Universidade do Minho
Escola de Engenharia

Marta Susete Carvalho Batista Nogueira

# Modelação Ágil para Sistemas de Big Data Warehousing

Dissertação de Mestrado

Mestrado Integrado Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação da

Professora Doutora Maribel Yasmina Santos

Outubro 2019

## DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

## RESUMO

Os Sistemas de Informação, com a popularização do conceito de *Big Data* começaram a considerar aspetos relativos às infraestruturas capazes de lidar com a recolha, armazenamento, processamento e análise de vastas quantidades de dados heterogéneos, como pouca estrutura (ou nenhuma) e gerados a velocidades cada vez maiores. Estes têm sido os desafios inerentes à transição do *Data Modelling* em *Data Warehouses* tradicionais para ambientes de *Big Data.*

O estado-de-arte reflete que a área científica de *Big Data Warehousing* é recente, ambígua e apresenta lacunas relativas a abordagens para a conceção e implementação destes sistemas; deste modo, nos últimos anos, vários autores motivados pela ausência de trabalhos científicos e técnicos desenvolveram estudos na área com o intuito de explorar modelos adequados (representação de componentes lógicas e tecnológicas, *data flows* e estruturas de dados), métodos e instanciações (casos de demonstração recorrendo a protótipos e *benchmarks*).

A presente dissertação está inserida no estudo da proposta geral dos padrões de *design* para sistemas de *Big Data Warehousing* (M. Y. Santos & Costa, 2019) e, posteriormente, é efetuada a proposta de um método, em vista a semiautomatização da proposta de *design* dos autores referidos, constituído por sete regras computacionais, apresentadas, demonstradas e validadas com exemplos baseados em contextos reais. De forma a apresentar o processo de modelação ágil, foi criado um fluxograma para cada regra, permitindo assim apresentar todos passos. Comparando os resultados dos exemplos obtidos após aplicação do método e dos resultantes de uma modelação totalmente manual, o trabalho proposto apresenta uma proposta de modelação geral, que funciona como uma sugestão de modelação de *Big Data Warehouses* para o utilizador que, posteriormente, deve validar e ajustar o resultado tendo em consideração o contexto do caso em análise, as *queries* que pretende utilizar e as características dos dados.



**PALAVRAS-CHAVE:** *Big Data Warehousing, Big Data Modelling, Data Engineering.*

## ABSTRACT

Information Systems, with the popularization of Big Data, have started to consider the aspects related to infrastructures capable of dealing with collection, storage, processing and analysis of vast amounts of heterogeneous data, with little or no structure and produced at increasing speed. These have been the challenges inherent to the transition from Data Modelling into traditional Data Warehouses for Big Data environments.

The state-of-the-art reflects that the scientific field of Big Data Warehousing is recent, ambiguous and that it shows a few gaps regarding the approaches to the design and implementation of these systems; thus, in the past few years, several authors, motivated by the lack of scientific and technical work, have developed some studies in this scientific area in order to explore appropriated models (representation of logical and technological components, data flows and data structures), methods and instantiations (demonstration cases using prototypes and benchmarks).

This dissertation is inserted in the study of the general proposal of design standards for Big Data Warehousing systems. Late on, the proposed method is comprised of seven sequential rules which are thoroughly explained, demonstrated and validated with relevante exemples based on common real use-cases. For each rule, step-by-step flowchart is provider an agile modelling process. When compared a fully manual example, the proposed work offered a correct but genereal resulting model that works best as a first modelling effort that should then be validated by a use-case expert.

**KEYWORDS:** Big Data Warehouse, Big Data Modelling, Data Engineering.

# ÍNDICE

# Índice de Figuras

## ÍNDICE DE TABELAS

## Lista de Abreviaturas, Siglas e Acrónimos

ACID – *Atomicity, Consistency, Isolation, and Durability*

ADM – *Analytical Data Model*

API – *Application Programming Interface*

BD – *Big Data*

BDW – *Big Data Warehouse*

BI – *Business Intelligence*

BWW – *Bunge-Wand-Weber*

CAP – *Consistency, Availability, Partition Tolerance*

CEP – *Collection, Preparation, and Enrichement*

DDL – *Data Definition Language*

DER – Diagrama Entidade Relacionamento

DSRM – *Design Science Research Methodology*

DW – *Data Warehouse*

FK – *Foreign Key*

GFS – *Google File System*

HDFS – *Hadoop Distributed File System*

JDBC – *Java Database Connectivity*

JSON – *JavaScript Object Notation*

MDM – *Multidimensional Data Models*

MPP – *Massively Parallel Processing*

NDFS – *Nutch Distributed File System*

NoSQL – *Not only SQL*

ODBC – *Open Database Connectivity*

OLAP – *Online Analytical Processing*

OLPT – *Online Transaction Processing*

PK – *Primary Key*

SQL – *Structured Query Language*

XML – *Extensible Markup Language*

# 1. INTRODUÇÃO

Neste capítulo introdutório é apresentado o enquadramento e motivação inerentes ao tema, os principais objetivos e a abordagem metodológica a ser adotada. Por fim, é definido o planeamento das tarefas da dissertação, assim como o processo de revisão de literatura, os contributos associados a este trabalho e a definição da estrutura do documento.

## 1.1 Enquadramento e Motivação

Um fator importante a ter em consideração é que o volume dos dados está a crescer a um ritmo significativamente rápido com a grande quantidade de recolha e armazenamento de dados e com o surgimento de novas fontes de dados como, por exemplo, as redes sociais e sensores (Cassavia, Dicosta, Masciari, Saccà, 2014). Os desafios introduzidos por esse aumento introduzem o conceito de *Big Data,* que pode ser definido como grandes quantidades de dados disponíveis em vários graus de complexidade, gerados em diferentes velocidades e com diferentes graus de ambiguidade, que podem não se adequar a tecnologias, métodos e algoritmos tradicionais (Krishnan, 2013).

Num ambiente tradicional de *Data Warehousing* os repositórios assentam em bases de dados relacionais e modelos de dados estruturados, que atualmente já não são capazes de suportar a explosão de dados. Estes novos paradigmas da era do *Big Data* trouxeram dificuldades ao nível do armazenamento tradicional de dados e no processamento destes por parte das tecnologias existentes. Com o objetivo de dar resposta a esses desafios, surge o recente tópico de investigação por *Big Data Warehouses*, por meio do qual se procura a adoção de novos modelos lógicos que permitam escalabilidade, desempenhos significativos em tempo útil, armazenamento flexível, capacidade de lidar com o crescente volume, variedade e velocidade dos dados e, por fim, uma melhor gestão de dados não estruturados e desnormalizados; por exemplo, os modelos utilizados nas bases de dados NoSQL (Costa, Andrade, & Santos, 2018; Di Tria, Lefons, & Tangorra, 2014).

Em ambientes de *Big Data Warehousing* a estrutura dos dados pode mudar ao longo do tempo, devido aos requisitos analíticos e de armazenamento; por isso, é importante considerar um *Data Modelling* adequado, para garantir que as necessidades analíticas são devidamente consideradas. Contudo, o conceito de *Big Data Modelling* é relativamente recente e apresenta escassez de artigos científicos, porque, até ao momento, os estudos desenvolvidos nesta área científica não são concordantes no que se refere ao conceito de *Data Modelling* em contextos de *Big Data*, assim como à sua descrição

de abordagem e métodos. Por este motivo (M. Y. Santos & Costa, 2019) apresentaram a primeira proposta de abordagem de *Data Modelling* para sistemas de *Big Data Warehousing.* Deste modo, e, com o objetivo de agilizar os padrões de *design* definidos pelos autores, surgiu o trabalho desta dissertação de mestrado.

## 1.2 Objetivos e Resultados Esperados

O objetivo desta dissertação é a inferência de um método constituído por um conjunto de regras computacionais em vista a semiautomatização da proposta de abordagem de *Data Modelling* em contextos de *Big Data Warehousing.* Deste modo, a presente dissertação focar-se-á nos seguintes objetivos:

- Compreender a proposta da abordagem geral do método de *Data Modelling* para sistemas de *Big Data Warehouse,* elaborado pelos autores (Santos & Costa, 2019);
- Definir uma base de conhecimento com padrões de *design* possíveis, que já estão definidos pelos autores (Santos & Costa, 2019) e as regras que devem ser aplicadas em cada um;
- Desenvolver um conjunto de regras que infiram o método apropriado, tendo em consideração as características dos dados (regras guardadas na base de conhecimento).

## 1.3 Abordagem Metodológica

A abordagem metodológica a ser seguida no desenvolvimento desta dissertação tem como base a *Design Science Research Methodology* (DSRM) *for Information Systems* proposta por (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), visto ser uma metodologia de investigação enquadrada na área de Sistemas de Informação e adequada aos objetivos e tempo útil de desenvolvimento desta dissertação.

Como é possível verificar na Figura 1, a DSRM apresenta um processo nominal sequencial composto por seis fases, nomeadamente: a identificação do problema e motivação da dissertação, a definição de objetivos, o desenvolvimento do artefacto, a demonstração do artefacto como solução para o problema, a avaliação dessa solução e, por fim, a comunicação dos resultados de investigação que, neste contexto, resultará na publicação da dissertação e de outros contributos científicos relevantes. O modelo desta abordagem apresenta elevada flexibilidade, na medida em que pode ser iniciado num de quatro possíveis pontos de entrada da investigação, consoante o enquadramento do desafio de investigação, e permite uma abordagem iterativa ao processo de investigação, entre a avaliação/comunicação e a definição de proposta/desenvolvimento de uma solução.

O processo de investigação desta dissertação é iniciado no primeiro ponto de partida proposto pela abordagem metodológica, com foco na identificação do problema e nas principais motivações.

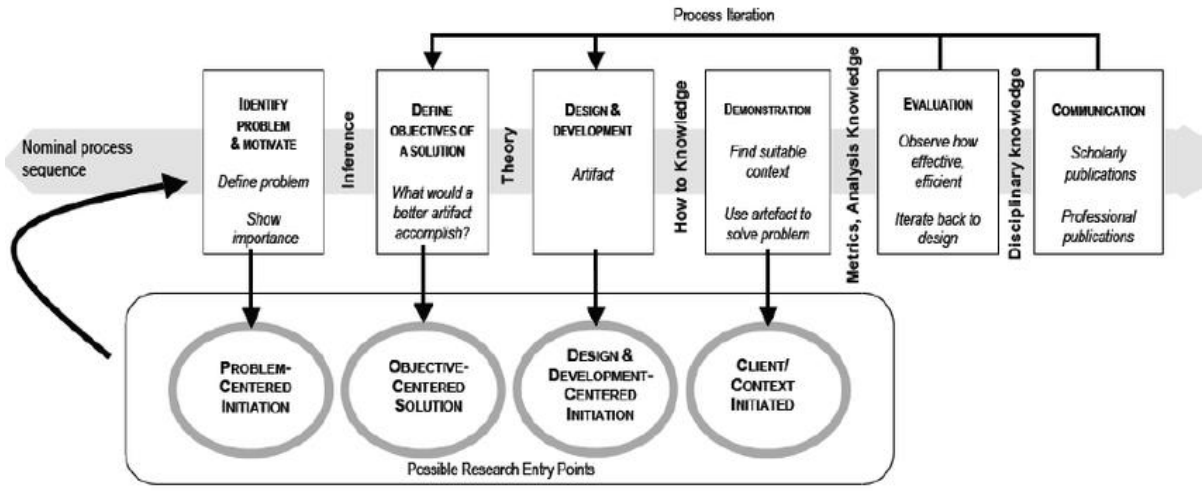



Figura 1 - Modelo DSRM [retirado de (Peffers et al., 2007)]

### 1.3.1 Fases da Abordagem de Investigação

Tendo em consideração a abordagem metodológica e as datas de entregas previstas pelo departamento, serão apresentadas nos pontos abaixo descritos as fases da abordagem de investigação:

1. Identificação do Problema e Motivação – nesta fase, é definido o problema relativo ao tema da dissertação e descrita a importância de uma solução para esse problema como contributo científico.

2. Definição dos Objetivos da Solução – nesta fase, são definidos os objetivos a atingir de forma a responder ao problema identificado.

3. Revisão de Literatura – nesta fase é feita a leitura referente aos conceitos relacionados com o tema da dissertação, de forma a compreender e conhecer o trabalho científico já desenvolvido na área. A elaboração do documento, revisão de literatura, é desenvolvida nesta fase da metodologia.

4. Enquadramento Tecnológico – nesta fase são caraterizadas algumas ferramentas tecnológicas fundamentais para a contextualização dos ambientes *Big Data*; deste modo, o estudo incide sobre três conceitos associados tipicamente ao ecossistema Hadoop.

5. Conceção e Desenvolvimento – nesta fase, é criado o artefacto que consiste num método que, segundo Peffers et al. (2007), constitui um conjunto de etapas usadas para executar tarefas. Desta forma, o método infere um conjunto de regras computacionais em vista a semi-

automatização da modelação; o método segue os padrões de *design* de *data modelling* propostos pelos autores Santos et al. (2019).

6. Demonstração – nesta fase, é demonstrada a aplicação do artefacto em três exemplos diferentes de *benchmarking* e um exemplo de uma base de dados aberta que monitoriza notícias de quase todo o mundo; estes exemplos foram previamente usados pelos autores já referidos.
7. Avaliação – avaliação da solução desenvolvida na fase anterior, na medida em que esta responde ao problema identificado. Se os resultados obtidos não forem de encontro aos definidos ou forem necessárias melhorias, o processo de desenvolvimento poderá ser reiterado.
8. Comunicação – esta fase acompanha toda a escrita e desenvolvimento da dissertação por meio da comunicação e da apresentação dos resultados obtidos a profissionais da área.

### 1.3.2 Processo de Revisão de Literatura

A criação de um processo de revisão de literatura foi fundamental para o enquadramento conceptual e tecnológico inerente a esta dissertação. O processo de elaboração da revisão do estado-de-arte envolveu a pesquisa em várias bases de dados de referência, uso de palavras-chave na pesquisa e no método de seleção dos artigos científicos.

No contexto desta dissertação, foram utilizadas palavras-chave para a pesquisa de artigos científicos, nomeadamente "*Big Data", "Big Data" AND Warehouse(ing), "Big Data Warehouse(ing)" AND Modelling*. Estas palavras chave foram usadas nas várias bases de dados de referência, designadamente "Scopus", "IEEE Xplore", "Science Direct", "ACM Digital Library", "Google Scholar", "Web of Science", "repositoriUM" e Google (pesquisa de sites oficiais).

Os conceitos-chave considerados nesta dissertação são relativamente recentes; como tal, foi estabelecida uma preferência por um período temporal para a recolha dos documentos compreendida entre 2001 e 2019, salvo exceções de maior relevância aos temas em investigação.

No método de seleção e recolha dos artigos científicos, foi feita, numa primeira fase, a análise do título e *abstract* do documento; se o mesmo for considerado relevante, é guardado. Numa segunda fase foram lidas as introduções e conclusões dos documentos guardados; as que apresentam interesse ao contexto da dissertação, foram lidas na íntegra.

## 1.4 Contributos do trabalho

O trabalho desenvolvido no âmbito desta dissertação permitiu a publicação do seguinte contributo científico:

1. Nogueira, M., Galvão, J., & Santos, M. Y. (2019). *An Agile Data Modelling for Big Data Warehouses.* In *16th European Mediterranean & Middle Eastern Conference (EMCIS).*

## 1.5 Organização do Documento

O presente documento está dividido em sete capítulos. No Capítulo 1, referente à introdução, são expostos o enquadramento e motivação à realização desta dissertação, os objetivos e resultados esperados e apresentada a abordagem metodológica, que inclui a descrição das fases da metodologia. É apresentado também o processo de revisão de literatura, os contributos científicos do trabalho realizado e a organização do documento aqui presente.

O Capítulo 2, denominado de enquadramento conceptual, introduz conceitos relevantes para a dissertação através de uma investigação sobre o estado-de-arte de três grandes temáticas associadas ao desafio de investigação: *Big Data*, *Big Data Warehouse* e *Big Data Modelling*.

No Capítulo 3 é apresentado o enquadramento tecnológico, numa abordagem semelhante ao anterior, incidindo fundamentalmente sobre uma contextualização do ecossistema Hadoop. Deste, são apresentadas e analisadas as ferramentas HDFS e Hive, assim como as bases de dados NoSQL.

O Capítulo 4 consiste na apresentação do processo de desenvolvimento do método. Na primeira secção do capítulo é apresentado um enquadramento global do método e uma breve introdução sumária das regras desenvolvidas. Na secção seguinte é detalhada individual e sequencialmente cada uma das regras do método.

O Capítulo 5, por sua vez, centra-se na demonstração da aplicação do método em diferentes casos de uso relevantes, sendo resultado de cada uma dessas aplicações comparada com o resultado dos autores Santos et al. (2019) para os mesmos casos no Capítulo 6, consagrando a avaliação do que foi desenvolvido.

Por fim, no Capítulo 7 são apresentadas as conclusões da dissertação, com uma reflexão sobre os resultados obtidos, uma síntese do trabalho realizado e uma apresentação de considerações sobre vertentes de investigação que possam ser exploradas no futuro.

# 2. ENQUADRAMENTO CONCEPTUAL

Com o objetivo de enquadrar conceitos inerentes a esta dissertação, o presente capítulo propõe uma visão geral relativa aos diversos conceitos da área. Os principais conceitos relativos ao tema da dissertação "Modelação Ágil para Sistemas de *Big Data Warehousing*" serão expostos nas secções seguintes. O capítulo atenta principalmente no conceito de *Big Data Modelling* e para compreensão do mesmo também serão sistematizados os conceitos de *Big Data* e *Big Data Warehouse*.

## 2.1 Big Data

O *Big Data* começou a ser referenciado em 1970 (Gali & Henk, 2012); contudo, só nos últimos anos é que o conceito ganhou relevo devido ao significativo aumento de fontes de dados, métodos e aplicações de recolha de dados, que, inevitavelmente, acabaram por ser responsáveis por gerar um elevado volume de dados oriundos maioritariamente de sensores, redes sociais, dispositivos eletrónicos, emails, páginas *web*, entre outros (Golab & Johnson, 2014; Mathur, Sihag, Bagaria, & Rajawat, 2014; Zikopoulos & Eaton, 2011).

O grande aumento de dados está diretamente relacionado com a origem do conceito *Big Data* e o autor Krishnan (2013) sugere que o *Big Data* pode ser definido como "quantidades de dados disponíveis em vários graus de complexidade, gerados em diferentes velocidades e com diferentes graus de ambiguidade, que podem não se adequar a tecnologias, métodos, algoritmos tradicionais".

A rápida evolução do conceito e de tecnologias *Big Data* originou alguma confusão (Gandomi & Haider, 2015); como tal, ao longo dos anos, e de forma a acompanhar o crescimento, foram vários os autores que definiram o conceito de *Big Data*, geralmente com opiniões distintas e envoltas em ambiguidade; no entanto, os autores De Mauro, Greco, & Grimaldi (2016) apresentaram uma proposta de uma definição formal, sugerindo que o *Big Data* pode ser caracterizado por um vasto volume, velocidade e variedade de dados, ao ponto de requererem tecnologias e métodos analíticos inerentes à sua transformação.

As três principais características de *Big Data* já referidas: volume, velocidade e variedade dos dados, estão retratadas no modelo dos 3Vs, conforme a Figura 2.

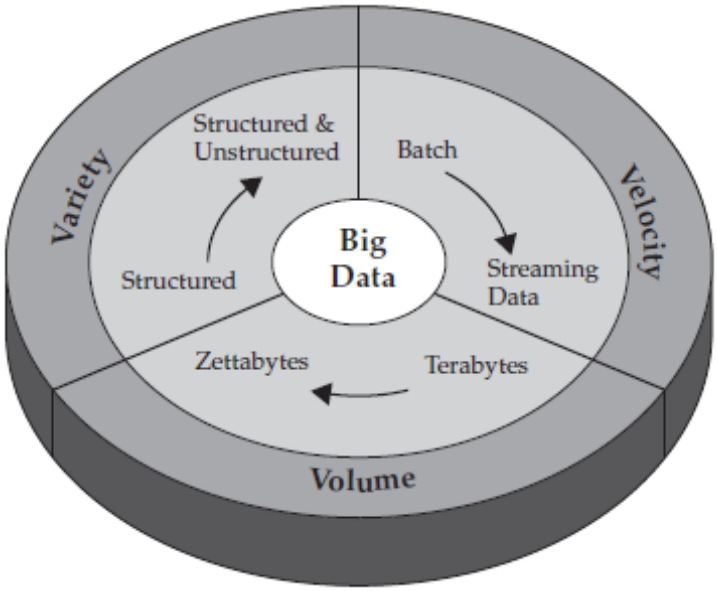



Figura 2 - Modelo 3Vs (volume, velocidade e variedade) [retirado de (Zikopoulos & Eaton, 2011)]

O volume é a primeira característica associada a *Big Data* e pode ser caracterizado pela grande quantidade de dados existentes atualmente, os que são gerados continuamente e os que serão recolhidos no futuro (Chandarana & Vijayalakshmi, 2014; Zikopoulos & Eaton, 2011). Segundo (Zikopoulos (2012), existem mais dados do que alguma vez existiram e o autor apresenta elementos sobre o seu volume de dados. Refere que no ano de 2000 existiam 800000 *petabytes* [PB] de dados armazenados em todo o mundo, sendo é expectável que em 2020 esse número venha a atingir os 35 *zettabyttes* [ZB] de dados armazenados).

A segunda característica, a variedade, diz respeito à receção de dados estruturados, não estruturados e semiestruturados, visto não existir controlo sobre o formato ou estrutura em que os mesmos vão surgindo. Os dados estruturados podem surgir, por exemplo, de bases de dados relacionais; no caso de serem dados não estruturados, surgem, por exemplo, de imagens, vídeos e textos; os dados semiestruturados, por exemplo, surgem da linguagem XML (Gandomi & Haider, 2015; Krishnan, 2013; Zikopoulos & Eaton, 2011).

A terceira caraterística, a velocidade, refere-se à rapidez com que os dados chegam e são guardados, tendo associadas, desse modo, taxas de criação de dados (Zikopoulos & Eaton, 2011). Tradicionalmente, os dados eram analisados em *batches;* estes, eram obtidos ao longo do tempo, processados e armazenados na base de dados ou ficheiro de destino, ao contrário do *Big Data* em que os fluxos de dados são analisados, quase em tempo real, de forma a que o tempo entre a recolha e o seu processamento seja o mais curto possível, sendo assim possível tomar decisões com base nos dados (Krishnan, 2013; Zikopoulos & Eaton, 2011).

Ao modelo dos 3V's, o autor Chandarana & Vijayalakshmi (2014) sugeriu serem acrescentadas mais duas características, a veracidade e o valor, desenvolvendo assim o modelo dos 5 V's. Posteriormente, Khan, Uddin, & Gupta (2014) apresenta o modelo dos 7 V's acrescentado as características de validade e volatilidade. A Figura 3 ilustra as sete características referidas.

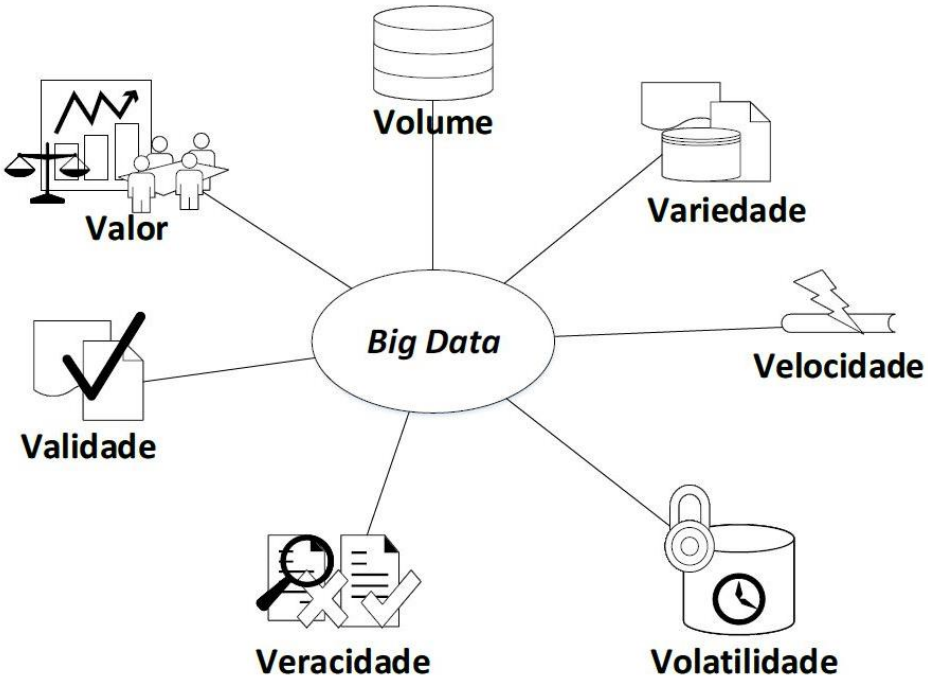



Figura 3 - 7 Vs do *Big Data* [retirado de (Lima, Francisca Vale, 2017)]

As organizações tiram proveito do armazenamento e tratamento de grandes quantidades de dados; dados que proporcionam oportunidades de aumentar a eficiência operacional, um fornecimento de serviços cada vez mais personalizado e o acompanhamento das alterações do mercado de forma a identificar e desenvolver produtos de acordo com as necessidades do consumidor; possibilita, ainda, a identificação de novos clientes e mercados, entre outros. (Fan, Han, & Liu, 2014; Labrinidis & Jagadish; Philip Chen & Zhang, 2014).

## 2.2 Big Data Warehouse

Com a rápida evolução e aceitação do *Big Data* começaram a surgir entraves relacionados com as grandes quantidades de dados gerados por diversas fontes, métodos e aplicações de recolha. Consequentemente, os tradicionais *Data Warehouse* (DW) e tecnologias inerentes ao mesmo não têm capacidade de suportar processamentos de grandes quantidades de dados (Kaisler, Armour, Espinosa, & Money, 2013; Krishnan, 2013; Zikopoulos & Eaton, 2011). Desta forma, surge o conceito de *Big Data Warehouse* (BDW) que apresenta diferenças substanciais relativamente aos *Data Warehouses*

tradicionais, na medida em que os esquemas desenvolvidos através de novos modelos lógicos acabam por ser mais flexíveis do que os modelos relacionais (F. Di Tria et al., 2014).

Os autores Di Tria et al. (2014) sugerem também que existem diferenças entre o *Big Data Warehouses* e os *Data Warehouses* tradicionais, afirmando que os seus modelos de dados devem ser baseados num *design* mais flexível. Portanto, os autores propõem uma metodologia que considera entidades e relacionamentos baseados no modelo *key-value,* e também é definido um conjunto de regras de que transforme os dados no modelo referido anteriormente; isto significa que, em vez de serem usados esquemas em estrela ou em floco de neve usados nos *Data Warehouses* tradicionais, os autores proponham o uso de modelos orientados em colunas e em documentos.

Para facilitar a compreensão entre os conceitos de *Data Warehouse* e *Big Data Warehouse*, os mesmos serão sistematizados em pontos distintos, de forma a explicitar a razão dos *Data Warehouses* tradicionais já não terem capacidade de resposta face aos desafios dos dias de hoje, bem como a forma de responder a esses mesmos desafios com o aparecimento do conceito de *Big Data Warehouse*.

### 2.2.1 Abordagem de Data Warehouse

Os autores Jing Han, Haihong E, Guan Le, & Jian Du (2011) referem-se ao *Data Warehouse* como um repositório que se mantém separado dos sistemas operacionais da organização, integrando informação, proveniente das mais variadas fontes de dados, que disponibiliza uma plataforma de dados históricos consolidados, com o objetivo de analisar e auxiliar o processo de tomada de decisão. No sistema de *Data Warehousing* tradicional, os sistemas operacionais *Online Transaction Processing* (OLTP) têm como finalidade registarem as transações que ocorrem. Estas mesmas operações são estruturadas, repetitivas e exportadas periodicamente para sistemas analíticos *On-Line Analytical Processing* (OLAP), passando por um processo de Extração-Carregamento-Transformação (ETL) (Kimball & Ross, 2013; Krishnan, 2013).

Um *Data Warehouse*, segundo os autores Golfarelli, Rizzi, & Pagliarani (2009), surge como uma consequência do grande volume de dados armazenados e pela necessidade crescente de utilizar estes mesmos dados como respostas que vão para além dos objetivos e tarefas inerentes a processos diários.

O autor Inmon (2015) define *Data Warehouse* como:

- Uma compilação de registos orientados por assunto e as necessidades analíticas das diferentes áreas da organização;

- Os dados integrados, selecionados e armazenados no *Data Warehouse* provêm de diferentes fontes de dados que são agregados de forma a delinear uma visão coerente;
- Os dados no *Data Warehouse* são estáveis, ou seja, depois de serem carregados não podem ser alterados ou eliminados, apenas serem adicionados dados entretanto acumulados em bases de dados operacionais;
- As variáveis ao longo do tempo e a informação fornecida é referente a uma perspetiva histórica; como tal, os dados têm de dispor de uma dimensão temporal.

Nas últimas décadas, os *Data Warehouses* foram reconhecidos como *enterprise data assets*; contudo, a evolução do tipo de análises (como por exemplo, o *data mining*, estatísticas e *queries* complexas), o aumento do volume de dados e a necessidade de analisar novos dados que surgem em tempo real, estão a implicar mudanças nas arquiteturas dos *Data Warehouses* (Russom, 2014). Na era do *Big Data,* os *Data Warehouses* estão a evoluir de forma a ampliar e modernizar, tendo como finalidade o suporte dos avanços tecnológicos e dos requisitos inerentes às organizações (como por exemplo, vantagens competitivas, eficiência operacional, tomada de decisão com base em dados e analítica). Atualmente, o processo de modernização apresenta desafios relacionados com a falta de competências, custos na implementação de novas tecnologias e dificuldades em desenvolver novas soluções que tenham capacidade de processar o volume cada vez maior de dados, assim como os seus diferentes tipos (Russom, 2016).

### 2.2.2 Abordagem de Big Data Warehousing

O *Big Data Warehousing* permite recolher, integrar e armazenar os grandes volumes de dados provenientes de diferentes fontes de dados, sejam eles estruturados ou não estruturados, e que são gerados com uma elevada velocidade, tendo variedade e complexidade nos dados (F. Di Tria et al., 2014).

Recentemente, os autores Francesco Di Tria, Lefons, & Tangorra (2018) apresentaram uma estrutura capaz de avaliar metodologias de *design* de *Big Data Warehouses,* tendo sido desenvolvido um conjunto de critérios como aplicação, agilidade, uma abordagem ontológica*,* paradigmas e uma modelação lógica. Os autores sugerem, igualmente, formas de dividir metodologias em classes (semiautomático, incremental e não relacional), assim como definiram as características a serem abordadas pela metodologia (valor, variedade e velocidade).

Os *Big Data Warehouses*, relativamente aos *Data Warehouses*, requerem novas características e mudanças na sua concretização segundo (Goss & Veeramuthu, 2013; Mohanty, S., Jagadeesh, M., & Srivatsa, H., 2013 ), com capacidades tais como:

- Processamento de dados altamente distribuídos;
- A baixo custo, suportar a escalabilidade;
- Analisar grandes volumes de dados sem recorrer a amostras;
- Melhorar o processo de tomada de decisão podendo, em tempo real, processar e visualizar dados;
- Integrar diversas estruturas de dados, seja de fontes de dados internos ou externos à organização;
- Gerir compensações entre consistência, disponibilidade e tolerância a partições;
- Suportar cargas de trabalho extremas, como consultas de amplitude (consultas *ad hoc*, *data mining* e análises estratégicas);
- Carregamento e processamento de dados *batch* (neste processamento, os dados são lidos na fonte de dados, processados e armazenados na base de dados ou ficheiro de destino; contudo, não é capaz de executar tarefas de forma interativa ou recursiva. Como tal, caracteriza-se por executar uma tarefa de cada vez e sequencialmente) ou *streaming* (neste processamento, os dados estão continuamente a ser processados, obtendo-se resultados à medida que novos dados surgem).

Os autores Di Tria et al. (2014) sugerem que as metodologias de *Big Data Warehouse* devem priorizar novos modelos lógicos, como os que são usados pelas bases de dados *Not Only SQL* (NoSQL), garantindo assim a escabilidade, flexibilidade e um alto desempenho do sistema. Para tal, são utilizadas técnicas e tecnologias *Big Data* que têm como finalidade responder às difíceis e complexas cargas de trabalho analíticas (por exemplo, análises em tempo útil, *ad hoc querying*, visualização dos dados, *data mining*, simulações) (C. Costa & Santos, 2018).

Atualmente, vários trabalhos na área evidenciam que o *design* de *Big Data Warehouses* tanto se deve focar numa camada física (infraestruturas) como numa camada lógica (modelos de dados e compatibilidade entre os componentes). Em termos gerais, pode ser implementado seguindo duas estratégias; "*lift and shift*" consiste na capacidade de gerir os tradicionais *Data Warehouses* recorrendo

a tecnologias de *Big Data* (por exemplo, Hadoop e NoSQL); "*rip and replace*" consiste em substituir totalmente os tradicionais *Data Warehouses* por tecnologias de *Big Data*.

Contudo, é necessária uma avaliação dos modelos e métodos de *design* e arquiteturas de *Big Data Warehousing*, pois, atualmente, os conhecimentos e práticas carecem de trabalhos de cariz científico (Costa & Santos, 2018; Russom, 2014; Russom 2016).

## 2.3 Big Data Modelling

As organizações estão a recolher mais dados com diferentes tipos de formatos e velocidades. Quando os mesmos são utilizados e analisados corretamente, apresentam um impacto seminal, dado permitirem que as organizações se adaptem à estratégia de negócio que vá de encontro aos melhores resultados. Atualmente, o volume e a estrutura dos dados são as características que acabam por desafiar a capacidade de processamento dos tradicionais *Data Warehouses*. Os dados já não são centralizados em sistemas OLTP das organizações, visto que, nos dias de hoje, são altamente distribuídos com diferentes estruturas e apresentam uma taxa de crescimento exponencial. Desta forma, os *Big Data Warehouses* diferem substancialmente dos *Data Warehouses* na medida em que as estruturas devem ser baseadas em novos modelos lógicos mais flexíveis do que os modelos relacionais (E. Costa, Costa, & Santos, 2017; Krishnan, 2013).

O *Big Data* é um conceito emergente tanto a nível científico como técnico, embora existam alguns esforços de padronizar construções e componentes lógicos para sistemas de *Big Data (NIST Big Data Public Working Group Definitions and Taxonomies Subgroup*, 2015). Do mesmo modo, o conceito de *Big Data Warehouse* é recente e evidencia ambiguidade e falta de padrões de abordagem. Desta forma, os autores C. Costa & Santos (2018) definiram características e padrões de abordagem para colmatarem a lacuna científica. As mesmas serão apresentadas nos pontos que se seguem:

- *Massively Parallel Processing* (MPP);
- Análises mistas e complexas (*queryng ad-hoc, data mining, text mining*);
- Armazenamento flexível de forma a suportar dados de várias fontes de dados;
- Operações em tempo útil (processamento em *streaming*, baixa latência e atualizações com significativa frequência);
- Alto desempenho, com respostas em tempo útil;
- Escalabilidade para armazenar dados, utilizadores e análises;

- Uso de *commodity hardware* para reduzir custos;
- Interoperabilidade de múltiplas tecnologias.

O conceito de *Big Data Modelling* é recente, pelo que os artigos científicos referentes ao mesmo são escassos e desprovidos de coesão entre si. Assim, ainda não existe uma abordagem estruturada capaz de descrever o design e a implementação de sistemas de *Big Data Warehousing*. Consequentemente, e porque a compreensão deste conceito é fundamental no contexto desta dissertação, será efetuado um enquadramento do mesmo. Inicialmente, serão apresentados artigos com base no conceito de *Big Data Modelling*; depois, e uma vez que há poucos artigos científicos, surgiu a necessidade de complementar o trabalho com uma nova pesquisa sobre *Data Warehouse Modelling*.

### 2.3.1 Abordagem de Data Warehouse Modelling

*Data Warehouse* é uma tecnologia de soluções integradas e baseadas na gestão e implementação de dados que, ao longo dos anos, se tornou cada vez mais complexa. Como tal, e por vários anos, o conceito de *Data Warehouse* foi alvo de várias pesquisas e trabalhos científicos com o objetivo de contribuir com novos *Data Warehouse Modelling,* a fim de acompanharem o rápido crescimento de tecnologias, comunicações e informações (Zhou & Xiao, 2009).

A finalidade de um sistema de *Data Warehouse* passa pelo armazenamento de dados relativamente estáticos, ou seja, dados que não mudam regularmente, definidos por Inmon (2005) como "*time-invariant*", termo que se reflete na estrutura principal do *Data Warehouse Modelling,* estrutura em estrela, que consiste em uma ou mais tabelas de dimensões ligadas à(s) tabela(s) de facto através de chaves estrangeiras (Nikolov, 2007).

O conceito de modelação de dados em sistemas de *Data Warehouse* é baseado na modelação multidimensional e em ferramentas OLAP que acedem diretamente às estruturas multidimensionais, permitindo, desta forma, suportar a tomada de decisão. Assim, a qualidade do modelo de dados tem influência em todo o *Data Warehouse*. Os esquemas multidimensionais organizam os dados em factos e dimensões. Os factos contêm medidas de um processo de negócio (por exemplo, vendas e entregas), enquanto as dimensões (por exemplo, produto, cliente e tempo) representam o contexto de análise de um facto (Luján-Mora, Trujillo, & Song, 2006; Šuman, Jakupović, & Kuljanac, 2016). A modelação multidimensional é amplamente utilizada em ambientes tradicionais de *business intelligence* e diversos *Data Warehouses* estão disponíveis para suportar o processo de apoio à tomada de decisão (Maribel Yasmina Santos, Martinho, & Costa, 2017).

Os autores He, Chen, Meng, & Liu (2011) sugerem que a modelação conceptual multidimensional é o *core* do *Data Warehouse* e que o método de modelação conceptual multidimensional nem sempre se foca apenas em garantir a qualidade do *Data Warehouse*, mas também na melhoria da tomada de decisão. O estudo científico dos autores centra-se num método ontológico de modelação conceptual para *Data Warehouses*, representado pelo sistema BWW (*The Bunge-Wand-Weber*).

Os tradicionais *Data Warehouses* foram concebidos, segundo os autores (F. Di Tria et al., 2014a; Francesco Di Tria, Lefons, & Tangorra, 2014b), seguindo, habitualmente, uma das abordagens apresentadas nos seguintes pontos:

- *Data-driven* (orientada por dados) – esta abordagem é orientada aos dados; preocupa-se com a utilização dos mesmos, sendo criado um modelo multidimensional baseado nas fontes de dados disponíveis. Nesta abordagem, a interação com quem toma decisões nas organizações é minimizada.
- *Requirement-driven* (orientada por requisitos) – esta abordagem também é conhecida como *demand-driven;* visa definir esquemas multidimensionais baseados nos objetivos de negócio dos responsáveis pela tomada de decisão.

### 2.3.2 Data Modelling em ambientes de Big Data

Os autores Kimball & Ross (2013) sugerem que, apesar do *Data Modelling* se focar nos *Data Warehouses* tradicionais, é relevante que sejam apresentadas práticas de planear um *Data Warehouse* em ambientes de *Big Data*, como as que se seguem:

- Considerar análises complexas e não só relatórios ou queries *ad hoc;* evitar, tanto quanto possível, as mudanças tecnológicas; e promover o uso de *sandbox results,* porque permitem aos *data scientists* um trabalho mais flexível.
- Planear *data highways,* ou seja, diferentes caches com diferentes requisitos de latência; extrair dados, mesmo que não sejam estruturados; e implementar mecanismos de *streaming*.
- Abordagem de um problema de modelação, como o das dimensões e factos, e a integração de dados estruturados e não estruturados.

Segundo os autores Francesco Di Tria et al. (2018) a metodologia a aplicar em contextos de Big Data deve ser híbrida, automática, incremental, baseada em ontologias e modelos não relacionais:

- Híbrida – esta abordagem emergiu de abordagens orientadas por dados e orientadas por requisitos, permitindo a recolha das potencialidades de cada uma. É uma abordagem mais complexa, dada a necessidade de integrar as duas abordagens; desta forma, integra e considera, em simultâneo, as fontes de dados e os requisitos de negócio.
- Automática – a fim de tornar a sua conceção mais rápida, mesmo em contexto de várias fontes de dados;
- Incremental – com base em abordagens ágeis;
- Ontologias – esta abordagem pode ser utilizada para resolver possíveis incoerências sintáticas e semânticas na integração de fontes de dados, mesmo que não sejam estruturadas;
- Modelos não relacionais – permitem um *design* mais flexível que o dos tradicionais modelos relacionais, visto apresentarem dados desnormalizados e sem esquema.

O estudo experimental dos autores E. Costa, Costa, & Santos (2017) consistiu na avaliação e discussão do modelo de dados mais adequado, na análise dos impactos dos diferentes modelos de dados, e a organização das estratégias de *Data Warehouses* para Hadoop (Hive) em ambientes de *Big Data*. Desta forma, os resultados do estudo vão ajudar no planeamento de sistemas *Big Data Warehouses* e contribuir na determinação de aspetos metodológicos para o *Data Modelling* nas organizações. O trabalho de investigação dos autores conclui que o Hive, em sistemas de *Data Warehouses* construídos com esquemas em estrela e em *Data Warehouses* construídos com tabelas desnormalizadas, beneficia com a utilização de uma estratégia de *Data Modelling* desnormalizada, uma vez que os *benchmarks* mostraram significativas vantagens em todos os fatores de desempenho, em relação à estratégia de modelação dimensional. Segundo os autores, apesar de ser viável implementar *Data Warehouses* eficientes, em Hive, utilizando esquemas em estrela, o mesmo pode não ser o padrão de *design* mais eficiente, visto que as tabelas desnormalizadas mostram melhor desempenho do que os esquemas em estrela ao utilizar sistemas específicos de *Multidimensional Data Model* (MDM) de *Data Warehouse*. Os autores reforçam ainda que – apesar dos inúmeros esforços aplicados em sistemas SQL-on-Hadoop, de forma a suportar queries multidimensionais – é notório que *Data Warehouses* baseados em Hive ainda favorecem estruturas desnormalizadas que não dependem de operações *joins* para responder a queries OLAP. Esta abordagem evita o custo de realizar operações de *join* em ambientes de *Big Data*.

O estudo científico desenvolvido pelos autores Santos, Martinho, & Costa (2017) propõe um conjunto de regras de modelação de *Big Data Warehouses* que permitem a transformação de MDM em tabelas Hive, integrando dados em diferentes níveis de detalhe. Além disso, para implementar as tabelas

selecionadas, os autores também conceberam uma arquitetura tecnológica que facilita a migração de um ambiente de *business intelligence* (BI) tradicional para um ambiente de *Big Data Analytics*. As regras, definidas pelos autores, podem ser utilizadas para transformar MDM *Analytical Data Models* em ambientes de *Big Data* e passam por identificar tabelas primárias de dados, e passam por gerar tabelas desnormalizadas. Posteriormente, as tabelas com dados derivados agregados podem ser identificadas, reduzindo, assim, o número de *data records*. Posteriormente, estas tabelas de dados derivados podem ser implementadas e disponibilizar capacidades de *querying ad hoc*.

Os autores C. Costa & Santos (2018) apresentaram o resultado de um estudo científico que apoia a análise e compreensão de vários padrões de *design* e tendências em *Big Data Warehousing* com o intuito de promover futuras pesquisas e dar suporte à escolha de *design* para implementações de *Big Data Warehouse.* Foram discutidos os *trade-offs* entre esquemas em estrela e tabelas desnormalizadas, recorrendo a pequenas e grandes dimensões, a utilidade e eficiência de estruturas *nested* em *Big Data Warehouse*, assim como várias considerações relativas ao desempenho do armazenamento em *streaming*. Os principais resultados obtidos pelos autores são os seguintes:

- Tabelas desnormalizadas, tendem a superar esquemas em estrela, ambos testados com dimensões pequenas e grandes; contudo, em certos contextos, os esquemas em estrela mostraram vantagens;
- Estruturas *nested,* trazem vários benefícios para *Big Data Warehouses*; apesar disso, não são eficientes quando se utilizam atributos nas estruturas *nested* para aplicar filtros ou funções de agregação;
- Hive, tende a superar o Cassandra como um sistema de armazenamento em *streaming*; contudo, depois de um determinado período, o número de pequenos ficheiros gerados acabam por sobrecarregar o HDFS que, quando executa operações de metadados, estes acabam por penalizar o desempenho.
- Sistemas interativos SQL-on-Hadoop, como o Presto, podem combinar de forma eficiente tabelas de factos em *streaming*, armazenadas em bases de dados NoSQL (Cassandra), com as dimensões armazenadas em *batch* (Hive), obtendo um desempenho semelhante às tabelas desnormalizadas;
- Periodicamente, os dados armazenados em *streaming* devem ser movidos para o armazenamento em *batch* de um *Big Data Warehouse*, a fim de manter a interatividade do sistema;

- Em ambientes *multi-tent* deve ser tido em consideração o *trade-off* entre os recursos do cluster e os intervalos entre os micro *batch* e *streaming*.

Os autores Santos et al. (2019), motivados pela inexistência de uma abordagem estruturada que descrevesse os padrões de *design* e a implementação de *Big Data Warehouses*, desenvolveram, entre outros, um método de *data modelling*, para projetar estruturas de dados armazenados em sistema de *Big Data Warehouse*. De forma a compreender a proposta de *Data Modelling* do autor os conceitos como *analytical objects* (que inclui *descriptive* e *analytical attributes*), *complementary analytical objects, materialized objects, granularity keys, atomic values, collections, partition keys* e *bucketing/clustering keys* serão apresentados nos pontos abaixo.

*Analytical Objects* – são definidos como assuntos isolados com propósitos analíticos. Portanto, os mesmos podem conter, no seus *descriptive attributes*, os atributos que correspondem à *granularity key* de outro objeto. Os *analytical objects* são altamente desnormalizados e possuem estruturas autónomas capazes de responder a *queries* sem a necessidade constante de fazer *join* com as dimensões e a tabela de factos. Uma organização pode identificar os objetos analíticos examinando as fontes de dados disponíveis (*data-driven*) ou examinando os objetivos atuais de negócio, começando, posteriormente, a recolher os dados que correspondam aos objetivos analíticos (*requirements-driven*).

I. *Descriptive Attributes* – os atributos descritivos fornecem uma forma de interpretar atributos analíticos através de diferentes perspetivas, por exemplo, designação de produtos ou clientes.

II. *Analytical Attributes* – os atributos analíticos, em contraste com os atributos descritivos, fornecem valores numéricos (às vezes incorporados em dados complexos/estruturas de dados *nested* que contêm dados de texto) podendo ser analisados, recorrendo ao uso de diferentes atributos descritivos, e agrupados ou filtrados por estes atributos. Podem incluir atributos factuais e preditivos.

O *record* de um *analytical objects* armazena todos os valores que correspondam a um evento associado ao mesmo, tendo em consideração diferentes atributos. Desta forma, os atributos descritivos e analíticos podem conter *atomic values* ou *collections.*

I. *Atomic values* – armazenam os dados de uma forma simples, por exemplo, como inteiro, *float*, *double, string* ou *varchar.*

II. *Collections* – armazenam estruturas de dados mais complexos, por exemplo, *arrays, maps* ou objetos JSON.

A *granularity key* está associada aos *analytical objects* e permite identificar o nível de detalhe dos dados que serão armazenados em cada *record.* A *granularity key* de um objeto é definida por um ou mais atributos descritivos que apenas identificam o *record*.

*Complementary Analytical Objects* – são considerados objetos analíticos complementares, se a *granularity key* estiver incluída num *analytical object.*

A diferença entre a *partition key* e *bucketing/clustering key:*

I. *Partition key* – os atributos que são utilizados para particionar o *analytical object* constituem a *partition key.*

II. *Bucketing/clustering key* – garante que o intervalo de *record* seja armazenado no mesmo grupo ou classificado.

*Materialized Objects* – podem ser armazenados em *batch* ou em *streaming,* sendo capaz de armazenar o resultado do tempo utilizado por *queries,* aumentado o desempenho do sistema de *Big Data Warehousing.* Os *materialized objects* podem ser análogos aos cubos OLAP em ambientes tradicionais de *Data Warehouse*, contendo dados pré-agregados destinados a serem consumidos de maneira mais rápida e eficiente.

Relativamente ao processamento do *Data Modelling*, há três tipos de processamento: o *batch, interactive* e *streaming*. Segundo os autores, os mesmos são definidos como:

I. *Batch processing* – este tipo de processamento envolve um intervalo de latência compreendido entre minutos e horas.

II. *Interactive processing* – este tipo de processamento é usado para fornecer à execução de *queries* intervalos de tempo compreendidos entre os milissegundos e algumas dezenas de segundos, dependendo das infraestruturas e do volume dos dados.

III. *Stream processing* – este tipo de processamento é relativo apenas ao processo de dados CPE, isto é, os utilizadores não têm acesso instantâneo aos dados *streamed.* Em vez disso, os dados *streaming* são armazenados num componente de *streaming*, ficando, de imediato, disponíveis para o utilizador.

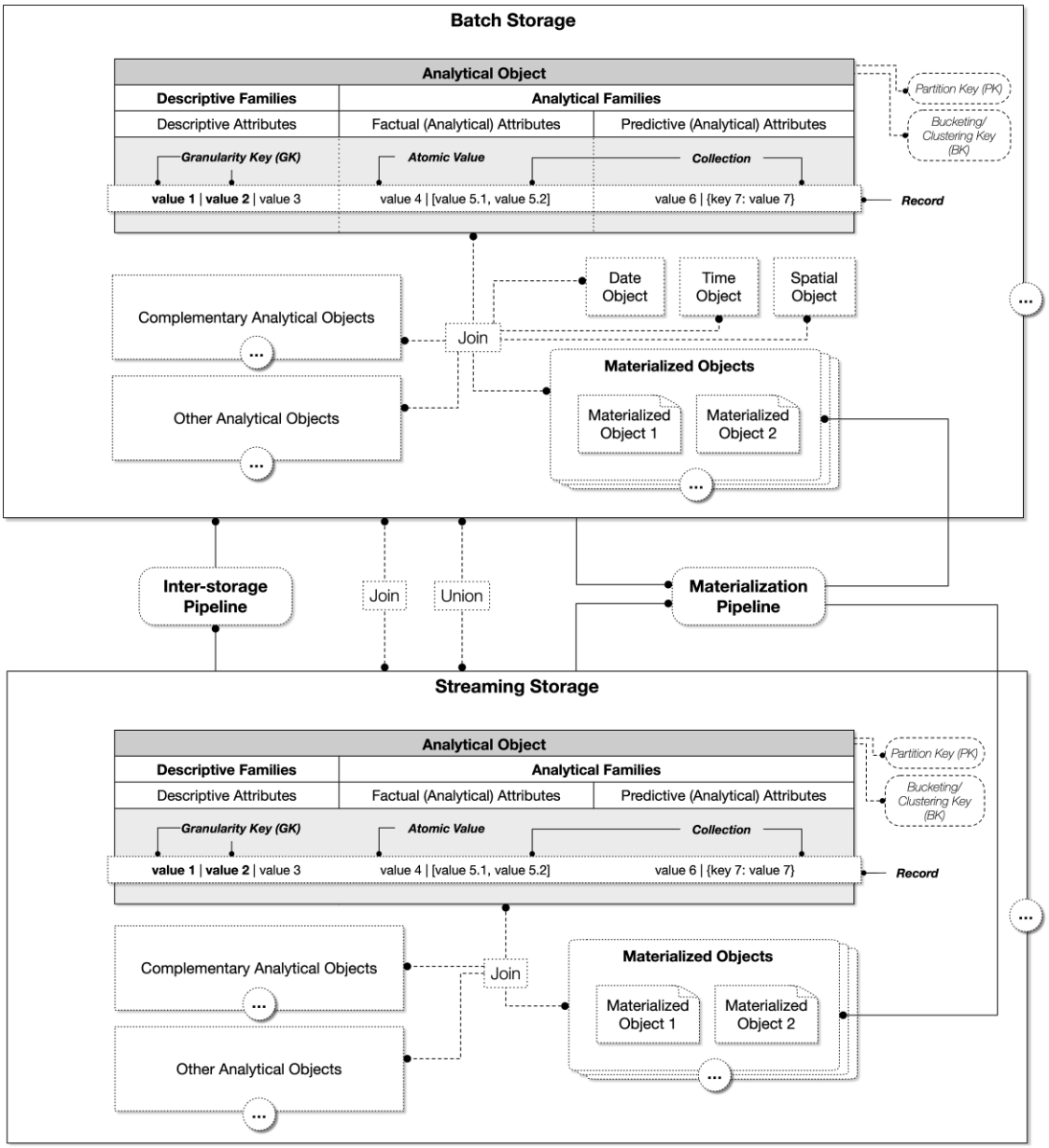



Figura 4 - Modelo geral de dados [retirado de (M. Y. Santos & Costa, 2019)].

No seguimento do estudo científico desenvolvido (M. Y. Santos & Costa, 2019), apresentam-se, abaixo, as melhores práticas de *Data Modelling* a aplicar em modelos de dados de *Big Data Warehouse* de forma a clarificar questões relacionadas com o *design* e implementação, assim como as vantagens e desvantagens do *Data Modelling*.

- Melhores Práticas – as melhores práticas a serem aplicadas a um modelo de dados *Big Data Warehouse*, com a finalidade de esclarecerem questões relacionadas com a conceção e implementação de *use null values, spatial and temporal attributes* e *modelling of records as immutable events*, são as seguintes:

I. *Use Null Values* – a utilização de valores nulos em sistemas de *Big Data Warehouse* não é proibida e, nalgumas situações, é até oportuna. É o caso dos atributos analíticos em que é recomendável a utilização de *null,* de forma a indicar ausência de valores, visto que as *queries*, sistemas OLAP e tecnologias de visualização, ignoram os valores nulos, quando agregam os dados. Por outro lado, no caso dos atributos descritivos (por exemplo, dados de texto), são mais apropriados para a utilização de "Desconhecido" ou "Não Aplicável", salvo exceções.

II. *Spatial and Temporal attributes* – os *date objects* e *time objects*, apresentados na Figura *4*, incluem vários atributos que complementam os objetos analíticos armazenados num sistema de *Big Data Warehouse.* A forma mais adequada seria o uso de um padrão para as datas (por exemplo, "yyyy-mm-dd") e o uso de um padrão de horas (por exemplo, "hh:mm"); desta forma, todos os objetos analíticos permitem o *join* com os objetos data e hora. Quanto ao uso de *spatial objects*, mostram-se úteis para criar padrões de atributos espaciais por meio de objetos analíticos do sistema de *Big Data Warehouse*.

III. *Immutable vs Mutable* – um *immutable event* evita atualizações nos dados existentes, de forma a simplificar sistemas de *Big Data Warehousing.* Em contraste, os *mutable events* permitem a atualização dos dados.

- Vantagens – a abordagem de modelação é baseada em desnormalização, sendo um passo crucial no alcance de um armazenamento flexível num sistema de *Big Data Warehouse*. Quando comparada com as abordagens de modelação de dados relacionais em *Data Warehouses* tradicionais, o autor identifica as vantagens apresentadas nos pontos a seguir:

  I. Garante um melhor desempenho na execução de queries;

  II. Fornece um modelo desnormalizado flexível;

  III. Concentra-se, preferencialmente, na modelação de objetos analíticos, por exemplo, um conjunto de *immutable events*;

  IV. Evita problemas relacionados com a modelação tradicional de dados ETL e considera vários tipos de dimensões que, em contexto de *Big Data*, são desnecessários. Desta forma, economizam espaço de armazenamento e obtêm modelos de dados menos redundantes.

V. Destaca relevantes estruturas *nested* em determinados modelos e aplicações de dados do sistema de *Big Data Warehouse*.

- Desvantagens – o método de *data modelling* proposto por Santos et al. (2019)tem características que podem ser consideradas como desvantagens, quando comparadas com os métodos acima mencionados para projetar *Data Warehouses*, que incluem:

  I. O tamanho total de *Big Data Warehouses* (tipicamente, aqueles cujas fontes de dados são altamente dimensionais por reutilizarem dimensões com muita frequência) podem aumentar drasticamente devido à desnormalização dos dados. Por essa razão, a abordagem introduz o conceito de objetos de data/hora, objetos espaciais, objetos analíticos complementares e famílias descritivas.

  II. Se a fonte de dados associada ao objeto analítico for assente numa base de dados relacional, as cargas de trabalho de CPE para esse objeto podem precisar de várias operações de *join*, sendo executadas na origem (como *queries* SQL). No entanto, em contextos *Big Data*, consideráveis fontes de dados não são relacionais (dados de sensores, bases de dados NoSQL, ficheiros XML e JSON), tornando o método de *Data Modelling* mais simples em sistemas de *Big Data Warehouse*.

De forma a sintetizar e representar esquematicamente todos os conceitos da proposta de *Data Modelling* para sistemas de *Big Data Warehouse*, a Figura *4* compreende os mesmos e permite mostrar, de modo mais intuitivo, o *design* do *Data Modelling*. Como tal, o método proposto pelos autores Santos et al. (2019) de modelação de dados sugere uma forma de estruturar sistemas de *Big Data Warehouse* em *batch* e *streaming,* utilizando o mesmo construtor, independentemente da tecnologia subjacente que suporta o armazenamento do sistema, fornecendo uma camada de abstração na qual os profissionais podem confiar para modelar *Big Data Warehouses* suportados por HDFS/Hive, bases de dados NoSQL/NewSQL, Kudu, Druid, entre outras tecnologias.

No seguimento da apresentação do método do *Data Modelling* proposto no estudo científico dos autores Santos et al. (2019), o exemplo apresentado na Figura *5* foi escolhido com o objetivo de demonstrar a aplicação, num possível contexto do mundo real, os conceitos descritos e representados na Figura *4*.

Após aplicar o método de *Data Modelling*, num exemplo que utiliza uma base de dados relacional OLTP de uma empresa fictícia que fabrica e vende bicicletas, o resultado alcançado pelos autores, após a aplicação do mesmo, pode ser visto na Figura *5*. Contém 7 *analytical objects* (*"employee history",*

*"sales line"*, *"product review"*, *"product vendor history"*, *"purchase line"*, *"product inventory"* e *"work order"*), 3 *complementary analytical objects* (*"product"*, *"vendor"*, e *"special offer"*), 1 *date object*, 1 *time object*, e 2 *spatial objects* (*"city"* e *"territory"*). A proposta de abordagem do exemplo, que usa a base de dados *Adventure Works*, representa a maioria dos conceitos do método acima descritos.

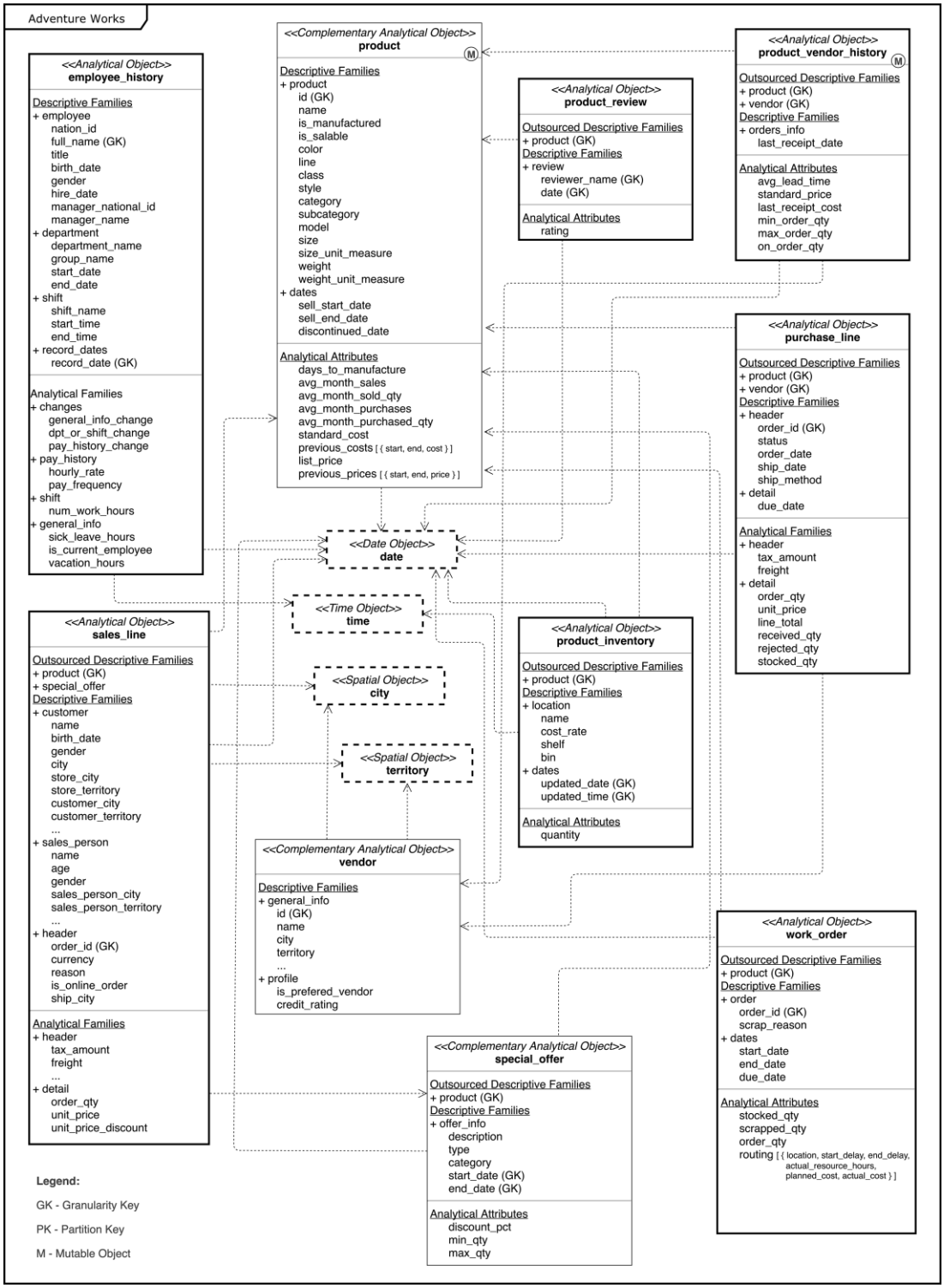



Figura 5 - Modelo de dados BDW [retirado de (M. Y. Santos & Costa, 2019)].

Concluindo, esta é uma área científica recente; como tal, os artigos científicos relacionados com o *Data Modelling* em sistemas de *Big Data Warehouse,* assim como a otimização do processamento, ainda são escassos. Segundo os autores Santos et al. (2019), nos últimos anos esta lacuna científica e técnica motivou vários autores a desenvolverem estudos científicos com o intuito de colmatarem a falta de estruturas e abordagens de modelos capazes de descrever como conceber e implementar sistemas de *Big Data Warehouses* por meio de avaliações adequadas dos modelos (representação de componentes lógicas e tecnológicas, *data flows* e estruturas de dados), métodos e instanciações (casos de demonstração recorrendo a protótipos e *benchmarks*).

# 3. Enquadramento Tecnológico

O enquadramento tecnológico inerente a este trabalho visa caraterizar algumas ferramentas tecnológicas fundamentais para o entendimento dos ambientes *Big Data*. Estas são soluções que tipicamente compõe sistemas para contextos de *Big Data* e referidas em respostas a desafios de investigação e desenvolvimento de *Big Data Warehousing*. No âmbito desta dissertação, considera-se como essencial o entendimento de três conceitos associados tipicamente ao popular ecossistema *Hadoop*.

## 3.1 Hadoop Distributed File System (HDFS)

O Hadoop tem uma estrutura *open source* (significa que é uma ferramenta de livre acesso, sem qualquer custo de licença) e surgiu como uma opção para resolver problemas relacionados com o processamento de *Big Data* em plataformas de baixo custo, sendo possível a utilização de *commodity hardware*. A ferramenta utiliza o paradigma de computação *MapReduce* para analisar e transformar conjuntos significativos de dados e gerir todos os *inputs* e *outputs* de dados armazenados num cluster; o Hadoop tem como componente chave o Hadoop Distributed File System (HDFS) (Capriolo, Wampler, & Rutherglen, 2012; Holmes, 2012; Shvachko, Kuang, Radia, & Chansler, 2010).

No ecossistema Hadoop estão presentes várias ferramentas de análise, consulta e *data mining,* incluindo algoritmos de *machine learning.* Este ecossistema apresentou resultados satisfatórios quando organizações que apresentam um volume significativo de dados, como a Amazon, Facebook, Yahoo!, AOL e New York Times, o utilizaram (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009; Gupta, 2015; Luo, Luo, Guan, & Zhou, 2013).

O HDFS é um sistema de ficheiros distribuído, capaz de lidar com altas taxas de transferência, leitura e escrita, para significativos volumes de dados que podem atingir os *petabytes* em *clusters* de milhares de nós (Holmes, 2012). O autor Krishnan (2013) sugere que o HDFS se distingue dos já existentes pela sua tolerância a falhas, pela grande quantidade de dados que suporta e por poder ser utilizado com *hardware* de baixo custo. Além disso, caracteriza-se pela sua redundância, pelo facto de armazenar cópias dos dados em nós distintos do *cluster*. Este facto aumenta a sua tolerância a falhas, mas também pode permitir melhor desempenho na apresentação de resultados de queries ao sistema.

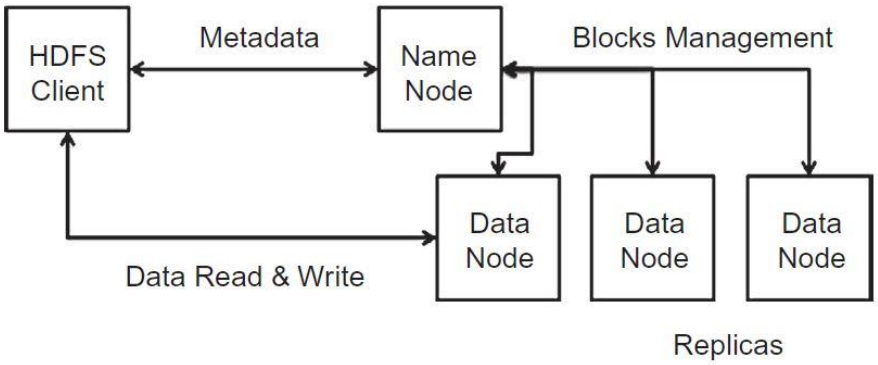



Figura 6 - Arquitetura do HDFS [retirado de (Krishnan, 2013)].

A arquitetura HDFS evolui da arquitetura *Nutch Distributed File System* (NDFS), que é baseada na arquitetura Google File System (GFS). A *Figura 6* representa uma arquitetura básica de um sistema HDFS; cada bloco de dados é replicado várias vezes, de acordo com o parâmetro *replication factor* (definido por defeito como três) de forma a prevenir a perda de dados ou a aumentar a capacidade de processamento dos mesmos no caso de surgir uma falha no disco ou no servidor. A arquitetura HDFS é constituída pelos *NameNodes (master node)* e *DataNodes (slave nodes)*. Os *NameNodes* são responsáveis pela gestão do sistema de metadados, pelos sistemas de ficheiros e pela gestão do acesso dos clientes. Os *DataNodes* são responsáveis pela gestão e armazenamento dos blocos de dados para cada ficheiro (Krishnan, 2013).

## 3.2 Hive

O Hive é a ferramenta, *open source,* que melhor se adequa para a implementação de um *Data Warehouse* em contextos de Big Data. Esta ferramenta pertence ao ecossistema Hadoop e foi desenvolvida pelo Facebook, em janeiro de 2007, com o objetivo de consultar e gerir grandes volumes de dados armazenados em ambientes distribuídos. O Hive organiza os dados em tabelas, podendo estas estar particionadas por partições e/ou *buckets* (Cassavia, Dicosta, Masciari, & Saccà, 2014; Maribel Yasmina Santos & Costa, 2016; Thusoo et al., 2010).

Os modelos de dados do Hive, segundo (Du, 2018; Santos & Costa, 2016), fornecem uma estrutura em tabelas de alto nível sobre o HDFS e sustentam as estruturas de dados descritas a seguir indicados:

- Tabelas – apresentam similaridade às tabelas das bases de dados relacionais (estruturas comuns, com colunas e linhas). As tabelas podem ser filtradas, projetadas e unidas e os dados de cada tabela são armazenados numa diretoria HDFS.

- Partições – cada tabela pode ter uma ou mais partições que acabam por determinar a distribuição dos dados. As mesmas permitem dividir os dados horizontalmente e acelerar o processamento de *queries*.
- *Buckets* – correspondem a segmentos de ficheiros no HDFS, podendo ser aplicados apenas a um único atributo, e ajudam a organizar os dados em cada partição da sua divisão por vários ficheiros.

O Hive também suporta o conceito de tabelas externas, o que significa que uma tabela pode ser criada em arquivos ou diretorias do HDFS já existentes, fornecendo assim o local adequado para a DDL (*Data Definition Language*) da criação da tabela. Este suporta também a criação de tabelas temporárias que são automaticamente eliminadas quando terminada a sessão (Du, 2018).

A linguagem usada pelo Hive para desenvolver *queries* é o HiveQL (*Hive Query Language*), a mesma tem uma sintaxe semelhante à linguagem SQL (*Structure Query Language*). O HiveQL tem a singularidade de inspecionar várias tabelas em simultâneo, possibilitando aos utilizadores executarem múltiplas consultas nos mesmos dados de entrada recorrendo a uma única *query* HiveQL o que faz aumentar o desempenho dessas consultas (Du, 2018). As ferramentas de *querying* são fundamentais em sistemas de *Big Data Warehousing*, visto promoverem a interação das interfaces SQL para queries de dados armazenados em *batch* e *streaming* (Costa, C., 2018).

## 3.3 NoSQL

As bases de dados relacionais, baseiam-se na linguagem SQL e em propriedades transacionais que garantem propriedades ACID: atomicidade, consistência, isolamento e durabilidade; as mesmas têm sido utilizadas em estruturas de dados normalizadas. Contudo, a tecnologia tradicional já não suporta o processamento e as análises das significativas quantidades de dados que são produzidas atualmente (Alekseev et al., 2016). Como é o caso do Facebook que já foi detentor do Cassandra, a base de dados NoSQL mais usada. Contudo, o NoSQL inclui implementações SimpleDB, Google BigTable, Hadoop, o MapReduce, entre outros (Krishnan, 2013).

Com o desenvolvimento da Internet e da computação em *cloud*, surge uma nova abordagem de base de dados NoSQL, capaz de suportar o armazenamento de dados e o processamento de *Big Data* de forma eficiente, permitindo melhores resultados nas consultas (Cassavia et al., 2014).

Segundo o autor Cassavia et al., (2014), e segundo o seu modelo de dados, as bases de dados NoSQL podem ser categorizadas em quatro tipos:

- Bases de Dados *Key-Value* - indexam valores identificados por chaves;
- Bases de Dados *Column-Oriented* - contém uma coluna expansível com dados relacionais;
- Armazenamento Orientado a Documentos - permite que o utilizador adicionar os campos que pretender, seja qual for a dimensão;
- Bases de Dados Orientadas a Grafos - consistem na criação de nós.

Para armazenar e gerir dados não-estruturados ou dados não relacionais, as bases de dados NoSQL admitem abordagens específicas. Primeiro, o armazenamento e a gestão de dados são separados em duas partes. No armazenamento, as bases de dados NoSQL concentram-se na escabilidade do armazenamento de dados e com alto desempenho. Na gestão de dados, estas têm um mecanismo de acesso de baixo nível que permitem implementar tarefas na *application layer.* Desta forma, os sistemas NoSQL são flexíveis na modelação de dados e desenvolvem facilmente atualizações nas aplicações.

A evolução, o processamento e análise do volume dos dados tornaram as propriedades ACID insuficientes, sendo a introdução do teorema CAP (*Consistency, Availability, Partition Tolerance*) imprescindível. O teorema CAP afirma que um sistema de dados distribuídos garante, no máximo e ao mesmo tempo, duas das três propriedades a seguir mencionadas (Gessert, Wingerath, Friedrich, & Ritter, 2017; Krishnan, 2013):

- Consistência (C) – as execuções desta característica estão relacionadas com a atomicidade e isolamento. Isto significa que os utilizadores poderão ter, em qualquer momento, a mesma visão sobre os dados.
- Disponibilidade (A) – qualquer pedido feito por um nó do sistema resulta sempre numa resposta, mesmo que existam falhas na rede.
- Tolerância à Partição (P) – o sistema mantém as garantias de consistência; assim e, independentemente da disponibilidade ou da perda de dados de comunicação de uma partição, o sistema irá funcionar.

Em suma, o HDFS e o Hive estão orientados para, em tempo útil, terem um acesso sequencial (não acontece o mesmo com o acesso aleatório); isto traduz-se num aumento da dimensão do armazenamento em *batch,* o que também vai aumentar o intervalo entre a recolha dos dados e sua disponibilidade no sistema de *Big Data Warehousing*. Tendo isto em conta, não será menos conveniente optar pelas bases de dados NoSQL, pois as mesmas são capazes de resolver problemas em pequenos ficheiros Hadoop, embora provoquem mais complexidade nas infraestruturas e o tempo de execução das *queries* seja mais lento (Cattell, 2011; Mackey, Sehrish, & Wang, 2009).

# 4. Método para a Modelação de Big Data Warehouses

No presente capítulo será apresentado o desenvolvimento do artefacto do tema da dissertação, que consiste na proposta de um método para a modelação de *Big data Warehouses* constituído por um conjunto de sete regras computacionais em vista a semiautomatização do método de *Data Modelling*, proposto no estudo científico de Santos et al. (2019) descrito no capítulo 2, subsecção 2.3.2 deste documento. Deste modo, foi definida uma proposta de abordagem de *Data Modelling* para sistemas de *Big Data Warehousing*, pelo que o método foi desenvolvido com o objetivo de agilizar a proposta dos padrões de *design* desenvolvidos pelos autores.

Em primeiro lugar, e de forma a consolidar e explorar o método de *Data Modelling* proposto pelos autores Santos et al. (2019), foi efetuado um exercício com os mesmos exemplos por eles selecionado, a fim de facilitar a transição entre a teoria e a prática de possíveis aplicações do método. O resultado deste trabalho inicial foi comparado com o dos autores, verificando-se que a aplicação do método nos vários exemplos se traduziu numa clara compreensão do *Data Modelling*. Consequentemente, a repetição da aplicação do mesmo permitiu que fossem encontrando padrões dos objetos relativos aos relacionamentos e cardinalidade, culminando no método constituído por sete regras.

Mais tarde, foram desenhados os diagramas de entidades relacionamentos (ER) de todos os exemplos demonstrados, e iniciado o estudo de padrões e de casos pontuais que podiam criar uma regra. Daqui até ao resultado final do método, o desenvolvimento sofreu constantes mudanças, para que as ideias de possíveis regras fossem testadas em diferentes contextos e ordem, com o objetivo de comparar os resultados e de definir o melhor método. Este processo envolveu a conceção de regras que foram sendo descartadas ou refinadas, consoante os resultados que apresentavam após a sua aplicação, e permitiu definir igualmente a ordem pela qual cada uma era executada. A escolha de diagramas ER ficou a dever-se ao facto de os modelos de base de dados relacionais serem, atualmente os mais usados, segundo o *DB-Engines Ranking* (*DB-Engines Ranking*, 2019).

No contexto desta dissertação, o desenvolvimento do método centrou-se numa abordagem *top-down*, ou seja, o trabalho aqui apresentado refere-se, apenas, ao nível mais alto do *Data Modelling* sugerido pelos autores. Deste modo, centra-se essencialmente na identificação dos tipos de objetos, como é o caso dos *analytical object, complementary analytical object, date object, time object* e *spatial object*. Já o nível mais detalhado do método de *Data Modelling*, como é o exemplo das *descriptive families analytical families, outsourced descriptive families, materialized objects, granularity key, atomic value, collection, partition key e bucketing/clustering key*, poderá traduzir-se em possíveis trabalhos futuros.

## 4.1 Enquadramento Geral do Método

Esta secção apresenta o método, bem como a ordem pela qual as sete regras que o constituem o executam. O método aplica as regras continuamente, até verificar que o resultado final da aplicação dos passos das sete regras do último modelo é análogo ao resultado final do penúltimo modelo.

O método surge no seguimento da proposta de uma abordagem estruturada de *Data Modelling* para sistemas de *Big Data Warehouse* dos autores Santos et al. (2019). As regras do método são apresentadas nos pontos a seguir indicados:

R1. Proposta de Objetos do Tipo *Date, Time* e *Spatial.*

R2. Proposta de Objetos do Tipo *Analytical Objects*.

R3. Proposta de Objetos Integráveis.

R4. Proposta de Objetos com Relacionamentos Múltiplos.

R5. Proposta de Objetos Padronizáveis.

R6. Proposta de Objetos Autónomos.

R7. Proposta de Objetos do tipo *Complementary Analytical Objects*.

R1. Proposta de Objetos do Tipo *Date, Time* e *Spatial* - a primeira regra a ser executada propõe a verificação, criação e classificação de entidades[1] ou atributos *date/time objects* e de entidades *spatial objects*. Deste modo, R1 garante que todos os modelos que tenham entidades ou atributos do tipo *date* e/ou *time* sejam classificados. Segundo os autores Santos et al. (2019), incluir estes atributos nos *analytical objects* pode aumentar o espaço de armazenamento e, consequentemente, afetar a estabilidade e a performance de sistemas de *Big Data Warehousing*. A utilização de *spatial objects* é significativo para padronizar atributos do tipo *analytical object*, visto assegurarem, por exemplo, que uma cidade e um país tenham o mesmo significado em todo o *data model.*

R2. Proposta de Objetos do Tipo *Analytical Objects* - a segunda regra propõe a identificação e classificação de objetos como *analytical object,* sempre que verificar que os únicos relacionamentos que recebe é do tipo muitos.

[1] A partir deste ponto, os termos objeto e entidade são utilizados como sinónimos

R3. Proposta de Objetos Integráveis - a terceira regra propõe a desnormalização de objetos integráveis e surge de forma a verificar, em cada par de entidades a condição a seguir apresentada.

$$\sum FK\,(A\ +\ B) \geq\ 3\ AND\ (FK(A) =\ 1\ OR\ FK(B) =\ 1)$$

Isto é, se a soma das chaves estrangeiras do par de objetos, A-B, for maior ou igual a 3 e A ou B só podem ter uma chave estrangeira. Se a condição se verificar, o par é desnormalizado.

R4. Proposta de Objetos com Relacionamentos Múltiplos - a quarta regra do método, independentemente de como as objetos estão representados no modelo, é o típico caso de 'partir' um relacionamento do tipo M:N (Muitos-para-Muitos). Logo que um cenário destes é verificado, as entidades são desnormalizadas para a entidade que está a servir de ponte entre as mesmas.

R5. Proposta de Objetos Padronizáveis - a regra número cinco começa por verificar os objetos padronizáveis, isto é, entidades que só recebam relacionamentos do tipo um e tenham baixa cardinalidade; assim sendo, as que cumprirem as duas condições são classificadas. Logo que todas as entidades tenham sido percorridas, as que foram classificadas, e de forma a evitar erros na identificação de objetos padronizáveis, é necessitam de validação manual por parte do utilizador. O(s) objetos(s) selecionado(s) pelo utilizador será(ão) desnormalizado(s).

R6. Proposta de Objetos Autónomos - a quinta regra propõe a desnormalização de objetos autónomos, isto é, objetos isolados que existem exclusivamente para complementar outro objeto; deste modo, os mesmos recebem exclusivamente uma ligação de um. Se isto se verificar, a entidade é desnormalizada.

R7. Proposta de Objetos do tipo *Complementary Analytical Objects* – a sétima regra propõe a verificação e classificação de objetos do tipo *complementary analytical object;* para tal, a entidade tem de verificar uma das condições: receber, pelo menos, um relacionamento de um e pelo menos um relacionamento de muitos; ou entidades que recebam unicamente relacionamentos de um, mas verifiquem alta cardinalidade. Logo, se alguma das duas condições se verificar, a entidade é classificada como *complementary analytical object*.

A fase de desenvolvimento do método inclui fluxogramas para todas as regras, permitindo, assim, a representação esquemática de todo o processo lógico e dos passos inerentes a cada uma das mesmas, tornando a sua compreensão mais inteligível.

Nos passos de decisão dos sete fluxogramas apresentados nesta secção, a referência à palavra 'entidade' é sempre relativa à entidade que está a ser verificada. O mesmo acontece com a palavra

'atributo'. Esta, ao ser referida, significa a verificação de todos os atributos da entidade que está a ser analisada. Da mesma forma, nos passos de decisão, sempre que existe menção a 'relacionamentos do tipo muitos', é equivalente a ter relacionamentos do tipo M:1 (Muitos-para-Um) ou M:N. De forma similar, a referência a 'relacionamentos de um' é equivalente a ter relacionamentos do tipo 1:M (Um-para-Muitos) ou 1:1 (Um-para-Um).

A Figura *7* apresenta o modelo inicial do caso de demonstração descrito no TPC *Benchmark* H (TPC-H, 2018) que se foca em fornecer dados objetivos e relevantes para utilizadores do sector industrial. O caso TPC-H foi selecionado para demonstração da aplicação do método na fase de desenvolvimento por ser o mais simples e com o maior número de passos do método aplicado, relativamente aos casos utilizados na demonstração do método no capítulo 5.

Ao modelo inicial, ao qual será aplicado o método, seguir-se-á a demonstração nas figuras apresentadas (Figura *9, 11 , 14, 19 e 21*), o resultado da aplicação de cada uma das regras, no modelo. Com um objetivo similar, foram criadas tabelas (Tabela *1, 2, 3, 4* e *5*) que, a par das figuras referidas, permitem testar o processo inerente a cada regra.

Importa realçar ainda que os passos de cada fluxograma estão numerados, a fim de facilitar a aplicação passo-a-passo das regras; contudo, as tabelas da verificação formal das regras só contemplam os passos que implicam mudanças no modelo de demonstração TPC-H.

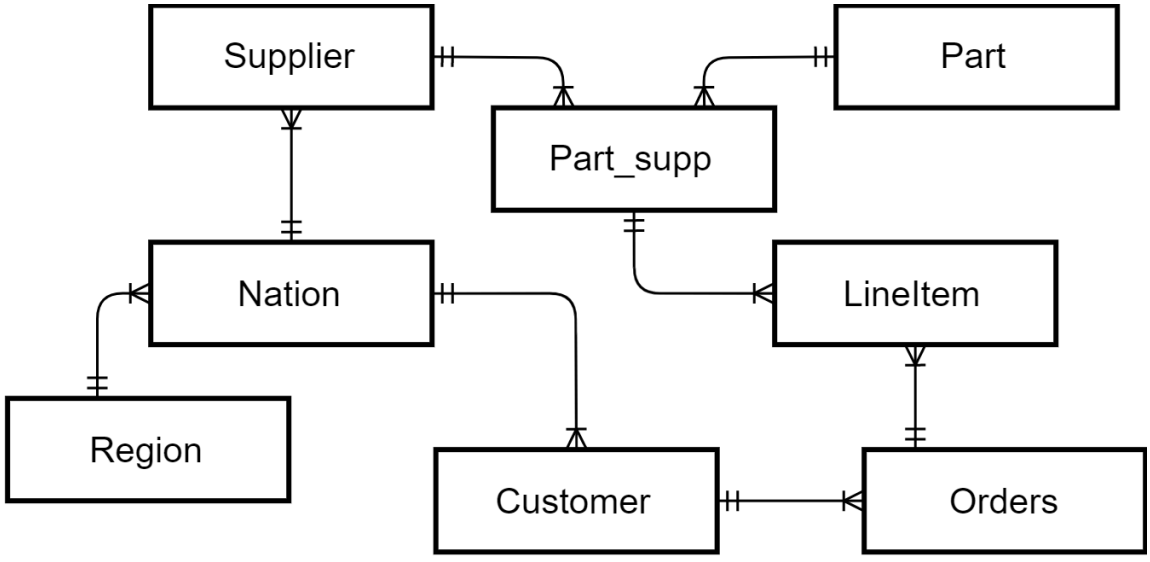



Figura 7 – Modelo inicial do TPC-H.

## 4.2 Proposta de Objetos do Tipo *Date, Time e Spatial*

Os autores já referidos, incentivam o uso de *date* e *time objects* para armazenar atributos temporais que podem ser usados pelos *analytical objects,* visto que estes objetos por serem consideravelmente pequenos, não afetarão significativamente a performance dos sistemas de *Big Data Warehousing*. O uso de *spatial objects* também é recomendado, porque os mesmos permitem uma uniformização significativa dos atributos espaciais em todos os *analytical objects* do *Big Data Warehouse*, assegurando que as cidades e países tenham o mesmo significado em todo o modelo de dados.

Assim, a primeira regra assegura a verificação, através da semântica, da existência de entidades ou atributos de cariz temporal e de entidades espaciais; assim sendo, o método de processamento de R1, Figura *8*, percorre todas as entidades três vezes. Desta forma, inicia com o objetivo de se certificar da existência de alguma entidade chamada *date*; porém, se se verificar, pela primeira vez é criada uma entidade, classificada como *date object,* adicionando um relacionamento entre a mesma e a entidade que tem o tipo de atributo *date*; se numa das seguintes entidades verificada tiver um atributo do tipo *date,* só é adicionado o relacionamento. Logo que todas as entidades sejam percorridas pela primeira vez, é iniciada uma nova verificação das mesmas, aplicando os mesmos passos já descritos; agora, no entanto, com o objetivo de procurar entidades ou atributos do tipo *time*. Por fim, todas as entidades são percorridas pela última vez, de modo a notar entidades do tipo *spatial* e, caso se verifique, é classificada como *spatial object;* se não, nenhuma entidade é classificada e termina a R1.

Considerando a *Tabela 1* tem-se, agora, a verificação formal da R1 que consiste em testar o processo em que cada uma das entidades *"Orders", "Customers", "Nation", "Region", "Supplier", "Part_supp", "Part" e "LineItem"*, do modelo TPC-H é sujeita em cada um dos passos da R1.

No caso da entidade *"Orders",* no passo 1.4, não verifica a condição de ser uma entidade do tipo *date;* como tal, avança para o passo 1.5 e verifica que cumpre a condição de ter atributo(s) do tipo *date*; por isso, avança para o passo 1.6; este confirma que ainda não existe uma entidade *date* criada; consequentemente, no passo 1.7, cria a entidade *date*; no passo 1.8 classifica-o como *date object* e no passo 1.9 adiciona o relacionamento, consoante a chave estrangeira; por fim, nos passos 1.11, 1.15 e 1.20, não verifica nenhuma das condições. A entidade *"LineItem"* é semelhante à *"Orders",* diferindo, apenas, no passo 1.6, porque a entidade *date* já está criada; por isso, passa logo para o passo 1.9, adicionando, assim, o seu relacionamento com a entidade *date*.

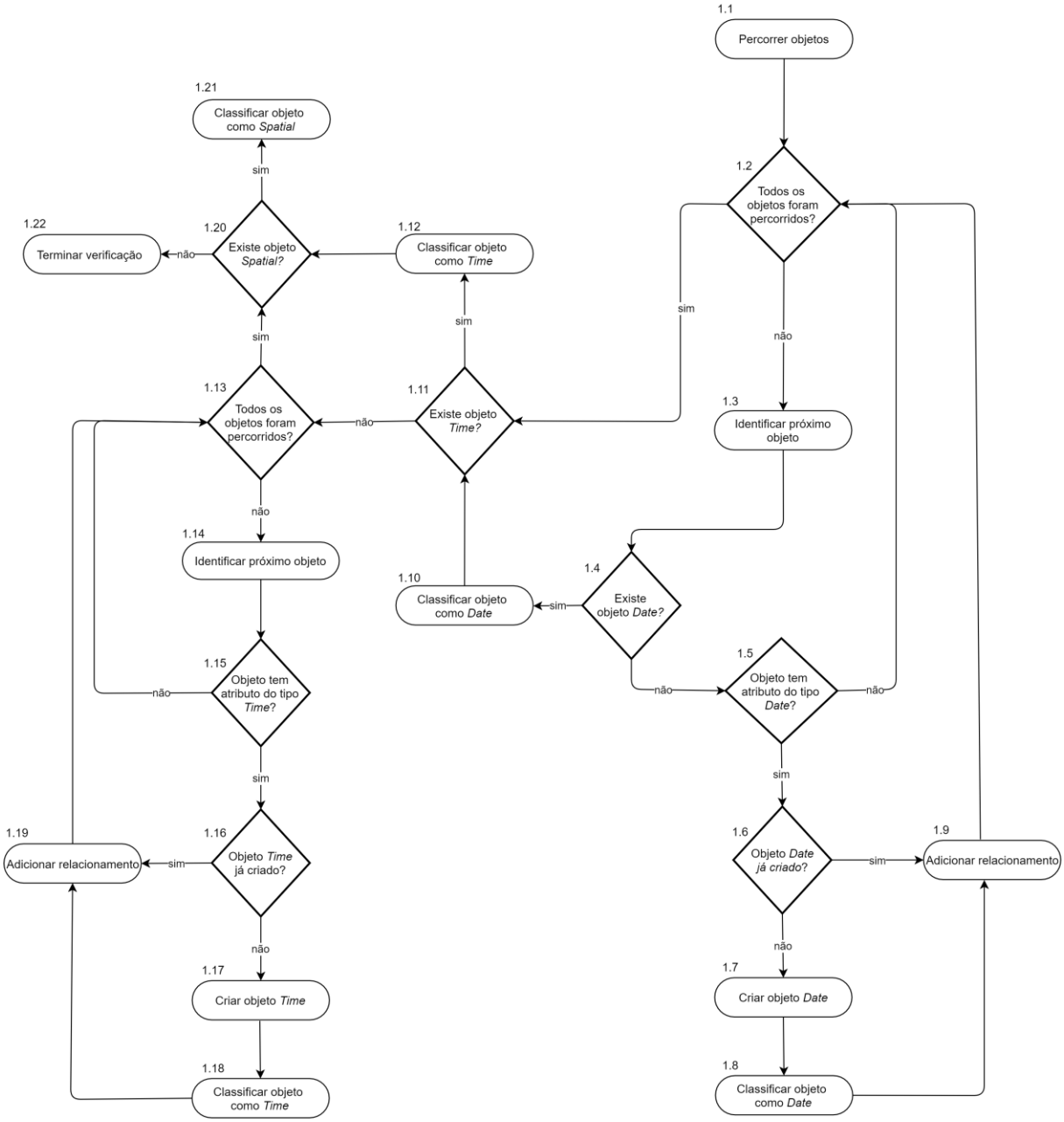



Figura 8 – Passos do método da R1.

As entidades *"Nation"* e *"Region",* na R1 do método, só verificam o passo 1.20 que é referente à condição de existir entidades do tipo *spatial object* e, consequentemente, são classificadas no passo 1.21.

Por fim, as entidades "*Customers", "Supplier", "Part_supp"* e *"Part"* não verificaram nenhuma das condições dos passos. A entidade *date,* como foi adicionada ao modelo e classificada, não é verificada.

Tabela 1 – Verificação formal da R1.

| | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 1.11 | 1.15 | 1.20 | 1.21 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Orders** | não | sim | não | ✓ | ✓ | ✓ | não | não | não | — |
| **Customer** | não | não | — | — | — | — | não | não | não | — |
| **Nation** | não | não | — | — | — | — | não | não | sim | ✓ |
| **Region** | não | não | — | — | — | — | não | não | sim | ✓ |
| **Supplier** | não | não | — | — | — | — | não | não | não | — |
| **Part_supp** | não | não | — | — | — | — | não | não | não | — |
| **Part** | não | não | — | — | — | — | não | não | não | — |
| **LineItem** | não | sim | sim | — | — | ✓ | não | não | não | — |
| **Date** | — | — | — | ✓ | — | — | — | — | — | — |

A Figura *9* apresenta o modelo, após a aplicação dos passos da R1 do método, identificando na mesma os passos em que ocorreram as modificações; o uso de cores sugere que as entidades foram classificadas. No caso da cor verde, significa que a entidade foi classificada como *date object,* e no caso da cor cinzenta, as entidades foram classificadas como *spatial objects*.

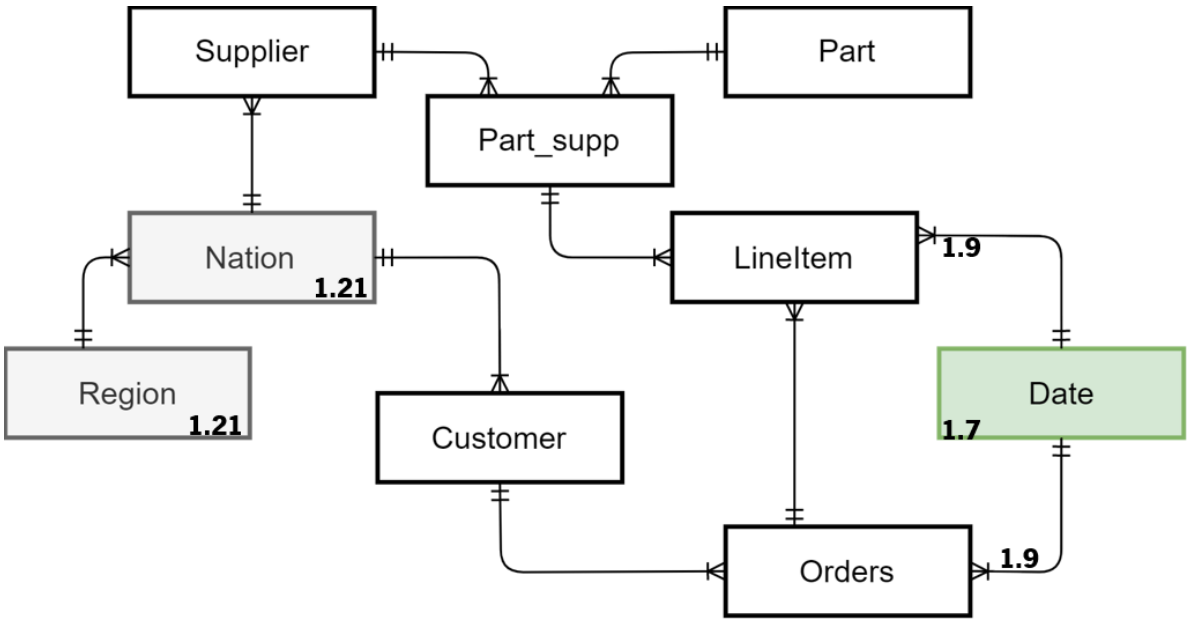



Figura 9 - Modelo após aplicação da R1.

## 4.3 Proposta de Objetos do tipo *Analytical Object*

O estudo de Santos & Costa (2016), propõe um conjunto de regras que automatizam a transformação de um modelo de dados multidimensional, tanto numa base de dados em colunas NoSQL como num esquema de dados Hive, que permite (o conjunto de regras) a implementação de um *Data Warehouse* num ambiente de *Big Data.* Posto este enquadramento do contexto inerente à proposta de regras feita pelos autores, as mesmas serviram de ponto de partida para a conceção da R2: Proposta de Objetos do Tipo *Analytical Objects,* do método e, consequentemente, para validação da mesma. A conceção da R2 seguiu as três primeiras regras da proposta dos autores; como tal, iniciou com a identificação das tabelas, tabelas de facto e dimensões; já o passo seguinte, identifica os atributos descritivos presentes nas dimensões, enquanto as tabelas de facto fornecem os atributos analíticos; por fim, são considerados os relacionamentos entre as tabelas de facto e tabelas de dimensões.

Assim, o ponto de partida do desenvolvimento do pensamento dos passos da R2 Figura *10* do método seguiu um processo semelhante ao apresentado anteriormente; deste modo, a R2 percorre todas as entidades com o objetivo de identificar as que possam ser classificadas com *analytical object*, o que, na proposta dos autores, corresponde às tabelas de facto. Para tal, são percorridas todas as entidades, de maneira a verificar o tipo de relacionamentos que cada uma recebe. Caso se verifique que a mesma só recebe relacionamentos de muitos, passa para o passo seguinte, que vai confirmar se existe mais do que um relacionamento; desta forma, se as duas condições forem cumpridas, a entidade é classificada como *analytical object,* isto é, entidades com um assunto isolado com relevância analítica; se não cumprir estas condições, não é classificada. A verificação das entidades termina após se verificar que todas foram percorridas.

Similarmente, à tabela apresentada anteriormente, a *Tabela 2* expõe a verificação formal da R2 do método, verificando assim se algum passo se aplica a alguma entidade do modelo. Neste caso, só a entidade *"LineItem"* verifica as condições dos passos 2.4 e 2.6; se o passo 2.4 verificar que o objeto recebe unicamente relacionamentos do tipo muitos passa para o passo 2.6 e verifica se o objeto recebe mais do que um relacionamento se sim o objeto transita para o passo 2.7, onde é classificada como *analytical object*. As entidades *"Orders", "Customers", "Supplier", "Part_supp" e "Part"* não verificam a condição 2.4; logo, passam para o passo 2.5. As entidades *"Date", "Nation"* e *"Region"* já foram classificadas; por isso, não voltam a ser percorridas.

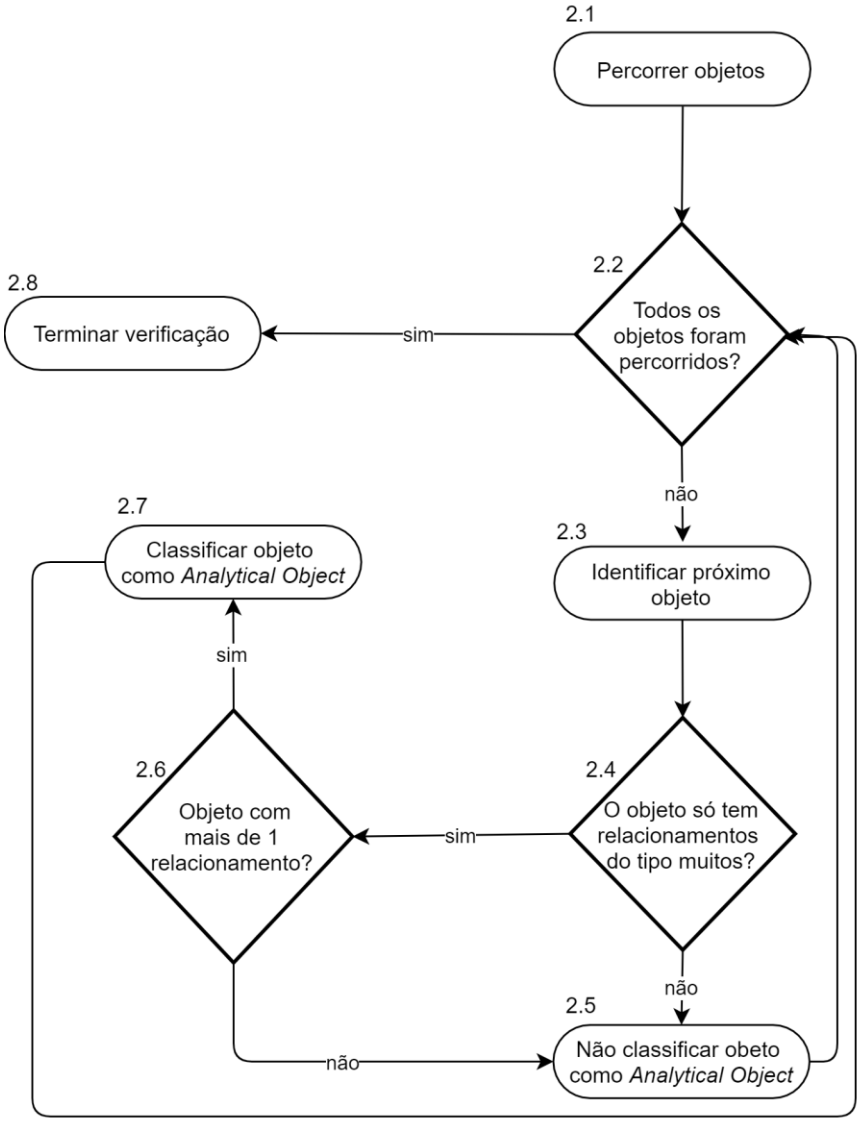



Figura 10 - Passos do método da R2.

Tabela 2 - Verificação formal da R2.

| | 2.4 | 2.5 | 2.6 | 2.7 |
|---|---|---|---|---|
| **Orders** | não | ✓ | — | — |
| **Customer** | não | ✓ | — | — |
| **Nation** | — | — | — | — |
| **Region** | — | — | — | — |
| **Supplier** | não | ✓ | — | — |
| **Part_supp** | não | ✓ | — | — |
| **Part** | não | ✓ | — | — |
| **LineItem** | sim | — | sim | ✓ |
| **Date** | — | — | — | — |

Tendo em consideração o modelo, após aplicação da R1, ilustrado nesta secção, as diferenças do mesmo para o modelo, após aplicação da R2 apresentado na Figura *11*, passam pela classificação da entidade *"LineItem"* como *analytical object.*

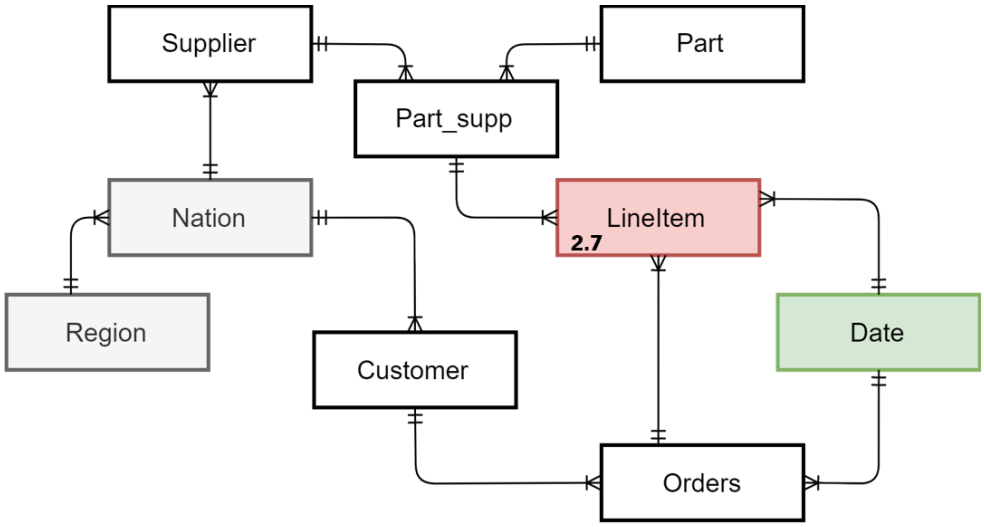



Figura 11 - Modelo após aplicação da R2.

## 4.4 Proposta de Objetos Integráveis

A abordagem do *Data Modelling* é baseada na desnormalização de dados, sendo a mesma vista como um passo crítico para o armazenamento flexível em sistemas de *Big Data Warehouse.* Deste modo, a terceira regra do método têm como finalidade encontrar entidades que possam ser desnormalizadas para que as redundâncias sejam mínimas.

Por esta razão, a R3 do método, proposta de objetos integráveis, surge com a finalidade de desnormalizar pares de entidades que verifiquem a condição.

$$\sum FK(A+B) \geq 3 \;\; AND \;\; (FK(A) = 1 \; OR \; FK(B) = 1)$$

Isto é, o somatório das chaves estrangeiras do par de objetos é maior ou igual a três, e um dos objetos do par só pode ter no máximo uma chave estrangeira. Para facilitar a compreensão da condição, a Figura *12* ilustra em a) o exemplo de um caso abstrato que cumpre a condição, e em b) o resultado do caso exemplo após aplicação da R3.

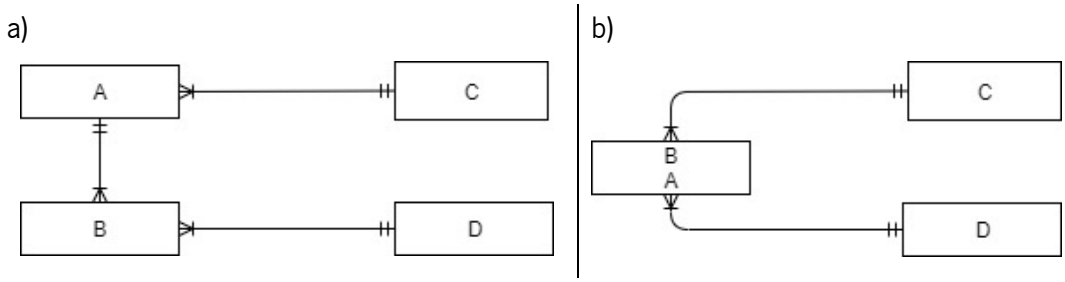



Figura 12 – a) Caso exemplo; b) Caso exemplo após aplicação da R3 do método.

Assim sendo, a Figura *13* representa o fluxograma referente aos passos inerentes à R3, ou seja, são percorridos todos os pares do modelo e se algum verificar a condição descrita anteriormente e representada em a) da Figura *12*, o par desnormaliza de acordo com chave estrangeira, tal como é apresentado em b); se a condição não se verificar, o par de entidades não é desnormalizado.

Considerando a verificação formal da R3 apresentada na Tabela *3*, a mesma serve verificando assim se a condição se aplica em algum par de entidades do modelo.. Assim sendo, no passo 3.4 foram verificados os pares "Orders-Customer", "Orders-LineItem", "Orders-Date", "Customer-Nation", "Nation-Region", "Supplier-Nation", "Supplier-Part_supplier", "Part-Part_supplier", "LineItem-Part_supplier" e "LineItem-Date". Os pares "Orders-Customer" e "Supplier-Part_supplier" verificaram a condição $\sum FK(A+B) \geq 3\ AND\ (FK(A) = 1\ OR\ FK(B) = 1)$; por isso, os mesmos prosseguiram para o passo 3.6, onde a entidade *"Customer"* foi desnormalizada na entidade *"Orders"* e a entidade *"Supplier"* foi desnormalizada na entidade *"Part-supplier"*, os restantes pares não verificaram a condição no passo 3.4; por isso passaram para o passo 3.5, e assim sucessivamente até todos os pares terem sido verificados.

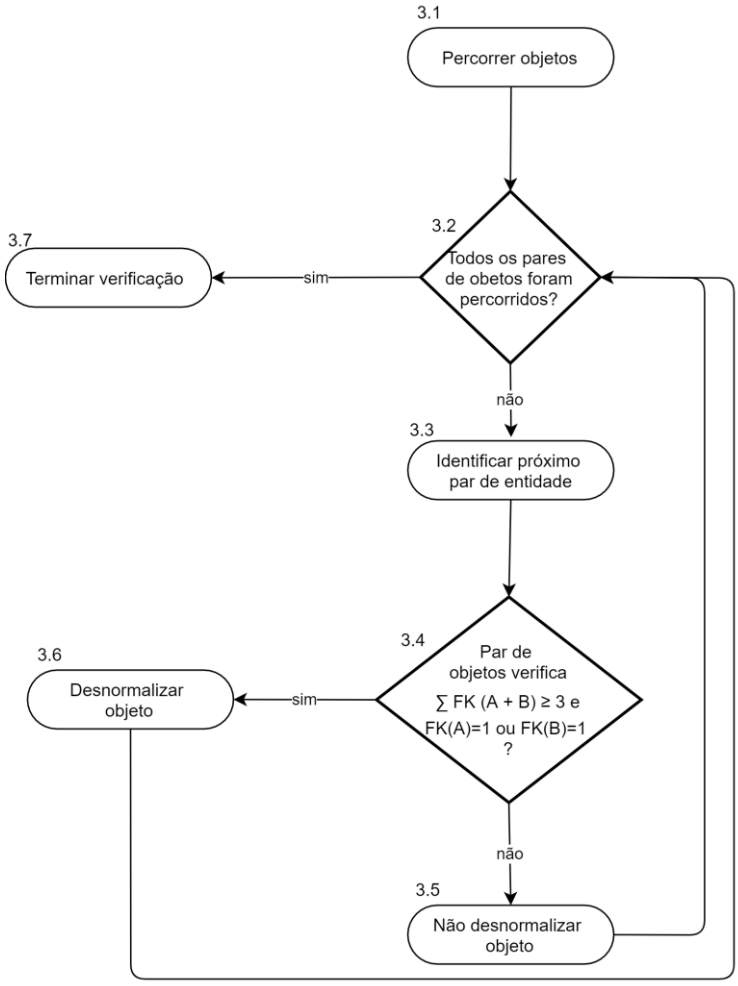



Figura 13 - Passos do método da R3.

A entidade *"Customer"* foi desnormalizada na entidade *"Orders"* e a entidade *"Supplier"* foi desnormalizada na entidade *"Part_supp"*. Podendo ser confirmado o mesmo na *Figura 14*, nas entidades de cor lilás e com a identificação do passo 3.6, da verificação formal, em que ocorreu a desnormalização.

A entidade *"Customer"* foi desnormalizada na entidade *"Orders"* e a entidade *"Supplier"* foi desnormalizada na entidade *"Part_supp"*. Podendo ser confirmado o mesmo na *Figura 14*, nas entidades de cor lilás e com a identificação do passo 3.6, da verificação formal, em que ocorreu a desnormalização.

Tabela 3 - Verificação formal da R3.

| | 3.4 | 3.5 | 3.6 |
|---|---|---|---|
| **Orders – Customer** | sim | — | ✓ |
| **Orders – LineItem** | não | ✓ | — |
| **Orders – Date** | não | ✓ | — |
| **Customer – Nation** | não | ✓ | — |
| **Nation – Region** | não | ✓ | — |
| **Supplier - Nation** | não | ✓ | — |
| **Supplier – Part_supp** | sim | — | ✓ |
| **Part – Part_supp** | não | ✓ | — |
| **LineItem – Part_supp** | não | ✓ | — |
| **LineItem - Date** | não | ✓ | — |

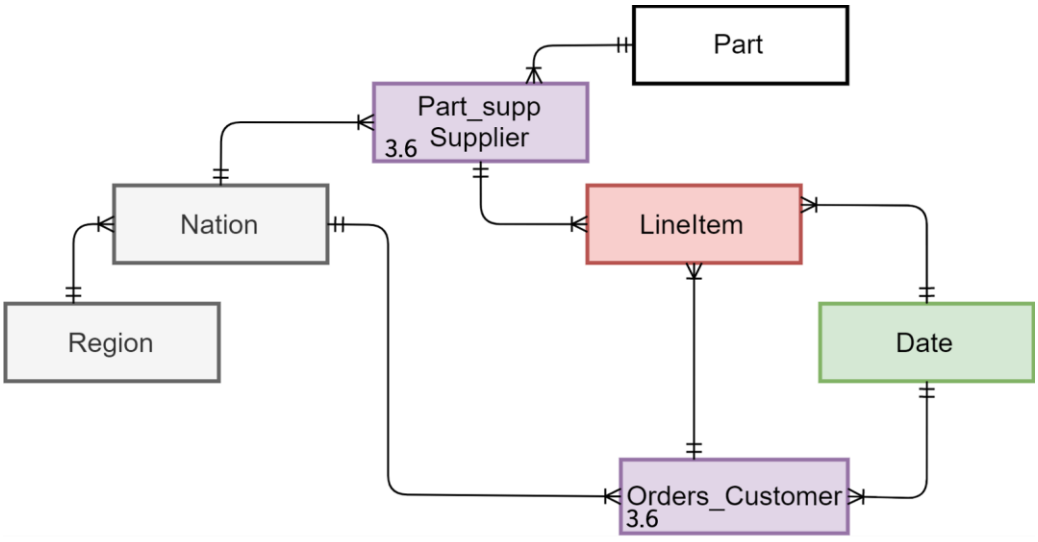



Figura 14 - Modelo após aplicação da R3.

## 4.5 Proposta de Objetos com Relacionamentos Múltiplos

Similarmente, à abordagem de *Data Modelling* apresentada na regra anterior, R3, a desnormalização de dados, a presente, R4 do método, surgiu com a finalidade de resolver um dos típicos problemas encontrados em base de dados relacionais referentes à existência de entidades com relacionamentos múltiplos, permitindo a desnormalização das que verifiquem a condição. Desta forma, este tipo de relacionamentos da R4 do método verifica se alguma das entidades verifica a condição.

*Select Count Distinct* (FK_1, FK_2,...) = 1

Assim, e de forma a facilitar a compreensão da R4 do método, a *Figura 15* apresenta dois exemplos, a) e b), semelhantes no sentido que uma entidade recebe dois relacionamentos de muitos, contudo, só o exemplo a) verifica a condição '*Select Count Distinct* (*PartKey*, *SupplierKey*) = 1', isto é, a existência do par dos atributos *"PartKey"* e *"SuppKey"* são simultaneamente chave primária (PK) e chave estrangeira (FK) da entidade *"Part_Supp"*; deste modo, a repetição do par só pode ocorrer no máximo uma vez, enquanto o caso de demonstração b) verifica a condição '*Select Count Distinct* (FK_1, FK_2,...) > 1', isto porque o par de atributos "*CustomerKey"* e *"Date*" são chave estrangeira da entidade *"Orders"* e podem-se repetir mais de uma vez, tendo como exemplo a figura b) é exequível um cenário em que um cliente possa fazer várias compras no mesmo dia.

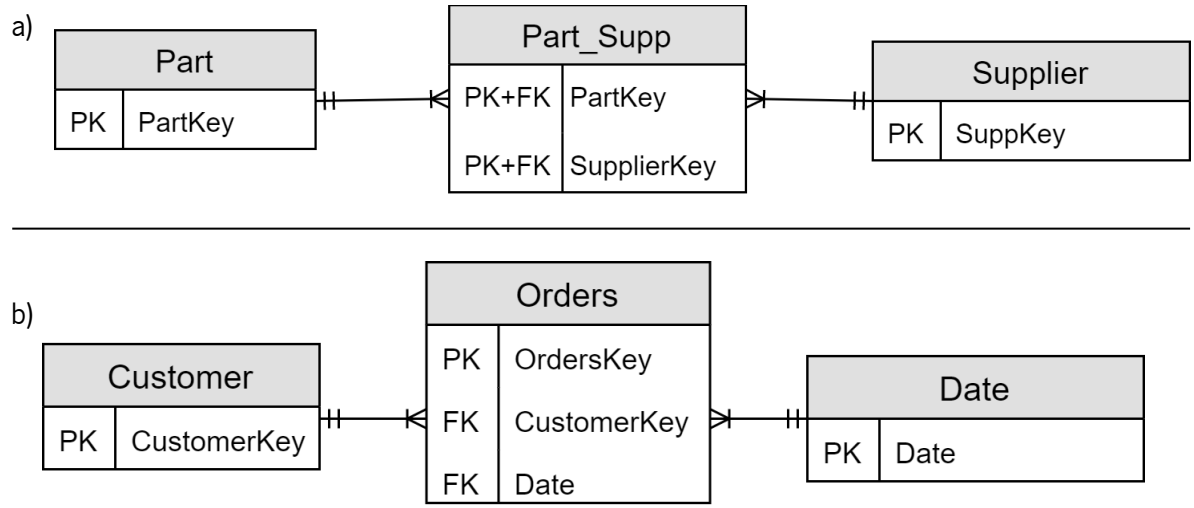



Figura 15 – Exemplo de explicação para a R4 do método.

Conforme já apresentado nesta subsecção, o processo de abordagem permite confirmar se alguma entidade verifica a condição '*Select Count Distinct* (FK_1, FK_2,...) = 1'; se sim, a mesma desnormaliza de acordo com as chaves estrangeiras e prossegue os restantes passos. No caso apresentado no exemplo a), em que verifica a condição, a entidade *"Part_Supp"* desnormaliza as entidades *"Part"* e *"Supplier";* no entanto, no caso do exemplo b) a condição não é verificada; por isso, não desnormaliza.

A *Figura 16* apresenta os passos da R4, o desenvolvimento da mesma teve como ponto de partida identificar entidades que verifiquem o passo 4.4. Caso se verifique que a mesma cumpre a condição do passo 4.4 e prossegue para o passo 4.6 onde a mesma é desnormalizada; se não passa para o passo 4.5 e continua a percorrer as restantes entidades.

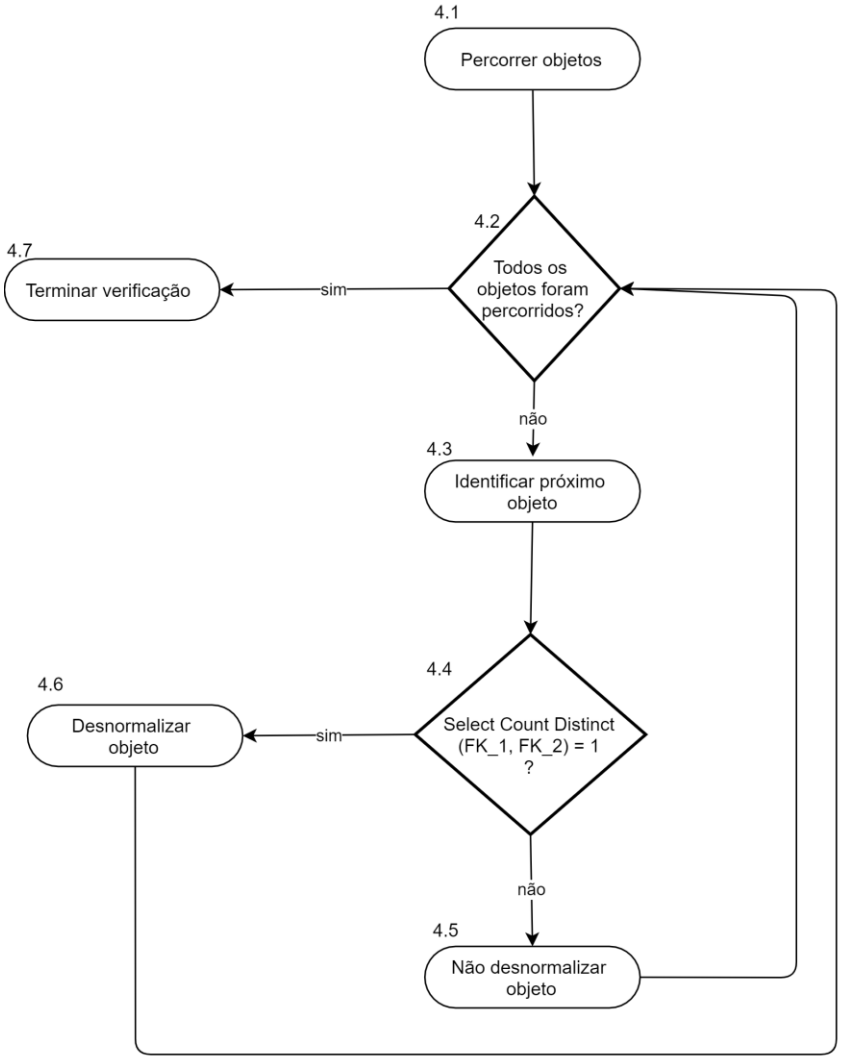



Figura 16 - Passos do método da R4.

## 4.6 Proposta de Objetos Padronizáveis

Similarmente ao que foi descrito na R3 e R4, a respeito da abordagem do método de *Data Modelling* baseado na desnormalização de dados, a R5 do método foi criada para identificar objetos padronizáveis. Deste modo, a R5 tem como objetivo desnormalizar objetos padronizáveis; isto é, objetos com poucos atributos, logo com baixa cardinalidade, o que, em contextos de *Big Data*, é recomendado que os mesmos sejam desnormalizados permitindo que as *queries* sejam significativamente rápidas.

A proposta de objetos padronizáveis, tendo em conta o contexto desta regra do método consiste, por exemplo, numa entidade que tenha como atributos os tipos de expedição de uma encomenda.

Para verificar a existência de objetos padronizáveis, são analisadas todas as entidades; as que cumprirem as condições de só terem relacionamentos do tipo um e baixa cardinalidade, são armazenadas num repositório de metadados. Após todas as entidades serem percorridas e verificadas, é gerada uma lista com todos os objetos classificados como 'objetos padronizáveis'. A desnormalização não é possível unicamente com a verificação das condições já descritas; por isso, e caso se aplique, é necessária uma validação manual por parte do utilizador, que terá de selecionar, de entre os objetos classificados, os que realmente são 'objetos padronizáveis'; os selecionados, posteriormente serão desnormalizados nas entidades que têm chave estrangeira do relacionamento.

A R5 do método não se aplica no caso de demonstração TPC-H, visto que nenhuma entidade verifica a condição para ser classificada como objeto padronizável.

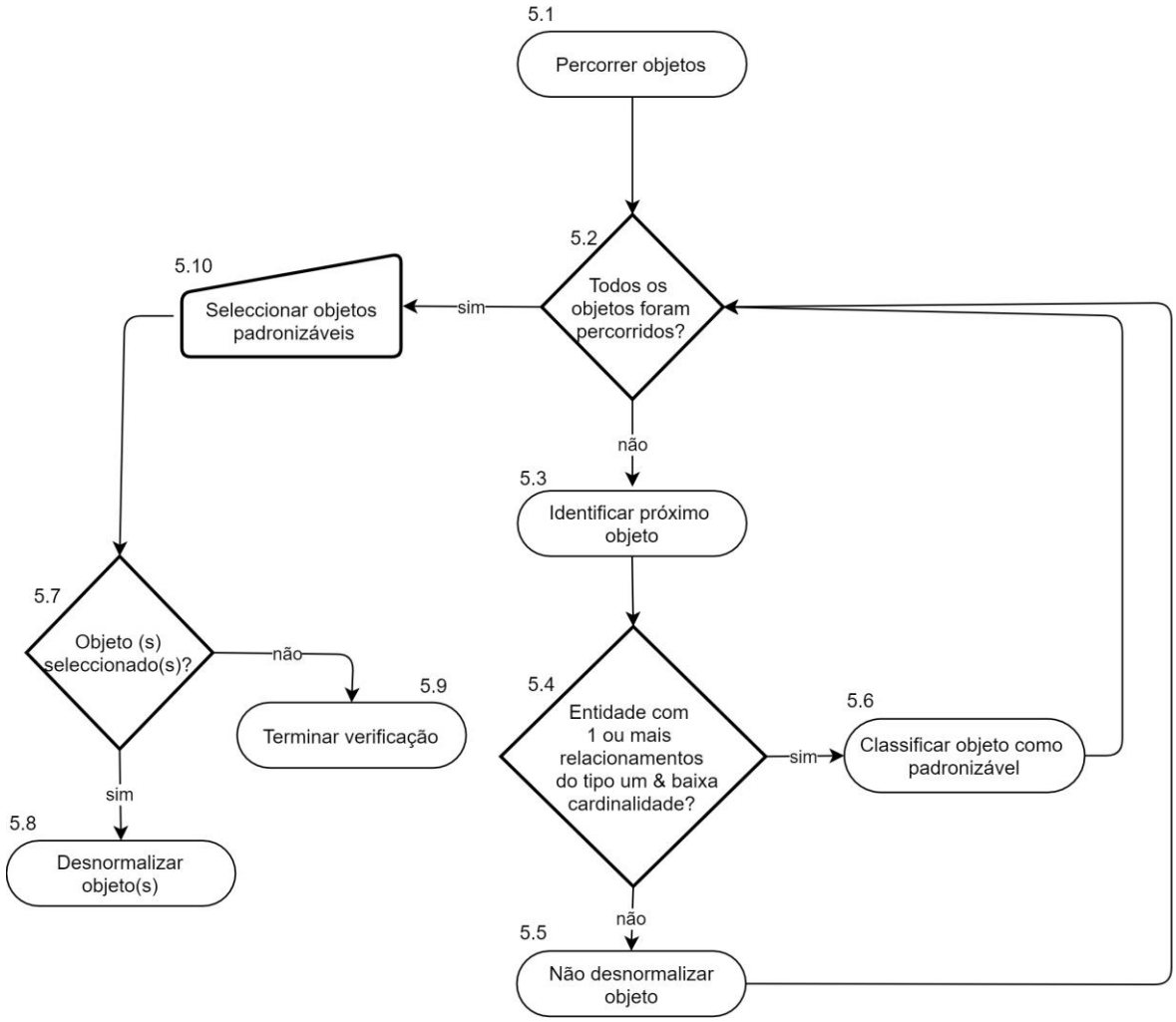



Figura 17 - Passos do método da R5.

## 4.7 Proposta de Objetos Autónomos

Similarmente ao que foi descrito nas três últimas regras, a R6 do método tem como meta desnormalizar objetos autónomos. Desta forma, a mesma percorre todas as entidades e se verificar em alguma que o único relacionamento que tem é de um, então desnormaliza-o na entidade que tem chave estrangeira; caso não se verifique a primeira condição passa para a segunda condição, e verifica se a entidade só tem um relacionamento de um e se os restantes relacionamentos que recebe são objetos especiais. Isto é, recebe relacionamentos *date object, time object* e ou *spatial object;* se for o caso, a entidade é desnormalizada. Se nenhuma das condições se verificar, as entidades não são desnormalizadas e é confirmado se as mesmas foram todas percorridas; se sim, termina a verificação; se não, continua a verificação da próxima entidade.

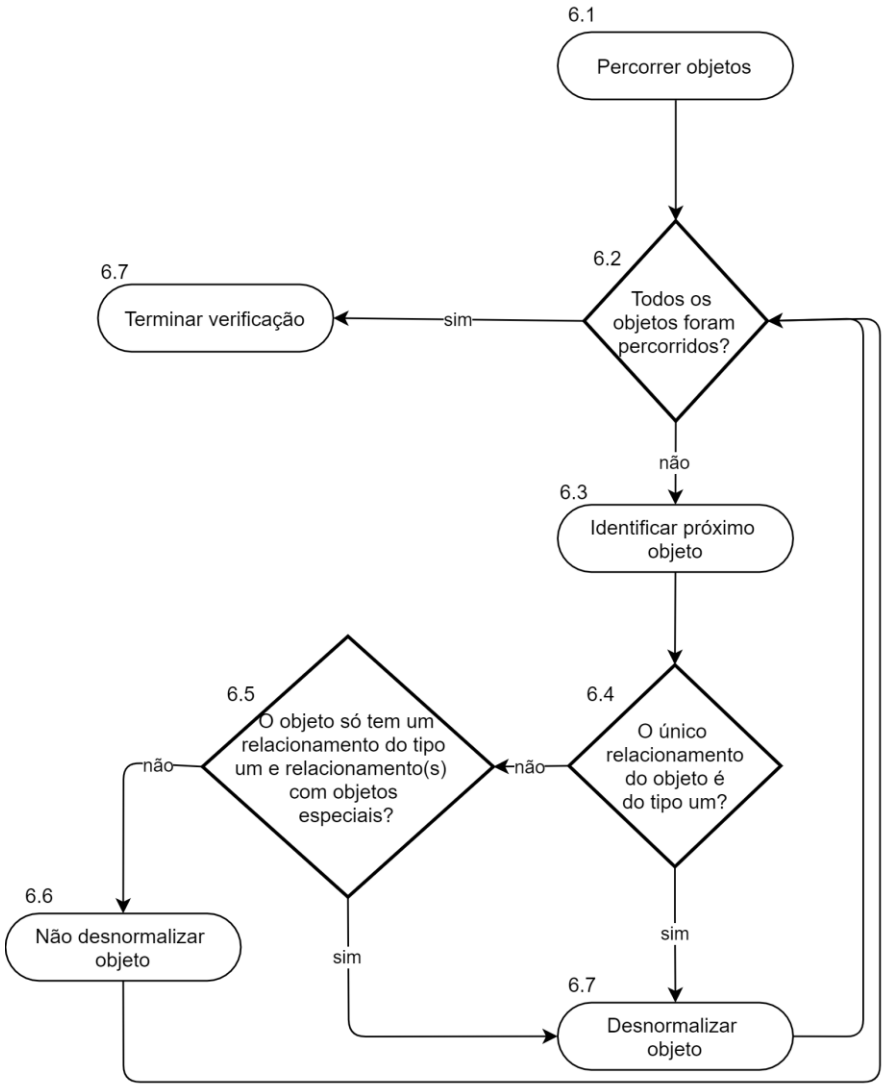



Figura 18 - Passos do método da R6.

Neste ponto é realizada a verificação formal, apresentada na *Tabela 4* de todas as entidades nos passos da R6 do método, mesmo as que já foram classificadas em regras anteriores. Como é o caso da entidade *"Region",* anteriormente classificada como *spatial object,* que no passo 6.4 verifica a condição de o único relacionamento que recebe é de um; por isso, avança para passo 6.7, onde é desnormalizada, de acordo com a chave estrangeira. O mesmo se sucede com a entidade *"Part"* que verificou a condição do passo 6.4, por isso avançou para o passo 6.7 e foi desnormalizada na entidade *"Part_Supp Supplier"*. As restantes entidades não verificam o passo 6.4; por isso, passam para o passo 6.5, em que confirmam ou não se a mesma só recebe um relacionamento de um e se os restantes relacionamentos correspondem a objetos especiais; no caso do TPC-H, nenhuma entidade verifica o passo 6.5; como tal, passam para o passo 6.6, e assim sucessivamente, até todas as entidades serem percorridas.

Tabela 4 - Verificação formal da R6.

| | 6.4 | 6.5 | 6.6 | 6.7 |
|---|---|---|---|---|
| **Orders_Customer** | não | não | ✓ | — |
| **Nation** | não | não | ✓ | — |
| **Region** | sim | não | — | ✓ |
| **Part_supp Supplier** | não | não | ✓ | — |
| **Part** | sim | não | — | ✓ |
| **LineItem** | não | não | ✓ | — |
| **Date** | — | — | — | — |

A entidade *"Region"* foi desnormalizada na entidade *"Nation"* e a entidade *"Part"* foi desnormalizada na entidade *"Part_Supp Supplier".* Podendo ser confirmado o mesmo na Figura *19*, nas entidades de cor lilás e com a identificação do passo 6.7, da verificação formal, em que ocorreu a desnormalização.

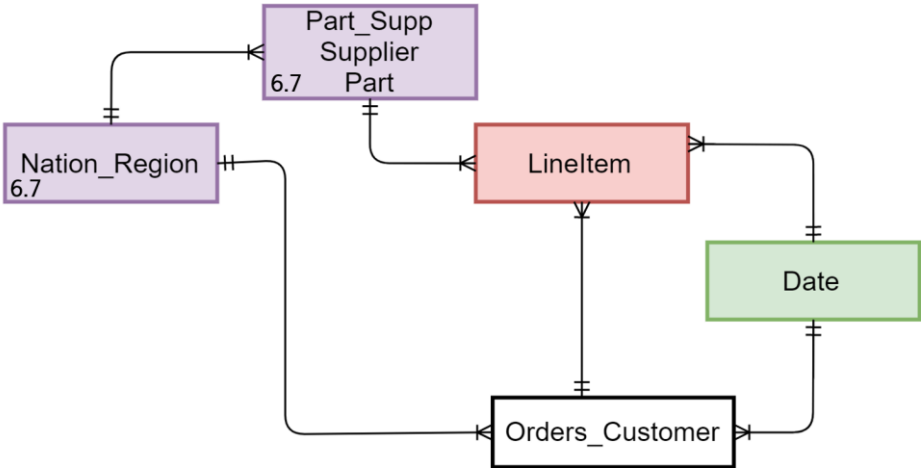



Figura 19 - Modelo após aplicação da R6.

## 4.8 Proposta de Objetos do Tipo *Complementary Analytical Objects*

Segundo o estudo de Santos et al. (2019), referido na subsecção 2.3.2, um *analytical object* deve conter, nos atributos descritivos, atributos que correspondam à *granularity key* de outro objeto.

Desta forma, a R7 do método percorre as entidades que ainda não foram classificadas, com a finalidade de verificar primeiramente se existem entidades com um ou mais relacionamentos de muitos e um ou mais relacionamentos de um; se a condição for cumprida, classifica a mesma como *complementary analytical object.* Em segundo lugar, se a entidade não verificar a condição descrita anteriormente, será verificado se a mesma só tem relacionamentos de um; se sim, é medida a cardinalidade e é classificada com *complementary analytical object,* se apresentar alta cardinalidade.

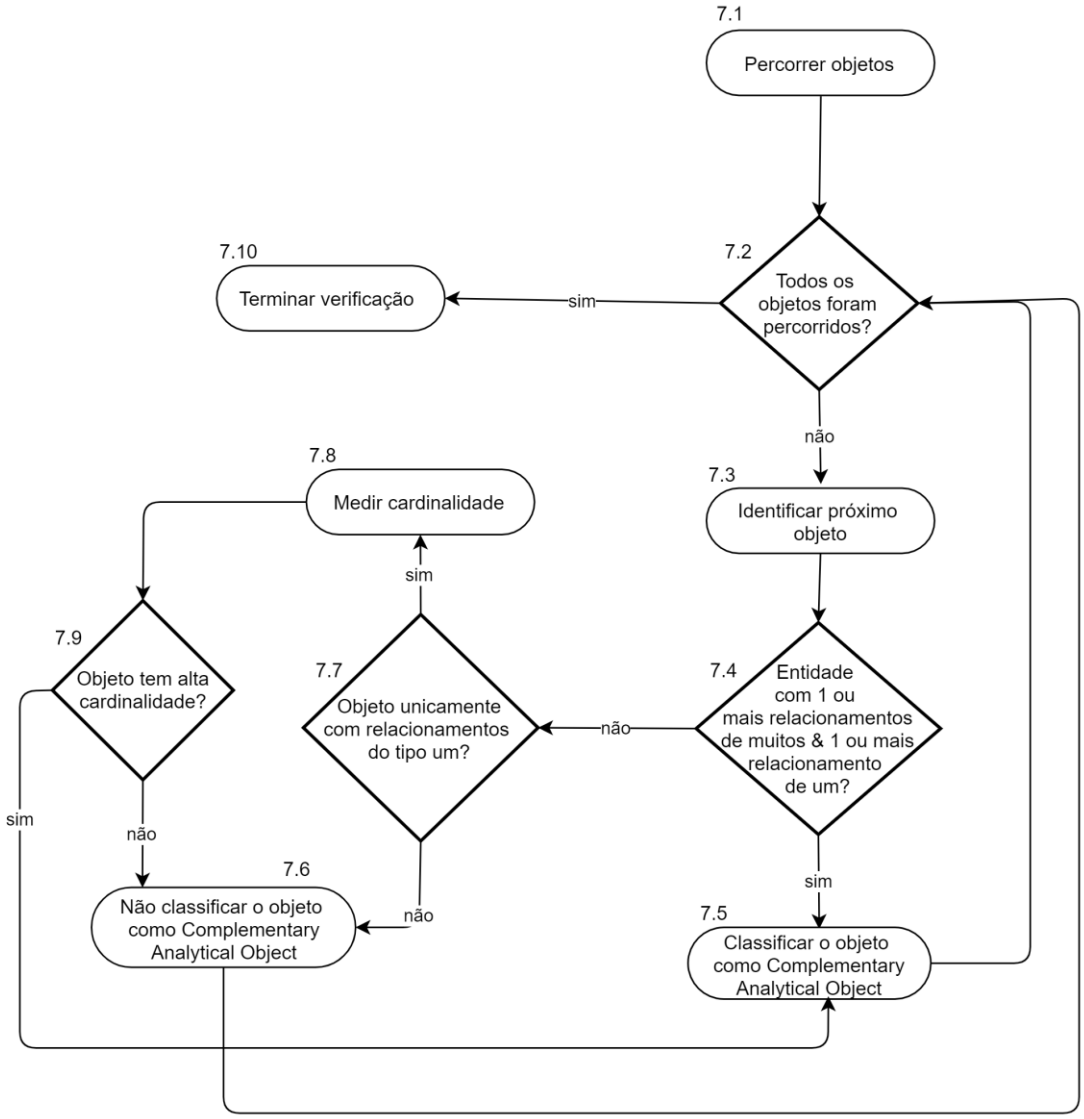



Figura 20 - Passos do método da R7.

Por último, a verificação formal da R7 do método verifica os passos da última regra, a fim de verificar e classificar as entidades ainda não classificadas em regras anteriores e que a mesma cumpra uma das

duas condições. No passo 7.4, as entidades *"Orders", "Customer"* e *"Part_Supp, Supplier, Part"* verificam a condição de possuírem um ou mais relacionamentos de muitos, e pelo menos, um relacionamento de um; desta forma, cada uma das três entidades passa para o passo 7.5, onde serão classificadas como *complementary analytical object*.

Tabela 5 - Verificação formal da R7.

| | 7.4 | 7.5 | 7.6 | 7.7 | 7.8 | 7.9 |
|---|---|---|---|---|---|---|
| Orders_ Customer | sim | ✓ | — | — | — | — |
| Nation_Region | — | — | — | — | — | — |
| Part_Supplier | sim | ✓ | — | — | — | — |
| LineItem | — | — | — | — | — | — |
| Date | — | — | — | — | — | — |

Após aplicação da R7 do método o resultado apresentado na Figura *21* corresponde ao modelo final. Nesta última iteração os retângulos azuis dizem respeito às entidades classificadas, no passo 7.5, como *complementary analytical objects.*

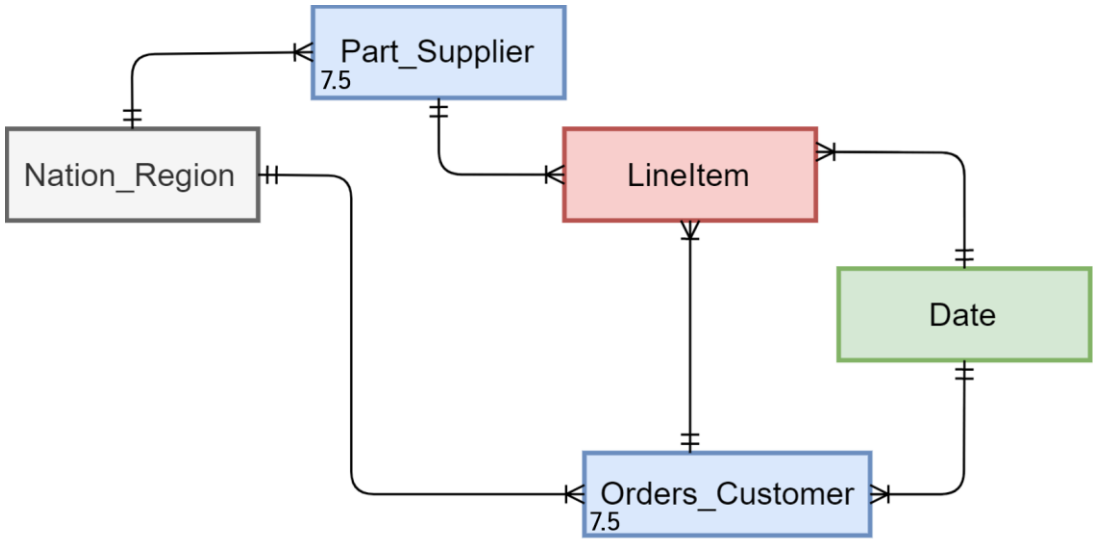



Figura 21 - Modelo após aplicação da R7.

Sistematizando, a aplicação do método no caso de demonstração TPC-H, a par das tabelas que fazem a verificação formal de cada regra, fornecem o resultado do processo dos passos dos fluxogramas após a aplicação de cada regra.

O modelo final, após a aplicação do método, apresenta todas as entidades classificadas: 1 *analytical object "LineItem"*, 2 *complementary analytical objects "Part_Supplier"* e *"Orders_Customer"*, 1 *date object* e 1 *spatial object*.

# 5. Demonstração e Avaliação do Método Proposto

Após a proposta do método de *Data Modelling*, apresentado no capítulo 4, o presente capítulo considera quatro casos de demonstração de potenciais aplicações do mundo real, tais como notícias e eventos mundiais, retalho, finanças e produção.

Este capítulo tem como objetivo demonstrar a aplicação do método em potenciais exemplos do mundo real, e o mesmo foi feito comparando as diferenças entre o diagrama do modelo inicial com o resultado após a aplicação do método no modelo, isto é, através da execução dos passos do mesmo. Consequentemente, em cada uma das quatro subsecções da demonstração do método, e de modo a enquadrar a proposta do conjunto de regras nos diferentes contextos, são identificados e descritos os passos aplicados em cada um dos exemplos. A secção de avaliação do método proposto utiliza os modelos TPC-DS, TPC-E e Projeto GDELT, de forma a validar o desenvolvimento do método e os resultados dos mesmos serem comparados com a proposta do método de *Data Modelling* para sistemas de *Big Data Warehouse* que os autores Santos et al. (2019) apresentaram, de cada um dos exemplos, no estudo cientifico. Em suma, a descrição de todo o processo e as respetivas considerações finais, em cada um dos exemplos, permitem depreender limitações do método que posteriormente podem ser consideradas e desenvolvidas em trabalho futuro.

## 5.1 Demonstração do Método

As três primeiras subsecções, da demonstração do método, apresentam o projeto GDELT (GDELT, 2018), o TPC *Benchmark* E (TPC-E, 2018) e o TPC *Benchmark* DS (TPC-DS, 2019), e foram selecionados, com o propósito de comparar o modelo inicial com os resultados do modelo após aplicação do método. A última subsecção contempla a demonstração do TPC *Benchmark* C (TPC-C, 2010) e foram selecionados com o propósito de testar e validar a aplicação do método em contextos mais pequenos relativamente aos primeiros três apresentados.

De forma a melhor organizar o documento, a demonstração da aplicação das regras entre o modelo inicial e o modelo final pode ser consultada na íntegra em ANEXOS, por não serem o foco do capítulo de demonstração, ou seja, podem ser consultadas todas as iterações ocorridas no modelo através da aplicação de cada regra.

### 5.1.1 Projeto GDELT

Nesta secção é apresentado o projeto GDELT[2]. Consiste numa base de dados aberta que monitoriza notícias, de quase todo o mundo, transmitidas, impressas, e publicadas na web identificando pessoas, locais, organizações, temas, pesquisas, emoções, imagens, entre outras informações presentes nas notícias.

A *Figura 22* demonstra o modelo inicial do projeto GDELT e o resultado do modelo após aplicação do método, permitindo identificar e comparar as diferenças ocorridas entre os mesmos. Entre o modelo inicial e o modelo final, ocorreram três iterações e o resultado das mesmas pode ser consultado na íntegra em Anexo 1 – Projeto GDELT. Para o resultado do modelo, após aplicação do método, foram executadas a R1, a R2, a R5 e, novamente, a R5, porque, da segunda vez que foi aplicado o método ao modelo, as entidades (*"Media", "Geo", "Event"* e *"Actor"*) verificaram a condição de o único relacionamento que recebem ser de um. A demonstração da aplicação do método ao modelo inicial do projeto GDELT está descrita nos pontos que se seguem:

- A R1 do método permitiu classificar a entidade "*Date"* como *date object e* as entidades *"Geo", "GeoName", "GeoType", "ADM1Code", "GeoCountry", "GeoFeature"* e *"Location"* como *spatial objects* e, por fim, verificou que existia um atributo *"Time"*; por isso, criou a entidade e classificou-a como *time object;*
- A R2 do método identificou a entidade *"GdeltMain"* como *analytical object;*
- Na R6 do método, as entidades *"Num Sources", "NumMentions", "NumArticles"* e *"AvgTone"* verificam a condição de o único relacionamento que recebem ser de um, por isso são desnormalizadas na entidade *"Media";* o mesmo acontece às entidades *"Geo", "GeoName", "GeoType", "ADM1Code", "GeoCountry", "GeoFeature"* e *"Location"* que verificam a condição e são desnormalizadas na entidade *"Geo";* o mesmo acontece com as entidades *"ActorCode", "ActorName", "Religion", "KnownGroup", "Type", "Ethnic"* e *"Country"* que, à semelhança das anteriores, também verifica a condição e as mesmas são desnormalizadas na entidade *"Actor";* do mesmo modo, as entidades *"QuadClass", "EventCode"* e *"GoldsteinScale"* verificam a condição e são desnormalizadas na entidade

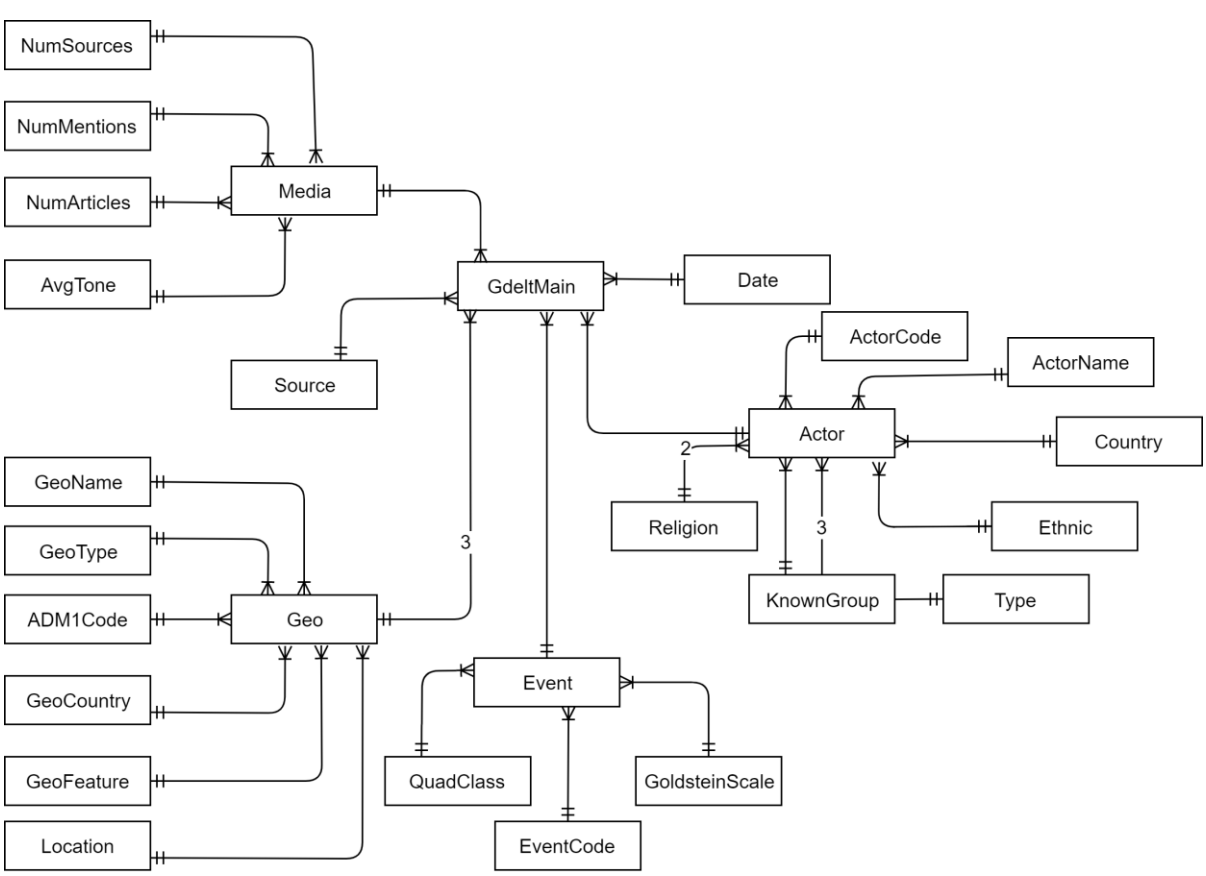
[2] https://www.gdeltproject.org/

*"Event";* por fim, a entidade *"Source"* também verifica a condição; por isso é desnormalizada na entidade *"GdeltMain";*

- Da segunda vez que foi aplicado o método a R5, voltou a verificar entidades em que o único relacionamento que recebem é de um; deste modo, as entidades *"Media", "Geo", "Event"* e *"Actor"* foram desnormalizadas na entidade *"GdeltMain".*

Sistematizando, após aplicar o método ao modelo inicial, todas as entidades foram classificadas, resultando 1 *analytical object*, 1 *date object*, 1 *time object* e 1 *spatial object.*

a)



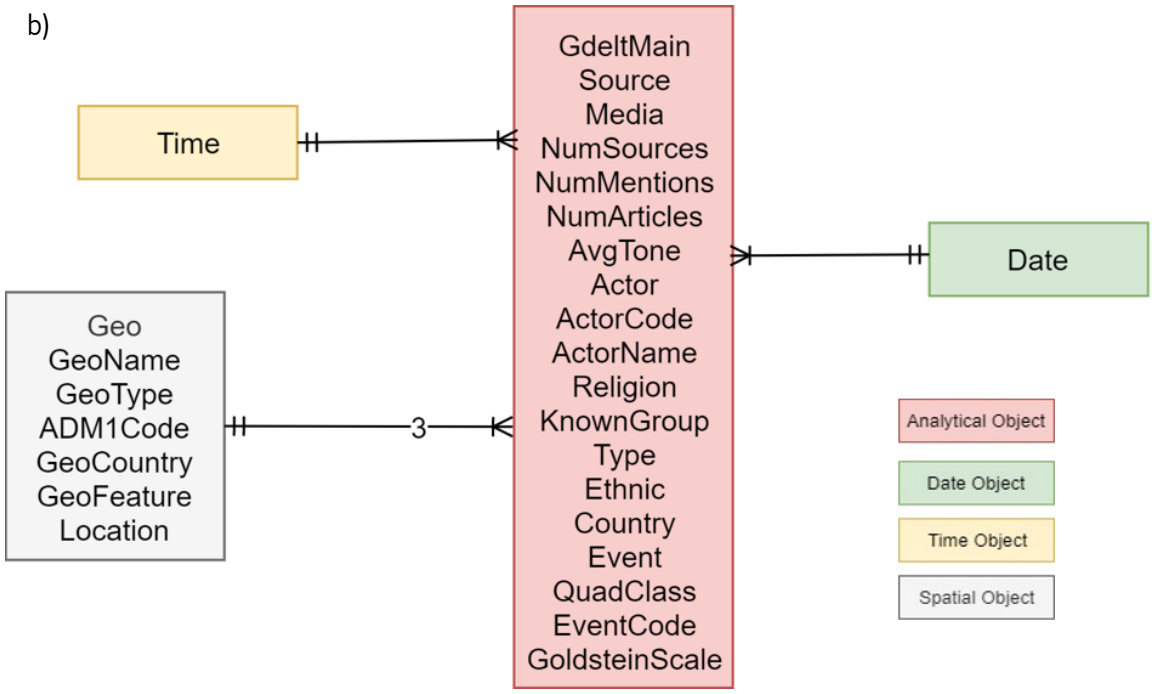



Figura 22 – Projeto GDELT – a) Modelo inicial vs b) Modelo após aplicação do método.

### 5.1.2 Caso TPC-E

O exemplo selecionado para demonstração da proposta do método apresentado nesta subseção, enquadra-se no sector financeiro, mais concretamente, numa corretora fictícia; o exemplo está descrito no TPC *Benchmark* E[3] (TPC-E) e o mesmo detalha, na integra, um sistema de base de dados transacional, no contexto financeiro de corretagem.

Comparando, na *Figura 23*, o modelo inicial e o modelo após a aplicação das seis iterações do TPC-E, o mesmo resultou na desnormalização das entidades *"Status_type", "Taxrate", "Trade_type"* e *"Trade_history",* na classificação de 13 *analytical objects*, 7 *complementary analytical objects*, 1 *date object*, 1 *time object* e 1 *spatial object.* O resultado da aplicação do método em cada uma das seis iterações do TPC-E pode ser analisado na íntegra em Anexo 2 – TPC-E.

A aplicação da R1, R2, R3, R5, R6 e R7 do método no caso TPC-E, encontrar-se-á descrita nos pontos que se seguem:

- A R1 do método não verifica nenhuma entidade *"Date"* e *"Time";* como tal, foi necessária a verificação de todos os atributos, e as entidades *"Trade", "Securrity", "Last_trade", "Daily_market", "Cash_transaction", "Settlement", "Financial", "Trade_history", "Company", "Customer", "Holding"* e *"News_item"* verificaram a condição; desta forma, a

[3] http://www.tpc.org/tpce/

primeira entidade que verificou a condição de ter atributos do tipo *date* criou uma nova entidade chamada *"Date",* classificou-a como *date object,* e as restantes entidades mencionadas adicionaram o relacionamento; o processo da entidade *"Time"* foi semelhante, à exceção que só as entidades *"News_items", "Trade_history", "Trade", "Holding", "Cash_transaction" e "Last_trade"* verificaram a condição; por fim, as entidades *"Address"* e *"ZP"* foram classificadas como *spatial objects*;

- A R2 do método classificou as entidades *"Watch_item", "Company_competitor", "Financial", "News_xref", "Last_trade", "Daily_market", "Customer_taxrate", "Settlement", "Comission_rate", "Holding_history", "Holding", "Cash_transaction"* e *"Trade_request"* como *analytical object,* por serem as entidades em que os únicos relacionamentos que recebem são do tipo muitos*;*
- A R3 do método verificou a condição num dos pares do modelo, isto é, o par de entidades *"Watch_item"* e *"Watch_list"* cumprem a condição $\sum FK(A+B) \geq 3\ AND\ (FK(A) = 1\ OR\ FK(B) = 1)$; como tal, a entidade *"Watch_list"* é desnormalizada na entidade *"Watch_item"*. Para facilitar a compreensão da mesma, a Tabela 6 verifica três exemplos de pares de entidades, *"Watch_item – Watch_list", "Financial - Company"* e *"Trade_type – Charge",* em cada uma das três condições. Em que, C1=$\sum FK(A+B) \geq 3$, C2= $FK(A) = 1$, C3= $FK(B) = 1$, C2+C3= $FK(A) = 1\ OR\ FK(B) = 1$ e o Resultado = C1 AND (C2+C3);

Tabela 6 - Caso de demonstração da R3.

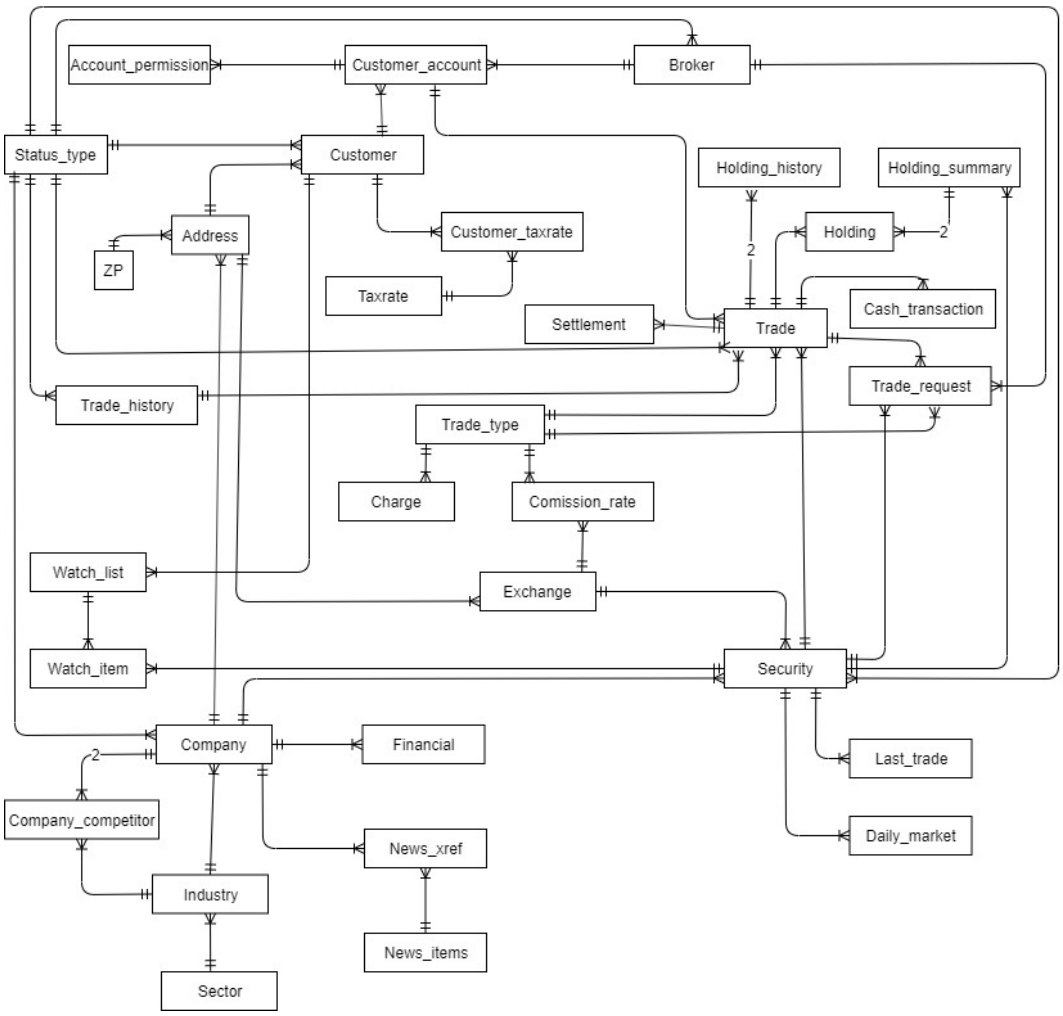
| | C1 | C2 | C3 | C2+C3 | Resultado |
|---|---|---|---|---|---|
| **Watch_list – Watch_item** | Verdadeiro | Verdadeiro | Falso | Verdadeiro | Verdadeiro |
| **Financial – Company** | Verdadeiro | Falso | Falso | Falso | Falso |
| **Trade_type – Charge** | Falso | Falso | Verdadeiro | Verdadeiro | Falso |

- Na R5 do método, as entidades *"Status_type"* e *"Trade_type"* verificaram a condição, e no passo manual foram selecionadas como objetos padronizáveis. A entidade *"Status_type"* foi desnormalizada nas entidades *"Customer", "Broker", "Trade", "Security", "Trade_type" e "Company";* a entidade *"Trade_type"* foi desnormalizada nas entidades *"Trade", "Comission_rate"* e *"Charge";*

- A R6 do método identificou cinco entidades isoladas, *"Sector", "ZP", "Taxrate", "Trade_history"* e *"News_items"* que foram desnormalizadas, de acordo com a chave;
- Por fim, a R7 do método classificou como *complementary analytical object* as entidades que cumpriam a condição, que são o caso *"Customer_account", "Customer", "Holding_summary", "Trade", "Security", "Exchange"* e *"Company".*

As entidades *"Account_permission", "Broker", "Charge"* e *"Industry",* após aplicação do método, ficaram sem classificação. As entidades *"Broker"* e *"Industry",* verificam a condição de os únicos relacionamentos que recebem ser de um; contudo, têm baixa cardinalidade.

a)



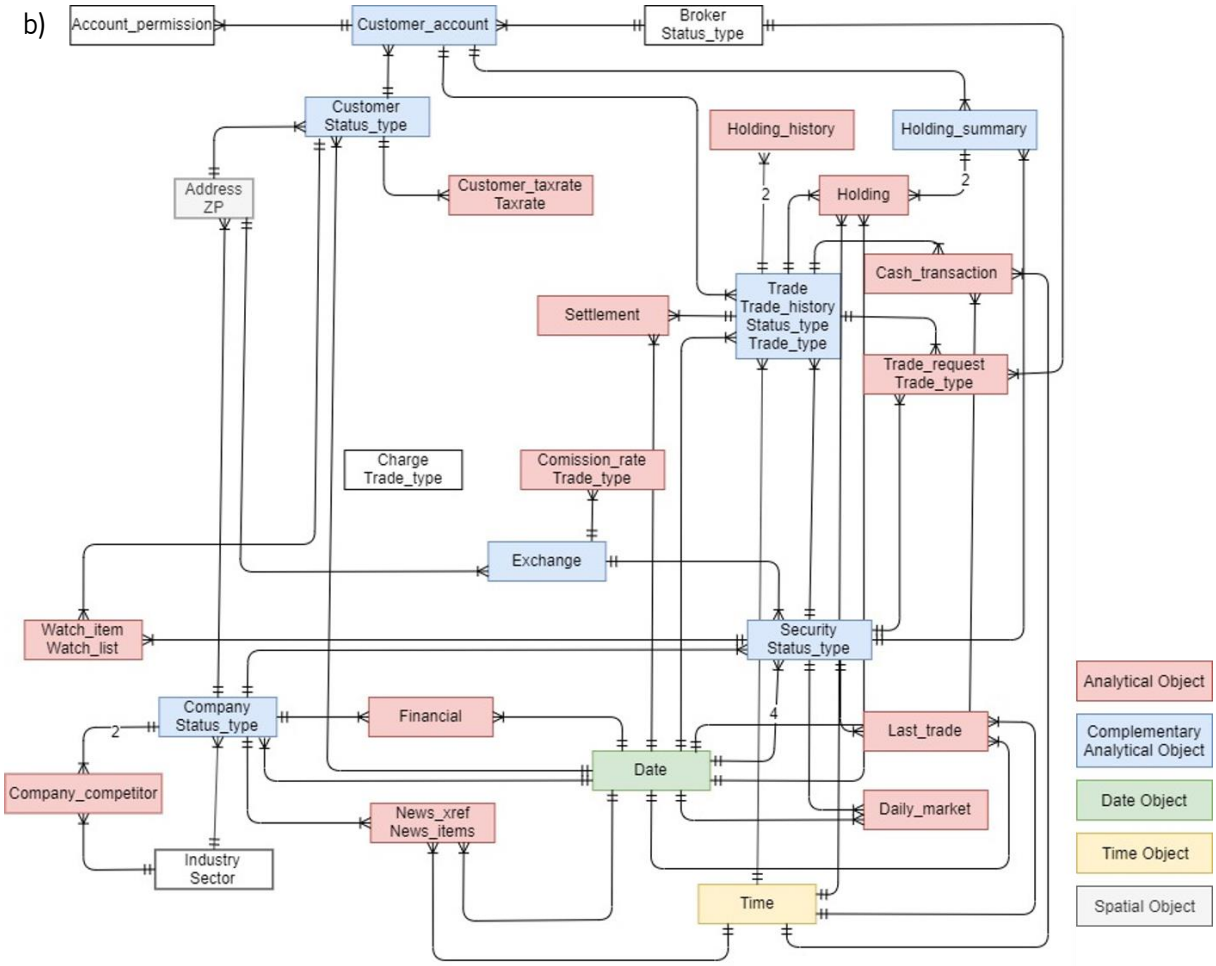



Figura 23 – TPC-E – a) Modelo Inicial vs b) Modelo após aplicação do método.

### 5.1.3 Caso TPC-DS

Esta secção tem como propósito demonstrar um exemplo que fornece detalhes específicos referentes a vendas e devoluções online, em loja e catálogo, inserido num contexto organizacional de retalho, descrito no documento TPC *Benchmark* DS[4] (TPC-DS).

Na *Figura 24* o modelo apresenta vários *analytical objects*, assim como *complementary analytical objects* inseridos no contexto de retalho, neste caso focam-se nas vendas, devoluções, promoções, clientes, itens e armazéns. As iterações do modelo estão descritas nos pontos que se seguem:

- A primeira iteração identifica a entidade *"Date"* e classifica-a como *date object;* identifica a entidade *"Time"* e classifica-a como *time object;* por fim, identifica as entidades *"Customer_demo", "Customer_add", "Household_demo"* como *spatial object;*

[4] http://www.tpc.org/tpcds/

- A R2 do método identifica os *analytical objects,* neste caso as entidades que verificam a condição de só receberem relacionamentos de muitos são *"Web_sales", "Catalog_sales", "Store_sales", "Web_returns", "Catalog_returns"* e *"Store_returns".* Por este razão, são classificadas como *analytical object;*

- A R5 do método verificou a condição dos objetos padronizáveis, que consiste na identificação de entidades que as únicas relações que recebem é de um e têm baixa cardinalidade. As entidades *"Ship_mode" e "Reason"* verificaram a condição; contudo, na verificação manual o utilizador só escolhe a entidade *"Ship_mode";* como tal, a mesma é desnormalizada de acordo com as chaves;

- Na R6 do método, a entidade *"Income_band"* verificou que as condições do único relacionamento que recebe é de um e foi desnormalizada na entidade *"Household_demo";*

- Na última iteração, são identificados os *complementary analytical objects;* desta forma, as entidades *"Catalog_page", "Web_site", "Web_page", "Call_center", "Promotion", "Customer", "Inventory"* cumprem a primeira condição de receberem, pelos menos, um relacionamento de muitos e, pelo menos, um relacionamento de um; por isso, são classificadas; as entidades *"Item", "Warehouse", "Store"* e *"Reason"* não cumprem a primeira condição e, por isso passam para o passo de verificar se só recebem relacionamentos de um e as mesmas cumprem esse passo, pelo que seguem para o passo de medir a cardinalidade; contudo, só as entidades *"Item",* e *"Store"* apresentam alta cardinalidade, pelo que são classificadas como *complementary analytical objects.* As entidades *"Warehouse"* e *"Reason"* não cumprem nenhuma condição do método, o que as deixa sem classificação;

Concluindo, o modelo após aplicação do método desnormalizou duas entidades (*"Ship_mode"* e *"Web_site"*) e apresenta 6 *analytical objects*, 10 *complementary analytical objects*, 1 *date object*, 1 *time object* e 3 *spatial objects.* O resultado das quatro iterações entre o modelo inicial e o modelo final pode ser consultado em Anexo 3 – TPC-DS.

Importa realçar que de forma a simplificar o diagrama ER apresentado na

*Figura 24*, as entidades *"Customer_demo", "Customer_add", "Household_demo"* estão dentro de um retângulo, por receberem os mesmos relacionamentos e terem um relacionamento com a entidade *"Customer"*.

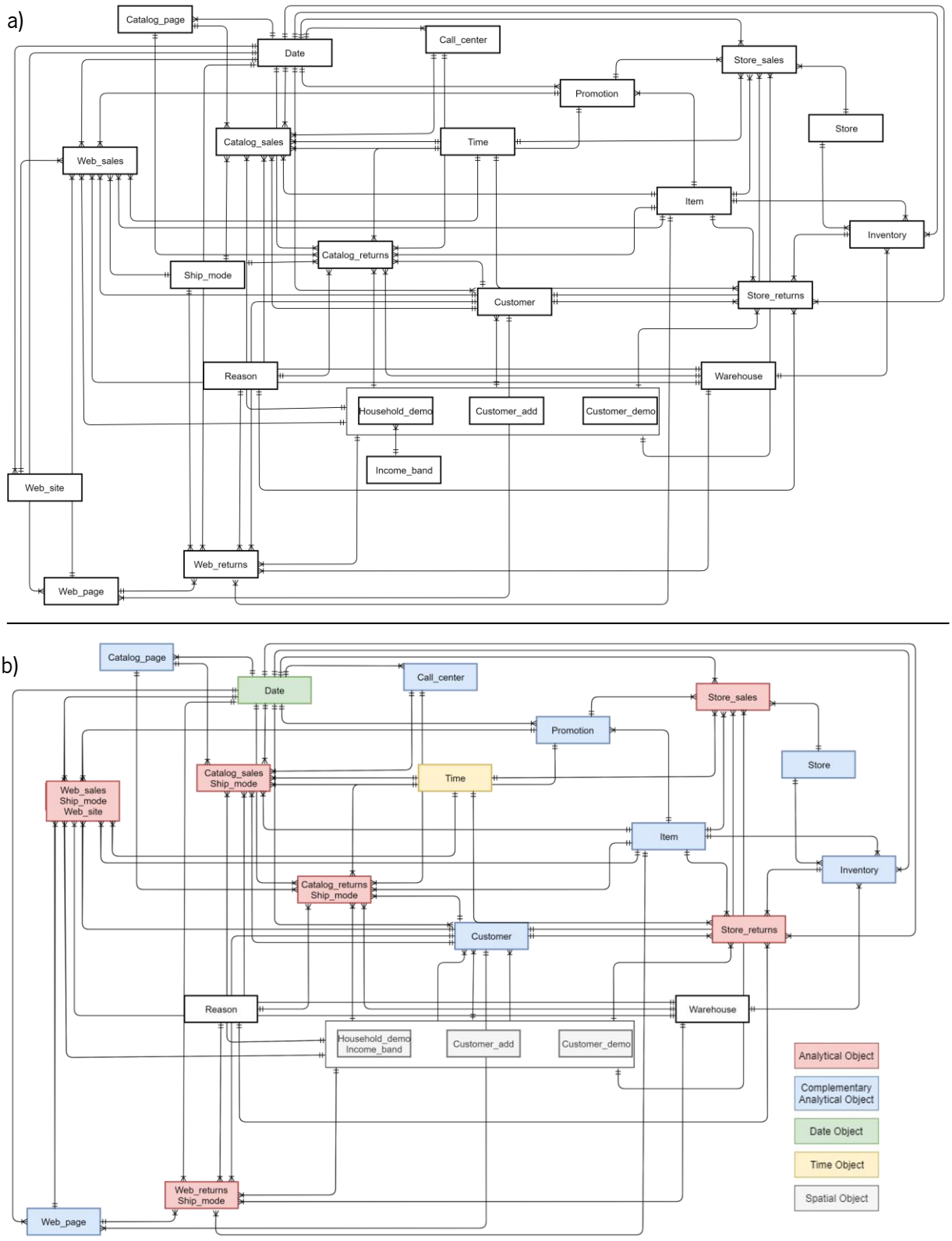



Figura 24 – TPC-DS – a) Modelo Inicial vs b) Modelo após aplicação do método.

### 5.1.4 Caso TPC-C

O TPC *Benchmark* C[5] (TPC-C) não representa uma atividade de um segmento de negócio em particular, mas sim de qualquer setor de gestão, venda ou distribuição de um produto ou serviço (por exemplo, distribuição de alimentos, aluguer de carros, etc.).

De acordo com a proposta de aplicação do método no modelo referente ao caso descrito no TPC-C, apresentado na Figura *25*, as regras aplicadas ao mesmo estão descritas nos pontos que se seguem:

- A R1 do método não identificou nenhuma entidade *"Date"* e *"Time"*; contudo, no passo referente à presença de atributos as entidades *"History", "Order"* e *"Order_line"* verificam o mesmo e, consequentemente, criam as entidades e classificaram-nas como *date object* e *time object*; por fim, a entidade *"District"* foi classificada como *spatial object;*
- Na R2, as entidades *"Stock", "History"* e *"Order_line"* verificaram a condição da mesma, pelo que foram classificadas como *analytical object*;
- A iteração a seguir corresponde à desnormalização de objetos autónomos, R6 do método, e a mesma permitiu a desnormalização da entidade *"Item"* na entidade *"Stock";*
- Por fim, a R7 do método verifica que as entidades *"Customer"* e *"Order"* cumprem o passo de receberem, pelo menos, um relacionamento de muitos e, pelo menos, um relacionamento de um; deste modo, são classificadas como *complementary analytical object*.

Após aplicação do método, as entidades *"Warehouse"* e *"New_order"* terminaram sem classificação. A entidade *"Warehouse"* verifica o passo da R6 e R7 de só receber relacionamentos de um; contudo, na R6 não é selecionada como e na R7 verifica que tem baixa cardinalidade; portanto, não é classificada como *complementary analytical object.* A entidade *"New_order"* não verifica nenhuma das regras do método, deixando em aberto possíveis melhorias ao método.

A aplicação do método neste exemplo finda com a identificação de 3 *analytical objects*, 2 *complementary analytical objects*, 1 *date object*, 1 *time object* e 1 *spatial object*. O resultado dos modelos, após aplicação da R1, R2 e R5 do método, não estão contemplados na figura abaixo; contudo, podem ser consultados em Anexo 4 – TPC-C.

[5] http://www.tpc.org/tpcc/

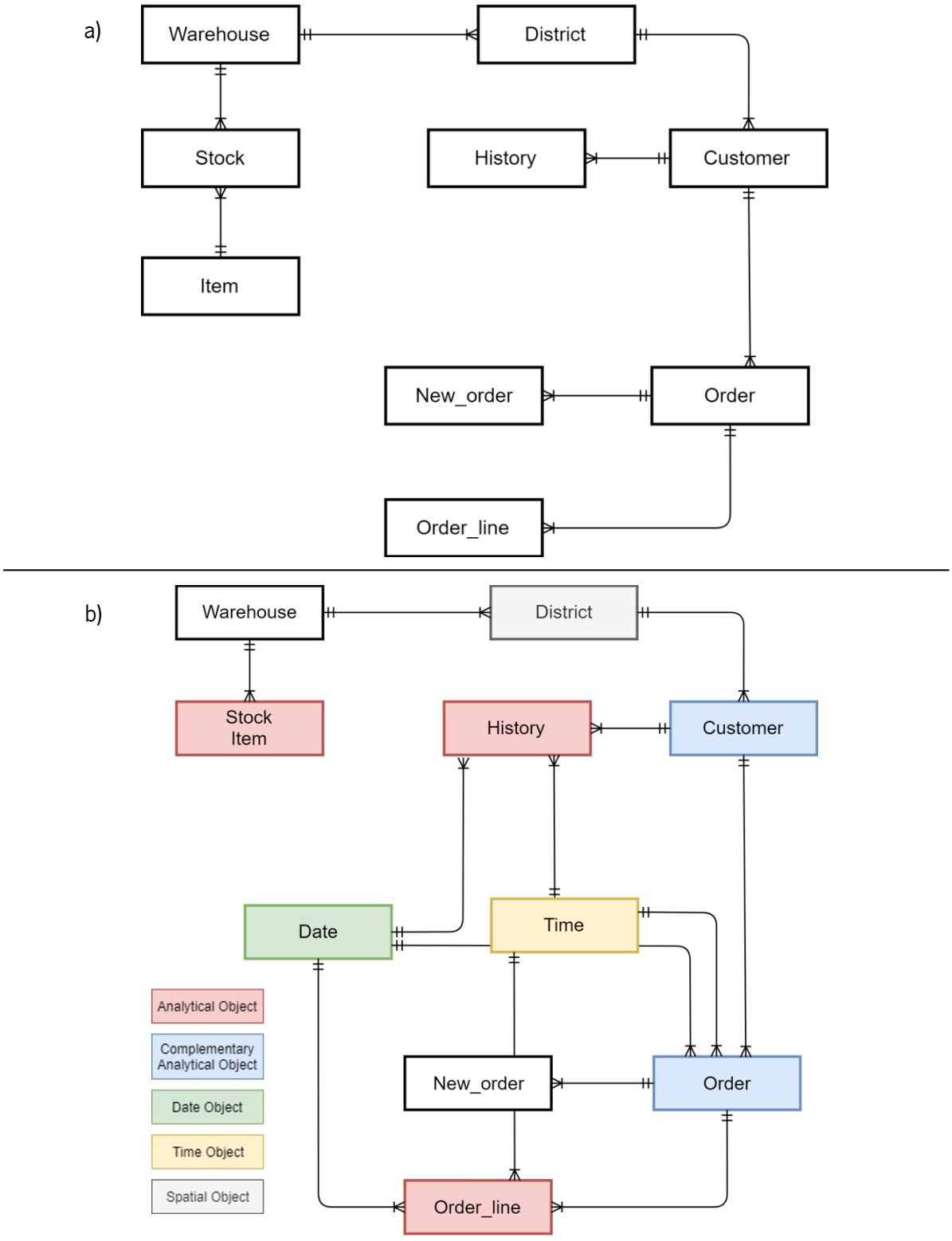



Figura 25 – TPC-C – a) Modelo Inicial vs b) Modelo após aplicação do método.

Os modelos, após aplicação do método TPC-C, TPC-DS e TPC-E, têm entidades que não estão classificadas, isto é, no modelo, após aplicação do método, as entidades podem não ter sido todas classificadas. Nestes casos cabe ao utilizador verificar se no contexto de modelação em causa aquela

entidade, mesmo não tendo cumprido nenhuma das condições das regras do método, apresente relevância ao modelo.

Em suma, a demonstração da aplicação do método nos quatro exemplos, na sua maioria relacionados com *benchmarks,* podem ser usados em contextos de *Big Data Warehousing.* O resultado, após aplicação do método no Projeto GDELT, TPC-E e TPC-DS, será usado na subsecção seguinte, de modo a validar o método.

## 5.2 Avaliação do Método

Na secção da avaliação do método, serão feitas considerações sobre os resultados dos modelos obtidos após aplicação do método, e de que forma é avaliada a qualidade, comparativamente com os resultados dos mesmos modelos concebidos pelos autores Santos et al. (2019). A avaliação dos modelos TPC-DS, TPC-E e Projeto GDELT, traduziu-se na discussão detalhada das diferenças dos resultados entre a aplicação prática do método de *Data Modelling* num exemplo em que os autores consideraram o contexto, a dimensão e casos particulares com uma proposta, desse mesmo exemplo, em que o resultado só considera a aplicação do método desenvolvido, seguindo os mesmos padrões de *design* definidos pelos autores já referidos, de regras computacionais, com vista a semiautomatização do mesmo.

Importa realçar que os diagramas ER dos modelos, após aplicação do método demonstrados no capítulo 5, foram adaptados para o mesmo formato que os autores utilizaram para conceber a proposta de abordagem do *Data Modelling*. Do mesmo modo, os modelos dos autores foram adaptados para só contemplarem *analytical objects*, *complementary analytical objects*, *date object*, *time object* e *spatial object*, visto serem os pontos de comparação com o trabalho desenvolvido nesta dissertação.

### 5.2.1 Projeto GDELT

Na fase de avaliação dos resultados do Projeto GDELT é apresentada a Figura *26* em que a) ilustra o resultado da aplicação do método de *Data Modelling* apresentado pelos autores referidos e b) representa o resultado alcançado após aplicação do método proposto nesta dissertação.

Segundo os autores já referidos, o *data model* apresentado em a) é composto por *date* e *time objects, "city"* como *spatial object* (que inclui dados desnormalizados referentes aos países) e *"event"* como *analytical object.* O *analytical object* é responsável por armazenar as notícias/eventos com dados dos eventos e dos atores envolvidos nos mesmos.

Do mesmo modo, o *data model* apresentado em b) é constituído por *date* e *time objects*, *"geo"* como *spatial object* (que inclui dados desnormalizados referentes aos países) e *"GdeltMain"* como *analytical object* (que inclui a desnormalização dos dados relativos aos atores, notícias/eventos e as fontes das mesmas).

Considerando os resultados do projeto GDELT de a) e b) da Figura *26* é possível concluir que os mesmos são semelhantes. Isto significa que o resultado do modelo do projeto GDELT apresentado pelos autores é igual ao resultado obtido após aplicação do método.

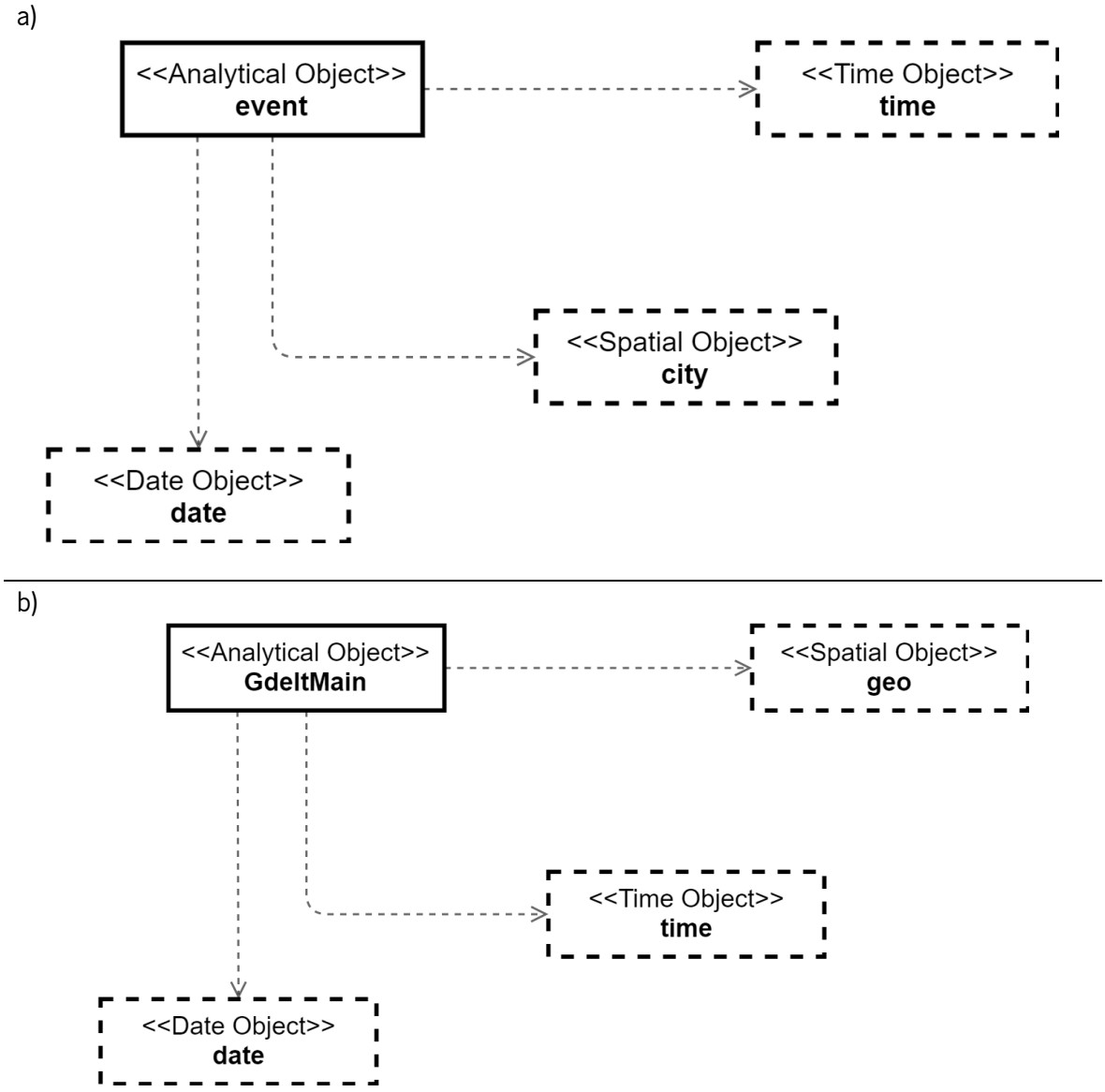



Figura 26 - Avaliação - Projeto GDELT

Através da comparação dos resultados em a) e b) do projeto GDELT, ambos propõe 1 *date object*, 1 *time object*, 1 *spatial object* e 1 *analytical object* que contêm a mesma estrutura. Assim sendo, a

aplicação, neste primeiro exemplo, da proposta de regras computacionais do método apresenta resultados similares aos resultados propostos pelos autores.

### 5.2.2 Caso TPC-E

Nesta seção serão comparados os resultados a) e b) do caso TPC-E apresentados na Figura *27*, com o objetivo de comparar o resultado a), proposto pelos autores já referenciados, com o resultado em b) do mesmo modelo, após aplicação do método proposto. Ressalva-se que a avaliação do caso TPC-E só inclui os resultados comparativamente com *data model* simplificado apresentado, isto é, os autores não representaram todos os *complementary analytical objects* no modelo a) da figura.

A proposta dos autores contempla 4 *analytical objects* (*whatch_list, trade, news_item* e *daily_market*), 3 *complementary analytical objects* (*customer_account, broker e security*), 1 *date object* e 1 *time object*.

Enquanto b), após aplicação da proposta do método, apresenta ao utilizador 13 *analytical objects* (*customer_taxrate, settlement, comisson_rate, holding, holding_history, cash_transaction, trade_request, last_trade, daily_market, financial, news_xref, company_competitor* e *watch_item*), 7 *complementary analytical objects* (*customer, customer_account, holding_summary, trade, security, company* e *exchange*), 1 *date object,* 1 time object e 1 *spatial object.*

Os autores, na demonstração do caso TPC-E, explicitam que os objetos *"customer"* e *"company"* podiam ser classificados como *complementary analytical objects;* contudo, não foram classificados devido à falta de valor analítico para o contexto específico do caso. O objeto *"customer"* foi desnormalizado em *"customer_account"* e *"company"* nos objetos *"news_xref"* e *"security"*

O modelo, com a aplicação da proposta do método, desnormalizou objetos integráveis (*watch_list*), padronizáveis (*status_type* e *trade_type*) e autónomos (*sector, ZP* e *trade_history*).

Por último, ficaram por classificar quatro entidades, incluindo a entidade *broker* que detêm atributos analíticos, mas que, e segundo as *guidelines* sugeridas pelos autores este tipo de objeto é recomendado ser criado como *complementary analytical object*.

Comparando o resultado do *Data Modelling* apresentado pelos autores, com o resultado após a aplicação da proposta do método no caso TPC-E, é possível concluir que, dos três exemplos apresentados nesta secção, é o que oferece a proposta significativamente mais ampla; por isso, vai exigir do utilizador uma análise mais cuidada do resultado, para que sejam utlizados, apenas, os objetos já classificados que respondam aos dados que o utilizador pretende caracterizar.

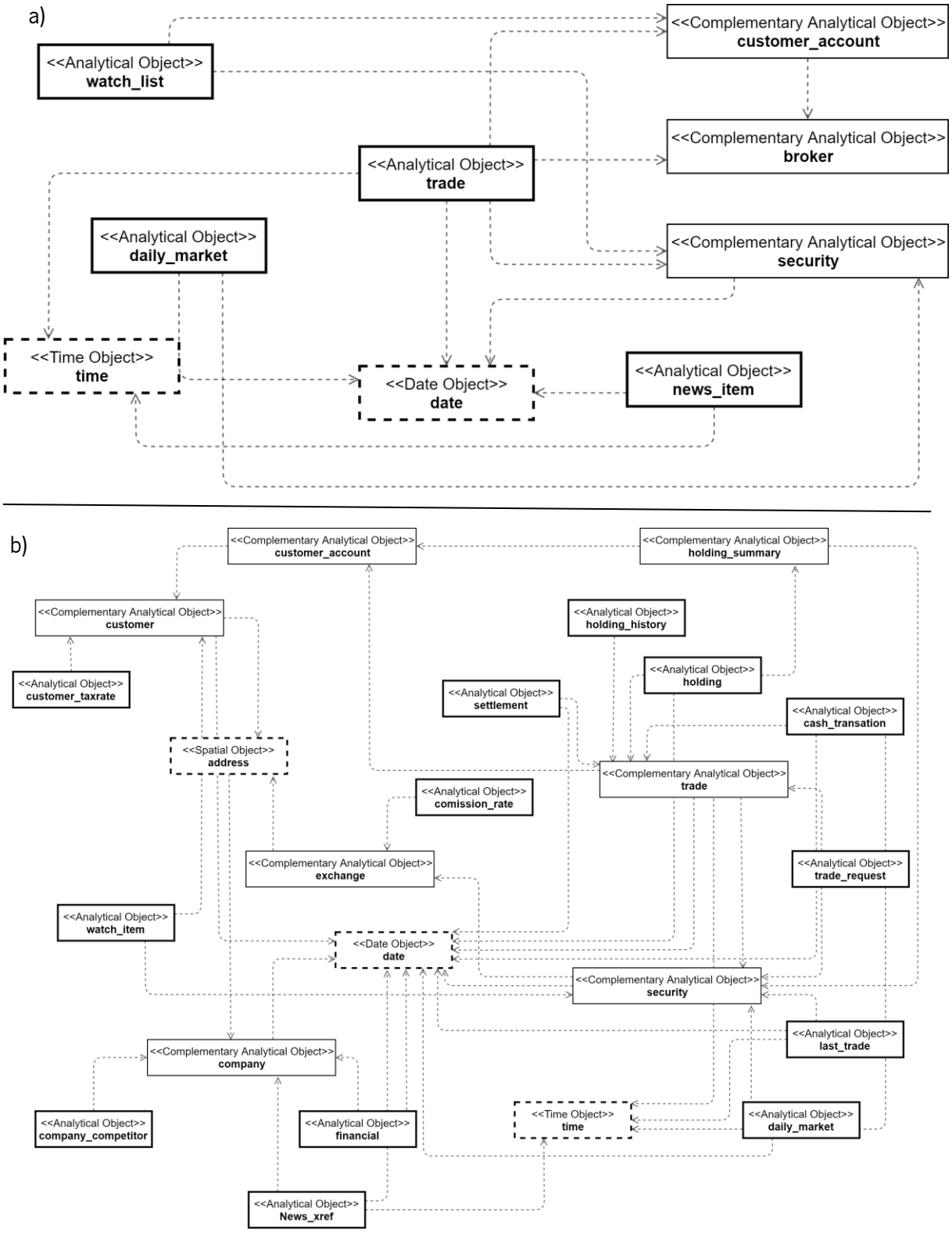



Figura 27 – Avaliação - Caso TPC-E

### 5.2.3 Caso TPC-DS

Considerando o último caso do capítulo a ser avaliado, o mesmo corresponde a um exemplo de *Big Data Warehouse* que suporta organizações de retalho com o foco em *"sales", "returns", "promotions", "customers", "items"* e *"warehouse"* descrito no TPC-DS, o mesmo está apresentado em a) e b) da *Figura 28*.

O *data model* do caso TPC-DS proposto pelos autores em a), considera a possibilidade de três prováveis *analytical objects store_sales, catalog_sales* e *web_sales* serem desnormalizados num *analytical object "sale"* particionado em tipos de *"sales".* A estrutura de *return* é semelhante à descrita anteriormente; desta forma, os três possíveis *analytical objects store_return, catalog_return* e *web_returns* são desnormalizados num *analytical object "return".* Deste modo, os autores propuseram *"sale"* como *complementary analytical object,* porque *"return"* inclui *granularity key* de "*sale".* Para além dos objetos já descritos, os autores consideraram como *analytical object "inventory"; "customer", "item", "warehouse"* e *"promotion"* como *complementary analytical object*; 1 *date object* e 1 *time object.* Em relação ao *spatial object*, os autores recomendam vivamente o seu uso; contudo, no contexto deste caso, cada cliente têm um endereço; no entanto, e com alguma frequência, as encomendas não são enviadas para o endereço padrão do cliente por este ter adicionado outro endereço à encomenda em si e não exclusivamente ao cliente. Neste exemplo apresentado, a informação do *spatial object* é granular (nome da rua, número de porta) ao invés da informação significativamente menos granular (nome da cidade e país) aconselhada pelos autores.

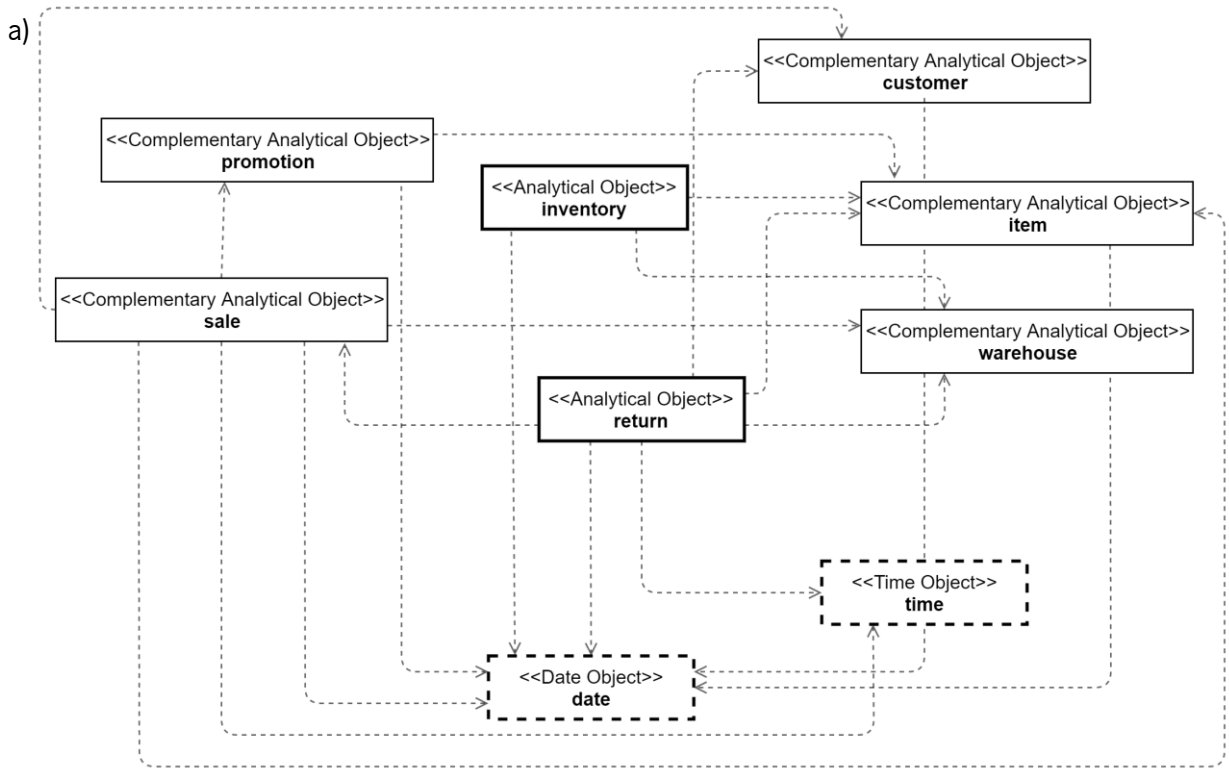
O modelo, após aplicação do método apresentado em b), contempla os três *analytical objects* de *"sales" (store, catalog e Web sales), assim* como os três *analytical objects* de *return (store, catalog e Web returns); "customer", "item", "promotion", "inventory", "store", "call_center", "catalog_page" e "web_page"* e 1 *date object,* 1 *time object* e 1 *spatial object.*

Comparando os resultados dos dois modelos apresentados na *Figura 28*, é notória a diferença do número de objetos classificados como *analytical objects* e *complementary analytical objects.* No caso dos *analytical objects,* deve-se a facto de os autores considerarem que a melhor modelação para o caso TPC-DS passa pela desnormalização das entidades *(store, catalog e Web sales)* e *(store, catalog e Web returns)* num objeto geral *"sale"* e *"return",* respetivamente. Do mesmo modo, os autores não consideraram que os objetos *"store", "call_center", "catalog_page" e "web_page",* classificados como *complementary analytical objects* na proposta do método de automatização, tivessem relevância analítica que justificasse a criação de um *complementary analytical objects.* Por último, a entidade *"warehouse"* é contemplada na modelação dos autores; contudo, a mesma não cumpre nenhuma das regras do

método proposto, porque, mesmo sendo *granularity key* de vários objetos *"warehouse",* tem baixa cardinalidade, isto é, poucos atributos; por esta razão, a entidade termina sem classificação, permitindo ao utilizador desnormalizá-la e ou classificá-la, se considerar que a mesma é relevante ou não para o caso em análise.

Existem diferenças entre os resultados dos modelos a) e b), visto que o primeiro modelo teve, de antemão, uma análise por parte dos autores relativa às características dos dados a analisar, o contexto e questões particulares do caso TPC-DS; desta forma, o resultado em a) foi concebido exclusivamente pelos autores. Enquanto no modelo b) é apresentada a proposta de modelação por meio da aplicação de regras computacionais, tendo em vista a semiautomatização do mesmo; deste modo, é apresentada ao utilizador uma proposta de modelação que o mesmo, após uma análise, pode ajustar, tendo em consideração o contexto do caso em análise, as *queries* que pretende utilizar e as características dos dados, como pode considerar que o resultado, após aplicação do método, responde a todas as questões pretendidas. Em síntese, a abordagem da proposta de modelação semiautomática é o suficiente flexível para permitir ao utilizador adaptar, se necessário, os resultados.

a)



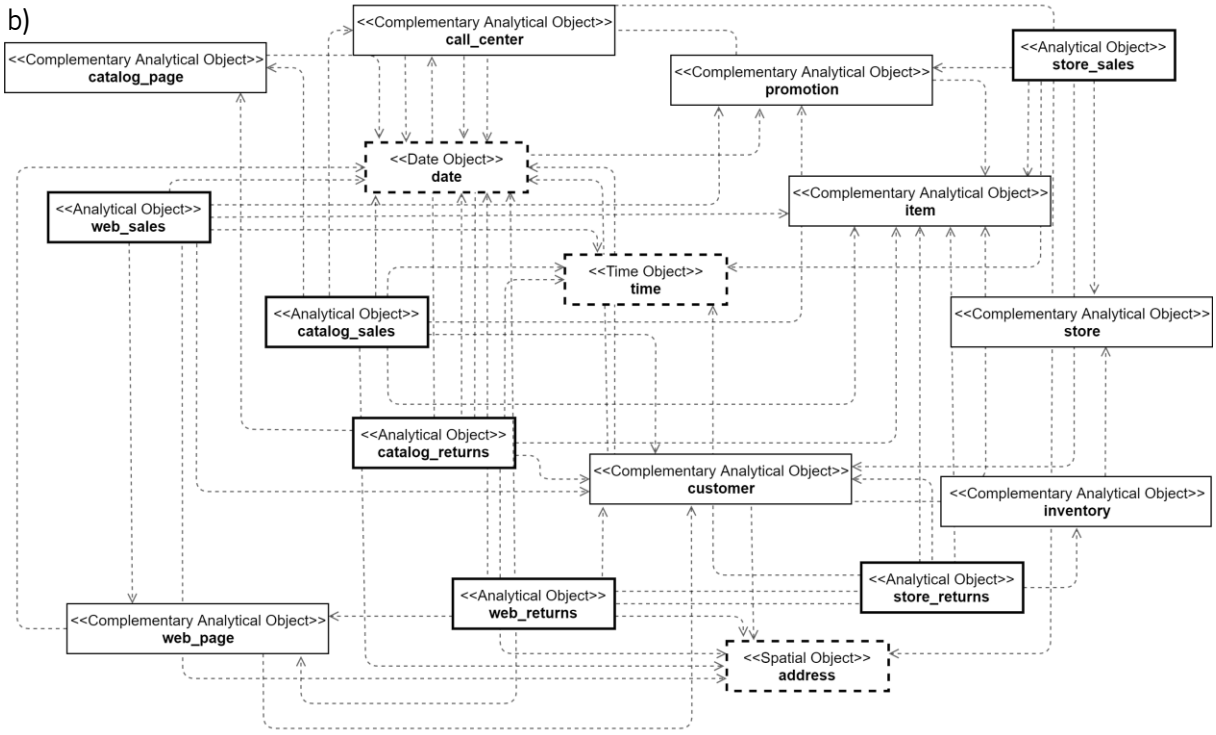



Figura 28 - Avaliação - Caso TPC-DS

A avaliação dos resultados apresentados neste capítulo permite a validação do método constituído por sete regras computacionais com a finalidade de semiautomatizar a modelação; contudo, a mesma, após apresentação do resultado, permite ao utilizador desnormalizar objetos com menos interesse analítico para o contexto e adaptar as classificações dos objetos. Deste modo, o método com regras computacionais tem em vista a semiautomatização da modelação.

Os resultados obtidos, após a aplicação do método, comparados com os resultados dos autores Santos et al. (2019), nos casos apresentados neste capítulo, permitem inferir que o método já apresenta um nível significativo de maturidade, isto é, a proposta de modelação semiautomática que executa regras, já é capaz de percorrer objetos e verificar, com significativo sucesso, a existência de *analytical objects, complementary analytical objects, date objects, time objects* e *spatial objects,* visto que os resultados, após aplicação do método, são semelhantes aos resultados apresentados pelos autores.

# 6. CONCLUSÕES

A visão geral do trabalho desenvolvido é apresentada neste capítulo final da dissertação subordinada ao tema: modelação ágil para sistemas de *Big Data Warehousing*. Neste capítulo também é descrito sucintamente o trabalho realizado e apresentado nos capítulos anteriores, assim como considerações relativas ao mesmo. Conclui, por fim, com a definição de potenciais vertentes do trabalho a explorar no futuro.

Nesta dissertação, começando por se fazer o enquadramento conceptual da área de *Big Data,* passou a definir-se o conceito, fazendo, depois, ligação da definição com o conceito de *Data Warehouse* e *Big Data Warehouse*. De seguida, foi apresentado o conceito de *Big Data Modelling,* fundamental no contexto desta dissertação; contudo, a falta de estudos científicos levou a uma pesquisa do conceito de *Data Warehouse Modelling*. São explicitadas as motivações dos autores, assim como a descrição do *Data Modelling* e a aplicação do mesmo num contexto específico. Após o enquadramento conceptual, foi delimitado o ecossistema Hadoop com foco no HDFS, Hive e bases de dados NoSQL.

Em seguida, foi apresentado o método e a ordem pela qual o mesmo deve executar as regras computacionais; deste modo, e com objetivo de facilitar a compreensão do processo inerente a cada regra do método, o mesmo foi demonstrado através de fluxogramas. Posteriormente, o método foi aplicado em diferentes casos demonstrativos. Na última secção, relativa à avaliação, foi realizada uma análise aos resultados após aplicação do método, sendo estes comparados com os resultados obtidos pelos autores, de forma a validar o trabalho desenvolvido e demonstrado no capítulo anterior.

## 6.1 Trabalho Realizado

Os autores Santos et al. (2019), motivados pela inexistência de uma abordagem estruturada que descrevesse o *design* e a implementação de *Big Data Warehouses,* desenvolveram padrões de *data modelling* para projetar estruturas de dados armazenados em sistema de *Big Data Warehouse*. Em tal questão se centra o artefacto da dissertação que consiste em desenvolver um método para a modelação de *Big Data Warehouses* constituído por um conjunto de sete regras computacionais em vista a semiautomatização da abordagem de *Data Modelling* proposta pelos autores. Para tal, foram seguidos os padrões de *data modelling* propostos pelos autores e o método foi aplicado em casos de possíveis contextos do mundo real de sistemas de *Big Data Warehouse*.

No âmbito do trabalho realizado, o método com sete regras computacionais, em vista a semiautomatização da modelação, é apresentado nos pontos a seguir indicados:

R1. Proposta de Objetos do Tipo *Date, Time* e *Spatial.*

R2. Proposta de Objetos do Tipo *Analytical Objects*.

R3. Proposta de Objetos Integráveis.

R4. Proposta de Objetos com Relacionamentos Múltiplos.

R5. Proposta de Objetos Padronizáveis.

R6. Proposta de Objetos Autónomos.

R7. Proposta de Objetos do tipo *Complementary Analytical Object*.

A demonstração da proposta do método consiste na aplicação da proposta final das sete regras do método em cinco modelos diferentes de dados de *Big Data Warehouse* de potenciais problemas do mundo real, tais como notícias e eventos mundiais, retalho, finanças e produção. A proposta do método foi aplicada em casos fictícios de *benchmarking* (TPC-E, TPC-DS, TPC-C e TPC-H) e a base de dados do projeto GDELT. Em cada um dos casos, foi aplicado o método, sendo posteriormente descritos e comparados os resultados, após a aplicação de cada regra no modelo.

A avaliação da proposta do método consistiu na discussão detalhada das diferenças dos resultados dos modelos obtidos após aplicação do método comparativamente com os resultados (dos mesmos modelos) concebidos pelos autores Santos et al. (2019). Por comparação de resultados, a avaliação dos modelos TPC-DS, TPC-E e Projeto GDELT permitiu a validação da proposta do método semisemiautomático que verifica, com significativo sucesso, a existência de *analytical objects, complementary analytical objects, date objects, time objects* e *spatial objects.*

Em suma, ao longo da história o conceito de *Data Warehouse* teve um papel analítico de tal modo significativo nas organizações que, ainda hoje, um número considerável de indústrias utilizam modelos tradicionais e sistemas de *Data Warehousing e Data Modelling* como estratégia. A proposta do método apresentado neste trabalho constitui uma mais-valia na transição de uma abordagem tradicional para uma abordagem de *Data Modelling* em contextos de *Big Data Warehousing.* Deste modo, e com a ressalva que no decorrer do processo do método é necessária uma verificação manual, por parte do utilizador, a aplicação das sete regras, num modelo de dados tradicional, de modo automático propõe um modelo de dados que segue padrões de *design* e de implementação em *Big Data Warehouses.*

Após o término da dissertação e depois de uma análise cuidada aos resultados obtidos por comparação, na secção de avaliação, entre os resultados alcançados manualmente por Santos et al. (2019) e os resultados após aplicação da proposta do método, é possível concluir que os objetivos

propostos no capítulo 1, relativo à Introdução, foram atingidos; como tal, este trabalho é um contributo para uma área de *Data Modelling* em contextos de *Big Data Warehousing*.

## 6.2 Trabalhos Futuros

Com o término da dissertação, ficam alguns âmbitos que poderão ser explorados em trabalhos futuros, com o objetivo de acrescentar e continuar o estudo científico numa área significativamente recente. Tal como descrito na secção 2.3.2, no método de *Data Modelling* proposto por Santos & Costa, (2019) são considerados outros tipos de objetos que não apenas *analytical objects, complementary analytical object, date object, time object* e *spatial object*. Apesar destes serem os conceitos principais da modelação da sua proposta para *Big Data Warehousing*, uma futura investigação poderá incidir sobre os conceitos *materialized objects, granularity keys, atomic values, collections, partition keys* e *bucketing/clustering key*, de forma a expandir o método resultante desta dissertação.

Além disso, destaca-se a necessidade de continuar a ajustar cada uma das regras, aplicando-as em exemplos de diferentes contextos aos já apresentados neste trabalho, facultando, assim, a possibilidade de encontrar padrões que permitam completar o método com regras que proponham verificar e classificar os conceitos em falta, de forma a que os resultados do modelo de dados sejam o mais completos possível.

O método é constituído por sete regras computacionais, tendo em vista a semiautomatização da modelação; deste modo, é relevante a discussão do aspeto "semi" do método semiautomatizado, tendo em conta os resultados da discussão é pertinente a transição da proposta conceptual do método para uma aplicação que apresente ao utilizador uma proposta de modelação ágil que cumpram os padrões de *design* propostos pelos autores.

## Referências Bibliográficas


Alekseev, A. A., Osipova, V. V., Ivanov, M. A., Klimentov, A., Grigorieva, N. V., & Nalamwar, H. S. (2016). Efficient data management tools for the heterogeneous big data warehouse. *Physics of Particles and Nuclei Letters*, *13*(5), 689–692. https://doi.org/10.1134/S1547477116050022

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599–616. https://doi.org/10.1016/j.future.2008.12.001

Capriolo, E., Wampler, D., & Rutherglen, J. (2012). *Programming Hive: Data Warehouse and Query Language for Hadoop*. O'Reilly Media, Inc.

Cassavia, N., Dicosta, P., Masciari, E., & Saccà, D. (2014). Data Preparation for Tourist Data Big Data Warehousing: *Proceedings of 3rd International Conference on Data Management Technologies and Applications*, 419–426. https://doi.org/10.5220/0005144004190426

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, *39*(4), 12. https://doi.org/10.1145/1978915.1978919

Chandarana, P., & Vijayalakshmi, M. (2014). Big Data analytics frameworks. *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, 430–434. https://doi.org/10.1109/CSCITA.2014.6839299

Costa, C., Andrade, C., & Santos, M. Y. (2018). Big Data Warehouses for Smart Industries. Em S. Sakr & A. Zomaya (Eds.), *Encyclopedia of Big Data Technologies* (pp. 1–11). https://doi.org/10.1007/978-3-319-63962-8_204-1

Costa, C., & Santos, M. Y. (2018). Evaluating Several Design Patterns and Trends in Big Data Warehousing Systems. Em J. Krogstie & H. A. Reijers (Eds.), *Advanced Information Systems Engineering* (Vol. 10816, pp. 459–473). https://doi.org/10.1007/978-3-319-91563-0_28

Costa, E., Costa, C., & Santos, M. Y. (2017). Efficient Big Data Modelling and Organization for Hadoop Hive-Based Data Warehouses. Em M. Themistocleous & V. Morabito (Eds.), *Information Systems* (Vol. 299, pp. 3–16). https://doi.org/10.1007/978-3-319-65930-5_1

DB-Engines Ranking. (2019). Obtido 30 de Setembro de 2019, de DB-Engines website: https://db-engines.com/en/ranking

De Mauro, A., Greco, M., & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library Review*, *65*(3), 122–135. https://doi.org/10.1108/LR-06-2015-0061

Du, D. (2018). *Apache Hive Essentials: Essential techniques to help you process, and get unique insights from, big data, 2nd Edition*. Packt Publishing Ltd.

Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data analysis. *National Science Review*, *1*(2), 293–314. https://doi.org/10.1093/nsr/nwt032

Gali, H., & Henk, F. M. (2012). The Evolution of Big Data as a Research and Scientific Topic: Overview of the Literature - Research Trends. Obtido 4 de Fevereiro de 2019, de https://www.researchtrends.com/issue-30-september-2012/the-evolution-of-big-data-as-a-research-and-scientific-topic-overview-of-the-literature/

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

GDELT. (2018). Obtido 15 de Julho de 2019, de The GDELT Project website: https://www.gdeltproject.org/

Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL database systems: A survey and decision guidance. *Computer Science - Research and Development*, *32*(3), 353–365. https://doi.org/10.1007/s00450-016-0334-3

Golab, L., & Johnson, T. (2014). Data stream warehousing. *2014 IEEE 30th International Conference on Data Engineering*, 1290–1293. https://doi.org/10.1109/ICDE.2014.6816763

Golfarelli, M., Rizzi, S., & Pagliarani, C. (2009). *Data warehouse design: Modern principles and methodologies*. New York, NY: McGraw-Hill.

Goss, R. G., & Veeramuthu, K. (2013). Heading towards big data building a better data warehouse for more data, more speed, and more users. *ASMC 2013 SEMI Advanced Semiconductor Manufacturing Conference*, 220–225. https://doi.org/10.1109/ASMC.2013.6552808

Gupta, A. (2015). Big Data analysis using Computational Intelligence and Hadoop: A study. *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1397–1401.

He, L., Chen, Y., Meng, N., & Liu, L. Y. (2011). An Ontology-Based Conceptual Modeling Method for Data Warehouse. *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, 130–133. https://doi.org/10.1109/ICM.2011.171

Holmes, A. (2012). *Hadoop in Practice*. Greenwich, CT, USA: Manning Publications Co.

Inmon, W. H. (2005). *Building the Data Warehouse: Getting Started ()*. 19.

Jing Han, Haihong E, Guan Le, & Jian Du. (2011). Survey on NoSQL database. *2011 6th International Conference on Pervasive Computing and Applications*, 363–366. https://doi.org/10.1109/ICPCA.2011.6106531

Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big Data: Issues and Challenges Moving Forward. *2013 46th Hawaii International Conference on System Sciences*, 995–1004. https://doi.org/10.1109/HICSS.2013.645

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, 1–5. https://doi.org/10.1109/ASEEZone1.2014.6820689

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd edition). Indianapolis, Ind: Wiley.

Krishnan, K. (2013). *Data Warehousing in the Age of Big Data*. Newnes.

Labrinidis, A., & Jagadish, H. V. (sem data). *Challenges and opportunities with big data*. 2.

Luján-Mora, S., Trujillo, J., & Song, I.-Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, *59*(3), 725–769. https://doi.org/10.1016/j.datak.2005.11.004

Luo, Y., Luo, S., Guan, J., & Zhou, S. (2013). A RAMCloud Storage System based on HDFS: Architecture, implementation and evaluation. *Journal of Systems and Software*, *86*(3), 744–750. https://doi.org/10.1016/j.jss.2012.11.025

Mackey, G., Sehrish, S., & Wang, J. (2009). Improving metadata management for small files in HDFS. *2009 IEEE International Conference on Cluster Computing and Workshops*, 1–4. https://doi.org/10.1109/CLUSTR.2009.5289133

Mathur, A., Sihag, A., Bagaria, E. G., & Rajawat, S. (2014). A new perspective to data processing: Big Data. *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, 110–114. https://doi.org/10.1109/IndiaCom.2014.6828111

Mohanty, S., Jagadeesh, M., & Srivatsa, H. (2013). Big Data Imperatives: Enterprise 'Big Data' Warehouse, 'BI' Implementations ... - Soumendra Mohanty, Madhu Jagadeesh, Harsha Srivatsa— Google Books. Obtido 30 de Dezembro de 2018, de https://books.google.pt/books?hl=en&lr=&id=FP4RMWVDqrMC&oi=fnd&pg=PP3&dq=Big+data+Imperatives:+Enterprise+Big+Data+Warehouse&ots=jayuDShNqG&sig=aVsOWRwtJrU2yn31y8xBnffowdw&redir_esc=y#v=onepage&q=Big%20data%20Imperatives%3A%20Enterprise%20Big%20Data%20Warehouse&f=false

Nikolov, N. (2007). *A new look into data warehouse modelling*. *DISI*, 540–543. Obtido de Scopus.

NIST Big Data Public Working Group Definitions and Taxonomies Subgroup. (2015). *NIST Big Data Interoperability Framework: Volume 1, Definitions* (N. NIST SP 1500-1). https://doi.org/10.6028/NIST.SP.1500-1

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Philip Chen, C. L., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Santos, M. Y., & Costa, C. (2019). *Big Data: Concepts, Warehousing and Analytics*.

Santos, Maribel Yasmina, & Costa, C. (2016). Data Warehousing in Big Data: From Multidimensional to Tabular Data Models. *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering - C3S2E '16*, 51–60. https://doi.org/10.1145/2948992.2949024

Santos, Maribel Yasmina, Martinho, B., & Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics*, *4*(2), 111–129. https://doi.org/10.1080/23270012.2017.1304292

Shafer, J., Rixner, S., & Cox, A. L. (2010). The Hadoop distributed filesystem: Balancing portability and performance. *2010 IEEE International Symposium on Performance Analysis of Systems Software (ISPASS)*, 122–133. https://doi.org/10.1109/ISPASS.2010.5452045

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. https://doi.org/10.1109/MSST.2010.5496972

Šuman, S., Jakupović, A., & Kuljanac, F. G. (2016). Knowledge-Based Systems for Data Modelling: *International Journal of Enterprise Information Systems*, *12*(2), 1–13. https://doi.org/10.4018/IJEIS.2016040101

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Zhang, N., … Murthy, R. (2010). Hive—A petabyte scale data warehouse using Hadoop. *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 996–1005. https://doi.org/10.1109/ICDE.2010.5447738

TPC-C - Homepage. (2010). Obtido 15 de Julho de 2019, de http://www.tpc.org/tpcc/

TPC-DS - Homepage. (2019). Obtido 15 de Julho de 2019, de http://www.tpc.org/tpcds/

TPC-E - Homepage. (2018). Obtido 15 de Julho de 2019, de http://www.tpc.org/tpce/

TPC-H - Homepage. (2018). Obtido 15 de Julho de 2019, de http://www.tpc.org/tpch/

Tria, F. Di, Lefons, E., & Tangorra, F. (2014). Design process for Big Data Warehouses. *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 512–518. https://doi.org/10.1109/DSAA.2014.7058120

Tria, Francesco Di, Lefons, E., & Tangorra, F. (2014). Big Data Warehouse Automatic Design Methodology. *Big Data Management, Technologies, and Applications*, 115–149. https://doi.org/10.4018/978-1-4666-4699-5.ch006

Tria, Francesco Di, Lefons, E., & Tangorra, F. (2018). A Framework for Evaluating Design Methodologies for Big Data Warehouses: Measurement of the Design Process. *International Journal of Data Warehousing and Mining (IJDWM)*, *14*(1), 15–39. https://doi.org/10.4018/IJDWM.2018010102

Zhou, Q., & Xiao, Q. (2009). The Study on Data Warehouse Modelling and OLAP for Highway Management. *2009 International Conference on Measuring Technology and Mechatronics Automation*, *2*, 416–419. https://doi.org/10.1109/ICMTMA.2009.587

Zikopoulos, P., & Eaton, C. (Eds.). (2011). *Understanding big data: Analytics for enterprise class Hadoop and streaming data;* New York, NY: McGraw-Hill.

# ANEXOS

## Anexo 1 – Projeto GDELT

A *Figura 29*, Figura *30*, Figura *31* apresentam a aplicação da R1, R2 e R6 do método no modelo do Projeto GDELT.

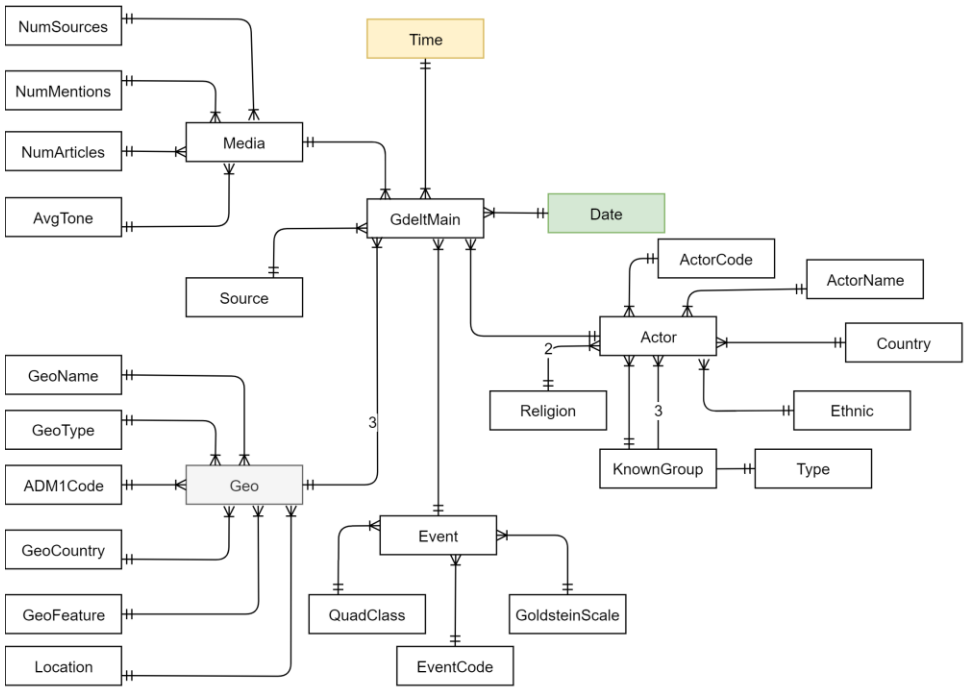



Figura 29 – Projeto GDELT após aplicação da R1.

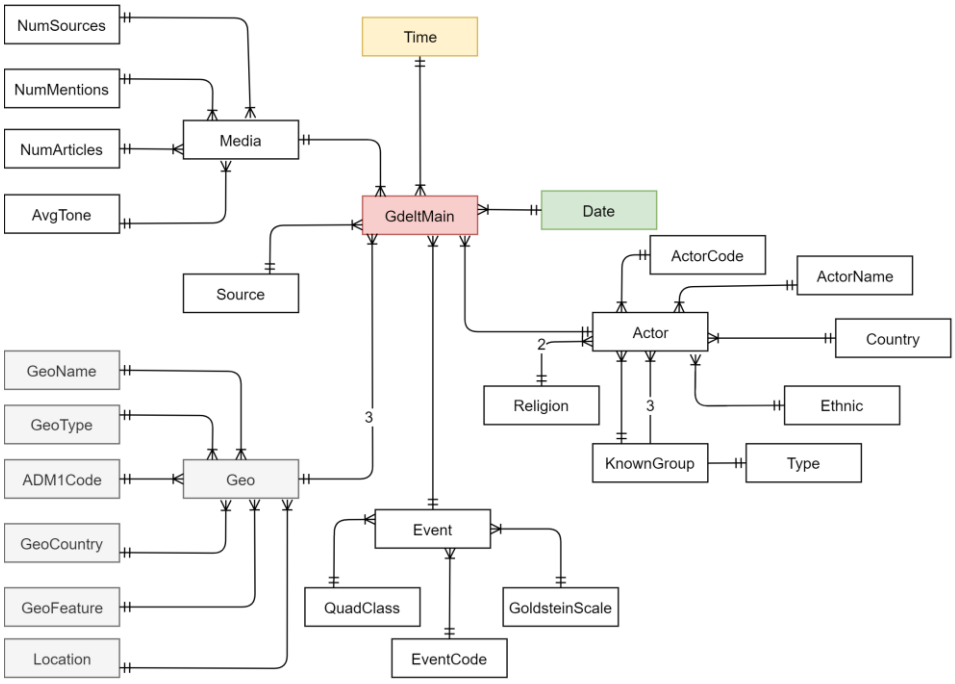



Figura 30 - Projeto GDELT após aplicação da R2.

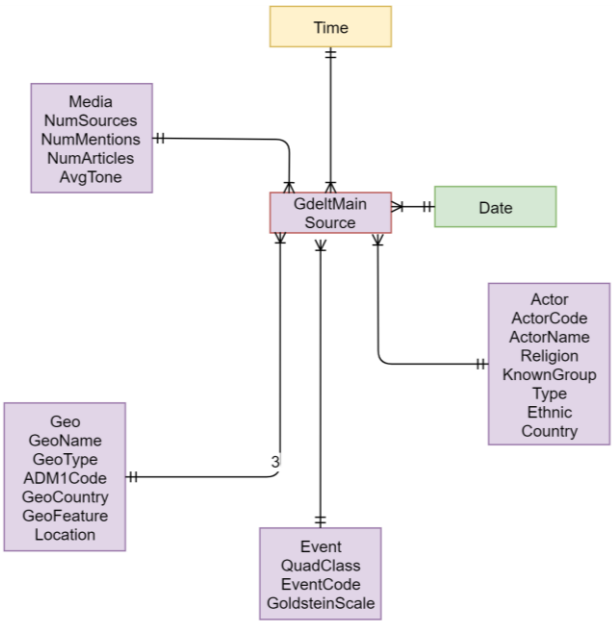



Figura 31 - Projeto GDELT após aplicação da R2.

## Anexo 2 – TPC-E

A Figura *32*, Figura *33*, Figura *34*, Figura *35*, Figura *36* apresentam a aplicação da R1, R2, R3, R5 e R6 do método no modelo do caso de demonstração TPC-E.

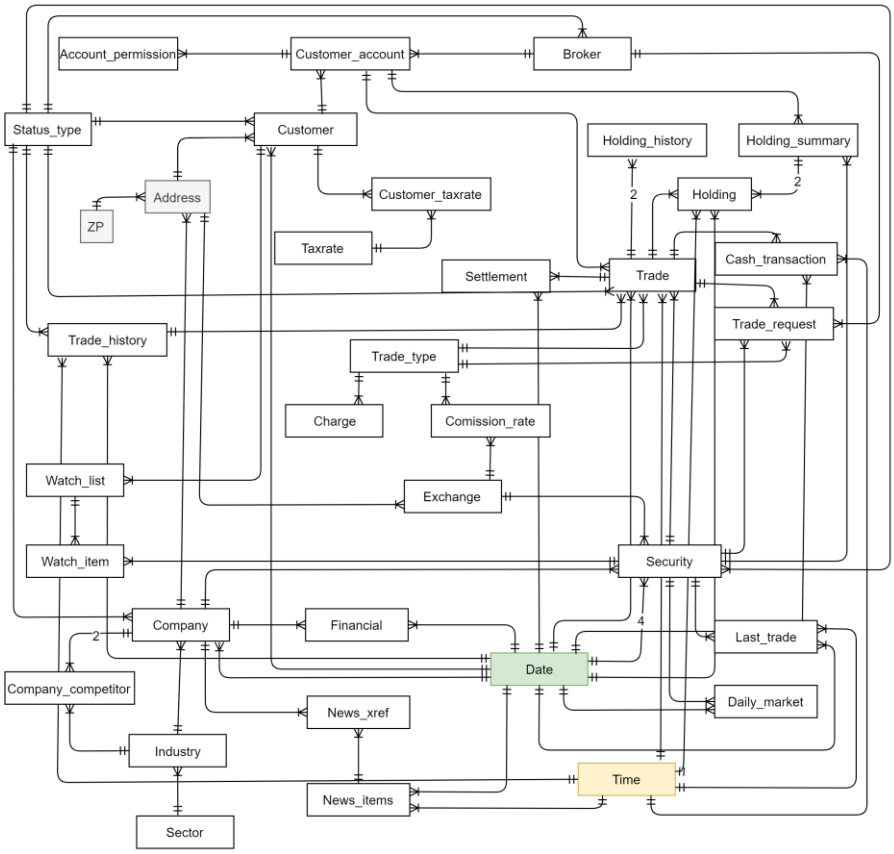



Figura 32 – TPC-E após aplicação da R1.

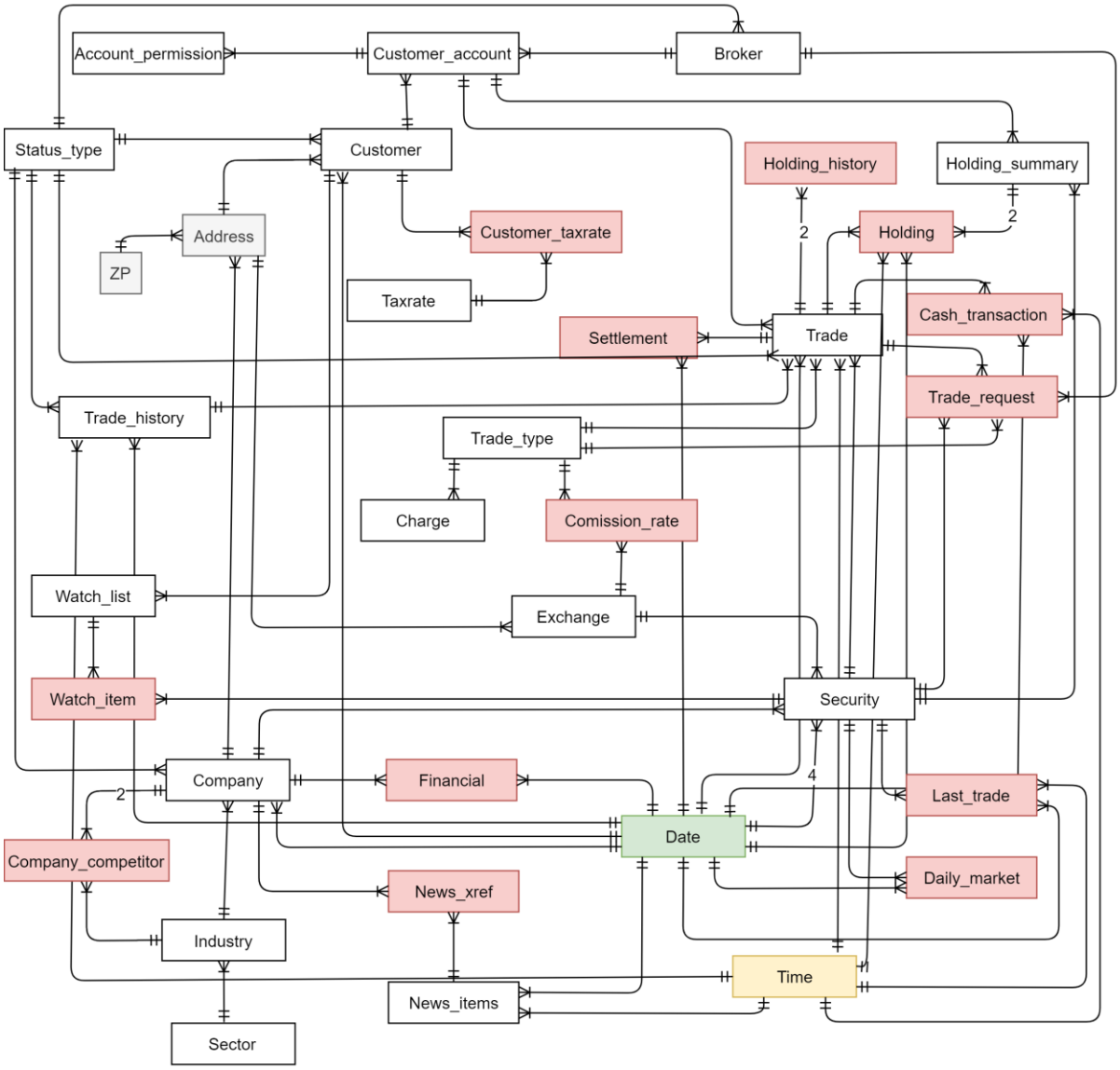



Figura 33 - TPC-E após aplicação da R2.

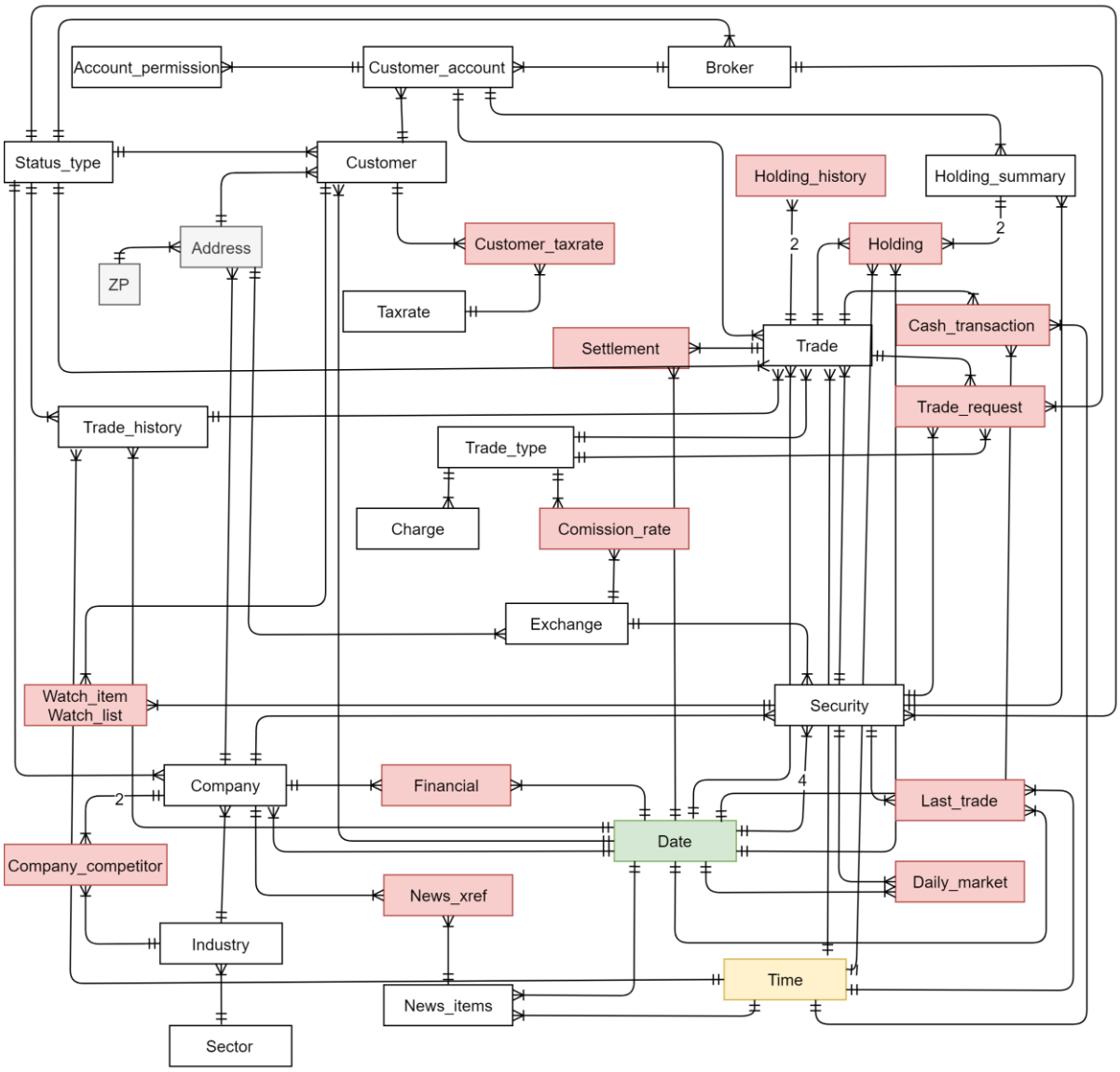



Figura 34 – TPC-E após aplicação da R3.

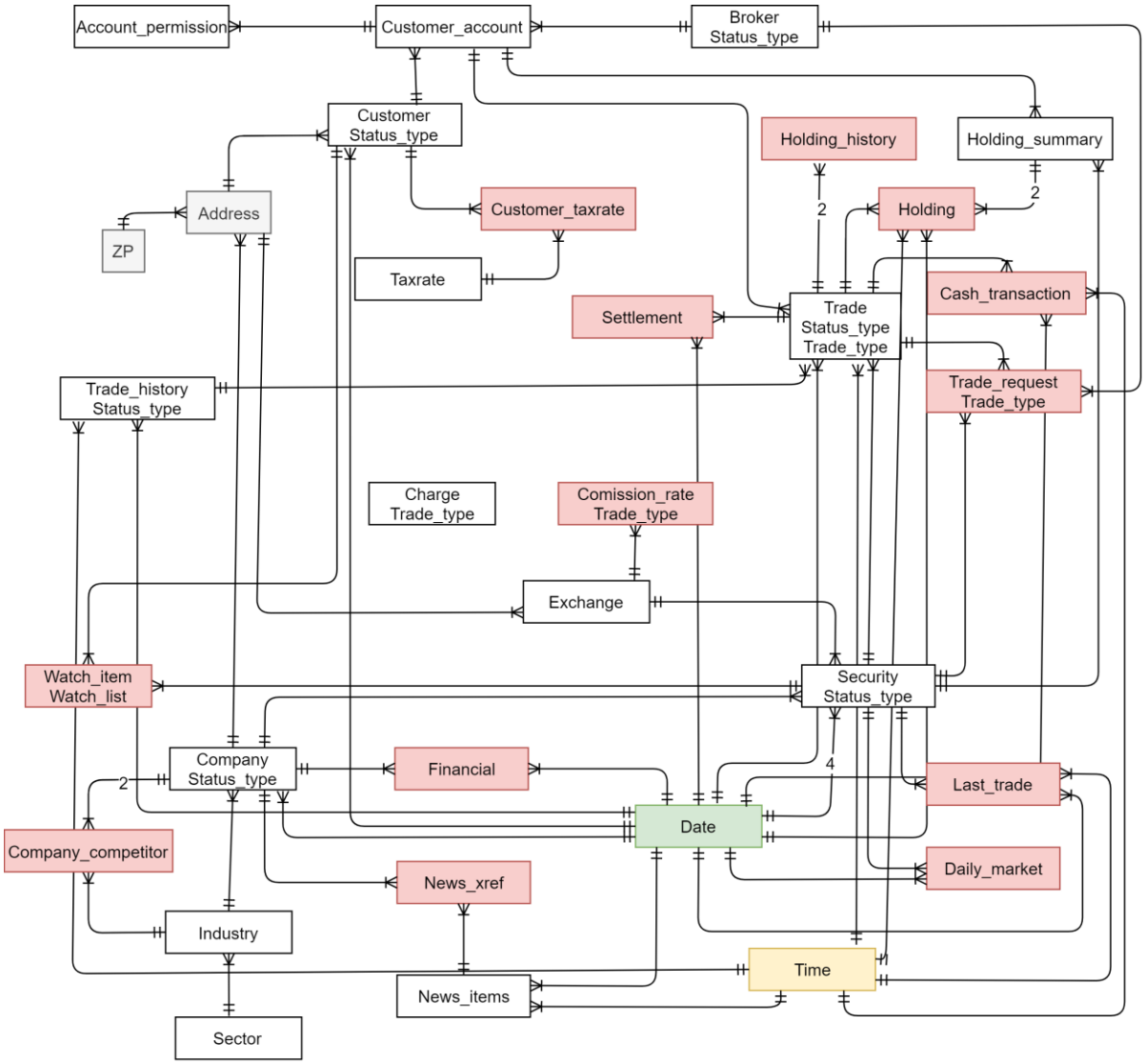



Figura 35 – TPC-E após aplicação da R5.

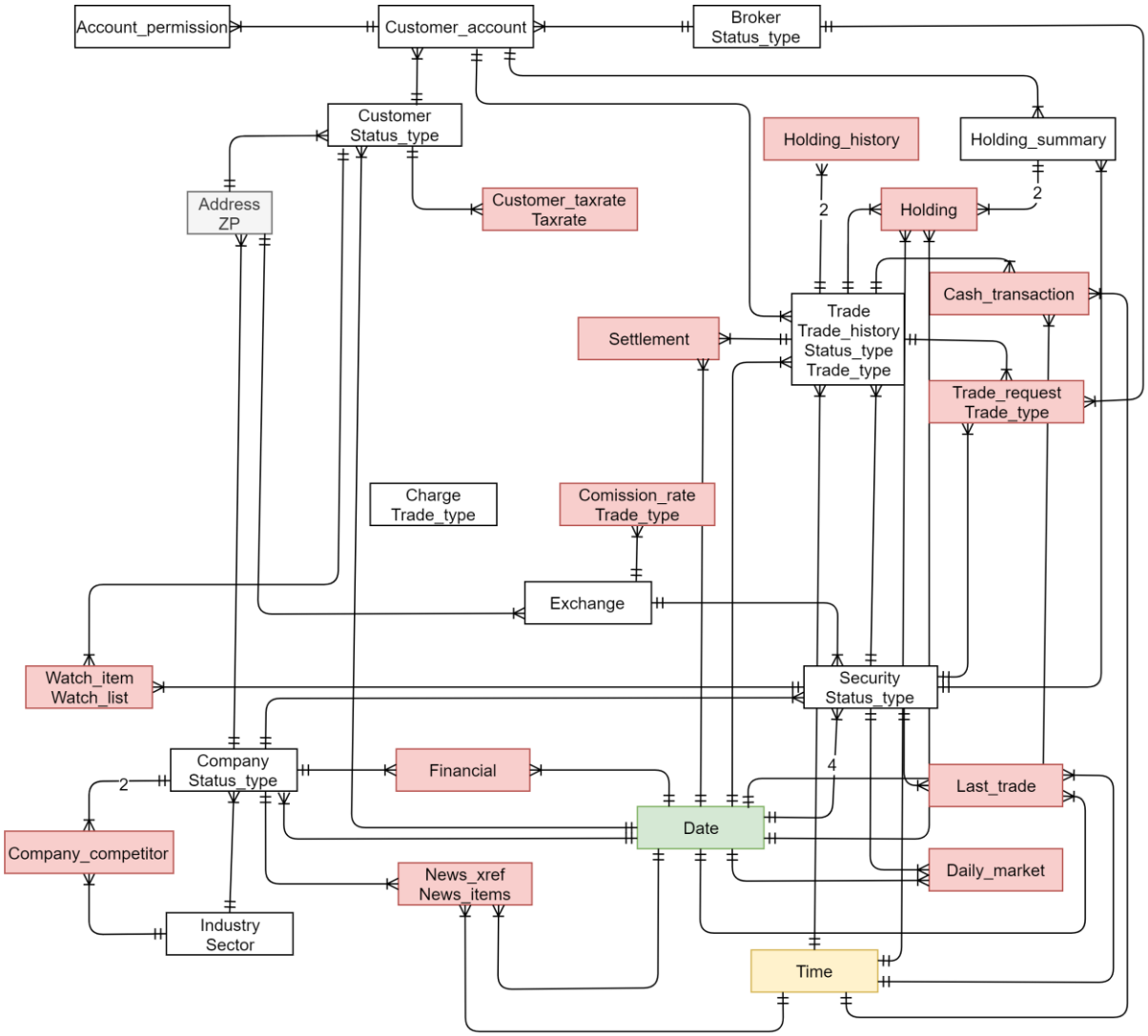



Figura 36 – TPC-E após aplicação da R6.

## Anexo 3 – TPC-DS

A Figura *37*, Figura *38*, Figura *39*, Figura *40* apresentam a aplicação da R1, R2, R5 e R6 do método no modelo do caso de demonstração TPC-DS.

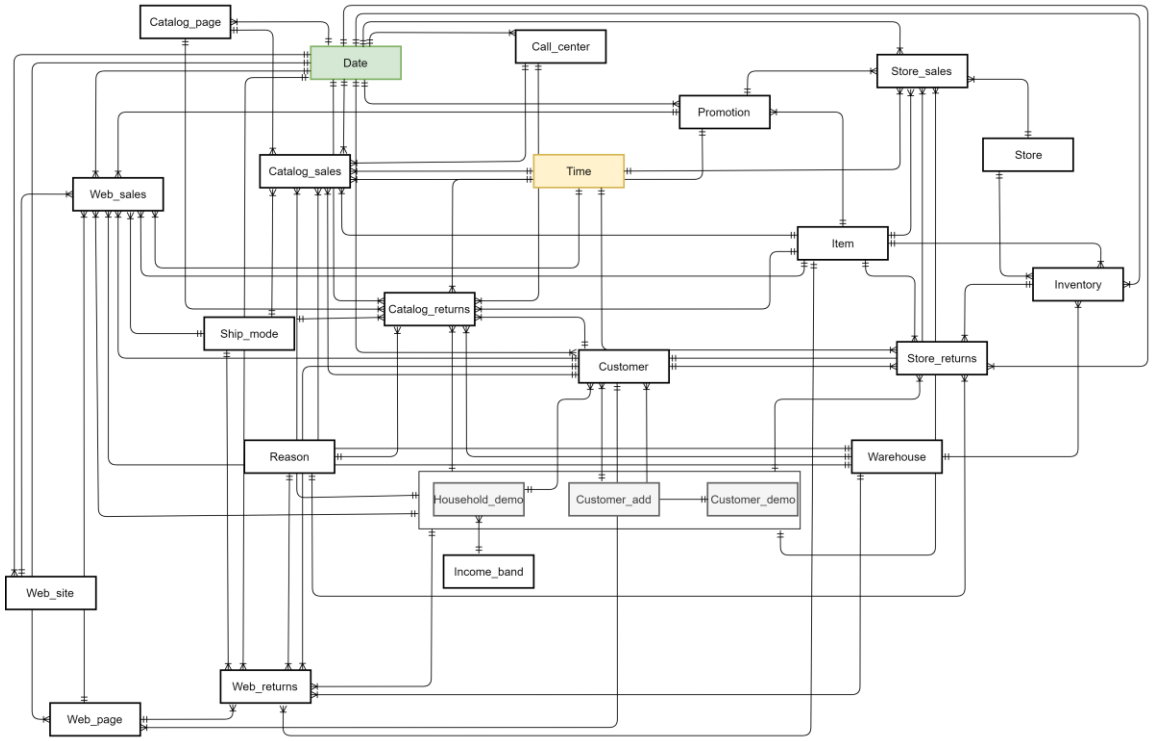



Figura 37 – TPC-DS após aplicação da R1.

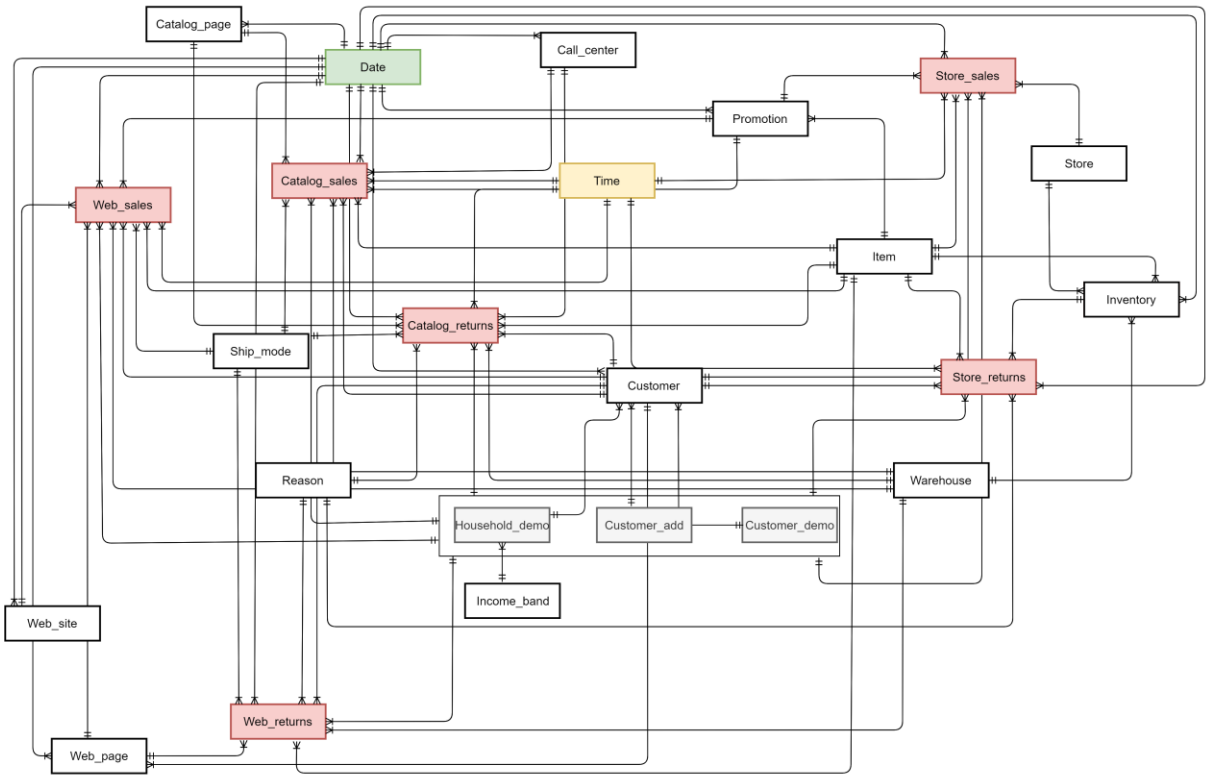



Figura 38 – TPC-DS após aplicação da R2.

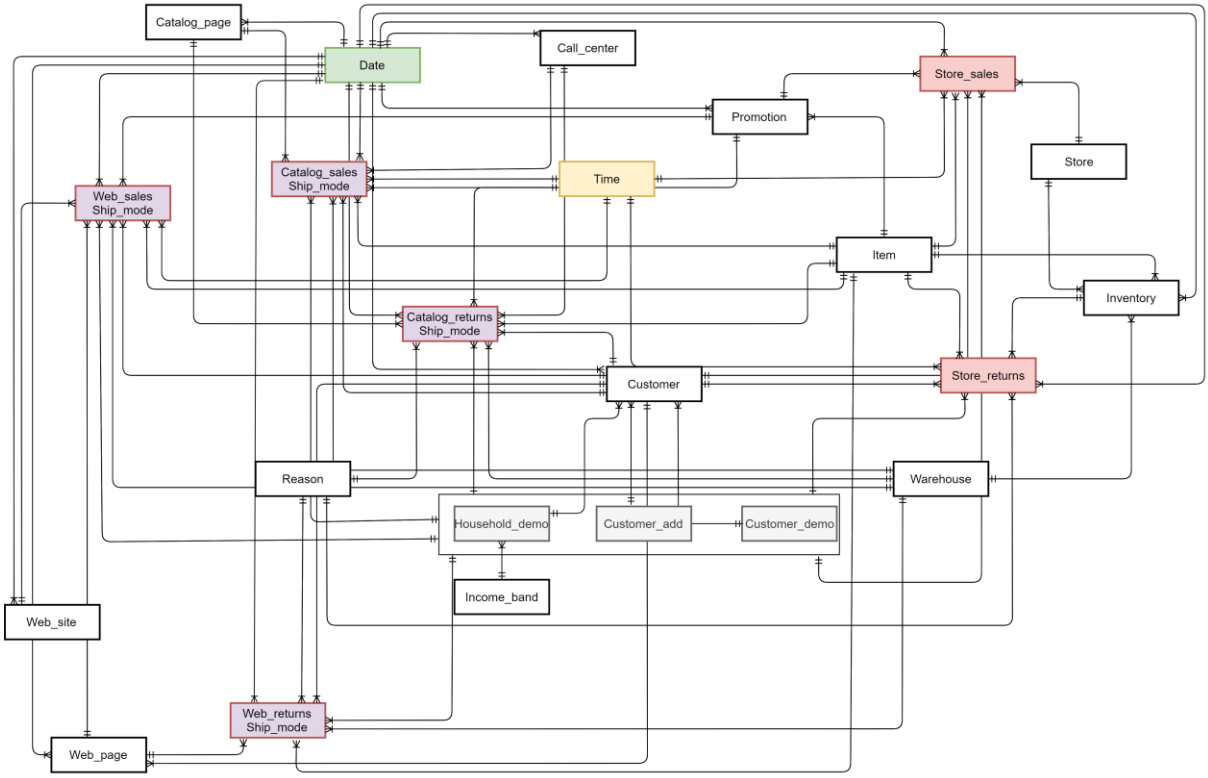



Figura 39 – TPC-DS após aplicação da R5.

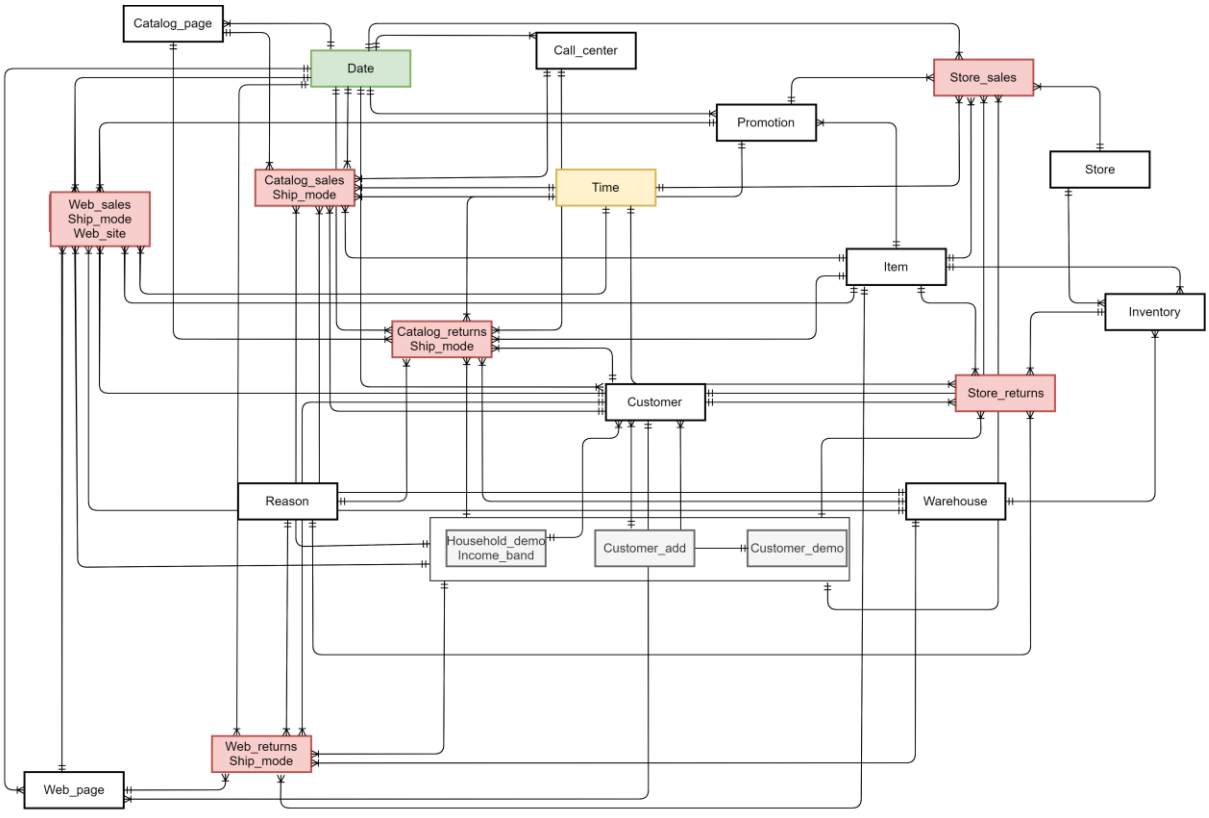



Figura 40 – TPC-DS após aplicação da R6.

## Anexo 4 – TPC-C

A Figura *41*, Figura *42*, Figura *43* apresentam a aplicação da R1, R2 e R6 do método no modelo do caso de demonstração TPC-C.

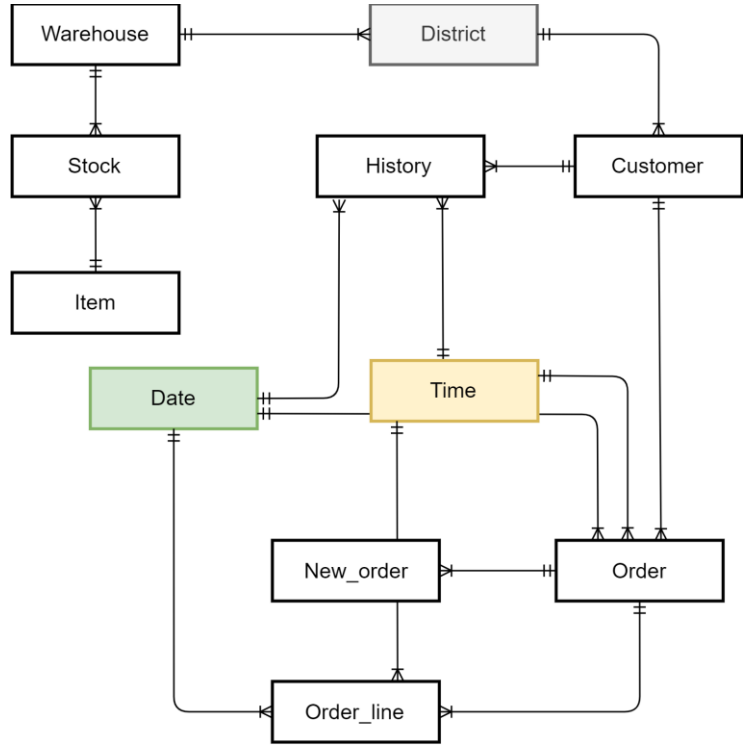



Figura 41 – TPC-C após aplicação da R1.

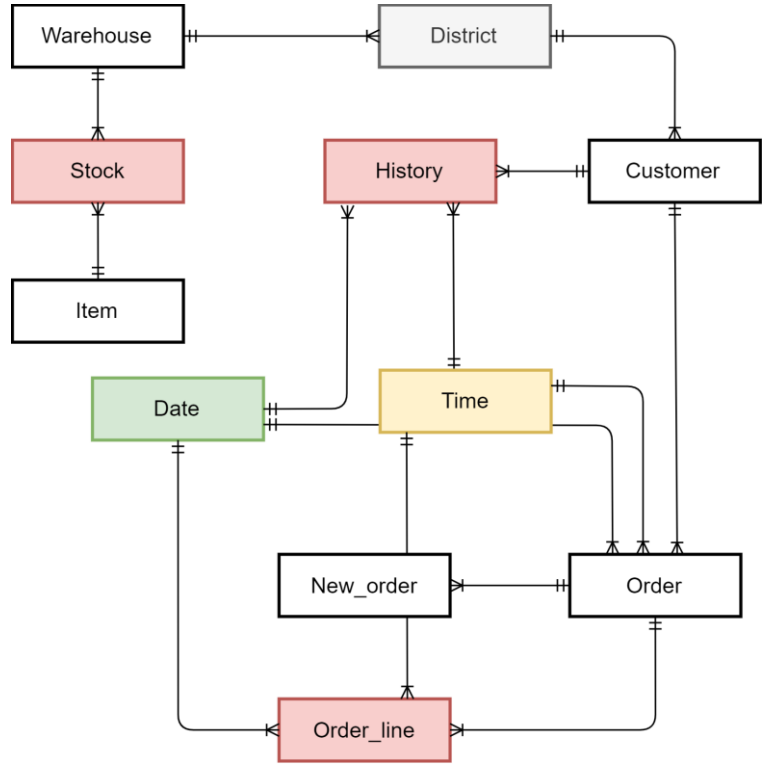



Figura 42 – TPC-C após aplicação da R2.

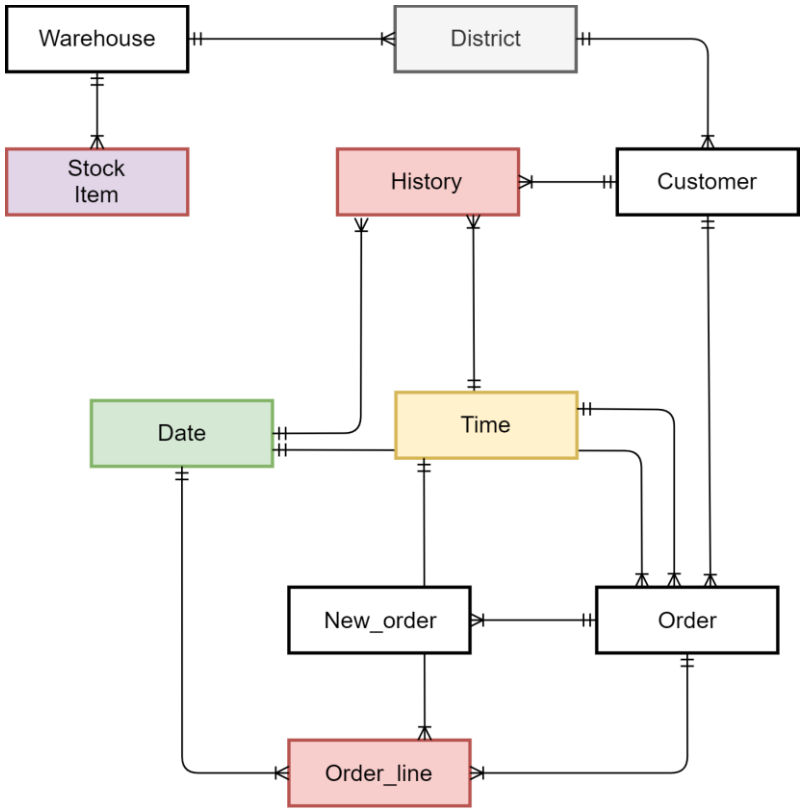



Figura 43 – TPC-C após aplicação da R6.