

## CAPÍTULO 1

# **Introdução**

Este capítulo contextualiza os principais assuntos abordados neste trabalho, apresenta as motivações, os objetivos gerais e específicos da proposta desta pesquisa, bem como sua justificativa.

## 1.1 Contextualização

Desde a década de 1970, com a criação do modelo relacional por Edgar Frank Codd, a estrutura de armazenamento adotada por muitos desenvolvedores de sistemas da área de tecnologia da informação tem se baseado no conceito de entidade e relação proposto por Codd. A maioria dos sistemas gerenciadores de banco de dados que possui aceitação no mercado fazem uso desse modelo, por exemplo o MySQL, Oracle e Microsoft SQL Server. Porém, os requisitos para o desenvolvimento de ferramentas de software modernas têm mudado significativamente, especialmente com o aumento das aplicações Web [Nas12]. Este segmento de aplicações exige requisitos com alta escalabilidade e vazão, onde sistemas que utilizam um armazenamento com esquema relacional não conseguem atender satisfatoriamente. Em resposta a isso, novas abordagens de armazenamentos de dados utilizando o termo de NoSQL tornaram-se popular.

O termo NoSQL é constantemente interpretado como *"Not Only SQL"*, cujo SQL refere-se a linguagem de manipulação de dados dos gerenciadores de armazenamento de dados relacionais (RDBMS - Relational Database Management System) - Structure Query Language [Nas12]. O grande propósito das abordagens NoSQL é oferecer alternativas onde os esquemas relacionais não apresentam um bom desempenho. Esse termo abrange diferentes tipos de sistemas. Em geral, banco de dados NoSQL usam modelo de dados não-relacionais, com poucas definições de esquema, são executados em clusters e aplicados a alguns bancos de dados recentes como o Cassandra, o Mongo, o Neo4J e o Riak [FS13].

Muitas empresas coletam e armazenam milhares de gigabytes de dados por dia, no qual a análise desses dados torna-se uma vantagem competitiva no mercado. Por isso, há uma grande necessidade de uma nova arquitetura para o gerenciamento de suporte à decisão que possa alcançar melhor escalabilidade e eficiência [LTP13]. Para auxiliar no processo de gerenciamento de suporte à decisão uma das formas mais utilizadas é a criação de um ambiente data warehousing que é responsável por providenciar informações estratégicas e esquematizadas a respeito do negócio [CD97].

Segundo a definição de [KR02], data warehouse (DW) é uma coleção de dados para o processo de gerenciamento de suporte à decisão orientado a assunto, integrado, variante no tempo e não volátil. Os dados de diferentes fontes de sistemas são processados em um data warehouse central através da Extração, Transformação e Carga (ETL) de maneira periódica.

Ferramentas de ETL são sistemas de software responsáveis por extrair dados de diversas fontes, transformar e customizar os dados e inseri-los no data warehouse. Comumente, esses processos são executados periodicamente, onde a otimização do seu tempo de execução torna-se importante [PAS05].

O projeto de ETL consome cerca de 70% dos recursos de implantação de um DW, pois desenvolver esse projeto é crítico e custoso, tendo em vista que gerar dados incorretos pode acarretar em más decisões. Porém, por algum tempo pouca importância foi dada ao processo de ETL pelo fato de ser visto somente como uma atividade de suporte aos projetos de DW. Apenas a partir do ano 2000, a comunidade acadêmica passou a dar mais importância ao tema [dS12].

Tradicionalmente, o DW é implementado em uma base de dados relacional, onde o dado é armazenado nas tabelas fato e tabelas dimensões, na qual forma um esquema em estrela

[KR02]. Por isso, é comum que as ferramentas de ETL utilizadas no mercado atualmente só dêem suporte aos esquemas relacionais. Para oferecer suporte aos sistemas que necessitem utilizar um esquema não relacional de BDs NoSQL em DW, a proposta desse trabalho é especificar um framework programável, flexível e integrado para modelagem e execução de processos ETL em BDs NoSQL.

## 1.2 Motivação

Uma alternativa para organizar e manipular grandes volumes de dados sem utilizar um modelo relacional e ainda processá-los e armazená-los de maneira distribuída é fazer o uso de BDs NoSQL [dCS16]. Com isso, surge a necessidade de se promover meios para o uso desses BDs em DWs.

As pesquisas presentes na literatura sobre extração de dados em BDs NoSQL mostram que não há uma ferramenta que seja integrada para o uso de BDs NoSQL, as ferramentas existentes no mercado apenas oferecem a possibilidade para alguns SGBDs NoSQL, ficando a cargo da equipe de implantação do projeto de DW todo o trabalho de modelagem e programação ao se utilizar BDs NoSQL.

[dS12] aponta em sua pesquisa que muitas empresas evitam ferramentas de ETL disponíveis no mercado, e adotam o desenvolvimento dos processos a partir de uma linguagem de programação de propósito geral, pelo fato dessas ferramentas terem uma longa curva de aprendizagem e grande complexidade no seu uso.

O aumento do uso de banco de dados com esquemas não relacionais baseados no paradigma NoSQL e a falta de uma ferramenta programável, flexível e integrada, independente de plataforma que dê suporte à extração, transformação e carga em data warehouses para esses esquemas é a grande motivação deste trabalho.

Dessa forma, encontrar uma solução que seja programável, flexível e integrada para extração, transformação e carga dos dados em BDs NoSQL é a proposta deste trabalho.

## 1.3 Objetivos

Nesta seção serão apresentados os objetivos geral e específicos desta pesquisa.

### 1.3.1 Objetivo Geral

Esta pesquisa propõe o ETL4NoSQL que é um framework para desenvolvimento de aplicações ETL para sistemas que utilizam bases de dados NoSQL, com poucas definições de esquemas e não relacionais. Bases de dados NoSQL são utilizadas em muitas aplicações modernas, principalmente aplicações Web, onde possuem um grande volume e variedade de dados.

Para que seja possível a extração, transformação e carga dos dados armazenados em bancos de dados que utilizam um dos paradigmas de NoSQL é preciso que seja definido o esquema no qual os dados necessários estão armazenados. Dessa forma, algumas questões importantes são abordadas:

- Quanto os BDs NoSQL diferem dos BDs suportados pelas ferramentas de ETL?
- Como é possível oferecer suporte para ETL em BDs NoSQL?

As diferenças dos BDs NoSQL são abordadas nos capítulos de fundamentação teórica. Assim, fica explícito o problema de como é possível converter os modelos de dados NoSQL em modelos relacionais que possam ser lidos por qualquer ferramenta de ETL. [Nas12] define uma tabela como uma representação de uma coleção de instâncias de entidades comparáveis. Então, possibilitar suporte para armazenamentos de dados NoSQL é permitir extração e importação de instâncias de entidades comparáveis em tabelas relacionais. Porém, os problemas a serem resolvidos abordados por [Nas12] para atingir isso incluem:

1. Como permitir o usuário especificar as entidades comparáveis e seus atributos?
2. Como, e de onde, extrair exatamente?

A solução sugerida pelo autor é um esquema ser deduzido por meio de uma amostra do banco de dados. Este esquema pode, então, ser apresentado ao usuário que procede selecionando quais atributos importar do que foi apresentado - exatamente como no caso do RDBMS. É claro que também deverá ser possível ao usuário editar os esquemas apresentados tendo em vista que a amostra pode não ser perfeita. Essa abordagem geral é referida como dedução de esquema - *schema inference*.

Construir uma amostra com todo o banco de dados pode ser muito custoso, por isso é preferível que faça uma amostra com pequenas estruturas, como por exemplo, uma entidade inteira. É claro que muitas vezes não há esse tipo de estrutura, então é possível que a amostra seja colocada em clusters. Múltiplas entidades podem ser divididas cada uma em um cluster, e assim, é possível ter um esquema de entidade por cluster.

Dessa forma, dado o que extrair, consultar e recuperar dados de um RDBMS é direto por causa do SQL. Sistemas NoSQL geralmente tem diferentes interfaces que suportam diferentes tipos de consultas. Então, não há nenhuma sugestão de solução geral para isso. Ao invés disso, uma investigação de cada interface particular dos sistemas deve ser conduzida para resolver o problema [Nas12].

Para suprir o problema a respeito das várias interfaces a serem lidas na extração de dados de bases NoSQL, sugerimos a criação de um ambiente programável que oferecesse interfaces previamente selecionadas e permitisse a inserção de novas interfaces por meio de linguagem de programação. Essas interfaces são fundamentadas nos princípios de flexibilidade, reuso e inversão de controle adotadas pela literatura para desenvolvimento de frameworks [EC97] [V.03] [ECE99] [Sam97] [dS12]. Assim, a arquitetura do ETL4NoSQL dispõe de uma interface de programação com elementos para a importação dos dados das várias bases NoSQL, o mapeamento e manipulação desses dados para um modelo relacional que possa ser lido por qualquer ferramenta de ETL, possibilita também a criação de modelos não relacionais para o uso de DWs não relacionais, por meio da interface de programação, além de elementos que encapsulam os operadores de transformação, carga de dados que são fundamentos de ETL presentes na literatura [KC04] [dS12] [Hei01]. Para proporcionar a especialização dos elementos de importação e mapeamento do framework para diversos modelos de dados utilizamos uma abordagem

baseada em padrões, que possibilita a implementação do framework utilizando uma linguagem de programação de propósito geral.

Com isso, o objetivo principal desta pesquisa é especificar um framework programável, flexível e integrado para modelagem e execução de processos ETL em BDs NoSQL.

### 1.3.2 Objetivo Específico

O primeiro objetivo específico desta pesquisa é estender a proposta do framework para facilitar a carga de dados de dois sistemas de BD NoSQL distintos baseado no mesmo paradigma NoSQL em um DW relacional modelados pelo esquema estrela, tendo em vista que este é o esquema de dados dimensionais mais recomendado pela literatura [KR02] [?]. O outro objetivo específico é ao invés de dar carga em um DW relacional fazer uso do mesmo sistema em um DW NoSQL, seguindo a metodologia adotada por [CMK<sup>+</sup>15] em seu trabalho de pesquisa. Para isso, desenvolvemos dois frameworks especializados a partir do ETL4NoSQL em conformidade às peculiaridades dos processos de ETL nessas duas áreas de aplicação, os quais também são objetos de validação do ETL4NoSQL.

## 1.4 Justificativa

A integração de dados e os processos de ETL são procedimentos cruciais para a criação de data warehouses e sistemas BI (business intelligence). Porém, os sistemas para ETL e integração de dados são tradicionalmente desenvolvidos para dados estruturados em modelos relacionais que representam apenas uma pequena parte dos dados mantidos por muitas empresas [DBcRA05] [Russom 2007, Pedersen 2009]. Dessa forma, existe uma demanda crescente para integrar os dados não estruturados e semi estruturados em um repositório unificado. Devido a complexidade desses dados, novos desafios estão surgindo quando lidamos com dados heterogêneos e distribuídos no ambiente de integração [Salem, 2012].

Além disso, muitas empresas encontram dificuldades para lidar com as ferramentas ETL disponíveis no mercado. Aprender a lidar com essas ferramentas pode ser muito custoso em termos financeiros e de tempo, e por isso, acabam optando desenvolver os seus processos por meio de uma linguagem de programação de propósito geral [Awad et al., 2011; Muñoz et al., 2009].

Portanto, este trabalho propõe um framework programável para desenvolvimento de sistemas de ETL que possibilita a integração de dados não estruturados e semi estruturados armazenados em bases NoSQL. O framework possui um ambiente integrado para a importação e mapeamento dos dados, além da modelagem e customização dos processos de ETL. Os processos de importação e mapeamento do framework integram dados não estruturados e semi estruturados. Esses processos possibilitam a leitura e manipulação de dados de bases NoSQL, e também o armazenamento desses dados em bases deste tipo, oferecendo uma alternativa não relacional para a construção de DWs.

## **1.5 Organização do Trabalho**