os atributos podem ser classificados pela sua cardinalidade como **monovalorado** ou **multivalorado**,

ou seja, quantos atributos deste mesmo tipo uma entidade pode ter. Por exemplo, uma pessoa possui apenas uma data de nascimento, entretanto ela pode ter nenhum, um ou múltiplos telefones de contato.

Os atributos ainda podem ser considerados simples ou compostos, ou seja, os atributos simples são atômicos ou indivisíveis, por exemplo a idade de uma pessoa e os atributos compostos são aqueles que podem ser divididos em dois ou mais atributos, como por exemplo o endereço de uma pessoa que pode ser dividido em tipo do logradouro, nome do logradouro, número, complemento, bairro, código postal etc. Ainda podemos dizer que um atributo é derivado quando o valor deste depende do valor de um ou mais atributos. Por exemplo o atributo idade, ele é determinado pela data de nascimento e pela data corrente, ou seja, ele deriva da diferença entre a data que está sendo medida e a data de nascimento.

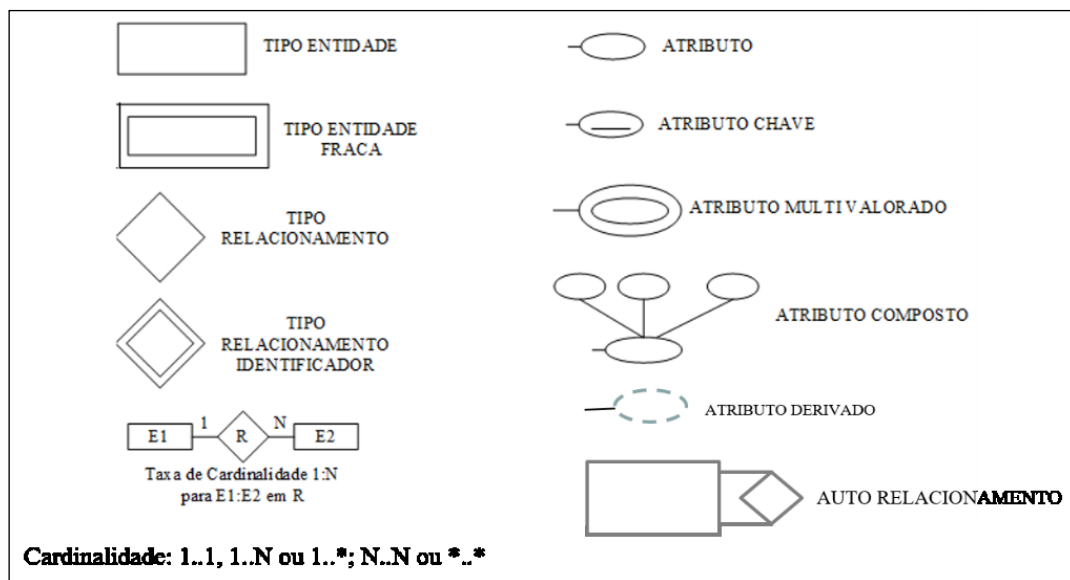
Os relacionamentos são as associações entre as entidades, ou seja, refere-se a como entidade se relaciona com outra entidade. Em geral representam ações que ocorrem entre duas ou mais entidades. Os relacionamentos se classificam conforme sua cardinalidade, de acordo com a lista abaixo:

- Relacionamento 1..1 (um para um): cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em uma base de currículos, cada usuário cadastrado pode possuir apenas um currículo, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.
- Relacionamento 1..n ou 1..* (um para muitos): uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada à uma unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal.

- Relacionamento n..n ou *.* (muitos para muitos): neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos.

O MER é graficamente representado pelo Diagrama de entidade-relacionamento (DER) conforme notação apresentada na Figura 5.

Figura 5 – Formas usadas na representação dos elementos do DER.



Fonte: ELMASRI e NAVATHE (2005).

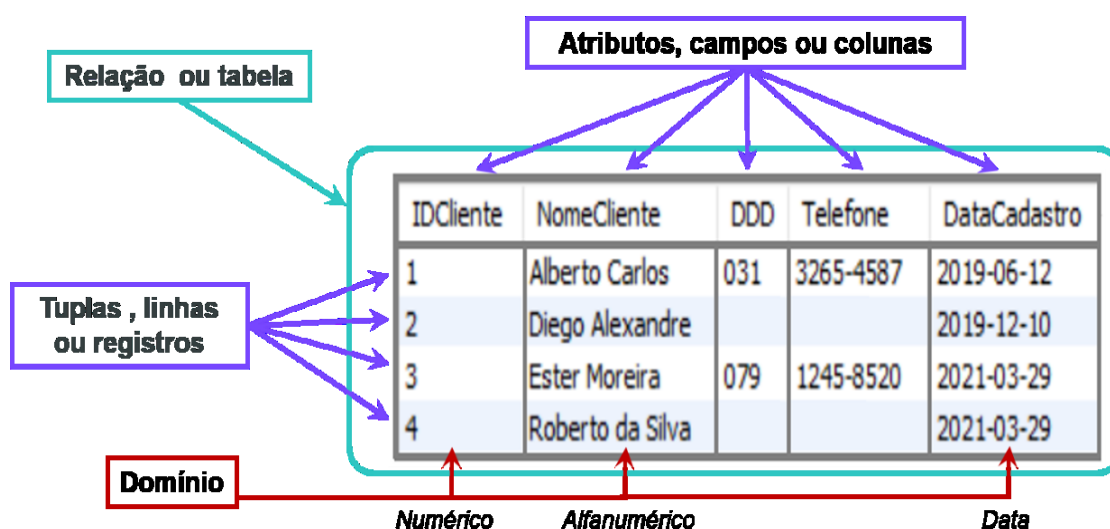
Modelo lógico

Um modelo derivado do modelo conceitual que leva em consideração algum paradigma tecnológico – paradigma hierárquico, em rede, relacional, orientado a objetos – mas não uma ferramenta em si. O modelo lógico descreve as estruturas que estarão contidas no banco de dados, mas sem considerar ainda nenhuma característica específica de SGBD, resultando em um esquema lógico de dados.

Modelo relacional

O modelo relacional é um modelo lógico que visa representar os dados necessários a um sistema, usando o paradigma relacional definido por Codd em 1970. Na sua estrutura fundamental Figura 6, o modelo relacional possui a **relação** (entidade no modelo conceitual e tabela no modelo físico) e as associações ou **relacionamentos** entre as relações. Uma relação é constituída por um ou mais **atributos** (campos ou colunas no modelo físico) que traduzem o tipo de dados a armazenar. Cada instância ou ocorrência do esquema de dados é chamada de **tupla** (registro ou linha no modelo físico). Um atributo possui ainda um **domínio** vinculado a ele, ou seja, um conjunto de valores atômicos que o atributo pode assumir (ELMASRI, NAVATHE, 2005; HEUSER, 2008; SILBERSCHATZ, KORTH, SUDARSHAN, 2012).

Figura 6 – Elementos de um modelo relacional.



Desta forma, em um modelo relacional, as estruturas do banco de dados relacional são vistas como um conjunto de relações ou tabelas, onde cada instância da relação é referenciada como uma tupla. Além disso, uma relação é caracterizada pelo seu conjunto de atributos que possuem um domínio específico. As relações se associam pelos relacionamentos. Além disso, no modelo relacional podem ser definidas restrições de integridade que ajudam a garantir a coerência e consistência

dos valores que os dados que serão armazenados (ELMASRI, NAVATHE, 2005). As regras de integridade são:

- *Integridade de Chave*: define que os valores da chave primária e alternativa devem ser únicos e não nulos.
- *Integridade de Domínio*: define os valores que podem ser assumidos pelos campos de uma coluna.
- *Integridade de Vazio*: especifica se os campos de uma coluna podem ou não serem vazios (nulos).
- *Integridade Referencial*: define que os valores dos campos que aparecem numa chave estrangeira devem aparecer na chave primária (candidata) da tabela referenciada.
- *Integridade de Unicidade*: define que o valor do campo ou campos são únicos.

O processo de modelagem relacional, quando não derivado de um modelo conceitual como o MER, envolve, antes de tudo, o entendimento sobre o negócio que será modelado. Isso pode ser pela compreensão dos requisitos do sistema, pela compreensão das regras de negócio, ou os dois no melhor dos casos.

Modelo de dados físico

O modelo de dados físico, como mencionado anteriormente, descreve, os dados e as estruturas que serão armazenadas do ponto de vista da tecnologia que foi escolhida para implementar o modelo de dados. É criado por meio de alguma linguagem, que descreve como será feita a armazenagem dos dados no SGBD. Neste nível de modelagem se escolhe qual SGBD será usado, levando em consideração o modelo lógico adotado. Por exemplo, para um modelo lógico relacional, pode ser

escolhido um SGBD relacional. Assim, para melhor compreender o modelo físico relacional, antes é necessário entender o que são os SGBD relacionais.

Sistemas gerenciadores de bancos de dados relacionais

Os bancos de dados relacionais são os mecanismos de persistência de dados mais adotado por empresas nas últimas décadas. É um SGBD que implementa o modelo relacional de dados. Fundamentado na teoria de conjuntos e nas possíveis relações entre os conjuntos, permitindo operações de junção, união, retorno seletivo de dados e diversas outras operações matemáticas. Até recentemente, este modelo foi considerado o mais adequado ao solucionar os vários problemas que se colocam no nível da concepção e implementação da base de dados para tratamento de dados estruturados (ELMASRI, NAVATHE, 2005; HEUSER, 2008).

Os bancos de dados relacionais são adequados para solucionar problemas que se colocam no nível da concepção e normalmente o uso mais comum deste tipo de banco é para implementar funcionalidades do tipo CRUD (do inglês Create, Read, Update e Delete), ou seja, criar ou inserir, ler ou selecionar, alterar e excluir um dado. As funcionalidades dos sistemas de informação fazem interface com os bancos de dados utilizando estas quatro operações básicas (CRUD) e geralmente por múltiplas aplicações em paralelo. Além disso, uma determinada funcionalidade pode envolver múltiplas operações, assim temos o conceito de transação em um SGBD. Uma transação é uma sequência de operações executadas como uma única unidade lógica de trabalho.

Propriedades ACID dos SGBDs relacionais

Para garantir que as transações sejam realizadas de forma que o banco de dados continue íntegro, os SGBDs relacionais implementam as propriedades conhecidas como ACID (Atomicidade, Consistência, Isolamento e Durabilidade) (ELMASRI, NAVATHE, 2005):

- **Atomicidade:** uma transação é composta por múltiplas operações, assim, a transação é uma unidade atômica de processamento. Quando realizada, ou se faz tudo, ou nada, sem meio termo. Todas as operações só serão aplicadas e persistidas se – e somente se – todo o conjunto de alterações for concluído com sucesso.
- **Consistência:** os dados estarão sempre consistentes e completos, respeitando as regras de integridade, relacionamentos e restrições configuradas no banco. Tem por objetivo garantir que o banco de dados antes da transação esteja consistente e, que após a transação, o banco permaneça consistente, sem problemas de integridade.
- **Isolamento:** a manipulação dos dados é realizada de forma isolada, garantindo que não haja interferência externa por outra transação, sendo realizada no mesmo instante. Desta forma, uma transação deve aguardar que a outra termine para poder acessar e manipular os dados.
- **Durabilidade:** uma vez que o banco de dados retornou a informação de que o dado está salvo, ele não será mais perdido.

Existem vários bancos de dados relacionais no mercado, entre os mais confiáveis e robustos podemos destacar o PostgreSQL, o IBM DB2, o MySQL, o Oracle e o SQL Server.

Modelo de dados relacional físico

Modelo derivado do modelo lógico relacional, que descreve as estruturas físicas de armazenamento de dados, tais como: tamanhos de campos, índices, tipos de dados, nomenclaturas, etc. Este modelo leva em consideração limites impostos pelo SGBD relacional escolhido e pelos requisitos não funcionais dos programas que acessam os dados (ELMASRI, NAVATHE, 2005).

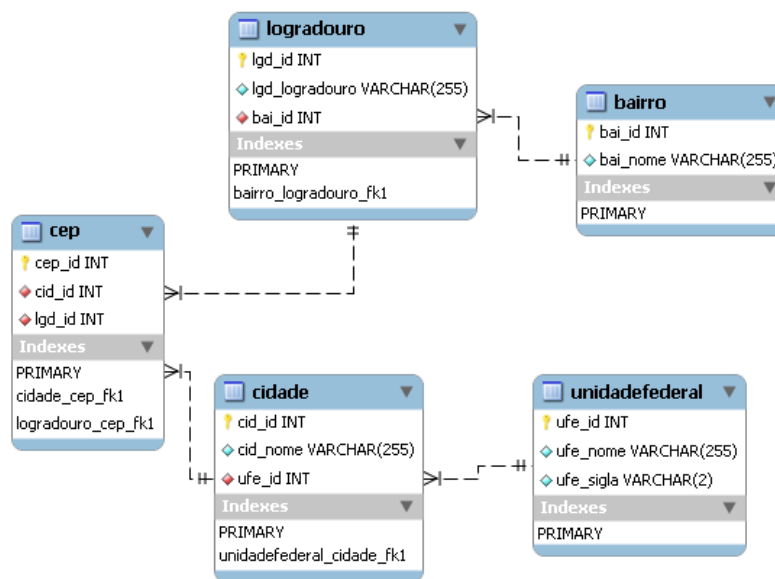
Por exemplo, no caso de escolher o SGBD Oracle, deve-se ter em mente que ele implementa tipos de dados diferentes do SQL Server. No Oracle temos um tipo

de dados chamado VARCHAR2 e no SQL Server o tipo de dados semelhante seria o VARCHAR. Outra diferença é que no SQL Server podemos criar uma coluna com a opção auto-incremento, já no Oracle não temos esta opção, mas temos um objeto chamado SEQUENCE para gerar dados sequenciais. É importante ter em mente que estas diferenças alteram o modelo físico pois são características específicas do SGBD escolhido.

Apresentamos na Figura 7 um exemplo de modelo relacional desenhado com o apoio da ferramenta MySQL Workbench⁴. Conforme a figura, observamos 5 relações ou tabelas chamadas *unidadefederal*, *cidade*, *cep*, *bairro*, *logradouro*. Cada uma delas possui uma chave primária que é identificada com o desenho de uma “chave” em frente ao nome do atributo identificado como chave. Por exemplo, na relação *unidadefederal* o atributo *ufe_id* é o atributo chave. Os demais atributos são precedidos por um losango azul. No caso dos losangos vermelhos, eles identificam que o atributo é uma chave estrangeira, por exemplo o atributo *ufe_id* na relação *cidade*.

Figura 7 – Exemplo de modelo relacional.

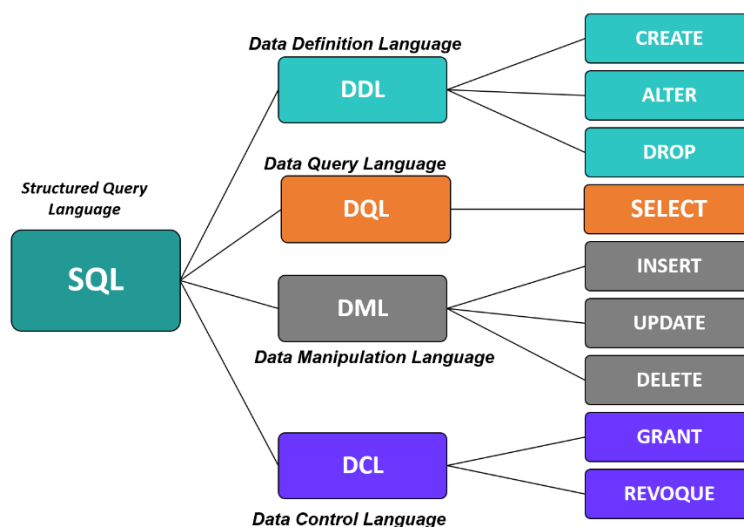
⁴ Disponível para download em: <http://fabforce.eu/dbdesigner4/>



Capítulo 4. Linguagem SQL

A Linguagem Estruturada de Consultas mais conhecida como linguagem SQL (do inglês *Structured Query Language*) surgiu no início dos anos 70 com o objetivo de fornecer uma interface mais amigável ao usuário para acesso aos bancos de dados. Foi um grande sucesso, sendo que a maioria dos SGBDs relacionais atuais a utilizam. A linguagem SQL é uma linguagem criada para interagir com os bancos de dados relacionais, pois, assim como os próprios bancos de dados, a linguagem é criada com o conceito de teoria de conjuntos. SQL serve para criar tanto as estruturas como os dados nos bancos de dados. Esta linguagem é dividida em sub-linguagens conforme ilustrado na Figura 8 a seguir, e também ilustra os principais comandos de cada uma.

Figura 8 – Divisões da linguagem SQL.



- Linguagem de Definição de Dados: DDL (Data Definition Language) é a parte da linguagem SQL utilizada para criação ou definição dos elementos ou estrutura do banco de dados, assim como a modificação e remoção das estruturas. Ou seja, permite criar, alterar e excluir as estruturas como tabelas, índices e procedimentos.

- Linguagem de Consulta de Dados: DQL (Data Query Language) é um conjunto de instruções usado para consultar dados nas estruturas dos objetos armazenados.
- Linguagem de Manipulação de Dados: também conhecida como DML (Data Manipulation Language), é o subconjunto da linguagem SQL, utilizada para realizar as operações de manipulação de dados, como inserir, atualizar e apagar dados.
- Linguagem de Controle de Dados: DCL (Data Control Language) é a parte da linguagem SQL que controla os aspectos destinados a autorização de acesso aos dados para manipulação de dados dentro do BD.

Capítulo 5. Data warehouse e modelagem dimensional

Um armazém de dados ou data warehouse (DW) é um esquema de banco de dados organizado em estruturas lógicas dimensionais, construído para atender sistemas de apoio à tomada de decisão, e possibilitando o seu processamento analítico por sistemas como os *Online Analytical Processing* (OLAP) e *Data Mining*. A definição original é dada por William H. Inmon (Bill Inmon) ao caracterizar os DW como um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais (BARBIERI, 2001, 2011).

- **Orientado a assuntos:** significa que os dados do DW dão informações sobre um assunto particular em vez de sobre operações contínuas da companhia. Por exemplo, um DW que lida com vendas de produtos a diferentes tipos de clientes, ou atendimentos e diagnósticos de pacientes.
- **Integrado:** os dados que são reunidos no DW partem de uma variedade de origens e assim podem apresentar diferentes nomenclaturas, formatos e estruturas das fontes de dados, estes dados precisam ser transformados em um único esquema ou formato para prover uma visão unificada e consistente da informação.
- **Não volátil:** os dados de um DW são estáveis, ou seja, não são modificados como em sistemas transacionais (exceto para correções), mas somente carregados e acessados para leituras, e em regra geral nunca removidos. Isso capacita à análise de séries históricas dos negócios.
- **Variável em relação ao tempo:** todos os dados no DW são identificados com um período de tempo particular, que é referente ao fato em análise.

Outro conceito relevante para entendermos sobre DW é o conceito de *data mart* ou “mercado de dados”. Segundo Inmon, um *data mart* corresponde às necessidades de informações de uma determinada comunidade de usuários (BARBIERI, 2011). Ou seja, *data mart* se refere a um subconjunto do DW que contém

dados sobre um setor específico da empresa (departamento, direção, serviço, gama de produto etc.), um assunto específico (compras, vendas, estoque, contábil etc), ou diferentes níveis de sumarização (Vendas Anual, Vendas Mensal).

Modelo dimensional de dados

A modelagem dimensional é uma técnica de modelagem, focada na elaboração do modelo lógico usado para projetos de DW e data marts. No modelo multidimensional, ou dimensional como às vezes é chamado, o foco não é a coleta dos dados e a normalização das estruturas de dados, mas sim a consulta aos dados. Esta é uma das grandes diferenças dos modelos dimensionais para os modelos relacionais. Ou seja, para melhorar o desempenho, há redundância planejada dos dados, compensando os gastos com armazenamento e atualização das informações. A redundância de dados aqui se torna uma vantagem competitiva (BARBIERI, 2001, 2011). O modelo dimensional permite visualizar dados abstratos de forma simples e relacionar informações de diferentes setores da empresa. Esta abordagem lida basicamente com três elementos: fatos, dimensões e métricas.

Os **fatos** são os assuntos que serão analisados, ou seja, acontecimentos do negócio que são merecedores de análise e controle na organização. A tabela de fatos é a principal tabela de um modelo dimensional, onde as métricas estão armazenadas. É composta por uma chave primária (formada por uma combinação única de valores de chaves de dimensão) e pelas métricas de interesse para o negócio. A tabela de fatos deve representar uma unidade do processo do negócio, e não devem misturar assuntos diferentes numa mesma tabela.

As **dimensões** são entidades do negócio que apresentam alguma influência sobre o fato em análise, assim, são variáveis de análise de um acontecimento. Estes dois elementos são representados no modelo como tabelas. A tabela de dimensão é composta de atributos e contém a descrição do negócio. Seus atributos são fontes das restrições de consultas, agrupamento dos resultados e cabeçalhos para

relatórios. Ela possui aspectos pelos quais se pretende observar as métricas relativas ao processo modelado. Imagine que está analisando uma compra (o fato) e será analisado conforme as dimensões Produto (o que foi comprado?), Filial (onde foi comprado?), Cliente (quem comprou?) e Tempo (quando comprou?).

As **métricas** são as medidas que são analisadas em um assunto de acordo com as variáveis de análise e são representadas no modelo como atributos da tabela fato. As métricas podem ser de diferentes tipos:

- **Aditiva:** permitem operações matemáticas como adição, subtração e média de valores independente das suas dimensões de análise. Por exemplo, a quantidade e valores de produtos. Podem ser sumarizados por data (dia, mês ou ano), local, clientes sem perder o sentido da informação.
- **Semiaditiva:** pode ser somada ou sumarizadas por parte das suas dimensões de análise. Por exemplo, saldo de conta bancária ou saldo de itens em estoque. O saldo é um valor que reflete a situação atual da conta, que pode ter o saldo credor ou devedor e pode variar por dia ao longo do mês. Assim, não faz sentido somar os saldos de todos os dias de um mês para uma determinada conta bancária.
- **Não aditiva:** não podem ser somadas e não podem ter agregações, pois perdem a veracidade do valor. Por exemplo, valores percentuais. Não faz sentido algum somar o percentual de vendas de um produto com o outro.

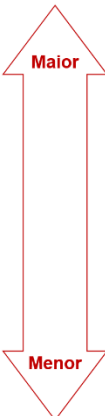
Outro fator importante a ser observado no modelo dimensional refere-se à granularidade do fato a ser analisado, ou seja, o nível de detalhe ou de resumo contido nas unidades de dados existentes no fato em análise. Um grão corresponde a um registro na tabela fato, e o detalhamento do fato afeta diretamente o volume de dados armazenado e consequentemente o tempo de resposta de uma consulta. Quanto mais detalhe existir, mais baixo será o nível de granularidade, consequentemente, quanto menos detalhe existir, mais alto será o nível de granularidade (BARBIERI, 2001, 2011).

Para exemplificar a ideia de granularidade, considere o exemplo da Figura 9, podemos analisar a criminalidade por tipo anualmente ou a cada semestre. Quando avaliamos por ano estamos avaliando os crimes com menos detalhes, ou seja, maior granularidade.

Figura 9 – Exemplo de granularidade.

Ocorrência	2017	2018	2019
Assaltos à mão armada	123	109	158
Furtos em residências	75	90	101
Furtos de veículos	243	250	332
Assassinatos	89	77	167
Estupros	2	24	69

Ocorrência	1º sem 2017	2º sem 2017	1º sem 2018	2º sem 2018	1º sem 2019	2º sem 2019
Assaltos à mão armada	50	73	45	64	70	88
Furtos em residências	40	35	60	30	71	30
Furtos de veículos	121	123	120	130	165	167
Assassinatos	40	49	37	40	67	100
Estupros	2	0	14	10	30	39



Além disso, no modelo multidimensional, os Fatos e dimensões podem ser dispostos segundo diferentes configurações: Esquema Estrela (*Star Schema*), o esquema original, onde as dimensões não são normalizadas. Ou Esquema Floco de Neve, (*Snow flake*), onde as dimensões são estruturadas de forma normalizadas, ou decompostas de ou mais dimensões que possuem hierarquias (BARBIERI, 2001, 2011).

Por exemplo, conforme Figura 10, a dimensão *lojas* no esquema estrela está desnormalizada, enquanto no modelo floco de neve ela foi normalizada dando origem às dimensões *estados* e *regiões*. O mesmo ocorre com a dimensão *produtos*, que após ser normalizada deu origem às dimensões *fabricantes* e *modelos*.

No esquema estrela a navegação pelas tabelas é mais simples, fácil e rápida, porém desperdiça espaço repetindo as mesmas descrições ao longo de toda a estrutura. O Esquema Floco de Neve reduz a redundância mas aumentam a complexidade do esquema e consequentemente a compreensão por parte dos usuários. Dificultam as implementações de ferramentas de visualização dos dados. A

decisão pela adoção destes dois esquemas deve considerar o tempo de resposta com o uso ou não da normalização e espaço em disco, pensando também em estratégias como uso de fatos agregados e alteração na granularidade dos dados.

Figura 10 – Exemplo de esquema estrela versus floco de neve⁵.



Para maiores detalhes sobre DW e modelagem dimensional, sugere-se a leitura do material disponível no Blog: <https://rafaelpiton.com.br/blog/>.

Extração, Transformação e Carga (ETC)

O processo conhecido como Extração, Transformação e Carga (ETC), mais conhecidos como ETL, sigla originária do inglês *Extract, Transform and Load*, tem sua origem com os projetos de DW, e envolve a coleta, a padronização e a consolidação de dados. ETC se referem às múltiplas rotinas construídas e executadas com a

⁵ Modelos retirados de:

<https://msdn.microsoft.com/pt-br/library/cc518031.aspx#XSLTsection126121120120>

finalidade de sistematizar a coleta ou extração dos dados originados dos diversos sistemas organizacionais, o tratamento (limpeza, transformação, integração, padronização) dos dados e a carga destes dados nas estruturas analíticas (DW).

Online Analytical Processing

Em processos de análises de dados é comum a necessidade de agrupar, agregar e juntar dados. Essas operações em bancos de dados relacionais consomem muitos recursos. Para otimizar estes cruzamentos de dados surgiu o *OnLine Analytical Processing* (OLAP), processamento online analítico, que se refere a um conjunto de técnicas de análise de dados desenvolvidas para analisar dados em data warehouses (BARBIERI, 2001, 2011).

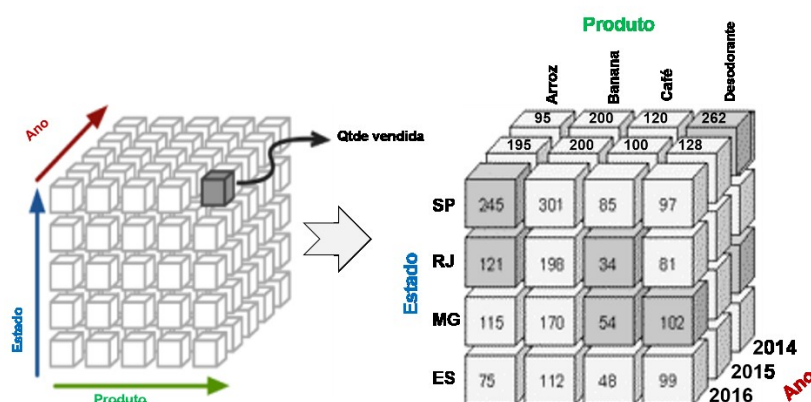
O processo de data warehousing é o processo de extrair dados de sistemas transacionais e transformá-los em informação organizada em um formato amigável. O conceito central do OLAP é a ideia de um cubo de dados. Baseia-se na ideia de analisar um acontecimento por múltiplas dimensões que surge a ideia de modelo multidimensional, que usa a metáfora do cubo (uma figura de três dimensões) para falar do conjunto de dados que será coletado (BARBIERI, 2001, 2011), conforme ilustrado na

Figura 11 a seguir.

Por exemplo, em um data mart de vendas, algumas das dimensões relevantes são o momento da venda (dia, mês e ano), local da venda, produto e vendedor. Na

Figura 11 consideramos apenas as três primeiras dimensões. Cada ponto em um cubo de dados armazena uma métrica consolidada dos valores de quantidade vendida. Em geral, essas dimensões são hierárquicas; o momento da venda pode ser organizado como uma hierarquia dia-mês-trimestre-ano, ou local poderia ser a loja, cidade, estado.

Figura 11 – Exemplo de cubo.



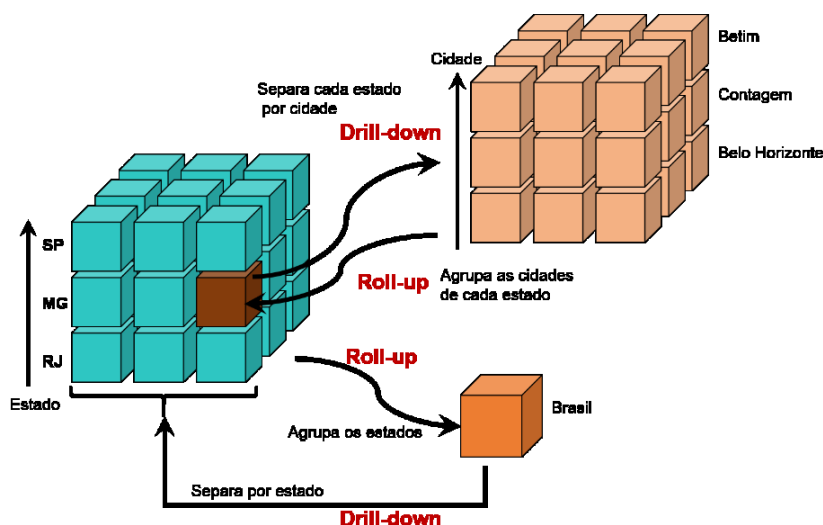
Na figura acima observamos que no ano de 2016, a Banana foi o produto mais vendido no estado de São Paulo, seguido pelo Arroz. Além disso, quando avaliamos a quantidade vendida de cada produto no estado de São Paulo em cada ano, observamos que os produtos Café e Desodorante tiveram uma queda nas vendas.

As operações OLAP típicas incluem drill down, roll-up, pivot e slice and dice, que são as formas de interagir com o cubo de dados para análise de dados multidimensionais. A operação roll-up também é conhecida como “consolidação” ou “agregação”, na operação drill-down é o oposto do processo de roll-up, ou seja, os dados são fragmentados em partes menores. Estas operações podem ser realizadas ao longo de cada dimensão e você pode “subir ou descer” dentro do detalhamento do dado, como, por exemplo, analisar uma informação tanto diariamente quanto anualmente, partindo da mesma base de dados. Ou seja, você pode navegar no cubo por diferentes níveis de granularidade. A operação de roll-up (

Figura 12) não é limitado pelo grão máximo e os dados podem ser agregados mesmo após se chegar a este limite superior. Já na operação de drill-down (

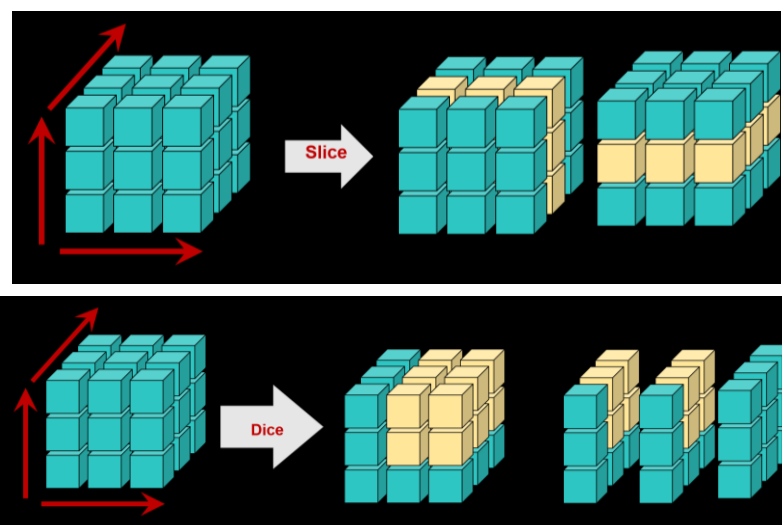
Figura 12) é limitado pelo grão mínimo e navega no sentido de explorar maior granularidade do cubo.

Figura 12 – Operações OLAP: roll-up e drill-down.



Nas operações slice and dice (Figura 13), também conhecidas como seleção e projeção, a ideia é gerar um sub-cubo limitando a análise a uma parte dos dados. No caso do slice, a análise é de apenas uma fatia, ou seja, restringe os valores de uma dimensão aplicando um filtro de seleção de dados, mas não diminui a cardinalidade do cubo. Já na operação dice, ocorre a redução das dimensões ou a cardinalidade de um cubo por meio da eliminação ou filtro em uma ou mais dimensões. Em suma, estas operações consistem na rotação do cubo, possibilitando a combinação de quaisquer dimensões.

Figura 13 – Operações OLAP: slice.



Na operação pivot, você gira os eixos de dados para fornecer uma apresentação de dados substituta, oferecendo assim diferentes perspectivas sobre o mesmo conjunto de dados. Por exemplo, ao invés de ver os dados na dimensão Produto x Estado é possível trocar para Estado x Produto.

Os dados de um DW promoverem a capacidade de analisar dados estruturados, padronizados e centralizados, fomentando os processos de gerência e tomada de decisão. Entretanto, o Big Data trouxe novos desafios que os DW's não conseguem atender em sua plenitude, principalmente pela diversidade de estruturas e formatos de dados, para suprir tal demanda, surge o *Data lake*.

Data lake, data swamp e data pond

Data lake ou lago de dados, termo cunhado por James Dixon⁶ ex-diretor de tecnologia da Pentaho para contrastar com a ideia de DW mas focando nas características do Big Data. Como o DW, o *Data lake* é um repositório de dados centralizado, geralmente com grandes quantidades de dados, mas contrastando com o DW possuem grande variedade de dados, incluindo dados estruturados, semiestruturados e não estruturados no mesmo repositório.

O objetivo do *Data lake* é centralizar os dados coletados das diversas fontes de dados, como planilhas, ERPs, CRMs, redes sociais etc., e acomodá-los em um único repositório. Se caracteriza por ser um modelo não estruturado e abrangente onde os dados são armazenados no seu estado bruto e estão disponíveis para qualquer pessoa que precise realizar uma análise. Assim, uma vez que os dados se encontram em repositório único, eles podem ser submetidos a diferentes processos de análise (CANALTECH, 2015).

⁶ Disponível em: <https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/>

Figura 14 – Esquema de funcionamento do *Data lake*.



Fonte: Traduzido de CANALTECH (2015)

Os dados inseridos no *Data lake* são atrelados a metadados (tags) para permitir a identificação, localização e utilização destes dados nos diferentes processos de análises. É o que podemos chamar de “governança” de lagos de dados. Quando um lago de dados é criado sem qualquer governança, ou seja, sem metadados que irão permitir por exemplo a localização de um dado, dizemos que não se trata mais de um lago e sim um pântano de dados (*data swamp*). Assim, *Data Swamp*, em contraste ao *Data lake*, apresenta pouca ou nenhuma organização ou nenhuma estrutura de organização. Os *Data Swamps* não têm curadoria, incluindo pouco ou nenhum gerenciamento ativo em todo o ciclo de vida dos dados e pouco ou nenhum metadado contextual e governança.

Data pond é outro conceito relacionado ao *data lake*. Refere-se a um subconjunto de dados do *data lake* organizados em uma estrutura que possa ser analisada. Quando os dados em seu estado bruto e formato nativo são carregados no *data lake*, uma vez processados nos lagos, estes dados podem então ser refinados e distribuídos para os *data ponds* (lagoas ou tanques de dados).

Capítulo 6. Bancos de Dados NOSQL

Durante a última década, a necessidade de lidar com grandes volumes de dados demandou novas tecnologias de armazenamento que em muitos casos sacrificam propriedades como consistência de dados para garantir baixo tempo de resposta em consultas. Neste contexto, surgiu uma nova categoria de SGBD não relacional visando atender aos requisitos de gerenciamento de dados, semiestruturados ou não estruturados, garantindo alta disponibilidade e escalabilidade, são os SGBDs NoSQL e NewSQL.

Sistemas gerenciadores de bancos de dados NoSQL.

O acrônimo **NoSQL**, significa *Not Only SQL* (não somente SQL, em tradução livre), em uma alusão a ideia de um movimento que prega que nem todos os cenários de dados são adequados ao uso de SGBDs relacionais, ou mais especificamente ao uso da linguagem SQL para consulta e manipulação de dados.

Os SGBDs NoSQL são considerados uma boa opção para aplicações fundamentadas no Big Data, uma vez que fornecem recursos eficientes para armazenamento de dados estruturados e não-estruturados, facilidade de acesso, alta escalabilidade e disponibilidade, e baixo custo. Quando comparados aos SGBDs relacionais, as tecnologias NoSQL geralmente usam interfaces de consulta de baixo nível e não padronizadas, o que torna mais difícil a integração em aplicativos existentes que esperam uma interface SQL (FOWLER, 2016; STROHBACH *et al.*, 2016).

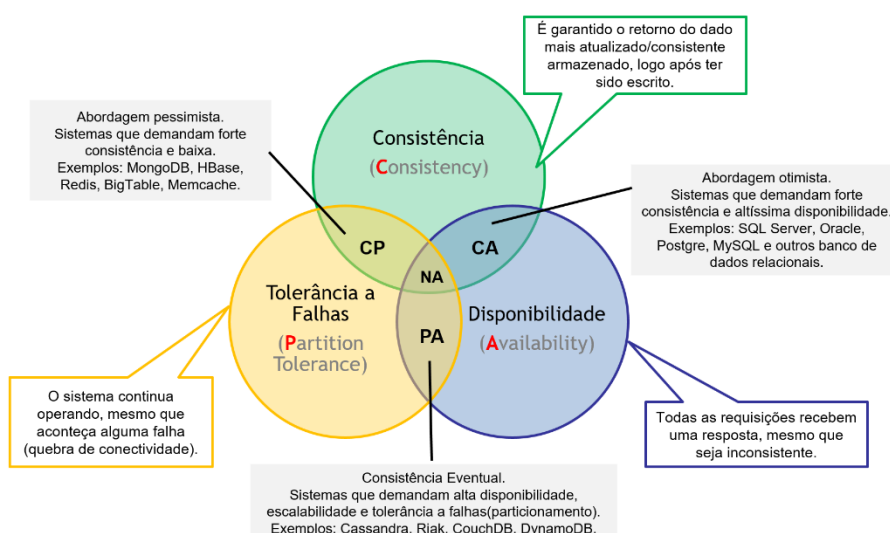
Teorema CAP

Os SGBDs NoSQL se fundamentam no Teorema CAP ou Teorema de Brewer, que descreve o comportamento de um sistema distribuído quando acontece uma requisição de escrita de dados seguida de uma requisição de leitura (consulta).

Conforme a Figura 15, dado um par de requisições, uma escrita seguida por uma leitura é impossível que o armazenamento de dados distribuído garanta simultaneamente mais de duas das três seguintes características: Consistência, Disponibilidade e Tolerância a falhas (STROHBACH *et al.*, 2016).

Em outras palavras, o teorema CAP afirma que, na presença de uma partição da rede, é preciso escolher entre consistência e disponibilidade. Observe que a consistência conforme definida no teor de CAP é bastante diferente da consistência garantida em transações de bases de dados ACID. Nenhum sistema distribuído está protegido contra falhas de rede, portanto a partição geralmente deve ser tolerada. Na presença de partições, são dadas duas opções: consistência ou disponibilidade. Ao escolher consistência em relação à disponibilidade, o sistema retornará um erro ou um tempo limite se não puder garantir que informações específicas sejam atualizadas devido à sua partição na rede. Ao escolher disponibilidade sobre consistência, o sistema sempre processará a consulta e tentará retornar a versão disponível mais recente da informação, mesmo que não possa garantir que ela esteja atualizada devido às partições.

Figura 15 – Esquema explicativo do teorema CAP.



Propriedades BASE

Os SGBDs NoSQL não aderem necessariamente às propriedades transacionais ACID, promovem as propriedades conhecidas como BASE (*Basically Available, Soft state, Eventual consistency*), que distribui os dados em diferentes repositórios tornando-os sempre disponíveis, não se preocupa com a consistência de uma transação, delegando essa função para a aplicação, porém sempre garante a consistência dos dados em algum momento futuro à transação. São projetados para garantir escalabilidade e disponibilidade, sacrificando a consistência dos dados (SADALAGE, FOWLER, 2013; STROHBACH *et al.*, 2016).

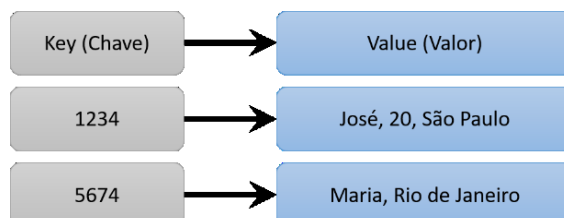
- Basically Available (Basicamente Disponível): o BD é desenhado para estar sempre disponível e respondendo principalmente às operações de escrita, mesmo que nem todos os seus dados estejam disponíveis para consulta naquele momento. Esta característica garante que os dados sejam recebidos pela aplicação sejam escritos e que sempre haja uma resposta válida para uma consulta.
- Soft state (em um estado flexível): uma informação armazenada em um BD pode ter uma relevância temporal ou em relação ao seu acesso. Isso faz com que uma determinada informação seja alterada ou descartada pelo próprio sistema, não garantindo que ela esteja salva da mesma forma como foi inserida pelo usuário. A localização atual de um usuário pode ser descartada ou ter o status alterado em função do tempo em que ela foi recebida.
- Eventual consistency (Eventualmente Consistente): haverá um momento em que todos os dados estarão consistentes, porém haverá um momento em que parte da informação pode estar ausente ou desatualizada em relação ao todo. Isso pode ocorrer em função de um atraso na atualização dos dados ou pela queda momentânea de um dos nós da aplicação de banco de dados.

Categorias de Bancos de Dados NoSQL

Devido aos diferentes cenários nos quais os SGBDs NoSQL são necessários, diferentes soluções e abordagens foram criadas e disponibilizadas para o mercado. Os SGBDs NoSQL podem ser distinguidos pelos modelos de dados que eles usam (BDW, 2014):

Armazenamento por chave-valor (key-value store): é o modelo mais simples tanto em termos de funcionamento quanto de entendimento. Permite o armazenamento de dados sem esquema definido. Os dados podem ser não-estruturados ou estruturados e são acessados por uma única chave. Nesta categoria, um determinado dado ou valor é acessado através de uma chave identificadora única (Figura 16). Estas chaves podem ser tratadas de forma literal ou armazenadas como hash's por parte dos BDs, e os valores tipicamente são tratados como sequenciais binários, não sendo interpretados ou trados pelo banco de dados.

Figura 16 – Exemplo esquema NoSQL key-value.



Existem vários SGBDs que implementam o paradigma Chave-Valor (Key-Value), dentre eles citamos: DynamoDB⁷, Redis⁸, Riak⁹, Voldemort¹⁰ e Memcached¹¹.

Armazenamento em coluna ou colunar (columnar stores): o modelo colunar define a estrutura de valores como um conjunto predefinido de colunas, ou seja, permite o armazenamento de tabelas de dados como seções de colunas de

⁷ <https://aws.amazon.com/pt/dynamodb/>

⁸ <http://redis.io/>

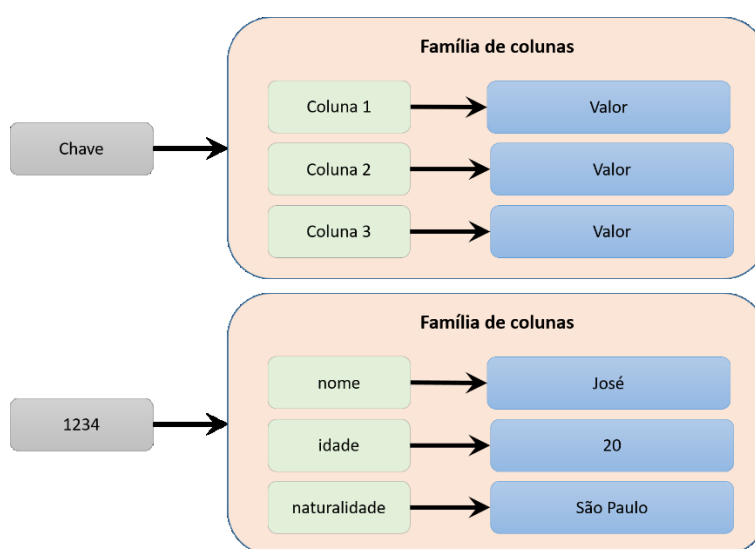
⁹ <http://basho.com/>

¹⁰ <https://www.project-voldemort.com/voldemort/>

¹¹ <https://memcached.org/>

dados e não como linhas de dados, como a maioria dos SGBD relacionais. Suportam várias linhas e colunas e também permitem subcolunas, assim, ao invés de definir antecipadamente as colunas necessárias para armazenar um registro, o responsável pela modelagem de dados define o que é chamado de “famílias de colunas” (Figura 17). As famílias de colunas são organizadas em grupos de itens de dados que são frequentemente usados em conjunto em uma aplicação.

Figura 17 – Exemplo esquema NoSQL colunar.



Nos SGBDs colunares não é necessário que cada linha de dados possua o mesmo número de colunas, dando maior flexibilidade de inserir as colunas que considerar necessárias em cada registro armazenado, sem precisar alterar a estrutura já existente. Alguns SGBDs colunar, são: BigTable do Google¹², Cassandra¹³ e HBase¹⁴.

Armazenamento orientado a Documentos (document databases): este modelo armazena coleções de documentos estruturados, mas sem a exigência de requisito para um esquema comum que todos os documentos devem aderir. Os

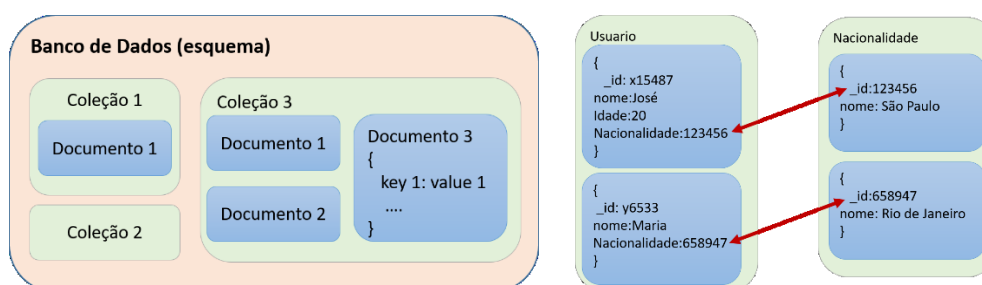
¹² <https://cloud.google.com/bigtable/>

¹³ <http://cassandra.apache.org/>

¹⁴ <https://hbase.apache.org/>

documentos são conjuntos de atributos e valores (semelhante ao esquema chave-valor), onde um atributo pode ser multivalorado. As chaves dentro dos documentos são únicas. Cada documento contém um identificador, que é único dentro do conjunto (Figura 18). Uma característica deste modelo é a independência de um esquema rígido predefinido, permitindo a atualização na estrutura do documento sem causar problemas ao banco de dados. Além disso, dois documentos de uma mesma coleção podem possuir um conjunto de chave-valores distinto.

Figura 18 – Exemplo de esquema NoSQL orientado a documentos.



Cada documento possui uma chave única de identificação que pode ser usada como referência de relação entre documentos distintos (Figura 18). Esse modelo costuma armazenar os valores em uma estrutura como JSON (*JavaScript Object Notation*) ou XML (*Extensible Markup Language*). Dentre os diversos SGBDs orientado a documentos, podemos citar: Couchbase¹⁵, CouchDB¹⁶, MongoDB¹⁷ e BigCouch¹⁸.

Armazenamento orientados por grafos: os SGBDs orientados por grafos armazenam os dados em estruturas definidas conforme a teoria dos grafos. Possuem três componentes básicos: os nós (são os vértices do grafo) para armazenar os dados dos itens coletados, as arestas que armazenam os relacionamentos entre os dados

¹⁵ <http://www.couchbase.com/>

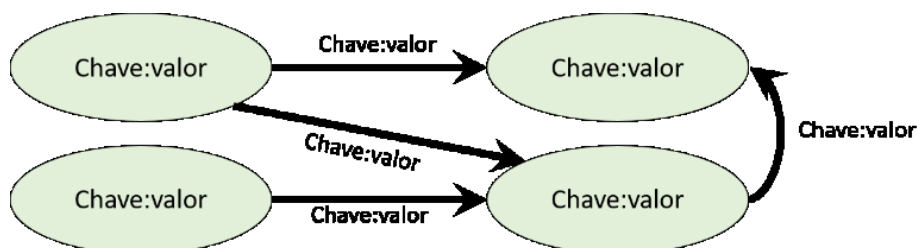
¹⁶ <http://couchdb.apache.org/>

¹⁷ <https://www.mongodb.com>

¹⁸ <https://bigcouch.cloudant.com/>

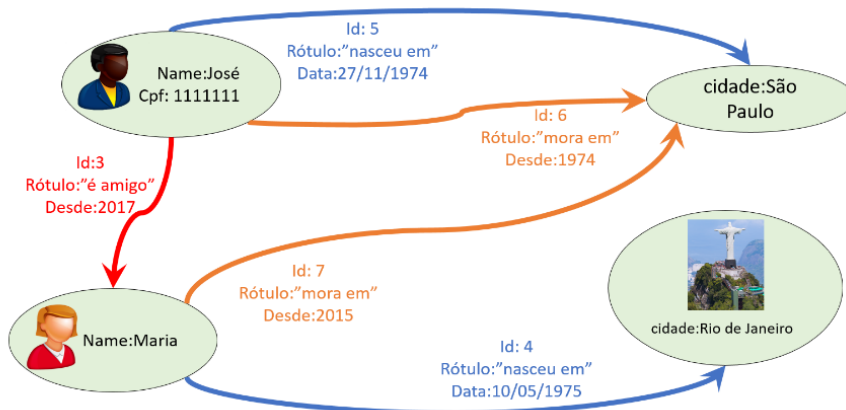
(vértices) e as propriedades (ou atributos) dos nós e relacionamentos. As propriedades são definidas conforme o par chave-valor (Figura 19).

Figura 19 – Exemplo esquema NoSQL orientado por grafos.



Os vértices e arestas podem ter múltiplas propriedades (Figura 20). Um conjunto de vértices conectados por meio de arestas definem um caminho no grafo. Este modelo suporta a utilização de restrições de integridade de dados, garantindo assim as relações entre elementos de forma consistente.

Figura 20 – Exemplo de dados organizados em um esquema NoSQL orientado por grafos.



Estes BDs são ideais para gerenciar relações entre diferentes objetos ou que se encontram em tipos de dados diferentes, ou quando a interconectividade dos dados é importante como a web semântica. Dentre os SGBDs orientado por grafos temos: Virtuoso¹⁹, Infinitegraph²⁰, AllegroGraph²¹, Titan²², ArangoDB²³ e Neo4J²⁴.

¹⁹ <https://virtuoso.openlinksw.com/>

²⁰ <https://www.objectivity.com/products/infinitegraph/>

²¹ <http://franz.com/agraph/allegrograph/>

²² <https://titan.thinkaurelius.com/>

²³ <https://www.arangodb.com/>

²⁴ <https://neo4j.com/>

Capítulo 7. Fundamentos de Análise de dados

Análise de dados nada mais é do que um conjunto de técnicas empregadas para a transformação de dados e informações em conhecimento para um propósito específico. A análise de dados visa encontrar uma forma eficiente de conhecimento (padrões) em conjuntos de dados, seja para compreender o comportamento de pessoas ou para identificar novas oportunidades de negócio.

Inicialmente, com o surgimento dos DWs, eram empregadas técnicas de análise a fim de compreender o que aconteceu e o motivo pelo qual os eventos aconteceram. Mais tarde isso já não era suficiente, pois surgiu a necessidade de tentar prever o que poderia acontecer com os negócios antes de acontecer e, assim, antecipar algumas ações. Como o volume de dados ultrapassava a capacidade humana de interpretar e compreender tanta informação, foi necessário criar mecanismos automáticos para processar tantos dados.

Principais tipos de análise de dados

De acordo com a classificação do Gartner, existe uma cadeia de evolução em análise de dados, variando de descritiva a diagnóstica, até preditiva, e culminando com prescritiva. Esses tipos de análise de dados ajudam as organizações a compreender dois momentos sobre seus negócios: passado e futuro.

Análise descritiva

Análise descritiva de dados, às vezes descrita como análise exploratória, tem como objetivo é entender o cenário atual da organização a partir da análise de seus dados históricos. Trabalha com histórico de dados, cruzando informações com o objetivo de gerar um panorama claro e preciso dos temas relevantes para a empresa no presente momento a partir de seu passado. Em geral utilizam métricas e técnicas estatísticas simples ou avançadas para entender e explicar como os dados são,

buscando explicar o que está acontecendo ou aconteceu em uma determinada situação (TUKEY, 1977).

Análise diagnóstica

Algumas referências incluem a análise diagnóstica como parte da descritiva, isso porque a análise diagnóstica visa explicar os eventos que ocorreram e foram descritos no modelo de análise anterior. Este modelo de análise tentar responder à pergunta “*Por que isso aconteceu?*”. Neste modelo de análise o foco está na relação de causas e consequências percebidas ao longo do tempo, sobre de um determinado assunto ou evento, cruzando informações com o objetivo de entender quais fatores influenciaram o resultado atual.

Análise preditiva

A análise preditiva é utilizada para prever tendências baseadas nos dados. Segundo o Gartner, a análise preditiva é uma forma de análise avançada que examina dados ou conteúdo para responder à pergunta: “*O que vai acontecer?*”, ou mais precisamente, “*O que é provável que aconteça?*”. Este tipo de análise é o mais indicado para quem precisa prever algum tipo de comportamento ou resultado. Essa técnica busca analisar dados relevantes ao longo do tempo, buscando padrões comportamentais e suas variações de acordo com cada contexto, a fim de prever como será o comportamento de seu público ou mercado no futuro, dadas as condições atuais. Muito útil para avaliar tendências de consumo e flutuações econômicas. É caracterizada por técnicas como análise de regressão, previsão, estatísticas multivariadas, correspondência de padrões, modelagem preditiva e previsão.

Análise prescritiva

A análise prescritiva vai um pouco além da preditiva, porém a lógica envolvida é semelhante. Esta forma de análise examina dados ou conteúdo para responder à pergunta: “*O que deve ser feito?*”, ou “*O que podemos fazer para fazer algo acontecer?*”. Um pouco mais profunda que a análise preditiva, a análise prescritiva traduz as previsões em planos viáveis para o negócio. Ou seja, foca em prever as possíveis consequências para as diferentes escolhas que forem feitas, e desta forma, este tipo de análise pode recomendar melhores caminhos a serem seguidos. E é caracterizada por técnicas como análise de gráficos, simulação, processamento de eventos, redes neurais, mecanismos de recomendação, heurística e aprendizado de máquina.

Análise Exploratória de Dados

Após a coleta e a digitação de dados em um banco de dados apropriado, o próximo passo é a análise descritiva ou análise exploratória de dados. Seu intuito é observar os dados previamente à aplicação de qualquer técnica estatística (DATA SCIENCE GUIDE, 2021). Assim, o analista consegue um entendimento básico de seus dados e das relações existentes entre as variáveis analisadas. Em suma, a AED consiste em sumarizar e organizar os dados coletados por meio de tabelas, gráficos ou medidas numéricas, e a partir desta sumarização/organização procura por alguma regularidade ou padrão nas observações, ou seja, faz a interpretação dos dados²⁵(HECKERT,FILLIBEN, 2003; TUKEY, 1977).

²⁵ Para detalhes, acesse:

http://leg.ufpr.br/~fernandomayer/aulas/ce001n-2016-01/02_Analise_Exploratoria_de_Dados.html

População e amostra

Entende-se como população o conjunto dos elementos que representam pelo menos uma característica comum, no qual deseja-se analisar o comportamento de interesse. Ou seja, a população é o conjunto global sobre o qual se deseja chegar a conclusões. A amostra refere-se ao subconjunto finito de uma população sobre o qual são feitas observações. A amostra é qualquer conjunto de elementos retirado da população, que não seja vazio e tenha um menor número de elementos que a população. Uma amostra tem que ser representativa em relação à população para que os resultados não sejam distorcidos (MEDRI, 2011).

Variável

Entende-se como variável qualquer característica de interesse associada aos elementos de uma população. Estas variáveis podem ser (MEDRI, 2011):

Variáveis qualitativas: variáveis que assumem valores categóricos, classes ou rótulos, ou seja, por natureza, dados não numéricos. Estas variáveis denotam características individuais das unidades sob análise, tais como sexo, estado civil, naturalidade, raça, grau de instrução, dentre outras; permitindo estratificar as unidades para serem analisadas de acordo com outras variáveis. Podem ser subdivididas como nominal, onde as categorias não possuem uma ordem natural (Ex. nomes, cores, sexo) ou como ordinal, onde as categorias podem ser ordenadas (Ex. classe social, grau de instrução, estado civil) (MEDRI, 2011).

Variáveis quantitativas: variáveis que assumem valores numéricos, intervalar ou de razão, por exemplo: idade, salário, peso etc. As variáveis quantitativas podem ser classificadas como discretas, quando assumem um número finito de valores, em geral valores inteiros (Ex.: número de irmãos, número de passageiros), ou contínuas, quando assumem um número infinito de valores dos números reais, geralmente em intervalos (Ex.: peso, altura, pressão) (MEDRI, 2011).

Medidas

A análise exploratória de dados consiste em um conjunto de cálculos de medidas estatísticas que visam resumir as características dos dados. Dentre as medidas estatísticas as mais utilizadas são as medidas de posição central (de tendência central), medida de posição, medida de dispersão e medidas de assimetria (MEDRI, 2011).

Medidas de Posição Central: representam os fenômenos pelos seus valores médios, em torno dos quais tendem a concentrar-se os dados. Dentre todas as medidas de tendência central, temos: Média, mediana e moda.

- *Moda:* é o valor (ou atributo) que ocorre com maior frequência.
- *Média:* soma de todos os valores da variável dividida pelo número de observações.
- *Mediana:* valor que deixa 50% das observações à sua esquerda

Medida de Posição: são medidas que dividem a área de uma distribuição de frequências em regiões de áreas iguais. As principais medidas de posição são: Quartil, Percentil, Mínimo e Máximo.

- *Máximo (max) e Mínimo (min):* a maior e a menor observação de valor dos dados.
- *Quartis:* divide um conjunto de valores dispostos em forma crescente em quatro partes. Primeiro Quartil (Q1): valor que deixa 25% das observações à sua esquerda. Terceiro Quartil (Q3): valor que deixa 75% das observações à sua esquerda.

Medidas de Dispersão: é um valor que busca quantificar o quanto os valores da amostra estão afastados ou dispersos relativos à média amostral. A dispersão é a variabilidade que os dados apresentam entre si, ou seja, se todos os valores forem

iguais, não existe dispersão; agora, se os dados não são iguais, existe dispersão entre os dados. As medidas utilizadas para representar dispersão são:

- *Amplitude*: diferença entre o valor máximo e o valor mínimo.
- *Intervalo-Interquartil*: é a diferença entre o terceiro quartil e o primeiro quartil, ou seja, $Q3 - Q1$.
- *Variância*: média dos quadrados dos desvios em relação à média aritmética.
- *Desvio Padrão*: mede a variabilidade independente do número de observações e com a mesma unidade de medida da média.
- *Coefficiente de Variação*: mede a variabilidade em uma escala percentual independente da unidade de medida ou da ordem de grandeza da variável.

Medidas de Assimetria e Curtose: as medidas de assimetria possibilitam analisar uma distribuição de acordo com as relações entre suas medidas de moda, média e mediana, quando observadas graficamente ou analisando apenas os valores. Ou seja, uma distribuição é considerada simétrica quando apresenta o mesmo valor para a moda, a média e a mediana. Da mesma forma, é considerada assimétrica quando essa igualdade de medidas não ocorre. Curtose é o grau de achatamento da distribuição em relação a uma distribuição padrão, denominada de curva normal. Ou seja, o quanto uma curva de frequência será achatada em relação a uma curva normal de referência.

Web mining

Web Mining (Mineração na Web) corresponde à aplicação de técnicas de Data Mining (Mineração de Dados) à Web. Ou seja, Web Mining é o processo de extração de conhecimento, a partir dos dados da Web. São três tipos de Web Mining: 1) Mineração de conteúdo da Web (*Web Content Mining*): extrai informação do

conteúdo dos recursos Web. 2) Mineração da estrutura da Web (*Web Structure Mining*): tem como objetivo principal extrair relacionamentos, previamente desconhecidos, entre recursos web. 3) Mineração de uso da Web (*Web Usage Mining*): utiliza técnicas de mineração de dados para encontrar analisar ou descobrir padrões de navegação do usuário nos sites. O objetivo é melhorar a experiência do usuário nas aplicações Web (SCIME, 2005).

Text mining

A Mineração de Texto (*Text Mining*), consiste na aplicação de técnicas de mineração de dados para obtenção de informações importantes em um texto. É um processo que utiliza algoritmos capazes de analisar coleções de documentos texto escritos em linguagem natural, com o objetivo de extrair conhecimento e identificar padrões. Dentre as técnicas utilizadas, destaca-se o processamento de linguagem natural (SILVA, 2002; TAN, 1999).

Capítulo 8. Coleta e preparação de dados

Coleta e Preparação dos dados são duas das atividades principais do Engenheiro de Dados. Os dados podem ser coletados de várias fontes de dados diretas e indiretas, internas e externas à organização. A preparação de dados refere-se ao conjunto de atividades realizadas para melhorar a qualidade do dado ou transformar os dados brutos em um formato plausível para ser usado e analisado (REHMAN *et al.*, 2016).

Coleta de dados

A coleta de dados precede as demais atividades de análise de dados. Pois tudo começa com a obtenção dos dados. A coleta de dados nada mais é que o processo de obtenção de dados de uma ou mais fontes de dados (REHMAN *et al.*, 2016). Para realizar a coleta de dados é necessário identificar as fontes de dados e os respectivos tipos de dados que cada fonte provê.

As fontes podem ser internas, os dados disponíveis na organização tais como dados oriundos dos CRMs, ERPs, SCMs entre outros sistemas de processamento de transação usados pela empresa; ou externas, como os dados disponíveis na WEB, dados adquiridos junto a empresas especializadas e dados abertos disponibilizados, por exemplo, por órgãos governamentais.

A coleta de dados nas bases internas pode ser realizada utilizando a linguagem SQL e aplicações desenvolvidas em ferramentas de ETL como o Pentaho Data Integration²⁶, a Plataforma Knime Analytics ou em linguagens como Java e Python.

²⁶ Para detalhes acesse https://help.pentaho.com/Documentation/7.1/0D0/Pentaho_Data_Integration ou <https://www.infoq.com/br/articles/pentaho-pdi/>

A coleta de fontes externas que merece destaque é a Web, neste caso, não apenas os dados abertos, mas também as mídias sociais em geral. Em geral, estas coletas podem ser realizadas utilizando APIs de coleta de dados ou rotinas de rastreamento e raspagem (web crawler e web scraping) (BENEVENUTO, ALMEIDA, SILVA, 2011; MUNZERT *et al.*, 2014).

Preparação de dados

A preparação de dados é uma das etapas mais importante do pipeline de big data, pois é a etapa em que são realizadas as operações para melhorar a qualidade dos dados. A qualidade dos dados vai determinar a eficiência e acuracidade das análises de dados. Os conjuntos de dados são susceptíveis a ruídos, valores faltantes e outras inconsistências. Desta forma, a preparação de dados visa, acima de tudo, transformar os dados brutos em um formato plausível para ser utilizado nas análises.

As operações de preparação consistem em limpar, enriquecer, normalizar, integrar e combinar dados para análise, e incluem uma ampla gama de métodos que são usados principalmente para os seguintes fins:

Limpeza dos dados: operações de tratamento sobre os dados pré-existentes, de forma a assegurar a qualidade (completude, consistência, veracidade e integridade). Consiste em resolver problemas como a retirada de dados duplicados, correção de dados corrompidos ou inconsistentes, tratamento de valores ausentes ou inaplicáveis; detecção e remoção de anomalias (ruídos, outliers, valores de dados irregulares, incomuns e indesejados).

Enriquecimento de dados: operações que visam agregar aos dados existentes mais dados (detalhes) tornando os dados mais ricos, de modo que possam contribuir no processo de descoberta de conhecimento. Em geral, a partir de dados existentes é possível coleta novos dados externos, que tenham algum grau correlação, por meio de processos de integração e combinação de dados.

Transformação de dados: operações que visam transformar os dados conforme alguma regra, por exemplo, padronização e normalização dos dados, conversão de valores categóricos em numéricos e vice-versa, geração de hierarquia de conceitos.

Integração de dados: operações que visam a fusão de dados de fontes distintas em um único conjunto de dados.

Redução dos dados: operações que visam criar um conjunto reduzido da série de dados que produz (quase) o mesmo efeito nas análises. Pode ser por redução de dimensionalidade ou redução no volume de dados.

APIs de coleta de dados

O acrônimo API corresponde às palavras em inglês “*Application Programming Interface*”, em português “Interface de Programação de Aplicações”. Uma API é um middleware ou software intermediário que permite que dois aplicativos se comuniquem. Quando você usa um aplicativo de mídia social como o Facebook ou Twitter, ou envia uma mensagem instantânea ou verifica o clima em app no seu telefone, você está usando uma API.

Uma API é composta por um conjunto de rotinas (programas) que são responsáveis por realizar várias operações previamente conhecidas e divulgadas pelo fornecedor da própria API. Por meio das APIs é possível utilizar suas funcionalidades seguindo os protocolos previamente definidos. As APIs permitem que os desenvolvedores economizem tempo aproveitando a implementação de uma plataforma para realizar o trabalho. Isso ajuda a reduzir a quantidade de código que os desenvolvedores precisam criar e também cria mais consistência entre aplicativos para a mesma plataforma.

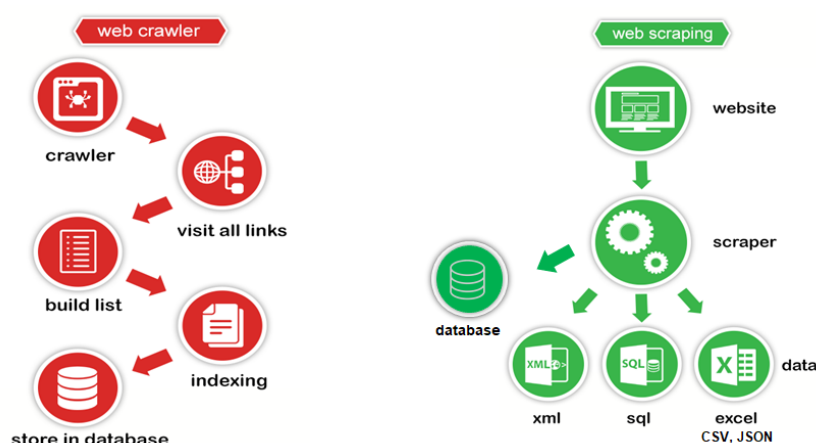
Web Crawler e web scraping

A raspagem (Scraping) e o rastreamento (crawling) são dois métodos muito utilizados para coletar dados na web (VANDEN BROUCKE,BAESENS, 2018). O Web Crawling ou rastreamento da web é o processo de localização de informações na World Wide Web (WWW), indexando todas as palavras de um documento, adicionando-as a um banco de dados, seguindo todos os hiperlinks e índices e adicionando essas informações também ao banco de dados (KAUSAR,DHAKA,SINGH, 2013; THELWALL, 2001).

Já o Web Scraping ou raspagem da web é o ato de recuperar informações específicas da World Wide Web ou Websites (diferentes sites), ou seja, processo de solicitar automaticamente um documento da Web e coletar informações dele. Em geral, para fazer scraping na web, pode ser necessário realizar algum crawling na web para navegar entre os sites (MITCHELL, 2018).

Apesar de serem semelhantes e até sobrepostos, podemos ver algumas diferenças conceituais entre estes métodos (Figura 21).

Figura 21 – Diferença entre Scraping e Crawling.



Fonte: <http://proweb scraping.com/web-scraping-vs-web-crawling>.

Referências

ALMEIDA, M. B. d. Revisiting ontologies: A necessary clarification. *Journal of the American Society for Information Science and Technology*, 64, n. 8, p. 1682-1693, 2013.

AZEVEDO, A. I. R. L.; SANTOS, M. F. *KDD, SEMMA and CRISP-DM: a parallel overview*. IADS-DM, 2008.

BARBIERI, C. P. *BI2 - Business Intelligence: Modelagem e Qualidade*. 1. ed. Rio de Janeiro: Elsevier, 2011. 416 p p. 978-8535247220.

BARBIERI, C. P. *BI-business intelligence: modelagem e tecnologia*. Axcel Books, 2001. 8573231483.

BDW, B. D. W. *Introduction about NoSQL Data Models*. Big Data World. online. 2019 2014.

BENEVENUTO, F.; ALMEIDA, J. M.; SILVA, A. S., 2011, Campo Grande, Brasil. *Explorando redes sociais online: Da coleta e análise de grandes bases de dados às aplicações*. Sociedade Brasileira de Computação, 2011. 63-102.

BERNERS-LEE, T. *Linked Data*. 2006. Disponível em: <<https://www.w3.org/DesignIssues/LinkedData.html>>. Acesso em: 11 mai. 2021.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific american*, 284, n. 5, p. 28-37, Apr 26 2001.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, p. 205-227, 2009.

CANALTECH. *EMC oferece solução de armazenamento e análise de Data Lake - Infra*. online, 2015-04-03 2015. Disponível em: <<https://canaltech.com.br/infra/EMC-oferece-solucao-de-armazenamento-e-analise-de-Data-Lake/>>. Acesso em: 11 mai. 2021.

CHEN, P. P. *Modelagem de dados: a abordagem entidade-relacionamento para projeto lógico*. Makron Books do Brasil, 1990.

CHEN, P. P. The entity-relational model toward a unified view of data. *ACM Trans, on Database Systems*, 1, n. 1, p. 1-49, 1976.

COUGO, P. S. *Modelagem conceitual e projeto de banco de dados*. 1ª Ed. 18ª Reimp. ed. Rio de Janeiro: Elsevier, 1997.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. *Sistemas Distribuídos: Conceitos e Projeto*. Bookman Editora, 2013.

COX, M.; ELLSWORTH, D., 1997, English, Phoenix, AZ, USA. *Application-controlled demand paging for out-of-core visualization*. IEEE Computer Society Press. 235-ff. Disponível em: <<https://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf>>. Acesso em: 11 mai. 2021.

CURRY, E. The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches. In: CAVANILLAS, J. M.; CURRY, E. e WAHLSTER, W. (Ed.). *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*. Springer, 2016. p. 29-37. Disponível em: <https://link.springer.com/content/pdf/10.1007%2F978-3-319-21569-3_3.pdf>.

Acesso em: 11 mai. 2021.

DAMA. *DAMA-DMBOK: Data management body of knowledge* Bas King Ridge, New Jersey, USA: Technics Publications LLC, 2017. 624 p.

DAMA. *The DAMA Guide to The Data Management Body of Knowledge (DAMA-DMBOK Guide)*. Bas King Ridge, New Jersey, USA: Technics Publications LLC, 2009. 406 p.

DATA SCIENCE GUIDE, D. *Exploratory data analysis*. Data science guide. online. 2021.

DAVENPORT, T. H. *Ecologia da informação: por que só a tecnologia não basta para o sucesso na era da informação*. Tradução ABRÃO, B. S. 2ª Edição ed. São Paulo: Futura, 1998. 292 p..

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 4ª Ed. São Paulo: Addison Wesley, 2005.

FARINELLI, F. *Improving semantic interoperability in the obstetric and neonatal domain through an approach based on ontological realism*. Orientador: ALMEIDA, M. B. d. 2017. 256 f. Doctoral (Doctor in Information Science) - School of Information Science Federal University of Minas Gerais at Brazil, Belo Horizonte. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/BUBD-AX2J5B>>. Acesso em: 11 mai. 2021.

FOWLER, M. **Nosql Definition**. 2016. Disponível em: <<https://martinfowler.com/bliki/NosqlDefinition.html>>. Acesso em: 11 mai. 2021.

HEATH, T.; BIZER, C. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1, n. 1, p. 1-136, 2011.

HECKERT, N. A.; FILLIBEN, J. J. *NIST/SEMATECH e-Handbook of Statistical Methods*; Chapter 1: Exploratory Data Analysis. 2003.

HEUSER, C. A. *Projeto de Banco de Dados*. 6. ed. Porto Alegre: Bookman, 2008. 282 p.

ISOTANI, S.; BITTENCOURT, I. I. *Dados Abertos Conectados: Em busca da Web do Conhecimento*. Novatec Editora, 2015.

KAUSAR, M. A.; DHAKA, V. S.; SINGH, S. K. Web crawler: a review. *International Journal of Computer Applications*, 63, n. 2, 2013.

LANEY, D. 3D data management: Controlling data volume, velocity and variety. *META Group Research Note*, 6, p. 70-73, 2001.

MAYER-SCHÖNBERGER, V.; CUKIER, K. *Big data: a revolution that will transform how we live, work, and think*. Boston: Houghton Mifflin Harcourt, 2013 2013. 242 p.

MEDRI, W. *Análise exploratória de dados*. Apostila do curso de especialização em Estatística. Londrina: Universidade Estadual de Londrina 2011.

MITCHELL, R. *Web scraping with Python: Collecting more data from the modern web*. O'Reilly, 2018.

MUNZERT, S.; RUBBA, C.; MEIßNER, P.; NYHUIS, D. *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons, 2014.

NASCIMENTO, J. P. B. *A carreira dos profissionais de Ciência de Dados, Engenharia de Dados e Machine Learning*. 2017. Disponível em: <<http://igti.com.br/blog/carreira-big-data-engenharia-dados-machine-learning/>>. Acesso em: 11 mai. 2021.

OKI, O. K. I. *Guia de Dados Abertos*. 2019. Disponível em: <http://opendatahandbook.org/guide/pt_BR/>. Acesso em: 11 mai. 2021.

PARUCHURI, V. *What is Data Engineering?*. 2017. Disponível em: <<https://www.dataquest.io/blog/what-is-a-data-engineer/>>. Acesso em: 11 mai. 2021.

REHMAN, M. H. u.; CHANG, V.; BATOOL, A.; WAH, T. Y. Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management*, 36, n. 6, Part A, p. 917-928, 2016.

SADALAGE, P. J.; FOWLER, M. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2013.

SCIME, A. *Web Mining: applications and techniques*. IGI Global, 2005.

SHAFIQUE, U.; QAISER, H. A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12, n. 1, p. 217-222, 2014.

SHIVALINGAIAH, D.; NAIK, U. Comparative Study of Web 1.0, Web 2.0 and Web 3.0. In: *6th International CALIBER*, 2008, Allahabad, Índia. p. 499-507.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistemas de banco de dados*. 6ª. ed. São Paulo: Elsevier, 2012.

SILVA, E. M. *Descoberta de conhecimento com o uso de text mining : cruzando o abismo de moore*. 2002. Universidade Católica de Brasília. Disponível em: <<https://bdtd.ucb.br:8443/jspui/handle/123456789/1462>>. Acesso em: 11 mai. 2021.

SOUSA, F. R.; MOREIRA, L. O.; MACÊDO, J. A. F. d.; MACHADO, J. C. Gerenciamento de dados em nuvem: Conceitos, sistemas e desafios. *Topicos em*

sistemas colaborativos, interativos, multimedia, web e bancos de dados, Sociedade Brasileira de Computacao, p. 101-130, 2010.

SOUSA, F. R.; MOREIRA, L. O.; MACHADO, J. C. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI)*, p. 150-175, 2009.

STROHBACH, M.; DAUBERT, J.; RAVKIN, H.; LISCHKA, M. Big Data Storage. *In: CAVANILLAS, J. M.; CURRY, E. e WAHLSTER, W. (Ed.). New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*. Cham: Springer International Publishing, 2016. p. 119-141.

TAN, A.-H., 1999, *Text mining: The state of the art and the challenges*. sn. 65-70. Disponível em: <http://www.ntu.edu.sg/home/asahtan/papers/tm_pakdd99.pdf>. Acesso em: 11 mai. 2021.

TAURION, C. *Big data*. Rio de Janeiro: Brasport, 2013.

TAURION, C. *Cloud computing-computação em nuvem*. Brasport, 2009.

THELWALL, M. A web crawler design for data mining. *Journal of Information Science*, 27, n. 5, p. 319-325, 2001.

TUKEY, J. W. *Exploratory data analysis*. Reading, Mass., 1977.

VANDEN BROUCKE, S.; BAESENS, B. From web scraping to web crawling. *In: Practical Web Scraping for Data Science*: Springer, 2018. p. 155-172.

WHITE, T. *Hadoop: The definitive guide*. O'Reilly Media, 2012.