CENTRO UNIVERSITÁRIO DE ANÁPOLIS - UniEVANGÉLICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

ANDRÉ COSTA RIBEIRO
LUCAS HANANNI DE MELO SENA

APLICAÇÃO DE TÉCNICA DE *DATA WAREHOUSE* PARA GERENCIAMENTO LOGÍSTICO DE CENTRO DE DISTRIBUIÇÃO

ANÁPOLIS
2018 - 01

ANDRÉ COSTA RIBEIRO
LUCAS HANANNI DE MELO SENA

APLICAÇÃO DE TÉCNICA DE *DATA WAREHOUSE* PARA GERENCIAMENTO LOGÍSTICO DE CENTRO DE DISTRIBUIÇÃO

Projeto de Pesquisa apresentado ao Curso de Bacharelado em Engenharia da Computação do Centro Universitário de Anápolis - UniEVANGÉLICA como requisito parcial à aprovação na disciplina Trabalho de Conclusão de Curso II sob orientação da Prof. Esp. Aline Dayany de Lemos.

ANÁPOLIS
2018 - 01

ANDRÉ COSTA RIBEIRO
LUCAS HANANNI DE MELO SENA

APLICAÇÃO DE TÉCNICA DE *DATA WAREHOUSE* PARA GERENCIAMENTO LOGÍSTICO DE CENTRO DE DISTRIBUIÇÃO.

Projeto de Pesquisa apresentado ao Curso de Bacharelado em Engenharia da Computação do Centro Universitário de Anápolis - UniEVANGÉLICA como requisito parcial à aprovação na disciplina Trabalho de Conclusão de Curso II sob orientação da Prof. Esp. Aline Dayany de Lemos.

Banca Examinadora

.....
Prof. Esp. Aline Dayany de Lemos
Orientador

.....
Prof. Dra. Renata Dutra Braga
Convidado

.....
Prof. Me. Luciana Nishi
Convidado

Nota:.....

Anápolis, 11 de Julho de 2018.

RESUMO

Este trabalho demonstra o processo de implementação de um sistema de apoio a decisão em um banco de dados de centro de distribuição. Considerando a importância do gerenciamento de dados para uma empresa, observou-se a relevância de um sistema de apoio a decisão neste meio, para analisar a movimentação de seus produtos com o objetivo de aumentar sua produtividade e consequentemente os lucros. Sabendo disso, este trabalho propõe apresentar a técnica de *Data Warehouse*, um poderoso sistema de apoio a decisão de sistemas transacionais, que a partir de um bom planejamento, é possível analisar a empresa em várias perspectivas diferentes. Tal planejamento juntamente a sua implementação e resultados obtidos foram detalhados em ordem cronológica com objetivo de demonstrar o passo a passo e vantagens desta técnica.

Palavras-chave: *Data Warehouse*, Sistemas de Apoio a Decisão, Banco de dados, Centro de distribuição

ABSTRACT

This work demonstrates the implementing process of a decision support system in a distribution center database. Considering the importance of data management for a company, it was observed the relevance of a decision support system in this environment, to analyze the movement of its products in order to increase its productivity and consequently the profits. Knowing this, this paper proposes to present the technique of Data Warehouse, a powerful decision support system of transactional systems, that from a good planning, it is possible to analyze the company in several different perspectives. Such planning together with its implementation and results obtained were detailed in chronological order, to demonstrate the step-by-step and benefits of this technique.

Keywords: *Data Warehouse, decision support system, database, distribution center*

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS

BI	*Business Inteligence*
BD	Banco de Dados
DM	*Data Mining*
DW	*Data Warehouse*
EIS	*Executive Information System*
ETL	*Extract Transformation Load*
MER	Modelo de Entidade Relacional
SQL	*Structured Query Language*
CEP	Código de Endereçamento Postal
SGBD	Sistema de Gerenciamento de Banco de Dados
ODS	*Operation Data Storage*

SUMÁRIO

1 INTRODUÇÃO

É possível verificar que a era da informação está trazendo diversas transformações na sociedade, e uma delas seria a importância do conhecimento dentro do mercado de trabalho. Para que uma empresa possa crescer economicamente nos dias atuais, é praticamente necessário a busca incessante de novos conhecimentos.

Segundo Turban (2013, p.396) "[...] já se conhece que os dados são um ativo da empresa, mesmo que sua manutenção possa também representar um ônus. Por isso, em termos de informação e conhecimento, o uso dos dados é poder. "

Todos os centros de distribuições para que tenham um bom lucro no final do mês, devem ter uma boa gerência de todo o seu trabalho desenvolvido e suas informações. A venda de produtos não se limita apenas na compra e venda de uma mercadoria, mas sim de toda uma gerência bem organizada.

Para um bom gerenciamento é necessária uma boa tomada de decisão que por sua vez, depende das informações obtidas pelo gerente até aquele momento. Uma informação é muito importante e o bom ou mau uso desta, pode causar muitas consequências diferentes.

Segundo Gomes (2014, p.12-13) sobre decisões complexas:

> Tomar decisões complexas é, de modo geral, uma das mais difíceis tarefas enfrentadas individualmente ou por grupos de indivíduos, pois quase sempre tais decisões devem atender a múltiplos objetivos, e frequentemente seus impactos não podem ser corretamente identificados.

Geralmente o dever de decidir em uma empresa, é de responsabilidade de um gerente que precisa analisar bem a situação antes de tomar uma decisão. O que as empresas muitas vezes não sabem, é que a base de informações para isto, está em seu próprio banco de dados, e com a utilização do *Data Warehouse* para o gerenciamento, muitas decisões que seriam complexas se tornam fáceis e tranquilas para serem solucionadas.

Diante de um centro de distribuição que possui muitas tabelas pode gerar complexidade, o que é consequência de diversas dependências entre elas. Muitos bancos de dados têm deficiência na apresentação de informações que por sua vez podem ser relevantes para a resolução de problemas logísticos da empresa. Alguns exemplos práticos de informações logísticas de um centro de distribuição a serem resolvidos seriam: Como diminuir os seus gastos? Como aumentar suas vendas? Como atingir um maior público? Como diminuir a perda de produtos? Entre outros. O que fazer para facilitar as respostas logísticas sobre esses tipos de informações? Qual seria o procedimento ideal para retirar essas informações do banco de dados

sem prejudicar sua velocidade de processamento? Como estruturar essas informações periodicamente de maneira automatizada?

Turban (2013) define que o objetivo do *Data Warehouse* é criar um repositório de dados relevantes de uma empresa para as atividades de processamento analítico e também para o apoio a tomada de decisão, resolvendo desta forma cada um dos problemas levantados acima. Dessa forma o objetivo deste estudo é aplicar e demonstrar o quão poderoso é o uso do *Data Warehouse*.

Este trabalho aborda a importância da informação. Também apresenta os conceitos de *Data Warehouse* e toda a sua arquitetura e ferramentas a serem utilizados para sua aplicação. É levado em consideração um banco de dados de um centro de distribuição disponibilizada por uma empresa logística, estudando desta forma os processos essenciais para o desenvolvimento de um *Data Warehouse*. Dessa forma foram feitas análises sobre a eficiência do banco transacional e sua complexidade no que se refere o quão difícil é a obtenção de uma determina informação para um gerente específico. O estudo comprova o quão bom o *Data Warehouse* é e possibilita o conhecimento para qualquer pessoa que tenha a curiosidade de saber um pouco sobre seu conceito, desenvolvimento e aplicação.

A seguir serão apresentadas as atividades desenvolvidas neste trabalho. No primeiro momento será abordado conceitos teóricos sobre o *Data Warehouse*, retirados de livros e artigos. Logo após será apresentado o estudo de campo levando em consideração a aplicação do *Data Warehouse*, assim como a utilização das técnicas e ferramentas que auxiliaram na análise do banco de dados. E para finalizar será demonstrado os resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologia e Sistemas para Gerenciamento de Informações

A cada dia que passa, novas tecnologias são criadas, desenvolvidas e consequentemente é requerido uma estrutura de software mais elevada. As empresas constantemente buscam sistemas robustos e complexos. Tendem a pensar na qualidade do desenvolvimento de suas atividades.

Nery (2006, p.15) diz:

> O foco das técnicas estruturadas manteve uma busca incessante pela qualidade de software, sempre à procura de aplicação perfeita, com nível de erro mínimo, rotinas exaustivas e testes, bancos de dados e por aí vai. Mas sempre na busca de estabelecer elementos de controle operacional para a empresa. Sempre com direcionamento para automação de processos.

Devido a constante evolução e busca pelo conhecimento, as informações começaram a gerar complexidade para a tomada de decisão, principalmente se tratando da logísticas empresarial. Segundo Ballou (2009), A logística empresarial leva em consideração a melhora do desenvolvimento dos serviços prestados em uma empresa através dos estudos em administração, provendo desta forma o controle sobre o fluxo de produtos e assim ajudando na tomada de decisão.

É possível verificar que muitos dos problemas que são desenvolvidos pela logística, são problemas mais abstratos. A Abstração de tais problemas devem ser estudados mais cautelosamente para que assim possam identificar a melhor tomada de decisão para aquela situação. Sabendo da importância sobre estas manipulações de dados e sua utilização em sistemas empresariais, foi observado a necessidade de criar dois bancos de dados separados.

Segundo Silberschatz (2012, p.559) os dois tipos de banco de dados existentes: “De modo geral, as aplicações de banco de dados podem ser classificadas em sistemas de processamento de transação de dados e apoio a decisão”

Os dados tratados por sistemas transacionais são os famosos dados utilizados em um banco de dados com modelagem entidade e relacionamento (MER). Seriam os bancos já conhecidos e utilizados pelas empresas, onde possuem todo o gerenciamento de suas atividades. Silberschatz (2012, p.559) define sistemas transacionais: “[...] são sistemas que registram informações sobre transações, como informação de vendas de produtos para empresas, ou registro de curso e informação de notas das universidades”

Os dados tratados para a tomada de decisão são dados mais abrangentes, que tem por objetivo relacionar o tempo histórico de uma determinada entidade do banco de entidade-relacionamento.

Silberschatz (2012, p.559) define sistemas para apoio de decisão:

> Os sistemas de apoio a decisão visam obter informações de alto nível a partir de informações detalhadas armazenadas nos sistemas de processamento de transação, além de usar as informações de alto nível para tomar uma série de decisões.

Apesar da existência dos dois tipos de sistemas com seus respectivos bancos, muitas empresas buscam aprimorar mais os seus sistemas transacionais do que os sistemas sobre a tomada decisão. O motivo para tal, é porque muitas vezes não consideram a informação como um agente mais importante para sua empresa, mas sim a tecnologia dos sistemas transacionais, buscando a qualidade em operações de seu sistema.

Segundo Nery (2006) as falhas estruturais e os custos de desenvolvimento de sistemas, sempre deixaram para o último lugar as necessidades executivas de informação, porém nunca deixou de ser necessário um sistema que possa gerenciar as informações de tal forma a dar suporte a tomada de decisão.

Uma empresa que apresenta qualquer tipo de problema, seja este no meio sistêmico utilizado ou em um conflito empresarial, podem acabar tendo que confiar na tomada de decisão de um administrador. A responsabilidade aqui imposta depende de uma boa base de dados e de um bom domínio de negócio. Um administrador que não tem essa qualidade, pode levar a empresa a um estado ainda pior.

Para o gerenciamento de dados e informações de uma empresa pode ser utilizado diversos tipos de sistemas diferentes. De acordo com Nery (2006) um sistema utilizado nos anos de 1970 e 1980 foi o sistema *Executive Information System* (*EIS*), que só era utilizado por uma empresa em momentos de crise. Os sistemas foram desenvolvidos e trazem consigo novas tecnologias, como *Data Warehouse* (*DW*), *Business Inteligence* (*BI*), *Data mining* (*DM*), *Big data*, entre outros.

2.2 Diagnóstico e Problemas Empresariais

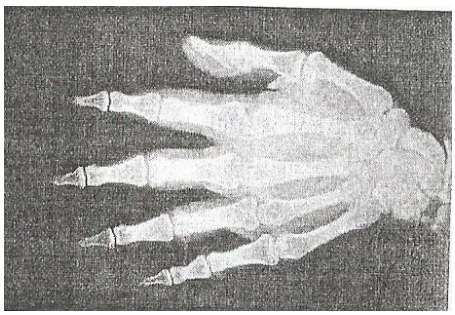
Quando um empreendedor está diante de um problema que necessite uma tomada de decisão, este pode buscar por auxilio em pessoas, ferramentas, técnicas entre outros. Determinar como resolver um problema, não é uma tarefa fácil, mas com a utilização de ferramentas que gerenciam as informações, pode acabar diminuindo a complexidade da situação e tornando assim a decisão para aquela solução, uma tarefa mais fácil.

Nery (2006, p.16), enfatiza:

> Da mesma forma que, quando vamos a um médico, ficamos satisfeitos quando o diagnóstico é realizado após exames detalhados de nossas condições físicas (exames de sangue, fezes, urina, raios X, etc.), os executivos necessitam, para diagnosticar e administrar as tendências de negócio, de um ambiente que lhes permita executar exames nos seus dados com a mesma capacidade de profundidade, transparência e evolução.

A analogia expressa pelo autor mostra que para atender um paciente, são feitas várias analises sobre o problema que foi exposto. Diante de vários exames, se obtém os resultados que por sua vez possibilitam o diagnóstico, definindo assim qual a doença e a cura para aquela situação. Como exemplo a figura 1 mostra um exemplo de raio X que permite as análises mencionadas a cima

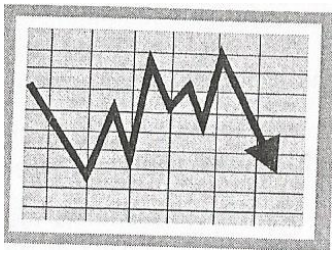
Figura 1 – Raio X utilizado para análise médica



Fonte: Nery, 2006, p.19.

Uma empresa que está com problema e precisa de uma solução, age da mesma forma. O empresário solicita ferramentas de análise de informações, que por sua vez, gera resultados e permite mostrar ao administrador quais são os conflitos e qual seria a melhor solução para aquele problema. Como exemplo de tal análise pode ser observado na figura 2.

Figura 2 – Análise história que pode ser utilizada entre dois ou mais parâmetros



Fonte: Nery, 2006, p.19.

Diante de uma doença o médico ainda pode solicitar uma análise histórica. Ao ser tratada a Diabetes, por exemplo, o médico pode verificar se esta é uma doença genética ou não através do histórico da vida do paciente, verificando seus antepassados e determinando qual o tipo de diabetes aquela determinada pessoa tem.

Em uma empresa é possível se ter a mesma percepção. Uma ferramenta de análise de informações, verifica o histórico e agrupa as informações de forma a demonstrar possíveis problemas e determinar a melhor decisão a ser tomada diante da situação.

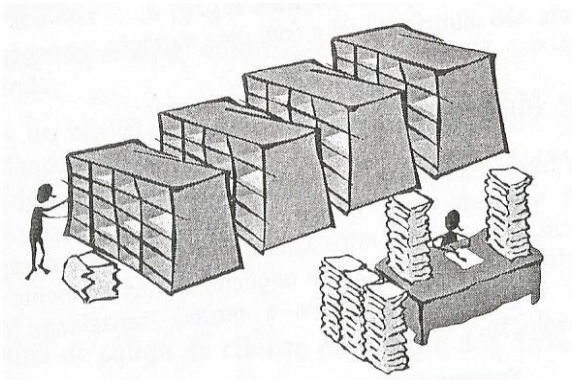
Segundo Nery (2006) uma análise de dados históricos pode nos apresentar indicadores de crescimento ou sinalizadores de perigo nos negócios.

2.3 *Data Warehouse*

Para análise de informações, uma das técnicas a serem utilizadas é o *Data Warehouse*. Pereira (2004, p.213) define *Data Warehouse* como:

> *Data Warehouse* (Literalmente quer dizer "Armazém de dados") pode ser definido como um conjunto de dados integrados, não-voláteis, mas que podem variar de tempos em tempos e orientados ao assunto. Esse conjunto de dados é utilizado com finalidade analítica e num processo de tomada de decisão de negócios, nos diversos níveis organizacionais de uma empresa. [...]

Figura 3 – Depósito de dados



Fonte: Nery, 2006, p.20.

Como pode ser observado na figura 3 acima, o objetivo de um *Data Warehouse* é armazenar as informações dentro de um banco de dados. Os dados devem estar disponíveis para, quando necessário, ser utilizados para análise e assim ajudar na tomada de decisão.

Segundo Nery (2006) a crescente utilização do *Data Warehouse* pelas empresas, se deve a crescente necessidade do domínio das informações estratégicas, levando em consideração a sua manipulação, pois assim garante respostas e ações rápidas, assegurando a competitividade de um mercado altamente disputado e mutável.

2.3.1 Características do *Data Warehouse*

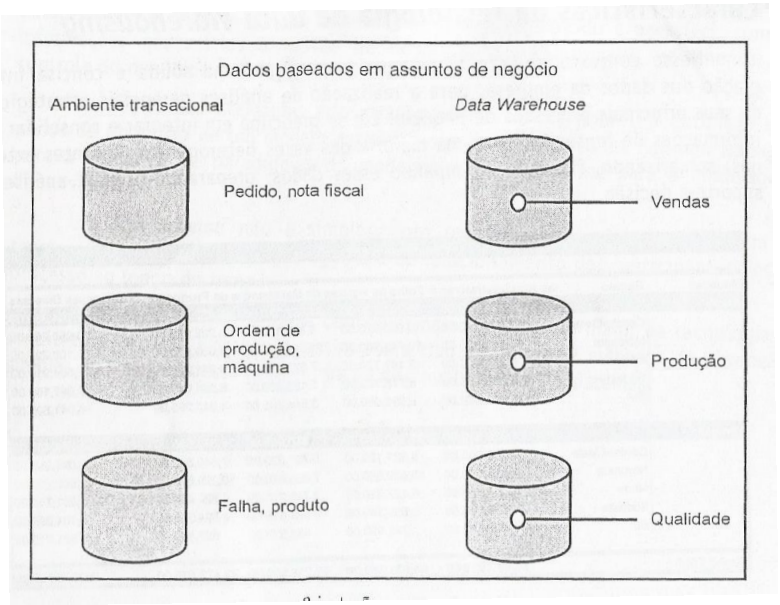
Para que o *Data Warehouse* possa atingir seus objetivos, é necessário que não apenas armazene os dados, mas também tenha o poder de integrar e consolidar as informações. Nery (2006, p.27) diz:

> O *Data Warehouse* proporciona uma sólida e concisa integração dos dados da empresa, para a realização de análises gerenciais estratégicas de seus principais processos de negócio. Ele se preocupa em integrar e consolidar as informações de fontes internas, na maioria das vezes heterogêneas, e fontes externas, sumarizando, filtrando e limpando esses dados, preparando-os para análise e suporte à decisão.

Conforme Colaço (2004) o *Data Warehouse* possui algumas características necessárias para que consiga ajudar o empresário na tomada de decisão. As características são:

2.3.1.1 Orientação por Assunto ou Tema

Figura 4 – Orientação por assunto ou tema



Fonte: Nery, 2006, p.28.

O *Data Warehouse* é orientado por assunto como pode ser visto na figura 4. Diferentemente de sistemas transacionais, um sistema orientado a assunto define um determinado negócio a ser trabalho no banco. Colaço (2004, p.16) diz:

> O *Data Warehouse* armazena informações necessárias para o processo de suporte a decisão. Essas informações são organizadas pelos temas importantes para o negócio da empresa. Em uma rede de restaurantes, por exemplo, os temas são: produtos, clientes, funcionários, etc.

A orientação por assunto leva em consideração os principais objetivos da empresa que por sua vez possuem diversos processos que inicialmente não foram criados para a tomada de decisão em si. Nery (2006, p.28) fala sobre os processos:

> É o processo que mostra desempenho e que possui indicadores de sua evolução. Eles podem e devem ser compreendidos e controlados para o sucesso e competitividade da organização. Esse controle é o principal objetivo dos sistemas de apoio a decisão.

Nem todos os processos de um sistema sustentam o *Data Warehouse*, muitos deles foram desenvolvidos apenas para manter as transações que são realizadas todos os dias. De acordo com Nery (2006) um desenvolvedor de *Data Warehouse* deve levar em consideração

apenas processos que são importantes para a tomada de decisão. Os processos mais abordados geralmente são os que possuem relações com atividades críticas.

2.3.1.2 Variação de Tempo

Os dados presentes em um banco de dados de um sistema transacional podem ser atualizados a qualquer momento. Ao existir um desconto ou acontecer uma crise, o valor do produto pode ser alterado no mercado, porém ao vender tal produto o que será levado em consideração é apenas o valor estipulado naquele momento, não importando se este estava mais caro ou mais barato no dia anterior.

Observe que a análise feita sobre o produto é apenas de qual é o seu valor momentâneo. Segundo Colaço (2004, p.17) afirma sobre variação de Tempo:

> Em um *Data Warehouse* os dados são carregados como fotos da base de dados operacional do momento, ou seja, cada ocorrência e cada mudança são consideradas como um novo registro. Os dados não são atualizados e podem ser comparados ao longo do tempo.

Em um *Data Warehouse* os dados adquiridos não devem ser substituídos e sim acrescentados de tal forma que sua identidade seja relacionada com a data e a hora em que aquele novo valor foi atribuído.

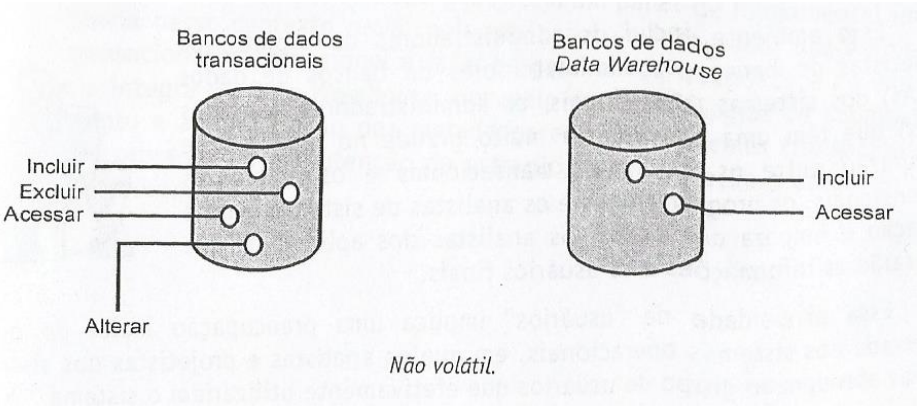
2.3.1.3 Não Volátil

Colaço (2004, p.17) define a não volatilidade do *Data Warehouse*:

> Teoricamente, depois que os dados estão no *Data Warehouse* (*DW*) não poderão ser atualizados ou alterados, apenas acessados. Os novos dados serão absorvidos, integrando-se com os dados existentes. O *Data Warehouse* permite apenas a carga inicial dos dados e a consulta aos mesmos. Contraditoriamente, existe no ambiente operacional uma grande volatilidade, visto que os dados são atualizados registro a registro a qualquer momento.

Os sistemas transacionais são baseados na volatilidade dos dados, contrapondo a idéia dos sistemas voltados a tomada de decisão. Os bancos de dados baseados em modelo de entidade relacional (MER) podem ter dados alterados, excluídos, inseridos ou consultados. Nos sistemas de tomada de decisão, como pode ser verificado na figura 5 abaixo, só possuem a opção de incluir e consultar dados.

Figura 5 – Não volatilidade de um *Data Warehouse*



Fonte: Nery, 2006, p.30.

2.3.1.4 Integração

Diante de um *Data Warehouse* pode ser analisado não apenas um banco de dados e seu histórico. Dependendo do tipo de análise, pode ser levado em consideração vários bancos de vários tipos de sistemas diferentes. O papel de integração nesta situação é muito importante, isso porque é necessário que a filtragem das informações de vários bancos, sejam concisas e consigam retirar o dado adequadamente.

Quando levamos em consideração dois bancos de dados diferentes que tratam uma coluna chamada "sexo" com divergência de valores entre os sistemas, como exemplo: Em um sistema o sexo é divido entre Homem e Mulher, e no outro como Masculino e Feminino. É de papel exclusivo da característica de integração conseguir manipular as informações de tal forma a adequar corretamente ao *Data Warehouse* quais são as informações que podem se referir a mesma informação.

Segundo Nery (2006, p.31) diz:

> Em ambientes de múltiplas plataformas sistêmicas, a característica de integração se torna fundamental, pois necessitamos de unicidade de informações. A existência de sistemas mais antigos com padrões de codificação de dados, leva à existência de diferentes padrões entre os sistemas operacionais, que quando da carga do *DW* são resolvidos pelos processos de filtragem e agregação.

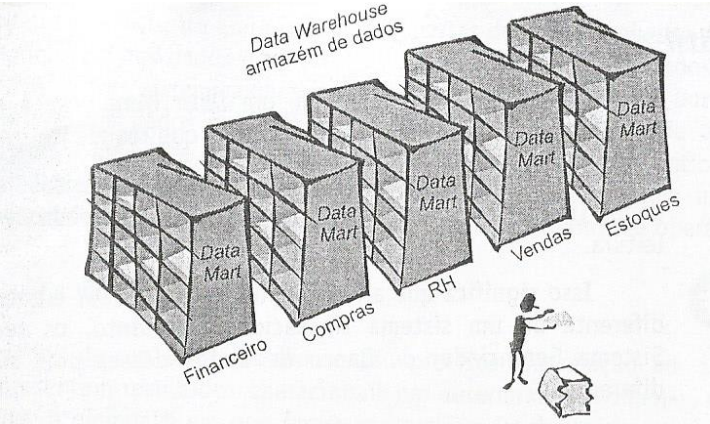
2.4 *Data Mart*

Um *Data Warehouse* guarda diversos tipos de informações requisitadas por um administrador. Dentro dele seria ideal que houvesse uma organização, até mesmo para facilitar a busca de dados.

Segundo Nery (2006) o *Data Warehouse* é constituído de vários subconjuntos de dados que por sua vez são chamados de *Data Mart*. Cada *Data Mart* é responsável por tratar um

assunto específico relacionado a alguma área ou até mesmo a um departamento de uma empresa como demonstrado na figura 6. Esses subconjuntos de dados facilita um rápido retorno de informações para a tomada de decisão, permitindo assim o usuário final de uma implantação do *Data Warehouse* verificar e avaliar os benefícios extraídos de seu investimento.

Figura 6 – *Data Warehouse* formado por vários *Data Marts*



Fonte: Nery, 2006, p.33.

2.5 Modelagem Multidimensional

Dentro do banco de dados, a primeira iniciação para o seu desenvolvimento geralmente é a criação do modelo de Entidade – Relacionamento (MER). Este tipo de modelo MER deve ser sempre normalizado. Amaral (2016, p.25) diz:

> O modelo relacional é baseado na álgebra relacional. No seu processo de modelagem, um banco de dados deve ser normalizado. A normalização permite que os dados sejam armazenados de forma consistente, reduzindo a redundância e garantido a integridade das informações.

Sabendo disso, o modelo relacional cumpre seu papel de gerenciar os dados de tal forma a garantir a integridade, a não redundância e sua consistência, e por esses motivos é percebido que tal modelo é utilizado até os dias de hoje.

Quando tratamos da modelagem multidimensional a ideia sobre MER não faz mais nenhum sentido. Colaço (2004, p.49) fala sobre este divergência:

> Na projeção de bases de dados para *Data Warehouse*, deve-se quebrar o paradigma da eliminação de redundâncias em um modelo de dados (normalização) e buscar armazenamento histórico. Desnormalizando algumas tabelas, o projetista do *DW* busca ganhar desempenho nas consultas.

O *Data Warehouse* não leva em consideração a manipulação dos dados dentro de um sistema transacional, porém ele se baseia nas informações históricas que este banco pode passar. Quanto mais dados a organização tiver, melhor será a aplicação do *Data Warehouse*. As

análises levantadas pelo *Data Warehouse* podem prever e diminuir a complexidade de uma tomada de decisão.

Nery(2006, p.97) mostra os objetivos do *Data Warehouse*:

> O objetivo de um *Data Warehouse* é obter um grande depósito de informações para utilização em aplicações não transacionais, por isso o denominamos de sistema de apoio à decisão, o qual deve apresentar informações de séries históricas, que refletem a evolução de fatos do dia-a-dia de negócios de uma organização.

Sabendo da importância de negócios para um *Data Warehouse*, é possível verificar a independência entre um MER e um modelo multidimensional. O modelo multidimensional busca modelar os negócios importantes de uma empresa para a tomada de decisão, enquanto o MER busca modelar as funcionalidades de um sistema transacional.

Nery(2006, p.79) mostra a definição de modelagem multidimensional:

> A modelagem multidimensional é uma técnica de concepção e visualização de um modelo de dados de um conjunto de medidas que descrevem aspectos comuns de negócios. É utilizada especialmente para sumarizar e reestruturar dados e apresenta-los em visões que suportem a análise dos valores desses dados.

2.6 Modelo ou Esquema Estrela

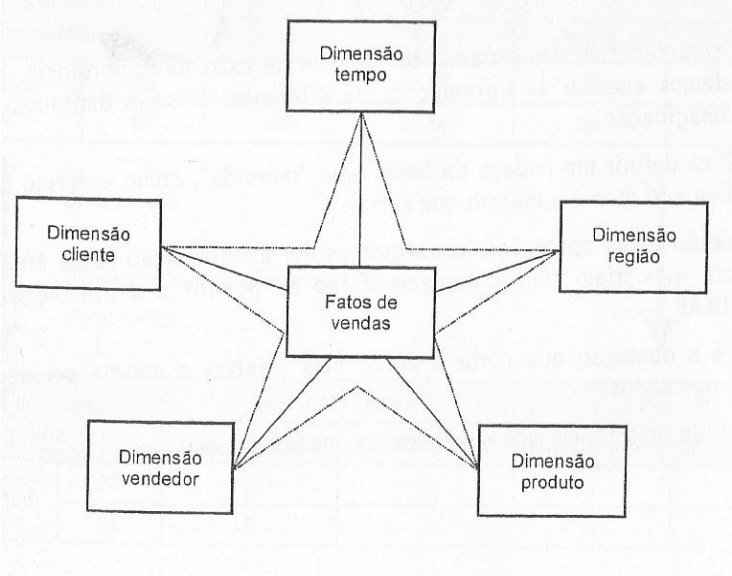
O modelo estrela é utilizado para estudar as necessidades levantadas pelo administrador. O administrador expõe as dificuldades da tomada de decisão de sua empresa, e diante das informações adquiridas é papel dos desenvolvedores do *Data Warehouse*, analisar e criar o modelo estrela. (NERY, 2006)

Todo modelo estrela é composto por três características que são chamadas de fatos, dimensões e medidas. As uniões destes três dados possibilitam o desenvolvimento e análise de negócio da empresa.

Colaço (2004, p.50) mostra a composição de um modelo estrela:

> O nome *Star Schema* foi adotado pela semelhança com uma estrela. Este esquema é composto de uma tabela dominante, chamada tabela de fatos, no centro, rodeado por tabelas auxiliares, chamadas de tabelas de dimensão. A tabela fatos conecta-se às demais por múltiplas junções e as tabelas de dimensões se conectam com apenas uma junção à tabela de fatos.

Figura 7 – Modelo ou esquema estrela



Fonte: Nery, 2006, p.93.

Com a figura 7 acima é possível verificar os componentes de uma estrela. Para o desenvolvimento de um *Data Warehouse*, o desenvolvimento de uma Estrela é o primeiro passo, é através dela que será definido quais serão os dados críticos analisados pelo administrador.

2.6.1 Fatos

Amaral (2016, p.42) mostra a definição e algumas características de um fato:

> O fato é a informação central, o tema ao qual se quer analisar. Um fato possui medidas que são valores a serem analisados e pré-calculados. Um fato também possui dimensões que são os diversos pontos de vista sobre o qual se quer analisar o fato.

Fato é o centro de uma estrela onde todas as informações adquiridas irão se movimentar em volta dele. É o fato quem define o centro do assunto discutido, é sobre ele que os negócios serão medidos e analisados, o modelo estrela sempre deve possuir um e somente um fato. Nery (2006, p.100) mostra outra característica de um fato:

;“Outra característica importante para identificar um fato é que ele é evolutivo, muda suas medidas com o tempo, podendo ser sempre questionado sobre essa evolução ao longo de um espaço de tempo.”

2.6.2 Dimensões

As dimensões giram em torno dos fatos, são eles quem definem o contexto em que aquele fato se encontra. Cada dimensão determina uma ponta da estrela, porém as quantidades

de pontas não são pré-determinadas. Pode existir estrelas que possuem 5 pontas (dimensões), ou possuir mais ou menos do que essa quantidade. (NERY, 2006)

2.6.2.1 Membros de Dimensões.

Dentro das dimensões existem alguns dados que podem ser utilizados para representar uma mesma dimensão, porém em proporções diferentes. Esses dados podem proporcionar informações mais precisas dependendo da forma que são manipulados.

Nery (2006, p.80) fala sobre os membros de uma dimensão:

> Um membro de dimensão é um nome diferente utilizado para determinar a posição de um item de dado. Por exemplo, todas as ocorrências de ano, trimestre e mês fazem a dimensão tempo, e todas as cidades, estados e regiões fazem a dimensão geográfica.

2.6.3 Medidas ou Métricas

Todo fato possui medidas que por sua vez são manipulados pelas dimensões, um fato que não possui medidas, não pode ser caracterizado como fato. As medidas podem variar, podem ser em função do tempo, de dinheiro, de porcentagem ou de qualquer outro número mutável.

Nery (2006, p.135) define medidas:

> A principal utilização de um *Data Mart* é para consultar dados históricos que estão normalmente sumarizados por períodos de tempo e as mais variadas combinações de uma informação. Mas geralmente o que se deseja ver são valores numéricos e sua evolução ou não, em um espaço de tempo, com cálculos de transformação desses dados. Esses valores numéricos são denominados de medidas ou métricas.

As medidas podem ser divididas em três tipos: Aditivos quando podem ser aplicados as operações de soma, subtração e média, como número de crime, número de acidentes. Semiaditivas quando as vezes pode ser aplicado algumas operações, e não-aditivos quando se trata de medidas que não podem ser manipuladas, como porcentagem, índice entre outros. (AMARAL, 2016)

2.7 Visão de um modelo multidimensional: Cubo

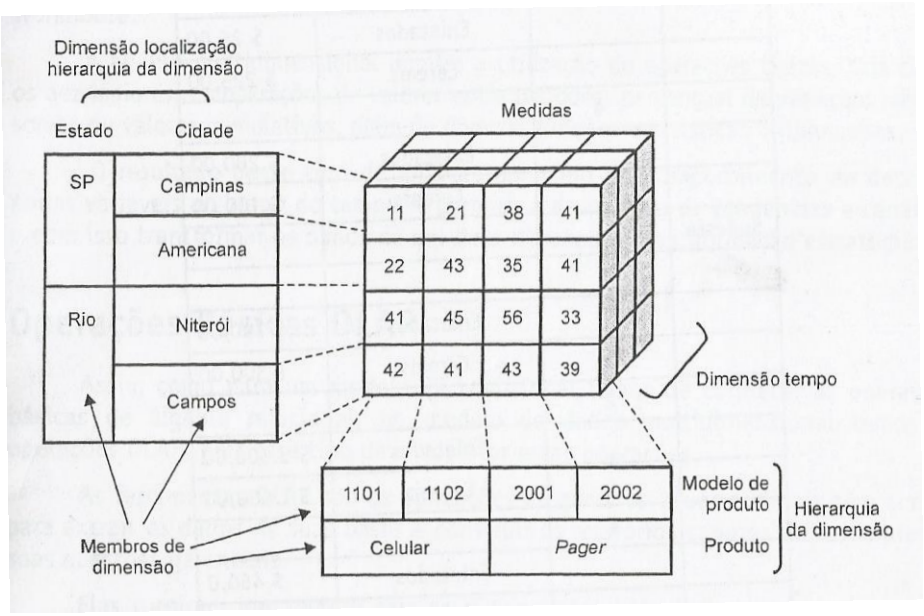
Depois de desenvolvido o modelo estrela, o Data Mart pode ser transformado em um CUBO. Nery (2006, p.53) mostra a representação de um modelo multidimensional como cubo:

> Nós podemos representar um modelo tridimensional por um cubo, entretanto usualmente um modelo dimensional consiste em mais de três dimensões, o que é definido como um hipercubo. Visualizar graficamente um hipercubo é muito difícil, desta forma utiliza-se a referência a cubo para qualquer modelo multidimensional.

Dentro do cubo é encaixado cada característica abordada de um modelo estrela. O fato representa o cubo em si. As dimensões, representam os lados de um cubo. Ao analisar duas

dimensões e suas intersecções o cubo é divido em subcubos que por sua vez possuem valores que são definidos como medidas. A ideia do cubo é a possibilidade de analisar o negócio abordado, de diversas formas diferentes e assim diante dos resultados expostos tomar uma decisão importante sobre um determinado problema. (NERY, 2006). Observe como exemplo de análise a figura 8 abaixo:

Figura 8 – Representação multidimensional com cubo



Fonte: Nery, 2006, p.82.

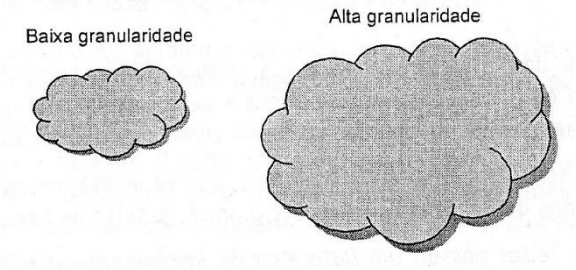
2.8 Granularidade no *Data Warehouse*

Diante de um projeto de *Data Warehouse*, a granularidade é o fator mais importante durante o planejamento, pois é o que torna esse sistema o diferencial para a tomada decisões. Nery (2006, p.59) conceitua a granularidade como:

> A granularidade de dados refere-se ao nível de sumarização dos elementos e de detalhe disponíveis nos dados, considerado o mais importante aspecto do projeto de um *Data Warehouse*. Quanto mais detalhe existir, mais baixo será o nível de granularidade. Quanto menos detalhe existir, mais alto será o nível de granularidade.

Em outras palavras, a granularidade representa o nível de abstração de informação, se a informação mostrada está generalizada no qual representa uma visão mais geral, sem muitos detalhes, ela possui granularidade alta, e por outro lado, se a informação apresenta detalhes mais específicos e em menor abstração, sua granularidade é baixa. Tal informação pode ser vista na figura 9 abaixo:

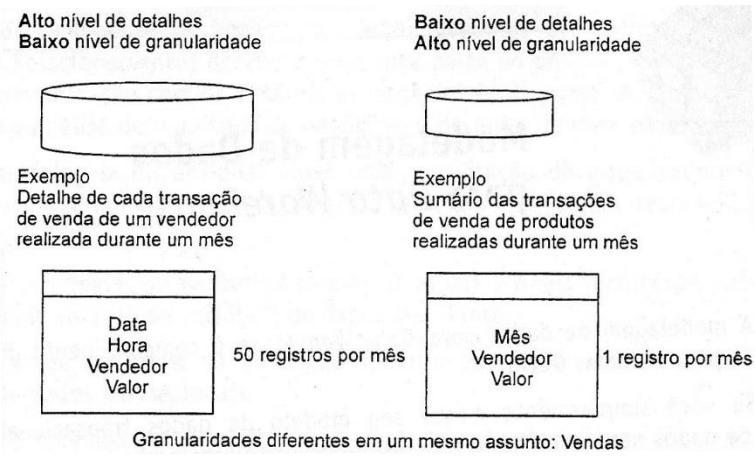
Figura 9 – Representação de granularidade



Fonte: Nery, 2006, p.59.

Na figura 10, Nery (2006) expõe um exemplo comum na aplicação da granularidade em projetos de *Data Warehouse*, no qual observa-se uma análise sobre a dimensão tempo em relação ao assunto vendas. Percebesse que ao tratar as informações de período em data e hora, os detalhes serão maiores, e terá uma quantidade maior de registros, e ao analisar as vendas em uma perspectiva mensal, será gerado apenas um registro sem muitos detalhes, porém com uma visão geral do que aconteceu no departamento de vendas durante esse mês.

Figura 10 – Aplicação da granularidade em projetos de *Data Warehouse*



Fonte: Nery, 2006, p.60.

Além do tempo, a granularidade também pode ser aplicada em diversas outras situações. Segundo Nery (2006, p.62):

"O mais importante da granularidade em um projeto é entendermos que ela não limita somente a tempo, mas a todos os fatores de classificação da informação que estiverem sendo utilizados".

Em projetos de *Data Warehouse*, por ser um sistema de apoio a decisões, é economicamente viável manter a granularidade em nível alto, pois de acordo com Nery (2006), com granularidade baixa seria como se fosse a realização de armazenamento histórico dos

sistemas transacionais, e para isso existem técnicas mais baratas como *backup* em fitas magnéticas, além de que obter uma perspectiva geral sobre determinado assunto é mais interessante para a tomada de decisões.

2.9 Cronograma e Implementação

Todo *Data Warehouse* precisa ter um cronograma de desenvolvimento e implementação. A necessidade de tal atividade é para fornecer documentação e melhor gerenciamento dos processos do *Data Warehouse*. Desta forma é possível acompanhar as atividades que devem ser realizadas durante todo o ciclo de vida do *DW*. Neste cronograma deve ser relatado desde o levantamento gerencial a tomada de decisão até o treinamento da utilização do produto final desta técnica. Segundo Colaço (2004, p.123)

> A implementação do *DW* deverá ser gradual e constituída por etapas consecutivas. Em cada fase deverão ser disponibilizadas as informações e os recursos destinados a apoiar as atividades de uma área alvo. O processo de implementação deverá contemplar a adequação do ambiente de informação convencional, com o intuito de: Incorporar os dados tratados pelos aplicativos departamentais; compatibilizar o significado de um mesmo dado quando tratado por mais de um aplicativo; disponibilizar os dados adicionais necessários para gerar certos tipos de informações previstas pelo ambiente de suporte à decisão.

2.10 Análises de Sistemas de Apoio a Decisão

Para uma implementação com excelência de um *Data Warehouse*, existe alguns processos de análises bem definidos a serem considerados, para conhecer bem o cenário e verificar a aplicabilidade dos dados que estão sendo tratados. Colaço (2004, p.102) cita os objetivos das Análises de Sistemas de Apoio a Decisão:

"Os objetivos da Análise de sistemas de apoio a decisão são: compreender quais dados são de interesse dos usuários finais, como extrair esses dados das bases operacionais e como disponibilizar essas informações para os usuários finais."

Ao detalharmos esse processo temos: análise de processo - define as etapas de carga e extração; análise de fonte de dados - mapeia os principais itens de interesse do usuário e também a forma de manipulação dos dados que serão transformados e limpados; análise de carga de dados - concentra-se em analisar como os dados serão carregados no *Data Warehouse*; análise de consulta de dados - preocupa-se como os usuários finais irão utilizar os dados gerados. (COLAÇO, 2004)

Como os dados dentro dos sistemas de apoio a decisão já estão modelados e bem apurados, suas consultas realizadas possibilitam determinados tipos de análises de usuário. Colaço (2004) diz que as análises podem ser: Análise estatísticas (cálculos, médias), Análise

Multivariável (comparações para observar padrões), Simulação e modelagem (validação de hipóteses) e Previsão (apoio à decisão para valores futuros).

2.11 Importância do Conhecimento do Volume de Dados do Sistema Transacional

Como preocupação de estruturar o ambiente que irá comportar um sistema de apoio a decisão, ter o conhecimento do volume de dados dos sistemas transacionais que estão sendo analisados, é de extrema importância para projetar tal ambiente. Colaço (2004, p.107) diz:

> As entrevistas também devem identificar os volumes de dados dos sistemas fontes para fins de projeção do espaço em disco que deverá existir no servidor do *Data Warehouse.* Os volumes devem ser obtidos a partir de dados do ambiente de produção e representar um dia ou mês aleatórios, podendo haver desvios (a maior ou a menor) em relação ao valor médio

2.12 Ambiente de Hardware e Software para um *Data Warehouse*

Quando se trata de ambiente de hardware para comportar um sistema de apoio a decisão como o *Data Warehouse*, é necessário seguir alguns requisitos mínimos para que sua implementação não ocorra futuros problemas. Colaço (2004) fala que o ambiente do *Data Warehouse* deve ser em um banco de dados separado fisicamente dos outros bancos do sistema transacional e possuir algumas características como escalabilidade e recursos para acesso de grandes volumes.

Devido à grande quantidade de informações presente no *Data Warehouse*, o tempo de resposta do processamento das consultas devem demorar, chegando até em minutos, o que depende da velocidade do canal de atendimento. E por esse motivo, o *Data Warehouse* precisa ficar em um ambiente fisicamente separado, para não prejudicar a operabilidade dos bancos de dados dos sistemas transacionais (COLAÇO 2004).

Além das especificações físicas, Colaço (2004, p.113) diz:

> Para a implementação de um *DW* será necessária a utilização de um conjunto mínimo de softwares, composto por um Sistema Gerenciador de Banco de Dados, software para geração do modelo multidimensional (pode estar acoplado à ferramenta de acesso ou não) e ferramenta de acesso para os usuários finais.

Na primeira fase de implementação do *Data Warehouse*, é importante um ambiente de testes, para que os desenvolvedores envolvidos realizem operações para a modelagem do sistema sem afetar o desempenho do ambiente em produção. Caso seja necessário, é possível criar um ambiente dedicado ao *Data Warehouse* apenas para consulta e manipulações básicas (COLAÇO, 2004).

2.12.1 SGBD (Sistemas de Gerenciamento de Banco de dados)

Os Sistemas de Gerenciamento de Banco de Dados(SGBD), segundo Date (2004), são *softwares* que trata todo o acesso ao banco de dados, como por exemplo, recebe e interpreta requisições de pedidos de usuário utilizando uma sub linguagem de dados, no qual geralmente é o *SQL* (*Structured Query Language*).

Para projetos de *Data Warehouse*, além da utilização do SGBD para interpretar operações de banco de dados, é comum a utilização da linguagem de programação *SQL* como meio de requisições. Date (2004, p.71) diz:

"SQL é a linguagem padrão para se lidar com bancos de dados relacionais, e é aceita por quase todos os produtos existentes no mercado. "

2.12.2 PostgreSQL

Dentre os SGBD's, destaca-se o PostgreSQL. Segundo a empresa mantenedora (POSTGRESQL, 2018c):

"O PostgreSQL é um poderoso sistema de banco de dados objeto-relacional de código aberto com mais de 30 anos de desenvolvimento ativo que lhe garantiu uma forte reputação de confiabilidade, robustez de recursos e desempenho."

Dentre as funções da linguagem *SQL* para a realização de conexões entre banco de dados, foi considerado o "dblink". Segundo a mantenedora (POSTGRESQL, 2018b):

"O dblink é um módulo que suporta conexões com outros bancos de dados do PostgreSQL de dentro de uma sessão de banco de dados."

2.12.3 *pgAdmin*4

Para manipular e gerenciar os dados de banco, foi necessário a utilização de uma ferramenta de gerenciamento. Para a utilização do banco de dados PostgreSQL, a fabricante também disponibiliza um software para esta manipulação. Segundo a documentação do POSTGRESQL (POSTGRESQL, 2018a):

> O *pgAdmin* é a principal ferramenta de gerenciamento de código aberto do Postgres. O *pgAdmin* 4 foi projetado para atender às necessidades dos usuários novatos e experientes do Postgres, fornecendo uma poderosa interface gráfica que simplifica a criação, a manutenção e o uso de objetos de banco de dados.

2.13 A Arquitetura do *Data Warehouse*

A Arquitetura do *Data Warehouse* é uma etapa que pode precisar ser cuidadosamente escolhida. O motivo para isso é porque a escolha é influenciada por diversas variáveis. Essas variáveis são descritas por Nery (2006, p.47)

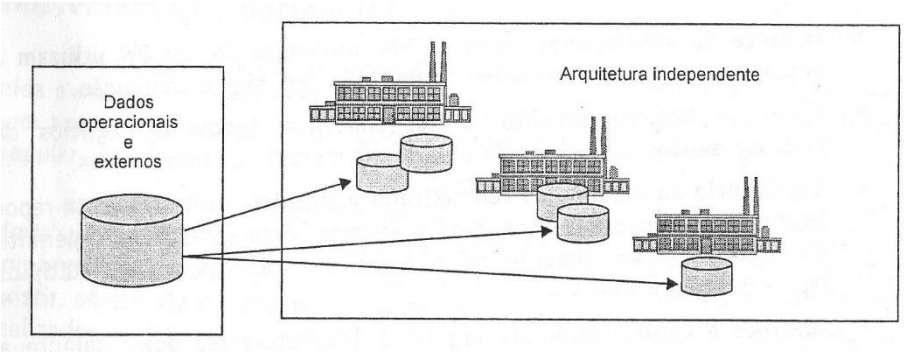
> A escolha da arquitetura é uma decisão gerencial do projeto, e está normalmente baseada nos fatores relativos à infraestrutura disponível, ao ambiente de negócios (porte da empresa), concomitantemente com o escopo de abrangência desejado, assim como a capacitação dos empregados da empresa e dos recursos disponibilizados ou projetados para investimento

Acontece que uma arquitetura mal planejada pode causar dramáticos impactos sobre o sucesso de um *Data Warehouse*. Segundo Nery (2006) muitas variáveis afetam diretamente a escolha de implementação da arquitetura do *Data Warehouse*, como exemplo pode ser citado o tempo para a execução do projeto, o retorno do investimento, a velocidade dos benefícios para a empresa ao ser utilizado as informações e consequentemente a satisfação do administrador e usuário das informações obtidas.

2.13.1 Arquitetura Independente

A arquitetura independente é uma arquitetura de um *Data Mart* que não influencie outro *Data Warehouse*. São geralmente arquiteturas que levam em consideração apenas uma necessidade de um departamento específico não se relacionando com outras áreas da empresa. Nery (2006) diz que a arquitetura independente implica em *Data Marts Stand Alone* controlados por um grupo específico de usuários e que atende somente as suas necessidades especificas e departamentais, sem foco corporativo nenhum. A figura abaixo mostra que um *Data Warehouse* com arquitetura independente não se relaciona com outras empresas, ou seja, os dados partem do banco para empresa como pode ser observado na figura 11 abaixo.

Figura 11 – Arquitetura independente



Fonte: Nery, 2006, p.53.

Essa arquitetura se caracteriza pela entrega do produto final do *Data Mart* em menor tempo e menor impacto sobre os recursos de tecnologia de informação, portanto não permite nenhuma visão global e também não aceita qualquer integração corporativa (Nery, 2006).

2.14 Modelo de Implementação do *Data Warehouse*

O escolha do modelo de implementação assim como a arquitetura do *Data Warehouse* leva em consideração o ambiente e as variáveis presentes. Nery (2006, p.52) afirma:

> A opção por um tipo de abordagem de implementação é influenciada por fatores como a infraestrutura de Tecnologia da Informação, a arquitetura escolhida, o escopo da implementação, os recursos disponíveis e principalmente pela necessidade ou não de acesso corporativo dos dados, assim como pelo retorno de investimento desejado e velocidade de implementação.

2.14.1 Implementação *Bottom Up*

A Implementação *Bottom Up* leva em consideração o desenvolvimento de *Data Marts* para depois o desenvolvimento de *Data Warehouse*. Segundo Nery (2006, p.54):

> Esse tipo de implementação permite que o planejamento e o desenho dos *Data Marts* possam ser realizados sem esperar que seja definida uma infraestrutura corporativa para *Data Warehouse* na empresa. Essa infraestrutura não deixará de existir, só que ela poderá ser implementada incrementalmente conforme forem sendo realizados os *Data Marts*

O primeiro processo nessa implementação é a utilização da extração, transformação e a integração dos dados para um *Data Mart.* Nery (2006) diz que a prática desta implementação possibilita um desenvolvimento mais rápido e consequentemente respostas mais rápidas, ele também permite a extração de informações mais relevantes por se tratar de um problema específico assim como a facilidade de implementação.

Segundo Nery (2006) esta implementação possui desvantagens, podendo causar conflitos entre os padrões estabelecidos por cada *Data Mart* levando assim a uma dificuldade quanto a integração de novos *Data Marts* criados.

2.15 *ETL*

O processo *ETL* (*Extract*, *Transformation & Load*), no português Extração, transformação e Carregamento se trata da etapa onde os dados são traduzidos de um sistema transacional para serem inseridos dentro do *Data Warehouse*. Segundo Nery (2006, P.35), ele conceitua o processo *ETL* como:

> A extração, organização e integração dos dados devem ser realizadas com o propósito de garantir a consistência e integridade das informações, construindo desta forma uma base de dados de alta qualidade e confiabilidade, que retrate efetivamente a realidade de negócios da empresa.

Na criação de um *Data Warehouse*, estimasse que parte do tempo gasto no desenvolvimento se passa na *ETL* e é comum que este ocupe 80% do tempo de toda implementação. Sabendo disso, tal procedimento se distribui em 3 etapas principais, sendo eles a extração, organização (transformação) e integração (carregamento) (INMON, 1997).

2.15.1 Extração

Fase em que as informações importantes são retiradas dos bancos de dados transacionais de acordo com a modelagem do *Data Warehouse* e inseridas na *Staging Area* (ALMEIDA, 2006 apud ABREU, 2007). Na primeira implementação, verifica-se uma carga, geralmente grande, de dados a serem extraídos com o objetivo de popular o histórico das informações que estão sendo tratadas, e no decorrer do tempo essa extração é periódica (KIMBALL, 1998 apud ABREU, 2007).

2.15.2 Organização

Também conhecido como transformação, essa etapa acontece após a extração, e tem como finalidade transformar, traduzir e limpar os dados que estão sendo tratados, de forma que a informação não perca sua integridade (GONÇALVES, 2003 apud ABREU, 2007).

Segundo Gonçalves (2003) é comum em banco de dados transacionais a utilização de valores booleanos ou iniciais para definir uma informação, como por exemplo tratar gênero como 0 e 1 ou M e F. Outras situações de transformação e padronização de valores como centímetros ou metros, gramas ou quilos também são observados. Essas informações precisam estar bem definidas e traduzidas ao serem inseridas no *Data Warehouse*.

2.15.3 Integração

Essa etapa também conhecida como carregamento ou carga, de acordo com Almeida (2006), é a fase em que os dados provenientes da transformação dentro da *Staging Area*, são inseridas no *Data Warehouse*. É a etapa, que em relação as demais, utiliza mais processamento, pois exige desempenho do banco para inúmeras inserções. Essas cargas são feitas periodicamente, de acordo com a modelagem planejada do *Data Warehouse*.

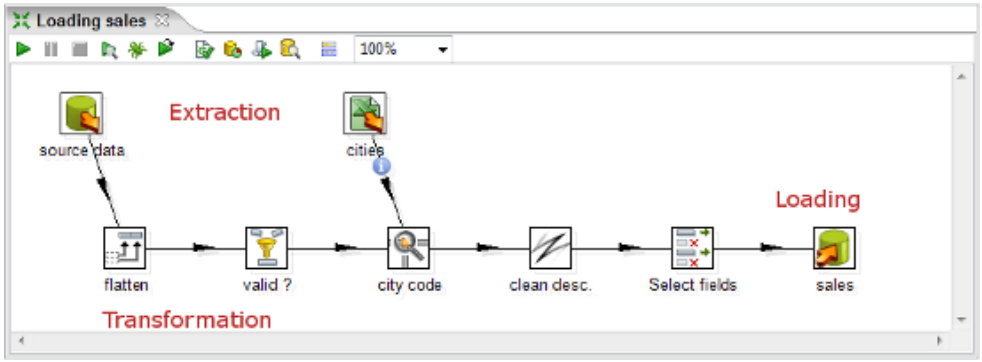
2.16 Ferramentas *ETL*

Como mencionado, o processo *ETL* é uma etapa duradoura quando se trata de desenvolvimento de um *Data Warehouse*, por isso, há no mercado diversas ferramentas que prometem facilitar este processo.

2.16.1 *Pentaho Data Integration*

Pentaho Data Integration (PDI) desenvolvido em 2004 pela *Pentaho Corporation* e mantida pela *Hitachi Insight Group* desde 2015. A PDI oferece usabilidade intuitiva, bibliotecas avançadas de componentes que ajudam na transformação de dados, integrações que suporta qualquer tipo de fonte de dados e oferece funcionalidades que ultrapassam os conceitos de *ETL* como por exemplo analisar dados, realizar *backups*, suporte a *Big Data* entre outros.

Figura 12 – Processo *ETL* do Pentaho



Fonte: PENTAHO (2011)

A figura 12 refere-se ao “Spoon”, uma ferramenta do Pentaho responsável por mapear o fluxo do processo *ETL*, com ajuda de componentes para facilitar modelagem e visualização. Outras ferramentas pertencentes ao *Pentaho Data Integration* como Pan e Kitchen, são responsáveis por executar o Spoon, realizando as transformações e funcionalidades previamente mapeadas (PENTAHO, 2011).

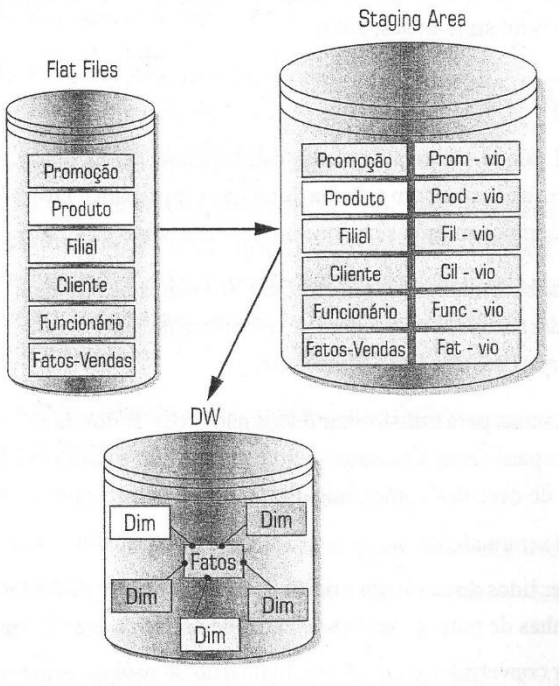
2.17 ***Operation Data Storage* ou *Staging Area – ODS***

Como primeiro passo para a manipulação de dados para o *Data Warehouse* será necessário a utilização da *Staging área* que permite a transformação dos dados em informações úteis a serem utilizadas no *Data Warehouse*. Segundo Nery (2006, p.42) *Staging Area* é:

> Um ambiente intermediário de armazenamento e processamento dos dados oriundos de aplicações OLTP e outras fontes, para o processo de extração e transformação e carga (*ETL*), possibilitando o seu tratamento, e permitindo sua posterior integração em formato e no tempo, evitando problemas após a criação do *Data Warehouse* e a concorrência com o ambiente transacional no consumo de recursos.

A *Staging Area* pode ser chamada também de *Operation Data Storage* (*ODS*). Esta área é a intermediária entre os sistemas transacionais e o *Data Warehouse*, e permite que apenas dados específicos sejam armazenados. É neste ambiente que o processo da *ETL* será aplicado. A figura 13 abaixo permite uma melhor visualização do que seria a *ODS*.

Figura 13 – Intermediadora *Staging Area*



Fonte: Nery, 2006, p.60.

O *ODS* possibilita a manipulação não só de um sistema transacional, levando em consideração diversos outros setores que por sua vez podem possuir sistemas diferentes e consequentemente dados heterogêneos (dados que demonstram os mesmos resultados, porém de forma diferente). O fato de manipular as informações de diversos sistemas transacionais, pode consequentemente trazer transtornos a empresa, pois a extração de muitos dados pode gerar um processamento adicional do computador e assim a inutilização deste durante o processo. A *Staging Area* sabe tratar este problema, extraindo as informações em horários marcados e por este motivo que se torna viável a utilização junto ao *Data Warehouse*. Nery (2006, p.38) diz:

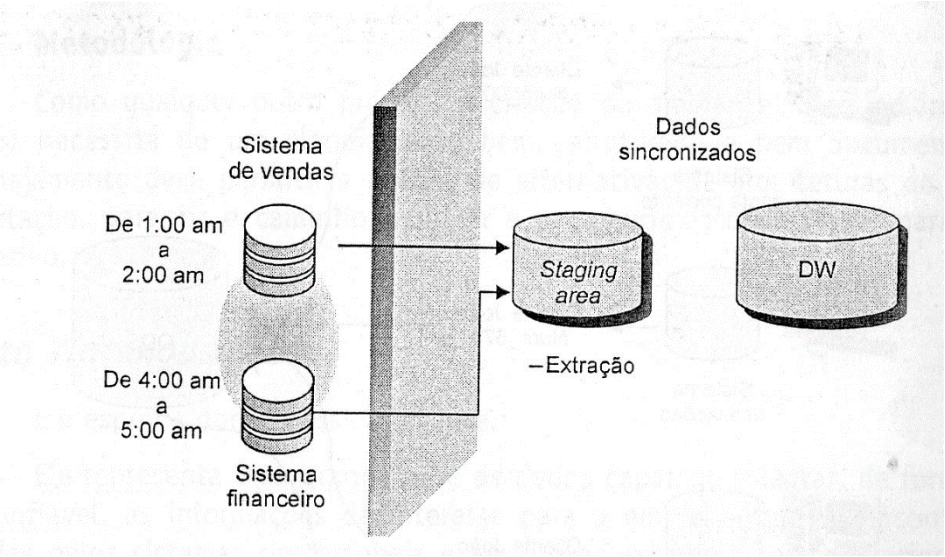
"Uma *Staging Area* permitirá ao administrador do *Data Warehouse* extrair os dados no momento em que estão disponíveis e posteriormente integrá-los. Isso facilita as extrações dos sistemas operacionais durante períodos fora de pico de operações."

Nery (2006, p.38) exemplifica essa situação:

> Por exemplo, os dados de vendas podem estar disponíveis para extração apenas entre 1:00 a.m. e 2:00 a.m. (o processamento das vendas é terminado e os dados estão em um estado estável e sincronizado), enquanto os dados financeiros estão disponíveis apenas entre 4:00 a.m. e 5:00 a.m.

2.18 Sistema de carga

Figura 14 – Sistema automatizado de carga



Fonte: Colaço, 2004, p.120.

O sistema de carga é responsável por analisar todo o processo de inserção de dados no *Data Warehouse*. A figura 14 acima mostra que o sistema de carga é uma operação automatizado e que por si mesma gera informações referentes ao processo de carga no sistema. Segundo Methanias Colaço (2004, p.120) diz:

> A operação de um Sistema de Carga consiste basicamente na montagem de uma Agenda de Carga, Liberação da Agenda e Monitoração da Carga. Periodicamente (diariamente, mensalmente, etc.) os arquivos (Flat Files) contendo as informações dos sistemas legados são enviados para o servidor de *Data Warehouse*. A agenda deve ser montada para a data à qual se referem os arquivos a serem carregados.

Através de um sistema de carga é possível verificar qual a situação dos carregamentos realizados no *Data Warehouse*. Desta forma é possível analisar se o processo de carga está funcionando corretamente ou se está tendo algum impedimento para a realização de suas atividades.

2.18.1 Ferramenta de Carga - *pgAgent*

Como sequência de configuração de um *Data Warehouse*, é necessário a criação de procedimentos periódicos para a realização da *ETL* automatizada, em períodos previamente declarados no planejamento especificados na reunião com o cliente. Para isso, existem algumas ferramentas para a realização desse procedimento automatizado, sendo uma delas o *pgAgent*. Segundo POSTGRESQL (POSTGRESQL, 2018d):

“O *pgAgent* é um agente de agendamento de tarefas para bancos de dados Postgres, capaz de executar scripts em lotes ou shell de várias etapas e tarefas *SQL* em agendas complexas.”

Segundo a documentação do POSTGRESQL (POSTGRESQL, 2018d) o *pgAgent* é um serviço do *pgAdmin* no qual executa *Jobs* (ações), que se dividem por *steps* (etapas), sendo para cada etapa um script *SQL* a ser executado de ordem configurada e de acordo com uma *shedule* (cronograma). Sendo assim, é possível realizar a configuração no *pgAgent* no qual é responsável por gerar as *procedures* para disparar as ações configuradas de acordo com um período especificado.

3 DESENVOLVIMENTO

Para execução deste TCC foi criado uma relação com a empresa onBlox, empresa responsável pelo desenvolvimento de Sistemas Logísticos. Seus sistemas trabalham junto a centros de distribuições que por sinal necessitam de diversos tipos de relatórios que são desenvolvidos pela empresa.

O sistema da empresa a ser abordando nesta pesquisa, se trata do *Warehouse Managment System* (WMS), no português "Sistema de gerenciamento de Armazém".

Entre os itens que os sistemas de WMS tratam, estão principalmente: endereçamento físico de produtos no armazém, recebimento (entrada) e expedição (saída) de produtos no estoque, tarefas para usuários, inventários (balanço) e etc. Através disso, sistemas de WMS trouxeram maior facilidade no controle para armazéns, de forma que todo procedimento ali realizado, gera dados no sistema.

Diante da proposta deste trabalho, a empresa apoiou o projeto tendo em vista a aplicação do *Data Warehouse* como suporte a tomada de decisão de seus clientes. Uma vez acordado o sistema para tal empresa, foi feita uma nova reunião junto ao consultor logístico da onBlox, onde foi requerido para o trabalho um centro de distribuição e uma de suas principais necessidades junto a tomada de decisão. A conversa foi gravada e anexada a este trabalho em forma de áudio (APÊNDICE A).

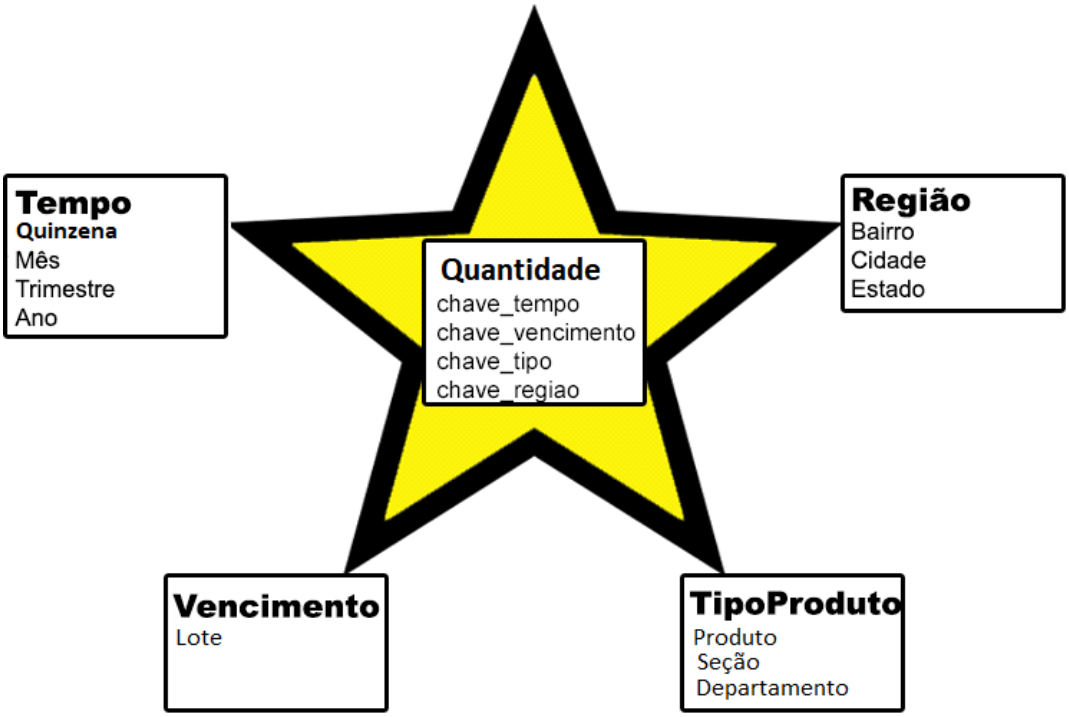
A principal necessidade levantada foi referente a dificuldade que os administradores de um centro de distribuição possuem para verificar quais são os produtos que mais saem durante um determinado período, levando em consideração o seu vencimento.

Diante do problema, foi definido a arquitetura e o modo de implementação que o *Data Warehouse* deveria seguir, levando em consideração as necessidades do cliente. O tempo para desenvolvimento do *data Warehouse* foi então definido em 6 meses aproximadamente. Diante do que foi exposto pelo cliente, foi possível verificar que o problema se relaciona apenas a quantidade de produto que sai e entra no centro de distribuição.

As características definidas pelo cliente definem exatamente a idéia de arquitetura independente com a implementação *BottomUp*. São técnicas que possuem características de desenvolvimento ágil e que podem ser utilizadas para a resolução de um problema específico. Tais características ignoram qualquer influência de departamentos externos ou qualquer interferência de outra empresa sobre *Data Mart* desenvolvido.

Definido a arquitetura e realizado algumas análises, foi desenvolvido a etapa de modelagem de um *Data Mart*, que por sua vez é uma característica da implementação *BottomUp*. A primeira modelagem, levou em consideração o modelo estrela que tem como dimensões definidas junto ao consultor as seguintes entidades: Região, Vencimento, Tempo, tipo de produto, como demonstrado na figura 15.

Figura 15 – Modelo ou esquema estrela do estudo de caso

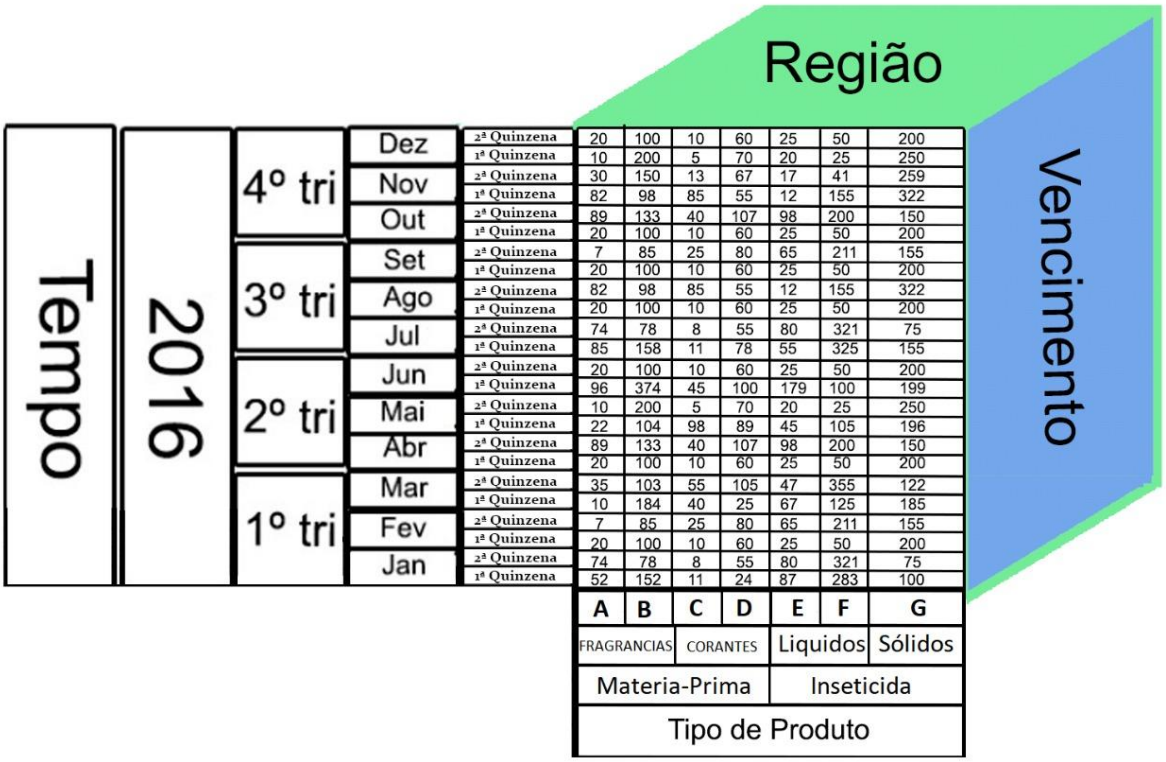


Fonte: RIBEIRO; SENA. 2017

Observando a necessidade do cliente foi definido algumas hierarquias ou granularidades que podem ser vistas dentro das dimensões, conforme demonstrado na figura 15. A partir do modelo estrela, foi realizado alguns testes para a modelagem do cubo.

O teste foi realizado através de cenários. Estes cenários são situações criadas com o intuito de provar se as dimensões do cubo, atingem as expectativas geradas pelo cliente. As expectativas foram levadas em consideração segundo as informações adquiridas da reunião com o cliente, que foi gravada e está como APÊNDICE A. Alguns exemplos destes cenários seria: Quantos produtos foram perdidos devido ao vencimento no mês de janeiro? Quais são os produtos que mais saem no meio do ano? Qual região que mais adquire mercadorias no ano? Entre outras. Observando estes cenários é possível verificar que existe um relacionamento entre as dimensões do modelo estrela, tais relacionamentos permitiu uma maior compreensão sobre o desenvolvimento do cubo, O cubo gerado pode ser observado na figura 16.

Figura 16 – Representação multidimensional com cubo sobre o estudo de caso



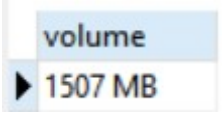
Fonte: RIBEIRO; SENA. 2017

Dentro do cubo foi especificado cada dimensão e suas hierarquias respectivamente. Foi levado em consideração que a quantidade de produtos representaria o fato do *Data* Mart. Com o cubo desenvolvido, a capacidade de análise para o desenvolvimento do *Data Mart* se tornou mais fácil.

Para continuidade, foi disponibilizado um *backup* do banco de dados do centro de distribuição em estudo. Com o intuito de analisar os dados do sistema transacional para mensurar o impacto do desenvolvimento do *Data Warehouse*.

Como exemplo de tal analise, foi analisado no banco transacional qual seria o seu tamanho real. O resultado da pesquisa, verificou que o tamanho utilizado é de 1507MB. Saber o tamanho do banco, possibilita uma melhor analise do impacto do *Data Warehouse* sobre a eficiência do sistema transacional. A figura 17 comprova o tamanho do banco mencionado.

Figura 17 – Volume do sistema transacional



Fonte: RIBEIRO; SENA. 2018

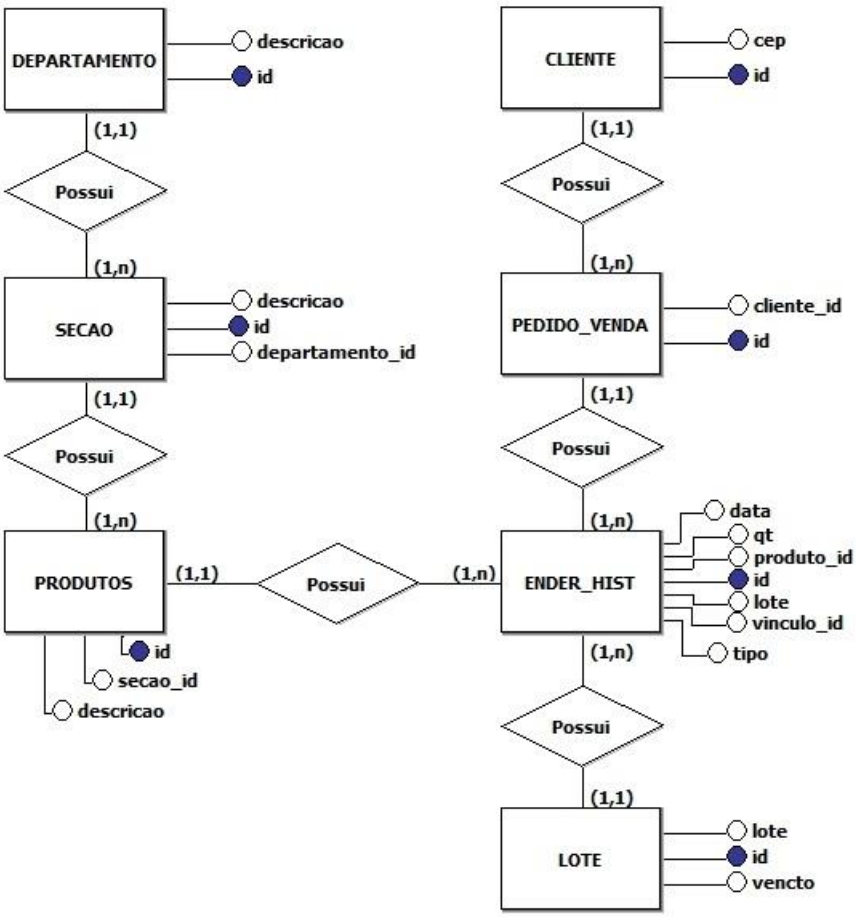
O próximo passo foi a análise do *backup* realizado, para definir quais dados do banco transacional deveriam ser incluídos no *Data Warehouse*. O intuito desta análise é identificar diante do banco cada uma das dimensões desenvolvidas e assim as suas hierarquias ou granularidades. Uma vez identificado, os processos de *ETL* poderiam ser realizados.

O processo de *ETL* garante que todos os dados mencionados no cubo e no modelo estrela possam ser carregados no *Data Warehouse*. Para o auxílio de tal procedimento foi utilizado algumas ferramentas como: a linguagem *SQL*, SGBD *PostgreSQL*, *pgAdmin*, *DbLink* e *pgAgent* que serão mencionados suas aplicações mais a frente.

O processo de *ETL* depende da *Staging Area* para o seu processo de manipulação e transformação de dados. Por enquanto foi definido como a *Staging Area* uma máquina da empresa Onblox, e que caso fosse necessário, poderia futuramente ser migrado para um servidor do centro de distribuição, ficando assim a escolha do cliente.

Conforme definido anteriormente, todo o desenvolvimento de um *Data Mart* tem como foco principal suas dimensões, granularidades, e seu fato. Sabendo disso, para uma melhor compreensão sobre a implementação dos processos de *ETL*, foi desenvolvido um MER parcial do sistema transacional. Esse MER não utiliza todas as tabelas e informações contidas no banco, pelo contrário, busca demonstrar apenas as informações que serão utilizadas em cada dimensão, conforme figura 18.

Figura 18 – Representação do MER do banco em estudo



Fonte: RIBEIRO; SENA. 2018

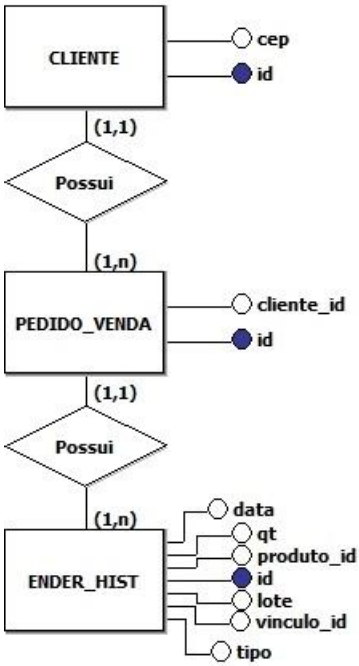
O processo de implementação do *Data Warehouse*, iniciou-se com o desenvolvimento de cada dimensão separadamente. As separações das dimensões possibilitou melhor análise das informações necessárias para cada granularidade assim como maior facilidade do desenvolvimento dos processos de *ETL*. Para ajuda entender os processos de *ETL* e as análises realizadas, o MER da figura 18 foi divido por dimensões e será demonstrado a seguir as suas particularidades diante de suas implementações

Como todo *Data Mart* necessita de um fato, o MER acima possui uma tabela que fornece os dados do fato. O fato tratado no *Data Mart* em estudo é a quantidade de produto, desta forma a tabela responsável por passar este dado é a entidade ENDER_HIST. Diante disso, todas as dimensões que foram criadas, consideraram a utilização desta tabela.

A dimensão região, possui como granularidade os atributos estado, cidade e bairro. As tabelas utilizadas do MER da figura 18, podem ser observados na figura 19. No banco transacional, não possui as informações da granularidade desta dimensão, porém na tabela

cliente existe o atributo CEP que com a utilização da linguagem *SQL* possibilita o processo de manipulação da *ETL*.

Figura 19 – MER do sistema transacional utilizado para a dimensão região



Fonte: RIBEIRO; SENA. 2018

Para resolver o problema foi adquirido através do Correios uma base de dados com informações de CEP e seus respectivos endereços. Dessa forma, dentro da *Staging Area* foi feito uma junção dos dados transacionais disponibilizado com a base de dados dos Correios.

No código de *SQL* abaixo (Código 1), foi feito uma relação entre a entidade ENDER_HIST e a tabela CLIENTE levando como intermediador a entidade PEDIDO_VENDA.

Código 1 – Código da dimensão Região

```
select
SUM(a.qt) qt_saida,
(CASE WHEN g.bairro is null then 'Não informado' WHEN g.bairro is not null then
g.bairro END)as bairro,
(CASE WHEN g.cidade is null then 'Não informado' WHEN g.cidade is not null then
g.cidade END)as cidade,
(CASE WHEN g.uf is null then 'Não informado' WHEN g.uf is not null then g.uf
END)as estadofrom ender_hist as a
FULL OUTER JOIN pedido_venda e on a.vinculo_id = e.id
FULL OUTER JOIN clientes f on e.cliente_id = f.id
FULL OUTER JOIN cep g on f.cep::int = g.cep::int
where  a.tipo = 'SE'
```

```
group by g.bairro, g.cidade, estado
```

Fonte: RIBEIRO; SENA. 2018

O resultado da compilação deste código é demonstrado na figura 20.

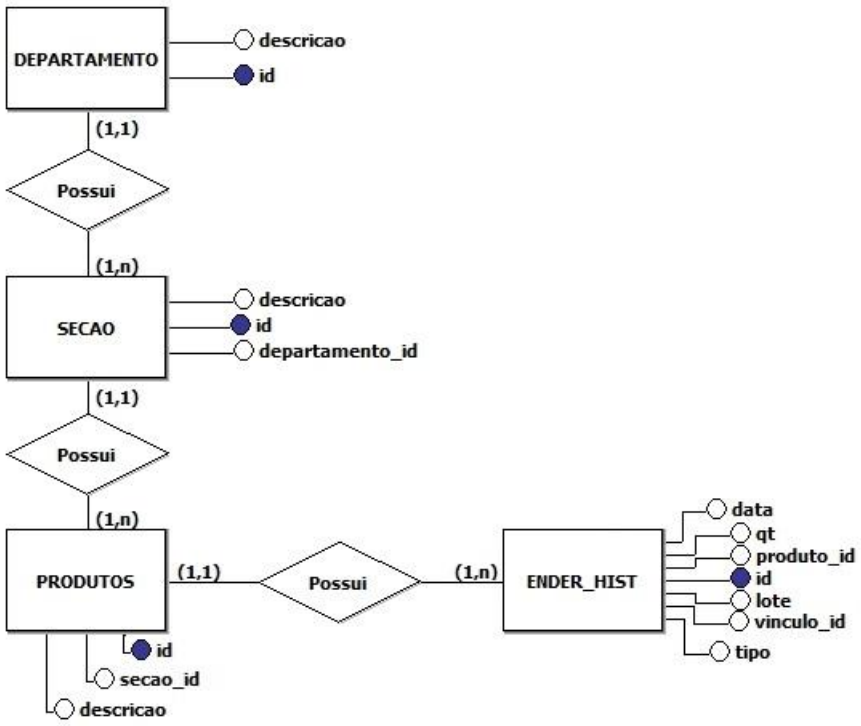
Figura 20 – Resultado da execução do código *SQL* referente a dimensão região

qt_saida	bairro	cidade	estado
3	Abolião	Rio de Janeiro	RJ
208	Aeroviário	Goiânia	GO
105	Alagadio	Fortaleza	CE
268	Alagoinhas Velha	Alagoinhas	BA
43	Alameda	Várzea Grande	MT
112	Aldeota	Fortaleza	CE
990	Alecrim	Natal	RN
30	Alphaville Empresarial	Barueri	SP
307	Alto Alegre	Cascavel	PR

Fonte: RIBEIRO; SENA. 2018

As granularidades que constituem a dimensão Tipo de Produto são departamento, seção e produto. As entidades retiradas do MER para a análise desta dimensão podem ser visualizadas na figura 21. Nesta dimensão não houve a necessidade de realizar nenhum processo de transformação da *ETL*. Na base de dados transacional, os dados já se encontram com as granularidades necessárias, conforme figura 21.

Figura 21 – MER do sistema transacional utilizado para a dimensão tipo de produto



Fonte: RIBEIRO; SENA. 2018

ao execução do código *SQL* (Código 2) resulta as informações desta dimensão, a entidade ENDER_HIST, onde se encontra a quantidade do produto, é relacionada com as entidades PRODUTOS, SECAO e DEPARTAMENTO, levando em consideração a descrição de cada entidade.

Código 2 – Código da dimensão do tipo de produto

```
select d.nome as produto,
(CASE WHEN i.descricao is null then 'Não informado' WHEN i.descricao is not null
then i.descricao END) as secao,
(CASE WHEN j.descricao is null then 'Não informado' WHEN j.descricao is not null
then j.descricao END) as departamento,
(CASE WHEN c.qt_entrada is null then 0 WHEN c.qt_entrada is not null then
c.qt_entrada END) as qt_entrada,
SUM(a.qt) qt_saida
from ender_hist a
full outer join
(
select
a.produto_id,
SUM(a.qt) qt_entrada
from ender_hist as a
where a.tipo = 'ER'
group by a.produto_id
) as c
on a.produto_id = c.produto_id
full outer join produtos d on a.produto_id = d.id
FULL OUTER JOIN secao as i on d.secao_id = i.id
FULL OUTER JOIN departamento as j on d.departamento_id = j.id
where  a.tipo = 'SE'
group by d.nome, secao, departamento,c.qt_entrada
```

Fonte: RIBEIRO; SENA. 2018

Como resultado da consulta 2, temos os dados demonstrados na figura 22

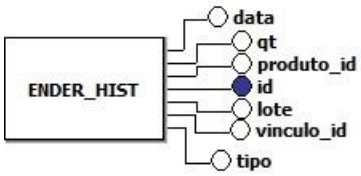
Figura 22 – Resultado da execução do código *SQL* referente a dimensão tipo de produto

produto	secao	departamento	qt_entrada	qt_saida
AKB 480 SACHE 1X100X10ML	LIQUIDOS	INSETICIDA	496	452
AKB GLIFOSATO 480 SACHE 24X5X10ML	HOUSE GARDEN	INSETICIDA	882	436
BARAKELL- PRO GEL 24X30GRS	LIQUIDOS	INSETICIDA	114	75
CUPINICIDA 12X1LT	LIQUIDOS	INSETICIDA	163	123
ESPANTA POMBO E MORCEGO 200GR (BISNAGA)	HOUSE GARDEN	INSETICIDA	354	298

Fonte: RIBEIRO; SENA. 2018

A dimensão tempo possui as granularidades de ano, trimestre e mês. Para a construção desta dimensão necessitou-se apenas da entidade ENDER_HIST e seus atributos conforme demonstrado na figura 23.

Figura 23 – MER do sistema transacional utilizado para a dimensão tempo



Fonte: RIBEIRO; SENA. 2018

A tabela ENDER_HIST possui um atributo a data e através do código 3(este atributo foi utilizado para obter as informações referentes a granularidade ano e mês. O trimestre por sua vez foi construído na junção de 3 meses do ano e nomeado como 1º Trimestre e assim sucessivamente..

Código 3 – Código da dimensão tempo

```
select
            c.lote,
            c.qt_entrada,
            SUM(a.qt) qt_saida,
            (
                        CASE
                              when (to_char(a.data,'DD')::int > 0)  AND
(to_char(a.data,'DD')::int <= 15) THEN '1ª QUINZENA'
                              when (to_char(a.data,'DD')::int >= 16)  AND
(to_char(a.data,'DD')::int <= 31) THEN '2ª QUINZENA'
                        END
            ) as quinzena,
            to_char(a.data,'TMMONTH') as mes,
            to_char(a.data,'MM') as numeromes,
            (
                        CASE
                              when (to_char(a.data,'MM')::int >=1) AND
(to_char(a.data,'MM')::int <=3) THEN '1º TRIMESTRE'
                              when (to_char(a.data,'MM')::int >=4) AND
(to_char(a.data,'MM')::int <=6) THEN '2º TRIMESTRE'
                              when (to_char(a.data,'MM')::int >=7) AND
(to_char(a.data,'MM')::int <=9) THEN '3º TRIMESTRE'
                              when (to_char(a.data,'MM')::int >=10) AND
(to_char(a.data,'MM')::int <=12) THEN '4º TRIMESTRE'
```

```
                        END
            ) as trimestre,
            to_char(a.data,'YYYY') as ano
from ender_hist as a
full outer join
(
select
b.lote,
SUM(a.qt) qt_entrada
from ender_hist as a
full outer join lote as b
on a.lote = b.lote
where a.tipo = 'ER'
group by b.lote
) as c
on a.lote = c.lote
where  a.tipo = 'SE'
--and a.lote = '4004'
group by quinzena,trimestre, mes, ano, c.qt_entrada, c.lote,numeromes
order by ano, numeromes,lote, quinzena
```

Fonte: RIBEIRO; SENA. 2018

A figura 24 é o resultado da execução do código 3.

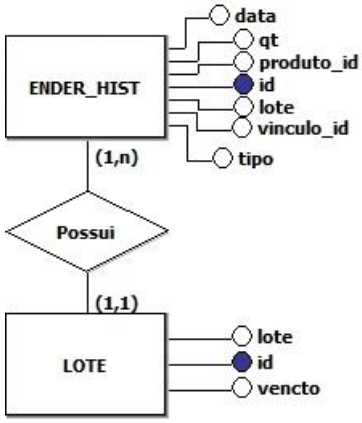
Figura 24 – Resultado da execução do código *SQL* referente a dimensão tempo

lote	qt_entrada	qt_saida	quinzena	mes	numeromes	trimestre	ano
4004	950	143	2ª QUINZENA	DEZEMBRO	12	4º TRIMESTRE	2015
4004	950	1	1ª QUINZENA	JANEIRO	01	1º TRIMESTRE	2016
4004	950	81	2ª QUINZENA	JANEIRO	01	1º TRIMESTRE	2016
4004	950	249	1ª QUINZENA	FEVEREIRO	02	1º TRIMESTRE	2016
4004	950	412	2ª QUINZENA	FEVEREIRO	02	1º TRIMESTRE	2016
4004	950	52	1ª QUINZENA	MARÇO	03	1º TRIMESTRE	2016
4004	950	1	2ª QUINZENA	MARÇO	03	1º TRIMESTRE	2016
4004	950	2	1ª QUINZENA	ABRIL	04	2º TRIMESTRE	2016

Fonte: RIBEIRO; SENA. 2018

A dimensão vencimento por sua vez possui apenas uma granularidade chamada de lote. Diante do MER da figura 18 as entidades a serem utilizadas nesta dimensão são as entidades presentes na figura 25. A entidade lote possui o atributo vencto que determina o vencimento de um lote específico, por isso, todos os produtos que estiverem dolote vencerão na mesma data.

Figura 25 – MER do sistema transacional utilizado para a dimensão vencimento



Fonte: RIBEIRO; SENA. 2018

O código *SQL* a seguir (Código 4) é a construção da relação entre a tabela ENDER_HIST e a tabela LOTE. O código identifica os atributos de quantidade de produto, vencimento, e o número do lote.

Código 4 – Código da dimensão vencimento

```
select
a.lote,
to_char(a.vencto, 'DD/MM/YYYY') as Vencimento,
(CASE WHEN c.qt_entrada is null then 0 WHEN c.qt_entrada is not null then
c.qt_entrada END) as qt_entrada,
SUM(a.qt) qt_saida
from ender_hist as a
full outer join lote as b on a.lote = b.lote
full outer join
(
select
b.lote,
SUM(a.qt) qt_entrada
from ender_hist as a
full outer join lote as b
on a.lote = b.lote
where a.tipo = 'ER'
group by b.lote
order by b.lote
) as c
on a.lote = c.lote
where  a.tipo = 'SE'
```

```
group by a.lote, a.vencto,c.qt_entrada
```

Fonte: RIBEIRO; SENA. 2018

Ao executar o código é mostrado uma planilha semelhante a figura 26.

Figura 26 – Resultado da execução do código *SQL* referente a dimensão vencimento

lote	vencimento	qt_entrada	qt_saida
4034	08/12/2017	251	251
4035	08/12/2017	251	251
4053	08/12/2017	116	116
4274	07/03/2018	250	250
4301	09/03/2018	250	250
4352	28/03/2018	501	501
4359	31/03/2018	296	296
4360	30/03/2018	250	250
4361	31/03/2018	249	249

Fonte: RIBEIRO; SENA. 2018

Para considerar a quantidade de entrada e saída como fato do *Data Warehouse*, foi necessário analisar algumas siglas utilizados pelo banco de dados do sistema transacional. A tabela ENDER_HIST é responsável por armazenar todos os dados de movimentação de produtos no centro de distribuição, e afim de selecionar apenas os tipos de entrada e saída de mercadoria, foi identificado que o banco transacional possui um atributo chamado “tipo” que permite fazer a distinção dos variados tipos de movimentações. De acordo com informações fornecidas pelo cliente, foram considerados os registros do tipo “SE” (Saída Expedição) como quantidade de saída, e registros do tipo “ER” (Entrada Recepção) como quantidade de entrada.

Diante de todas as granularidades obtidas e suas dimensões o próximo passo é a construção do *Data Warehouse*. Uma tabela com campos para todos os dados adquiridos sendo criada pela junção de todos os códigos *SQL* desenvolvidos para cada dimensão. Para a criação do *Data Warehouse*, foi considerado a criação de uma tabela contendo todas as informações das dimensões, da menor até a maior granularidade esclarecida para esse trabalho.

Sabendo disso, tal tabela contêm os seguintes atributos: produto, secao, departamento, lote, vencimento, qt_entrada, qt_saida, quinzena, mes, trimestre, ano, bairro, cidade e estado. O *script* da realização da criação da tabela pode ser visualizado no código 5.

Código 5 – Código da criação da tabela do *Data Warehouse*

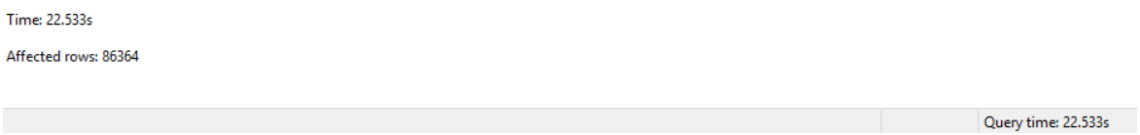
```
create table data_warehouse (
produto varchar(100),
secao varchar(50),
```

```
departamento varchar(50),
lote varchar(20),
vencimento date,
qt_entrada integer,
qt_saida integer,
quinzena varchar(20),
mes varchar(20),
trimestre varchar(20),
ano integer,
bairro varchar(50),
cidade varchar(50),
estado varchar (20)
)
```

Fonte: RIBEIRO; SENA. 2018

Para a realização da primeira carga, na qual se caracteriza em popular todo o histórico do sistema transacional no *Data Warehouse*, foi realizado uma inserção manual através de linguagem *SQL*. Tal procedimento foi executado em 22 segundos, e um total de 86.364 mil registros foram inseridos no *Data Warehouse*. A figura 27 detalha as informações de execução deste processo.

Figura 27 – Quantidade de registros inseridos no *Data Warehouse*



Fonte: RIBEIRO; SENA. 2018

Dessa maneira, foi criado o *Data Warehouse* na menor granularidade em relação a todas as dimensões específicadas neste trabalho. Os relatórios podem ser apresentados de acordo com o agrupamento dessas informações em relação a perspectiva desejada. A figura 28apresenta algumas das informações que foram inseridas no *Data Warehouse*:

Figura 28 – *Data Warehouse*

produto	secao	departamento	lote	vencimento	qt_entrada	qt_saida	quinzena	mes	trimestre	ano	bairro	cidade	estado
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	2	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Centro	Imperatriz	MA
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	1	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Cidade Jardim	Campinas	SP
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	1	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Jardim Alvorada	Barretos	SP
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	8	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Jardim Nova São C	São Carlos	SP
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	5	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Setor Aeroporto	Luziânia	GO
KELLMAT GRANULAD	GRANULADOS	INSETICIDA	5235	20/01/2019	149	3	2ª QUINZENA	MARÇO	1º TRIMESTRE	2017	Vila Progresso	Campo Grande	MS

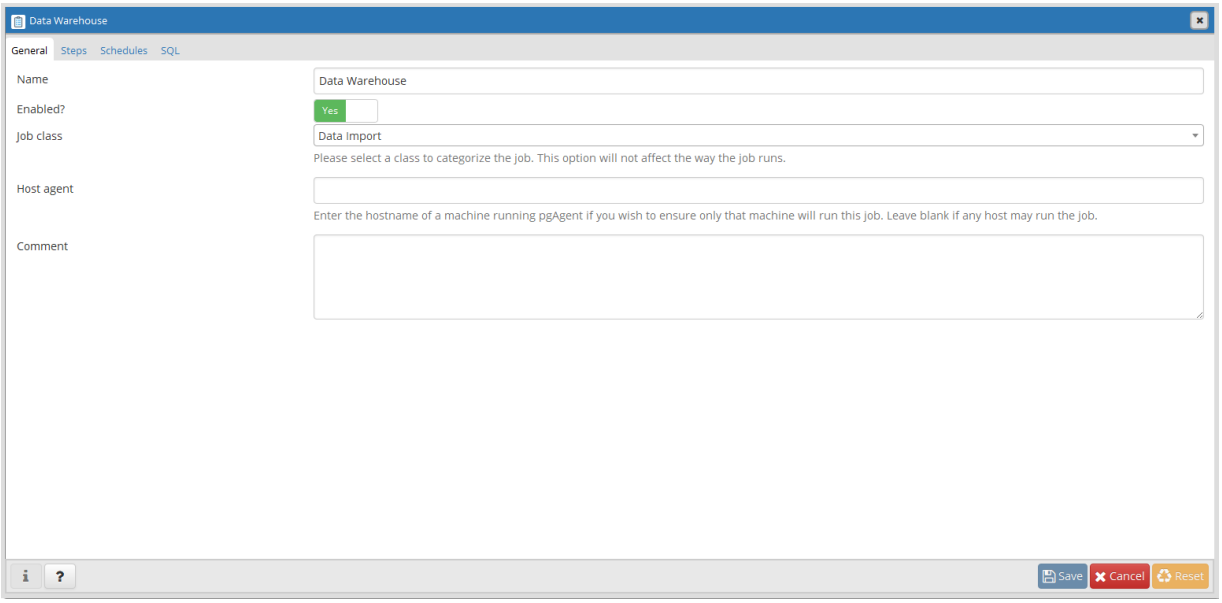
Fonte: RIBEIRO; SENA. 2018

Ao analisarmos as informações, podemos visualizar uma entrada fixa para um determinado lote de um produto, e suas saídas em relação a menor granularidade das outras dimensões. Ou seja, verificando somente a perspectiva “Estado” da granularidade de Região ignorando os “Bairros” e “Cidades”, por exemplo, as linhas do relatório serão reduzidas devido a essa informação estar mais generalizada.

Para a realização das cargas periódicas neste projeto de *Data Warehouse*, foi utilizado um serviço do *pgAdmin*4. *PgAgent*, é o serviço responsável por gerenciar ações, controlar e executar o cronograma de carga. Para a realização da configuração do serviço, é necessário seguir 3 etapas principais.

A primeira etapa, são as informações gerais relevantes e informativas referente ao que se trata, como por exemplo o nome do serviço, tipo, comentários. A figura 29 demonstra os detalhes dessas informações presentes nessa etapa.

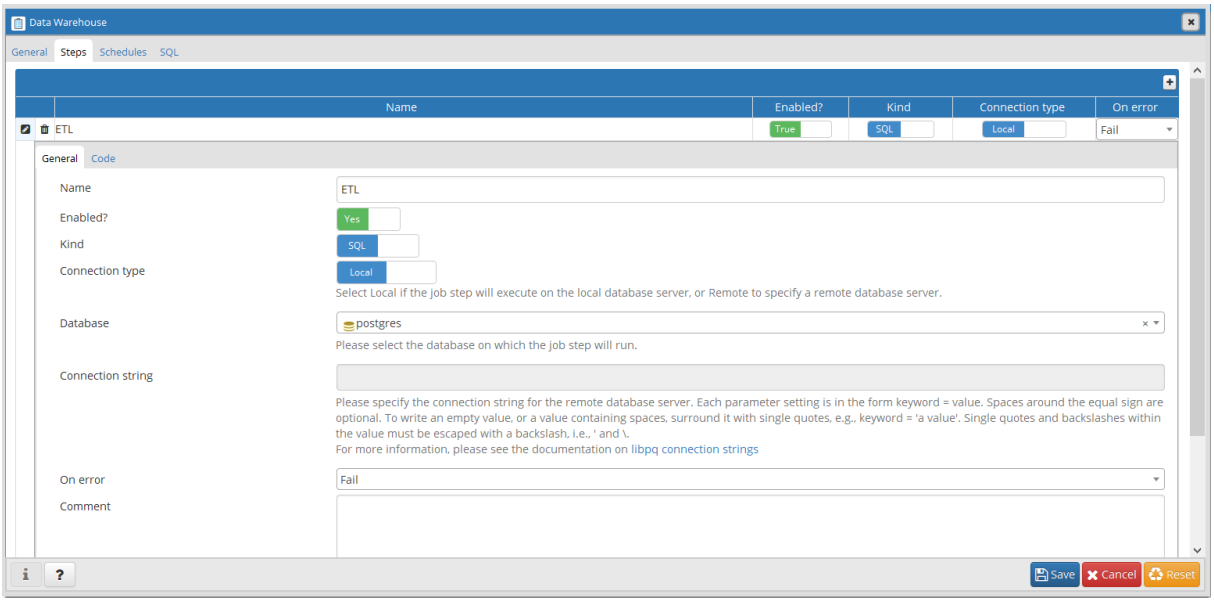
Figura 29 – Primeira etapa de configuração do *pgAgent*



Fonte: RIBEIRO; SENA. 2018

A segunda etapa se trata em especificar ao *pgAgent*, o que deverá ser feito. Por se tratar de uma linguagem a ser compilada pelo SGBD, essa etapa deve ser configurada e inserida com um ou mais comandos *SQL*. Neste projeto de *Data Warehouse*, foi inserido apenas uma ação a ser executada, no qual se trata da automatização do processo *ETL*. Na figura 30 observa-se a configuração referente a segunda etapa da criação do serviço, no qual é necessário adicionar informações gerais e o comando *SQL*.

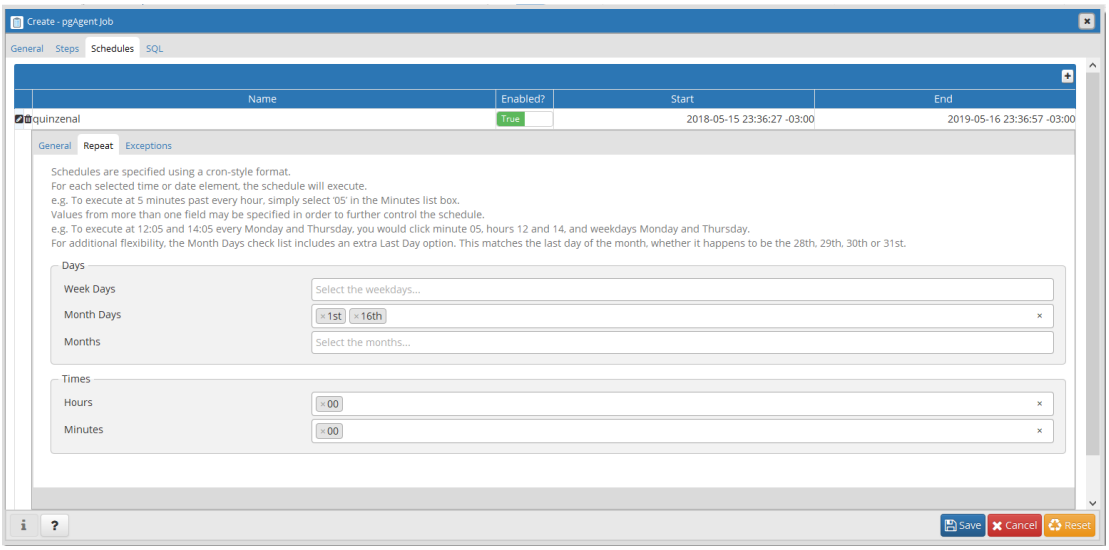
Figura 30 – Segunda etapa de configuração do *pgAgent*



Fonte: RIBEIRO; SENA. 2018

Por fim, a etapa da criação do serviço, é definir o cronograma de execução da ação previamente configurada. Nesse projeto de *Data Warehouse* o script *SQL foi agendado* em relação a menor granularidade da dimensão tempo. A menor granularidade se trata de quinzenas, ou seja, os comandos *serão executados* todo dia 01 e 16 de cada mês. Para que as informações estejam consolidadas de acordo com essa granularidade. Para que a execução da procedure, código 6, o horário agendado não deve impactar o funcionamento do sistema transacional, sendo agendado para as 00:00hs nas datas pré- Na figura 31, pode-se observar a configuração desta etapa:

Figura 31 – Terceira etapa de configuração do *pgAgent*



Fonte: RIBEIRO; SENA. 2018

No *Data Warehouse* foram realizadas duas cargas quinzenais de acordo com o cronograma configurado. Tais cargas, foram executadas no dia 16/05/2018 e 01/06/2018 no horário agendado, com duração de 2 a 5 segundos aproximadamente e o sucesso como status de resposta, conformefigura 32, demonstra-se os logs de execução do serviço:

Figura 32 – Log de execução da carga periódica

Run	Status	Start time	Duration	End time
232	s	2018-06-01 00:00:04.946621-03	00:00:04.971681	2018-06-01 00:00:09.918302-03
220	s	2018-05-16 00:00:03.331562-03	00:00:02.012465	2018-05-16 00:00:05.344027-03

Fonte: RIBEIRO; SENA. 2018

Código 6 – Código do sistema de carga

```
DO $$
DECLARE
    jid integer;
    scid integer;
BEGIN
-- Creating a new job
INSERT INTO pgAgent.pga_job(
    jobjclid, jobname, jobdesc, jobhostagent, jobenabled
) VALUES (
    2::integer, 'Data Warehouse'::text, ''::text, ''::text, true
) RETURNING jobid INTO jid;
-- Steps
-- Inserting a step (jobid: NULL)
INSERT INTO pgAgent.pga_jobstep (
    jstjobid, jstname, jstenabled, jstkind,
    jstconnstr, jstdbname, jstonerror,
    jstcode, jstdesc
) VALUES (
    jid, 'carga quinzenal'::text, true, 's'::character(1),
    ''::text, 'postgres'::name, 'f'::character(1),
    'insert into data_warehouse
select *
from dblink
   (
    ''dbname=backup
     hostaddr=-
     user=-
     password=-
```

```
     port=5432'',

 ''select d.nome as produto,

(CASE WHEN i.descricao is null then ''''Não informado'''' WHEN i.descricao is not
null then i.descricao END) as secao,

(CASE WHEN j.descricao is null then ''''Não informado'''' WHEN j.descricao is not
null then j.descricao END) as departamento,

a.lote,

to_char(a.vencto, ''''DD/MM/YYYY'''') as Vencimento,

(CASE WHEN c.qt_entrada is null then 0 WHEN c.qt_entrada is not null then
c.qt_entrada END) as qt_entrada,

SUM(a.qt) qt_saida,

(            CASE

                   when (to_char(a.data,''''DD'''')::int > 0)  AND
(to_char(a.data,''''DD'''')::int <= 15) THEN ''''1ª QUINZENA''''

                   when (to_char(a.data,''''DD'''')::int >= 16)  AND
(to_char(a.data,''''DD'''')::int <= 31) THEN ''''2ª QUINZENA''''

             END

) as quinzena,

to_char(a.data,''''TMMONTH'''') as mes,

(

             CASE

                   when (to_char(a.data,''''MM'''')::int >=1) AND
(to_char(a.data,''''MM'''')::int <=3) THEN ''''1º TRIMESTRE''''

                   when (to_char(a.data,''''MM'''')::int >=4) AND
(to_char(a.data,''''MM'''')::int <=6) THEN ''''2º TRIMESTRE''''

                   when (to_char(a.data,''''MM'''')::int >=7) AND
(to_char(a.data,''''MM'''')::int <=9) THEN ''''3º TRIMESTRE''''

                   when (to_char(a.data,''''MM'''')::int >=10) AND
(to_char(a.data,''''MM'''')::int <=12) THEN ''''4º TRIMESTRE''''

             END

) as trimestre,

to_char(a.data,''''YYYY'''') as ano,

(CASE WHEN g.bairro is null then ''''Não informado'''' WHEN g.bairro is not null
then g.bairro END)as bairro,

(CASE WHEN g.cidade is null then ''''Não informado'''' WHEN g.cidade is not null
then g.cidade END)as cidade,

(CASE WHEN g.uf is null then ''''Não informado'''' WHEN g.uf is not null then g.uf
END)as estado

from ender_hist as a

full outer join lote as b on a.lote = b.lote

full outer join

(select

b.lote,
```

```
SUM(a.qt) qt_entrada
from ender_hist as a
full outer join lote as b
on a.lote = b.lote
where a.tipo = ''''ER''''
group by b.lote
order by b.lote
) as c
on a.lote = c.lote
full outer join produtos d on a.produto_id = d.id
FULL OUTER JOIN pedido_venda e on a.vinculo_id = e.id
FULL OUTER JOIN clientes f on e.cliente_id = f.id
FULL OUTER JOIN cep g on f.cep::int = g.cep::int
FULL OUTER JOIN secao as i on d.secao_id = i.id
FULL OUTER JOIN departamento as j on d.departamento_id = j.id



where  a.tipo = ''''SE''''
and a.dt_insert >= CURRENT_DATE - 15
group by d.nome, a.lote, a.vencto, b.data_fabricacao,c.qt_entrada,
quinzena,trimestre, mes, ano, g.bairro, g.cidade, estado,secao, departamento
order by ano ''
   ) as resultado(produto varchar,secao varchar, departamento varchar,lote
varchar,vencimento date,qt_entrada numeric,
                                                        qt_saida numeric,
quinzena varchar, mes varchar, trimestre varchar, ano integer, bairro varchar,
                                                        cidade varchar, estado
varchar);
'::text, ''::text
) ;
-- Inserting a schedule
INSERT INTO pgAgent.pga_schedule(
    jscjobid, jscname, jscdesc, jscenabled,
    jscstart, jscend,    jscminutes, jschours, jscweekdays, jscmonthdays,
jscmonths
) VALUES (
    jid, 'quinzenal'::text, 'Carga quinzenal Data Warehouse'::text, true,
    '2018-05-15 23:36:27-03'::timestamp with time zone, '2019-05-16 23:36:57-
03'::timestamp with time zone,
    -- Minutes
    ARRAY[true, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false,
```

```
false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false]::boolean[],

    -- Hours

    ARRAY[true, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false, false,
false, false, false]::boolean[],

    -- Week days

    ARRAY[false, false, false, false, false, false, false]::boolean[],

    -- Month days

    ARRAY[true, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, true, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false, false]::boolean[],

    -- Months

    ARRAY[false, false, false, false, false, false, false, false, false, false,
false, false]::boolean[]

) RETURNING jscid INTO scid;

END

$$;
```

Fonte: RIBEIRO; SENA. 2018

4 CONSIDERAÇÕES FINAIS

No início do projeto, foi definido com os gestores logísticos, os pontos mais importantes a serem abordados em um sistema de apoio a decisão. Diante disso foi analisado as deficiências logísticas de um centro de distribuição e quais informações seriam necessárias para a resolução de seus principais problemas.

Através das informações retiradas nas reuniões com o cliente, foi definido o modelo estrela e o cubo multidimensional, como forma de planejamento para o início da construção do *Data Warehouse*. Paralelamente, buscou-se utilizar boas práticas para a escolha da arquitetura e modelagem de implementação de sistemas de apoio a decisão.

Mesmo com o estudo de ferramentas automatizadas que abrange todo o processo de *ETL*, este trabalho focou em demonstrar a formação do *Data Warehouse* através da linguagem *SQL*, permitindo sua automatização apenas no processo de agendamento das cargas periódicas através de uma ferramenta presente no SGBD PostgreSQL.

Para a sua implementação técnica, foi avaliado uma dificuldade de bibliografias como base para o desenvolvimento, pois como se trata de uma ferramenta poderosa e bem específica de empresa a empresa. Não é comum que estejam disponíveis para estudo, no entanto, este baseou-se principalmente em questões teóricas do *Data Warehouse* e orientações de profissionais especializados no assunto.

Além das dificuldades em referenciais teóricos, a linguagem *SQL* e o entendimento do sistema da empresa, foram outros fatores que impactaram no desenvolvimento. Foi necessário um estudo aprofundado da regra de negócio para conseguir suprir as necessidades levantadas pelos gestores logístico, para que dessa forma, as informações geradas no *Data Warehouse* não houvessem problemas de integridade e inconsistência.

Por fim, a implementação do *Data Warehouse* em um banco de dados referente a operações de um centro de distribuição possibilita uma análise geral em relação a movimentação de produtos, sendo possível verificar picos de demanda e perdas de acordo com determinadas perspectivas (dimensões).

Dessa forma, proporciona aos gestores uma base histórica para auxiliar em cálculos de projeções de compras de seus produtos, de maneira que estes não faltem em estoque e também evite a compra excessiva no qual pode-se ter prejuízo como consequência, devido a estes terem a possibilidade de ultrapassarem da data de vencimento o que inviabiliza a sua comercialização.

Além das vantagens relacionado a tomada de decisões que o *Data Warehouse* proporciona, verificou-se também uma alta performance nas buscas destas informações na base de dados, isso acontece porque o *Data Warehouse* consolidou as informações do sistema transacional em um banco de dados separado esua manipulaçãonão interfere no banco transacional do cliente.

TRABALHOS FUTUROS

Para o constante incremento do *Data Warehouse*, como trabalhos futuros também se tem interesse na análise de novas situações a serem levantadas como problemas logísticos do centro de distribuição do cliente para a integração de novos *Data Marts.*

Tem-se também a possibilidade de criação e integração de um *BI*, no qual pretende disponibilizar ferramentas gráficas para facilitar as consultas nessa base de dados, aumentando a variedade de opções de visualização.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, Fábio S. G. G. ***Estudo de usabilidade do software:*** *Talend Open Studio como ferramenta padrão para ETL dos sistemas clientes da aplicação PostGeoOlap*. 2007. Monografia (Graduação em Sistemas de Informação) – Faculdade Salesiana Maria Auxiliadora, Macaé, 2007.

ALMEIDA, Alexandre M. ***Proposição de indicadores para avaliação técnica de projetos de Data Warehouse:*** *um estudo de caso no Data Warehouse da plataforma Lattes*. 2006. Monografia (PósGraduação em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

AMARAL, Fernando. **Introdução à Ciência de Dados:** Mineração de Dados e *Big Data*. 1ª Ed, Rio de Janeiro: Alta Books. 2016.

BALLOU, H. Ronald. **Logística Empresarial**. São Paulo: Atlas S. A., 2009

COLAÇO, Methanias J. **Projetando Sistemas de Apoio à Decisão Baseados em *Data Warehouse***. 1ª Ed. Rio de Janeiro: Axcel Books. 2004.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 8ª Ed., Rio de Janeiro: Campus, 2004.

GOMES, F. A. M.; GOMES, C. F. S.; ALMEIDA, A. T. de. **Tomada de Decisão Gerencial:** Enfoque Multicritério. 5ª ed. São Paulo: Atlas, 2014.

GONÇALVES, Marcio. **Extração de dados para Data Warehouse**. Rio de Janeiro: Axcel Books, 2003.

INMON, W. H.; HACKATHORN, Richard D. **Como Usar o Data Warehouse**. Rio de Janeiro: Infobook, 1997.

KIMBALL, Ralph. **Data Warehouse Toolkit**. São Paulo: Makron Books, 1998.

NERY, Felipe R. M. **Tecnologia e Projeto de Data Warehouse.** 2ª Ed. São Paulo: Érica. 2006.

PEREIRA, William A. **Fundamentos de Bancos de Dados**. 2ª Ed. São Paulo: Érica. 2004.

POSTGRESQL, ***pgAdmin 4****.* Disponível em: <https://www.*pgAdmin*.org/docs/*pgAdmin*4/dev/>. Acesso em: 28 de abril de 2018a.

POSTGRESQL, ***PostgreSQL 9.3.23 Documentation***. Disponível em: <https://www.postgresql.org/docs/9.3/static/dblink.html>. Acesso em: 28 de abril de 2018b.

POSTGRESQL, ***POSTGRESQL: THE WORLD'S MOST ADVANCED OPEN SOURCE RELATIONAL DATABASE***. Disponível em: <https://www.postgresql.org/>. Acesso em: 28 de abril de 2018c.

POSTGRESQL, ***Creating a pgAgent Job*** Disponível em: < https://www.*pgAdmin*.org/docs/*pgAdmin*4/dev/*pgAgent*_jobs.html>. Acesso em: 28 de abril de 2018d.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 6ª Ed. Rio de Janeiro: Elsevier. 2012.

TURBAN, Efraim; McLEAN, Ephraim; WETHERBE, James. **Tecnologia da Informação para Gestão:** Em Busca Do Melhor Desempenho Estratégico e Operacional. 8ªed. Porto Alegre: Bookman, 2013.

VERGARA, C. V. **Projetos e Relatórios de Pesquisa em Administração**. 14ª ed. São Paulo: Atlas S. A., 2013.

LISTA DE ANEXOS

LISTA DE APÊNDICES

ANEXO A

ANEXO A – Carta de Autorização

Carta de Autorização

Anápolis, 20 de Novembro de 2017

Autorizamos o Sr. André Ribeiro Costa e o Sr. Lucas Hananni de Melo Sena, a utilização do nome e dados estatísticos da empresa onBlox Software Logístico para serem utilizadas na pesquisa: Aplicação de *Data Warehouse* para o Gerenciamento Logístico de Centro de Distribuição.

Esta autorização está concedida aos pesquisadores, comprometendo-se os mesmos a utilizar os dados para fins científicos, garantindo a não utilização das informações em prejuízo da empresa.

Thiago Silva Cruz
onBlox Software Logístico