



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

CAIO BULHÕES SABINO

**O USO DE DATA WAREHOUSE NO AUXÍLIO DE GESTÃO COM
BUSINESS INTELLIGENCE**

**Assis/SP
2020**



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

CAIO BULHÕES SABINO

**O USO DE DATA WAREHOUSE NO AUXÍLIO DE GESTÃO COM
BUSINESS INTELLIGENCE**

TCC apresentado ao Curso de Bacharelado em Ciência da
Computação do Instituto Municipal de Ensino
Superior de Assis, como requisito do Curso de
Graduação

Aluno: Caio Bulhões Sabino

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Área de Concentração: Informática

**Assis/SP
2020**

AGRADECIMENTOS

Agradeço a Deus, pois sem ele eu não teria a força de vontade para me dedicar e passar noites e noites estudando para completar o curso.

A minha mãe e irmã, que são exemplos de vida, fonte de inspiração e que sempre me apoiaram nos dias mais complicados.

A minha querida e amada namorada, presente em quase todos os momentos da minha vida, sem ela seria difícil passar por tantos obstáculos.

A todos os professores do Curso de Ciência da Computação da Fundação Educacional do Município de Assis (FEMA), visto que não faltaram esforços no compartilhar o melhor de seus conhecimentos e experiências.

Em especial ao Professor Dr. Alex Sandro Romeo de Souza Poletto pela gentileza em aceitar ser orientador do presente estudo.

Aos colegas, pelos momentos que juntos vivenciamos, alguns contatos permanecerão, outros quem sabe, se perderão pelo caminho, mas, foi muito bom este tempo.

RESUMO

As empresas que auxiliam o gerenciamento das atividades do negócio, têm alguns problemas na agilidade das informações para auxiliar os coordenadores em processos decisórios. Decisões baseadas em informações incompletas ou incorretas, possivelmente poderão prejudicar o resultado de estratégia da empresa. No entanto, o uso do Data Warehouse auxilia a gestão apresentando uma arquitetura consolidada de dados transacionais, para suportar e alocar as informações em um único ambiente integrado. O objetivo desse trabalho é o estudo e a criação de um Data Warehouse para auxiliar a gestão de empresas que necessitam tomar decisões corretas. O experimento é demonstrado através de uma aplicação da Microsoft chamado Power BI.

Palavras-chave: Data Warehouse, Power BI, Microsoft.

ABSTRACT

Companies that assist in the management of business activities, have some problems in the agility of information to assist coordinators in decision-making processes. Decisions based on incomplete or incorrect information can possibly harm the outcome of the company's strategy. However, the use of the Data Warehouse helps management by presenting a consolidated transactional data architecture, to support and allocate information in a single integrated environment. The objective of this work is the study and creation of a Data Warehouse to assist the management of these companies that need to make correct decisions and are demonstrated through a Microsoft application called Power BI.

Keywords: Data Warehouse, Power BI, Microsoft.

LISTA DE FIGURAS

Figura 1 - Esquema Estrela.....	16
Figura 2 - Visualizações dos componentes arquiteturais básicos	17
Figura 3 - Fato Venda e VendaDetalhe	27
Figura 4 – Esquema Lógico.....	27
Figura 5 – Tabelas do Data Warehouse.....	28
Figura 6 - Procedimentos realizados para a importação dos dados para o Data Warehouse	30
Figura 7 - Dashboard de Vendas	30

LISTA DE QUADROS

Tabela 1 - Modelo de dados do data Warehouse.....	15
Tabela 2 - Componentes Arquiteturais Básicos	17
Tabela 3 - Vantagens e desvantagens da implementação do Bottom UP	19

LISTA DE ANEXOS

Anexo 01 – Criação do corpo do Data Warehouse.....	36
Anexo 02 – Criação do Data Stage e Log Administração.....	39
Anexo 03 – Criação Tabela Importa Vendas.....	40
Anexo 04 – Criação Dimensão Cliente.....	41
Anexo 05 – Criação Dimensão Geografia.....	42
Anexo 06 – Criação Dimensão Funcionário.....	43
Anexo 07 – Criação Dimensão Produto.....	44
Anexo 08 – Criação Dimensão Data.....	45
Anexo 09 – Criação Fato Venda.....	46
Anexo 10 – Criação Fato Detalhe de Venda.....	48

SUMÁRIO

1. INTRODUÇÃO	11
1.1. OBJETIVOS	12
1.2. JUSTIFICATIVA	12
1.3. MOTIVAÇÃO.....	12
1.4. PERSPECTIVA DE CONTRIBUIÇÃO.....	13
1.5. METODOLOGIA	13
2. REFERENCIAL TEÓRICO.....	14
2.1 DATA WAREHOUSE.....	14
2.1.1 Escopo do Data Warehouse	15
2.1.2 O Data Warehouse.....	15
2.1. ARQUITETURA DO DATA WAREHOUSE	16
2.1.1. Camada de dados operacionais e dados externos.....	18
2.1.2. Camada de extração, transformação e carregamento de dados (ETL) .	18
2.1.3. Camada do Data Warehouse	18
2.2. IMPLEMENTAÇÃO	19
2.2.1. Bottom up	19
2.3. GRANULARIDADE (NÍVEL DE AGREGAÇÃO).....	20
2.4. COMPONENTES DE UM DATA WAREHOUSE.....	20
2.4.1. Surrogate key (chave substituta)	20
2.4.2. Integrado	21
2.4.3. Orientado a assunto	21
2.4.4. Variável no tempo.....	21
2.4.5. Não volátil.....	22
3. MATERIAIS E MÉTODOS	23
3.1. MATERIAIS.....	23
3.1.1. SQL Server 2014.....	23

3.1.2. SQL Developer	23
3.1.3. Power BI	24
3.2. MODELAGEM E IMPLEMENTAÇÃO.....	24
3.2.1. Modelagem de dados dimensional	25
3.2.2. Criação do ambiente e arquitetura.....	25
4. RESULTADOS E DISCUSSÕES	26
4.1. APRESENTAÇÃO DO DATA WAREHOUSE	26
4.1.1. Arquitetura do Data Warehouse	26
4.1.2. Modelagem do Data Warehouse	26
4.1.3. Projeto lógico	27
4.1.4. Projeto físico	28
4.2. CRIAÇÃO DO AMBIENTE DE ETL E DATA WAREHOUSE	28
4.2.1. ETL	29
4.2.2. Extração, Transformação e Carga nos Dados.....	29
4.3. CRIAÇÃO DE DASHBOARD	30
5. CONCLUSÃO	31
6. REFERÊNCIAS.....	32

1. INTRODUÇÃO

Segundo uma pesquisa realizada por Tech Trends (2019), nos últimos anos, com o exponencial crescimento da área de inteligência de negócios, mais de 70% das empresas de tecnologia do Brasil ainda se baseiam em dados imprecisos para as tomadas de decisões mais importantes do negócio. Essa pesquisa, realizada com várias empresas no Brasil, indicou que apenas 28% das empresas de tecnologia participantes adotam ferramentas de Business Intelligence que atuam na geração de informações de inteligência para o gerenciamento de dados, relatórios e gestão de desempenho. Esse é um dado preocupante, uma vez que a inteligência de negócio permite o crescimento sustentável dos negócios.

O objetivo de sistemas de apoio a decisões é capturar dados, informações, conhecimento e responder a eventos de negócios que precisam tomar decisões rápidas e melhores. (POPA, 2010). Entretanto, nas soluções de apoio a decisões existentes nas empresas, nota-se certa demora no processamento das informações causando lentidão no sistema, concorrência entre ferramentas e a carga extra para o banco de dados de transação, principalmente quando envolve um grande período de análise por estas ferramentas.

Dessa forma, com o uso da tecnologia pode-se manter o controle, extraindo melhor estas informações entregando indicadores que antes seriam difíceis de imaginar devido à complexidade envolvida, trazendo um ambiente propício para melhor tomada de decisão.

1.1. OBJETIVOS

O presente projeto tem por objetivo desenvolver um modelo de Data Warehouse iniciando um Data Mart orientado por assunto, no auxílio de gestão. O trabalho proposto conta com uma implementação de uma arquitetura e ambiente ETL (Extração, Transformação e Carga) para o tratamento dos dados e por fim um ambiente para a tomada de decisão.

1.2. JUSTIFICATIVA

O desenvolvimento do Data Warehouse justifica-se por resolver problemas de lentidão no processamento de informações gerenciais com base no passado.

O uso de serviços em nuvem disponibiliza poder de processamento, garantindo disponibilidade do serviço, elasticidade para aumentar ou diminuir os recursos de acordo com a necessidade de processamento e armazenamento, garantindo a continuidade do negócio e menor preocupação para a gestão de TI (Tecnologia da Informação).

1.3. MOTIVAÇÃO

Devido ao alto grau de competitividade no mercado, os gestores têm percebido que a necessidade de uma boa e rápida tomada de decisão pode influenciar nos benefícios e ganhos de uma organização. A partir dessa percepção, e com o avanço da tecnologia, observa-se que cada vez mais as informações e dados são utilizados para

que empresas possam se destacar, utilizando-os como uma forma de se manter na contínua competitividade do mercado. Com isso, a implantação de Data Warehouses e ferramentas de Business Intelligence, tornaram-se um requisito de negócio para grandes, médias e até pequenas empresas.

1.4. PERSPECTIVA DE CONTRIBUIÇÃO

Ao término deste trabalho, pretende-se aumentar a inteligência de negócios da empresa, utilizando ferramentas para manipular e analisar os dados que podem ser feitos consultas, relatórios, ou quaisquer outras análises, visando o aumento de lucro e diminuição das despesas. Pretende-se também publicá-la em artigos e em instituições de ensino para que as pessoas com interesse nesta área possam aproveitar esta pesquisa em seus trabalhos e observarem os resultados obtidos através da inteligência de negócios.

1.5. METODOLOGIA

O Data Warehouse se dá em ambiente de forma centralizada no ambiente da empresa. O Modelo de Dados Dimensional irá ser o Star Schema (Esquema em Estrela) por oferecer mais benefícios na geração de relatórios se comparado com o Modelo Entidade-Relacionamento (E/R). Utilização da técnica top/down no desenvolvimento do trabalho, os níveis de granularidade (agregação), pensando no menor tempo de processamento. O processamento ETL para a carga, extração e transformação dos dados onde na origem e destino usa-se SGBD SQL Server 2014.

Com as necessidades levantadas constrói-se o Data Mart e dimensões de acordo com o método Segundo Kimball *et al.* (1998) onde:

1. Escolher o data mart;

2. Escolher a granularidade (detalhamento);
3. Escolher as dimensões;
4. Escolher a tabela fato.

2. REFERENCIAL TEÓRICO

O referencial teórico é apresentado nesse capítulo para fundamentar o desenvolvimento do data warehouse proposto. Trata-se da arquitetura e modelo de data warehouse e a preparação dos dados.

2.1 DATA WAREHOUSE

Como definição de Data warehouse Date(2004) empregam o termo no qual significa “é um depósito de dados orientado por assunto, integrado, não volátil, variável com o tempo, que fornece suporte á tomada de decisões”.

Segundo Elmasri e Navathe (2011, p. 422), vê-se que “o data warehouse tem como característica distintiva de servir principalmente para aplicações de apoio a decisão”.

De acordo com estes autores, o uso de banco de dados com a finalidade de oferecer apoio á decisão, são sistemas que auxiliam a análise de informação, onde a meta é definir tendências, apontando os problemas e tomando decisões inteligentes, baseados nessas informações.

Pode-se afirmar que o Data warehouse é um banco de dados para armazenar, extrair e processar os dados, dando suporte para gerar informações.

2.1.1 Escopo do Data Warehouse

Segundo Singh (2001), “o data warehouse pode funcionar como um repositório central para alocar grandes quantidades de informações”, pode se criar data marts também contém dados sumarizados e tabelas (Dimensões e fato) para o suporte as ferramentas de processamento analítico online (OLAP).

2.1.2 O Data Warehouse

Para implantar o Data Warehouse, existem algumas regras de design e algumas boas práticas. O Quadro 01 mostra detalhadamente os modelos e as características de um DW.

MODELO	CARACTERÍSTICAS
1-E/R	Representação do modelo utilizando entidades e os relacionamentos entre estas entidades. Apresentação do modelo através do diagrama E/R. Principais componentes: entidades, relacionamentos e atributos
2-Estrela	É a estrutura básica do modelo dimensional. Basicamente é composto de uma grande tabela central (tabela fato) e um conjunto de pequenas tabelas (tabelas de dimensão) dispostas ao redor da tabela fato Dentro de cada categoria, existe uma única tabela fato histórica simples, contendo detalhes e dados resumidos, armazenados nos níveis de estrutura indicados em cada tabela dimensional A chave primária da tabela fato contém somente uma coluna de cada dimensão Cada chave é uma chave gerada pelo sistema Cada dimensão é representada por uma única tabela fato, usando também uma chave gerada pelo sistema
3-Snowflake	O modelo dimensional típico inicia com uma tabela fato e um único nível de várias dimensões ao redor. O modelo <i>snowflake</i> é o resultado da decomposição de uma ou mais dessas dimensões. Algumas vezes, esta decomposição pode acontecer em uma dimensão que já foi decomposta, formando uma hierarquia

Tabela 1 - Modelo de dados do data Warehouse

Na imagem a baixo (Figura 01), vemos um esquema estrela e suas tabelas lógicas de dimensões e fatos para armazenar informações de vendas (tabela de fatos vendas).

É um esquema comum, projetado para uso por aplicações OLAP, na forma normal de Boyce-Codd (FBNC). A maior parte dos dados ficam na tabela fato, sem redundância, segundo Ramakrishnan e Gehrke (2008, p.514) “a FBNC garante que nenhuma redundância pode ser detectada usando-se apenas as informações da DF” (dependência funcional), sendo a forma normal mais desejável partindo do ponto de vista de não existir redundância na tabela fato.

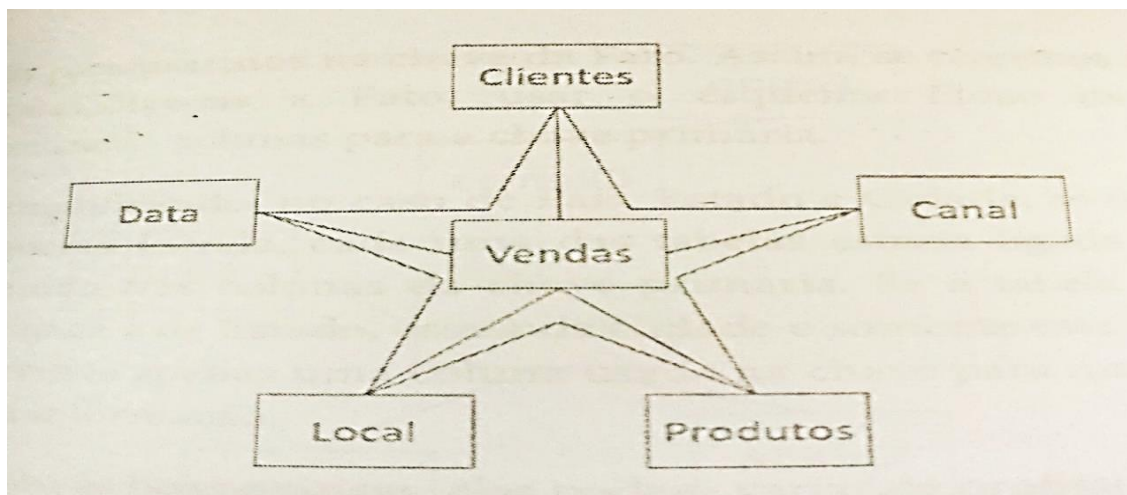


Figura 1 - Esquema Estrela

2.1. ARQUITETURA DO DATA WAREHOUSE

A arquitetura visa cobrir uma faixa de tecnologia e aplicações para o gerenciamento de todo o ciclo de vida dos dados, o fluxo das atividades se inicia com a extração dos dados das origens. Esses dados são armazenados temporariamente na Staging Area, onde são tratados com as regras e padrões predeterminados para então prosseguir para a etapa de carga em que os dados são carregados no DW. Por fim, essas informações são normalmente consultadas através de ferramentas de análises, para encontrar as respostas e insights necessários para as tomadas de decisões.

Verificam-se no Quadro 2 e na Figura 2, os componentes arquiteturais básicos de BI (Business Intelligence):

COMPONENTE	DESCRIÇÃO
Ferramenta de extração, transformação e carregamento de dados (ETL)	Esse componente é encarregado de coletar, filtrar, integrar e agregar dados operacionais a serem salvos em um armazenamento de dados otimizado para o suporte a decisões. Como o nome diz, esse componente extrai os dados, filtra os dados extraídos para selecionar os registros relevantes e empacota os dados no formato certo para ser adicionado ao componente de armazenamento de dados.
Armazenamento de dados	O armazenamento de dados é otimizado para o suporte a decisões e costuma ser representado por um <i>data warehouse</i> ou <i>data mart</i> . Ele contém dados de negócios extraídos de bancos operacionais e de fontes externas. Esses dados são armazenados em estruturas otimizadas para a velocidade de análise e consulta. As fontes de dados externos fornecem dados que não podem ser encontrados no interior da empresa, mas que são relevantes para os negócios como preço de ações, indicadores de mercado, informações de marketing (como as demográficas) e dados de competidores.
Ferramenta de consulta e análise de dados	Esse componente executa as tarefas de recuperação, análise e mineração, utilizando os dados no armazenamento e os modelos de análise de dados de negócios. Tal componente é utilizado pelo analista de dados para criar as consultas que acessam o banco. Dependendo da implementação, a ferramenta de consulta acessa tanto o banco de dados operacional como, o que é mais comum, o armazenamento de dados. Essa ferramenta orienta o usuário sobre quais dados selecionar e como construir um modelo de dados confiável. Tal componente costuma aparecer na forma de uma ferramenta OLAP.
Ferramenta de apresentação e visualização de dados	Esse componente é encarregado de apresentar os dados ao usuário final de vários modos. É utilizado pelo analista de dados para organizar e apresentar os dados. Essa ferramenta ajuda o usuário final a selecionar o formato de apresentação mais adequado, como relatório resumido, mapa, gráfico de pizza ou barra ou gráfico mistos. A ferramenta de consulta e a ferramenta de apresentação são a extremidade final do ambiente de BI.

Tabela 2 - Componentes Arquiteturais Básicos

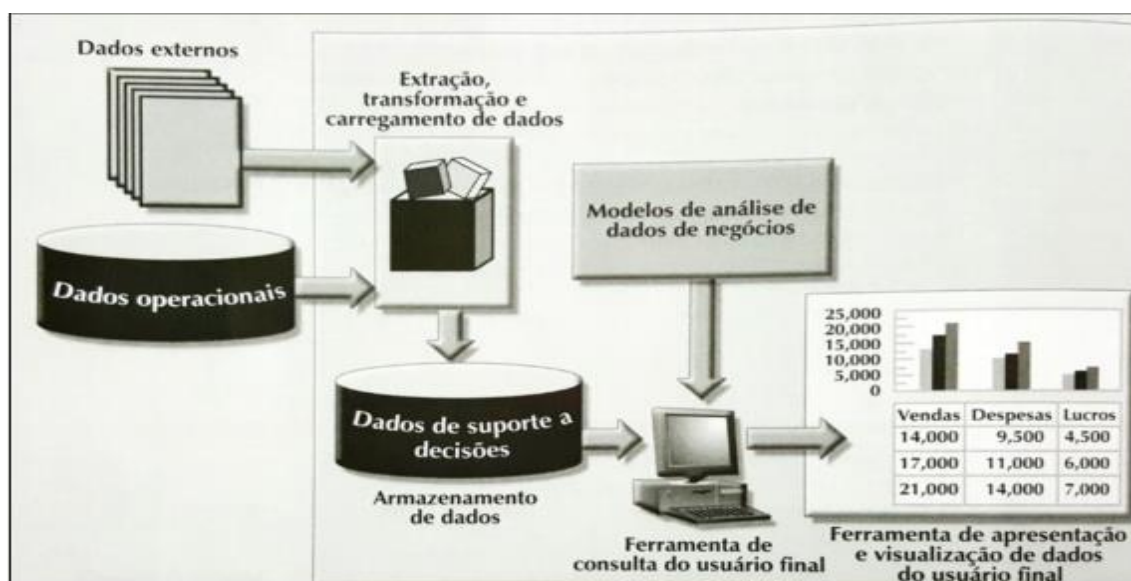


Figura 2 - Visualizações dos componentes arquiteturais básicos

Após a arquitetura do Data Warehouse ser descrita, é importante apresentar a camada de dados operacionais e dados externos para compor o data warehouse.

2.1.1. Camada de dados operacionais e dados externos

Nesta camada constam os dados operacionais e as fontes de dados externas, nos quais serão tratados e integrados para compor o data warehouse.

Ela inclui todos os processos necessários para selecionar,editar,resumir,combinar e carregar o Data Warehouse a partir de dados operacionais e de bancos de dados externos, minimizando o impacto do banco de dados operacional, onde os recursos de extração de dados devem permitir a extração de dados em batch e programada.

2.1.2. Camada de extração, transformação e carregamento de dados (ETL)

Extração, transformação e carregamento de dados (ETL), são ferramentas cuja função é a extração de dados de diversos sistemas, transforma esses dados para selecionar os registros relevantes e por fim o empacota os dados no formato certo para ser adicionado ao componente de armazenamento de dados, o Data Warehouse. (ROB, CORONEL, 2011).

2.1.3. Camada do Data Warehouse

Camada do Data Warehouse é a camada onde os dados são armazenados e manipulados, oferecendo acesso para as ferramentas de consultas utilizarem os dados armazenados.

De acordo com Rob e Coronel (2011), o esquema deve proporcionar representação de dados complexos, a recuperação das consultas de leitura deve ser otimizada com suporte a recursos como índices de bitmap e particionamento de dados, que aumentam a velocidade das buscas e suporte a estruturas não normalizadas e complexas.

2.2. IMPLEMENTAÇÃO

A escolha de implementar o Data Warehouse parte da idéia de ter um baixo custo de software e hardware e um curto tempo de implementação, e essa escolha traz a forma de implementação do Bottom Up que é uma metodologia que contribui para um melhor desempenho operacional.

2.2.1. Bottom up

Segundo Clemes(2001), “a opção por esta implementação permite resultados mais rápidos, pois está baseada na construção de data marts, que tem uma menor complexidade do projeto do que um data warehouse global” Observa-se no Quadro 03, as vantagens e desvantagens do uso.

VANTAGENS	DESVANTAGENS
Apresenta resultados imediatos	Pode ocorrer redundância de dados e inconsistência entre os <i>data marts</i>
Menor complexidade na elaboração do projeto	A integração dos <i>data marts</i> em um <i>data warehouse</i> global pode ser difícil até uma determinada etapa do projeto
Necessidades iniciais de hardware e outros recursos têm custo relativamente baixo	Pode ser necessário retrabalho quando novas áreas são atendidas, ou quando mudanças naquelas que já estão implementadas são necessárias

Tabela 3 - Vantagens e desvantagens da implementação do Bottom UP

2.3. GRANULARIDADE (NÍVEL DE AGREGAÇÃO)

A granularidade afeta diretamente no volume de dados armazenados, na velocidade das consultas e no nível de detalhamento das informações do DW. Quanto maior for o detalhamento, maior será a flexibilidade para se obter respostas. Porém, maior será o volume e menor a velocidade das consultas. Já quanto menor for o detalhamento, menor será o volume, maior a sumarização dos dados e melhor será a performance.

De acordo com Singh(2001, p.147) sobre agregação, “agregação é o processo pelo qual os dados de nível baixo são previamente sumarizados e incluídos em tabelas que armazenam informações “agregadas” ou “sumarizadas”.

2.4. COMPONENTES DE UM DATA WAREHOUSE

A seguir apresentam-se algumas definições e componentes de um data warehouse de acordo com Rob e Coronel (2011).

2.4.1. Surrogate key (chave substituta)

Lima (2011, p.5) descreve sobre chave substituta “levando esse conceito para uma modelagem dimensional, aprendemos que as chaves compostas em dimensões causam uma certa complexidade na estruturação de um SQL, além de afetar o tempo de execução desse SQL”.

2.4.2. Integrado

Um dos aspectos mais importantes do Data Warehouse é que os dados criados dentro de um ambiente Data Warehouse são integrados, sempre, com nenhuma exceção. A integração mostra-se em diferentes maneiras na convenção consistente dos nomes, na forma consistente das variáveis, na estrutura consistente de códigos, nos atributos físicos consistentes dos dados.

Do mesmo modo que um elemento de dado é medido em centímetros em uma aplicação, em polegadas em outra, ele será convertido para uma representação única ao ser colocado no Data Warehouse.

2.4.3. Orientado a assunto

Os dados no data warehouse são dispostos e otimizados fornecendo respostas a perguntas derivadas de diversos setores da empresa, resumidos por tema, cada tema é relacionado a assuntos de interesse específicos, sendo essa forma de organização de dados diferente das disposições mais orientada a processos ou funcional comum das transações.

2.4.4. Variável no tempo

O Data warehouse possibilita a análise de grandes volumes de dados coletados dos sistemas transacionais. São as chamadas séries históricas que possibilitam uma melhor análise de eventos passados, oferecendo suporte às tomadas de decisões

presentes e a previsão de eventos futuros. Por definição, os dados em um data warehouse não são voláteis, ou seja, eles não mudam, salvo quando é necessário fazer correções de dados previamente carregados.

Os dados estão disponíveis somente para leitura e não podem ser alterados.

2.4.5. Não volátil

Quando inserido um dado no data warehouse o mesmo nunca será removido, representando o histórico da empresa, com os dados operacionais inseridos continuamente, crescendo em volume de dados com vários gigabytes, se diferenciando dos bancos de dados operacionais (transações). (ROB, CORONEL, 2011).

3. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e o método utilizado para a realização deste trabalho. Os materiais, referem-se às tecnologias como linguagens e ferramentas para a modelagem e a implementação do sistema. O método contém as etapas com os principais procedimentos utilizados para o desenvolvimento do data warehouse, abrangendo o levantamento dos requisitos aos testes.

3.1. MATERIAIS

As ferramentas e as tecnologias utilizadas para compor a modelagem, a arquitetura e a implementação do data warehouse são:

- a) SQL Server 2014 para armazenar o banco de dados do data warehouse;
- b) SQL Developer para administrar consultas SQL;
- c) Power BI ferramenta para demonstrar a consulta ao data warehouse;

3.1.1. SQL Server 2014

Responsável por armazenar as tabelas fato e dimensões no data warehouse.

3.1.2. SQL Developer

Utilizado para criar as consultas, tabelas fato, tabelas dimensões, consultas e estrutura do data warehouse.

3.1.3. Power BI

Power BI ferramenta para coletar, tratar e organizar todos os dados do data warehouse, dessa forma, os resultados e as informações sejam entregues da maneira desejada.

3.2. MODELAGEM E IMPLEMENTAÇÃO

O método utilizado para a realização deste trabalho expões de maneira sequencial as atividades para obter o resultado.

- **Análise**
 - Identificação dos requisitos de negócio e riscos
 - Definir origens, qualidade e latência das informações
 - Definir regras de negócio corporativas e departamentais
- **Desenvolvimento**
 - Data Warehouse (ou Datamart)
 - Desenho
 - Teste de Mesa
 - Implantação
 - Processos de Carga
 - Desenho e Regras
 - Implantação
 - Relatórios e dashboards
 - Dashboard comercial
 - Relatório de apoio ao vendedor

3.2.1. Modelagem de dados dimensional

A modelagem de dados dimensional em estrela aplicada para agrupar os dados, criar as tabelas de dimensões e fato para atender os requisitos.

3.2.2. Criação do ambiente e arquitetura

O ambiente e arquitetura se dá em forma centralizada em ambiente *on premise*, com recursos do computador local.

4. RESULTADOS E DISCUSSÕES

Este capítulo apresenta o data warehouse desenvolvido como resultado deste trabalho. Inicialmente é apresentada a sua descrição e em seguida é apresentada a sua modelagem.

4.1. APRESENTAÇÃO DO DATA WAREHOUSE

Abaixo, apresenta-se o desenvolvimento do projeto lógico e físico do data warehouse.

4.1.1. Arquitetura do Data Warehouse

Ambiente desenvolvido para que se tenha um sistema de data warehouse utilizando banco de dados relacional, podendo trabalhar na fase de carga dos dados no data warehouse (ETL).

4.1.2. Modelagem do Data Warehouse

O Desenho do data warehouse inicialmente foi definido um modelo em estrela para implementar o *data stage*.

Verifica-se na Figura 3 um assunto venda, onde existem duas tabelas fato histórica (F_VENDA E F_VENDADETALHE), contendo os dados das dimensões.

As tabelas de Fato possuem uma Primary Key composta com todas as demais chaves das dimensões que se relacionam com ela.

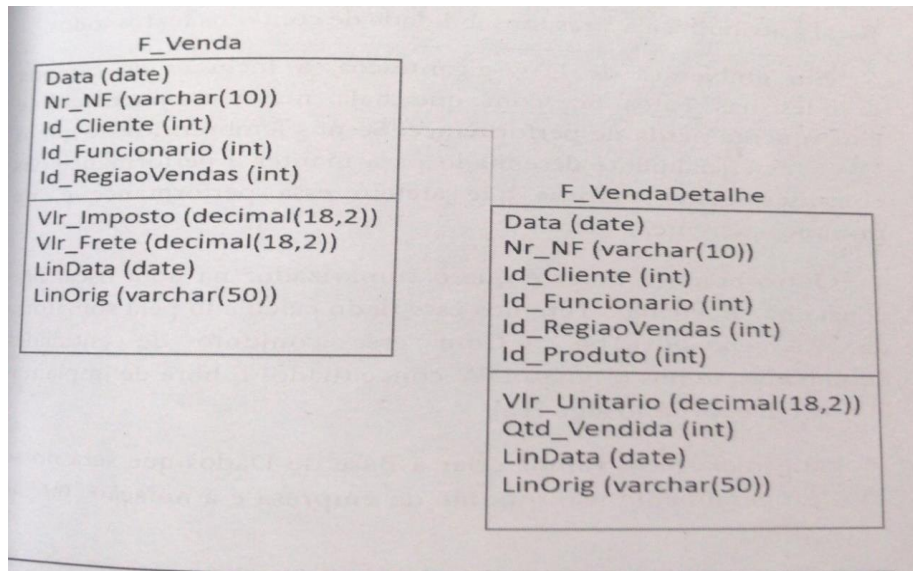


Figura 3 - Fato Venda e VendaDetalhe

4.1.3. Projeto lógico

Na Figura 4 está elencada o diagrama em estrela visualizando as tabelas envolvidas no Schema Lógico

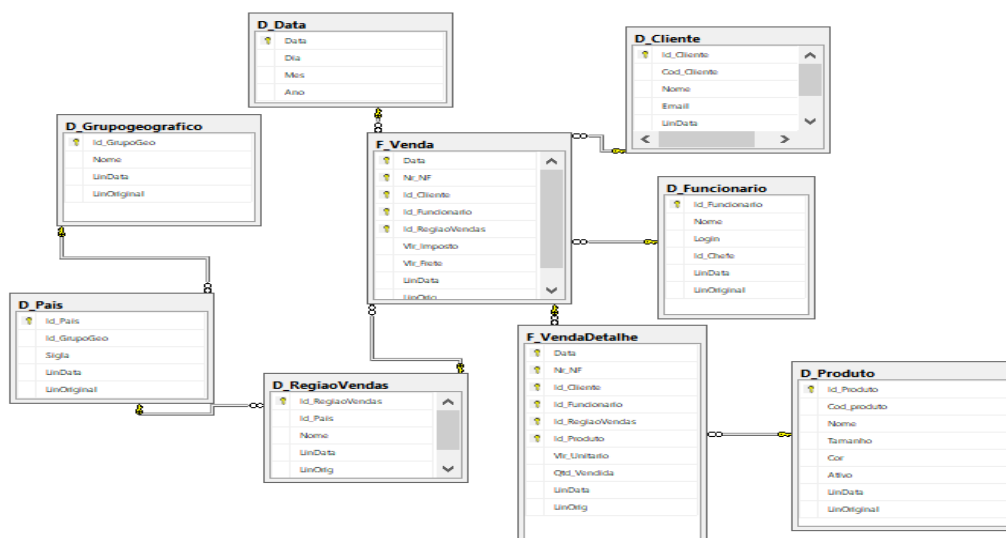


Figura 4 – Esquema Lógico

4.1.4. Projeto físico

Os códigos SQL utilizados se encontram no Anexo 1. Na figura a baixo vê-se o projeto físico, e na Figura 5 os objetos (tabelas) criadas no data warehouse:



Figura 5 – Tabelas do Data Warehouse

4.2. CRIAÇÃO DO AMBIENTE DE ETL E DATA WAREHOUSE

Nesta etapa se faz a preparação dos dados para que sejam alimentadas no data Warehouse.

Os indicadores e respostas que serão implementados para a validação do ambiente de data Warehouse são:

- Dashboard de Vendas contendo
 - Análise por produto
 - Top 10 Produtos por Valor de Venda
 - Vendas por cliente
 - Volume de Vendas

4.2.1. ETL

Abaixo consta o resumo dos comandos executados no Management Studio para extrair, transformar e carregar os dados para as tabelas dimensões do data Warehouse:

1. Criação do corpo do Data Warehouse
2. Criação do Data Stage e Log Administração
3. Criação Tabela Importa Vendas
4. Criação Dimensão Cliente
5. Criação Dimensão Geografia
6. Criação Dimensão Funcionário
7. Criação Dimensão Produto
8. Criação Dimensão Data
9. Criação Fato Venda
10. Criação Fato Detalhe de Venda

4.2.2. Extração, Transformação e Carga nos Dados

Após definir quais dados serão extraídos, transformados e carregados no data warehouse, os procedimentos armazenados no Data Stage enviam os dados da origem para o destino como demonstra na Figura 6.

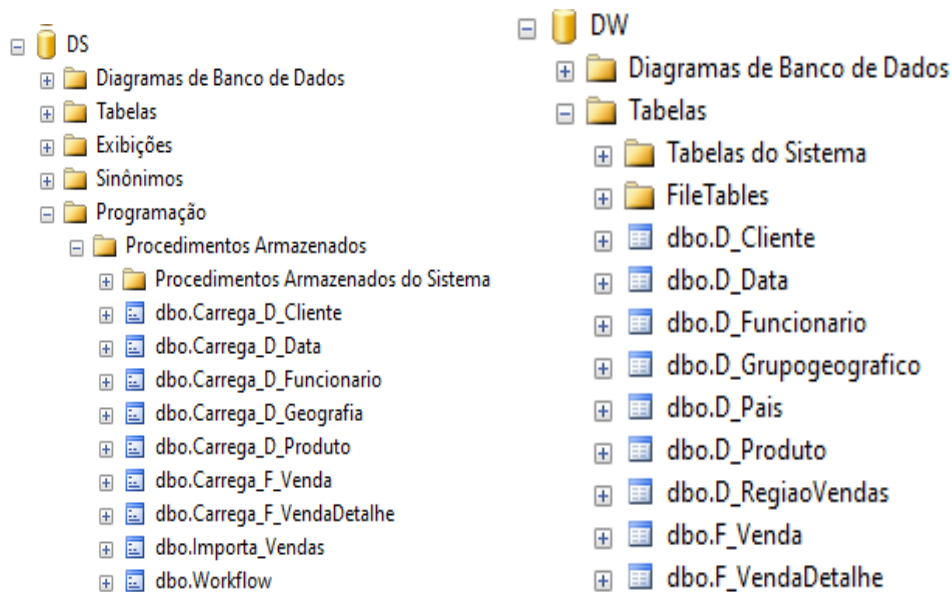


Figura 6 - Procedimentos realizados para a importação dos dados para o Data Warehouse

4.3. CRIAÇÃO DE DASHBOARD

Já com o ambiente de Data Warehouse criado, utiliza-se o Microsoft Power BI Desktop para criar os dashboards visualizando os dados com uma nova percepção e um entendimento melhor com gráficos e tabelas.

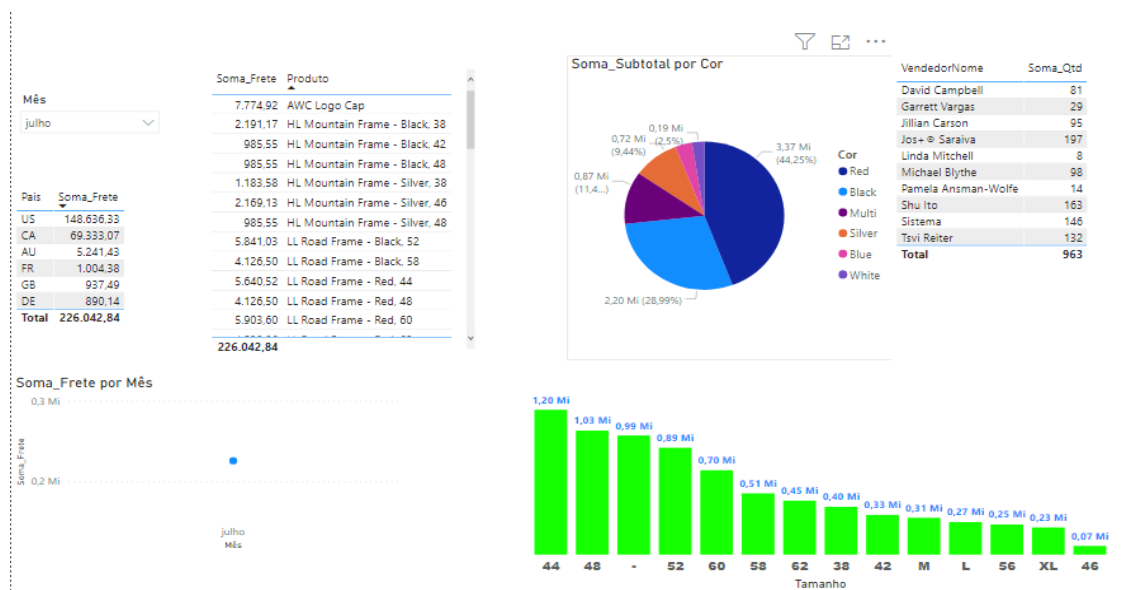


Figura 7 - Dashboard de Vendas

5. CONCLUSÃO

O Data Warehouse tem como objetivo o auxílio na tomada de decisão de uma gestão empresarial, já que apresenta uma arquitetura de consolidação de dados transacionais para suportar e dar acesso as informações em um único ambiente integrado.

No presente estudo pode-se concluir que é possível criar um banco de dados totalmente funcional a custo zero.

A arquitetura do ambiente de extração, transformação e carregamento foi criado para suportar e dar acesso as informações e consolidar suas informações para não sobrecarregar o banco de dados transacional.

Criar um Data Stage de acordo com o assunto e a necessidade do negócio viabiliza o tempo de entrega do Data Warehouse e justifica a sua implantação para a gestão da empresa.

6. REFERÊNCIAS

BRAGHITTONI, Ronaldo. Business Intelligence: Implementar do jeito certo e a custo zero. São Paulo: 1 Ed, 2017.

DATE, C. J. Introdução a Sistemas de Bancos de Dados. 15. ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson Education, 2011

KIMBALL, R.; ROSS, M. *The Data Warehouse toolkit: the complete guide to dimensional modeling*. New York: John Wiley & Sons, 2002.

LIMA, Carlos Alberto. Conceito de Surrogate Key – Chaves Substitutas (2009). Disponível em: . Acesso em: já. 2017.

POPA, C. *Business Intelligence Performance and Capacity Impact*. 2010. 10 f. Universidade Politécnica de Bucarest. Disponível em: https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full8324.pdf

RAMAKRISHNAN, R.; GEHRKE, J. Sistemas de gerenciamento de banco de dados. 3. ed. McGraw-Hill: São Paulo, 2008.

ROB, P. CORONEL, C. Sistemas de banco de dados: Projeto, Implementação e gerenciamento. São Paulo: Cengage Learning, 2011.

SINGH, H. S. *Data Warehouse: Conceitos, Tecnologias, Implementação e Gerenciamento*. São Paulo: Makron Books, 2001.

TECHTRENDS: *Benchmarks de Ferramentas Para Empresas De Tecnologia*. Disponível em: <http://materiais.resultadosdigitais.com.br/techtrends>

ANEXOS

ANEXO 1 – Criação do corpo do Data Warehouse.

```
USE [DW]
GO

CREATE TABLE [dbo].[D_Cliente](
  [Id_Cliente] [int] identity(1,1) NOT NULL,
  [Cod_Cliente] [varchar](10) NOT NULL,
  [Nome] [varchar](50) NOT NULL,
  [Email] [varchar](50) NOT NULL,
  [LinData] [date] NOT NULL,
  [LinOrig] [varchar](50) NOT NULL,
  CONSTRAINT [PK_D_Cliente] PRIMARY KEY CLUSTERED
(
  [Id_Cliente] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_Data](
  [Data] [date] NOT NULL,
  [Dia] [char](2) NOT NULL,
  [Mes] [char](2) NOT NULL,
  [Ano] [char](2) NOT NULL,
  CONSTRAINT [PK_D_Data] PRIMARY KEY CLUSTERED
(
  [Data] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_Funcionario](
  [Id_Funcionario] [int] identity(1,1) NOT NULL,
  [Nome] [varchar](50) NOT NULL,
  [Login] [varchar](50) NOT NULL,
  [Id_Chefe] [int] NULL,
  [LinData] [date] NOT NULL,
  [LinOriginal] [varchar](50) NOT NULL,
  CONSTRAINT [PK_D_Funcionario] PRIMARY KEY CLUSTERED
(
  [Id_Funcionario] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_Grupogeografico] (
  [Id_GrupGeo] [int] identity(1,1) NOT NULL,
  [Nome] [varchar](50) NOT NULL,
  [LinData] [date] NOT NULL,
  [LinOriginal] [varchar](50) NOT NULL,
  CONSTRAINT [PK_D_Grupogeografico] PRIMARY KEY CLUSTERED (
  [Id_GrupGeo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_Pais] (
  [Id_Pais] [int] identity(1,1) NOT NULL,
  [Id_GrupGeo] [int] NOT NULL,
  [Sigla] [char](2) NOT NULL,
  [LinData] [date] NOT NULL,
  [LinOriginal] [varchar](50) NOT NULL,
  CONSTRAINT [PK_D_Pais] PRIMARY KEY CLUSTERED
(
  [Id_Pais] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_Produto] (
  [Id_Produto] [int] identity(1,1) NOT NULL,
  [Cod_produto] [varchar](20) NOT NULL,
  [Nome] [varchar](5) NOT NULL,
  [Tamanho] [varchar](5) NOT NULL,
  [Cor] [varchar](20) NOT NULL,
  [Ativo] [char](1) NOT NULL,
  [LinData] [date] NOT NULL,
  [LinOriginal] [varchar](50) NOT NULL,
  CONSTRAINT [PK_D_Produto] PRIMARY KEY CLUSTERED
```

```

(
[Id_Produto] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[D_RegiaoVendas](
[Id_RegiaoVendas] [int] identity(1,1) NOT NULL,
[Id_Pais] [int] NOT NULL,
[Nome] [varchar](20) NOT NULL,
[LinData] [date] NOT NULL,
[LinOrig] [varchar](50) NOT NULL,
CONSTRAINT [PK_D_RegiaoVendas] PRIMARY KEY CLUSTERED
(
[Id_RegiaoVendas] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[F_Venda] (
[Data] [date] NOT NULL,
[Nr_NF] [varchar](10) NOT NULL,
[Id_Cliente] [int] NOT NULL,
[Id_Funcionario] [int] NOT NULL,
[Id_RegiaoVendas] [int] NOT NULL,
[Vlr_Imposto] [decimal] (18,2) NOT NULL,
[Vlr_Frete] [decimal] (18,2) NOT NULL,
[LinData] [date] NOT NULL,
[LinOrig] [varchar](50) NOT NULL,
CONSTRAINT [PK_F_Venda] PRIMARY KEY CLUSTERED
(
[Data] ASC,
[Nr_NF] ASC,
[Id_Cliente] ASC,
[Id_Funcionario] ASC,
[Id_RegiaoVendas] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[F_VendaDetalhe](
[Data] [date] NOT NULL,
[Nr_NF] [varchar](10) NOT NULL,
[Id_Cliente] [int] NOT NULL,
[Id_Funcionario] [int] NOT NULL,
[Id_RegiaoVendas] [int] NOT NULL,
[Id_Produto] [int] NOT NULL,
[Vlr_Unitario] [decimal](18,2) NOT NULL,
[Qtd_Vendida] [int] NOT NULL,
[LinData] [date] NOT NULL,
[LinOrig] [varchar](50) NOT NULL,
CONSTRAINT [PK_F_VendaDetalhe] PRIMARY KEY CLUSTERED
(
[Data] ASC,
[Nr_NF] ASC,
[Id_Cliente] ASC,
[Id_Funcionario] ASC,
[Id_RegiaoVendas] ASC,
[Id_Produto] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE NONCLUSTERED INDEX [IX_D_Pais] ON [dbo].[D_Pais]
(
[Id_GruposGeo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE =
OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

CREATE NONCLUSTERED INDEX [IX_D_RegiaoVendas] ON [dbo].[D_RegiaoVendas]
(
[Id_Pais] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE =
OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

ALTER TABLE [dbo].[D_Pais] WITH CHECK ADD CONSTRAINT [FK_D_Pais_D_GruposGeografico] FOREIGN KEY
([Id_GruposGeo]) REFERENCES

```

```

[dbo].[D_GrupoGeografico] ([Id_GrupoGeo])
GO

ALTER TABLE [dbo].[D_Pais] CHECK CONSTRAINT [FK_D_Pais_D_GrupoGeografico]
GO

ALTER TABLE [dbo].[D_RegiaoVendas] WITH CHECK ADD CONSTRAINT [FK_D_RegiaoVendas_D_Pais] FOREIGN KEY
([Id_Pais]) REFERENCES
[dbo].[D_Pais] ([Id_Pais])
GO

ALTER TABLE [dbo].[D_RegiaoVendas] CHECK CONSTRAINT [FK_D_RegiaoVendas_D_Pais]
GO

ALTER TABLE [dbo].[F_Venda] WITH CHECK ADD CONSTRAINT [FK_F_Venda_D_Cliente] FOREIGN KEY ([Id_Cliente])
REFERENCES
[dbo].[D_Cliente] ([Id_Cliente])
GO

ALTER TABLE [dbo].[F_Venda] CHECK CONSTRAINT [FK_F_Venda_D_Cliente]
GO

ALTER TABLE [dbo].[F_Venda] WITH CHECK ADD CONSTRAINT [FK_F_Venda_D_Data] FOREIGN KEY ([Data]) REFERENCES
[dbo].[D_Data] ([Data])
GO

ALTER TABLE [dbo].[F_Venda] CHECK CONSTRAINT [FK_F_Venda_D_Data]
GO

ALTER TABLE [dbo].[F_Venda] WITH CHECK ADD CONSTRAINT [FK_F_Venda_D_Funcionario] FOREIGN KEY
([Id_Funcionario]) REFERENCES
[dbo].[D_Funcionario] ([Id_Funcionario])
GO

ALTER TABLE [dbo].[F_Venda] CHECK CONSTRAINT [FK_F_Venda_D_Funcionario]
GO

ALTER TABLE [dbo].[F_Venda] WITH CHECK ADD CONSTRAINT [FK_F_Venda_D_RegiaoVendas] FOREIGN KEY
([Id_RegiaoVendas]) REFERENCES
[dbo].[D_RegiaoVendas] ([Id_RegiaoVendas])
GO

ALTER TABLE [dbo].[F_Venda] CHECK CONSTRAINT [FK_F_Venda_D_RegiaoVendas]
GO

ALTER TABLE [dbo].[F_VendaDetalhe] WITH CHECK ADD CONSTRAINT [FK_F_VendaDetalhe_D_Produto] FOREIGN KEY
([Id_Produto]) REFERENCES
[dbo].[D_Produto] ([Id_Produto])
GO

ALTER TABLE [dbo].[F_VendaDetalhe] CHECK CONSTRAINT [FK_F_VendaDetalhe_D_Produto]
GO

ALTER TABLE [dbo].[F_VendaDetalhe] WITH CHECK ADD CONSTRAINT [FK_F_VendaDetalhe_F_Venda] FOREIGN KEY
([Data], [Nr_NF], [Id_Cliente],
[Id_Funcionario], [Id_RegiaoVendas]) REFERENCES [dbo].[F_Venda] ([Data], [Nr_NF],
[Id_Cliente],[Id_Funcionario], [Id_RegiaoVendas])
GO

ALTER TABLE [dbo].[F_VendaDetalhe] CHECK CONSTRAINT [FK_F_VendaDetalhe_F_Venda]
GO

```

ANEXO 2 – Criação do Data Stage e Log Administração

```
use [DS]
go

CREATE TABLE [dbo].[Adm_Log](
  [Id_Log] [uniqueidentifier] primary key NOT NULL,
  [Data] [datetime] NOT NULL,
  [Passo] [varchar](50) NOT NULL,
  [SucessoFalha] [char](1) NOT NULL,
  [mensagem] [varchar](255) NOT NULL)
```

ANEXO 3 – Criação Tabela Importa Vendas

```
CREATE TABLE [dbo].[TbImp_Vendas](
[NrNf] [varchar](50) NULL,
[DataVenda] [varchar](50) NULL,
[CodCliente] [varchar](50) NULL,
[NomeCliente] [varchar](50) NULL,
[EmailCliente] [varchar] (50) NULL,
[RegiaoVendas] [varchar] (50) NULL,
[Pais] [varchar] (50) NULL,
[GrupoGeografico] [varchar] (50) NULL,
[VendedorLogin] [varchar] (50) NULL,
[VendedorNome] [varchar] (50) NULL,
[VendedorChefeNome] [varchar] (50) NULL,
[Cod_Produto] [varchar] (20) NULL,
[Produto] [varchar] (50) NULL,
[Tamanho] [varchar] (50) NULL,
[Linha] [varchar] (50) NULL,
[Cor] [varchar] (50) NULL,
[SubTotal] [varchar] (50) NULL,
[ImpTotal] [varchar] (50) NULL,
[Frete] [varchar] (50) NULL,
[PrecoUnitario] [varchar] (50) NULL,
[Qtd] [varchar] (50) NULL
) ON [PRIMARY]

USE [DS]
Create procedure [dbo].[Importa_Vendas]
as
Declare @STR as nvarchar(1000)

truncate table TbImp_Vendas

set @STR = 'Move c:\BI\*.rpt c:\BI\MassaDados2005.rpt'

exec xp_cmdshell @STR

bulk insert [dbo].[TbImp_Vendas]
from 'C:\BI\MassaDados2005.rpt'
with (
    FIRSTROW = 3,
    FIELDTERMINATOR = '|',
    ROWTERMINATOR = '\n'
)

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa MassaDados2005.rpt', 'S', 'Arquivo
importado com sucesso')

declare @nomearquivo varchar(50)
Set @nomearquivo = (Select cast(year(getdate()) as char(4)) + right('00' + cast(month(getdate())
as varchar(2)),2)
+ right('00' + cast(day(getdate()) as varchar(2)),2) + '_Vendas.rpt')

Set @STR = 'move c:\BI\MassaDados2005.rpt c:\BI\Historico\' + @nomearquivo

exec xp_cmdshell @STR

USE [DS]
GO
exec Importa_Vendas
```

ANEXO 4 - Criação Dimensão Cliente

```
CREATE TABLE [dbo].[D_Cliente](
    [Cod_Cliente] [varchar](10) NOT NULL,
    [Nome] [varchar](50) NOT NULL,
    [Email] [varchar](50) NOT NULL,
    [LinData] [date] NOT NULL,
    [LinOrig] [varchar](50) NOT NULL
)
GO
create index IX_Cod_Cliente on DS..D_Cliente (Cod_Cliente)
GO
create index IX_Cod_Cliente on DW..D_Cliente (Cod_Cliente)

Create procedure [dbo].[Carrega_D_Cliente]
as
begin try
    truncate table [DS]..[D_Cliente]

    insert into [dbo].[D_Cliente]
    Select Distinct
    CodCliente as [Cod_Cliente],
    NomeCliente as [Nome],
    EmailCliente as [Email],
    Getdate() as [LinData],
    'Arquivo de vendas' as [LinOrig]
    from [DS]..[TbImp_Vendas]

    insert into [DW]..[D_Cliente]
    Select
    ds.Cod_Cliente,
    ds.Nome,
    ds.Email,
    ds.LinData,
    ds.LinOrig
    from [DS]..[D_Cliente] as ds left join [DW]..[D_Cliente] as dw
    on ds.Cod_Cliente = dw.Cod_Cliente
    where dw.ID_Cliente is null

    insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Cliente', 'S', 'Carga de D_Cliente
com sucesso.')
end try
begin catch
insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Cliente', 'F', 'Erro ao carregar
D_Cliente.')
end catch
```

ANEXO 5 – Criação Dimensão Geografia

```
USE [DS]
GO

Create procedure [dbo].[Carrega_D_Geografia]
as

begin try
    truncate table [DS]..[D_GrupoGeografico]
    truncate table [DS]..[D_Pais]
    truncate table [DS]..[D_RegiaoVendas]

insert into [dbo].[D_GrupoGeografico]
Select Distinct
GrupoGeografico, Getdate() as [LinData], 'Arquivo de Vendas' as [LinOrig] from [DS]..[TbImp_Vendas]

insert into [DW]..[D_GrupoGeografico]
Select Distinct
ds.Nome, ds.LinData, ds.LinOrig from [DS]..[D_GrupoGeografico] as ds left join DW..D_GrupoGeografico as dw
on ds.[Nome] = dw.[Nome] where dw.[Id_GrupoGeo] is null

insert into [dbo].[D_Pais]
Select Distinct
dw.Id_Grupogeo, ds.Pais, Getdate() as [LinData], 'Arquivo de Vendas' as [LinOrig] from [DS]..[TbImp_Vendas]
as ds inner join
[DW]..[D_GrupoGeografico] as dw on ds.GrupoGeografico = dw.Nome

insert into [DW]..[D_Pais]
select
ds.[Id_GrupoGeo], ds.[Sigla], ds.[LinData], ds.[LinOrig] from [DS]..[D_Pais] as ds left join [DW]..[D_Pais]
as dw on ds.Id_GrupoGeo = dw.Id_GrupoGeo
and ds.Sigla = dw.Sigla where dw.Id_Pais is null

insert into [dbo].[D_RegiaoVendas]
select Distinct
dw.Id_Pais, ds.RegiaoVendas, Getdate() as [LinData], 'Arquivo de Vendas' as [LinOrig] from
[DS]..[TbImp_Vendas] as ds inner join [DW]..[D_Pais] as dw
on ds.Pais = dw.[Sigla]

insert into [DW]..[D_RegiaoVendas]
Select
ds.[Id_Pais], ds.[Nome], ds.[LinData], ds.[LinOrig]
from [DS]..[D_RegiaoVendas] as ds left join [DW]..[D_RegiaoVendas] as dw
on ds.Id_Pais = dw.Id_Pais and ds.Nome = dw.Nome where dw.Id_RegiaoVendas is null

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Geografia', 'S', 'Carga das tabelas de
geografia com sucesso.')

end try
begin catch
insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Geografia', 'F', 'Erro a carregar tabelas
de geografia.')
end catch

execute Carrega_D_Geografia

USE [DS]

select
a.[Nome] as RegGeo,
b.[Sigla] as Pais,
c.[Nome] as RegVendas
from [D_GrupoGeografico] as a inner join [D_Pais] as b
on a.[Id_GrupoGeo] = b.[Id_GrupoGeo]
inner join [dbo].[D_RegiaoVendas] as c on b.[Id_Pais] = c.[Id_Pais]

USE [DS]
GO
```


ANEXO 6 – Criação Dimensão Funcionário

```
CREATE TABLE [dbo].[D_Funcionario](
[Nome] [varchar](50) NOT NULL,
[Login] [varchar](50) NOT NULL,
[Id_Chefe] [int] NULL,
[LinhaData] [date] NOT NULL,
[LinhaOrig] [varchar](50) NULL
)

create procedure [dbo].[Carrega_D_Funcionario]
as
begin try
    truncate table [DS]..[D_Funcionario]

    insert into [dbo].[D_Funcionario]
    ([Nome],[Login],[LinhaData],[LinhaOrig])
    Select Distinct
    VendedorNome, VendedorLogin, Getdate() as [LinhaData], 'Arquivo de Vendas' as [LinhaOrig] from
    [DS]..[TbImp_Vendas]

    insert into [DW]..[D_Funcionario]
    Select
    ds.[Nome], ds.[Login], ds.[Id_Chefe], ds.[LinhaData], ds.[LinhaOrig] from [DS]..[D_Funcionario]
    as ds left join [DW]..[D_Funcionario] as dw
    on ds.[Login] = dw.[Login] where dw.Id_Funcionario is null

    declare @Id_funcionario int
    declare @LoginFuncionario varchar(50)
    declare @NomeFuncionario varchar(50)
    declare @NomeChefe varchar(50)
    declare @Id_Chefe int

    declare cur_AtualizaChefe cursor for
        Select Id_Funcionario, Login, Nome from DW..D_Funcionario
    Open cur_AtualizaChefe
    fetch next from cur_AtualizaChefe into @Id_Funcionario, @LoginFuncionario, @NomeFuncionario
    while @@FETCH_STATUS = 0
    Begin
        Set @NomeChefe = (Select Distinct VendedorChefeNome from [DS]..[TbImp_Vendas] as b where
        b.VendedorLogin = @LoginFuncionario)
        Set @Id_Chefe = (Select Id_Funcionario from [DW]..[D_Funcionario] as b where b.Nome = @NomeChefe)
        If @NomeChefe != @NomeFuncionario
        Begin
            update [DW]..[D_Funcionario]
            Set Id_Chefe = @Id_Chefe where Id_Funcionario = @Id_funcionario
        End
    End
    fetch next from cur_AtualizaChefe into @Id_funcionario, @LoginFuncionario, @NomeFuncionario
    End
    close cur_AtualizaChefe
    deallocate cur_AtualizaChefe

    insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Funcionario','S', 'Carga de D_Funcionario
    com sucesso.')

end try
begin catch
insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Funcionario','F', 'Erro ao carregar
D_Funcionario,')
end catch
```

ANEXO 7 – Criação Dimensão Produto

```
USE [DS]
GO

CREATE TABLE [dbo].[D_Produto](
[Cod_Produto] [varchar](20) NOT NULL,
[Nome] [varchar](50) NOT NULL,
[Tamanho] [varchar](5) NOT NULL,
[Cor] [varchar](20) NOT NULL,
[Ativo] [char](1) NOT NULL,
[LinhaData] [date] NOT NULL,
[LinhaOrigem] [varchar](50) NOT NULL
)

GO

create index IX_ProdutoNM on DS..D_Produto (nome)
create index IX_ProdutoNM on DW..D_Produto (nome)
create index IX_ProdutoCod on DS..D_Produto (Cod_Produto)
create index IX_ProdutoCod on DW..D_Produto (Cod_Produto)

create procedure [dbo].[Carrega_D_Produto]
as

begin try
truncate table [DS]..[D_Produto]
insert into [dbo].[D_Produto]
Select Distinct
Cod_Produto as [Cod_Produto],
Produto as [Nome],
Tamanho as [Tamanho],
Cor as [Cor],
Ativo = 1,
Getdate() as [LinhaData],
'Arquivo de Vendas' as [LinhaOrigem] from [DS]..[TbImp_Vendas]

insert into DW..D_Produto
Select
ds.Cod_Produto,
ds.Nome,
ds.Tamanho,
ds.Cor,
ds.Ativo,
ds.LinhaData,
ds.LinhaOrigem
from [DS]..[D_Produto] as ds left join [DW]..[D_Produto] as dw
on ds.Cod_Produto = dw.Cod_Produto where dw.Cod_Produto is null

declare @CodProduto varchar(20)
declare @NomeProduto varchar(50)
declare @Tamanho varchar(5)
declare @Cor varchar(20)
declare @LinhaData date
declare @LinhaOrigem varchar(50)
declare cur_AtualizarProduto cursor for
Select ds.Cod_Produto, ds.Nome, ds.Tamanho, ds.Cor, ds.LinhaData, ds.LinhaOrigem
from DS..D_Produto as ds left join DW..D_Produto as dw on ds.Cod_Produto = dw.Cod_Produto
where ds.Nome != dw.Nome and dw.Ativo = 1
Open cur_AtualizarProduto fetch next from cur_AtualizarProduto into @CodProduto, @NomeProduto,
@Tamanho, @Cor, @LinhaData, @LinhaOrigem while @@FETCH_STATUS = 0
Begin
update dw..D_Produto set Ativo = 0 where Cod_Produto = @CodProduto

insert into DW..D_Produto values (@CodProduto, @NomeProduto, @Tamanho, @Cor, 1, @LinhaData, @LinhaOrigem)

fetch next from cur_AtualizarProduto into @CodProduto, @NomeProduto,@Tamanho, @Cor, @LinhaData, @LinhaOrigem
end
close cur_AtualizarProduto deallocate cur_AtualizarProduto

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Produto','S', 'Carga de D_Produto com
sucesso.')

end try
begin catch

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Produto','F', 'Erro ao carregar
D_Produto.')
end catch
```

ANEXO 8 – Criação Dimensão Data

```
USE [DS]
GO

CREATE TABLE [dbo].[D_Data](
[Data] [date] Primary key NOT NULL,
[Dia] [char](2) NOT NULL,
[Mes] [char](2) NOT NULL,
[Ano] [char](4) NOT NULL,
)

USE [DS]
GO

Create procedure [dbo].[Carrega_D_Data]
as
begin try
    truncate table [DS]..[D_Data]

    insert into [dbo].[D_Data]
    Select distinct
    Cast(DataVenda as date) as Data,
    right('00'+ cast(Day(DataVenda) as varchar(2)),2) as Dia,
    right('00'+ cast(Month(DataVenda) as varchar(2)),2) as Mes,
    Year(DataVenda) as Ano
    from [DS]..[TbImp_Vendas]

    insert into [DW]..[D_Data]
    Select
    ds.data,
    ds.Dia,
    ds.Mes,
    ds.Ano
    from [DS]..[D_Data] as ds left join [DW]..[D_Data] as dw on ds.Data = dw.Data where dw.Data is null

    insert into [dbo].[Adm_Log] values (newid(), getdate(), ' importa Data', 'S', 'Carga de D_Data com sucesso.')

    end try
    begin catch
    insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Data', 'F', 'Erro ao carregar D_Data.')
    end catch

USE DS
GO

exec [dbo].[Carrega_D_Data]
```

ANEXO 9 – Criação Fato Venda

```
USE [DS]
GO
CREATE TABLE [dbo].[F_Venda](
[Data] [date] NOT NULL,
[Nr_NF] [varchar](10) NOT NULL,
[Id_Cliente] [int] NOT NULL,
[Id_Funcionario] [int] NOT NULL,
[Id_RegiaoVendas] [int] NOT NULL,
[Vlr_Imposto] [decimal](18,2) NOT NULL,
[Vlr_Frete] [decimal](18,2) NOT NULL,
[LInData] [date] NOT NULL,
[LInOrig] [varchar](50) NOT NULL,
CONSTRAINT [PK_F_Venda] PRIMARY KEY CLUSTERED
(
[Data] ASC,
[Nr_NF] ASC,
[Id_Cliente] ASC,
[Id_Funcionario] ASC,
[Id_RegiaoVendas] ASC
))

USE [DS]
GO

Create procedure [dbo].[Carrega_F_Venda]
as

begin try
    truncate table [DS]..[F_Venda]

    declare @idclienteGenerico int = (Select Id_Cliente from DW..D_Cliente where Cod_Cliente = '999999')
    declare @idFuncionarioGenerico int = (Select ID_Funcionario from DW..D_Funcionario where Login = 'Não
    Aplica')
    declare @idRegiaoGenerico int = (Select Id_RegiaoVendas from DW..D_RegiaoVendas where Nome = 'Não Aplica')

    insert into DS..F_venda
    Select
    Data,
    Nr_NF,
    Id_Cliente,
    Id_Funcionario,
    Id_RegiaoVendas,
    sum(Vlr_Imposto),
    sum(Vlr_Frete),
    Getdate() as [LInData],
    'Arquivo de Vendas' as [LInOrig]
    from (
    Select Distinct
    isnull (a.DataVenda, '1900/01/01') as Data,
    isnull (a.NrNF, '000000') as Nr_NF,
    isnull (c.Id_Cliente, @idclienteGenerico) as Id_Cliente,
    isnull (d.Id_Funcionario, @idFuncionarioGenerico) as Id_Funcionario,
    isnull (e.Id_RegiaoVendas, @idRegiaoGenerico) as Id_RegiaoVendas,

    cast (ImpTotal as decimal (18,2)) as Vlr_Imposto,
    cast (Frete as decimal (18,2)) as [Vlr_Frete]
    from DS..TbImp_vendas as a left join DW..D_Data as b on a.DataVenda = b.Data
    left join DW..D_Cliente as c on a.CodCliente = c.Cod_Cliente
    left join DW..D_Funcionario as d on a.VendedorLogin = d.Login
    left join DW..D_RegiaoVendas as e on a.RegiaoVendas = e.Nome ) as x
    group by
    Data,
    Nr_NF,
    Id_Cliente,
    Id_Funcionario,
    Id_RegiaoVendas

    insert into DW..F_Venda
    Select

    ds.[Data],
    ds.[Nr_NF],
    ds.[Id_Cliente],
    ds.[Id_Funcionario],
    ds.[Id_RegiaoVendas],
    ds.[Vlr_Imposto],
    ds.[Vlr_Frete],
    ds.[LInData],
    ds.[LInOrig]
    from DS..F_Venda as ds left join DW..F_Venda as dw on ds.Data = dw.Data
    and ds.Nr_NF = dw.Nr_NF
```

```
and ds.Id_Cliente = dw.Id_Cliente
and ds.Id_Funcionario = dw.Id_Funcionario
and ds.Id_RegiaoVendas = dw.Id_RegiaoVendas
where dw.Data is null

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Fatos de Venda','S', 'Carga de F_Venda com
sucesso.')

end try
begin catch
insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Fatos de Venda','F', 'Erro ao carregar
F_Venda.')
end catch

exec [dbo].[Carrega_F_Venda]
```

ANEXO 10 – Criação Fato Detalhe de Venda

```
CREATE TABLE [dbo].[F_VendaDetalhe](
[Data] [date] NOT NULL,
[Nr_NF] [varchar](10) NOT NULL,
[Id_Cliente] [int] NOT NULL,
[Id_Funcionario] [int] NOT NULL,
[Id_RegiaoVendas] [int] NOT NULL,
[Id_Produto] [int] NOT NULL,
[Vlr_Unitario] [decimal](18,2) NOT NULL,
[Qtd_Vendida] [int] NOT NULL,
[LinhaData] [date] NOT NULL,
[LinhaOrig] [varchar](50) NOT NULL,
CONSTRAINT [PK_F_VendaDetalhe] PRIMARY KEY CLUSTERED
(
    [Data] ASC,
    [Nr_NF] ASC,
    [Id_Funcionario] ASC,
    [Id_RegiaoVendas] ASC,
    [Id_Produto] ASC
)
)

USE [DS]
GO

create procedure [dbo].[Carrega_F_VendaDetalhe]
as
begin try

declare @idclienteGenerico int = (Select Id_Cliente from DW..D_Cliente where Cod_Cliente = '999999')
declare @idfuncionarioGenerico int = (Select Id_Funcionario from DW..D_Funcionario where Login = 'Não
Aplica')
declare @idregiaoGenerico int = (Select Id_RegiaoVendas from DW..D_RegiaoVendas where Nome = 'Não Aplica')
declare @idprodutoGenerico int = (Select Id_Produto from DW..D_Produto where Cod_Produto = '999999' and
[Ativo] = '1')

insert into DS..F_VendaDetalhe

Select
Data,
Nr_NF,
Id_Cliente,
Id_Funcionario,
Id_RegiaoVendas,
Id_Produto,
sum(Vlr_Unitario),
sum(Qtd_Vendida),
Getdate() as [LinhaData], 'Arquivo de Vendas' as [LinhaOrig]
from(
    Select Distinct
        isnull(a.DataVenda, '1900/01/01') as Data,
        isnull(a.NrNF, '000000') as Nr_NF,
        isnull(c.Id_Cliente, @idclienteGenerico) as Id_Cliente,
        isnull(d.Id_Funcionario, @idfuncionarioGenerico) as Id_Funcionario,
        isnull(e.Id_RegiaoVendas, @idregiaoGenerico) as Id_RegiaoVendas,
        isnull(f.Id_Produto, @idprodutoGenerico) as Id_Produto,
        isnull(cast(a.PrecoUnitario as decimal(18,2)),0) as Vlr_Unitario,
        isnull(cast(a.Qtd as int),1) as Qtd_Vendida,
        Getdate() as [LinhaData], 'Arquivo de Vendas' as [LinhaOrig] from DS..TbImp_Vendas as a left join
DW..D_Data
as b on a.DataVenda = b.Data left join DW..D_Cliente as c on a.CodCliente = c.Cod_Cliente left join
DW..D_Funcionario
as d on a.VendedorLogin = d.Login left join DW..D_RegiaoVendas as e on a.RegiaoVendas = e.Nome left join
DW..D_Produto as f
on a.Cod_Produto = f.Cod_Produto where f.Ativo = '1'
) as x

Group by
Data,
Nr_NF,
Id_Cliente,
Id_Funcionario,
Id_RegiaoVendas,
Id_Produto

insert into DW..F_VendaDetalhe
Select
ds.[Data],
ds.[Nr_NF],
ds.[Id_Cliente],
ds.[Id_Funcionario],
ds.[Id_RegiaoVendas],
ds.[Id_Produto],
```

```
ds.[Vlr_Unitario],
ds.[Qtd_vendida],
ds.[LinData],
ds.[LinOrig]
from DS..F_VendaDetalhe as ds left join DW..F_VendaDetalhe as dw
on ds.Data = dw.Data
and ds.Nr_NF = dw.Nr_NF
and ds.Id_Cliente = dw.Id_Cliente
and ds.Id_Funcionario = dw.Id_Funcionario
and ds.Id_RegiaoVendas = dw.Id_RegiaoVendas
and ds.Id_Produto = dw.Id_Produto
where dw.Data is null

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Fatos de Detalhe de Venda', 'S', 'Carga de
F_VendaDetalhe com sucesso.')

end try
begin catch

insert into [dbo].[Adm_Log] values (newid(), getdate(), 'Importa Fatos de Venda Detalhe', 'F', 'Erro ao
carregar F_VendaDetalhe.')
end catch
```