UNIVERSIDADE DE CAXIAS DO SUL

ALEXANDRE TOMÁS HÜBNER

BIG DATA SOB A ÓTICA DE MODELAGEM DE DADOS MULTIDIMENSIONAL PARA SOLUÇÕES DE BUSINESS INTELLIGENCE

CAXIAS DO SUL
2020

ALEXANDRE TOMÁS HÜBNER

BIG DATA SOB A ÓTICA DE MODELAGEM DE DADOS MULTIDIMENSIONAL PARA SOLUÇÕES DE BUSINESS INTELLIGENCE

Trabalho de conclusão de curso apresentado como requisito parcial à obtenção do título de Bacharel em Sistemas da Informação na área do conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul

Orientadora Profa. Dra. Helena Graziottin Ribeiro

CAXIAS DO SUL
2020

ALEXANDRE TOMÁS HÜBNER

BIG DATA SOB A ÓTICA DE MODELAGEM DE DADOS MULTIDIMENSIONAL PARA SOLUÇÕES DE BUSINESS INTELLIGENCE

Trabalho de conclusão de curso apresentado como requisito parcial à obtenção do título de Bacharel em Sistemas da Informação na área do conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul

Aprovado em 02/07/2020

Banca Examinadora

_____
Prof. Dra. Helena Graziottin Ribeiro
Universidade de Caxias do Sul – UCS

_____
Prof. Dr. Daniel Luis Notari
Universidade de Caxias do Sul – UCS

_____
Prof. Ms. Iraci Cristina da Silveira de Carli
Universidade de Caxias do Sul – UCS

Dedico esta produção a todos aqueles que, como eu, acreditam na força do conhecimento. E, sobretudo, na necessidade de que este gere resultados práticos para o dia-a-dia daqueles que fazem a economia girar: as empresas e seus profissionais.

AGRADECIMENTOS

Meu maior agradecimento a Deus pelo dom da vida e a oportunidade de viver esta experiência. À minha esposa Josiane pela incansável parceria, confiança e amor com que me apoia há 18 anos! À minha filha Ana Helena por tolerar minha ausência e me convidar diariamente à resiliência. Aos meus pais Irmo e Mônica pelas infinitas lições, incentivo e suporte.

Menção especial ao amigo e mentor Ildo de Barros, pelas oportunidades e desafios compartilhados, sem os quais jamais haveria me desenvolvido. Ao também amigo e prof. Dr. Gelson Rech, cujo incentivo foi fundamental para realização desta etapa. E por fim, à inesgotável professora Helena Ribeiro, pela paciência e tolerância com que me assistiu no desenvolvimento desta missão.

“[...] se você mostrar às pessoas os problemas e mostrar a elas as soluções, elas serão movidas a agir”
Bill Gates

RESUMO

A velocidade da propagação das tecnologias da era global e a mutação constante das características dos consumidores, proporcionam às organizações significativos desafios do ponto de vista da estratégia de sobrevivência. Por conseguinte, se torna absolutamente relevante a gestão de suas informações – já não sendo isto objeto de mera escolha. Conceitos tecnologicamente já consolidados como *Business Intelligence* (BI) devem ser parte da rotina das empresas. Considerando, contudo, os expressivos volumes de dados, oriundos de fontes como redes sociais, internet das coisas e indústria 4.0, já não são simplesmente experimentais as abordagens como a do Big Data. Esta obra tem por premissa a fundamentação destes elementos e suas dependências, bem como o desenvolvimento de um novo modelo de processos que combina abordagens oriundas do ELT (*Big Data*) e do ETL (*Business Intelligence*): o ELTL (Extrair, Carregar, Transformar e Carregar). Tal modelo visa obter a adoção otimizada e combinada daquilo que, de melhor, cada um destes contextos pode apresentar. Foram realizados testes de velocidade e qualidade, utilizando as famílias de ferramentas SQL e Hadoop. Tudo com o objetivo de validar a aplicabilidade desse novo modelo e evidenciar que BI e BDT são conceitos que apesar de distintos e com objetivos diferentes, podem ser utilizados em conjunto, de forma não excludente, para melhor análise e compreensão dos dados.

Palavras-chave: Business Intelligence, Big Data, Data Warehouse, Data Lake, Tomada de Decisão

ABSTRACT

The speed of the spread of global age technologies and the ever-changing consumer characteristics present organizations with significant challenges from a survival strategy perspective. As a result, the management of your information becomes absolutely relevant - this is no longer a matter of mere choice. Technologically consolidated concepts such as Business Intelligence (BI) should be part of the routine of companies. Considering, however, the significant volumes of data from sources such as social networks, the Internet of things, and industry 4.0, approaches such as Big Data are no longer simply experimental. This work is premised on the foundation of these elements and their dependencies, as well as the development of a new process model that combines approaches from ELT (Big Data) and Business Intelligence (ETL): the Extract, Load, Transform and ELTL. To charge). Such a model aims to achieve the optimal and combined adoption of what, best, each of these contexts can present. Speed and quality tests was performed, using SQL and Hadoop family tools. All with the objective of validating the applicability of this new model and showing that BI and BDT are concepts that, although distinct and with different objectives, can be used together, in a non-exclusive way, for better analysis and understanding of the data.

Keywords: Business Intelligence, Big Data, Data Warehouse, Data Lake, Decision Making.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

3NF	*Terceira Norma Formal*
BD	*Banco de Dados Relacional*
BDT	*Big Data*
BI	*Business Intelligence*
CRM	*Client Relationship Management*
DL	*Data Lake*
DW	*Data Warehouse*
ELT	*Extract, Load and Transform*
ELTL	*Extract and Load, Transform and Load*
ER	*Entidade-Relacionamento*
ERP	*Enterprise Resource Planning*
ETL	*Extract, Transform and Load*
GUI	*Graphical User Interface*
OLAP	*Online Analytical Processing*
SADs	*Sistemas de Apoio à Decisão*
SK’s	*Surrogate Key(s)*

SUMÁRIO

1 INTRODUÇÃO

A saga milenar da humanidade por evolução tem sintomas observados nas mais diversas esferas do relacionamento e interação. Nas organizações, este comportamento pode ser observado na constante busca pela otimização de processos, e redução de custos, para manutenção da competitividade. Em um cenário de disputas comerciais mais acirradas há que se desempenhar as tarefas com destreza e qualidade.

Em seu prefácio, Chiavenato (2003, p. 8) menciona que “estamos vivendo uma era de mudanças, incertezas e perplexidade. A era da informação está trazendo novos desafios para as organizações, e, sobretudo, para sua administração”. O avanço exponencial das tecnologias digitais, e a proeminente mudança de *MindSet*[1] que acompanha a geração de pessoas em evidente ativação econômica (chamados *Geração Z*[2]), deixa claro que embora redigida há mais de 10 anos a afirmativa de Chiavenato (2003) segue predominante, e em expressiva relevância. E este é o contexto em que se apresenta o processo de tomada de decisões.

O processo decisório tem sido também alvo de observações há décadas. Chiavenato (1999) ao analisar os modelos administrativos, cita a expectativa da então maior empresa de software empresarial do Brasil (Datasul), de expansão de 30% em seu faturamento, a partir da criação da solução *Business Intelligence System*, integrada ao software de gestão Datasul-EMS, cuja característica mencionada é o **apoio completo para tomada de decisão**. Decisões são necessárias em todas as esferas das organizações – da estratégica à operacional – e, a análise de informações, é elemento essencial para seu sucesso.

Lima, Camargo *et al.* (2017), observam que:

> Com a evolução tecnológica, especialmente nas últimas décadas do século passado, os sistemas de informações ganharam vulto. O grande avanço proporcionado pela tecnologia aos sistemas de informações pode ser resumido na capacidade de processar uma elevada quantidade de dados, gerar e disseminar informações de forma consistente e integrada para todos os gestores das diferentes áreas de uma empresa.

[1] MindSet é um termo popular utilizado para descrever o “modelo mental” ou “mentalidade”.
[2] Geração Z: “grupo de aproximadamente 80 milhões de pessoas (Estados Unidos), de pessoas nascidas após 1998, com características de diversidade peculiares.” GRUBB (2018).

A necessidade de otimização se dá, não exclusivamente ao trabalho da área e dos profissionais de TI. Fundamentalmente, os usuários de negócio devem ter sua rotina melhorada pelas ferramentas e abordagens de tecnologia. E deste ponto de vista, este trabalho de conclusão de curso, contribuirá de forma significativa. Afinal, a premissa do *Self Service*[3] é mais do que latente.

A elevada capacidade de geração de informações, pelos sistemas mencionados, apresenta significativos desafios às mais diversas áreas de negócio. E nisto podem ser constituídas as oportunidades de desenvolvimento relevantes para o contexto de dados tratado no presente objeto. Ou seja, profissionais das áreas de negócio e de tecnologia têm a oportunidade da troca de conhecimentos e atuação colaborativa, visando abordagens que ofereçam altos índices de retorno às organizações.

1.1 DO PROBLEMA

O problema aqui abordado, diz respeito à falta de discernimento com que os termos *Business Intelligence* (BI) e *Big Data* (BDT) são empregados pelos profissionais e empresas de tecnologia, trazendo assim dúvidas e ambiguidade para o mercado consumidor. Esta falta de clareza, ocasiona expectativa – incorreta – de que as duas soluções sejam alternativas, uma à outra.

Ao mesmo tempo em que elucidam questões globais, parte das literaturas contribuem para a negativa macropercepção citada. Marquesone (2016) explanando acerca de BDT, cita que “os dados eram mantidos em silos em um *Data Warehouse*”. A conjugação do verbo no tempo passado, referindo-se a um dos principais elementos do processo de BI – o *Data Warehouse* (DW) – é um exemplo típico da percepção sistematicamente distorcida – de sucessão cronológica evolutiva. Ou seja, trata-se da tendência à exclusão das prerrogativas do BI, quando da análise sob a ótica do BDT.

Os artefatos tecnológicos disponíveis têm, também, contribuição nesta problemática, uma vez que adotam nomenclaturas similares. Haja vista siglas como “ETL x ELT”, “*Data Warehouse* x *Data Lake*”.

Considerando que ambos os conceitos têm sido relacionados ao papel dos sistemas de informação no apoio ao processo de decisão, faz-se fundamental trazer

[3] Self Service é um termo em inglês, cuja tradução livre e literal significa autosserviço. Ele representa a capacidade de um indivíduo de ser autônomo com relação a determinado tema.

luz sobre eles, esclarecendo características, objetivos, requisitos, benefícios e sinergias, congruências e divergências. Por fim, há que se destacar que esta monografia “é sobre o conhecimento gerado, e não sobre o sistema em si”, utilizando aqui do raciocínio apresentado por Wazlawick (2008, p. 108). Assim sendo, o destaque será dado ao modelo e conceitos, e não às ferramentas específicas e especialistas empregadas.

1.2 OBJETIVO GERAL

Esta monografia – objeto de trabalho de conclusão de curso – tem por finalidade evidenciar que BI e BDT são conceitos que se complementam e, portanto, que a sua utilização pode ser integrada. Partindo dessa constatação, o objetivo é propor um modelo de processo híbrido entre ELT e ETL - duas técnicas bastante difundidas, porém utilizadas, até então, de forma isolada na Tecnologia da Informação. A proposta de resolução do problema será um modelo ELTL, baseado na junção de *paradigmas*[4] dos dois contextos (BI e BDT), a fim de resultar em uma melhor performance de execução de processamento dos dados.

“A combinação de etapas de dois conceitos distintos (ELT x ETL) pode gerar ganhos de performance no processamento e na análise de dados, dentro da uma estratégia de inteligência de negócios?”). Com o objetivo de responder a esta questão, a validação do novo modelo proposto ocorrerá através da medição da qualidade dos dados e a velocidade do processamento, em um ambiente controlado, a partir da submissão de uma base de dados contendo minimamente 15 milhões de registros e com a utilização das ferramentas de mercado, a serem definidas conforme cronograma proposto.

1.3 OBJETIVOS ESPECÍFICOS

Para o pleno atingimento do objetivo geral, são estabelecidos como objetivos específicos: identificar o papel de cada elemento no contexto do novo modelo de processos (ELTL) proposto; validar sua aplicabilidade e demonstrar a melhor

[4] Paradigma, conforme Ferreira (2008, p. 370) significa Modelo, Padrão

eficiência destes paradigmas tecnológicos quando utilizados de forma combinada – elevando assim a condição de geração de inteligência de negócio.

1.4 ESTRUTURA DO TRABALHO

O trabalho está organizado em seis capítulos. No primeiro, consta a introdução e abertura ao desenvolvimento da monografia. O segundo capítulo tem a finalidade de apresentar os conceitos relacionados à BI e DW. O terceiro capítulo, relaciona os conceitos que dizem respeito à BDT e DL.

Estabelecida a fundamentação, a partir do quarto capítulo, está estabelecida a apresentação de tendências e correlações entre as duas abordagens. É onde efetivamente está discriminado e detalhado o processo (modelo) proposto neste TCC. O quinto capítulo, descreve as atividades relacionadas à execução e desenvolvimento do que fora anteriormente proposto. Por fim, no sexto capítulo nada mais consta além da conclusão sobre o que aqui está sendo apresentado.

1.5 MÉTODO DE TRABALHO

Sendo o propósito desta seção “descrever os passos necessários para a demonstração de atingimento do objetivo” (conforme entendimento de Wazlawick (2008, p. 40)), o método adotado para este desenvolvimento é a “pesquisa exploratória”. Também caracterizado na literatura como “pesquisa experimental”, este método tem por característica fundamental a condução e o desenvolvimento de testes, que evidenciem os comportamentos propostos.

A sistemática inicia-se com a revisão bibliográfica, com foco na definição de termos e abordagem notórias. Após isto, é proposto novo processo/modelo de fluxo de carga e transformação de dados. Por fim, a proposição é posta à prova através da experimentação.

Considerando que todas as áreas de estudo devam ter por finalidade não o estudo em si, mas seu resultado prático – a resolução de algum problema – é de grande valor observar que a capacidade primária humana de execução, é o que respalda os métodos: seja o científico ou outro qualquer. Neste aspecto, os registros literários têm papel fundamental para a transferência do conhecimento, pois são o mais eficiente meio para registro e crítica dos feitos.

Esta revisão, está organizada em partes fundamentais, que trazem clareza aos temas abordados nesta monografia. Os capítulos 2 e 3, apresentam os fundamentos de BI e BDT, respectivamente.

O texto tem como finalidade demonstrar que as técnicas empregadas no BDT – como o *Extract, Load and Transform* (ELT), de forma isolada, não resultam em inteligência aos negócios, sendo necessária sua utilização junto a técnicas de BI como o *Extract, Transform and Load* (ETL). Pesem aqui o método de organização e tratamento dos dados. Para tal, será necessária a identificação do papel de cada elemento destas técnicas (ETL e ELT), no contexto da aplicação proposta. Assim, é discriminado de forma detalhada cada um destes paradigmas.

2 BUSINESS INTELLIGENCE

Este capítulo tem por finalidade descrever e apresentar conceitos e fundamentos do BI. Da sua finalidade aos atributos de processo, baseado no que há de mais clássico, acadêmico e, portanto, consolidado.

BI é um termo relacionado a um amplo processo, que abrange as mais diversas áreas de conhecimento e departamentos de uma organização. Contudo, para que seja viável e factível a referida análise de informações, é fundamental a utilização dos sistemas (de informação). Cabe citar, mas não limitando-se aqui, desde os sistemas de apoio aos processos operacionais e administrativos, até os sistemas de gestão de campanhas de marketing digital/online. Partindo-se da premissa de que dados são gerados por sistemas de informação, em termos práticos, não há possibilidade da realização de BI sem que se adote de forma organizada e estruturada a utilização destas ferramentas. Ou seja, teoricamente, o processo de BI pode ser realizado com base em modelos analógicos. Na realidade, isto é impraticável.

2.1 O QUE É BI

Para descrever o que é BI, necessariamente se faz necessário delimitar o escopo de abordagem, tendo em vista a tradução do termo, e seus sinônimos possíveis. Em tradução literal, o termo *Business Intelligence* nos leva em direção à “Inteligência do Negócio”, ou “Inteligência Empresarial”. Estes, por sua vez, proporcionam um amplo espectro, na medida em que envolvem não somente o processo decisório, como fundamentalmente as relações humanas. Em sua dissertação Parreiras e Matheus (2004), p. 4, chegam ao seguinte entendimento: “[...] a inteligência empresarial é mais ampla, e não é focada na tecnologia, mas no ser humano e nas organizações”.

De acordo com o instituto Gartner, em livre tradução, “BI é um termo “guarda-chuva” (abrangente), no qual são inclusas a infraestrutura, as ferramentas e as melhores práticas que tornam possível o acesso à análise da informação para melhorar as decisões”. Em simples definição para leigos: BI é uma forma de melhor organizar as informações, para que estas auxiliem as pessoas nas suas atividades diárias.

A presente monografia adotará o conceito tecnológico de BI. Assim, a expressão e o campo de estudo ficam delimitados aos sistemas informacionais, cuja finalidade seja a geração e apresentação de conteúdos relevantes à tomada de decisões.

A Figura 1, apresenta o caminho na jornada do BI, na visão de Sultan (2017). Ele afirma ainda que “BI provê aos gestores do negócio, acesso aos dados e modelos que possibilitam a condução de análises que levam à tomada de decisão”. A ilustração apresentada pelo autor, é organizada em cinco etapas, sendo sequencialmente interdependentes:

a) Dados: refere-se à coleta e organização dos dados;
b) Informação: refere-se à geração de informações a partir dos dados coletados;
c) Conhecimento: refere-se à produção de conhecimento organizacional, a partir das informações geradas;
d) Decisão: diz respeito à sustentação do processo decisório, que tem por pilar o conhecimento produzido na etapa anterior;
e) Lucro: estágio de maturidade, consequência das melhores decisões, tomadas em virtude do conhecimento adquirido com as informações, produzidas pelos dados.

Figura 1 – Trajetória do indivíduo

Fonte: Sultan (2017)

2.2 PAPEL DO BI NA TOMADA DE DECISÃO

É notória a necessidade das organizações contemporâneas – disruptivas ou não – analisarem dados. Em tempos de economia tendendo ao liberalismo, pode-se admitir – sem ônus moral – que esta análise de dados, tem por finalidade sustentar melhores decisões à organização. Em termos práticos, isto significa proporcionar às empresas, melhores condições de competitividade e lucratividade.

Assim sendo, com a democratização das tecnologias e plena difusão da internet, tornou-se obrigatório a qualquer empresa – seja ela uma grande corporação ou mesmo um pequeno negócio, conhecer com profundidade sua própria realidade. Adicionalmente, já não se pode considerar um diferencial a análise de dados de comportamento de mercado. A pena para a desatenção a estes aspectos, é o potencial fracasso do negócio.

Para que seja relevante, esta análise tem por premissa, a transformação dos dados em informações, que tem por sua vez, o potencial da geração de conhecimento à organização. Este fluxo, faz parte do processo de BI.

Não obstante, o processo de BI vai muito além da simples apresentação de indicadores e informações gerenciais. Para Loh (2014), BI tem como propósito responder perguntas. Ou seja, elucidar as razões para determinados comportamentos. E, para isto, é necessário que a organização tenha em sua cultura o processo de análise de informações. Também, é necessário que sejam propostas correlações e comparações de fatos e ocorrências. Nesta seara, se faz fundamental o emprego de técnicas e abordagens de cunho estatístico, cujo aprofundamento não faz parte do presente escopo.

Na avaliação de Williams (2017, p. 103), “o primeiro passo na tomada de decisão é identificar e definir o problema”. Sua abordagem, entretanto, apresenta um método complexo, que atravessa sete etapas até a tomada de decisão racional. Independentemente da extensão do processo de decisão que cada negócio escolha por adotar, o que é comum a todos eles, é a dependência de informações. Estas informações, podem ser apresentadas por um sistema de BI. Para Williams (2017, p. 374), estes sistemas, nominados Sistema de Informação Executiva (SIE), tem por finalidade “fornecer informações precisas, completas, relevantes e no momento certo para os gestores”. Indiferentemente do nome que lhe seja atribuído, o sistema de BI, é fundamental para as decisões.

2.3 REQUISITOS DO BI

Evidentemente que os sistemas de informação tradicionais como os *Enterprise Resource Planing* (ERPs), e os *Client Relashionship Management* (CRM) são fundamentais na rotina das organizações. Através deles, os processos cíclicos, que necessitam submissão – em maior ou menor grau – às regras de negócio, são executados de forma sistemática e estruturada. Como exemplo prático, a geração de notas fiscais e controle de estoque. Estes sistemas, por comum, proporcionam ao usuário, limitado conjunto de informações através de relatórios ou painéis gráficos predefinidos. Isto, entretanto, não é o suficiente para designar tal abordagem como BI.

Para caracterizar um sistema como BI, alguns requisitos devem ser atendidos. Dentre os principais, destacam-se a modelagem de arquitetura para um DW como o principal; a capacidade de criar correlações entre áreas de negócio distintas; a condição de apresentação das informações, sob óticas distintas e, preferivelmente não esperadas; a possibilidade de comparar dados de fontes diversas.

2.4 DATA WAREHOUSE

Kimball e Ross (2002) definem o DW como sendo um sistema de suporte à tomada de decisões. Para Inmon (1997, p. 33) o “*Data Warehouse* é o alicerce do processamento dos SADs”. Partindo destas afirmativas, é estabelecida a conexão entre o DW e o BI. DW tem papel fundamental para que haja possibilidade de usufruir dos benefícios de um BI.

Kimball e Ross (2002) citam que dentre as nobres causas do DW, está a de tornar as informações de uma empresa compreensíveis, intuitivas e óbvias para o usuário da área de negócios. A vasta literatura existente sobre o tema, contudo, é contundente ao afirmar que este armazém de dados deve apresentar as informações de forma consistente e confiável. Característica típica de um DW é sua não volatilidade. Sperley (1999, p. 13) afirma que “enquanto um sistema operacional (transacional) realiza atualizações, exclusões e inclusões nos dados armazenados, o *Data Warehouse* realiza carga de grandes volumes de dados que não são alterados”.

Para viabilizar a utilização das informações confiáveis, em larga escala, com relativa simplicidade, se faz necessário um modelo de dados específico, que se

mantenha distante do paradigma de estrutura de dados dos ERPs. Este modelo é o DW, que em uma tradução livre pode ser classificado como o Armazém de Dados Corporativos. Kimball e Ross (2002) referem-se aos bancos de dados, que são meras cópias dos sistemas operacionais e transacionais, como “impostores que prejudicam o conceito de Data Warehouse”. Neste contexto, o prejuízo não é somente do conceito, mas fundamentalmente do resultado.

Por que o DW contribui tanto no processo de tomada de decisão? Porque leva informação com relativa simplicidade ao usuário de negócio. É isto que o modelo dimensional proporciona, sem onerar os sistemas transacionais – requisito que as consultas diretas às bases de dados dos ERPs não conseguem atender. Adicionalmente, é possível descomplicar o viés técnico do modelo de ER, através do modelo dimensional – descrito mais à frente.

No que diz respeito ao seu desenvolvimento, o DW não é necessariamente simples. Mas, mais do que complexo, ele é trabalhoso – o que eventualmente torna o processo financeiramente oneroso. Sperley (1999) entende que para justificar a implantação de um DW, deve se utilizar como indicador, o tempo de trabalho que será poupado dos profissionais da organização. De acordo com ele, a economia resultante deve exceder o custo com o projeto, para que este possa avançar.

Como grande vantagem a seu favor, um DW devidamente estruturado, entrega à organização e às áreas de negócio, a possibilidade de análises e cruzamentos, de forma razoavelmente interativa e simplificada. Uma vez que a complexidade técnica do modelo formal – como a Terceira Norma Formal (3NF[5]) é desconstruída, não há necessidade de especialistas em tecnologia para as construções das visões. Ou seja, pessoas das áreas de negócio, com pouco treinamento, produzem suas visões de forma autônoma.

2.4.1 Fontes de dados de um DW

Considerando que o propósito do DW é a unificação de dados de diversas fontes, cabe destacar algumas das possibilidades. Tradicionalmente, as fontes de dados para um DW são os sistemas de transação como ERP, CRM, MRP, RH. Considerando ainda a alta probabilidade de que estes sistemas tenham por base

[5] A Terceira Norma Formal é uma norma global definida para a padronização de modelos de dados de sistemas transacionais como ERPs

tecnologias distintas e Sistemas Gerenciadores de Banco de Dados (SGBDs), distintos, é importante garantir que as ferramentas utilizadas, deem conta de sua junção.

Kimball e Ross (2002, p. 9) referem-se aos sistemas operacionais e transacionais de origem, afirmando que estas aplicações são naturalmente independentes. E complementa: “foi feito um investimento mínimo no compartilhamento de dados comuns como produto, cliente, geografia ou agenda, com outros sistemas operacionais da empresa” – referindo-se à falta de integração entre eles.

2.4.2 Datamarts

Sultan (2017), em tradução livre define que “o *Datamart* é um subconjunto do *Data Warehouse*, utilizado para atender as necessidades específicas de um grupo de pessoas ou time de uma organização”. Inmon (1997) refere-se ao *Datamart* como a representação da perspectiva departamental dos dados do DW. De outro modo, *Datamart* pode ser compreendido como uma fração virtual do DW, composta por um conjunto de entidades (tabelas), que fazem sentido entre si e respondem a um determinado tema.

2.4.3 Modelo Dimensional

O modelo dimensional tem por finalidade a simplificação do modelo de *entidade-relacionamento*[6], dos sistemas transacionais. De acordo com Kimball e Ross (2002, p. 15), “a modelagem dimensional dá conta do problema de esquemas extremamente complexos na área da apresentação”.

Conforme Kimball e Ross (2002), além da simplificação, um dos fundamentos do modelo dimensional é o reaproveitamento das dimensões. Ou seja, em um DW adequadamente projetado, as dimensões são utilizadas para diversos Datamarts.

Analogamente ao termo dimensão, está o entendimento da perspectiva. Ou seja, a premissa do modelo é de que se possa observar os fatos – as ocorrências – a partir de diversificadas perspectivas ou pontos de vista.

---

[6] Entidade-Relacionamento é o conceito utilizado para tratar da relação e organização entre tabelas de dados de um sistema

A simplificação do modelo, contudo, depende essencialmente da etapa de desnormalização. Esta etapa, tem por fundamento a descaracterização do modelo ER (normalizado – descrito previamente), tendo como referência o modelo em estrela (*Star Schema*), detalhado adiante.

2.4.3.1 Dimensões e fatos

Kimball e Ross (2002) atribuem a criação da terminologia das tabelas fatos e dimensões à um projeto de pesquisa realizado pela General Mills e Dartmouth University, na década de 1960. Formalmente nominadas entidades, tabelas fatos e tabelas dimensões são o exclusivo produto do processo ETL, adiante detalhado.

Conceitualmente falando, o DW é composto por nada além das tabelas fato e dimensões. As tabelas fato, são as entidades que contém as ocorrências de medição de um negócio. Por exemplo, a emissão de notas fiscais ou das ordens de compra. As dimensões, correspondem (em analogia simplificada), aos cadastros/pontos de vista a partir do qual se deseja avaliar as ocorrências. São exemplos de dimensões, a data/calendário, os clientes, os fornecedores, os produtos, os segmentos de atuação, as regiões geográficas, os representantes comerciais. Na Figura 4, a entidade central corresponde à tabela fato, e as entidades periféricas às dimensões.

As SK's (chaves substitutas) também são elemento de grande importância na implantação do modelo. Sua principal finalidade é servir de ligação entre as entidades (fatos e dimensões), proporcionando melhor desempenho do que poderia ser obtido com as chaves de negócio (legadas). Em geral, elas são criadas no processo de ETL, e consistem em campos do tipo inteiro, não nulos, com preenchimento auto incremental. Baseado nesta característica, a premissa de uma entidade fato, é que a mesma possua em seus campos somente os seguintes tipos de dados: datas, valores numéricos e chaves estrangeiras.

2.4.3.2 Cubo OLAP

A tecnologia que apresenta melhores resultados frente ao modelo apresentado, é aquela que se vale da utilização de bancos de dados multidimensionais (que vem do modelo dimensional) – que também podem ser chamados de Processamento Analítico Online – do inglês *Online Analytic Processing*

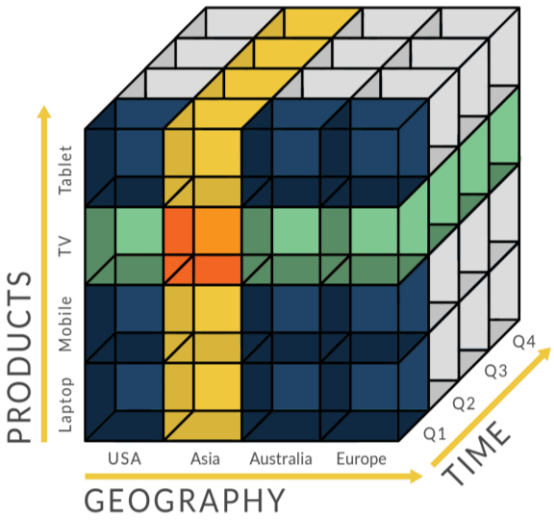
(OLAP), conforme Kimball e Ross (2002, p. 16). Diferentemente da estrutura de um banco relacional, a tecnologia OLAP tem como característica **manter em memória**, a partir de processamento previamente realizado, as formulações de cálculos (sumarizações, agregações, médias, medianas, divisões entre outros) pré-estabelecidas. Na Figura 2, podem ser observadas as dimensões (perspectivas) de um cubo OLAP. Embora no exemplo haja três dimensões (produtos, região e tempo), este não é um limite do modelo – que compreende **n** dimensões. Trata-se aqui de uma representação (simplificação) gráfica.

Em tradução livre, Garcia (2016, p. 2) refere-se ao Cubo OLAP:

> [...] estrutura multidimensional para acesso rápido da informação pré-processada. São estruturas de dados organizados e agregados, proporcionando-nos um acesso muito rápido e estruturado à informação que contém. Seus principais componentes são as dimensões e as medidas.

Este formato, entretanto, é significativamente oneroso, não somente em virtude das ferramentas – de alto custo, como também da necessidade de mão de obra especializada, e da menor difusão e adoção no mercado.

Figura 2 – Cubo OLAP



Fonte: https://olap.com/olap-definition/ (2019)

2.4.3.3 Esquema estrela

Mesmo tendo como pré-requisito o DW, um sistema de BI não depende essencialmente da adoção de ferramentas de OLAP. As características da simplicidade e fluidez do modelo dimensional, podem ser obtidas mesmo com um

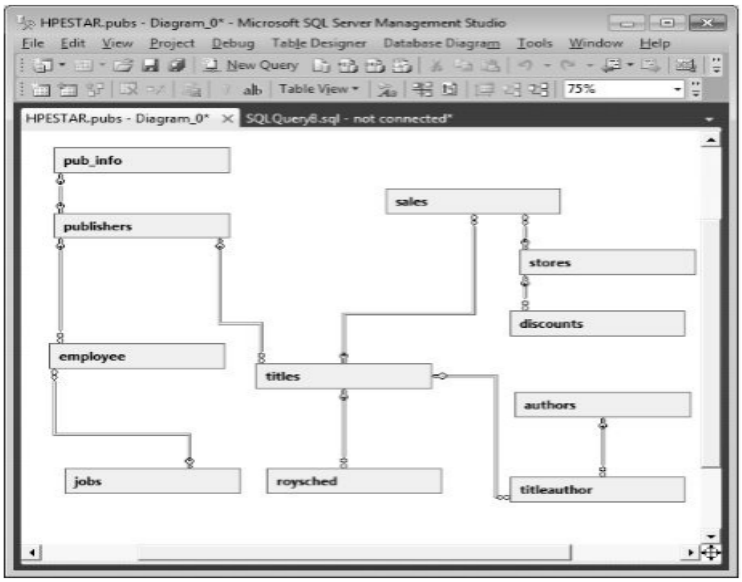
banco de dados relacional. A modelagem fiel à lógica do *Star Schema*[7] é, portanto, uma alternativa factível aos complexos bancos de dados OLAP.

Neste caso, também há que se observar a desnormalização do produto relacional, que geralmente acompanha os ERPs. Haja vista que o modelo normalizado, tem por finalidade a otimização do armazenamento dos dados, e assim, é comum observar relacionamentos de múltiplos e heterogêneos níveis, como representado na Figura 3.

A desnormalização do modelo consiste em abandonar normas como a 3NF, e proporcionar nível único de relacionamento entre as entidades fato e as dimensões. Em outras palavras, o esquema estrela, implica a não utilização de níveis de hierarquia e relacionamento dependente entre as dimensões. A Figura 4 apresenta o conceito de forma visual. Sperley (1999, p. 140) justifica:

> "Uma única entidade fato geralmente possui muitas dimensões de negócio associadas a si. Isto implica um modelo de dados onde a tabela fato é o centro para um amplo número de tabelas dimensão subordinadas"

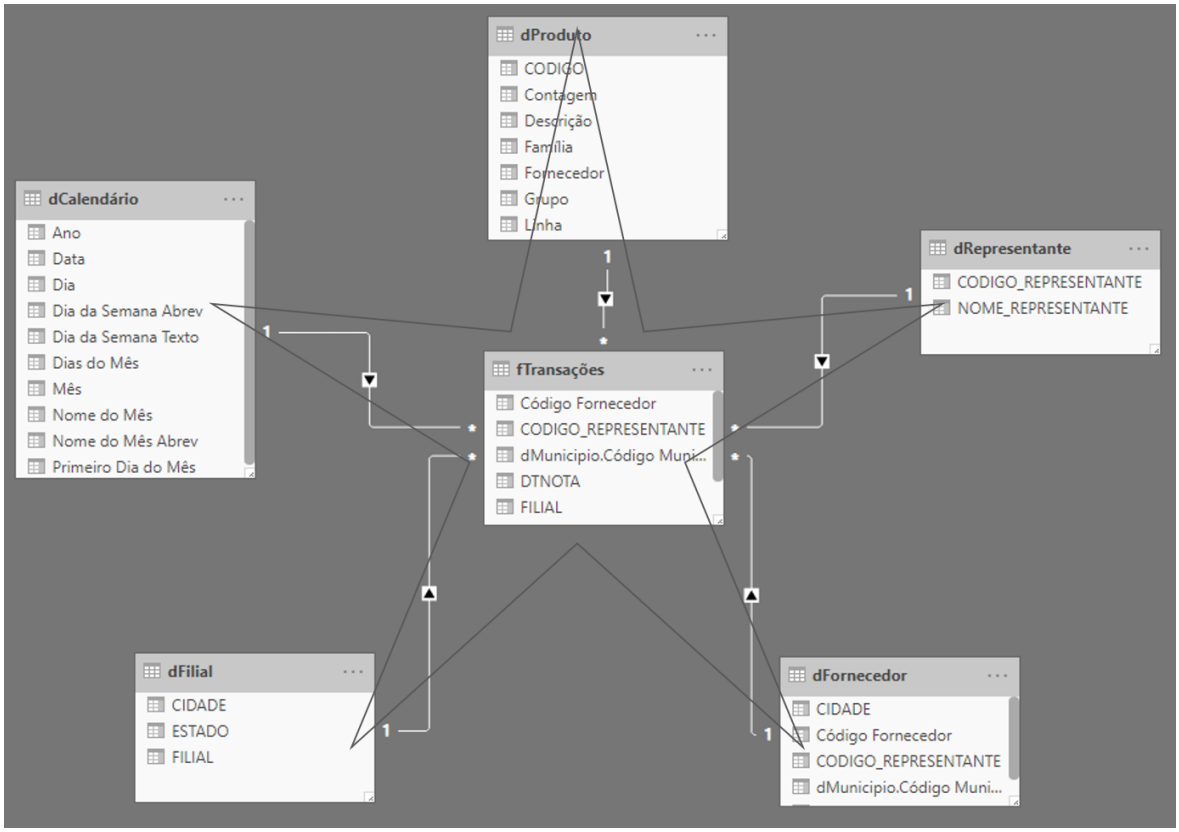
Figura 3 – Relacionamento múltiplos níveis



Fonte: Toth (2013)

Figura 4 – Representação *Star Schema*

[7] Star Schema: "Modelo Estrela", em tradução livre.



Fonte: HÜBNER (2019)

2.4.4 Granularidade

A granularidade dos dados é um elemento extremamente relevante no contexto do DW. Ela diz respeito ao nível de detalhe com que as ocorrências são armazenadas nas entidades fato. Grão é como é chamado o último nível de informação possível de se analisar – o nível mais detalhado – menos agrupado. Quanto mais granular, mais detalhado será o dado armazenado. Teremos grãos menores, mas, mais grãos. A isto é dado o nome Alta Granularidade. O oposto é verdadeiro. Ou seja, ao ser menos granular, o dado é menos detalhado. Logo, possuiremos menos grãos. Menos grãos significam menos dados a armazenar. A isto atribui-se o nome Baixa Granularidade.

Muito embora, do ponto de vista do armazenamento, seja vantajoso ter fatos com menor granularidade, Kimball e Ross (2002, p. 15) afirmam que “é totalmente inaceitável armazenar apenas dados de resumo em modelos dimensionais enquanto os *dados atômicos*[8] ficam confinados em modelos normalizados”. Com isto, os autores

[8] Dados Atômicos, no contexto dos autores, significam os dados com Alto Nível de Granularidade – e elevado poder de detalhamento.

deixam claro que cada problema deve ser cuidadosamente avaliado, para que as perguntas não fiquem sem respostas por decisões meramente tecnológicas. De acordo com Sperley (1999, p. 144) “a escolha do nível de granularidade correto, é uma decisão complexa, que envolve o negócio, hardware e software”. O Quadro 1 apresenta comparação das principais características do atributo em questão.

Quadro 1 – Granularidade

GRANULARIDADE	NÍVEL DE DETALHE	AGRUPAMENTO E TAMANHO DOS GRÃOS	QUANTIDADE DE GRÃOS	TEMPO DE PROCESSAMENTO	ÁREA PARA ARMAZENAMENTO
ALTA	ALTO - MUITO DETALHADO	BAIXO - DADOS POUCO AGRUPADOS	ALTA - MUITOS DADOS	ALTO - MAIOR VOLUME DE DADOS	ALTA - MAIOR VOLUME DE DADOS
BAIXA	BAIXO - POUCO DETALHADO	ALTO - DADOS MUITO AGRUPADOS	BAIXA - POUCOS DADOS	BAIXO - MENOS VOLUME DE DADOS	BAIXA - MENOR VOLUME DE DADOS

Fonte: HÜBNER (2019).

2.4.5 Verbosidade

Dentre as transformações recomendadas no processo de ETL, está a padronização dos dados, com a utilização de atributos com textos mais verbosos[9], como sugerem Kimball e Ross (2002). Ou seja, a utilização de códigos, que no ERP tem sua relevância (geralmente) garantida, no DW, perde espaço.

2.4.6 Paradigma ETL

A sigla ETL, oriunda da expressão em inglês “*Extract, Transform and Load*” significa, em tradução livre “Extração, Transformação e Carga”, nesta ordem. Pelos conceitos clássicos de BI, este é um processo ao qual são submetidos os dados, desde sua origem até a geração de informação. O resultado do processo de ETL é o DW.

O escopo de cada etapa do ETL pode variar de acordo com a ferramenta adotada e as experiências da equipe de profissionais responsáveis. Mas, de modo geral, como afirma Sultan (2017), citando Reinschmidt & Francoise (2000): “Extração é o processo de isolamento e encontro relevante dos dados de diferentes origens.” Costumeiramente, o resultado desta etapa é a chamada área de *Staging*. Nela, os

[9] Textos mais verbosos são aqueles nos quais há emprego de expressões de texto elaboradas com foco ao entendimento humano e não de máquinas.

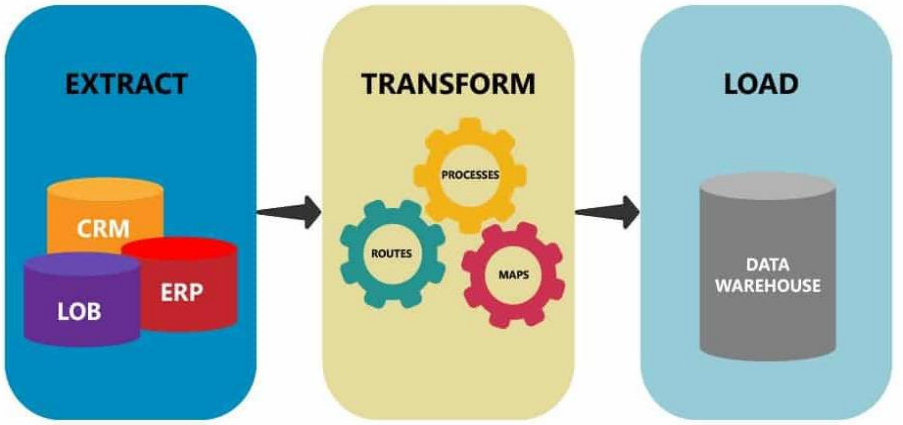
dados permanecem praticamente inalterados com relação à sua origem. Ou seja, ainda sem sofrer significativas alterações, mas já com *formato único*[10].

Na visão de Sultan (2017), a etapa de Transformação, tem a incumbência de, a partir dos dados temporariamente armazenados na *Staging Area*, implementar as regras de negócio – como as agregações, cálculos e correlações – corrigir erros, depurar dados e gerar informações consistentes para análise.

Como resultado do estágio de Transformação, obtém-se efetivamente o DW com suas *entidades*[11]. Esta é a etapa de Carga, que é a mais simples dentre as três, pois consiste em disponibilizar os dados no DW. Uma abordagem funcional do ETL pode ser observada na Figura 5.

A camada de apresentação visual dos dados, embora essencial para o consumo das informações, não compõe o processo de ETL. Esta área pode ser obtida com ferramentas visuais como *Microsoft Excel*, *QlickView*, *Tableau* ou *PowerBI.*

Figura 5 – Fluxo ETL



Fonte: Fatima (2019)

2.5 PROBLEMAS DA ABORDAGEM CLÁSSICA

Competência inerente à modelagem de um DW, é a possibilidade de realizar sumarizações. Ou seja, as tabelas de fatos ocorridos, não necessariamente apresentam os fatos ocorridos, mas sim, sumarizações e/ou médias de períodos ou grupos de dados (dimensões).

[10] O formato único mencionado no texto, refere-se ao formato padronizado e unificado dos dados na presente etapa. Isto significa, em um banco de dados comum.
[11] As entidades do DW são as tabelas fato e dimensões, detalhadas em tópico dedicado.

Ao mesmo tempo em que é identificado como competência (do ponto de vista da velocidade da análise a ser realizada), este comportamento é observado como fragilidade. Isto se dá, basicamente, sob o argumento da potencial falta de detalhamento dos dados. Pese neste contexto a redução expressiva de custo de armazenamento ao longo do séc. XXI. Ter os dados em seu maior detalhamento possível, pode ser, efetivamente útil do ponto de vista das descobertas.

2.6 BANCO DE DADOS ESTRUTURADO

O Banco de Dados Relacional Estruturado, tem por premissa básica conter estruturas com formato definido. Ou seja, cada entidade (tabela, visão ou função) tem atributos definidos em *formato*[12] e *variedade*[13]. Heuser (2019, p. 3), com o intuito de definir Banco de Dados, cita que este é “uma coleção de dados organizados, inter-relacionados e que atendem a uma aplicação ou família de aplicações”.

Este tipo de organização de dados, frequentemente adotada por sistemas transacionais (os ERPs), tem por característica estrutural, manter conjuntos de metadados. Isto é, as características de designação do conteúdo armazenado em si. Identificado por Heuser (2019) como Modelo de Dados, é o recurso que implementa a possibilidade e a garantia de que os dados armazenados acatem regras claras. Ainda, possibilita o armazenamento de informações de relacionamento entre as entidades, bem como *chaves de identificação exclusiva*[14] e *índices*[15].

[12] Os formatos de dados são, geralmente: textos, datas e valores.
[13] A variedade dos atributos, significa, neste contexto, os campos das tabelas. Ou seja, quais os dados estão ali armazenados.
[14] Conjunto de características que torna um dado único e exclusivo dentro de seu universo.
[15] Recursos comuns aos Bancos de Dados relacionais estruturados, que têm por finalidade elevar a performance das leituras de dados.

3 BIG DATA

De acordo com Marr (2015) as primeiras aplicações a adotarem o conceito do BDT como conhecemos hoje, datam de 1999. Segundo o autor, o marco de utilização do termo, contudo, se deu com o artigo publicado pelo site *Wired* onde Anderson (2008) aborda as escalas de armazenamento de dados como motivador do então novo paradigma. Inmon (1997) contudo, de forma indireta e inconsciente, já se referia ao BDT, ao citar em sua literatura “dados não estruturados” como objetos de estudo.

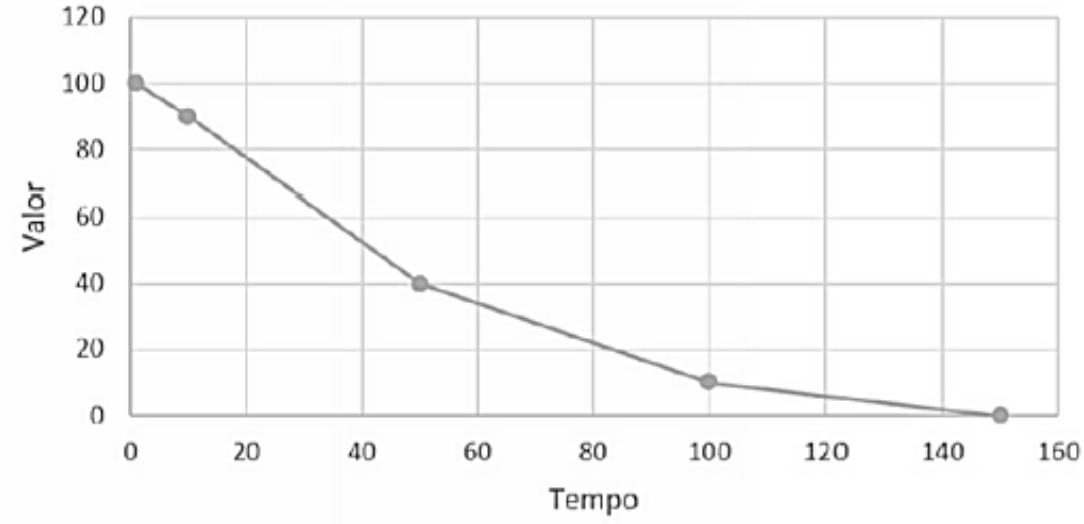
O advento das redes sociais, em meados dos anos 2000 – somado à popularização do acesso à internet, tornou iminente a necessidade de manipulação de dados com grandes expressões volumétricas. Consequentemente, a adoção de um novo paradigma.

No que tange ao processo de geração de informações, BDT – se adotado de modo isolado – não corresponde efetivamente aos resultados necessários. Amaral (2016) conduz a este entendimento na medida em que associa *Big Data* à Análise de Dados através da constante expressão “B&A” (*Big Data & Analytics*). Ao agregar à expressão o termo *Analytics***,** o autor deixa claro que, a adoção de processos acessórios ao Big Data, é de vital importância para sua geração de valor. Junto a esta perspectiva, se tornam relevantes questionamentos como o feito por Marquesone (2016): “Como extrair valor a partir dos dados?”.

Esta é uma área de estudo indispensável, tendo em vista que “analisar dados é uma questão de sobrevivência”, conforme Amaral (2016). Na mesma linha de defesa, Kimball e Ross (2002) citam que “um dos bens mais preciosos de qualquer empresa são suas informações”.

Amaral (2016) destaca ainda a relação entre o valor da informação e seu tempo de geração, conforme demonstrado na Figura 6. A análise do autor deixa explícita a importância da rápida análise dos dados, sob risco de perda de relevância.

Figura 6 – Relação entre valor da informação e seu tempo de produção



Fonte: Amaral (2016)

3.1 REQUISITOS

Atuar com BDT, exige (antes de mais nada) novo *MindSet*[1]. Marquesone (2016) confirma: “não há como atuar com Big Data estando resistente a mudanças”. Para a autora, um modelo de dados pode ser enquadrado na categoria *Big Data*, desde que atendidos os seguintes fundamentos: volume, variedade e velocidade – os três “Vs” do *Big Data* (BDT). A autora os define:

a) **Volume**: atributo mais significativo no conceito de BDT, faz referência à dimensão sem precedentes, no que diz respeito ao volume dos dados. Marquesone (2016), questiona sobre a aderência ao modelo de BDT: “Será que necessito de uma solução de Big Data somente se possuo *petabytes*[16] de dados? A resposta é não.”
b) **Variedade**: este requisito diz respeito à diversidade de formatos que, necessariamente, deverá não se limitar ao modelo de dados estruturados. Há que se considerar que aqui tenhamos dados não estruturados (definidos em tópico específico) e semiestruturados.
c) **Velocidade**: embora subjetivo, pois depende do requisito e da realidade de cada negócio, este aspecto também é detalhado por Marquesone

[16] Petabytes é a unidade de medida digital, correspondente a 1024 Terabytes.

> (2016) como requisito para um modelo de BDT. “O fator velocidade está se tornando tão importante, ao ponto que empresas que não conseguirem agilizar o tempo de análise dos dados terão dificuldades em se manterem competitivas no mercado”.

Amaral (2016) cita ainda a **Veracidade** e **Valor** como características inerentes ao BDT. Pela controvérsia e falta de unanimidade observada junto à literatura, esses aspectos não serão detalhados neste estudo.

Outra perspectiva importante do paradigma BDT, é o seu funcionamento com modelos multiprocessados. Sendo essa a característica técnica preponderante – distinta das anteriores, que são relacionadas à grandeza dos atributos dos dados. Isto pode ser observado na afirmação de Amaral (2016): “criou-se uma relação obrigatória entre *Big Data* e tecnologias de *MapReduce* como *Hadoop*. Estas tecnologias por sua vez, são baseadas nos conceitos de *cluster*[17].

Marquesone (2016) cita pesquisa realizada junto ao instituto Gartner, destacando a necessidade e a importância de evoluir sob a ótica dos processos: “Big Data faz referência não somente ao volume, mas também à variedade e à velocidade de dados, necessitando de estratégias inovadoras e rentáveis para extração de valor dos dados e aumento da percepção”.

3.2 DADOS NÃO ESTRUTURADOS

O conceito de dado não estruturado está diretamente ligado ao dos bancos de dados NoSQL (do inglês *Not Only Structured Query Language*). Fundamentais para o funcionamento da grande maioria dos sistemas Web modernos, esta forma de armazenamento é definida por Sadalage e Fowler (2019):

> “Os bancos de dados NoSQL atuam sem um esquema, permitindo que sejam adicionados, livremente, campos aos registros do banco de dados, sem ter de definir primeiro quaisquer mudanças na estrutura. Isso é especialmente útil ao lidar com dados não uniformes e campos personalizados, os quais faziam que os bancos de dados relacionais utilizassem nomes como customField6 (campoPersonalizado6) ou tabelas de campos personalizados, que são difíceis de processar e entender”.

[17] Cluster é o nome dado ao conjunto de mais de um computador e os softwares necessários à distribuição de atividades de processamento.

As formas de implementação são controversas. Sadalage e Fowler (2019) afirmam: “é provável que nunca haja uma definição coerente de ‘NoSQL’”. Isto deve-se à diversidade de alternativas de abordagem que podem ser caraterizadas como NoSQL. Marquesone (2016) destaca: “não existe um modelo de armazenamento único que seja adequado para todos os cenários de aplicações”.

Dados não estruturados têm por fundamento o componente da potencial variabilidade estrutural entre um e outro ciclo de carga e, assim sendo, dificilmente são submetidos a um modelo multidimensional (*Star Schema* ou OLAP), a não ser que sejam tratados. A elevada dinâmica nas atividades de leitura e gravação de dados, bem como a versatilidade na execução de alterações de modelagem durante o processo de desenvolvimento, favorecem a adoção do NoSQL, atraindo assim, muitos desenvolvedores.

O conceito, contudo, é antigo. Inmon (1997, p. 255) aborda os dados não estruturados, deixando claro, porém que seu destino deve ser o DW. Embora datado, do ponto de vista de geração da informação, seu raciocínio se mantém coerente. Permanecer utilizando o DW, é fundamental, afinal, de modo geral, a manutenção de um profissional cientista de dados para manipular diariamente dados não estruturados, através de códigos R ou Python, é extremamente dispendiosa e complexa.

3.3 DATA LAKE

Área que armazena e processa dados em formatos diferentes, com alta performance. O termo *Data Lake* (DL) foi criado por James Dixon, CTO do Pentaho[18]. Compreendendo a ideia dos dados “sempre disponíveis”, o termo se refere a um conceito (e não uma tecnologia, embora as soluções comerciais assim o deixem transparecer). Matos (2018) deixa claro que o verdadeiro valor que um DL pode oferecer, advém da capacidade e das habilidades de ciências de dados de quem o está utilizando. Alguns especialistas como Ribeiro (2018) afirmam que o dado armazenado no DL está pronto para o uso. O que está ligeiramente equivocado, uma vez que este dado, em seu formato bruto, em geral, não faz sentido para qualquer análise – consoante ao que fora fundamentado anteriormente.

[18] Pentaho é um dos maiores fabricante globais de soluções de processamento e análise de dados.

Data Lake poderia ser compreendido, de certo modo, como a área de *Staging* de um DW. A semelhança está no fato do armazenamento do dado em seu estágio anterior à transformação. As diferenças básicas estão nas seguintes características:

a) No DL são mantidos os dados em toda sua extensão. Ou seja, não somente as frações de incrementos (como ocorre na *Staging* do DW), mas sim todo o horizonte de período desejável para análise e disponível na fonte;
b) DL não tem como requisito a modelagem relacional;
c) DL não requer esquemas pré-definidos.

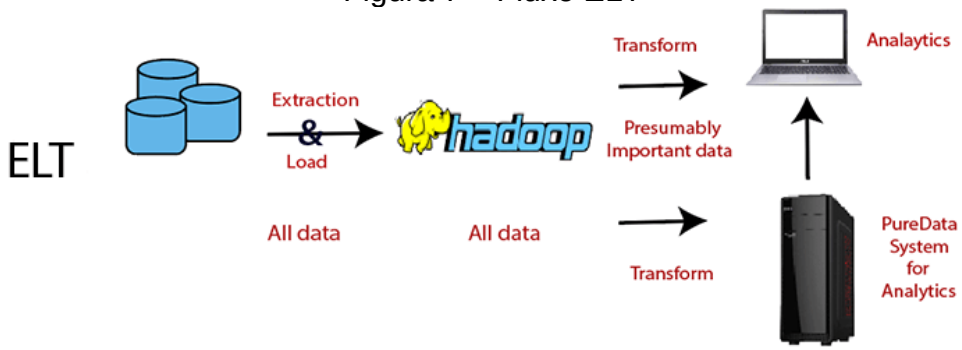
Pelas razões descritas, compreende-se a forte relação entre o fundamento de um Data Lake com relação aos dados não estruturados. O macroartefato DL, porém, compreende a possibilidade de processamento de dados com ferramentas como *Apache Spark*. Este recurso de processamento utiliza das qualidades da computação em nuvem, tal como a distribuição de tarefas em cluster para execução de suas funções. Além disso, DL tem por definição, a possibilidade de armazenar dados de estruturas clássicas, como bancos de dados relacionais, como cita Feinberg (2017).

3.4 PARADIGMA ELT

Em função das características essenciais ao BDT – anteriormente discriminadas, é latente a necessidade de evolução do aspecto dinâmico e de eficiência no processamento dos conjuntos de dados das organizações. Neste contexto as técnicas clássicas de extração e processamentos de dados, nominadas ETL (detalhadas anteriormente), já não entregam a performance ideal.

Assim, surge a abordagem de ELT. A semelhança do termo, contudo, não corresponde às importantes diferenças da proposta da Extração, Carga e Transformação (do inglês *Extract, Load & Transform*), para com o ETL. As literaturas são ricas ao descrever as etapas e ferramentas deste processo, embora não o sequenciem e identifiquem desta forma. Assim, a presente fundamentação empenhará utilização de analogias comparativas. Na Figura 7, é possível observar o conceito de fluxo de dados deste modelo, onde os dados são extraídos e carregados em uma área intermediária e, posteriormente, transformados.

Figura 7 – Fluxo ELT



Fonte: https://www.javatpoint.com/difference-between-etl-and-elt (2019)

3.4.1 Extração e carga

As etapas de Extração e Carga (EL) podem ser compreendidas como uma única. O mesmo artefato responsável pela extração, também se encarrega da carga dos dados no destino – o DL. Esta ação pode também ser identificada por Ingestão de Dados, ilustrada na Figura 8.

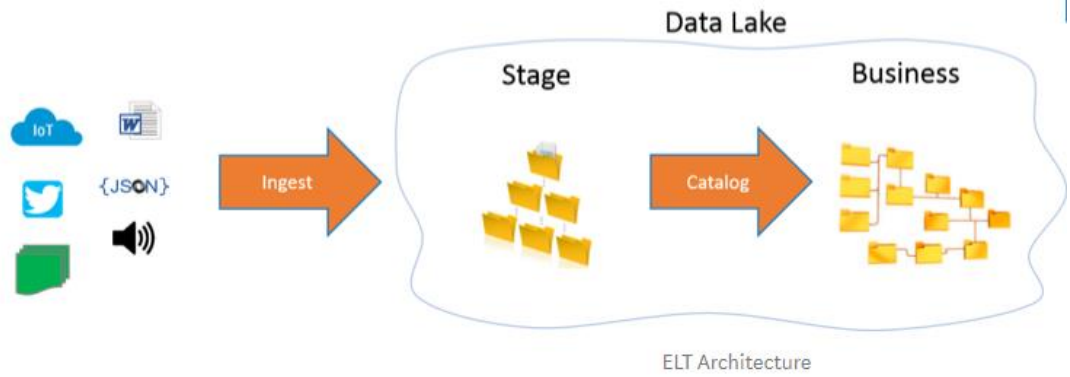
Não é possível abordar o processo de BDT e ELT, sem mencionar a tecnologia *Hadoop*. Taurion (2013) relata em sua obra o destaque do *Apache Hadoop* no cenário de BDT. Conforme o autor, esta tecnologia provê ainda um sistema de arquivos – o HDFS. Com base nesta tecnologia, ferramentas de mercado como *Microsoft Azure HDInsight*, fornecem a possibilidade efetiva de adoção da prática de "EL". Em termos da equivalência supra referida, firma-se aqui a correlação desta etapa.

3.4.2 Transformação

Converter os dados armazenados em informações, é a essência e o propósito desta etapa. Assim como no ETL, também para o ELT é este o momento em que há agregação de valor. De forma análoga, no contexto da tecnologia *Hadoop*, o paradigma *MapReduce*, é quem tem esta atribuição. Taurion (2013) destaca "manipulação de dados distribuídos em clusters de servidores usados de forma massivamente paralela", como funções desta etapa.

Para a abordagem de BDT é este o momento do processamento onde ocorre (embora de forma limitada) a aplicação das regras de negócio.

Figura 8 – Fluxo ingestão



Fonte: Marsh (2017)

3.5 PAPEL NA TOMADA DE DECISÃO

Dada a essência de comparação e análise crítica a que se propõe este objeto, por equidade, se faz necessário discorrer sobre o papel do BDT na tomada de decisões. Em tempos de alto nível de interação das organizações com clientes e *leads*[19] através de redes sociais, e marketing digital, entender o comportamento do consumidor já não é um diferencial mas, um requisito. Para manter a competitividade, é necessário observar tendências e agir com velocidade, acompanhando assim os movimentos de mercado. Proporcionar tal condição, é razão existencial do *Big Data*.

Sargut e McGrath[20] (2011 apud Loh), 2014), ao abordarem o processo de tomada de decisão e geração de insights,

> sugerem a gestores estabelecer um modelo que agregue três tipos de informação preditiva:
> • informações passadas: dados sobre o que já aconteceu, incluindo indicadores financeiros e de desempenho;
> • informações presentes: alternativas de caminhos, ações, estratégias, oportunidades ou decisões que podem ser tomados;
> • informações futuras: o que pode acontecer como consequência das alternativas, incluindo respostas do meio-ambiente ou mudanças internas"

É no aspecto das informações presentes e futuras que o BDT mostra seu grande valor, através da possibilidade de emprego de técnicas de *Machine Learning*

[19] Leads são os indivíduos atraídos pelas companhias ao processo de vendas, durante o processo de prospecção e marketing

[20] SARGUT, Gökçe; McGRATH, Rita Gunther. Learning to Live with Complexity. **Harvard Business Review**, special issue on Complexity, September 2011.

e Artificial Intelligence[21]. Afinal, para análise de informações passadas, dispõe-se do consolidado *Data Warehouse*.

3.6 O BIG DATA VAI SUBSTITUIR O BI?

Exercitar esta reflexão, exige minimamente ousadia. Predizer com algum índice de segurança os movimentos futuros do mercado de tecnologia, sobretudo em uma área de tão grande valor agregado, é algo a que modelos matemáticos dificilmente serão suficientes. Assim, seria necessário entrar no viés supositório, descaracterizando assim a essência científica a que se destina esta construção.

Sultan (2017) define BDT como um “transformador” do BI, assim como também o são as tecnologias cloud, melhores infraestruturas, entre outras. Outra característica mencionada pelo mesmo autor é a suposta independência dos usuários da área de TI para seus processos de BI. Seriam estes alguns recursos/características do “BI do futuro” (tradução livre):

a) Sistemas de Armazenamento Distribuídos (*Distributed Storage Systems*);
b) Tecnologias Em-Memória RAM (*In-memory Technologies*)
c) Análise de Diferentes Tipos de Dados (*Analysis of Different Types of Data*);
d) Sistemas de Auto-Serviço (*Self-Service Systems*);
e) Ciclos Controlados (*Control Loops*).

A maciça maioria das abordagens de BDT requerem conhecimentos expressivos em programação. Ou seja, esta não é uma atividade de alta produtividade que possa ser realizada pelo usuário da área de negócios.

Conforme Matos (2018) a exploração dos dados de um DL, cabe ao cientista de dados. Esta afirmação contribui de forma significativa com o entendimento da necessidade da evolução da abordagem de convenção do BDT. O usuário que consome o dado, não tem no DL e BDT uma possibilidade real e independente de consumo de dados. Para tal ainda prova seu valor o famigerado DW.

[21] Machine Learning (Aprendizado de Máquina) e Artificial Intelligence (Inteligência Artificial) são tecnologias que resultam em predição de comportamentos através da análise de dados e exercícios controlados.

A crítica a este modelo é que a simplificação, desnormalização e organização do dado não é mais apresentada como uma etapa do processo – coisa que no BI Clássico é fundamentalmente suprida pelo ETL.

Baú (2019, p. 63) cita que:

> as dimensões do fenômeno *Big Data* apenas geram resultados práticos quando aplicadas aos processos de extração de insights dos dados. Nesse contexto, um grande volume de dados não tem relevância se não puder ser tecnicamente absorvido e processado pela organização.

E continua:

> os dados de fontes variadas se tornam um problema sem um processo de aglutinação. E o oposto é verdadeiro. Ou seja, sem um grande volume de dados as análises ficam prejudicadas pela falta de triangulação.

Em sua relevante pesquisa sobre o impacto do BDT em uma cooperativa de crédito, Baú (2019, p. 64) traz depoimentos de seus entrevistados, onde fica destacada a importância do BI:

> No nosso BI, com a base dados dos últimos 5 anos que a gente tem, eu pedi para ele verificar para mim quais são os associados que, nesses últimos 5 anos mais deram retorno para a Cooperativa (Singular) [...]. Porque a gente fica míope quando olha um ano só, porque o associado, ele tem uma vida aqui dentro. Porque se olha o associado em um ano só:

Assim, muito embora o BDT traga contribuições relevantes ao contexto de manipulação de dados, especialmente de grandes volumes, as técnicas de armazenamento do BI (DW), continuarão sendo vitais para o processo de tomada de decisões.

Baú (2019, p. 75) cita ainda que:

> o valor dos dados também tem relação com o tempo, especialmente em termos qualitativos. Isso significa que quanto mais dados históricos são reunidos, maior é o valor percebido e sua utilidade para a tomada de decisão..."

Sabendo, pois, que, dados históricos são concernentes à BI (em processo e conceito), põe-se em perspectiva a continuidade da abordagem de BI, mesmo diante da acelerada evolução do BDT.

Grande parte dos entrevistados de Baú (2019) citam “BI” como a área ou processo que fornece as informações. Isto deixa claro que o BDT pode ser sim um excelente processo ou técnica intermediários. Mas o fim, é a inteligência de negócios (BI).

Assim, a divergência a ser destacada aqui é aquela que diz respeito às técnicas de processamento de cada uma das abordagens (BI com ETL, BDT com ELT). Big Data, em geral, demanda a utilização de conhecimentos de ciência de dados para a geração de informação, o que torna a organização refém deste profissional.

A etapa de Transformation do ELT, sugere que a Transformação ocorra “online”, diferentemente do que é proposto para concepção de DW, através do ETL. Este é um paradoxo tecnológico, cujas defesas subsidiam-se de impotentes argumentos – o que torna o embate irrelevante.

Fato é que não há análise em cima de dados brutos. Os dados brutos devem ser depurados, e alocados temporariamente em área específica designada (o que na abordagem clássica é realizado com a utilização de DW). Ainda, para que haja viabilidade para utilização de um *Self-Service*, este trabalho de organização dos dados é indispensável.

Assim, é correto afirmar que o BDT não substitui e nem substituirá o DW, e nem o BI. Eles são adjacentes e complementares.

4 PROPOSTA DE SOLUÇÃO

As soluções e o serviço de mercado disponíveis, indicam adoção concorrente e não concomitante das técnicas abordadas neste objeto. Embora não sejam sinônimos, BDT e BI, estão longe de serem absolutos antônimos. Como de antemão se observa, há dificuldade de entendimento entre os termos. E isto vai desde a abordagem comercial, passando pela formação específica dos profissionais, até os contextos de utilização. De artigos em blogs, à livros especializados.

O advento do BDT é incontestável. Por suas características fundamentais descritas na sessão 3, e pela iminente necessidade de evolução na dinâmica do processo de tomada de decisão. Ao mesmo tempo, esta abordagem não sana de forma ampla, autônoma e satisfatória a problemática da organização mínima dos dados. Este arranjo afeta diretamente a capacidade de interpretação das informações contidas nos conjuntos de dados, sendo, portanto, essencial para o processo decisório.

Loh (2014, p. 30) é esclarecedor ao dissertar a respeito do processo de BI:

> O processo de BI só se completa quando o conhecimento descoberto chega até as pessoas que precisam dele, no formato correto e no tempo exato. Se o processo demorar demais, se o resultado chegar num formato não adequado, o processo de decisão (razão da existência das informações) será comprometido.

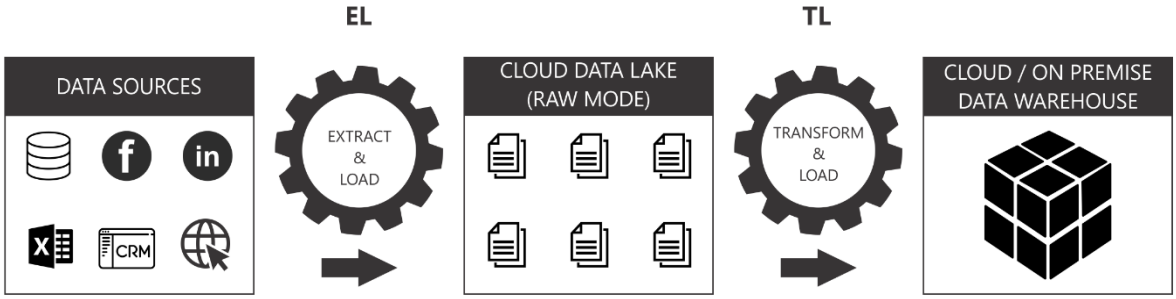
A afirmação de LOH deixa explícita a necessidade de agilidade, mas também de clareza e interpretabilidade na informação disponibilizada aos indivíduos decisores. Por esta razão, o modelo aqui proposto, se valerá de técnicas de BDT e BI. Ele converge ao estudo de Ranjan (2009), que apresenta discussão sobre os pontos fortes e fracos das abordagens ETL e ELT. O autor, no entanto, dedica elevado foco ao aspecto da performance computacional dos modelos, deixando claro que há diferença de desempenho. Ele deixa claro que tal distinção não se deve diretamente às abordagens (ETL x ELT) mas sim, à arquitetura de infraestrutura e localização das bases de dados. Ao fim do artigo, o pesquisador aponta para a necessidade de aprofundamento de um modelo híbrido.

Firma-se que para sua adoção devem ser atendidos alguns requisitos: precisar de análise de dados históricos (DW), e precisar de processamento veloz (BD). Ao ter como base (ponto de partida) o paradigma ETL, com eficiência já consolidada,

o autor identifica relevantes ganhos de performance com o modelo de ELT. Sugere, porém, como ideal (embora sem o devido aprofundamento), a combinação dos paradigmas, sendo assim designado o **ELTL**. O aprofundamento e detalhamento desta abordagem é, outrossim, objeto deste trabalho.

O grande acréscimo deste trabalho com relação ao estudo de Ranjan (2009) se dá na medida em que esta monografia tem por pilar macro a crítica ao método. Assim, com sua conclusão haverá um processo estabelecido e uma sequência lógica de etapas a serem seguidas – suas entradas, atribuições e saídas. A Figura 9, ilustra como ocorrerá o processo aqui apresentado. Seu detalhamento é apresentado a seguir.

Figura 9 – Fluxo de dados ELTL



Fonte: HÜBNER (2019)

4.1 MÉTODO DE DESENVOLVIMENTO

A presente proposta, tem por finalidade apresentar o modelo de fluxo de processamento de dados compatível com o conceito de Extração e Carga, Transformação e Carga – *Extract and Load, Transform and Load* (ELTL). A proposta é desenvolvida a partir da definição dos conceitos discriminados nos capítulos 2 e 3. Da definição das possíveis fontes de dados, ao posterior detalhamento e sequenciamento de ELTL.

Posterior à sintomática conceitual, será realizada a caráter experimental, a implementação funcional, seguida dos testes de desempenho e qualidade da solução. Por fim, a validação dos resultados, obtida a partir da medição de resultados.

4.2 DEFINIÇÃO DAS FONTES DE DADOS

Para que gere informação – e seja analisado, um conjunto de dados precisa estar estruturado, mesmo que em sua fonte ele não siga esta premissa. Em termos práticos as unidades de armazenamento, podem ser diversas, em localização e estrutura. Eis o *Big Data*.

Assim sendo, as fontes de dados comportadas nesta sistemática, são as fontes estruturadas (BD) e também as fontes não estruturadas e semiestruturadas – predominantes na utilização de BDT.

4.3 SEQUENCIAMENTO ELTL

O termo ELTL fora observado formalmente pela primeira vez no estudo de Ranjan (2009). Ele vem do inglês *Extract and Load, Transform and Load,* sendo resultante da exata sobreposição dos conceitos anteriormente discriminados ETL e ELT. Extração, Carga, Transformação e Carga: a tradução literal do termo, seria em resumidas palavras, a síntese desta proposta. Os nomes e termos técnicos, contudo, são o aspecto menos relevante – ficando como essência, portanto, a compreensão do conceito e do processo. O Quadro 2 apresenta de forma sintética características importantes para a compreensão do fluxo ELTL.

Quadro 2 – ELTL

ETAPA	DESCRIÇÃO	ORIGEM	DESTINO	AÇÃO	PARADIGMA ORIGINÁRIO
EL	EXTRACT AND LOAD	- DB RELACIONAL - ARQUIVOS SEMI-ESTRUTURADOS - ARQUIVOS NÃO ESTRUTURADOS (CLOUD E ON-PREMISSE)	DATA LAKE	CÓPIA/INGESTÃO	ELT
TL	TRANSFORM AND LOAD	DATA LAKE	DATA WAREHOUSE	TRANSFORMAÇÃO, PROCESSAMENTO, RELACIONAMENTO DOS DADOS; POPULAÇÃO BANCO DESTINO	ETL

Fonte: HÜBNER (2019).

4.3.1 Etapa de extração e carga

A extração é a etapa onde os dados serão obtidos de suas fontes, e armazenados – sem transformações. Tal como na abordagem de ETL, onde há a área de preparação (ou *Staging*), aqui teremos uma área com os dados em formato bruto.

Para a proponente abordagem, todavia, esta etapa se dará de acordo com as práticas de ELT, sendo chamado assim também de processo de ingestão. O dado será armazenado integralmente em um DL, sem que haja nesta etapa quaisquer cruzamentos ou sumarizações. Dentre as técnicas e ferramentas possíveis para a ingestão de dados, estão *Hadoop* e *CosmosDB*.

4.3.1.1 Dados estruturados

Os dados estruturados (vide seção 2.6) – frequentemente relegados pelas abordagens literárias de BDT são fundamentais para os processos de negócio das organizações. São inúmeros os casos de aplicação, como emissão de notas fiscais, controle de estoque e compras. É possível admitir, de modo geral, que as funções cumpridas pelos sistemas ERP, ERPII, CRM entre outros, são e continuarão sendo dependentes de Bancos de Dados Relacionais – de onde advém o conceito dos dados estruturados. Assim, sua participação na geração de informações para a tomada de decisões é incontestável.

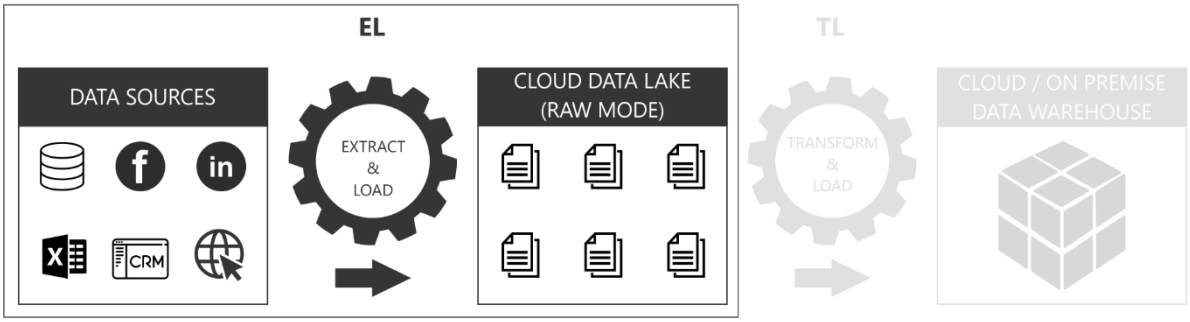
Tendo em vista a abordagem proposta, a ingestão dos dados estruturados, fará com que os mesmos tenham por destino o DL. A Figura 10 demonstra o fluxo de dados e a sistemática adotada.

4.3.1.2 Dados não estruturados

Dado o paradigma da ingestão dos dados, carregar aqueles de tipo não estruturado é a própria justificativa do conceito. Considerando que este tipo de conteúdo tem como origens comuns redes sociais e marketing online, faz-se cada vez mais necessária sua participação no processo de análise. Outras frequentes fontes de dados não estruturados são sensores IoT, dispositivos computacionais, textos e vídeos.

A linha de raciocínio diferencial desta monografia, se dá pela hipótese do cruzamento e comparação das informações oriundas destas fontes, com outros dados da organização, com origem nos dados estruturados (vide etapa de Transformação e Carga). A Figura 10, em corte, a seguir demonstra o fluxo lógico dos dados, de sua origem com destino ao DL.

Figura 10 – Fluxo de dados EL

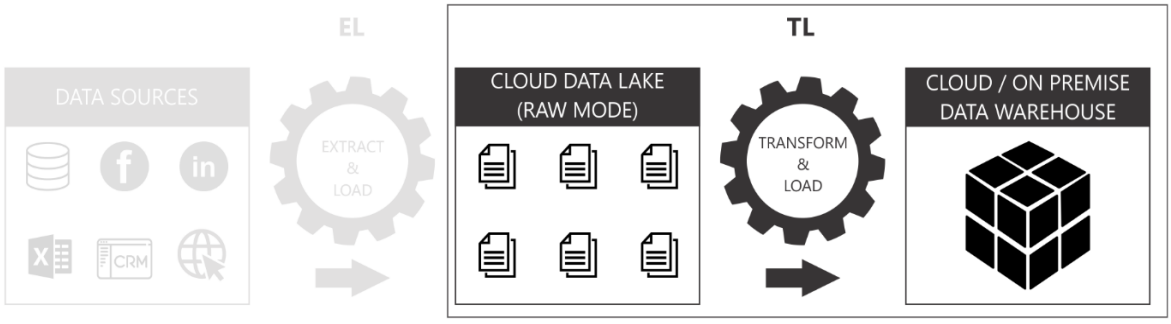


Fonte: HÜBNER (2019)

4.3.2 Etapa de transformação e carga

A Transformação e Carga, abstraída das duas últimas etapas do processo de ETL, é a ação chave para a riqueza de detalhes do modelo de dados gerado por este processamento. Responsável pela implementação do conceito multidimensional (cubo) de estruturas de dados, tem como finalidade a geração do DW que, por sua vez tem o objetivo de servir à organização como a “fonte da verdade”. Tendo por premissa o processo de extração, utiliza-se dos dados dispostos no DL. Complementa-se que mesmo os dados não estruturados, precisam ser estruturados em algum estágio do processo, para que possam assim ser manipulados e gerem valor. A Figura 11, apresenta de modo ilustrado e destacado, a etapa de Transformação e Carga dos dados no DW.

Figura 11 – Fluxo de dados TL



Fonte: HÜBNER (2019)

4.3.2.1 Transformação

A transformação dos dados, é a real responsável pela implementação do conceito multidimensional. É nesta etapa que os dados dos movimentos e cadastros

são relacionados, formando assim as dimensões e fatos. Sendo as fontes de dados oriundas do DL, resta oportuna utilização de ferramenta específica para este fim, cuja escolha pode variar de acordo com o contexto.

Como parte fundamental desta etapa está a adequação de modelagem (ou organização) dos dados não relacionais. Isto pode ser efetuado com a adoção de ferramentas como *Presto* e *Starbust*. Através destas, é possível consumir dados não estruturados com comandos *SQL*[22]. Por conta destas tecnologias, podemos vislumbrar com alto índice de exequibilidade o modelo corrente.

A grande distinção do proponente projeto, está na medida em que se compreende que os dados oriundos de fontes não estruturadas, possuem potencialmente valores de grandeza semelhantes aos conceituados para o DW. Assim sendo, definir os artefatos de dimensões e fatos, para tais dados, é a ação central indicada. Esta atividade é substancialmente sugerida para aqueles dados cuja modelagem (apesar de não estruturada), se repita em escala. Ou seja, não integrar dados cujo contexto seja desconhecido, é uma restrição desta implementação.

A interpretação dos dados brutos e não estruturados, deve ser realizada com a utilização da documentação fornecida pela sua fonte geradora. Em não havendo esta disponibilidade, esta tarefa deve ser efetuada por processo de *engenharia reversa*[23], semelhantemente ao que ocorre em análises de construção de projetos de BI. A partir deste entendimento, devem ser definidos os enquadramentos dos atributos e valores contidos no modelo, estabelecendo assim os referidos artefatos.

4.3.2.2 Etapa de carga

Como parte desta proposta, firma-se que o destino de toda a informação processada, independentemente de sua origem, se encontra no Data Warehouse. Como grande benefício desta técnica, está a disponibilização da informação à área de negócio da organização – que é o cliente de qualquer solução de BI – de forma relativamente simplificada, para que o mesmo possa realizar sua exploração utilizando do conceito do *Self Service*[3]. O objetivo é apresentar ao cliente final, maior dinâmica

[22] O consumo de dados não estruturados com linguagens como SQL, corrobora a necessidade da mínima estruturação de qualquer dado sobre o qual seja necessária a realização de análise.

[23] Esta técnica consiste, basicamente, na análise de amostras dos dados, acompanhada de execuções de processos de negócio e entrevistas com as áreas de negócio. Seu objetivo é identificar o comportamento dos dados no contexto.

de consumo de tais dados, se comparado ao modelo tradicional de BDT, onde a dependência de um profissional com conhecimentos em ciências de dados é muito grande.

Funcionalmente, só há a etapa de transformação. A carga, de forma efetiva, trata-se do destino dos dados transformados. Como resultado desta carga, será obtido o DW, dotado de relevante simplicidade – obtida através do processo de desnormalização, e aumento e padronização de verbosidade das estruturas de dados (contidos nas transformações).

4.3.2.3 Ferramentas

Para a operacionalização das etapas descritas, são necessárias ferramentas especialistas, cujas características podem ser extremamente distintas. Dentre as principais, podem ser citadas Microsoft SQL Integration Services, Pentaho Data Integration, Tableau, Informatica, Família de produtos Hadoop, Kafka, Google Big Query e Amazon Redshift.

4.4 SIMPLIFICAÇÕES

Há sutis alterações, através das quais os modelos propostos, podem ser simplificados, evitando assim a dispersão de empenho tecnológico. Ou seja, tanto BDT como DW oferecem a possibilidade de otimização de sua utilização, tornando a utilização híbrida, sujeita a características específicas.

4.4.1 Variação de granularidade e incrementalidade em Data Warehouse

Como alternativa ao problema das sumarizações – reclamado pelos cientistas de dados, e utilizado como argumentação favorável à utilização de DL – mas sem que para isto seja necessária a mudança de paradigma ou tecnologia, está a carga incremental de dados, em alto nível de *granularidade*[24]. Isto, porém, não é adequado para volumes extremos de dados, sob o risco de baixa performance de todas as consultas oriundas do DW.

[24] Alto nível de granularidade significa dados com mais detalhes – mais grãos e grãos menores – menos sumarizações

4.4.2 Organização de dimensões degeneradas globais

Análises de dados originados em fontes não estruturadas podem ocorrer sem a aplicação do modelo multidimensional – desde que sejam irrelevantes cruzamentos com atributos corporativos. Analogamente chamadas de “tabelões”, trazem grande diversidade de dados em estruturas simplificadas. Esta simplicidade, por sua vez, proporciona grande velocidade de implementação. Seu ônus, entretanto, é a limitação no que diz respeito à correlação com os demais dados da organização. Neste caso as dimensões a partir das quais um dado pode ser analisado, são em demasia limitadas.

4.5 IMPLEMENTAÇÃO E VALIDAÇÃO

Para implementação do presente modelo, serão utilizadas bases de dados demonstrativas, cuja definição, procedência e fornecimento serão estabelecidos no início da execução das atividades. Dentre as possíveis tecnologias empregadas para tal, estão Microsoft Azure Services, Microsoft SQL Server Database, Microsoft SQL Integration Services, Microsoft SQL Analisys Services, Pentaho Data Integration, Apache Hadoop, Amazon Redshift, Google BigQuery. Os critérios para a escolha das ferramentas serão: aderência com cada fundamento, mínima necessidade de diversificação, complexidade operacional e custo.

A validação da presente proposta será realizada através da comparação entre os resultados de seu emprego, frente a utilização modelos clássicos. Serão comparados três modelos:

a) ETL (DW/BI)
b) ELT (DL/BDT)
c) ELTL (DL/DW/BI)

Como parâmetros de comparação, serão utilizados três indicadores: tempo de processamento, integridade qualidade dos dados. O tempo de processamento será medido por aferição cronológica simples, tendo por base o início e o término da carga do modelo de dados escolhido. A aferição de integridade se dará com a comparação dos valores gerados nos modelos abordados. Já a análise de qualidade no contexto deste trabalho, levará em conta a simplicidade (verbosidade) do modelo resultante. A interpretação, tanto do modelo, como dos dados em si, é o que efetivamente pode

proporcionar autonomia de análise nas áreas de negócio. Assim, terá grande peso na avaliação.

A natureza do processo explicado nas sessões anteriores, leva à hipótese de que a abordagem de ETL, sob o aspecto do processamento, tenha performance inferior ao que possa ser proporcionado pela abordagem de ELT. Sob esta mesma ótica, a presunção sobre o aspecto da qualidade é de que aquela obtida no modelo de ETL seja superior à obtida no modelo de ELT. Assim, na reunião do que há de mais vantajoso em cada modelo, firma-se a hipótese de que o ELTL tenha o melhor resultado em uma avaliação ponderada entre os dois fundamentos (velocidade e qualidade).

5 DESENVOLVIMENTO DA PROPOSTA

Neste capítulo está descrita a implementação dos modelos abordados: ETL, ELT e ELTL (processo proposto). Ele tem a finalidade de apresentar, de forma descritiva, o aspecto prático do que fora anteriormente apontado. A partir dos critérios estabelecidos na seção 4.5 (aderência, mínima diversificação, complexidade e custo), foram empregadas para este desenvolvimento as seguintes plataformas: Microsoft SQL Server e Apache Hadoop.

Os dados utilizados para esta experimentação foram extraídos com a devida ciência e autorização, de uma indústria de porte médio, que utiliza como seu sistema de gestão o ERP TOTVS Datasul, sobre banco de dados Progress Open Edge. Trata-se de uma base de movimentação de estoque, com dados históricos que compreendem o período de 2015 até 2019. Do volume de 18 milhões de registros disponibilizados são obtidas, ao fim da execução, informações relativas à produção, compras e vendas desta organização. Pelas razões óbvias de confidencialidade e segurança da informação, o nome da empresa, bem como de seus funcionários, clientes e fornecedores, eventualmente abordados, serão na qualidade de pseudônimos.

A escolha deste conjunto de dados para o desenvolvimento tem por base o perfil de demandas cotidianas de uma empresa que atua na área de inteligência de dados – a saber, **Row Up Inteligência de Dados.** Assim, há que se absorver a realidade indesejável e equidistante no que diz respeito às teorias gerais dos sistemas de informação. Em outras palavras, a utilização de uma grande e principal tabela, dotada de enorme diversidade e quantidade de campos, é consciente, intencional e tem foco absolutamente prático.

5.1 EXCEÇÕES

O experimento é delimitado a um conjunto de dados estruturados, essencialmente pela necessidade da comparação quanto ao modelo clássico (ETL). Isto, contudo, não invalida a adoção do ELTL em conjuntos de dados originalmente não estruturados, como fotografias e textos. Afinal, todos estes que, em sua origem, não possuem estrutura clara e definida, necessitam passar por esta análise – geralmente manual – onde sejam definidos os parâmetros e características

resultantes, formando assim a estrutura do modelo – o que, por sua vez, o enquadra na proposta do ELTL. Estes desenvolvimentos, contudo, serão objeto de trabalho futuro posterior – eventualmente em pós graduação.

O aspecto da incrementalidade das cargas, fundamental em ambos os contextos, também não é objeto deste exercício, visando a redução de sua complexidade e foco ao que se quer provar: a utilização de modelo dimensional em contexto de Big Data. É notório que este fator interfere diretamente no tempo da carga e, portanto, teria forte influência sobre os resultados obtidos neste experimento. Mesmo que o tempo de processamento seja um indicador avaliado nesta implantação, a análise se dará sob a ótica da integralidade do conjunto de dados apresentado. Ou seja, para todas as análises, a incrementalidade não será levada em consideração, o que, por sua vez, traz ao exercício o equilíbrio necessário.

Não será abordada neste desenvolvimento a questão dos agendamentos das rotinas, fluxos e scripts de carga de dados. Este não é um aspecto relevante para aquilo se deseja provar com estes experimentos. As execuções serão controladas manualmente.

5.2 PREPARAÇÃO

Para a estruturação da base de dados inicial, no aspecto da alteração dos dados sensíveis, fora realizada macro substituição de valores, utilizando listas genéricas de nomes, obtidos na internet. Esta preparação foi realizada com a utilização de editor de planilhas, banco de dados SQL e integrador de dados. Por não ser o foco do desenvolvimento, esta etapa não será detalhada.

Para os exercícios aqui propostos é utilizado como fonte o banco de dados DBTC2, cujo conteúdo têm origem no ERP previamente citado. Ele é composto com as estruturas de apresentadas seguir. O detalhamento completo destas estruturas é apresentado no APÊNDICE A.

a) MovimentoEstoque: tabela principal do conjunto, contendo transações de estoque. Tipo da transação (entrada ou saída); espécie da transação (nota fiscal, reporte de produção, refugos, estorno, etc.); data da transação; usuário responsável; produto (código e descrição); cliente/fornecedor (código, nome, natureza, tipo); quantidade; valor (mão de obra, materiais,

gastos gerais), fazem parte desta tabela. Utiliza índices nos campos data da transação e espécie do documento;

b) FamiliaComercial: tabela auxiliar, com detalhes do cadastro do agrupamento dos produtos, no que diz respeito à sua comercialização;
c) FamiliaMaterial: tabela auxiliar, com detalhes do cadastro da família de materiais (produtos);
d) GrupoEstoque: tabela auxiliar, com detalhes do cadastro dos agrupamentos de estoque dos materiais (produtos). Este cadastro está “um nível acima” da família de materiais;
e) Produto: tabela auxiliar com detalhes do cadastro de materiais, e principalmente, sua relação com a taxonomia;

Para consumo dos dados nas ferramentas de fluxo são utilizadas exibições (*views*) que melhor possibilitam a manipulação dos parâmetros para fracionamento de dados, o que é fundamental para os exercícios preliminares. Esta abordagem é adotada para todas as tabelas. Porém, a única que sofre distinção quanto à sua formação original é a **MovimentoEstoque**, por conta do elevado volume de dados contidos e a necessidade de processamentos parciais no decorrer dos testes de desenvolvimento. Seu *script* está apresentado no APÊNDICE B. As demais exibições, entregam rigorosamente todo o conteúdo das tabelas correspondentes – sem transformações.

Apesar das divergências observadas nas bibliografias quanto à necessidade mínima de volume de dados para que se caracterize um BigData – seja em quantidade de registros, seja em *MegaBytes* armazenados – é possível afirmar que os 18 milhões de registros utilizados nesta implementação, que totalizam pouco mais de 90 *GigaBytes* não caracterizam de forma nítida um conjunto de BigData.

5.3 INFRAESTRUTURA

A realização das simulações e execução de cálculos computacionais depende, naturalmente, de recurso de *hardware* e *software*. Neste cenário é utilizado ambiente de virtualização Microsoft Hyper-V, rodando sobre plataforma PC.

5.3.1 Hardware

Para tanto, o *host* físico utilizado é um servidor Lenovo TS 150, dotado de um processador Xeon E3-1225 v5, com 4 núcleos físicos, *cache* de nível 3 com 8 *megabytes* e capacidade de processamento de 3,3 Giga Hertz. No armazenamento volátil, o equipamento contém 56GB de memória RAM, barramento DIMM, na versão DDR4 e taxa de transferência de 2133 Mega Hertz.

No que tange ao armazenamento, este servidor é equipado com controladora RAID *onboard*, modelo Intel C600+. Em seu montante primário, o equipamento em questão é dotado de 4 discos rígidos (HD) de 2TB cada. Modelo Seagate ST2000DM006, com velocidade física de 7200RPM e interface SATA, operando estes, em regime de RAID10. Adicionalmente, como unidades de alta taxa de leitura e escrita (IO), são dispostas no equipamento duas unidades de estado sólido (SSD), com interface SATA, de 240 e 960 Giga Bytes, sendo Kingston SUV400 e ADATA SU630 seus modelos, respectivamente.

Em seu barramento de rede, o servidor é equipado com duas interfaces, sendo uma integrada (*on-board*), de modelo Qualcomm Atheros AR8151 e capacidade de tráfego de 1 Gbps (1000 Mbps), e uma avulsa (*off-board*), modelo Realtek PCIe GBE e capacidade 1 Gbps (1000 Mbps). A primeira é dedicada à comutação das máquinas virtuais (VMs) descritas a seguir. Há no ambiente um *switch* gerenciável HP 1920 24 portas Full Gigabit, no qual o equipamento físico está conectado e por onde trafegam, inevitavelmente, os pacotes trocados entre as máquinas virtuais. O sistema operacional do *host* físico é o Microsoft Windows Server 2019 Datacenter 1809, Compilação 17763.107, 64 bits.

Este é um recurso de produção utilizado na empresa **Row Up Inteligência de Dados.** Portanto, além de hospedar os sistemas necessários ao presente desenvolvimento, ele compartilha recursos com as operações cotidianas da organização, como armazenamento de arquivos, autenticação de usuários, gerenciamento de rede, entre outros. Durante todas as simulações, os serviços fundamentais como autenticação e concessão de endereços de rede foram mantidos operacionais. Já serviços secundários como armazenamento de arquivos e execução de aplicações não críticas como wifi, foram interrompidos.

5.3.2 Máquinas Virtuais

A presente experimentação é baseada em duas máquinas virtuais (VM). A primeira, designada **DEV3H15,** compartilha a função de armazenamento das fontes de dados brutos e também de processamento de ETL. Já a segunda, tem o objetivo de viabilizar a utilização do ELT e ELTL. Ou seja, aquilo que se refere à Big Data. Sua designação: **CLOUDERA**. As alocações de recurso de hardware estabelecidas, o foram considerando o mínimo recurso necessário para a operação estável e contínua das aplicações.

5.3.2.1 DEV3H15

Dotada de sistema operacional Microsoft Windows 10 Professional, na versão 1903, compilação 18362.836, possui reserva de recursos no *host* físico. Quanto à memória RAM há reserva de 8 GB (oito *gigabytes*). Em termos de processamento, são 4 (quatro) os núcleos disponibilizados, com reserva de 10% (dez porcento) da capacidade do *host* físico. No armazenamento, os 2 (dois) discos virtuais empregados, estão dispostos no disco de estado sólido de maior capacidade do *host* físico. Isto lhes confere a melhor taxa de transferência possível no *hardware* em questão. Em termos de endereçamento de rede é utilizado de forma fixa o IP 10.10.1.244. Embora faça parte da mesma sub rede de produção, este endereço está fora do range de concessão DHCP existente no contexto, o que lhe confere a disponibilidade e estabilidade necessárias.

O objetivo desta VM é hospedar não somente os bancos de dados relacionais, fundamentais para formulação do Data Warehouse, como também o mecanismo de processamento de dados, que é utilizado para execução do ETL. Assim, está instalado nesta máquina o SGBD Microsoft SQL Server 2019 Enterprise Edition. Como parte deste produto, está disponível o mecanismo de processamento de dados (Microsoft SQL Integration Services) e também de agendamento de tarefas (Microsoft SQL Server Agent). Adicionalmente, as ferramentas de gerenciamento Microsoft SQL Server Management Studio e Microsoft Visual Studio Professional 2019 com pacote de Extensão SQL Server Integration Services Projects (VS Data Tools) dão condições às manipulações de dados necessárias. Complementarmente, este ambiente

comporta a execução da ferramenta de *frond-end*, através da qual são realizadas as validações de integridade e qualidade dos modelos: o Microsoft Power BI Desktop.

5.3.2.2 CLOUDERA

Originalmente obtida no padrão para hospedeiro Oracle Virtual Box, a VM em questão fora convertida para Hyper-V, para operação em consonância com a infraestrutura disponível. Trata-se de um sistema operacional Cloudera Express, versão 5.13.0, compilação 55. Esta é uma distribuição do Apache Hadoop, sobre Linux. Com reserva de memória RAM de 18 GB (dezoito *gigabytes*), possui 4 (quatro) núcleos de processamento, com reserva de 10% (dez porcento) da capacidade do *host* físico. Baseado em um único disco virtual, esta VM (assim como a VM anteriormente apresentada), utiliza o disco de estado sólido de 960GB do *host* físico. Isto lhe confere a melhor taxa de transferência de bytes possível dentro do contexto apresentado. No que diz respeito à comunicação em rede, fora alocado para este sistema, o IP 10.10.1.218, que igualmente se encontra fora do *range* de concessão DHCP. Destaque para a elevada exigência de hardware do Cloudera, que foi um dos dificultadores da sua operacionalização por causar eventual instabilidade e indisponibilidade da VM.

Muito embora o BDT tenha por essência a distribuição de processamento, caracterizando assim um cluster – não é este o caso desta experimentação. Por limitações de disponibilidade de hardware, este é um BDT operando *stand alone*. Contudo, esta não é uma limitação para a prova do modelo, pois é sabido que a performance da execução dependerá, primariamente, da disponibilidade de recursos de hardware nos ambientes em que venha a ser implementada.

Esta VM tem a finalidade de sediar tudo o que neste projeto diz respeito à Big Data. Como parte padrão da instalação Hadoop Cloudera estão os serviços de armazenamento, apresentação, ingestão, processamento e gerenciamento. No Quadro 3 são apresentados os termos da plataforma Hadoop correspondentes a estes e outros fundamentos relevantes à proposta.

Quadro 3 – Termos da plataforma Hadoop

TERMO	**DESCRIÇÃO**
BEELINE	Ferramenta que implementa a manipulação de dados do Hive
HDFS	Sistema de arquivos distribuído, responsável pelo armazenamento dos dados na plataforma

HIVE	Sistema de Data Warehouse, responsável por proporcionar estrutura de acesso (apresentação) aos dados armazenados, através de linguagem SQL
HUE	Sistema web de interface gráfica, através do qual é possível operar de forma mais simplificada a plataforma
IMPORT	Comando da ferramenta Sqoop para execução da importação (ingestão) de dados para a plataforma
MAP & REDUCE (OU MAPREDUCE)	Mecanismo de processamento de dados, construído para computação distribuída
SPARK	Mecanismo e ferramenta de processamento de dados, construído para computação distribuída
SQOOP	Ferramenta que faz o interfaceamento entre bancos de dados relacionais e a plataforma

Fonte: HÜBNER (2020)

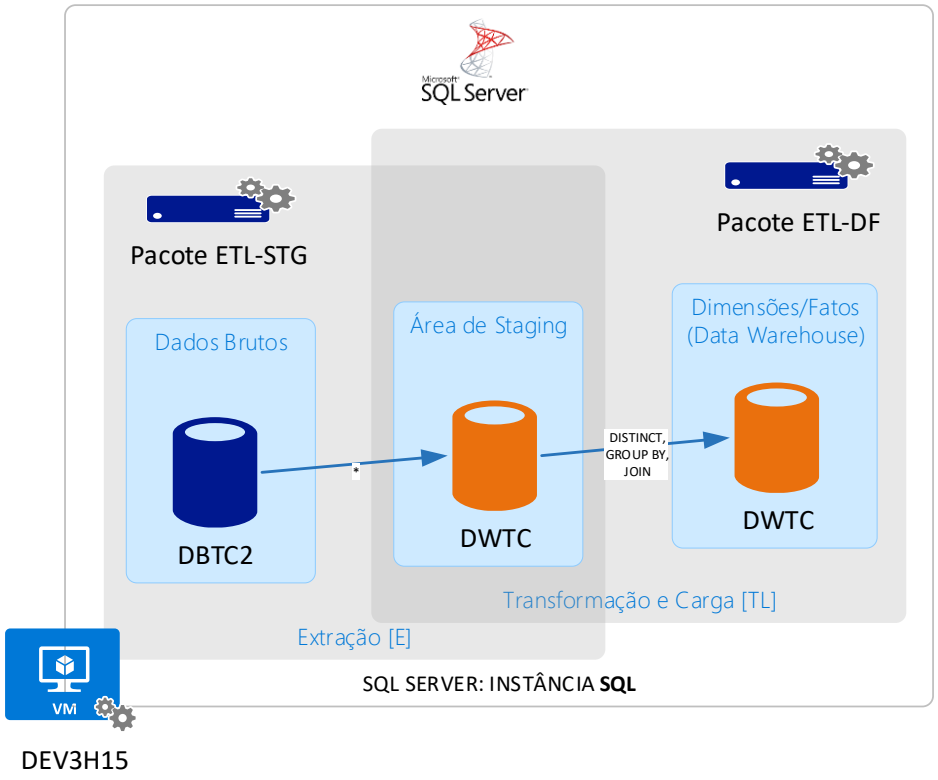
5.4 OBJETOS DE SAÍDA

Ao término da execução dos 3 modelos (ETL, ELT, ELTL), são entregues as análises de valor e quantidade de Compras, Vendas e Produção, agrupadas por período (ano e mês), filial e produto. Todas extraídas das tabelas brutas disponibilizadas. Isto se dará, essencialmente, através da separação dos registros conforme as espécies de documentos e grupos de estoque de produtos. As **compras** são compostas pelos registros cujas espécies possuem o código 21. As **vendas** são compostas pelos registros onde a espécie possui código 22 e os produtos pertencem aos grupos de estoque 80 e 81. Já a **produção** é obtida pelos registros onde a espécie seja 1, 8 e 25. Respeitadas as mesmas características de agrupamento em cada modelo exercitado, os valores devem ser absolutamente idênticos.

5.5 IMPLANTAÇÃO DO ETL

A lógica do macrofluxo de dados do paradigma ETL realizado pode ser observada na Figura 12. Sendo o mais tradicional de todos, o processo de ETL é implementado na VM DEV3H15, utilizando-se fundamentalmente do VS Data Tools. Ele se inicia pela Extração, que é materializada com a composição da área de Staging. Posteriormente, é realizada a Transformação, momento em que ocorre também a Carga dos dados transformados. A interação no experimento se dá entre os bancos de dados SQL DBTC2 e DWTC. Os pacotes referenciados nesta sessão (ETL-STG e ETL-DF) são conjuntos de fluxos, organizados e dispostos no contexto do VS Data Tools.

Figura 12 – Diagrama do ETL implementado



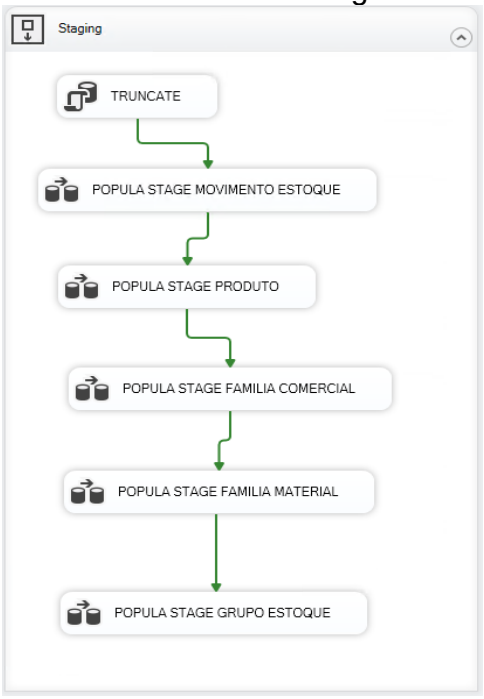
Fonte: HÜBNER (2020)

O desenvolvimento tem por premissa a realização do modelo estrela, sobre banco de dados relacional. Esta, por sua vez, é uma generalização do modelo dimensional nativo, oferecido pelas ferramentas OLAP, que não são alcançadas nesta implementação.

5.5.1 Extração

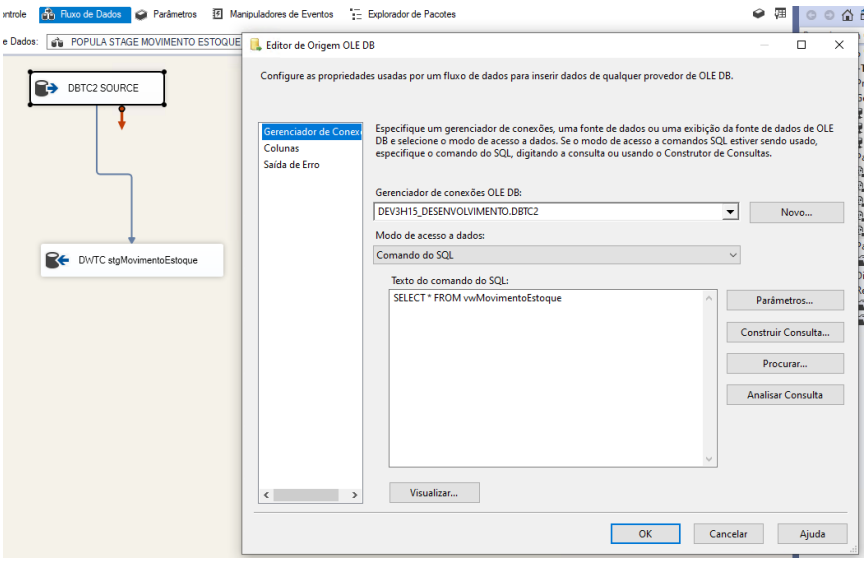
Esta etapa consiste na extração bruta dos dados da sua fonte e sua disposição na área de pré-processamento, no banco de dados designado ao Data Warehouse (DWTC). Sua implementação ocorre através de um fluxo de dados desenvolvido no VS Data Tools e ilustrado na Figura 13. A primeira ação deste fluxo é realizar a limpeza das tabelas dispostas na área de staging, através de um comando SQL "*truncate*". A segunda ação consiste na leitura dos dados na base de origem, simultaneamente à gravação na base corrente, conforme ilustra a Figura 14. O conjunto destas ações é representado pelo pacote **ETL-STG**. As tabelas stage resultantes deste movimento são stgMovimentoEstoque, stgProduto, stgFamiliaComercial, stgFamiliaMaterial, stgGrupoEstoque.

Figura 13 – Container de carga da staging



Fonte: HÜBNER (2020)

Figura 14 – Fluxo origem-destino com comando de seleção



Fonte: HÜBNER (2020)

5.5.2 Transformação e carga

Representada pelo pacote ETL-DF, esta etapa é onde ocorre o que de mais expressivo há no tratamento dos dados: a desnormalização. É através dela que o *datamart*, ou conjunto de dados, se torna humanamente razoável e inteligível.

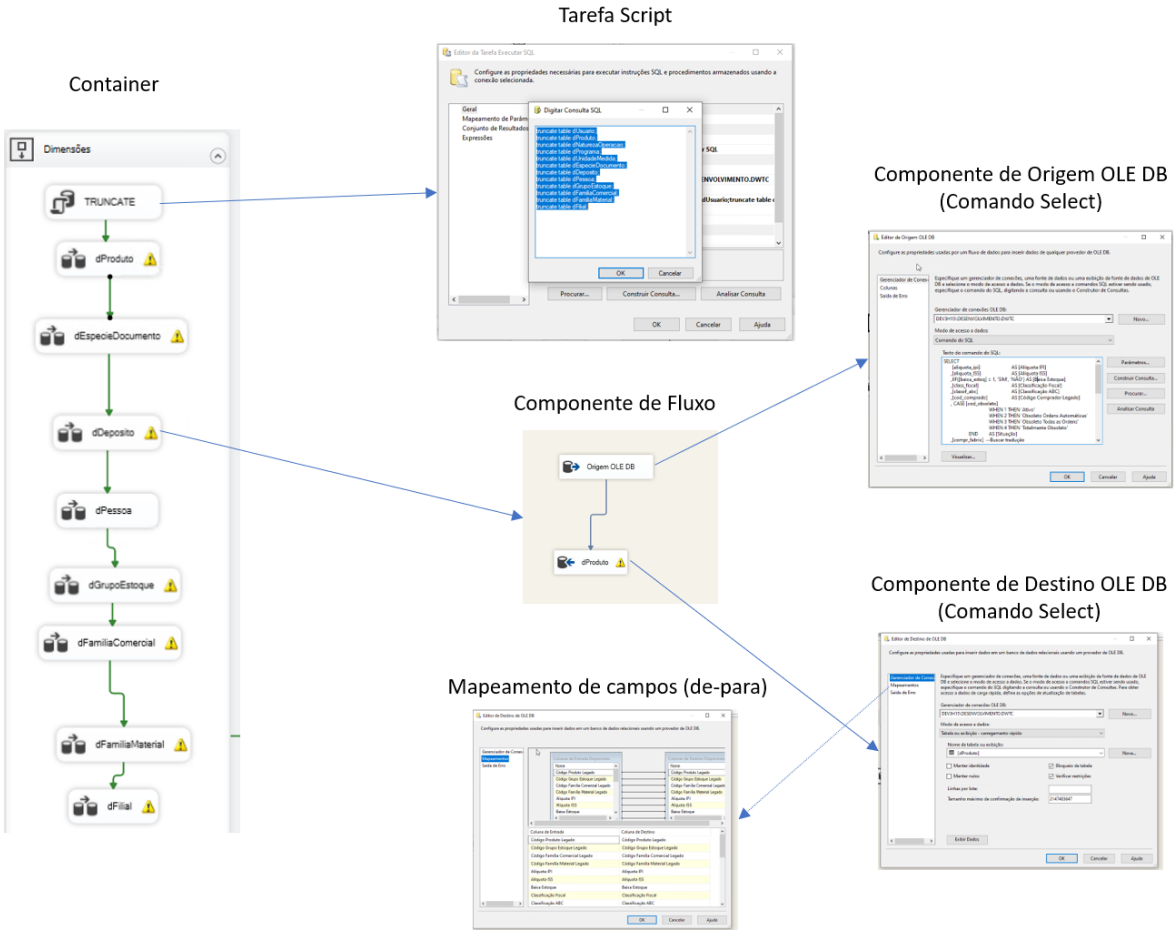
Partindo da premissa pré-estabelecida de que entidades (tabelas) fato devem possuir exclusivamente campos (atributos) de data, valores e chaves estrangeiras, cabe a esta etapa a criação das entidades (tabelas) dimensões. Elas terão a missão de proporcionar a visão dos fatos, sob as diversas óticas desejadas. É através das dimensões que se torna possível a visão aprofundada, correlacionada e fracionada das informações.

Neste exercício, as dimensões são compostas a partir de tabelas de cadastro – o que aqui é denominado “dimensões carregadas”, mas também pela seleção distinta de valores da tabela de movimentos – o que é denominado “dimensões extraídas”. Cada dimensão é representada por um fluxo dentro do contêiner, como disposto na Figura 15. Assim, após a limpeza destas entidades, concretizada pelo comando *truncate table*, é realizada a população delas. A mesma Figura 15 demonstra o fluxo de origem e destino de uma dimensão. Nela é possível observar o comando de seleção de dados de uma dimensão, com as suas transformações. Nestes comandos é implementada a elevação de verbosidade dos nomes dos atributos. Como exemplo, o campo “cod_obsoleto” do cadastro de produtos, refere-se à **situação** ou **status** do mesmo. Esta tradução é realizada para todos os campos.

As dimensões carregadas resultantes deste processo são: dProduto, dFamiliaMaterial, dFamiliaComercial e dGrupoEstoque. Há também as dimensões extraídas da staging de movimento – que advêm de procedimentos como o de seleção distinta: dEspecieDocumento, dDeposito, dPessoa e dFilial. Os códigos SQL completos da população das dimensões se encontram no APÊNDICE C. Neles podem ser observados os campos com seus nomes de origem e seus nomes alterados/substituídos. Estes nomes substituídos são, por si só, a explicação do que representam. Esta é efetivamente a implementação da elevação de verbosidade, anteriormente apresentada.

A implementação das SK’s ocorre através da definição das entidades de destino (tabelas de dimensões). Neste âmbito é que se define o campo auto incremental do tipo inteiro. Para as dimensões adotadas neste contexto o APÊNDICE D apresenta os códigos de implementação.

Figura 15 – Fluxo de transformação de dados expandido

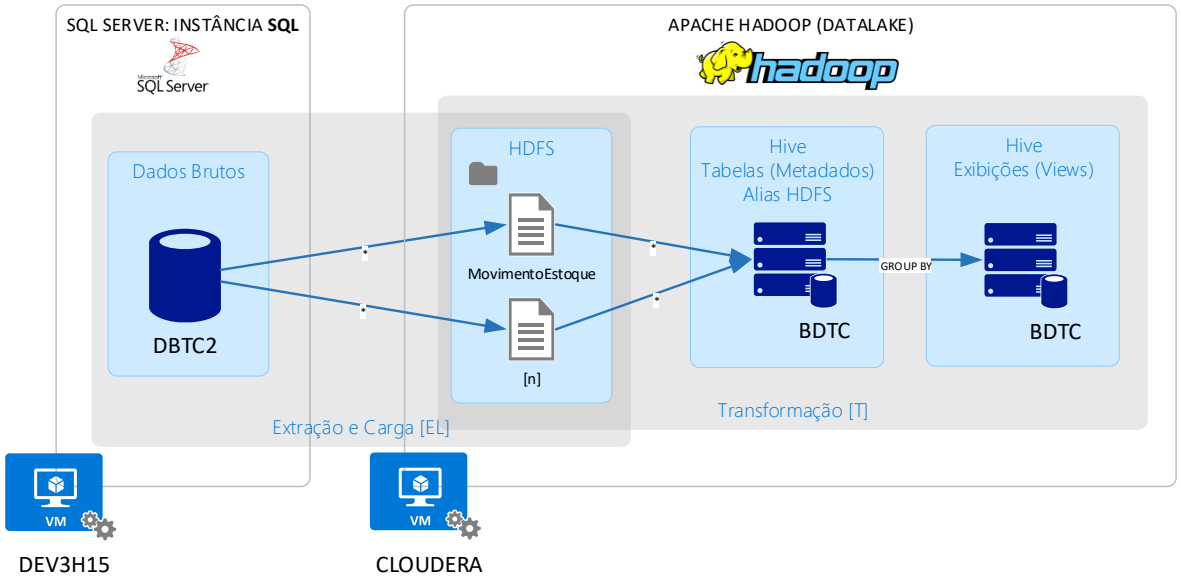


Fonte: HÜBNER (2020)

5.6 IMPLANTAÇÃO DO ELT

A execução das tarefas de processamento de dados ELT – que são aquelas que possuem característica de Big Data, no que diz respeito à presente execução, são de responsabilidade da VM CLOUDERA. O fluxo sintético desta operação está apresentado na Figura 16. Inicialmente, o conjunto de dados é ingerido para dentro do Hadoop e, posteriormente, é realizada a criação da estrutura e a transformação dos dados com a ferramenta Hive – ambos detalhados na sequência.

Figura 16 – Diagrama do ELT Implementado



Fonte: HÜBNER (2020)

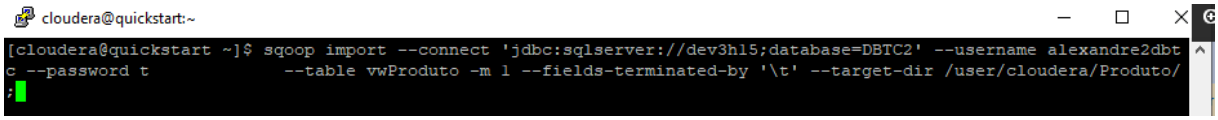
5.6.1 Extração e Carga

Como sugere a Figura 16, os dados importados são armazenados no formato de arquivos. Cada tabela de origem corresponde a um arquivo texto separado. Originalmente armazenados nas tabelas do banco de dados SQL na VM DEV3H15, estes dados são importados para o DL Hadoop na área HDFS (Hadoop Data File System) do CLOUDERA. Isto é realizado através da ferramenta “import”, por sua vez executada em terminal SSH, como demonstra a Figura 17. Este comando tem como parâmetros fundamentais a *string* de conexão com o banco de dados e sua autenticação; o nome da tabela/exibição da qual a leitura deve ser feita; o separador de campo desejado para o arquivo texto de destino; o diretório de destino no ambiente HDFS. Os códigos completos de importação estão dispostos no APÊNDICE E.

Este primeiro estágio do processamento não realiza críticas ou tratamentos de dados. Ou seja, a carga é integral, tal qual oferecido pela origem de dados. A partir da realização desta carga, os arquivos podem ser observados no sistema de arquivos como demonstrado na Figura 18, através da interface gráfica (GUI) disponibilizada na ferramenta HUE. Seu conteúdo também é passível de consulta, na mesma ferramenta, como demonstrado na Figura 19. Os arquivos de dados gerados nesta estrutura experimental são: FamiliaComercial, FamiliaMaterial, GrupoEstoque, MovimentoEstoque, Produto.

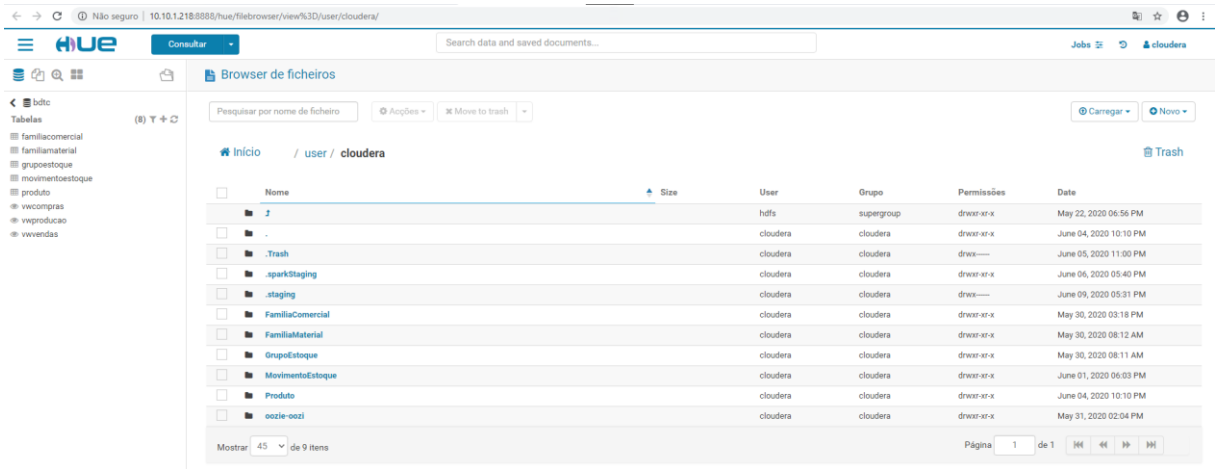
Outro fator relevante é que estes arquivos são armazenados sem metadados. Ou seja, não há cabeçalho nem tipagem definida. Assim, temos aqui um banco de dados NoSQL. Todavia, se faz fundamental haver compreensão sobre estas estruturas, seja através de exploração, análise dos metadados das fontes ou mesmo de documentação para, posteriormente, consumir os dados.

Figura 17 – Exemplo do comando Import



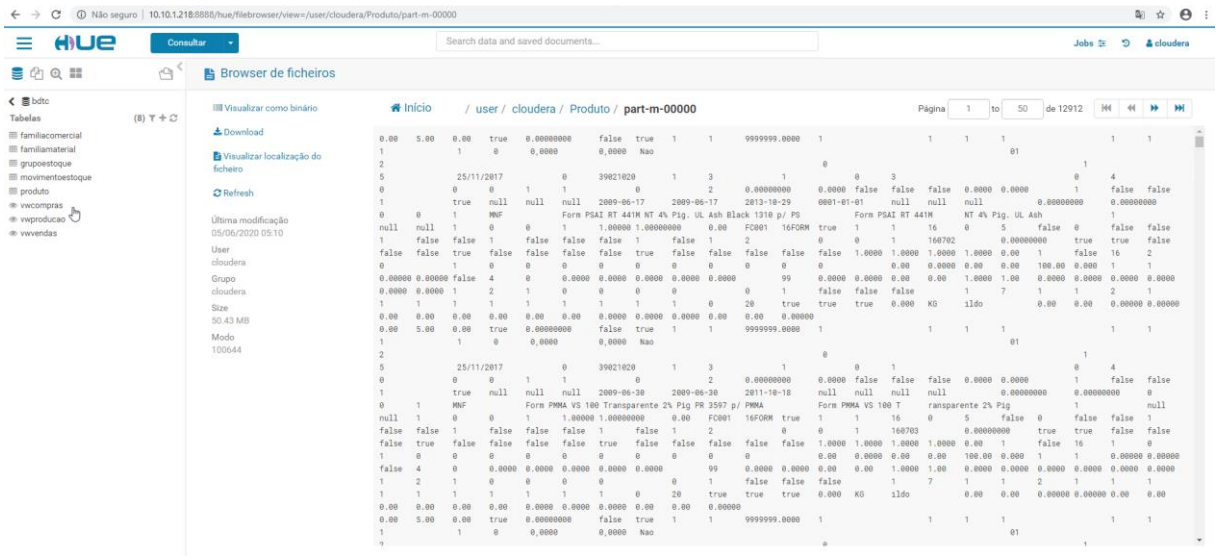
Fonte: HÜBNER (2020)

Figura 18 – Interface de operação HUE



Fonte: HÜBNER (2020)

Figura 19 – Consulta de conteúdo de arquivo na interface HUE



Fonte: HÜBNER (2020)

5.6.2 Transformação

Esta é a etapa onde são definidas as “máscaras” dos arquivos criados na etapa anterior. Estas máscaras são, na realidade, os metadados das estruturas. Sua finalidade é fundamental: para apresentação e consumo dos dados armazenados no DL com comandos no padrão SQL. Estes metadados seguem a estrutura original das tabelas, que neste caso eram conhecidas em função das estruturas de origem dos dados.

Entra em cena mais uma ferramenta da suíte Hadoop: o Hive. Nele, a orientação lógica é similar à de um banco de dados convencional. Afinal, são criadas tabelas, com campos de tipos definidos. E para a execução de procedimentos no Hive, embora seja possível a utilização do HUE, é disponibilizada dentro do pacote Hadoop, a ferramenta Beeline a ser executada em linha de comando (SSH), a partir da qual é utilizado o programa Sqoop.

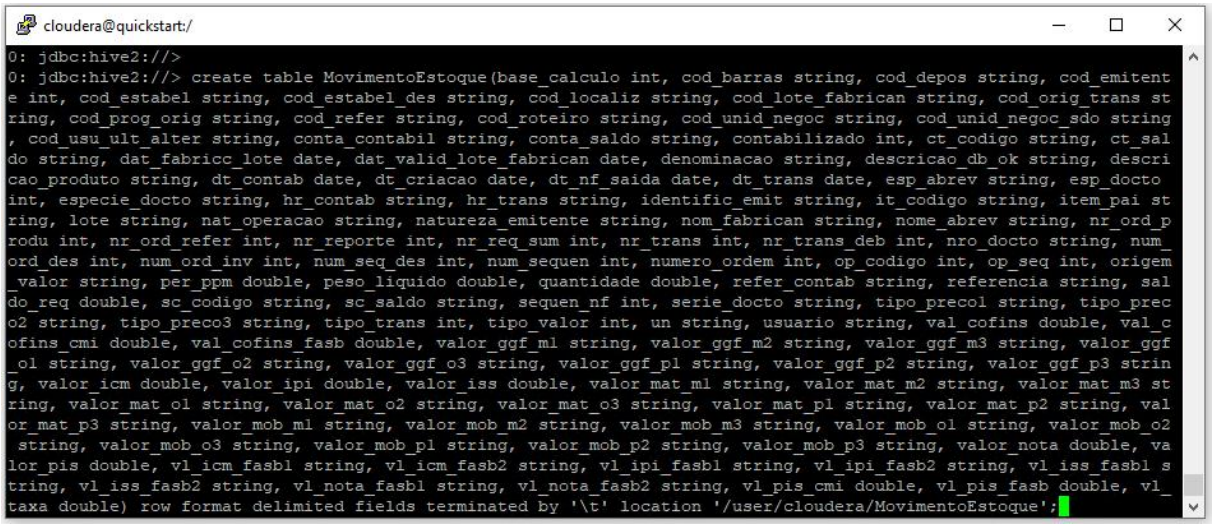
A primeira etapa é a criação do banco de dados. Para o presente exercício, fora criado o banco de dados **bdtc2**, com auxílio do comando “create database <dbname>”. A segunda etapa diz respeito à criação das tabelas. O comando para criação das tabelas e referência aos arquivos pode ser observado na Figura 20 e tem a seguinte sintaxe: “*create table <tName>(<field1> <type1>, <fieldn> <typen>) row format delimited fields terminated by '<same_separator_file>' location '/dir/<sourceFileName>'*”. Os parâmetros são basicamente o nome da tabela que se deseja visualizar no Hive; seus campos sempre seguidos pelos tipos; o delimitador de campos e a localização do arquivo de origem no HDFS. Os códigos completos constam no APÊNDICE F.

Estas tabelas fazem referência aos arquivos do HDFS. Ou seja, não há armazenamento efetivo no Hive. Portanto, excluídos os arquivos do HDFS, imediatamente são indisponibilizados os dados no Hive. Realizada esta atividade, está disponível para consulta com linguagem SQL a íntegra do conteúdo importado. Isto, por sua vez, pode ser analogamente referenciada à área de *staging* utilizada no ETL.

Considerando que os arquivos mencionados estão armazenados no HDFS em sua integralidade e, a informação que se deseja consumir corresponde a um determinado agrupamento de dados, faz-se necessária sua manipulação, implementada através de exibições (*views*). Tal qual um banco de dados relacional clássico, o Hive permite, além da criação de tabelas, a criação de exibições. Assim,

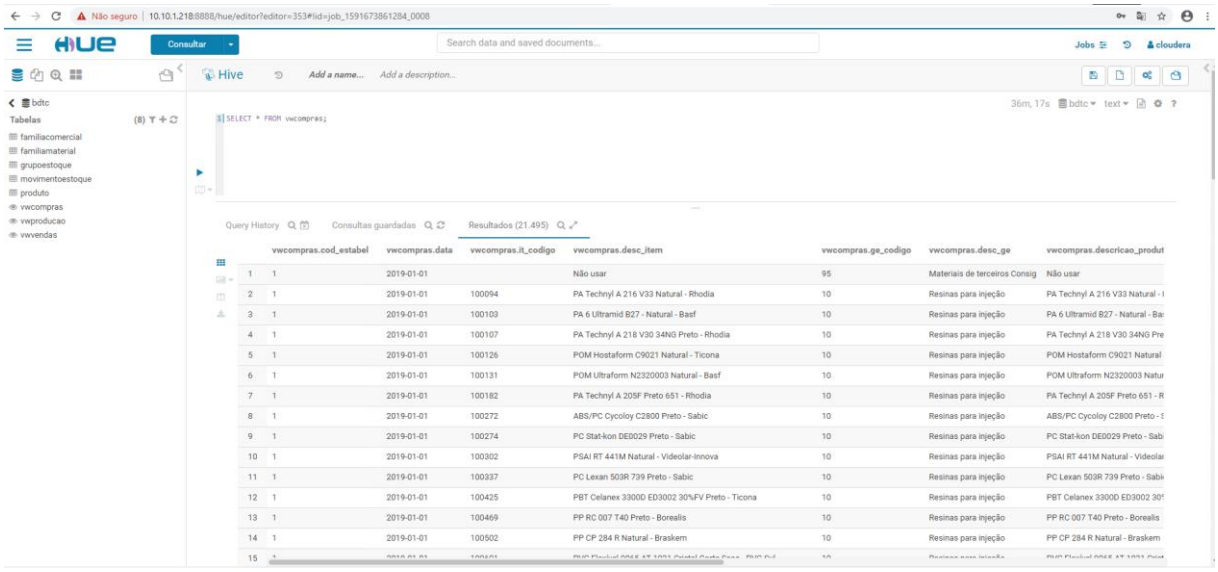
as exibições **vwcompras, vwvendas** e **vwproducao** são a implementação do paradigma ELT, que permitem o consumo objetivo das informações. Uma amostra de seu resultado pode ser observada na Figura 21. Destaque para o fato de que os campos das tabelas (e por consequência das exibições) mantém seus nomes de origem – falta do exercício da elevação de verbosidade – que não é previsto no processo de ELT. Os códigos de criação das exibições citadas estão disponíveis no APÊNDICE G.

Figura 20 – Comando de criação da tabela no Hive



Fonte: HÜBNER (2020)

Figura 21 – Amostra do resultado da seleção de uma exibição (view) no Hive

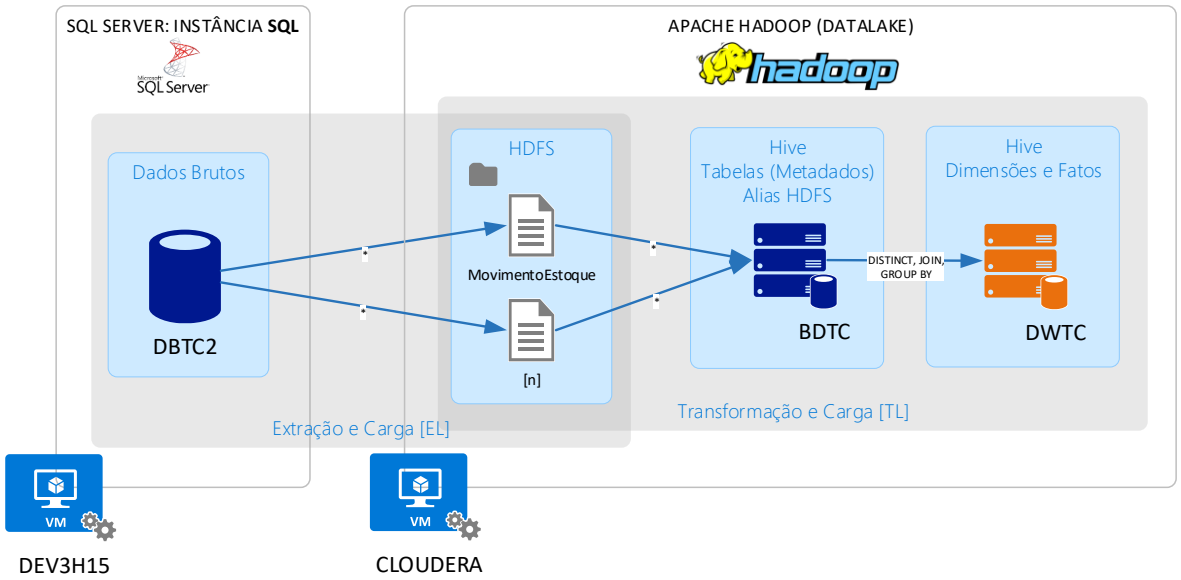


Fonte: HÜBNER (2020)

5.7 IMPLANTAÇÃO DO ELTL

O ELTL, por fim, é aqui relatado em termos práticos. Em termos de sequência lógica, sua abordagem é de veras similar à ELT. Esta semelhança ELT se deve ao fato de que sua implementação é fortemente apoiada na tecnologia de Big Data Hadoop. Sua distinção está, fundamentalmente, no fato de que a disponibilização da informação para o seu consumidor ocorre a partir de um banco de dados do Hive, chamado DWTC, e não mais das exibições (*views*) adotadas no ELT. Ou seja, as transformações de dados percorrem caminho distinto da abordagem convencional do BigData, resultando em um DataWarehouse, como se pode notar na Figura 22. Por essência, o DW segue premissas distintas de outra base convencional qualquer. Seu conteúdo são entidades (tabelas) de fato e dimensões. Sua formulação respeita a premissa do esquema estrela. E a tabela fato, além de conter somente datas, valores numéricos e chaves de dimensões, é ligada a estas através de SK's. Lembrando que estas chaves são campos numéricos do tipo inteiro, com geração automática.

Figura 22 – Diagrama do ELTL implementado



Fonte: HÜBNER (2020)

5.7.1 Extração e Carga

Em termos técnicos, esta etapa é absolutamente idêntica à sua correspondente no processo ELT. Ou seja, em nada se distingue a extração e carga realizada no ELT da que é realizada no ELTL. Assim sendo, em termos práticos, o

processamento não é repetido, pois os arquivos dispostos no DL HDFS descritos na página 62 são o ponto de partida para a etapa seguinte, esta sim distinta: transformação e carga.

5.7.2 Transformação e Carga

Esta é a etapa onde há, efetivamente, distinção com relação às abordagens convencionais. Ela é inspirada no conceito multidimensional. Porém, sendo o Hive uma generalização do banco de dados relacional, a incorporação consciente é do esquema estrela. Assim, a primeira atividade essencial é a estruturação dos arquivos texto, com a utilização do banco de dados BDTC, no Hive. Isto é realizado através da ação de importação no Hadoop HDFS descrita na página 62. Esta atividade será igualmente aproveitada, sendo desnecessária sua execução duplicada. Portanto, a ação de transformação efetivamente se vale do que fora anteriormente processado para o ELT, resultando assim na base de dados BDTC no Hive.

A ação seguinte vem a ser a criação do banco de dados DWTC no Hive, através do comando “create database dwtc”, executado em linha de comando, na ferramenta Beeline, previamente referenciada. Então, a partir do conteúdo integralmente acessível através das tabelas **familiacomecial, familiamaterial, grupoestoque, movimentoestoque** e **produto**, disposto no banco de dados BDTC, na área que pode ser analogamente definida como área de *staging* deste processo, são formuladas inicialmente as dimensões. Isto, contudo, vale para as dimensões carregadas.

É necessário pontuar sobre as dimensões extraídas. Ou seja, aquelas que advêm de uma seleção distinta com origem na tabela de movimentos. Estes casos, seguem um caminho sutilmente distinto. Em razão da criação das SK’s, que dependem do comando “row_number() over()”, com o terminador “CLUSTER BY <field_name>”, e, por sua vez, somente podem ser executados em seleções de listas completas – ou seja, não podem ser utilizadas junto ao comando “distinct”, neste caso há necessidade da utilização de exibições intermediárias. Para tal, é adotada a criação de um banco de dados intermediário (dwtcstg), no qual serão criadas exibições conforme detalhamento no APÊNDICE H.

A criação das dimensões segue as melhores práticas relacionadas à elevação de verbosidade, como se pode observar no exemplo da

Figura 23 onde o campo “cod_obsoleto” é ajustado para “Situação”, e seus valores numéricos (de 1 a 4), traduzidos em expressões humanamente compreensíveis (Ativo, Obsoleto Ordens Automáticas, Obsoleto Todas as Ordens e Totalmente Obsoleto). Além disto, nesta atividade são criadas as SK’s, com a utilização do comando “row_number() over()” e o terminador “CLUSTER BY <field_name>”. A íntegra dos códigos para criação das dimensões do DW Hive consta no APÊNDICE I.

Figura 23 – Exemplo de criação de dimensão no Hive

```
/*CRIA TABELA DIRETA COM EXPRESSÃO DA DIMENSÃO, NO DW*/
/*DIMENSÃO PRODUTO*/
CREATE TABLE dwtc.dProduto AS
    SELECT
        row_number() over () AS  `Código Produto`
      ,`aliquota_ipi`        AS `Alíquota IPI`
      ,`aliquota_ISS`        AS `Alíquota ISS`
      ,IF(`baixa_estoq` = 1, 'SIM', 'NÃO') AS `Baixa Estoque`
      ,`class_fiscal`        AS `Classificação Fiscal`
      ,`classif_abc`         AS `Classificação ABC`
      ,`cod_comprado`        AS `Código Comprador Legado`
      , CASE `cod_obsoleto`
        WHEN 1 THEN 'Ativo'
        WHEN 2 THEN 'Obsoleto Ordens Automáticas'
        WHEN 3 THEN 'Obsoleto Todas as Ordens'
        WHEN 4 THEN 'Totalmente Obsoleto'
      END   AS `Situação`
      ,`compr_fabric`  --Buscar tradução
      ,`curva_abc`           AS `Curva ABC`
      ,`data_implant`        AS `Data Implantação`
      ,`data_liberac`        AS `Data Liberação`
      ,`data_obsol`          AS `Data Obsolescência`
      ,`deposito_pad`        AS `Depósito Padrão`
      ,`desc_item`           AS `Descrição`
      ,`fm_cod_com`          AS `Código Família Comercial Legado`
      ,`fm_codigo`           AS `Código Família Material Legado`
      ,`ge_codigo`           AS `Código Grupo Estoque Legado`
      ,`it_codigo`           AS `Código Produto Legado`
      ,`lote_minimo`         AS `Lote Mínimo`
      ,`lote_multipl`        AS `Lote Múltiplo`
      ,`un`                  AS `Unidade Medida`
      ,`usuario_alt`         AS `Usuário Alteração`

      FROM bdtc.Produto
      CLUSTER BY `Código Produto`;
```

Fonte: HÜBNER (2020)

Tendo por referência as premissas estabelecidas, para o que neste contexto possa ser chamado de entidade fato em um DW, cabe a esta etapa eliminar da tabela resultante, qualquer dado diferente de datas, valores numéricos e chaves estrangeiras (SK). Tal qual se utiliza na implementação do ETL, esta substituição ocorre através do relacionamento prévio entre as dimensões, já estruturadas, e a tabela de dados brutos, que contém por sua vez, a totalidade de dados disponíveis na área de *staging*. Este relacionamento é o que proporciona à expressão, a possibilidade de gerar dados respeitando à premissa data, valores, chaves. Não obstante, esta mesma ação tem a finalidade de gerar os agrupamentos e filtros necessários, conforme a necessidade imposta pela regra de negócio. Um exemplo de comando utilizado para a criação de tabela fato no banco DWTC do Hadoop Hive, pode ser notada na Figura 24. Os

códigos completos de criação das tabelas fato deste contexto estão disponíveis para consulta no APÊNDICE J.

Figura 24 – Exemplo de criação de tabela fato ho banco DWTC no Hive

Fonte: HÜBNER (2020)

5.8 EXECUÇÃO E RESULTADOS

Foram realizadas mais de cinco execuções do processamento de cada um dos paradigmas, com cargas parciais de dados no decorrer da implantação, tendo sido possível observar o mesmo padrão de comportamento de tempo a seguir descrito em todas estas simulações. Considerando, contudo, que o objetivo deste experimento não é a avaliação dos resultados de cada modelo isoladamente, mas sim a comparação dos paradigmas entre si, em termos relativos e, considerando o alto volume de dados frente ao *hardware* disponível, foram realizadas três execuções completas dos fluxos, plenamente computadas.

Como premissa fundamental dos processamentos, respeitou-se a execução individual de cada fluxo de dados, para garantia da maior disponibilidade de recursos possível a cada etapa. O resultado destas execuções é apresentado nesta seção, sob a ótica de três atributos: performance, integridade e qualidade. O primeiro aspecto diz respeito, essencialmente, ao tempo de processamento de cada etapa de cada paradigma, tendo em vista a soma do tempo das etapas. O segundo aspecto diz respeito à quantidade de registros e à soma dos valores das tabelas (ou exibições) resultantes. O terceiro aspecto diz respeito à qualidade da estrutura de dados

oferecida ao usuário final – em que pese a facilidade de interpretação e correlação das tabelas e campos.

Nas avaliações não é levado em consideração o tempo de preparação das tabelas que compõe o banco de dados SQL DBTC2. Este, afinal, seria o banco de dados disponível, em um ambiente de produção. As medições, portanto, são computadas somente nas etapas seguintes à preparação.

As circunstâncias do presente desenvolvimento e o resultado das execuções, a seguir detalhados, permitem a confirmação das hipóteses apontadas na seção 4.5. O paradigma ELTL é superior ao ETL em performance, e superior ao ELT em qualidade.

5.8.1 Performance

A medição da performance leva em consideração a soma de tempo de cada etapa dentro do paradigma. Como observamos no Quadro 4, o melhor tempo agregado de execução do paradigma ELT foi de 32 minutos, na execução 3, ficando assim clara a sua vantagem neste quesito. O paradigma ELTL, por sua vez, apresenta tempo de carga de 127 minutos (também na execução 3). O paradigma ETL, em contrapartida, tem tempo computado de 190 minutos para o processamento do mesmo conjunto de dados, na mesma execução. Cabe destacar que o tempo de carga entre o conjunto final de dados de cada paradigma e a ferramenta de visualização de dados, não fazem parte desta análise, pois estão fora do escopo dos paradigmas exercitados. Em termos comparativos e relativos, as três execuções completas tiveram performance muito similar, com variação inferior a 5%.

Quadro 4 – Tempo de execução paradigmas

Duração (minutos)	**Execução**		
PARADIGMA/ETAPA	**1**	**2**	**3**
ELT	**33**	**32**	**32**
1 - IMPORT SQOOP	28	27	27
2 - CREATE TABLE BDTC	05	05	05
ELTL	**132**	**133**	**127**
1 - IMPORT SQOOP	27	28	27
2 - CREATE TABLE BDTC	07	06	05
3 - CRIAÇÃO DIMENSÕES CARREGADAS	08	09	07
4 - CRIAÇÃO DIMENSÕES EXTRAÍDAS	31	31	30
5 - CRIAÇÃO TABELAS FATO	59	59	58
ETL	**195**	**192**	**190**
1 - ETL-STG	189	188	187
2 - ETL-DF	06	04	03

Fonte: HÜBNER (2020)

5.8.2 Integridade

A aferição da integridade dos dados se deu através da leitura e comparação das tabelas resultantes em cada paradigma. Os conjuntos se mostraram coerentes como se pode observar nos valores apresentados no Quadro 5. Nele é possível observar cada objeto (que representa uma tabela distinta na origem dos dados ou uma tabela no conjunto resultante. Também se verifica o paradigma e ambiente, assim como a identificação de cada artefato. Por fim, a quantidade de registros, é o que efetivamente consolida a consistência dos conjuntos processados

Quadro 5 – Contagem de registros nas tabelas

OBJETO	PARADIGMA	AMBIENTE	IDENTIFICACAO	QT REG.
Cadastro Família Comercial	ELT	HADOOP BDTC	FamiliaComercial	43
Cadastro Família Comercial	ELTL	HADOOP DWTC	dFamiliaComercial	43
Cadastro Família Comercial	ETL	SQL DWTC	stgFamiliaComercial	43
Cadastro Família Material	ELT	HADOOP BDTC	FamiliaMaterial	570
Cadastro Família Material	ELTL	HADOOP DWTC	dFamiliaMaterial	570
Cadastro Família Material	ETL	SQL DWTC	stgFamiliaMaterial	570
Cadastro Grupo Estoque	ELT	HADOOP BDTC	GrupoEstoque	33
Cadastro Grupo Estoque	ELTL	HADOOP DWTC	dGrupoEstoque	33
Cadastro Grupo Estoque	ETL	SQL DWTC	stgGrupoEstoque	33
Cadastro Produto	ELT	HADOOP BDTC	Produto	31842
Cadastro Produto	ELTL	HADOOP DWTC	dProduto	31842
Cadastro Produto	ETL	SQL DWTC	stgProduto	31842
Fato Compras	ELT	HADOOP BDTC	vwCompras	91378
Fato Compras	ELTL	HADOOP DWTC	fcompras	91378
Fato Compras	ETL	SQL DWTC	fCompras	91378
Fato Produção	ELT	HADOOP BDTC	vwproducao	32070
Fato Produção	ELTL	HADOOP DWTC	fProducao	32070
Fato Produção	ETL	SQL DWTC	fProducao	32070
Fato Vendas	ELT	HADOOP BDTC	fProducao	27880
Fato Vendas	ELTL	HADOOP DWTC	fProducao	27880
Fato Vendas	ETL	SQL DWTC	fProducao	27880
Movimentos Estoque	ELT	HADOOP BDTC	fProducao	17838280
Movimentos Estoque	ETL	SQL DWTC	fProducao	17838280

Fonte: HÜBNER (2020)

Ainda como comparação de integridade dos dados, é possível notar que a soma do campo **quantidade saldo** de cada entidade fato, em cada paradigma, possui precisamente o mesmo valor. Independentemente da dimensão escolhida para análise, o valor total é correspondente: na fato **compras**, 415.217.688,16; na fato **vendas** -391.401.173,00; na fato **produção** 551.349.473.56.

A Figura 25 apresenta a visão resultante do paradigma ETL, na perspectiva da dimensão **município**, oriunda da **filial**. A Figura 26 apresenta o resultado do paradigma ELT sob a perspectiva do **ano**. Por fim, a Figura 27 corresponde ao

resultado do paradigma ELTL na perspectiva da dimensão **grupo estoque**. Esta análise é realizada com a utilização do Microsoft PowerBI Desktop, instalado no mesmo ambiente de hardware supracitado. Trata-se de uma leitura integral das tabelas conforme descrito a seguir:

a) ETL: leitura do banco de dados SQL “DWTC” hospedado na VM DEV3H15, com carga integral das tabelas dFamiliaComercial, dFamiliaMaterial, dGrupoEstoque, dProduto, fCompras, fVendas e fProducao;
b) ELT: leitura do banco de dados Hive “BDTC” hospedado na VM CLOUDERA, com carga integral das tabelas familiacomercial, familiamaterial, grupoestoque, produto, e das exibições vwcopmras, vwproducao e vwvendas.
c) ELTL: leitura do banco de dados hive “DWTC”, hospedado na VM CLOUDERA, com carga integral das tabelas dFamiliaMaterial, dGrupoEstoque, dProduto, fCompras, fVendas e fProducao;

Figura 25 – Valores resultantes do paradigma ETL

Município	Saldo Compras	Saldo Vendas	Saldo Produção
⊞ Blumenau	13.301.368,95	-23.213.369,00	32.118.768,00
⊞ Campinas	51.859.422,93	-48.830.792,00	49.305.201,87
⊞ Caxias do Sul	215.775.182,01	-186.801.760,00	219.931.671,69
⊞ Contagem	0,00		
⊞ Lajeado	7.016.234,36	-6.747.609,00	3.860.339,00
⊞ Londrina	2.248.341,60	-1.002.991,00	
⊞ Macaé	125.017.138,31	-124.804.652,00	246.133.493,00
Total	**415.217.688,16**	**-391.401.173,00**	**551.349.473,56**

Fonte: HÜBNER (2020)

Figura 26 – Valores resultantes do paradigma ELT

Ano	Saldo Compras	Saldo Vendas	Saldo Produção
⊞ 2015	82.976.952,39	-70.150.933,00	101.889.038,59
⊞ 2016	75.043.787,70	-72.528.183,00	98.893.177,64
⊞ 2017	80.830.642,89	-81.560.252,00	113.710.408,33
⊞ 2018	85.548.718,48	-84.424.105,00	117.435.619,40
⊞ 2019	90.817.586,71	-82.737.700,00	119.421.229,60
Total	**415.217.688,16**	**-391.401.173,00**	**551.349.473,56**

Fonte: HÜBNER (2020)

Figura 27 – Valores resultantes do paradigma ELTL

Grupo Estoque	Saldo Compras	Saldo Vendas	Saldo Produção
⊞ Aditivos, pigmentos e tintas	260.088,14		15,00
⊞ Componentes p/ montagem	183.170.808,47		4.368.238,00
⊞ Embalagens	34.305.194,04		8.132,00
⊞ Genérico	20.083,81		175,00
⊞ Manutenção e Imobilizado	145.668,61		
⊞ Materiais de terceiros Benefic	132.824.236,48		
⊞ Materiais de terceiros Consig	9.921.444,57		3,00
⊞ Materiais Consignação	1.616.804,00		
⊞ Material aux. de produção	12.053.256,59		
⊞ Material de manutenção	0,00		
⊞ Material de uso e consumo	82.215,75		
⊞ Matéria-prima p/ ferramentaria	665.276,78		
⊞ Moldes em Comodato	478,00		
⊞ Prod. Revenda Desenvolvimentos	32,00		5,00
⊞ Produtos em elaboração	11.139.589,00		154.145.092,14
⊞ Produtos para revenda	56,00		
⊞ Produtos prontos	5.568.546,00	-391.400.804,00	392.117.205,89
⊞ Produtos prontos ferramentaria	30,00	-369,00	705,00
⊞ Produtos Próprio em elaboração			68.688,00
⊞ Produtos Próprio prontos	224,00		37.669,00
⊞ Resinas Moídas	617.780,71		159.330,97
⊞ Resinas Moídas Extrusadas	238.149,60		444.214,56
⊞ Resinas para injeção	22.568.749,72		
⊞ Sucatas	6.575,00		
⊞ Testes/Amostras C/VALOR CML	2.594,00		
⊞ Testes/Amostras S/VALOR COML	9.806,90		
Total	**415.217.688,16**	**-391.401.173,00**	**551.349.473,56**

Fonte: HÜBNER (2020)

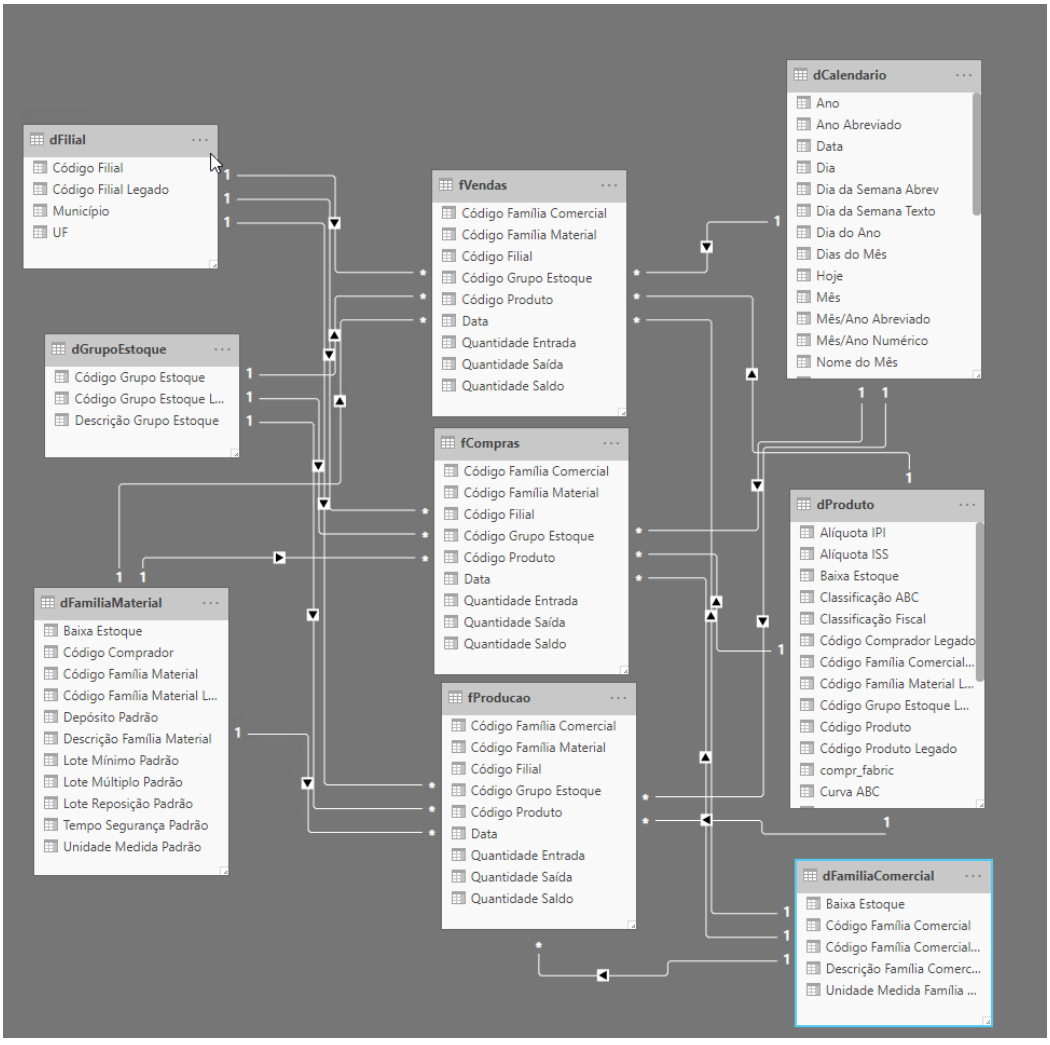
5.8.3 Avaliação qualitativa

As circunstâncias do presente desenvolvimento e as características do que fora proposto sugeriam, desde o princípio, o que se pode confirmar com as execuções: a qualidade do modelo resultante é idêntica no ETL e no ELTL. O ELT é o paradigma que resulta a pior avaliação neste quesito.

O resultado do paradigma ETL, que é o DW, aqui implementado em banco de dados relacional, seguindo o modelo conceitual em estrela, pode ser observado na Figura 28. Nela é possível notar a significativa simplificação dos nomes dos campos das entidades fato e dimensões que, por sua vez, induz não somente a execução dos relacionamentos como à facilidade na intepretação do conjunto de dados. O mesmo resultado, pode ser rigorosamente observado no paradigma ELTL, representado na mesma figura.

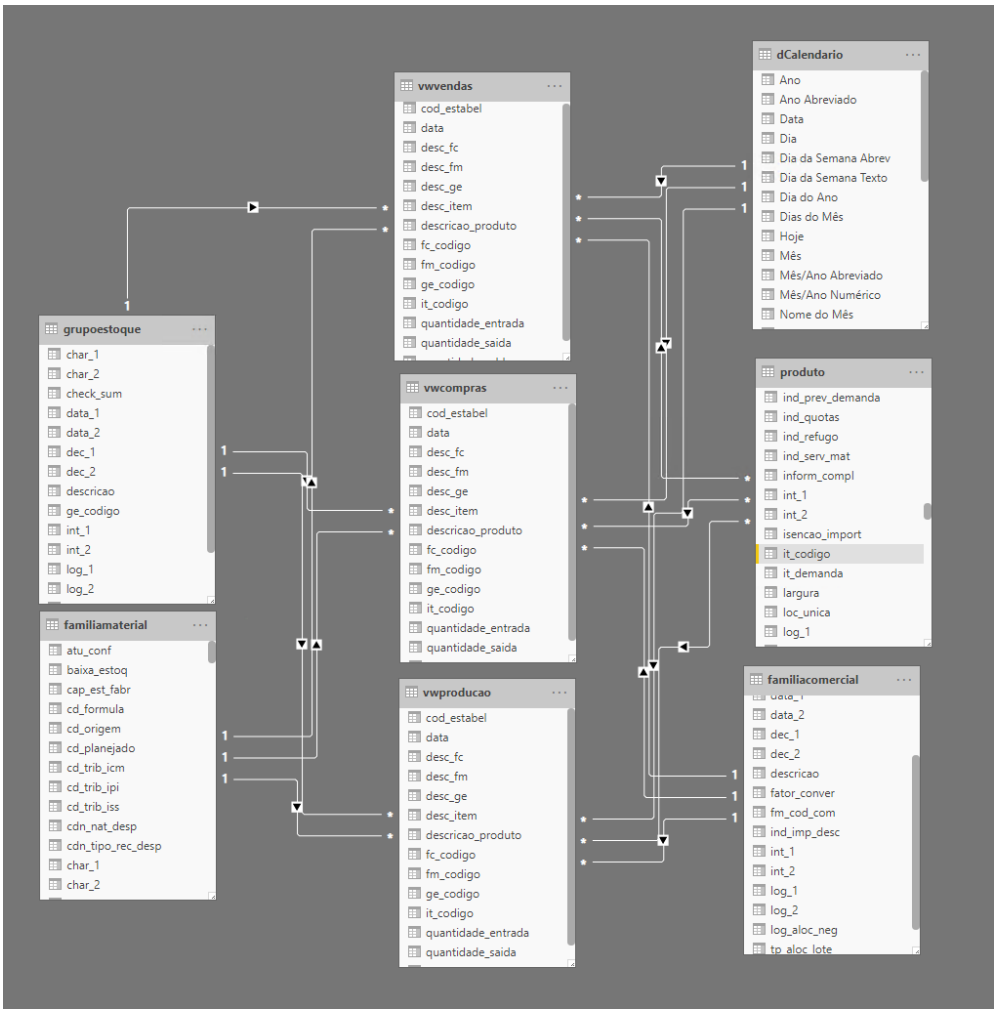
Na Figura 29 é possível observar os nomes de campos das tabelas (ou atributos das entidades), com relevante complexidade para interpretação. Ou seja, o ELT expõe as dificuldades da ausência do trabalho de elevação de verbosidade. Os nomes dos campos são mais complexos, o que torna o trabalho de análise menos assertivo, preciso e veloz – sobretudo para usuários das áreas de negócio, que são os verdadeiros consumidores da informação. Exemplo disto está na relação entre a tabela **familiacomercial** (onde o campo que corresponde à chave é o **fm_cod_com**), e as exibições (ou *views*), onde o campo correspondente é denominado **fc_codigo**. Destaque para a ausência da tabela de filiais, que não faz parte do conjunto de dados brutos.

Figura 28 – Conjunto de dados resultante do ETL e ELTL



Fonte: HÜBNER (2020)

Figura 29 – Conjunto de dados resultante do ELT



Fonte: HÜBNER (2020)

5.9 DIFICULDADES

As barreiras na implementação da proposta foram as mais diversas. A principal dificuldade, contudo, foi a curva de aprendizado relacionada ao contexto do Big Data e à ferramenta Hadoop. O conhecimento prévio no tema resumia-se aos conceitos e teorias. Assim, a descoberta dos primeiros passos demandou extensa pesquisa extraliterária, incluindo a aquisição e realização de cursos específicos através de plataformas populares como Udemy.

Do ponto de vista das simulações, a dificuldade mais expressiva foi o encontro da estabilidade do ambiente Hadoop. Sendo seu requisito de *hardware* extremamente elevado e a disponibilidade efetiva de hardware sensivelmente limitada, não foram incomuns as ocorrências de serviços abruptamente interrompidos e indisponíveis.

Ainda sobre o Hadoop, a impossibilidade da adoção de índices, embora seja conceitual e compreensível, em virtude do armazenamento no HDFS, contribuiu para a baixa performance dos modelos a ele relacionados.

Em termos de infraestrutura, para que a VM CLOUDERA se tornasse acessível no ambiente Windows disposto na VM DEV3H15, foi necessário criar uma entrada estática (*host* fixo), apontando o nome ‘QUICKSTART.CLOUDERA’ para o IP correspondente, no arquivo designado a esta função (%SYSTEMROOT%\System32\drivers\etc\hosts).

No processo de importação de tabelas para o sistema de arquivos Hadoop HDFS, observou-se relevante particularidade quanto à campos de textos grandes. A estes campos (como por exemplo a narrativa de um produto), é extremamente comum e regular a utilização de vírgulas, ponto e vírgulas, tabulações e quebras de linha, bem como quaisquer tipos de símbolos da tabela ASC. Com esta prática, os valores adotados para quebra de campos, implicaram resultados indesejados, tendo sido necessário remover tais campos do processo de importação.

A complexidade da implantação é algo subjetivo, pois depende da capacitação de cada profissional. Contudo, considerando-se para esta implantação que não havia conhecimento prático efetivo adquirido sobre BigData, certamente o ELT foi o paradigma de maior complexidade.

6 CONCLUSÃO

Apesar das considerações gerais quanto ao Big Data, é possível afirmar que o modelo proposto (ELTL) é viável e faz sentido. Por entregar performance geral superior ao ETL, e qualidade superior ao ELT, atende de forma satisfatória às expectativas. Certamente, seu amadurecimento com experimentações futuras – tanto em nível acadêmico como também no contexto organizacional (algo a que se propõe este trabalho, desde o seu princípio), será fundamental para sua confirmação e propagação.

Há que se considerar que, para as simulações propostas, notadamente não houvera caracterização dos 3VS fundamentais ao BigData. Ou seja, embora as tecnologias empregadas no contexto Hadoop sejam características de BDT, e, apesar de as literaturas serem divergentes (principalmente quando ao requisito de Volume para se caracterizar um BDT), o conjunto de dados utilizado neste desenvolvimento não pode ser tipificado puramente como BigData. Soma-se a isto o fato de que, para uma ideal execução de ambiente Hadoop, é essencial haver alta disponibilidade de Hardware – motivo pelo qual, no presente experimento, não fora adotado ambiente em *cluster*. Isto reforça suas premissas básicas como o processamento distribuído, através do qual poderá ser obtida eficiência no processamento de grandes volumes de dados com velocidades elevadas.

Não é possível afirmar que a escolha de ferramentas alternativas, teria ou não influência sobre os resultados obtidos. Mas é plausível pressupor que – se guardadas as características fundamentais de cada plataforma e observadas as etapas de processo descritas neste objeto – o resultado relativo seria similar. A futura realização de testes, com plataformas alternativas, será certamente esclarecedora.

Como valiosa lição, se faz necessário pontuar que a melhor experimentação possível – e, por conseguinte, os melhores resultados possíveis poderiam ter sido obtidos com a utilização de *hardware* dedicado especificamente a esta função. Os ambientes em nuvem avaliados se apresentaram monetariamente inacessíveis, o que inviabilizou este caminho.

Minha atividade profissional, em termos práticos e objetivos, sofre a partir deste momento, relevante agregação de valor. Considerando que a manipulação de bancos de dados relacionais para DW já faziam parte minha rotina, o conhecimento produzido para com relação à BigData no decorrer deste trabalho, já faz significativa

diferença na amplitude dos serviços oferecidos aos clientes da Row Up Inteligência de Dados.

São interessantes as perspectivas de trabalhos futuros a partir desta obra. Experimentações que abordem a “modelagem multidimensional com Big Data em ambientes distribuídos”, ou ainda assuntos como “interpretação dimensional em dados não estruturados”, “evolução do modelo multidimensional para estruturas corporativas”, “a utilização do Big Data nas áreas de negócio” são algumas das possibilidades de trabalhos futuros relacionados ao tema aqui apresentado.

É motivo de grande orgulho contribuir com o crescimento não somente do meu próprio conhecimento, como também da expertise acadêmica através deste trabalho, levando este objeto ao alcance da comunidade.

REFERÊNCIAS

AMARAL, F. **Big Data:** uma visão gerencial para executivos, consultores e gerentes de projetos. [S.l.]: Fernando Amaral, 2016. ASIN: B01MDSEP3V.

ANDERSON, C. The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. **Wired**, 2008. Disponivel em: <https://www.wired.com/2008/06/pb-theory/>. Acesso em: 31 out. 2019.

BAÚ, J. **Implicações do fenômeno Big Data na tomada de decisão baseada em dados em uma cooperativa de crédito**. Caxias do Sul: [s.n.], 2019. Dissertação (Mestrado em Administração) - Universidade Caxias do Sul.

CHIAVENATO, I. **Administração nos Novos Tempos**. 1a. ed. Rio de Janeiro: Campus, 1999.

CHIAVENATO, I. **Introdução à Teoria Geral da Administração**. 7a. ed. Rio de Janeiro: Elsevier, 2003. ISBN 85-352-1348-1.

FATIMA, N. ETL vs ELT: what's the difference? **Astera**, 2019. Disponivel em: <https://www.astera.com/type/blog/etl-vs-elt-whats-the-difference/>. Acesso em: 15 nov. 2019.

FEINBERG, D. Data Lake, Big Data, NoSQL – The Good, The Bad and The Ugly. **Gartner Blog Network**, 2017. Disponivel em: <https://blogs.gartner.com/donald-feinberg/2017/07/29/oh-terms-use-technology-need-new-hype/>. Acesso em: 02 nov. 2019.

FERREIRA, A. **AURÉLIO:** o dicionário da língua portuguesa. Curitiba - PR: Positivo, 2008. ISBN 978-85-385-0766-6.

GARCIA, A. **Cubos OLAP Best Practices**. [S.l.]: Armando Castanon Garcia, 2016. ASIN B01CO2M266.

GARTNER. **IT Glossary**. Disponivel em: <https://www.gartner.com/it-glossary/business-intelligence-bi/>. Acesso em: 24 set. 2019.

GRUBB, V. **Conflito de Gerações:** desafios e estratégias para gerenciar quatro gerações no ambiente de trabalho. [S.l.]: Autêntica Business, 2018. ASIN: B07DPQLNVC.

HEUSER, C. A. **Banco de dados relacional:** conceitos, SQL e administração. Porto Alegre: [s.n.], 2019. CDD 055.74 CDU 004.65.

INMON, W. **Como construir o data warehouse**. 2a. ed. Rio de Janeiro: Campus, 1997.

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit:** Guia completo para modelagem dimensional. 1a. ed. Rio de Janeiro: Campus, 2002.

LIMA, A. C. et al. **Tomada de Decisão nas Organizações**. 1. ed. [S.l.]: Saraiva, 2017. ASIN B076BY3FKQ.

LOH, S. **BI na era do big data para cientistas de dados:** Indo além de cubos e dashboards na busca pelos porquês, explicações e padrões. 1a. ed. Porto Alegre: [s. n.], 2014.

MARQUESONE, R. **Big Data:** Técnicas e tecnologias para extração de valor dos dados. [S.l.]: Casa do Código, 2016.

MARR, B. A brief history of big data everyone should read. **World Economic Forum**, 2015. Disponivel em: <https://www.weforum.org/agenda/2015/02/a-brief-history-of-big-data-everyone-should-read/>. Acesso em: 31 out. 2019.

MARSH, G. ELT Architecture in the Azure Cloud. **Aptitive**, 2017. Disponivel em: <https://blog.aptitive.com/elt-architecture-in-the-azure-cloud-50a90681036b>. Acesso em: 08 nov. 2019.

MATOS, D. Data Lake, a Fonte do Big Data. **Ciência e Dados**, 2018. Disponivel em: <http://www.cienciaedados.com/data-lake-a-fonte-do-big-data>. Acesso em: 21 out. 2019.

MATOS, D. NoSQL Database. **Ciência e Dados**, 2019. Disponivel em: <http://www.cienciaedados.com/nosql-database/>. Acesso em: 02 nov. 2019.

PARREIRAS, F.; MATHEUS, R. F. Inteligência Empresarial Versus Business Intelligence: Abordagens complementares para o apoio à tomada de decisão no Brasil. **KMBRASIL 2004 - Congresso Anual da Sociedade Brasileira de Gestão do Conhecimento**, Belo Horizonte, 2004.

QUADROS, C. M. et al. **Guia para elaboração de trabalhos acadêmicos**. 6a. ed. Caxias do Sul: SIBUCS, 2019.

RANJAN, V. A Comparative Study between ETL and ELT approach for loading data into a Data Warehose. **CSCI 693 Research Paper**, 17 dez. 2009.

RIBEIRO, R. Linkedin. **ETL vs ELT? Quando duas letras fazem muita diferença.**, 2018. Disponivel em: <https://www.linkedin.com/pulse/etl-vs-elt-quando-duas-letras-fazem-muita-diferença-rodrigo-r-g-/>. Acesso em: 24 out. 2019.

SADALAGE, P.; FOWLER, M. **NoSQL Essencial:** Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec Editora, 2019.

SPERLEY, E. **Enterprise data warehouse:** planning, building, and implementatio. Upper Saddle River: Prentice Hall PTR, 1999.

SULTAN, S. **Business Intelligence:** Harness BI tools for the decision-making and business forecasting. [S.l.]: Expert of Course, 2017. ASIN B076YS5P86.

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

TOTH, K. **Database Pro**. [S.l.]: Amazon, 2013. B00BDS2FFQ.

TURBAN, E. et al. **Business Intelligence:** um enfoque gerencial para a inteligência do negócio. Porto Alegre: Bookman, 2009. ISBN 9788577804252.

WANG, C. The Promise of ELT Over ETL: Analysis, Not Paralysis. **Fivetran**. Disponivel em: <https://fivetran.com/blog/elt_vs_etl>. Acesso em: 02 nov. 2019.

WAZLAWICK, R. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier, 2008. ISBN 978-85-352-3522-7.

WHAT is the definition of OLAP. **OLAP.com**. Disponivel em: <https://olap.com/olap-definition/>. Acesso em: 07 nov. 2019.

WILLIAMS, C. **Princípios de Administração**. 2a. ed. São Paulo: Cengage, 2017. ISBN 9788522126958.

APÊNDICE A – SCRIPTS DE CRIAÇÃO DAS ESTRUTURAS DO BANCO DE DADOS DBTC2

```
CREATE TABLE [dbo].[FamiliaComercial](
	[fm-cod-com] [varchar](16) NULL,
	[descricao] [varchar](60) NULL,
	[baixa-estoq] [int] NULL,
	[ind-imp-desc] [int] NULL,
	[un] [varchar](4) NULL,
	[char-1] [varchar](200) NULL,
	[char-2] [varchar](200) NULL,
	[dec-1] [numeric](23, 8) NULL,
	[dec-2] [numeric](23, 8) NULL,
	[int-1] [int] NULL,
	[int-2] [int] NULL,
	[log-1] [bit] NULL,
	[log-2] [bit] NULL,
	[data-1] [date] NULL,
	[data-2] [date] NULL,
	[tp-aloc-lote] [int] NULL,
	[fator-conver] [numeric](20, 5) NULL,
	[check-sum] [varchar](40) NULL,
	[log-aloc-neg] [bit] NULL)

CREATE TABLE [dbo].[FamiliaMaterial](
	[sit-aloc] [int] NULL,
	[vl-min-inv] [numeric](19, 4) NULL,
	[vl-max-inv] [numeric](19, 4) NULL,
	[var-transf] [numeric](17, 2) NULL,
	[var-req-menor] [numeric](17, 2) NULL,
	[var-req-maior] [numeric](17, 2) NULL,
	[var-rep] [numeric](17, 2) NULL,
	[var-quantid] [int] NULL,
	[var-qtd-perm] [numeric](17, 2) NULL,
	[var-preco] [int] NULL,
	[var-pre-perm] [numeric](17, 2) NULL,
	[var-mob-menor] [numeric](17, 2) NULL,
	[var-mob-maior] [numeric](17, 2) NULL,
	[un] [varchar](4) NULL,
	[tp-cons-prev] [numeric](15, 0) NULL,
	[tp-adm-lote] [int] NULL,
	[tipo-sched] [int] NULL,
	[tipo-requis] [int] NULL,
	[tipo-lote-ec] [int] NULL,
	[tipo-insp] [int] NULL,
	[tipo-est-seg] [int] NULL,
	[tipo-contr] [int] NULL,
	[tipo-con-est] [int] NULL,
	[tempo-segur] [int] NULL,
	[sazonalidade] [varchar](144) NULL,
	[ressup-fabri] [int] NULL,
	[res-int-comp] [int] NULL,
	[res-for-comp] [int] NULL,
	[res-cq-fabri] [int] NULL,
	[res-cq-comp] [int] NULL,
	[reporte-mob] [int] NULL,
	[rep-prod] [int] NULL,
	[quant-segur] [numeric](19, 4) NULL,
	[quant-perda] [numeric](19, 4) NULL,
	[qt-max-ordem] [numeric](19, 4) NULL,
	[prioridade] [int] NULL,
	[politica] [int] NULL,
	[perm-saldo-neg] [int] NULL,
	[periodo-fixo] [int] NULL,
	[perc-nqa] [numeric](17, 2) NULL,
	[per-rest-icms] [numeric](17, 2) NULL,
	[nr-linha] [int] NULL,
	[nivel] [int] NULL,
	[niv-rest-icms] [varchar](2) NULL,
	[moeda-padrao] [int] NULL,
	[lote-multipl] [numeric](19, 4) NULL,
	[lote-minimo] [numeric](19, 4) NULL,
	[lote-economi] [numeric](19, 4) NULL,
	[loc-unica] [bit] NULL,
	[invent-cla-c] [bit] NULL,
	[invent-cla-b] [bit] NULL,
	[invent-cla-a] [bit] NULL,
	[ind-inf-qtf] [bit] NULL,
	[ind-imp-desc] [int] NULL,
	[ind-cons-prv] [numeric](17, 2) NULL,
	[horiz-fixo] [int] NULL,
	[fraciona] [bit] NULL,
	[fm-codigo] [varchar](16) NULL,
	[fator-refugo] [numeric](17, 2) NULL,
	[fator-ponder] [varchar](48) NULL,
	[fator-conver] [numeric](20, 5) NULL,
	[emissao-ord] [int] NULL,
	[div-ordem] [int] NULL,
	[descricao] [varchar](60) NULL,
	[deposito-pad] [varchar](6) NULL,
	[demanda] [int] NULL,
	[curva-abc] [bit] NULL,
	[criticidade] [int] NULL,
	[contr-qualid] [bit] NULL,
	[codigo-orig] [int] NULL,
	[cod-servico] [int] NULL,
	[cod-comprado] [varchar](24) NULL,
	[classe-repro] [int] NULL,
	[class-fiscal] [varchar](20) NULL,
	[ciclo-contag] [int] NULL,
	[cd-trib-iss] [int] NULL,
	[cd-trib-ipi] [int] NULL,
	[cd-trib-icm] [int] NULL,
	[cd-planejado] [varchar](24) NULL,
	[cd-origem] [int] NULL,
	[cd-formula] [int] NULL,
	[cap-est-fabr] [numeric](19, 4) NULL,
	[baixa-estoq] [bit] NULL,
	[atu-conf] [bit] NULL,
	[cod-localiz] [varchar](20) NULL,
	[cod-lista-destino] [varchar](16) NULL,
	[log-atualiz-via-mmp] [bit] NULL,
	[cod-taxa] [int] NULL,
	[char-1] [varchar](400) NULL,
	[char-2] [varchar](400) NULL,
	[dec-1] [numeric](23, 8) NULL,
	[dec-2] [numeric](23, 8) NULL,
	[int-1] [int] NULL,
	[int-2] [int] NULL,
	[log-1] [bit] NULL,
	[log-2] [bit] NULL,
	[data-1] [date] NULL,
	[data-2] [date] NULL,
	[conv-tempo-seg] [bit] NULL,
	[nivel-apr-requis] [int] NULL,
	[nivel-apr-solic] [int] NULL,
	[nivel-apr-manut] [int] NULL,
	[nivel-apr-compra] [int] NULL,
	[ind-prev-demanda] [int] NULL,
	[ind-calc-meta] [int] NULL,
	[log-utiliza-batch-padrao] [bit] NULL,
	[qtd-batch-padrao] [numeric](19, 4) NULL,
	[check-sum] [varchar](40) NULL,
	[reporte-ggf] [int] NULL,
	[ind-refugo] [int] NULL,
	[ind-lista-mrp] [int] NULL,
	[num-id-familia] [int] NULL,
	[cod-classe-frete] [varchar](10) NULL,
	[pto-repos] [numeric](19, 4) NULL,
	[cod-reabast] [varchar](20) NULL,
	[Qtd-min-reabast] [numeric](17, 2) NULL,
	[Cod-var-tempo-res] [varchar](8) NULL,
	[Qtd-var-ressup] [numeric](17, 2) NULL,
	[cod-grupo-compra-p] [varchar](24) NULL,
	[cod-grupo-compra-o] [varchar](24) NULL,
	[cdn-tipo-rec-desp] [int] NULL,
	[cdn-nat-desp] [int] NULL,
	[politica-aps] [int] NULL,
	[geracao-ordem] [int] NULL,
	[cons-produto] [bit] NULL,
	[cons-saldo] [bit] NULL,
	[mp-restrit] [bit] NULL,
	[qtde-max] [numeric](19, 4) NULL,
	[qtde-fixa] [numeric](19, 4) NULL,
	[lote-repos] [numeric](17, 2) NULL,
	[cons-consumo] [bit] NULL,
	[cod-malha] [varchar](10) NULL,
	[cod-pulmao] [varchar](10) NULL,
	[tipo-formula] [int] NULL,
	[per-ppm] [numeric](19, 4) NULL,
	[log-control-estoq-refugo] [bit] NULL,
	[log-refugo-preco-fisc] [bit] NULL,
	[cod-item-refugo] [varchar](32) NULL,
	[val-relac-refugo-item] [numeric](20, 5) NULL,
	[log-multi-malha] [bit] NULL,
	[idi-classif-item] [int] NULL,
	[log-orig-ext] [bit] NULL,
	[log-altera-valid-lote] [bit] NULL,
	[log-inspec-lote] [bit] NULL,
	[dsl-formul-quant-cq] [varchar](500) NULL,
	[log-rotei-pend] [bit] NULL
)

CREATE TABLE [dbo].[GrupoEstoque](
	[tipo-ge] [int] NULL,
	[ge-codigo] [int] NULL,
	[descricao] [varchar](60) NULL,
	[char-1] [varchar](200) NULL,
	[char-2] [varchar](200) NULL,
	[dec-1] [numeric](23, 8) NULL,
	[dec-2] [numeric](23, 8) NULL,
	[int-1] [int] NULL,
	[int-2] [int] NULL,
	[log-1] [bit] NULL,
	[log-2] [bit] NULL,
	[data-1] [date] NULL,
	[data-2] [date] NULL,
	[check-sum] [varchar](40) NULL,
	[num-id-grup-estoque] [int] NULL,
	[log-orig-ext] [bit] NULL
)
```

```
CREATE TABLE [dbo].[Produto](
	[incentivado] [bit] NULL,
	[fator-reaj-icms] [numeric](23, 8) NULL,
	[tipo-atp] [int] NULL,
	[alt-refer] [bit] NULL,
	[sit-aloc] [int] NULL,
	[rot-refer] [bit] NULL,
	[rot-revis] [bit] NULL,
	[rot-quant] [bit] NULL,
	[capac-recip-beb] [int] NULL,
	[tipo-recip-beb] [int] NULL,
	[enquad-beb] [int] NULL,
	[esp-beb] [int] NULL,
	[valor-ipi-beb] [numeric](20, 5) NULL,
	[volume] [numeric](20, 5) NULL,
	[vl-mob-ant] [numeric](19, 4) NULL,
	[vl-mat-ant] [numeric](19, 4) NULL,
	[variac-acum] [numeric](19, 4) NULL,
	[var-transf] [numeric](17, 2) NULL,
	[var-req-menor] [numeric](17, 2) NULL,
	[var-req-maior] [numeric](17, 2) NULL,
	[var-rep] [numeric](17, 2) NULL,
	[var-mob-menor] [numeric](17, 2) NULL,
	[var-mob-maior] [numeric](17, 2) NULL,
	[cod-imagem] [varchar](60) NULL,
	[char-1] [varchar](400) NULL,
	[usuario-obsol] [varchar](24) NULL,
	[usuario-alt] [varchar](24) NULL,
	[un] [varchar](4) NULL,
	[curva-abc] [bit] NULL,
	[cod-tax] [int] NULL,
	[cod-tax-serv] [int] NULL,
	[desc-item] [varchar](120) NULL,
	[fase-medio] [int] NULL,
	[tx-importacao] [numeric](18, 3) NULL,
	[tp-desp-padrao] [int] NULL,
	[tp-cons-prev] [numeric](15, 0) NULL,
	[tp-aloc-lote] [int] NULL,
	[tp-adm-lote] [int] NULL,
	[tipo-sched] [int] NULL,
	[tipo-requis] [int] NULL,
	[tipo-lote-ec] [int] NULL,
	[tipo-insp] [int] NULL,
	[tipo-est-seg] [int] NULL,
	[tipo-desc-nt] [int] NULL,
	[tipo-contr] [int] NULL,
	[tipo-con-est] [int] NULL,
	[tempo-segur] [int] NULL,
	[sc-codigo] [varchar](40) NULL,
	[resumo-mp] [int] NULL,
	[ressup-fabri] [int] NULL,
	[responsavel] [varchar](24) NULL,
	[res-int-comp] [int] NULL,
	[res-for-comp] [int] NULL,
	[res-cq-fabri] [int] NULL,
	[res-cq-comp] [int] NULL,
	[reporte-mob] [int] NULL,
	[rep-prod] [int] NULL,
	[rendimento] [numeric](19, 4) NULL,
	[quant-segur] [numeric](19, 4) NULL,
	[quant-perda] [numeric](19, 4) NULL,
	[qt-max-ordem] [numeric](19, 4) NULL,
	[prioridade] [int] NULL,
	[preco-ul-ent] [numeric](19, 4) NULL,
	[preco-repos] [numeric](19, 4) NULL,
	[preco-fiscal] [numeric](19, 4) NULL,
	[preco-base] [numeric](19, 4) NULL,
	[pr-sem-tx] [numeric](19, 4) NULL,
	[politica] [int] NULL,
	[pm-ja-calc] [bit] NULL,
	[peso-liquido] [numeric](20, 5) NULL,
	[peso-bruto] [numeric](20, 5) NULL,
	[perm-saldo-neg] [int] NULL,
	[periodo-fixo] [int] NULL,
	[perc-nqa] [numeric](18, 3) NULL,
	[perc-demanda] [numeric](17, 2) NULL,
	[per-rest-icms] [numeric](17, 2) NULL,
	[per-rest-fora] [numeric](17, 2) NULL,
	[per-min-luc] [numeric](17, 2) NULL,
	[path] [varchar](80) NULL,
	[nr-linha] [int] NULL,
	[nr-item-dcr] [int] NULL,
	[nivel] [int] NULL,
	[niv-rest-icms] [varchar](2) NULL,
	[niv-rest-fora] [varchar](2) NULL,
	[niv-mps] [int] NULL,
	[niv-mais-bai] [int] NULL,
	[nat-despesa] [int] NULL,
	[moeda-padrao] [int] NULL,
	[lote-mulven] [numeric](19, 4) NULL,
	[lote-multipl] [numeric](19, 4) NULL,
	[lote-minimo] [numeric](19, 4) NULL,
	[lote-economi] [numeric](19, 4) NULL,
	[narrativa] [varchar](4000) NULL,
	[cod-localiz] [varchar](20) NULL,
	[loc-unica] [bit] NULL,
	[largura] [numeric](23, 8) NULL,
	[it-demanda] [varchar](32) NULL,
	[it-codigo] [varchar](32) NULL,
	[isencao-import] [int] NULL,
	[inform-compl] [varchar](32) NULL,
	[ind-serv-mat] [int] NULL,
	[ind-item-fat] [bit] NULL,
	[ind-ipi-dife] [bit] NULL,
	[ind-inf-qtf] [bit] NULL,
	[ind-imp-desc] [int] NULL,
	[ind-especifico] [bit] NULL,
	[ind-backorder] [bit] NULL,
	[id-grade] [bit] NULL,
	[horiz-fixo] [int] NULL,
	[ge-codigo] [int] NULL,
	[ft-conversao] [numeric](15, 0) NULL,
	[ft-conv-fmcoml] [numeric](15, 0) NULL,
	[fraciona] [bit] NULL,
	[fm-codigo] [varchar](16) NULL,
	[fm-cod-com] [varchar](16) NULL,
	[fator-refugo] [numeric](17, 2) NULL,
	[fator-conver] [numeric](20, 5) NULL,
	[emissao-ord] [int] NULL,
	[dt-ult-ben] [date] NULL,
	[dt-pr-fisc] [date] NULL,
	[div-ordem] [int] NULL,
	[descricao-2] [varchar](36) NULL,
	[descricao-1] [varchar](36) NULL,
	[desc-nacional] [varchar](72) NULL,
	[desc-inter] [varchar](72) NULL,
	[deposito-pad] [varchar](6) NULL,
	[demanda] [int] NULL,
	[dec-ftcon] [int] NULL,
	[dec-conv-fmcoml] [int] NULL,
	[de-codigo-prin] [varchar](32) NULL,
	[data-ult-sai] [date] NULL,
	[data-ult-rep] [date] NULL,
	[data-ult-ent] [date] NULL,
	[data-ult-con] [date] NULL,
	[data-obsol] [date] NULL,
	[data-liberac] [date] NULL,
	[data-implant] [date] NULL,
	[data-base] [date] NULL,
	[ct-codigo] [varchar](40) NULL,
	[criticidade] [int] NULL,
	[contr-qualid] [bit] NULL,
	[contr-plan] [int] NULL,
	[consumo-prev] [numeric](19, 4) NULL,
	[consumo-aad] [numeric](19, 4) NULL,
	[concentracao] [numeric](19, 4) NULL,
	[comprim] [numeric](23, 8) NULL,
	[compr-fabric] [int] NULL,
	[codigo-refer] [varchar](40) NULL,
	[codigo-orig] [int] NULL,
	[cod-servico] [int] NULL,
	[cod-refer] [varchar](16) NULL,
	[cod-produto] [varchar](32) NULL,
	[cod-obsoleto] [int] NULL,
	[cod-estabel] [varchar](10) NULL,
	[cod-comprado] [varchar](24) NULL,
	[cod-auxiliar] [varchar](80) NULL,
	[classif-abc] [int] NULL,
	[classe-repro] [int] NULL,
	[class-fiscal] [varchar](20) NULL,
	[ciclo-contag] [int] NULL,
	[cd-trib-iss] [int] NULL,
	[cd-trib-ipi] [int] NULL,
	[cd-trib-icm] [int] NULL,
	[cd-planejado] [varchar](24) NULL,
	[cd-origem] [int] NULL,
	[cd-formula] [int] NULL,
	[cd-folh-lote] [varchar](16) NULL,
	[cd-folh-item] [varchar](16) NULL,
	[cap-est-fabr] [numeric](19, 4) NULL,
	[calc-lead-time] [int] NULL,
	[calc-cons-prev] [int] NULL,
	[baixa-estoq] [bit] NULL,
	[atu-conf] [bit] NULL,
	[altura] [numeric](23, 8) NULL,
	[aliquota-ISS] [numeric](17, 2) NULL,
	[aliquota-ipi] [numeric](17, 2) NULL,
	[char-2] [varchar](400) NULL,
	[cd-referencia] [varchar](8) NULL,
	[dec-1] [numeric](23, 8) NULL,
	[dec-2] [numeric](23, 8) NULL,
	[INT-1] [int] NULL,
	[int-2] [int] NULL,
	[prefixo-lote] [varchar](10) NULL,
	[Nr-ult-peca] [int] NULL,
	[tp-lote-minimo] [bit] NULL,
	[tp-lote-multiplo] [bit] NULL,
	[tp-lote-econom] [bit] NULL,
	[quant-pacote] [numeric](19, 4) NULL,
	[conta-aplicacao] [varchar](34) NULL,
	[cd-tag] [varchar](32) NULL,
	[log-carac-tec] [bit] NULL,
	[cod-lista-destino] [varchar](16) NULL,
	[log-atualiz-via-mmp] [bit] NULL,
	[log-1] [bit] NULL,
	[vl-var-max] [numeric](17, 2) NULL,
	[vl-var-min] [numeric](17, 2) NULL,
	[qt-var-max] [numeric](17, 2) NULL,
	[qt-var-min] [numeric](17, 2) NULL,
	[nivel-apr-requis] [int] NULL,
	[reporte-ggf] [int] NULL,
	[log-2] [bit] NULL,
	[data-1] [date] NULL,
	[data-2] [date] NULL,
	[conv-tempo-seg] [bit] NULL,
	[ind-confprodcom] [bit] NULL,
```

```
[nivel-apr-solic] [int] NULL,
[nivel-apr-manut] [int] NULL,
[nivel-apr-compra] [int] NULL,
[ind-prev-demanda] [int] NULL,
[ind-calc-meta] [int] NULL,
[val-fator-custo-dis] [numeric](17, 2) NULL,
[qtd-refer-custo-dis] [numeric](17, 2) NULL,
[qtd-batch-padrao] [numeric](19, 4) NULL,
[log-utiliza-batch-padrao] [bit] NULL,
[ind-quotas] [bit] NULL,
[nr-pontos-quotas] [int] NULL,
[check-sum] [varchar](40) NULL,
[num-id-item] [int] NULL,
[ind-refugo] [int] NULL,
[log-necessita-li] [bit] NULL,
[dias-estoq-aloc] [varchar](16) NULL,
[pto-repos] [numeric](19, 4) NULL,
[aliquota-ii] [numeric](17, 2) NULL,
[cod-trib-ii] [int] NULL,
[geracao-ordem] [int] NULL,
[cons-produto] [bit] NULL,
[cons-saldo] [bit] NULL,
[mp-restrit] [bit] NULL,
[qtde-max] [numeric](19, 4) NULL,
[qtde-fixa] [numeric](19, 4) NULL,
[lote-repos] [numeric](17, 2) NULL,
[cons-consumo] [bit] NULL,
[cod-malha] [varchar](10) NULL,
[cod-pulmao] [varchar](10) NULL,
[politica-aps] [int] NULL,
[tipo-formula] [int] NULL,
[per-ppm] [numeric](19, 4) NULL,
[cod-tab-preco-aps] [varchar](16) NULL,
[cod-pulmao-proces] [varchar](16) NULL,
[log-tax-produc] [bit] NULL,
[log-control-estoq-refugo] [bit] NULL,
[log-refugo-preco-fisc] [bit] NULL,
[cod-item-refugo] [varchar](32) NULL,
[val-relac-refugo-item] [numeric](20, 5) NULL,
[log-multi-malha] [bit] NULL,
[cod-destaq] [int] NULL,
[dsl-destaq] [varchar](4000) NULL,
[idi-classif-item] [int] NULL,
[cod-unid-negoc] [varchar](6) NULL,
[log-consid-aloc-ativid] [bit] NULL,
[val-overlap] [numeric](17, 2) NULL,
[log-programac-sfc] [bit] NULL,
[cod-dcr-item] [varchar](24) NULL,
[log-orig-ext] [bit] NULL,
[log-altera-valid-lote] [bit] NULL,
[log-inspec-lote] [bit] NULL,
[cod-workflow] [varchar](200) NULL
)
```

APÊNDICE B – SCRIPTS DE CRIAÇÃO DAS VIEWS DO BANCO DBTC2

```
CREATE VIEW [dbo].[vwMovimentoEstoque]
AS

SELECT
 [base-calculo] AS [base_calculo]
,[cod-barras] AS [cod_barras]
,[cod-depos] AS [cod_depos]
,[cod-emitente] AS [cod_emitente]
,[cod-estabel] AS [cod_estabel]
,[cod-estabel-des] AS [cod_estabel_des]
,[cod-localiz] AS [cod_localiz]
,[cod-lote-fabrican] AS [cod_lote_fabrican]
,[cod-orig-trans] AS [cod_orig_trans]
,[cod-prog-orig] AS [cod_prog_orig]
,[cod-refer] AS [cod_refer]
,[cod-roteiro] AS [cod_roteiro]
,[cod-unid-negoc] AS [cod_unid_negoc]
,[cod-unid-negoc-sdo] AS [cod_unid_negoc_sdo]
,[cod-usu-ult-alter] AS [cod_usu_ult_alter]
,[conta-contabil] AS [conta_contabil]
,[conta-saldo] AS [conta_saldo]
,[contabilizado] AS [contabilizado]
,[ct-codigo] AS [ct_codigo]
,[ct-saldo] AS [ct_saldo]
,[dat-fabricc-lote] AS [dat_fabricc_lote]
,[dat-valid-lote-fabrican] AS [dat_valid_lote_fabrican]
,[denominacao] AS [denominacao]
,[descricao-db-ok] AS [descricao_db_ok]
,[Descrição Produto] AS [descricao_produto]
,[dt-contab] AS [dt_contab]
,[dt-criacao] AS [dt_criacao]
,[dt-nf-saida] AS [dt_nf_saida]
,[dt-trans] AS [dt_trans]
,[esp-abrev] AS [esp_abrev]
,[esp-docto] AS [esp_docto]
,[especie-docto] AS [especie_docto]
,[hr-contab] AS [hr_contab]
,[hr-trans] AS [hr_trans]
,[identific emit] AS [identific_emit]
,[it-codigo] AS [it_codigo]
,[item-pai] AS [item_pai]
,[lote] AS [lote]
,[nat-operacao] AS [nat_operacao]
,[natureza_emitente] AS [natureza_emitente]
,[nom-fabrican] AS [nom_fabrican]
,[nome-abrev] AS [nome_abrev]
,[nr-ord-produ] AS [nr_ord_produ]
,[nr-ord-refer] AS [nr_ord_refer]
,[nr-reporte] AS [nr_reporte]
,[nr-req-sum] AS [nr_req_sum]
,[nr-trans] AS [nr_trans]
,[nr-trans-deb] AS [nr_trans_deb]
,[nro-docto] AS [nro_docto]
,[num-ord-des] AS [num_ord_des]
,[num-ord-inv] AS [num_ord_inv]
,[num-seq-des] AS [num_seq_des]
,[num-sequen] AS [num_sequen]
,[numero-ordem] AS [numero_ordem]
,[op-codigo] AS [op_codigo]
,[op-seq] AS [op_seq]
,[origem-valor] AS [origem_valor]
,[per-ppm] AS [per_ppm]
,[peso-liquido] AS [peso_liquido]
,[quantidade] AS [quantidade]
,[refer-contab] AS [refer_contab]
,[referencia] AS [referencia]
,[saldo-req] AS [saldo_req]
,[sc-codigo] AS [sc_codigo]
,[sc-saldo] AS [sc_saldo]
,[sequen-nf] AS [sequen_nf]
,[serie-docto] AS [serie_docto]
,[tipo-preco1] AS [tipo_preco1]
,[tipo-preco2] AS [tipo_preco2]
,[tipo-preco3] AS [tipo_preco3]
,[tipo-trans] AS [tipo_trans]
,[tipo-valor] AS [tipo_valor]
,[un] AS [un]
,[usuario] AS [usuario]
,[val-cofins] AS [val_cofins]
,[val-cofins-cmi] AS [val_cofins_cmi]
,[val-cofins-fasb] AS [val_cofins_fasb]
,[valor-ggf-m1] AS [valor_ggf_m1]
,[valor-ggf-m2] AS [valor_ggf_m2]
,[valor-ggf-m3] AS [valor_ggf_m3]
,[valor-ggf-o1] AS [valor_ggf_o1]
,[valor-ggf-o2] AS [valor_ggf_o2]
,[valor-ggf-o3] AS [valor_ggf_o3]
,[valor-ggf-p1] AS [valor_ggf_p1]
,[valor-ggf-p2] AS [valor_ggf_p2]
,[valor-ggf-p3] AS [valor_ggf_p3]
,[valor-icm] AS [valor_icm]
,[valor-ipi] AS [valor_ipi]
,[valor-iss] AS [valor_iss]
,[valor-mat-m1] AS [valor_mat_m1]
,[valor-mat-m2] AS [valor_mat_m2]
,[valor-mat-m3] AS [valor_mat_m3]
,[valor-mat-o1] AS [valor_mat_o1]
,[valor-mat-o2] AS [valor_mat_o2]
,[valor-mat-o3] AS [valor_mat_o3]
,[valor-mat-p1] AS [valor_mat_p1]
,[valor-mat-p2] AS [valor_mat_p2]
,[valor-mat-p3] AS [valor_mat_p3]
,[valor-mob-m1] AS [valor_mob_m1]
,[valor-mob-m2] AS [valor_mob_m2]
,[valor-mob-m3] AS [valor_mob_m3]
,[valor-mob-o1] AS [valor_mob_o1]
,[valor-mob-o2] AS [valor_mob_o2]
,[valor-mob-o3] AS [valor_mob_o3]
,[valor-mob-p1] AS [valor_mob_p1]
,[valor-mob-p2] AS [valor_mob_p2]
,[valor-mob-p3] AS [valor_mob_p3]
,[valor-nota] AS [valor_nota]
,[valor-pis] AS [valor_pis]
,[vl-icm-fasb1] AS [vl_icm_fasb1]
,[vl-icm-fasb2] AS [vl_icm_fasb2]
,[vl-ipi-fasb1] AS [vl_ipi_fasb1]
,[vl-ipi-fasb2] AS [vl_ipi_fasb2]
,[vl-iss-fasb1] AS [vl_iss_fasb1]
,[vl-iss-fasb2] AS [vl_iss_fasb2]
,[vl-nota-fasb1] AS [vl_nota_fasb1]
,[vl-nota-fasb2] AS [vl_nota_fasb2]
,[vl-pis-cmi] AS [vl_pis_cmi]
,[vl-pis-fasb] AS [vl_pis_fasb]
,[vl-taxa] AS [vl_taxa]


  FROM [DBTC2].[dbo].[MovimentoEstoque] m
  --where m.[dt-trans] >= '2010-01-01' and m.[dt-trans] <=
'2010-01-05'
```

APÊNDICE C – SCRIPTS DE SELEÇÃO E TRANSFORMAÇÃO DAS DIMENSÕES DO DATAWAREHOUSE (ETL)

```
------------------------------------------------------ DIMENSÕES CARREGADAS ------------------------------------------------
/*DIMENSÃO PRODUTO*/
SELECT
      [aliquota_ipi]                  AS [Alíquota IPI]
     ,[aliquota_ISS]                  AS [Alíquota ISS]
     ,IIF([baixa_estoq] = 1, 'SIM', 'NÃO') AS [Baixa Estoque]
     ,[class_fiscal]                  AS [Classificação Fiscal]
     ,[classif_abc]                   AS [Classificação ABC]
     ,[cod_comprado]                  AS [Código Comprador Legado]
     , CASE [cod_obsoleto]
                WHEN 1 THEN 'Ativo'
                WHEN 2 THEN 'Obsoleto Ordens Automáticas'
                WHEN 3 THEN 'Obsoleto Todas as Ordens'
                WHEN 4 THEN 'Totalmente Obsoleto'
           END     AS [Situação]
     ,[compr_fabric]  --Buscar tradução
     ,[curva_abc]                     AS [Curva ABC]
     ,[data_implant]                  AS [Data Implantação]
     ,[data_liberac]                  AS [Data Liberação]
     ,[data_obsol]                    AS [Data Obsolescência]
     ,[deposito_pad]                  AS [Depósito Padrão]
     ,[desc_item]                     AS [Descrição]
     ,[fm_cod_com]                    AS [Código Família Comercial Legado]
     ,[fm_codigo]                     AS [Código Família Material Legado]
     ,[ge_codigo]                     AS [Código Grupo Estoque Legado]
     ,[it_codigo]                     AS [Código Produto Legado]
     ,[lote_minimo]                   AS [Lote Mínimo]
     ,[lote_multipl]                  AS [Lote Múltiplo]
     ,[un]                                          AS [Unidade Medida]
     ,[usuario_alt]                   AS [Usuário Alteração]
  FROM [dbo].[stgProduto]


  /*DIMENSÃO GRUPO ESTOQUE*/
  SELECT
     [descricao]             as [Descrição Grupo Estoque]
    ,[ge_codigo]             as [Código Grupo Estoque Legado]
  FROM [DWTC].[dbo].[stgGrupoEstoque]


  /*DIMENSÃO FAMÍLIA COMERCIAL*/
  SELECT
            [descricao]               AS [Descrição Família Comercial]
     ,[un]                                      AS [Unidade Medida Família Comercial]
     ,iif([baixa_estoq]=1,'SIM','NÃO') AS [Baixa Estoque]
     ,[fm_cod_com]          AS [Código Família Comercial Legado]
  FROM [DWTC].[dbo].[stgFamiliaComercial]


  /*DIMENSÃO FAMÍLIA MATERIAL*/
  SELECT
       [baixa_estoq]                  as [Baixa Estoque]
     ,[cod_comprado]                  as [Código Comprador]
     ,[deposito_pad]                  as [Depósito Padrão]
     ,[descricao]                     as [Descrição Família Material]
     ,[fm_codigo]                     as [Código Família Material Legado]
     ,[lote_minimo]                   AS [Lote Mínimo Padrão]
     ,[lote_multipl]                  AS [Lote Múltiplo Padrão]
     ,[lote_repos]                    AS [Lote Reposição Padrão]
     ,[tempo_segur]                   AS [Tempo Segurança Padrão]
     ,[un]                                                AS [Unidade Medida Padrão]
  FROM [DWTC].[dbo].[stgFamiliaMaterial]


------------------------------------------------------ DIMENSÕES EXTRAÍDAS ------------------------------------------------
/*DIMENSÃO ESPECIE DOCUMENTO*/
SELECT
         DISTINCT(s.[esp_docto]) as [Código Espécie Documento Legado]
         ,s.esp_abrev                           AS [Abreviatura Espécie Documento]
         ,s.especie_docto             AS [Descrição Espécie Documento]
FROM stgMovimentoEstoque s

/*DIMENSÃO DEPÓSITO */
SELECT
         DISTINCT(s.[cod_depos]) as [Código Depósito Legado]
         ,CASE s.[cod_depos]
                  WHEN 'SAL' THEN 'Saldo'
                  WHEN 'REJ' THEN 'Rejeição'
                  WHEN 'EXP' THEN 'Expedição'
                  WHEN 'ANC' THEN 'Área Não Conformidade'
                  WHEN 'AL2' THEN 'Almoxarifado 2'
                  WHEN 'MTG' THEN 'MTG'
                  WHEN 'FER' THEN 'Ferramentaria'
                  WHEN 'DCQ' THEN 'Qualidade'
                  WHEN 'MNF' THEN 'Fábrica'
                  WHEN 'PQI' THEN 'PQI'
                  WHEN 'RET' THEN 'Retrabalho'
                  WHEN 'EBC' THEN 'EBC'
                  WHEN 'EPI' THEN 'EPI'
                  WHEN 'ALM' THEN 'Almoxarifado 1'
         END       AS [Descrição Depósito]
FROM stgMovimentoEstoque s
```

```
/*DIMENSÃO PESSOA*/
SELECT
DISTINCT
          (s.[cod_emitente]) as [Código Pessoa Legado]
          ,s.[nome_abrev] AS [Nome Pessoa]
          ,s.natureza_emitente as [Natureza Pessoa]
          ,s.identific_emit   as [Tipo Pessoa]
FROM stgMovimentoEstoque s

/*DIMENSÃO FILIAL*/
select
          distinct(s.[cod_estabel]) as [Código Filial Legado]
          ,CASE s.[cod_estabel]
                    WHEN '1' THEN 'RS'
                    WHEN '2' THEN 'SC'
                    WHEN '3' THEN 'SP'
                    WHEN '4' THEN 'PR'
                    WHEN '5' THEN 'RJ'
                    WHEN '6' THEN 'RS'
                    WHEN '7' THEN 'MG'
          END       AS [UF]
          ,CASE s.[cod_estabel]
                    WHEN '1' THEN 'Caxias do Sul'
                    WHEN '2' THEN 'Blumenau'
                    WHEN '3' THEN 'Campinas'
                    WHEN '4' THEN 'Londrina'
                    WHEN '5' THEN 'Macaé'
                    WHEN '6' THEN 'Lajeado'
                    WHEN '7' THEN 'Contagem'
          END       AS [Município]
FROM stgMovimentoEstoque s
```

APÊNDICE D – CRIAÇÃO DAS DIMENSÕES

```
CREATE TABLE [dbo].[dDeposito](
          [Código Depósito] [int] IDENTITY(1,1) NOT NULL,
          [Código Depósito Legado] [varchar](6) NULL,
          [Descrição Depósito] [varchar](14) NULL
)

CREATE TABLE [dbo].[dEspecieDocumento](
          [Código Espécie Documento] [int] IDENTITY(1,1) NOT NULL,
          [Código Espécie Documento Legado] [int] NULL,
          [Abreviatura Espécie Documento] [nvarchar](255) NULL,
          [Descrição Espécie Documento] [nvarchar](255) NULL
)

CREATE TABLE [dbo].[dFamiliaComercial](
          [Descrição Família Comercial] [varchar](60) NULL,
          [Unidade Medida Família Comercial] [varchar](4) NULL,
          [Baixa Estoque] [varchar](3) NULL,
          [Código Família Comercial Legado] [varchar](16) NULL,
          [Código Família Comercial] [int] IDENTITY(1,1) NOT NULL
)

CREATE TABLE [dbo].[dFamiliaMaterial](
          [Baixa Estoque] [bit] NULL,
          [Código Comprador] [varchar](24) NULL,
          [Depósito Padrão] [varchar](6) NULL,
          [Descrição Família Material] [varchar](60) NULL,
          [Código Família Material Legado] [varchar](16) NULL,
          [Lote Mínimo Padrão] [numeric](19, 4) NULL,
          [Lote Múltiplo Padrão] [numeric](19, 4) NULL,
          [Lote Reposição Padrão] [numeric](17, 2) NULL,
          [Tempo Segurança Padrão] [int] NULL,
          [Unidade Medida Padrão] [varchar](4) NULL,
          [Código Família Material] [int] IDENTITY(1,1) NOT NULL
)

CREATE TABLE [dbo].[dFilial](
          [Código Filial Legado] [varchar](10) NULL,
          [UF] [varchar](2) NULL,
          [Município] [varchar](13) NULL,
          [Código Filial] [int] IDENTITY(1,1) NOT NULL
)

CREATE TABLE [dbo].[dGrupoEstoque](
          [Código Grupo Estoque] [int] IDENTITY(1,1) NOT NULL,
          [Descrição Grupo Estoque] [varchar](60) NULL,
          [Código Grupo Estoque Legado] [int] NULL
)

CREATE TABLE [dbo].[dPessoa](
          [Código Pessoa] [int] IDENTITY(1,1) NOT NULL,
          [Código Pessoa Legado] [int] NULL,
          [Nome Pessoa] [nvarchar](255) NULL,
          [Natureza Pessoa] [varchar](50) NULL,
          [Tipo Pessoa] [varchar](50) NULL
)

CREATE TABLE [dbo].[dProduto](
          [Código Produto] [int] IDENTITY(1,1) NOT NULL,
          [Código Produto Legado] [varchar](32) NULL,
          [Código Grupo Estoque Legado] [int] NULL,
          [Código Família Comercial Legado] [varchar](16) NULL,
          [Código Família Material Legado] [varchar](16) NULL,
          [Alíquota IPI] [numeric](17, 2) NULL,
          [Alíquota ISS] [numeric](17, 2) NULL,
          [Baixa Estoque] [varchar](3) NULL,
          [Classificação Fiscal] [varchar](20) NULL,
          [Classificação ABC] [int] NULL,
          [Código Comprador Legado] [varchar](24) NULL,
          [Situação] [varchar](27) NULL,
          [compr_fabric] [int] NULL,
          [Curva ABC] [bit] NULL,
          [Data Implantação] [date] NULL,
          [Data Liberação] [date] NULL,
          [Data Obsolescência] [date] NULL,
          [Depósito Padrão] [varchar](6) NULL,
          [Descrição] [varchar](120) NULL,
          [Lote Mínimo] [numeric](19, 4) NULL,
          [Lote Múltiplo] [numeric](19, 4) NULL,
          [Unidade Medida] [varchar](4) NULL,
          [Usuário Alteração] [varchar](24) NULL
)
```

APÊNDICE E – COMANDOS DE IMPORTAÇÃO PARA O HDFS

```
sqoop import --connect 'jdbc:sqlserver://dev3h15;database=DBTC2' --username alexandre2dbtc --password t --table
vwGrupoEstoque -m 1 --fields-terminated-by '\t' --target-dir /user/cloudera/GrupoEstoque/;

sqoop import --connect 'jdbc:sqlserver://dev3h15;database=DBTC2' --username alexandre2dbtc --password t --table
vwFamiliaComercial -m 1 --fields-terminated-by '\t' --target-dir /user/cloudera/FamiliaComercial/;

sqoop import --connect 'jdbc:sqlserver://dev3h15;database=DBTC2' --username alexandre2dbtc --password t --table
vwFamiliaMaterial -m 1 --fields-terminated-by '\t' --target-dir /user/cloudera/FamiliaMaterial/;

sqoop import --connect 'jdbc:sqlserver://dev3h15;database=DBTC2' --username alexandre2dbtc --password t --table vwProduto -m
1 --fields-terminated-by '\t' --target-dir /user/cloudera/Produto/;

sqoop import --connect 'jdbc:sqlserver://dev3h15;database=DBTC2' --username alexandre2dbtc --password t --table
vwMovimentoEstoque -m 1 --fields-terminated-by '\t' --target-dir /user/cloudera/MovimentoEstoque/;
```

APÊNDICE F – COMANDOS DE CRIAÇÃO DO BANCO E TABELAS NO HIVE

```
create database bdtc2
use bdtc2

create table MovimentoEstoque(base_calculo int, cod_barras string, cod_depos string, cod_emitente int, cod_estabel string,
cod_estabel_des string, cod_localiz string, cod_lote_fabrican string, cod_orig_trans string, cod_prog_orig string, cod_refer
string, cod_roteiro string, cod_unid_negoc string, cod_unid_negoc_sdo string, cod_usu_ult_alter string, conta_contabil
string, conta_saldo string, contabilizado int, ct_codigo string, ct_saldo string, dat_fabricc_lote date,
dat_valid_lote_fabrican date, denominacao string, descricao_db_ok string, descricao_produto string, dt_contab date,
dt_criacao date, dt_nf_saida date, dt_trans date, esp_abrev string, esp_docto int, especie_docto string, hr_contab string,
hr_trans string, identific_emit string, it_codigo string, item_pai string, lote string, nat_operacao string,
natureza_emitente string, nom_fabrican string, nome_abrev string, nr_ord_produ int, nr_ord_refer int, nr_reporte int,
nr_req_sum int, nr_trans int, nr_trans_deb int, nro_docto string, num_ord_des int, num_ord_inv int, num_seq_des int,
num_sequen int, numero_ordem int, op_codigo int, op_seq int, origem_valor string, per_ppm double, peso_liquido double,
quantidade double, refer_contab string, referencia string, saldo_req double, sc_codigo string, sc_saldo string, sequen_nf
int, serie_docto string, tipo_preco1 string, tipo_preco2 string, tipo_preco3 string, tipo_trans int, tipo_valor int, un
string, usuario string, val_cofins double, val_cofins_cmi double, val_cofins_fasb double, valor_ggf_m1 string, valor_ggf_m2
string, valor_ggf_m3 string, valor_ggf_o1 string, valor_ggf_o2 string, valor_ggf_o3 string, valor_ggf_p1 string, valor_ggf_p2
string, valor_ggf_p3 string, valor_icm double, valor_ipi double, valor_iss double, valor_mat_m1 string, valor_mat_m2 string,
valor_mat_m3 string, valor_mat_o1 string, valor_mat_o2 string, valor_mat_o3 string, valor_mat_p1 string, valor_mat_p2 string,
valor_mat_p3 string, valor_mob_m1 string, valor_mob_m2 string, valor_mob_m3 string, valor_mob_o1 string, valor_mob_o2 string,
valor_mob_o3 string, valor_mob_p1 string, valor_mob_p2 string, valor_mob_p3 string, valor_nota double, valor_pis double,
vl_icm_fasb1 string, vl_icm_fasb2 string, vl_ipi_fasb1 string, vl_ipi_fasb2 string, vl_iss_fasb1 string, vl_iss_fasb2 string,
vl_nota_fasb1 string, vl_nota_fasb2 string, vl_pis_cmi double, vl_pis_fasb double, vl_taxa double) row format delimited
fields terminated by '\t' location '/user/cloudera/MovimentoEstoque';

create table FamiliaComercial(baixa_estoq int, char_1 string, char_2 string, check_sum string, data_1 date, data_2 date,
dec_1 double, dec_2 double, descricao string, fator_conver double, fm_cod_com  string, ind_imp_desc int, int_1 int, int_2
int, log_1 int, log_2 int, log_aloc_neg int, tp_aloc_lote int, un string) row format delimited fields terminated by '\t'
location '/user/cloudera/FamiliaComercial';

create table FamiliaMaterial(atu_conf int, baixa_estoq int, cap_est_fabr double, cd_formula int, cd_origem int, cd_planejado
string, cd_trib_icm int, cd_trib_ipi int, cd_trib_iss int, cdn_nat_desp int, cdn_tipo_rec_desp int, char_1 string, char_2
string, check_sum string, ciclo_contag int, class_fiscal string, classe_repro int, cod_classe_frete string, cod_comprado
string, cod_grupo_compra_o string, cod_grupo_compra_p string, cod_item_refugo string, cod_lista_destino string, cod_localiz
string, cod_malha string, cod_pulmao string, cod_reabast string, cod_servico int, cod_taxa int, Cod_var_tempo_res string,
codigo_orig int, cons_consumo int, cons_produto int, cons_saldo int, contr_qualid int, conv_tempo_seg int, criticidade int,
curva_abc int, data_1 date, data_2 date, dec_1 double, dec_2 double, demanda int, deposito_pad string, descricao string,
div_ordem int, dsl_formul_quant_cq string, emissao_ord int, fator_conver double, fator_ponder string, fator_refugo double,
fm_codigo string, fraciona int, geracao_ordem int, horiz_fixo int, idi_classif_item int, ind_calc_meta int, ind_cons_prv
double, ind_imp_desc int, ind_inf_qtf int, ind_lista_mrp int, ind_prev_demanda int, ind_refugo int, int_1 int, int_2 int,
invent_cla_a int, invent_cla_b int, invent_cla_c int, loc_unica int, log_1 int, log_2 int, log_altera_valid_lote int,
log_atualiz_via_mmp int, log_control_estoq_refugo int, log_inspec_lote int, log_multi_malha int, log_orig_ext int,
log_refugo_preco_fisc int, log_rotei_pend int, log_utiliza_batch_padrao int, lote_economi double, lote_minimo double,
lote_multipl double, lote_repos double, moeda_padrao int, mp_restrit int, niv_rest_icms string, nivel int, nivel_apr_compra
int, nivel_apr_manut int, nivel_apr_requis int, nivel_apr_solic int, nr_linha int, num_id_familia int, per_ppm double,
per_rest_icms double, perc_nqa double, periodo_fixo int, perm_saldo_neg int, politica int, politica_aps int, prioridade int,
pto_repos double, qt_max_ordem double, qtd_batch_padrao double, Qtd_min_reabast double, Qtd_var_ressup double, qtde_fixa
double, qtde_max double, quant_perda double, quant_segur double, rep_prod int, reporte_ggf int, reporte_mob int, res_cq_comp
int, res_cq_fabri int, res_for_comp int, res_int_comp int, ressup_fabri int, sazonalidade string, sit_aloc int, tempo_segur
int, tipo_con_est int, tipo_contr int, tipo_est_seg int, tipo_formula int, tipo_insp int, tipo_lote_ec int, tipo_requis int,
tipo_sched int, tp_adm_lote int, tp_cons_prev double, un string, val_relac_refugo_item double, var_mob_maior double,
var_mob_menor double, var_pre_perm double, var_preco int, var_qtd_perm double, var_quantid int, var_rep double, var_req_maior
double, var_req_menor double, var_transf double, vl_max_inv double, vl_min_inv double) row format delimited fields terminated
by '\t' location '/user/cloudera/FamiliaMaterial';

create table GrupoEstoque(char_1 string, char_2 string, check_sum string, data_1 date, data_2 date, dec_1 double, dec_2
double, descricao string, ge_codigo int, int_1 int, int_2 int, log_1 int, log_2 int, log_orig_ext int, num_id_grup_estoque
int, tipo_ge int) row format delimited fields terminated by '\t' location '/user/cloudera/GrupoEstoque';

create table Produto(aliquota_ii double, aliquota_ipi double, aliquota_ISS double, alt_refer int, altura double, atu_conf
int, baixa_estoq int, calc_cons_prev int, calc_lead_time int, cap_est_fabr double, capac_recip_beb int, cd_folh_item string,
cd_folh_lote string, cd_formula int, cd_origem int, cd_planejado string, cd_referencia string, cd_tag string, cd_trib_icm
int, cd_trib_ipi int, cd_trib_iss int, char_1 string, char_2 string, check_sum string, ciclo_contag int, class_fiscal string,
classe_repro int, classif_abc int, cod_auxiliar string, cod_comprado string, cod_dcr_item string, cod_destaq int, cod_estabel
string, cod_imagem string, cod_item_refugo string, cod_lista_destino string, cod_localiz string, cod_malha string,
cod_obsoleto int, cod_produto string, cod_pulmao string, cod_pulmao_proces string, cod_refer string, cod_servico int,
cod_tab_preco_aps string, cod_tax int, cod_tax_serv int, cod_trib_ii int, cod_unid_negoc string, cod_workflow string,
codigo_orig int, codigo_refer string, compr_fabric int, comprim double, concentracao double, cons_consumo int, cons_produto
int, cons_saldo int, consumo_aad double, consumo_prev double, conta_aplicacao string, contr_plan int, contr_qualid int,
conv_tempo_seg int, criticidade int, ct_codigo string, curva_abc int, data_1 date, data_2 date, data_base date, data_implant
date, data_liberac date, data_obsol date, data_ult_con date, data_ult_ent date, data_ult_rep date, data_ult_sai date,
de_codigo_prin string, dec_1 double, dec_2 double, dec_conv_fmcoml int, dec_ftcon int, demanda int, deposito_pad string,
desc_inter string, desc_item string, desc_nacional string, descricao_1 string, descricao_2 string, dias_estoq_aloc string,
div_ordem int, dsl_destaq string, dt_pr_fisc date, dt_ult_ben date, emissao_ord int, enquad_beb int, esp_beb int, fase_medio
int, fator_conver double, fator_reaj_icms double, fator_refugo double, fm_cod_com string, fm_codigo string, fraciona int,
ft_conv_fmcoml double, ft_conversao double, ge_codigo int, geracao_ordem int, horiz_fixo int, id_grade int, idi_classif_item
int, incentivado int, ind_backorder int, ind_calc_meta int, ind_confprodcom int, ind_especifico int, ind_imp_desc int,
ind_inf_qtf int, ind_ipi_dife int, ind_item_fat int, ind_prev_demanda int, ind_quotas int, ind_refugo int, ind_serv_mat int,
inform_compl string, INT_1 int, int_2 int, isencao_import int, it_codigo string, it_demanda string, largura double, loc_unica
int, log_1 int, log_2 int, log_altera_valid_lote int, log_atualiz_via_mmp int, log_carac_tec int, log_consid_aloc_ativid int,
log_control_estoq_refugo int, log_inspec_lote int, log_multi_malha int, log_necessita_li int, log_orig_ext int,
log_programac_sfc int, log_refugo_preco_fisc int, log_tax_produc int, log_utiliza_batch_padrao int, lote_economi double,
lote_minimo double, lote_multipl double, lote_mulven double, lote_repos double, moeda_padrao int, mp_restrit int, nat_despesa
int, niv_mais_bai int, niv_mps int, niv_rest_fora string, niv_rest_icms string, nivel int, nivel_apr_compra int,
nivel_apr_manut int, nivel_apr_requis int, nivel_apr_solic int, nr_item_dcr int, nr_linha int, nr_pontos_quotas int,
Nr_ult_peca int, num_id_item int, path string, per_min_luc double, per_ppm double, per_rest_fora double, per_rest_icms
double, perc_demanda double, perc_nqa double, periodo_fixo int, perm_saldo_neg int, peso_bruto double, peso_liquido double,
pm_ja_calc int, politica int, politica_aps int, pr_sem_tx double, preco_base double, preco_fiscal double, preco_repos double,
preco_ul_ent double, prefixo_lote string, prioridade int, pto_repos double, qt_max_ordem double, qt_var_max double,
qt_var_min double, qtd_batch_padrao double, qtd_refer_custo_dis double, qtde_fixa double, qtde_max double, quant_pacote
double, quant_perda double, quant_segur double, rendimento double, rep_prod int, reporte_ggf int, reporte_mob int,
res_cq_comp int, res_cq_fabri int, res_for_comp int, res_int_comp int, responsavel string, ressup_fabri int, resumo_mp int,
rot_quant int, rot_refer int, rot_revis int, sc_codigo string, sit_aloc int, tempo_segur int, tipo_atp int, tipo_con_est int,
tipo_contr int, tipo_desc_nt int, tipo_est_seg int, tipo_formula int, tipo_insp int, tipo_lote_ec int, tipo_recip_beb int,
tipo_requis int, tipo_sched int, tp_adm_lote int, tp_aloc_lote int, tp_cons_prev double, tp_desp_padrao int, tp_lote_econom
int, tp_lote_minimo int, tp_lote_multiplo int, tx_importacao double, un string, usuario_alt string, usuario_obsol string,
```

```
val_fator_custo_dis double, val_overlap double, val_relac_refugo_item double, valor_ipi_beb double, var_mob_maior double,
var_mob_menor double, var_rep double, var_req_maior double, var_req_menor double, var_transf double, variac_acum double,
vl_mat_ant double, vl_mob_ant double, vl_var_max double, vl_var_min double, volume double) row format delimited fields
terminated by '\t' location '/user/cloudera/Produto';
```

APÊNDICE G – COMANDOS DE CRIAÇÃO DAS VIEWS NO BDTC2 NO HIVE

```
CREATE VIEW vwCompras AS
 SELECT
         s.cod_estabel
         ,to_date(concat(year(s.dt_trans),'-',month(s.dt_trans),'-','01')) as `data`
         ,s.it_codigo
         ,max(p.desc_item) as `desc_item`
         ,max(ge.ge_codigo) as `ge_codigo`
         ,max(ge.descricao) as `desc_ge`
         ,max(s.descricao_produto) as `descricao_produto`
         ,max(fc.fm_cod_com) as `fc_codigo`
         ,max(fc.descricao) as `desc_fc`
         ,max(fm.fm_codigo) as `fm_codigo`
         ,max(fm.descricao) as `desc_fm`

         ,SUM(if(s.tipo_trans = 1,s.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(s.tipo_trans = 1,0,s.quantidade)) as `quantidade_saida`
         ,SUM(IF(s.tipo_trans = 1,s.quantidade,s.quantidade * -1)) as `quantidade_saldo`
  FROM MovimentoEstoque AS `s`
    LEFT JOIN produto AS `p` on p.it_codigo = s.it_codigo
    LEFT JOIN grupoestoque AS `ge` on ge.ge_codigo = p.ge_codigo
    LEFT JOIN familiacomercial AS `fc` on fc.fm_cod_com = p.fm_cod_com
    LEFT JOIN familiamaterial AS `fm` on fm.fm_codigo = p.fm_codigo
  WHERE s.esp_docto in (21)
  GROUP BY
         s.cod_estabel
         , year(s.dt_trans)
         , month(s.dt_trans)
         , s.it_codigo;

-------------------------------------------------------------------------------------------------------------------------

CREATE VIEW vwVendas AS
 SELECT
         s.cod_estabel
         ,to_date(concat(year(s.dt_trans),'-',month(s.dt_trans),'-','01')) as `data`
         ,s.it_codigo
         ,max(p.desc_item) as `desc_item`
         ,max(ge.ge_codigo) as `ge_codigo`
         ,max(ge.descricao) as `desc_ge`
         ,max(s.descricao_produto) as `descricao_produto`
         ,max(fc.fm_cod_com) as `fc_codigo`
         ,max(fc.descricao) as `desc_fc`
         ,max(fm.fm_codigo) as `fm_codigo`
         ,max(fm.descricao) as `desc_fm`

         ,SUM(if(s.tipo_trans = 1,s.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(s.tipo_trans = 1,0,s.quantidade)) as `quantidade_saida`
         ,SUM(IF(s.tipo_trans = 1,s.quantidade,s.quantidade * -1)) as `quantidade_saldo`
  FROM MovimentoEstoque AS `s`
    LEFT JOIN produto AS `p` on p.it_codigo = s.it_codigo
    LEFT JOIN grupoestoque AS `ge` on ge.ge_codigo = p.ge_codigo
    LEFT JOIN familiacomercial AS `fc` on fc.fm_cod_com = p.fm_cod_com
    LEFT JOIN familiamaterial AS `fm` on fm.fm_codigo = p.fm_codigo
  WHERE s.esp_docto in (22)
    AND ge.ge_codigo in (80,81)
  GROUP BY
         s.cod_estabel
         , year(s.dt_trans)
         , month(s.dt_trans)
         , s.it_codigo;

-------------------------------------------------------------------------------------------------------------------------

CREATE VIEW vwProducao AS
 SELECT
         s.cod_estabel
         ,to_date(concat(year(s.dt_trans),'-',month(s.dt_trans),'-','01')) as `data`
         ,s.it_codigo
         ,max(p.desc_item) as `desc_item`
         ,max(ge.ge_codigo) as `ge_codigo`
         ,max(ge.descricao) as `desc_ge`
         ,max(s.descricao_produto) as `descricao_produto`
         ,max(fc.fm_cod_com) as `fc_codigo`
         ,max(fc.descricao) as `desc_fc`
         ,max(fm.fm_codigo) as `fm_codigo`
         ,max(fm.descricao) as `desc_fm`

         ,SUM(if(s.tipo_trans = 1,s.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(s.tipo_trans = 1,0,s.quantidade)) as `quantidade_saida`
         ,SUM(IF(s.tipo_trans = 1,s.quantidade,s.quantidade * -1)) as `quantidade_saldo`
  FROM MovimentoEstoque AS `s`
    LEFT JOIN produto AS `p` on p.it_codigo = s.it_codigo
    LEFT JOIN grupoestoque AS `ge` on ge.ge_codigo = p.ge_codigo
    LEFT JOIN familiacomercial AS `fc` on fc.fm_cod_com = p.fm_cod_com
    LEFT JOIN familiamaterial AS `fm` on fm.fm_codigo = p.fm_codigo
  WHERE s.esp_docto in (1,8,25)
  GROUP BY
         s.cod_estabel
         , year(s.dt_trans)
         , month(s.dt_trans)
         , s.it_codigo;
```

APÊNDICE H – COMANDOS DE CRIAÇÃO DE VIEWS INTERMEDIÁRIAS PARA CRIAÇÃO DE DIMENSÕES EXTRAÍDAS

```
CREATE VIEW dwtcstg.vwEspecieDocumento AS
    SELECT
          DISTINCT(s.esp_docto) as `Código Espécie Documento Legado`
          ,s.esp_abrev                          AS `Abreviatura Espécie Documento`
          ,s.especie_docto        AS `Descrição Espécie Documento`
    FROM bdtc.MovimentoEstoque AS `s`;


CREATE VIEW dwtcstg.vwFilial AS
    select
          distinct(s.cod_estabel) as `Código Filial Legado`
          ,CASE s.cod_estabel
                    WHEN '1' THEN 'RS'
                    WHEN '2' THEN 'SC'
                    WHEN '3' THEN 'SP'
                    WHEN '4' THEN 'PR'
                    WHEN '5' THEN 'RJ'
                    WHEN '6' THEN 'RS'
                    WHEN '7' THEN 'MG'
          END       AS `UF`
          ,CASE s.cod_estabel
                    WHEN '1' THEN 'Caxias do Sul'
                    WHEN '2' THEN 'Blumenau'
                    WHEN '3' THEN 'Campinas'
                    WHEN '4' THEN 'Londrina'
                    WHEN '5' THEN 'Macaé'
                    WHEN '6' THEN 'Lajeado'
                    WHEN '7' THEN 'Contagem'
          END       AS `Município`
    FROM bdtc.movimentoestoque AS `s`;
```

APÊNDICE I – COMANDOS DE CRIAÇÃO DAS DIMENSÕES NO DWTC NO HIVE

```
------------------- DIMENSÕES CARREGADAS ------------------
/*DIMENSÃO PRODUTO*/
CREATE TABLE dwtc.dProduto AS
    SELECT
        row_number() over () AS  `Código Produto`
      ,`aliquota_ipi`                 AS `Alíquota IPI`
      ,`aliquota_ISS`                 AS `Alíquota ISS`
      ,IF(`baixa_estoq` = 1, 'SIM', 'NÃO') AS `Baixa Estoque`
      ,`class_fiscal`                 AS `Classificação Fiscal`
      ,`classif_abc`                  AS `Classificação ABC`
      ,`cod_comprado`                 AS `Código Comprador Legado`
      , CASE `cod_obsoleto`
                  WHEN 1 THEN 'Ativo'
                  WHEN 2 THEN 'Obsoleto Ordens Automáticas'
                  WHEN 3 THEN 'Obsoleto Todas as Ordens'
                  WHEN 4 THEN 'Totalmente Obsoleto'
            END     AS `Situação`
      ,`compr_fabric`  --Buscar tradução
      ,`curva_abc`                    AS `Curva ABC`
      ,`data_implant`                 AS `Data Implantação`
      ,`data_liberac`                 AS `Data Liberação`
      ,`data_obsol`                   AS `Data Obsolescência`
      ,`deposito_pad`                 AS `Depósito Padrão`
      ,`desc_item`                    AS `Descrição`
      ,`fm_cod_com`                   AS `Código Família Comercial Legado`
      ,`fm_codigo`                    AS `Código Família Material Legado`
      ,`ge_codigo`                    AS `Código Grupo Estoque Legado`
      ,`it_codigo`                    AS `Código Produto Legado`
      ,`lote_minimo`                  AS `Lote Mínimo`
      ,`lote_multipl`                 AS `Lote Múltiplo`
      ,`un`                                           AS `Unidade Medida`
      ,`usuario_alt`                  AS `Usuário Alteração`

      FROM bdtc.Produto
      CLUSTER BY `Código Produto`;

-------------------------------------------------------------------------------
/*DIMENSÃO GRUPO ESTOQUE*/
CREATE TABLE dwtc.dGrupoEstoque AS
    SELECT
         row_number() over () AS  `Código Grupo Estoque`
        ,`descricao`                  as `Descrição Grupo Estoque`
        ,`ge_codigo`                  as `Código Grupo Estoque Legado`
      FROM bdtc.GrupoEstoque
    CLUSTER BY `Código Grupo Estoque`;

-------------------------------------------------------------------------------
/*DIMENSÃO FAMÍLIA COMERCIAL*/
CREATE TABLE dwtc.dFamiliaComercial AS
    SELECT
           row_number() over () AS  `Código Família Comercial`
          ,`descricao`               AS `Descrição Família Comercial`
      ,`un`                                    AS `Unidade Medida Família Comercial`
      ,if(`baixa_estoq`=1,'SIM','NÃO')  AS `Baixa Estoque`
      ,`fm_cod_com`          AS `Código Família Comercial Legado`
    FROM bdtc.FamiliaComercial
    CLUSTER BY `Código Família Comercial`

-------------------------------------------------------------------------------
/*DIMENSÃO FAMÍLIA MATERIAL*/
CREATE TABLE dwtc.dFamiliaMaterial AS
  SELECT
      row_number() over () AS  `Código Família Material`
      ,`baixa_estoq`                  as `Baixa Estoque`
      ,`cod_comprado`                 as `Código Comprador`
      ,`deposito_pad`                 as `Depósito Padrão`
      ,`descricao`                    as `Descrição Família Material`
      ,`fm_codigo`                    as `Código Família Material Legado`
      ,`lote_minimo`                  AS `Lote Mínimo Padrão`
      ,`lote_multipl`                 AS `Lote Múltiplo Padrão`
      ,`lote_repos`                   AS `Lote Reposição Padrão`
      ,`tempo_segur`                  AS `Tempo Segurança Padrão`
      ,`un`                                           AS `Unidade Medida Padrão`
  FROM bdtc.FamiliaMaterial
  CLUSTER BY `Código Família Material`

------------------- DIMENSÕES EXTRAÍDAS ------------------
/*DIMENSÃO ESPÉCIE DOCUMENTO*/
CREATE TABLE dwtc.dEspecieDocumento AS
    SELECT
        ROW_NUMBER() OVER () AS `Código Espécie Documento`
        ,vw.`código espécie documento legado`
        ,vw.`abreviatura espécie documento`
        ,vw.`descrição espécie documento`
    FROM dwtcstg.vwespeciedocumento vw
    CLUSTER BY `Código Espécie Documento`;

/*DIMENSÃO FILIAL*/
CREATE TABLE dwtc.dFilial AS
    SELECT
        ROW_NUMBER() OVER () AS `Código Filial`
        ,vw.`código filial legado`
        ,vw.`município`
        ,vw.uf
        FROM dwtcstg.vwFilial vw
```

APÊNDICE J - COMANDOS DE CRIAÇÃO DAS TABELAS FATO NO DWTC NO HIVE

```
--------- FATO VENDAS ------------
CREATE TABLE dwtc.fVendas AS
 SELECT
          to_date(concat(year(me.dt_trans),'-',month(me.dt_trans),'-','01')) as `data`
         ,fi.`código filial`
         ,p.`código produto`
         ,max(ge.`código grupo estoque`) as `código grupo estoque`
         ,max(fc.`código família comercial`) as `código família comercial`
         ,max(fm.`código família material`) as `código família material`
         ,SUM(if(me.tipo_trans = 1,me.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(me.tipo_trans = 1,0,me.quantidade)) as `quantidade_saida`
         ,SUM(IF(me.tipo_trans = 1,me.quantidade,me.quantidade * -1)) as `quantidade_saldo`
  FROM bdtc.MovimentoEstoque AS `me`
    LEFT JOIN dwtc.dfilial AS `fi` on fi.`código filial legado` = me.cod_estabel
    LEFT JOIN dwtc.dproduto AS `p` on p.`código produto legado` = me.it_codigo
    LEFT JOIN dwtc.dgrupoestoque AS `ge` on ge.`código grupo estoque legado` = p.`código grupo estoque legado`
    LEFT JOIN dwtc.dfamiliacomercial AS `fc` on fc.`código família comercial legado` = p.`código família comercial legado`
    LEFT JOIN dwtc.dfamiliamaterial AS `fm` on fm.`código família material legado` = p.`código família material legado`
    LEFT JOIN dwtc.despeciedocumento AS `ed` on ed.`código espécie documento legado` = me.esp_docto
  WHERE me.esp_docto in (22)
    AND ge.`código grupo estoque legado` in (80,81)
  GROUP BY
         fi.`código filial`
         , year(me.dt_trans)
         , month(me.dt_trans)
         , p.`código produto`;


--------- FATO COMPRAS ------------
CREATE TABLE dwtc.fCompras AS
 SELECT
          to_date(concat(year(me.dt_trans),'-',month(me.dt_trans),'-','01')) as `data`
         ,fi.`código filial`
         ,p.`código produto`
         ,max(ge.`código grupo estoque`) as `código grupo estoque`
         ,max(fc.`código família comercial`) as `código família comercial`
         ,max(fm.`código família material`) as `código família material`
         ,SUM(if(me.tipo_trans = 1,me.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(me.tipo_trans = 1,0,me.quantidade)) as `quantidade_saida`
         ,SUM(IF(me.tipo_trans = 1,me.quantidade,me.quantidade * -1)) as `quantidade_saldo`
  FROM bdtc.MovimentoEstoque AS `me`
    LEFT JOIN dwtc.dfilial AS `fi` on fi.`código filial legado` = me.cod_estabel
    LEFT JOIN dwtc.dproduto AS `p` on p.`código produto legado` = me.it_codigo
    LEFT JOIN dwtc.dgrupoestoque AS `ge` on ge.`código grupo estoque legado` = p.`código grupo estoque legado`
    LEFT JOIN dwtc.dfamiliacomercial AS `fc` on fc.`código família comercial legado` = p.`código família comercial legado`
    LEFT JOIN dwtc.dfamiliamaterial AS `fm` on fm.`código família material legado` = p.`código família material legado`
    LEFT JOIN dwtc.despeciedocumento AS `ed` on ed.`código espécie documento legado` = me.esp_docto
  WHERE me.esp_docto in (21)
  GROUP BY
         fi.`código filial`
         , year(me.dt_trans)
         , month(me.dt_trans)
         , p.`código produto`;

--------- FATO PRODUÇÃO ------------

CREATE TABLE dwtc.fProducao AS
 SELECT
          to_date(concat(year(me.dt_trans),'-',month(me.dt_trans),'-','01')) as `data`
         ,fi.`código filial`
         ,p.`código produto`
         ,max(ge.`código grupo estoque`) as `código grupo estoque`
         ,max(fc.`código família comercial`) as `código família comercial`
         ,max(fm.`código família material`) as `código família material`
         ,SUM(if(me.tipo_trans = 1,me.quantidade,0)) as `quantidade_entrada`
         ,SUM(if(me.tipo_trans = 1,0,me.quantidade)) as `quantidade_saida`
         ,SUM(IF(me.tipo_trans = 1,me.quantidade,me.quantidade * -1)) as `quantidade_saldo`
  FROM bdtc.MovimentoEstoque AS `me`
    LEFT JOIN dwtc.dfilial AS `fi` on fi.`código filial legado` = me.cod_estabel
    LEFT JOIN dwtc.dproduto AS `p` on p.`código produto legado` = me.it_codigo
    LEFT JOIN dwtc.dgrupoestoque AS `ge` on ge.`código grupo estoque legado` = p.`código grupo estoque legado`
    LEFT JOIN dwtc.dfamiliacomercial AS `fc` on fc.`código família comercial legado` = p.`código família comercial legado`
    LEFT JOIN dwtc.dfamiliamaterial AS `fm` on fm.`código família material legado` = p.`código família material legado`
  WHERE me.esp_docto in (1,8,25)
  GROUP BY
         fi.`código filial`
         , year(me.dt_trans)
         , month(me.dt_trans)
         , p.`código produto`;
```