

UNIVERSIDADE DE CAXIAS DO SUL
Centro de Computação e Tecnologia da Informação
Curso de Bacharelado em Sistemas de Informação

Rafael de Souza Vieira

IMPLANTAÇÃO DE BUSINESS INTELLIGENCE NO SISTEMA NL GESTÃO

Caxias do Sul

2012

Rafael de Souza Vieira

**IMPLANTAÇÃO DE BUSINESS INTELLIGENCE NO SISTEMA NL
GESTÃO**

Trabalho de Conclusão de
Curso para obtenção do Grau de
Bacharel em Sistemas de
Informação da Universidade de
Caxias do Sul.

Daniel Luís Notari

Orientador

Caxias do Sul

2012

**Dedico este trabalho ao
meu pai Roni Sebastião
Camargo Vieira.**

AGRADECIMENTOS

Além de dedicar também gostaria de agradecer ao meu pai por todo o apoio não só durante este trabalho, mas durante todo o tempo na graduação.

SUMÁRIO

| | |
|---|----|
| Resumo | 5 |
| Lista de Ilustrações | 6 |
| Lista de Quadros | 9 |
| Lista de Abreviaturas e Siglas | 10 |
| 1 Introdução..... | 11 |
| 1.1 Problema de Pesquisa | 11 |
| 1.2 Objetivo | 12 |
| 1.3 Organização do Trabalho..... | 13 |
| 2 Business Intelligence..... | 14 |
| 2.1 Data Warehouse | 16 |
| 2.1.1 Modelagem do <i>Data Warehouse</i> | 17 |
| 2.1.2 Características do <i>Data Warehouse</i> | 22 |
| 2.1.3 Granularidade | 23 |
| 2.1.4 Arquitetura do <i>Data warehouse</i> | 23 |
| 2.1.5 Carga dos Dados em um <i>Data Warehouse</i> | 24 |
| 2.1.6 Modelagem Dimensional..... | 25 |
| 2.2 Processamento Analítico Online (OLAP)..... | 28 |
| 2.2.1 Drill-down e Roll-Up..... | 29 |
| 2.2.2 Slice e Dice | 30 |
| 2.3 Considerações Finais | 31 |
| 3 Ferramentas | 33 |
| 3.1 Microsoft SQL Server Integration Services (SSIS)..... | 33 |
| 3.2 Microsoft SQL Server Analysis Services (SSAS)..... | 38 |
| 3.3 Tabelas dinâmicas do Microsoft Excel..... | 45 |
| 3.4 Considerações Finais | 50 |

| | | |
|-------|---|-----|
| 4 | Modelagem do BI..... | 52 |
| 4.1 | Arquitetura do Sistema de Gestão de Vendas..... | 52 |
| 4.2 | Definição do Processo de Negócio | 54 |
| 4.3 | Modelo Multidimensional..... | 55 |
| 4.4 | Tabelas de Fato | 56 |
| 4.5 | Definição das Dimensões..... | 57 |
| 4.6 | Considerações Finais | 61 |
| 5 | CRIAÇÃO DO BI..... | 63 |
| 5.1 | Criação do Processo de Carga dos Dados..... | 64 |
| 5.1.1 | ETL das Dimensões | 65 |
| 5.1.2 | ETL Vendas | 67 |
| 5.1.3 | ETL de Vendas por Forma de Pagamento..... | 77 |
| 5.2 | Carga dos Dados no DW | 80 |
| 5.3 | Criação do Cubo de Dados | 83 |
| 5.4 | Considerações Finais | 88 |
| 6 | Resultados obtidos..... | 91 |
| 6.1 | Horários de Picos de Vendas | 91 |
| 6.2 | Vendas por Forma de Pagamento | 92 |
| 6.3 | Ciclo de Vida de Produtos | 93 |
| 6.4 | Vendas por Unidade da Empresa..... | 94 |
| 6.5 | Vendas por Região..... | 95 |
| 6.6 | Considerações Finais | 96 |
| 7 | Conclusão..... | 98 |
| 7.1 | Trabalhos Futuros | 99 |
| 8 | Bibliografia..... | 100 |

RESUMO

No cenário atual grande parte das empresas possui um número relativamente grande de dados armazenados nas suas bases de dados operacionais. Estes dados, se trabalhados de maneira correta, podem gerar informações de apoio à tomada de decisões gerando um potencial estratégico. Para trabalhar estes dados e transformá-los em informações é utilizado o conceito de *Business Intelligence* (BI), que nada mais é do que um conjunto de arquiteturas, ferramentas, bancos de dados, aplicações e metodologias cujo principal objetivo é permitir o acesso interativo aos dados, proporcionar a manipulação dos dados e fornecer aos gerentes e analistas de negócios a capacidade de realizar a análise adequada destes dados. Este trabalho tem por objetivo implantar *Business Intelligence* em um sistema chamado NL Gestão produzido pela empresa NL Informática. Para cumprir este objetivo serão utilizadas as ferramentas de BI produzidas pela empresa Microsoft.

Palavras chave: Business Intelligence, Data Warehouse, data mart, Analysis Services, OLAP.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Ilustração 1 Arquitetura de Business Intelligence | 15 |
| Ilustração 2 Processo de Business Intelligence | 16 |
| Ilustração 3: Elementos básicos de um Data Warehouse | 17 |
| Ilustração 4: Modelo dimensional | 19 |
| Ilustração 5: Tipos de modelo dimensional | 19 |
| Ilustração 6 Dimensão de degeneração..... | 21 |
| Ilustração 7: Arquiteturas de <i>Data warehouse</i> | 24 |
| Ilustração 8: Processo de ETL | 25 |
| Ilustração 9 Processo de modelagem dimensional | 26 |
| Ilustração 10: Técnicas OLAP | 29 |
| Ilustração 11: Drill-down e Roll-Up..... | 30 |
| Ilustração 12: Slice | 30 |
| Ilustração 13: Dice | 31 |
| Ilustração 14 Tela de projeto do Microsoft <i>Integration Services</i> | 34 |
| Ilustração 15 Hierarquia de componentes do SSIS..... | 34 |
| Ilustração 16 Acesso à ferramenta Import and Export Wizard no SSMS | 35 |
| Ilustração 17 Criação de projeto do SSIS no BIDS | 36 |
| Ilustração 18 Tela principal do BIDS com um projeto SSIS..... | 36 |
| Ilustração 19 Exemplo de Fluxo de Controle | 37 |
| Ilustração 20 Exemplo de fluxo de dados..... | 37 |
| Ilustração 21 Arquitetura do Analysis Services..... | 39 |
| Ilustração 22 Particionamento do cubo de dados em diferentes servidores | 40 |
| Ilustração 23 Configurações do Data Source no SSAS | 42 |
| Ilustração 24 Designer de um Data Source View | 43 |
| Ilustração 25 Cube Designer no BIDS..... | 43 |
| Ilustração 26 Cube Designer - Paleta Dimension Usage | 44 |
| Ilustração 27 Cube Designer - Paleta Partitions | 44 |
| Ilustração 28 Cube Designer - Paleta Browser | 45 |
| Ilustração 29 Conexão do Excel com os serviços de análise do SSAS | 46 |

| | |
|---|----|
| Ilustração 30 Configuração do servidor de dados..... | 47 |
| Ilustração 31 Seleção de cubo para análise de dados no Excel | 47 |
| Ilustração 32 Tela inicial da tabela dinâmica | 48 |
| Ilustração 33 Seleção de segmentação de dados..... | 48 |
| Ilustração 34 Seleção de atributos para segmentação de dados..... | 49 |
| Ilustração 35 Exemplo de tabela dinâmica com segmentação de dados e gráfico | 50 |
| Ilustração 36 Arquitetura de rede de clientes..... | 53 |
| Ilustração 37 Fluxo de dados Gestão de Vendas | 54 |
| Ilustração 38 Fluxo de dados do Sistema de gestão de vendas..... | 54 |
| Ilustração 39 Cubo de Vendas | 55 |
| Ilustração 40 Fato de Vendas..... | 56 |
| Ilustração 41 Fato de Vendas por Forma de Pagamento | 57 |
| Ilustração 42 Dimensão de Unidade | 58 |
| Ilustração 43 Dimensão de Data | 59 |
| Ilustração 44 Dimensão de hora | 59 |
| Ilustração 45 Dimensão de produto | 60 |
| Ilustração 46 Dimensão de Operador | 60 |
| Ilustração 47 Dimensão de Vendedor..... | 61 |
| Ilustração 48 Dimensão de Forma de Pagamento..... | 61 |
| Ilustração 49 Tabelas do banco de dados NLDW..... | 63 |
| Ilustração 50 Chave da dimensão de Data com auto-incremento..... | 64 |
| Ilustração 51 Fluxo de carga de dimensão..... | 65 |
| Ilustração 52 Fluxo da ETL de dimensão de unidade..... | 66 |
| Ilustração 53 Configuração de saída de erro..... | 66 |
| Ilustração 54 ETL para o fato de Vendas | 67 |
| Ilustração 55 Script SQL para consulta de dados para o Fato de Vendas | 68 |
| Ilustração 56 Configurações de colunas derivadas na ETL de Vendas | 69 |
| Ilustração 57 Conversão de tipos de dados do Oracle para o SQL Server | 69 |
| Ilustração 58 Consulta de chave substituta na Dimensão de Data..... | 70 |
| Ilustração 59 Consulta de chave substituta na dimensão de hora | 71 |
| Ilustração 60 Consulta de chave substituta na dimensão de unidade | 71 |
| Ilustração 61 Consulta de chave substituta na dimensão de produto..... | 72 |
| Ilustração 62 Processo de consulta de chave substituta para Dimensão de Operador .. | 73 |

| | |
|--|----|
| Ilustração 63 Adição da coluna derivada KeyOperador | 73 |
| Ilustração 64 Conversão do campo KeyOperador de inteiro para numérico | 74 |
| Ilustração 65 Mapeamento de colunas para união do fluxo de dados do operador | 75 |
| Ilustração 66 Configuração da agregação de vendas | 76 |
| Ilustração 67 Configuração de conexão do Fato de Vendas | 76 |
| Ilustração 68 Mapeamento de colunas para tabela de fato de Vendas..... | 77 |
| Ilustração 69 Processo macro de ETL de vendas por forma de pagamento | 78 |
| Ilustração 70 Configuração de agregação para o fato de vendas por forma de pagamento..... | 79 |
| Ilustração 71 Configuração da consulta de chave substituta na dimensão de formas de pagamento..... | 79 |
| Ilustração 72 Mapeamento de colunas para inserção dos dados no fato de vendas por forma de pagamento | 79 |
| Ilustração 73 Execução da ETL para o fato de vendas..... | 81 |
| Ilustração 74 Tempo de execução da ETL do fato de vendas | 81 |
| Ilustração 75 Execução de ETL para o fato de vendas por forma de pagamento..... | 82 |
| Ilustração 76 Criação do projeto de Cubo de Vendas no BIDS..... | 83 |
| Ilustração 77 Data Sources do projeto | 83 |
| Ilustração 78 Estrutura Data Source View..... | 84 |
| Ilustração 79 Tela Inicial do Cubo de Vendas | 85 |
| Ilustração 80 Máscara de valor monetário para medidas..... | 85 |
| Ilustração 81 Edição da dimensão de Unidade | 86 |
| Ilustração 82 Hierarquia da dimensão de Unidade com granularidade por município..... | 86 |
| Ilustração 83 Hierarquia de Unidade | 87 |
| Ilustração 84 Hierarquia de data | 87 |
| Ilustração 85 Hierarquia de produto | 88 |
| Ilustração 86 Partições do cubo | 88 |
| Ilustração 87 Análise de horários de pico de vendas | 91 |
| Ilustração 88 Análise de Vendas por forma de pagamento | 92 |
| Ilustração 89 Análise de ciclo de vida de produto | 94 |
| Ilustração 90 Análise de vendas por unidade | 95 |
| Ilustração 91 Drill-Down na análise de volume de vendas | 95 |
| Ilustração 92 Análise de vendas por Região..... | 96 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1: Comparativo entre tipos de modelo Dimensional | 20 |
| Quadro 2 Vendas por Ano | 27 |
| Quadro 3 Vendas por ano e mês | 27 |
| Quadro 4 Configurações do servidor de ETL..... | 80 |
| Quadro 5 Resultado da ETL de dimensões..... | 80 |
| Quadro 6 Resultado da ETL do fato de vendas | 81 |
| Quadro 7 Resultado da ETL para o fato de vendas por forma de pagamento | 82 |

LISTA DE ABREVIATURAS E SIGLAS

| Sigla | Significado em Português | Significado em Inglês |
|--------------|--|--|
| BI | Inteligência de Negócios | Business Intelligence |
| BIDS | Ferramenta de Desenvolvimento de Inteligência de Negócios | Business Intelligence Development Studio |
| CID | Classificação Internacional de Doenças | |
| DSV | Visão de Fonte de Dados | Data Source View |
| DW | Armazém de Dados | Data Warehouse |
| ETL | Extração, Transformação e Carga | Extract, Transform and Load |
| HOLAP | Processamento Analítico Online Híbrido | Hybrid Online Analytical Processing |
| IDE | Ambiente de desenvolvimento Integrado | Integrated Development Environment |
| KPI | Principais Indicadores de Desempenho | Key Performance Indicator |
| MDX | Expressões Multidimensionais | Multidimensional Expressions |
| MOLAP | Processamento Analítico Online Multidimensional | Multidimensional Online Analytical Processing |
| OLAP | Processamento Analítico em tempo real | Online Analytical Processing |
| OLTP | Processamento de transações em tempo real | Online Transaction Processing |
| ROLAP | Processamento Analítico Online Relacional | Relational Online Analytical Processing |
| SQL | Linguagem de consulta estruturada | Structured Query Language |
| SSAS | Serviço de Análise do SQL Server | SQL Server Analysis Services |
| SSIS | Serviço de Integração do SQL Server | SQL Server Integration Services |
| SSMS | Ferramenta de Gerenciamento do SQL Server | SQL Server Management Studio |

1 INTRODUÇÃO

Na sociedade atual, quem possui a melhor informação conseguirá realizar o melhor negócio. As empresas investem muito dinheiro em sistemas de informação, visando ganhar tempo nas tarefas e gerenciar melhor sua organização. Com a preocupação da realização das tarefas operacionais, os sistemas deixaram em segundo plano a informação estratégica, concentrando-se apenas na manipulação de dados operacionais. Estes dados resultantes dos processos operacionais, se manipulados de maneira correta, podem gerar informações decisivas para a organização (BALLARD, *et al.*, 2006).

Para solucionar o problema este problema, as organizações vêm utilizando o conceito de *Business Intelligence* (BI), cujo objetivo é permitir o acesso interativo aos dados, proporcionar manipulação destes dados e fornecer aos analistas de negócios e gerentes a capacidade de analisar estes dados de forma adequada (TURBAN, *et al.*, 2009).

O processo de *Business Intelligence* acontece com a criação de um repositório de dados com informações adequadas ao negócio, à utilização de ferramentas específicas para criar visualizações destes dados que possam ser utilizadas pelos gestores e finalmente a avaliação destas informações por parte dos gestores na tomada de decisão (BALLARD, *et al.*, 2006).

Quando se fala na criação de um repositório de dados com informações adequadas ao negócio, tratamos do conceito de *data warehouse* que nada mais é do que um banco de dados com informações obtidas e transformadas a partir de diversos bancos de dados operacionais, e que têm por objetivo auxiliar no processo de criação de relatórios gerenciais e apoiar a tomada de decisão (INMON, 2005).

Nas seções a seguir será explicado o problema de pesquisa, objetivo e organização deste trabalho.

1.1 Problema de Pesquisa

O sistema NL Gestão é um sistema ERP (*Enterprise Resource Planning*) desenvolvido pela empresa NL Informática. Dentre os diversos módulos que compõe este sistema, existe o

módulo de Gestão Vendas, que é utilizado para agilizar o processo de vendas em empresas de varejo, através de uma solução de PDV (Ponto de Venda) e Retaguarda.

Certas informações da área de gestão de vendas podem ser significativamente relevantes na estratégia de negócio da empresa, informações estas, que podem ser visualizadas de forma interativa em uma ferramenta de *Business Intelligence*. Seguem alguns exemplos destas informações são:

1. Total de vendas por hora;
2. Total de vendas por vendedor;
3. Total de vendas por grupo de produtos;
4. Total de vendas por forma de pagamento;
5. Total de vendas por caixa e hora;
6. Total de vendas por período;
7. Total de cancelamento de vendas;
8. Total de descontos concedidos.

Algumas informações citadas acima podem ser obtidas através do cruzamento de diversos relatórios no sistema NL Gestão, outras não existem, para obtê-las é necessário o desenvolvimento de relatórios customizados.

Através dos relatórios atuais, ou desenvolvimento de novos relatórios, além de existir o custo de desenvolvimento de relatórios, existe o custo com o tempo do usuário interpretando informações de diferentes fontes, sem contar que a confiabilidade de informações obtidas de diferentes locais nem sempre é garantida.

1.2 Objetivo

Este trabalho tem por objetivo aplicar o conceito de *Business Intelligence* à ferramenta chamada NL Gestão produzida pela empresa NL Informática, e através disto, diminuir a necessidade dos usuários solicitarem novos relatórios ou customização dos relatórios atuais para a empresa.

Este objetivo será atingido com o desenvolvimento de um repositório de dados centralizado (*data warehouse*) onde os usuários do sistema poderão conectar ferramentas de visualização de dados de forma que possam trabalhar os dados de forma interativa.

1.3 Organização do Trabalho

O trabalho está organizado em oito capítulos. No capítulo 2 é conceituado o *Business Intelligence*, relatando todos os processos envolvidos na utilização deste conceito. No capítulo 3, são conceituadas as ferramentas utilizadas no desenvolvimento deste trabalho, além de introduzir as ferramentas, são descritas as principais funcionalidades de cada uma. No capítulo 4 são apresentados os passos de modelagem deste trabalho. No capítulo 5 é apresentado o processo de criação do BI, este capítulo contém a fase de desenvolvimento do trabalho. No capítulo 6 são apresentados os resultados obtidos com a utilização de *Business Intelligence*. No capítulo 7 é apresentada a conclusão deste trabalho e trabalhos futuros, e no capítulo 8 as referências bibliográficas utilizadas durante o desenvolvimento deste trabalho.

2 BUSINESS INTELLIGENCE

Business Intelligence (BI) é um termo novo, porém não é um conceito novo. O termo *Business Intelligence* foi lançado pelo Grupo Gartner por volta de 1990. Este termo é um termo “guarda-chuva” que inclui arquiteturas, ferramentas, bancos de dados, aplicações e metodologias. Significa coisas diferentes para pessoas diferentes, onde os principais objetivos são permitir o acesso interativo aos dados, proporcionar a manipulação dos dados e fornecer aos gerentes e analistas de negócios a capacidade de realizar a análise adequada. O processo de BI baseia-se na transformação de dados em informações, depois em decisões e finalmente em ações (TURBAN, *et al.*, 2009).

De forma simplificada, ele é um processo onde o usuário obtém informações consistentes de negócios a partir de um *data warehouse*, analisa esta informação de acordo com o contexto do seu negócio, identifica anomalias ou problemas, realiza previsões e/ou simulações, e por final, toma as decisões de forma mais rápida baseado em informações consistentes (BALLARD, *et al.*, 2006).

Para que o BI se torne realidade, é necessário em um primeiro momento definir quais são as informações relevantes para a empresa e modelar este BI definindo o cubo de dados. Para o preenchimento deste cubo com dados, são necessárias técnicas e ferramentas de *data warehouse*. Após o preenchimento deste cubo, o analista de negócios ou o tomador de decisão poderá trabalhar estes dados para obter a melhor visualização da informação através do processamento analítico online (OLAP).

Na Ilustração 1 é possível verificar a arquitetura do BI em quatro camadas. A primeira camada é a camada de dados, onde encontram-se as bases de dados operacionais na empresa. Na segunda camada, encontra-se o *Data Warehouse*, com os dados integrados e limpos. Estes dados estão armazenados em outro servidor com outro sistema gerenciador de banco de dados (SGBD) de melhor desempenho e capacidade. Na terceira camada encontra-se o servidor OLAP, aonde existe uma base de dados contendo os dados do *data warehouse* processados, ou seja, dados sumarizados, indexados e trabalhados de forma que possam ser melhor analisados. Na última camada, está a camada de visualização dos dados. Nesta camada são utilizadas ferramentas de relatório, ferramentas de KPI (*Key Performance Indicators*), ferramentas de análise e cruzamento de dados.

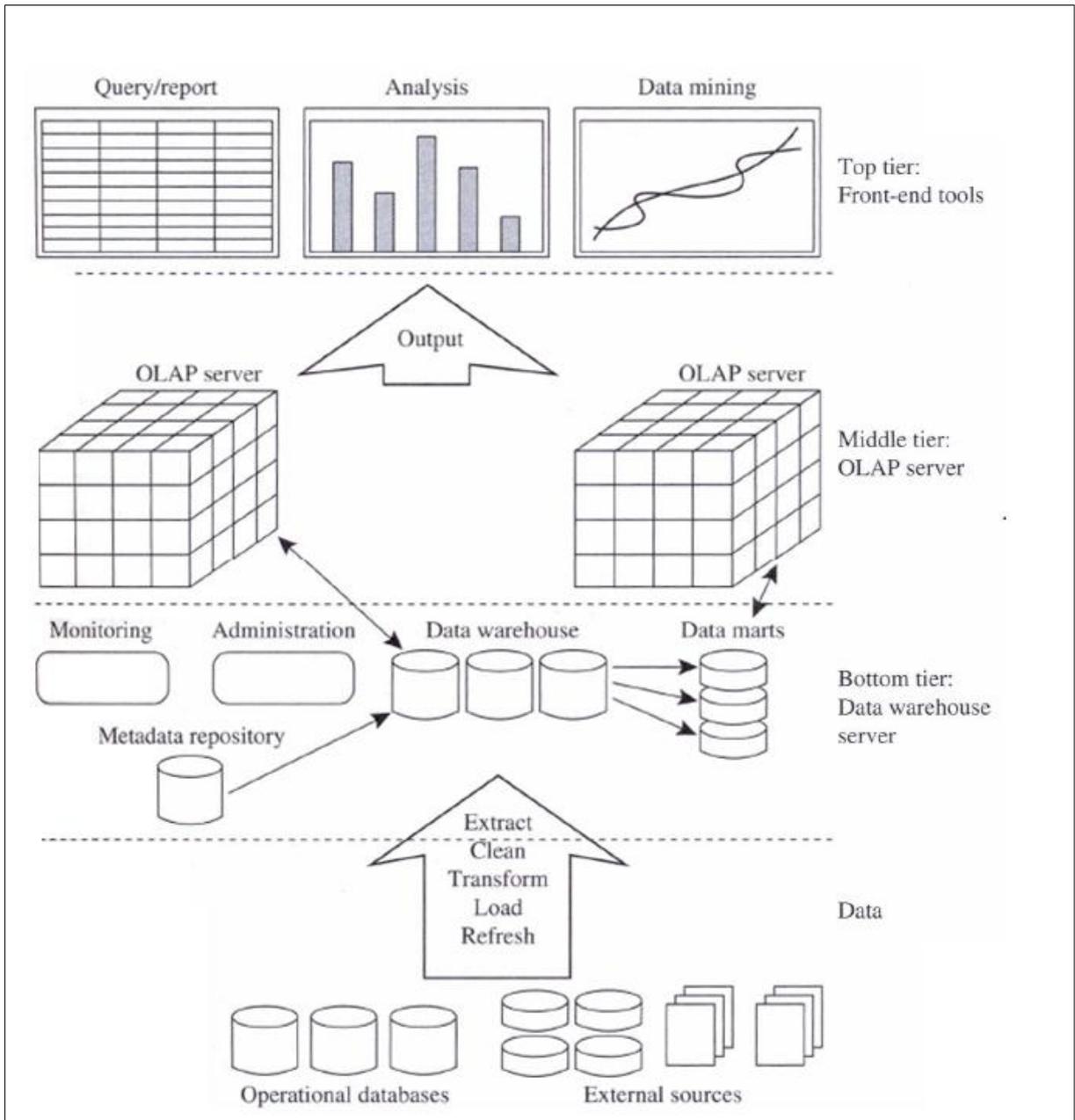


Ilustração 1 Arquitetura de Business Intelligence

Fonte: (JIAWEI, KAMBER e PEI, 2012)

Considerando o modelo em quatro camadas apresentado na Ilustração 1, percebe-se o processo de BI exibido na Ilustração 2.

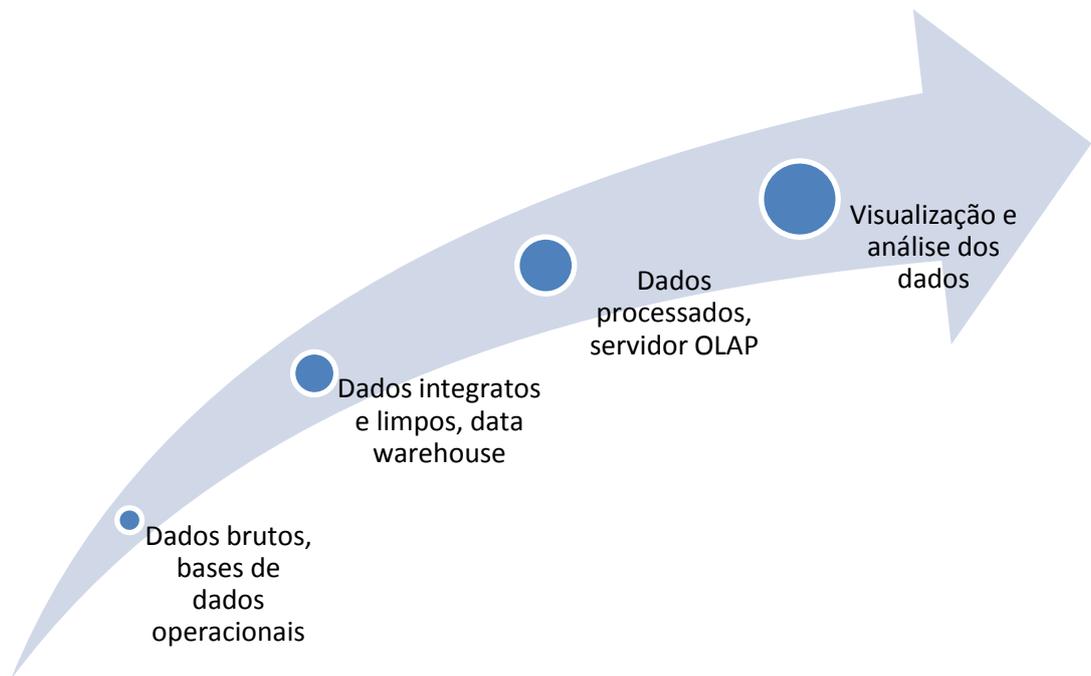


Ilustração 2 Processo de Business Intelligence

Nas seções a seguir são descritos os principais componentes de BI. Primeiro é introduzido o conceito de *data warehouse*, são explicados os modelos existentes, características, arquitetura, modelagem e processo de carga de dados. Após explicar os conceitos necessários ao entendimento de *data warehouse*, são explicados os conceitos de processamento analítico online e as operações possíveis de ser realizadas utilizando este conceito.

2.1 Data Warehouse

As primeiras tecnologias utilizadas pelas empresas na tentativa de melhorar o processo de decisão foram baseadas em computadores pessoais, planilhas eletrônicas e bancos de dados pessoais. Através destas tecnologias, usuários mais habilidosos retiravam dados dos sistemas operacionais da empresa, produzindo planilhas eletrônicas para servir como base a relatórios e gráficos, que seriam utilizados posteriormente auxiliando os tomadores de decisão. Tentando resolver estas questões, as empresas começaram a desenvolver repositórios de dados centralizados. Estes repositórios ficaram conhecidos como *data warehouse* (BALLARD, *et al.*, 2006).

Resumidamente pode-se entender *data warehouse* como um banco de dados reunido a partir de muitos sistemas, destinado a oferecer suporte à produção de relatórios gerenciais e à tomada de decisão (INMON, 2005).

Para a construção de um *data warehouse* é necessário primeiro modelar como os dados deverão ser apresentados, após a modelagem, deve ser definido um processo para realizar a carga dos dados a partir dos bancos de dados operacionais da empresa. Este processo de carga em um primeiro momento pode parecer simples, porém inclui tarefas de limpeza, padronização e transformação dos dados antes de realizar a carga dos dados (INMON, 2005).

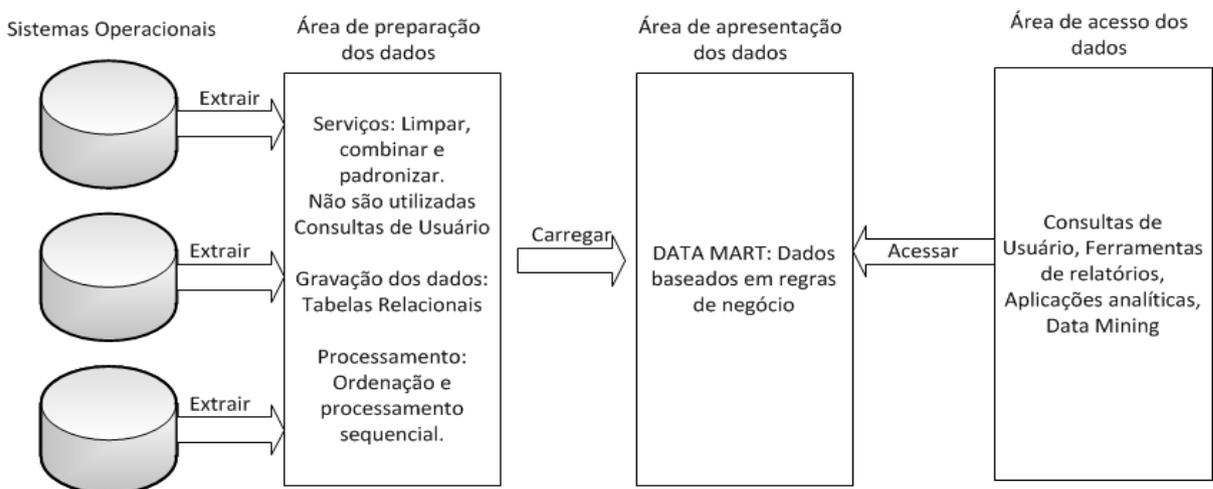


Ilustração 3: Elementos básicos de um Data Warehouse

Fonte: KIMBALL e ROSS, 2002

Na Ilustração 3 é possível visualizar os elementos básicos de um *data warehouse*, os dados sendo extraídos dos bancos de dados operacionais, em seguida carregados para a área de preparação dos dados onde é realizado o processo de transformação e em seguida a carga para *data marts* onde será realizado o processo de consulta e manipulação por parte dos analistas de negócio.

Nas seções a seguir são explicadas técnicas de modelagem e carga dos dados no *data warehouse*.

2.1.1 Modelagem do *Data Warehouse*

Na modelagem de dados se define as estruturas em que os dados serão armazenados, impactando diretamente no desempenho do banco de dados e na usabilidade destes dados. Um modelo é uma abstração da realidade, do processo de negócio. Através da modelagem somos

capazes de ver como ficará o processo de negócio pronto, e também validar os requisitos deste processo (BALLARD, *et al.*, 2006).

Para modelagem de dados é utilizado o modelo Entidade Relacionamento (ER). Este modelo é largamente utilizado na definição dos bancos de dados operacionais das empresas, em seu mais alto nível de normalização. Utilizar um nível alto de normalização significa dividir grandes tabelas em diversas pequenas tabelas visando diminuir a redundância das informações. Esta técnica é conhecida como terceira forma normal (3NF) (INMON, 2005).

No caso do *data warehouse*, a normalização dos dados acaba se tornando um problema, pois quando um analista de negócios tenta criar um relatório onde precisa unir oito ou mais tabelas, ele acaba sentindo dificuldade, ou talvez, nem consiga realizar esta união, o que para um usuário técnico não seria um grande problema, porém, o analista de negócios visa à informação e não é um grande conhecedor de técnicas de bancos de dados, portanto, a tarefa torna-se bastante complexa (BALLARD, *et al.*, 2006).

Para resolver este problema, é utilizado o modelo dimensional, também conhecido como estrela. Ele possui este nome, pois sua representação assemelha-se a uma estrela, onde no centro existe uma tabela com dados de um nível maior, e nas pontas, tabelas ligadas com dados de nível mais baixo (INMON, 2005). Esta técnica consiste em remover a normalização das tabelas do banco de dados, criando tabelas de fatos com os dados mais significativos e tabelas de dimensões para armazenar informações um pouco menos relevantes sobre estes dados. Através desta técnica as consultas realizadas no banco de dados possuem um desempenho muito melhor do que no modelo entidade relacionamento tradicional (KIMBALL e ROSS, 2002).

Através da Ilustração 4 é possível visualizar uma estrela, onde no centro encontra-se tabela de fatos e nas pontas da estrela as tabelas de dimensão com dados referentes a este fato. Basicamente existem três tipos de modelo dimensional (BALLARD, *et al.*, 2006) (Ilustração 5):

1. Modelo estrela;
2. Modelo floco de neve;
3. Modelo multi-estrela.

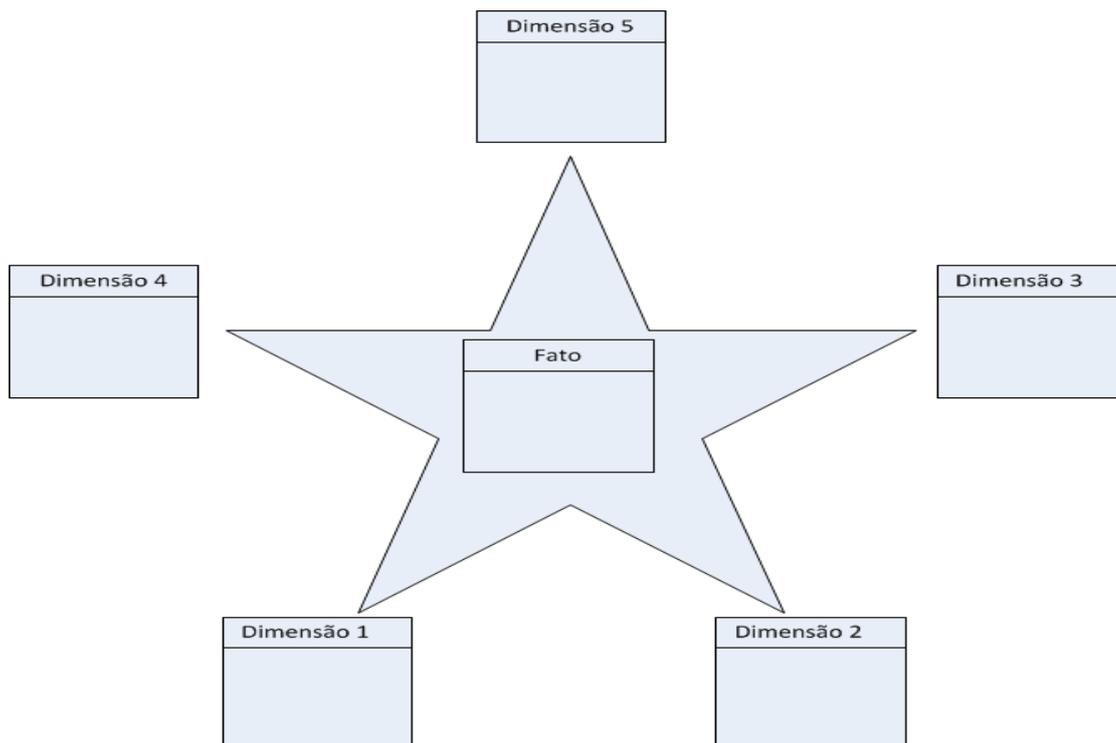


Ilustração 4: Modelo dimensional

O modelo estrela consiste em uma única tabela de fatos, e diversas tabelas de dimensão não normalizadas. O modelo floco de neve utiliza de técnicas de normalização em suas tabelas de dimensão. Este modelo é uma extensão do modelo estrela com dimensões normalizadas. Com a normalização das tabelas de dimensão o modelo se assemelha a um floco de neve, daí o nome do modelo. Já o modelo multi-estrela possui diversas tabelas de fatos unidas através de dimensões. Na Ilustração 5 fica clara a ligação de uma mesma tabela de dimensão com duas tabelas de fatos distintas. (BALLARD, *et al.*, 2006).

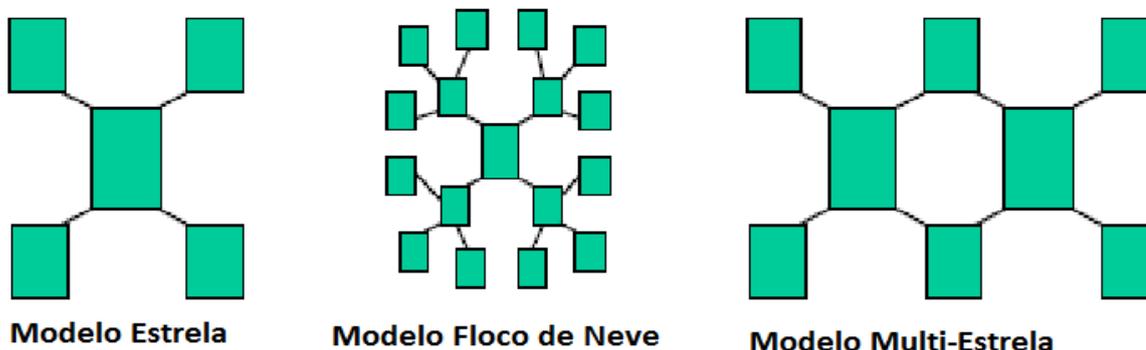


Ilustração 5: Tipos de modelo dimensional

Fonte: BALLARD, *et al.*, 2006

O Quadro 1 mostra um comparativo entre as principais características dos modelos dimensionais.

Quadro 1: Comparativo entre tipos de modelo Dimensional

| Características | Modelo Estrela | Modelo Floco de Neve | Modelo Multi-Estrela |
|------------------------|-----------------------|-----------------------------|-----------------------------|
| Tabela de Fatos | Única | Única | Diversas |
| Dimensões Normalizadas | Não | Sim | Não |
| Dimensões Relacionadas | Não | Sim | Não |

Uma característica comum a todos os modelos é a de que em alguns casos é utilizada uma dimensão abstrata dentro da tabela de fatos. Este tipo de dimensão é chamado de dimensão de degeneração. Para entender o funcionamento desta dimensão, pode-se considerar um cenário onde em alguns casos como em uma tabela de fato de vendas, para cada produto vendido é necessário armazenar o número do cupom fiscal ou nota fiscal que originou a venda deste item. Em um esquema de banco de dados tradicional, o número do cupom seria a chave do registro mestre do item vendido, registro este que conteria informações como a data da venda e a loja em que o item foi vendido por exemplo.

As informações citadas anteriormente provavelmente já se encontram em dimensões de unidade e data, portanto para a dimensão de cupom fiscal restaria apenas o número do cupom. Para evitar a criação de uma tabela de dimensão com somente o atributo chave, usa-se o conceito de dimensão de degeneração, onde é criado um atributo de chave na tabela fato, porém este atributo não tem relação com nenhuma tabela, ele serve somente para agrupar os itens na tabela de fatos (KIMBALL e ROSS, 2002).

O objetivo da utilização das dimensões de degeneração é simplesmente de eliminar a necessidade de uma tabela de dimensão contendo um único atributo. Economiza-se um pouco de espaço em disco e se elimina a necessidade de criação de um novo índice no banco de dados para uma informação duplicada visto que será somente uma cópia da tabela de fatos.

Na Ilustração 6 pode-se verificar no centro a tabela de fatos contendo quatro atributos com a notação “FK” à direita, e um atributo com a notação “DD”. Os que possuem FK são atributos chave que referenciam tabelas de dimensões. O atributo que possui DD é um atributo de Dimensão de Degeneração (DD), ou seja, não referencia nenhuma tabela de dimensão, porém faz parte da chave primária da tabela de fatos.

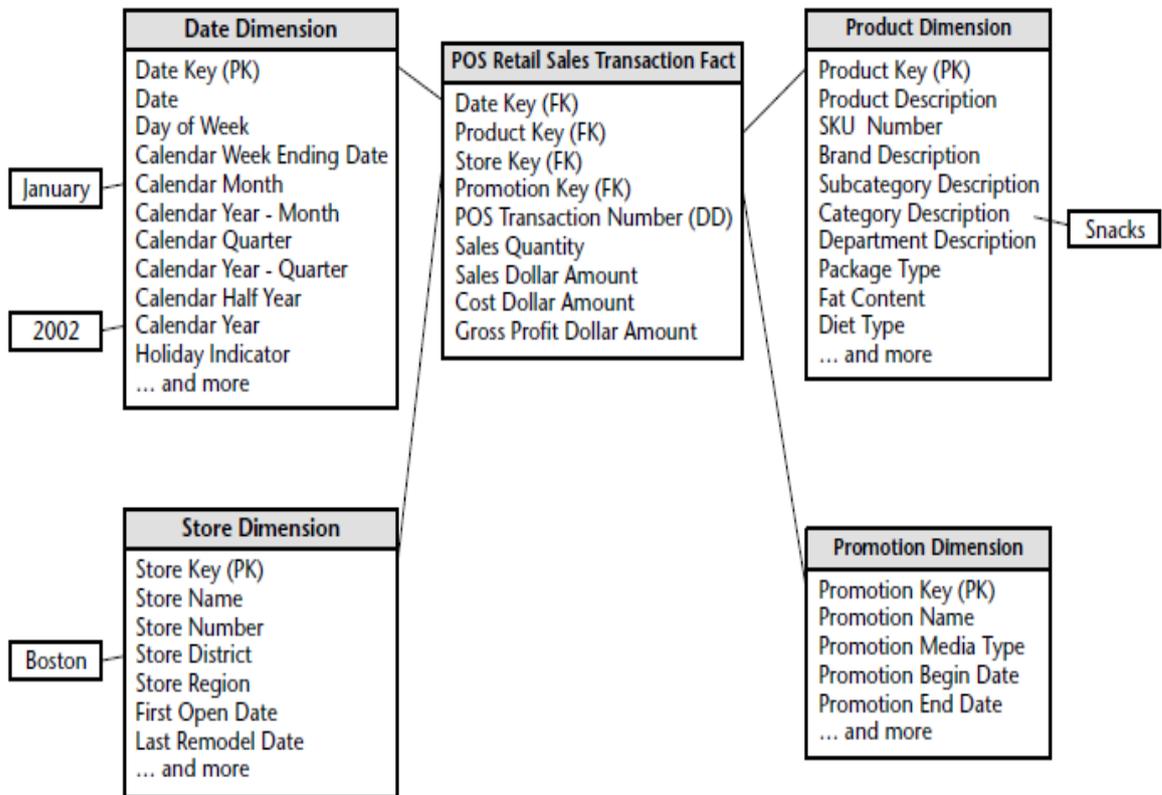


Ilustração 6 Dimensão de degeneração

Fonte: (KIMBALL e ROSS, 2002)

Outra característica comum a todos os modelos de DW é a utilização de chaves substitutas. Chaves substitutas são inteiros atribuídos sequencialmente ao atributo de chave primária da tabela de dimensão, por exemplo, em uma dimensão de produto, o primeiro registro receberia o valor 1, o segundo 2 e assim sucessivamente. O objetivo destas chaves é de ligar as tabelas de dimensão à tabela de fatos evitando que sejam feitas suposições sobre os dados simplesmente olhando para a chave da tabela de fatos. As chaves substitutas permitem a integração de dados de vários sistemas, de forma que não haja sobreposição de chaves, e também previne o problema de não haver chaves de origem consistentes (KIMBALL e ROSS, 2002). Existe uma vantagem desempenho na utilização de chaves substitutas, já que em parte dos sistemas as chaves geralmente são literais podendo conter diversos caracteres, a chave substituta será o menor inteiro possível, logo a tabela de fatos também será menor utilizando estas chaves. Como as dimensões são o meio de acesso aos dados, à utilização de chaves substitutas garante que o usuário não acesse os dados por outros meios, em vista de que os dados não ficarão claros se não forem acessados através da dimensão.

2.1.2 Características do *Data Warehouse*

Existem dois aspectos principais para a construção de um *data warehouse*: um é a interface com os bancos de dados operacionais, e outro o próprio projeto. O projeto caracteriza um processo incremental, pois num primeiro momento são carregados alguns dados. Então o usuário final os utiliza, analisa e dá uma resposta ao projetista sobre quais novos dados devem ser inseridos ou quais dados devem ser reformulados. Este ciclo existe durante toda a vida do *data warehouse* (INMON, 2005).

Pode ser considerado como as principais características de um *data warehouse* o fato de ele ser baseado em negócios, integrado, não volátil, e variável em relação ao tempo (INMON, 2005). Cada uma destas características é explicada a seguir:

1. **Baseado em negócios** - Esta característica se deve ao fato de que enquanto os bancos de dados operacionais são organizados entre as aplicações da empresa, o *data warehouse* é organizado em torno dos negócios destas aplicações. Por exemplo, numa empresa de seguros existem aplicações como automóvel, saúde e vida. Os principais negócios podem ser cliente, apólice, prêmio e indenização (INMON, 2005).
2. **Integrado** - Talvez esta seja a mais importante característica do DW. Durante anos as empresas construíram sistemas utilizando diferentes tipos e formatos de dados, quando é necessário realizar a carga destes dados ao DW, estes dados são corrigidos e padronizados, um exemplo disso pode ser o formato de dados de data, onde em alguns sistemas é utilizado o padrão ano, mês e dia (YYYYMMDD), e em outros, dia, mês e ano (DDMMYYYY). Para o *data warehouse*, deve ser definido um formato específico e então carregar os dados utilizando este formato (SPERLEY, 1999). Esta padronização dos dados é o que gera a característica de integração. Informações de diferentes sistemas são consolidadas em um único formato.
3. **Não volátil** - Nos bancos de dados operacionais, um mesmo dado pode ser acessado e atualizado durante o tempo. Logo, um dado que foi acessado em um ano, quando for acessado no próximo pode não ser igual devido há alguma atualização. Em um *data warehouse* os dados podem ser acessados, porém não podem ser atualizados, isto garante a não volatilidade dos dados.

4. **Variável em relação ao tempo** - Quando existe uma atualização de um dado, deve ser gravado um novo registro contendo informações atuais deste dado no *data warehouse*, portanto, ao longo do tempo, são armazenadas diversas versões de um dado, permitindo assim verificar a informação precisa de acordo com a época. Tudo isto é possível porque as chaves dos dados no *data warehouse* sempre possuem um elemento de tempo (KIMBALL e CASERTA, 2004).

2.1.3 Granularidade

A granularidade diz respeito ao nível de detalhe ou de agrupamento contido nas unidades de dados existentes no *data warehouse* (INMON, 2005).

Quanto mais detalhes, mais baixo é o nível de granularidade, quanto menos detalhe, mais alta é a granularidade. Para exemplificar este conceito basta imaginar uma simples transação de vendas, ela representa um baixo nível de granularidade, porém um registro contendo o total acumulado de todas as transações de vendas em um mês representa um alto nível de granularidade (KIMBALL e ROSS, 2002).

A granularidade dos dados é tão importante que impacta tanto no volume de dados armazenados pelo *data warehouse* quanto no tipo de consulta que poderá ser feito no mesmo (BALLARD, *et al.*, 2006).

2.1.4 Arquitetura do *Data warehouse*

Existem basicamente três tipos de arquitetura de *data warehouse* (BALLARD, *et al.*, 2006):

1. *Data Warehouse* Empresarial;
2. *Data Mart* Dependente;
3. *Data Mart* Independente.

Um *data mart* armazena dados de uma área específica de negócio, como por exemplo, produtos, clientes e compras, apesar de estarem relacionados, são negócios diferentes para a empresa (KIMBALL e CASERTA, 2004).

A utilização de um *data mart* em relação ao *data warehouse* empresarial é relativamente mais simples e barata, um *data warehouse* empresarial consiste em um repositório único com todas as informações de departamentos e negócios (BALLARD, *et al.*, 2006).

Basicamente existem dois tipos de *data mart*: dependentes e independentes. O dependente precisa obter dados a partir de um *data warehouse* empresarial, logo, ele depende dos dados centralizados com um nível mais baixo de granularidade e então armazena dados com o nível mais alto de granularidade, contém os dados agregados. O independente leva este nome porque não precisa estar conectado ao *data warehouse* empresarial. Este *data mart* consiste em pequenos repositórios de dados dirigidos a uma área específica, os dados são provindos dos bancos de dados operacionais da empresa (KIMBALL e CASERTA, 2004).

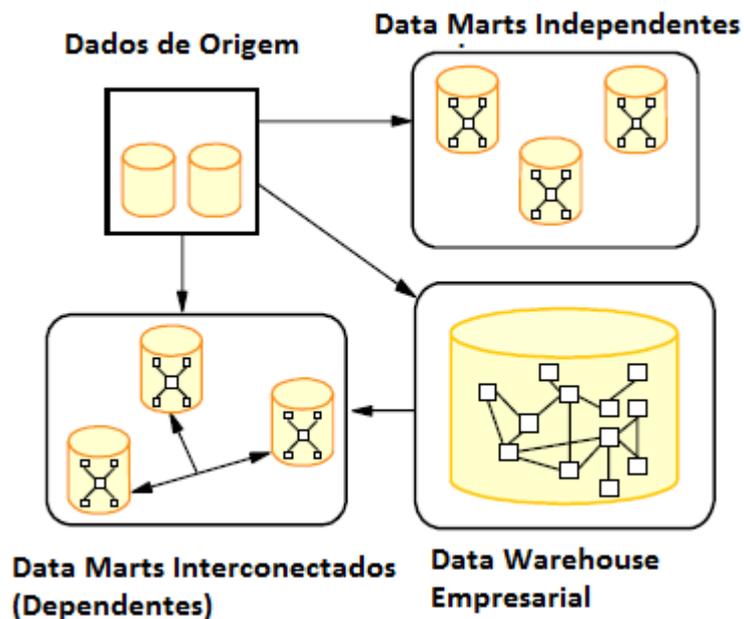


Ilustração 7: Arquiteturas de Data warehouse

Fonte: BALLARD, *et al.*, 2006

Na Ilustração 7 são exibidas as diversas arquiteturas de *data warehouse*.

2.1.5 Carga dos Dados em um Data Warehouse

Quando uma empresa decide implantar um *data warehouse* e tem este projeto bem definido, ela necessita então criar um processo para preencher o *data warehouse* com os dados dos bancos de dados operacionais, processo este que é chamado de integração de dados (JIAWEI, KAMBER e PEI, 2012).

Para realizar a integração dos dados podem ser utilizados sistemas de ETL (*Extract Transform Load*), ou a própria empresa pode desenvolver uma ferramenta para realizar este processo (KIMBALL e CASERTA, 2004). Um sistema de ETL consiste em três passos:

1. Extrair dados de sistemas de origem que podem ser os bancos de dados operacionais da empresa, planilhas, entre outros;
2. Transformar os dados aplicando padrões, limpar dados indesejados e estruturar estes dados;
3. Carregar os dados em um repositório, neste caso o *data warehouse*, para que possa ser acessado por aplicações de análise de negócios, *Business Intelligence*, entre outras (BALLARD, *et al.*, 2006).

A Ilustração 8 apresenta o processo de ETL, onde os dados são extraídos dos bancos de dados operacionais, servidores e arquivos nas estações, em seguida são transformados e então carregados para o *data mart*.

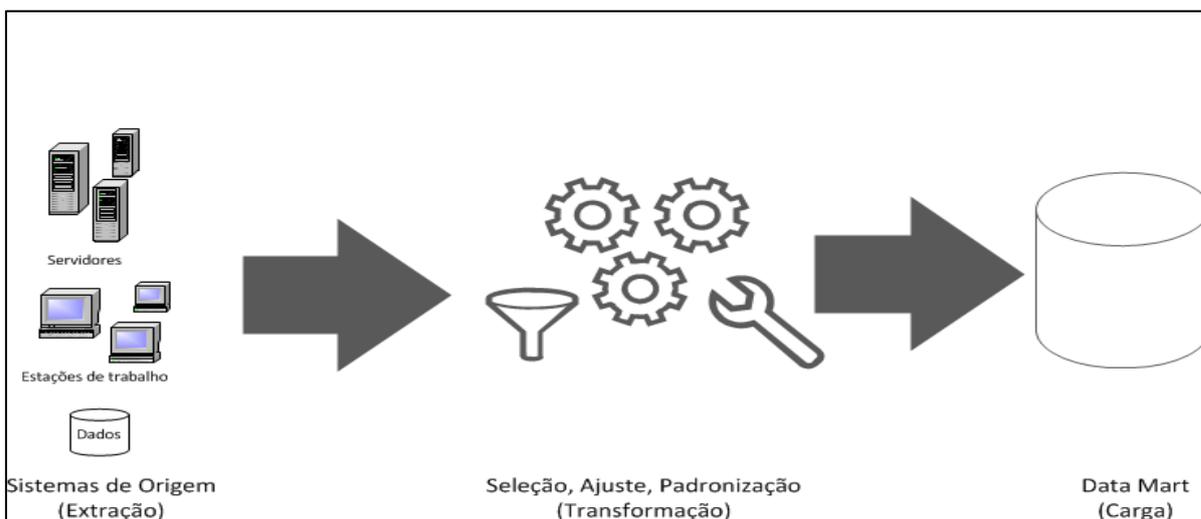


Ilustração 8: Processo de ETL

2.1.6 Modelagem Dimensional

Nesta seção é descrito um guia para auxílio na criação de um modelo dimensional. Todo o processo de modelagem dimensional basicamente consiste em quatro etapas distintas (KIMBALL e ROSS, 2002) (Ilustração 9):

1. Definição do processo de negócio;
2. Definição da granularidade dos dados (nível de detalhe);
3. Definição das dimensões (perspectivas de análise);

4. Definição das medidas (valores avaliados).

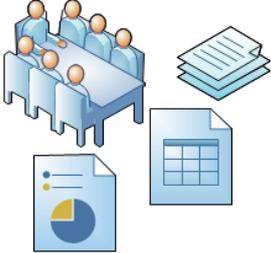
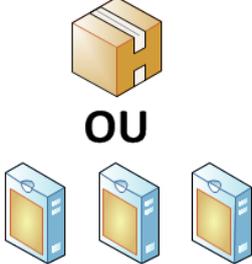
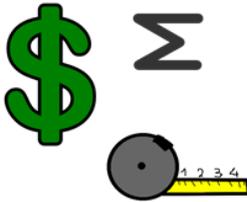
| | | | |
|---|---|---|---|
|  |  | <p>Departamento Produto Ano Loja ? Região Marca Cor Tamanho</p> |  |
| <p>Primeiro Passo Verificar Processos de Negócio</p> | <p>Segundo Passo Definir nível de Detalhamento</p> | <p>Terceiro Passo Definir Dimensões</p> | <p>Quarto Passo Definir as Medidas</p> |

Ilustração 9 Processo de modelagem dimensional

O processo de modelagem dimensional deve iniciar com a definição do processo de negócio. Para definir o processo de negócio é necessário ouvir os usuários do sistema atual, verificar quais indicadores de desempenho eles analisam a partir do sistema, quais relatórios utilizam e para que eles utilizam estes indicadores, tudo isto ajuda a identificar um processo de negócio. São exemplos de processos a compra de materiais, pedidos e estoque. Um processo de negócio não trata apenas de um departamento, ele trata de fontes comuns de dados. Por exemplo, se falar de compra de materiais, não devemos criar um modelo de compra de materiais para o setor de compras, outro modelo para o setor de almoxarifado e outro para o setor financeiro, deve ser criado um modelo de compra de materiais que seja útil aos três departamentos, dessa forma o modelo irá tratar de um processo de negócio.

Definir o modelo por processos de negócio evita a duplicidade de informações (um modelo para cada departamento), logo diminui também o esforço de desenvolvimento do processo de ETL.

Após definir o processo de negócio, deve ser definido em qual nível de detalhamento será necessário visualizar o processo. Tratando-se de vendas, por exemplo, se o nível de detalhe deverá ser o total de vendas por grupo de produtos, ou se o nível de detalhe deverá exibir cada venda de cada produto. O ideal é definir o nível de detalhe para informações atômicas no processo de negócio, atômicas porque estas informações não podem ser divididas, como um produto em um cupom fiscal, por exemplo, o produto não poderá ser subdividido. Quanto mais detalhado, mais coisas poderemos saber com certeza (KIMBALL e ROSS, 2002).

A definição do nível de detalhamento, impacta diretamente sob quais perspectivas de negócio serão possíveis de visualizar uma informação, ou também, em quais valores poderão ser exibidos neste nível de detalhe.

Após definir o nível de detalhamento, deve-se definir as dimensões dos fatos, ou seja, definir a partir de quais perspectivas os usuários gostariam de visualizar uma informação. Por exemplo, voltando a falar de vendas, as vendas podem ser visualizadas por loja, por mês, por departamento, por região do país, ou pelas lojas de uma determinada região em um determinado mês. Pensando desta maneira, é possível definir as dimensões para o fato.

Uma dimensão que estará presente em todos os projetos de *data warehouse* é a dimensão de data. As consultas em SQL (*Structured Query Language*) normais não tratam informações do tipo: se o dia é feriado, se é final de semana, também não permitem que se faça uma consulta somente por ano de uma maneira simples que qualquer usuário com pouco conhecimento consiga realizar a consulta (KIMBALL e ROSS, 2002).

Basicamente definir os atributos da dimensão, é definir quais cabeçalhos de linha iremos visualizar nas informações. Seguindo a ideia da dimensão de data, poderiam ser definidos os seguintes atributos:

1. Ano;
2. Mês;
3. Dia;
4. Final de Semana.

Considerando uma consulta de total de vendas de produtos utilizando o atributo de ano, teríamos a informação ilustrada no Quadro 2.

Quadro 2 Vendas por Ano

| Ano | Total de vendas em Reais |
|------------|---------------------------------|
| 2012 | R\$ 50.000,00 |

Porém, caso o analista quisesse descer a hierarquia desta consulta por mês, simplesmente adicionaria o atributo mês na consulta como é ilustrado no Quadro 3.

Quadro 3 Vendas por ano e mês

| Ano | Mês | Total de vendas em Reais |
|------------|------------|---------------------------------|
| 2012 | Maio | R\$ 10.000,00 |
| 2012 | Junho | R\$ 15.000,00 |
| 2012 | Julho | R\$ 25.000,00 |

O último passo para criar um modelo dimensional é definir os valores numéricos que deverão aparecer no fato, como por exemplo, o total das vendas ou a quantidade de itens vendidos de um produto. Estes valores numéricos serão as medidas da tabela de fato, é

importante ressaltar que todos deverão obrigatoriamente ser compatíveis com o nível de detalhe da informação definido no segundo passo.

2.2 Processamento Analítico Online (OLAP)

O processamento analítico *online* caracteriza ferramentas onde o usuário monta consultas dinâmicas através de bases de dados consolidadas, onde o objetivo desta consulta é a tomada de decisão. Esta característica é atingida através da análise multidimensional dos dados realizada de forma dinâmica. Estas consultas são utilizadas de forma casual e única, o usuário que as realiza não utiliza de nenhum método matemático ou científico, somente seu próprio conhecimento de negócio que é o mais necessário no momento da análise destes dados (INMON, 2005).

As ferramentas de processamento analítico online proporcionam uma rápida manipulação e consulta dos dados graças a uma técnica que consiste em armazenar os dados em forma de cubos, estes cubos possuem os valores subtotais calculados facilitando operações como agregação. Basicamente existem três variações de OLAP (JIAWEI, KAMBER e PEI, 2012):

1. MOLAP (Processamento Analítico Online Multidimensional) – Os dados são armazenados em cubos proprietários prontos para o processamento;
2. ROLAP (Processamento Analítico Multidimensional Relacional) – Os dados são armazenados em bancos de dados relacionais, as consultas são realizadas através da linguagem SQL (*Structured Query Language*);
3. HOLAP (Processamento Analítico Multidimensional Híbrido) – Os dados agregados são armazenados utilizando MOLAP, porém os dados detalhados através de ROLAP, por isso a nomenclatura de híbrido.

Nas seções a seguir serão explicadas as operações de processamento analítico *online* e as ferramentas que proporcionam estas operações.

Para a manipulação do cubo de dados OLAP, podem ser realizadas diversas operações como *Slice* e *Dice*, *Drill-Down* e *Roll-Up*, a Ilustração 10 exhibe o funcionamento destas operações que serão explicadas nas seções a seguir.

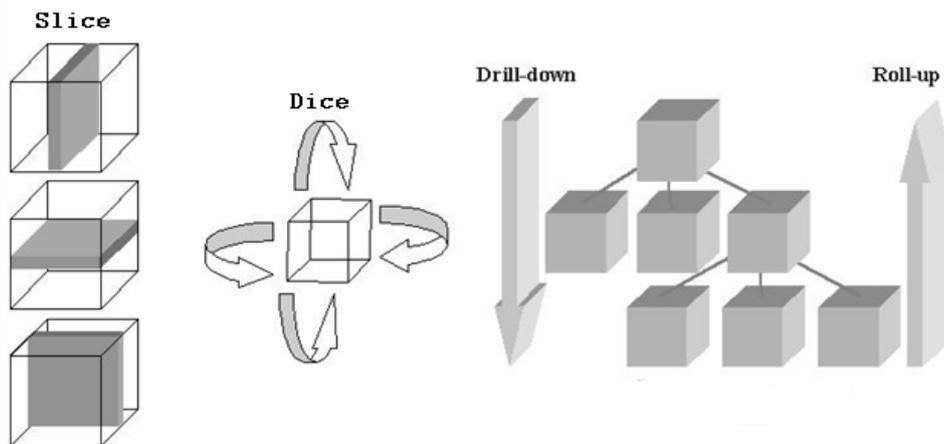


Ilustração 10: Técnicas OLAP

Além das operações OLAP, também existe uma linguagem chamada de *Multi Dimensional expression* (MDX) desenvolvida inicialmente pela empresa Microsoft para criação de consultas em bancos de dados OLAP (JANUS e FOUNCHÉ, 2010).

2.2.1 Drill-down e Roll-Up

As operações de *Drill-down* e *Roll-up* são operações de manipulação da granularidade da informação, onde é possível aumentar ou diminuir a granularidade da mesma.

A operação de *Drill-down* consiste em diminuir a granularidade da informação, por exemplo, se existisse uma tabela com o total de vendas, na operação de *Drill-Down* o usuário poderia visualizar vendas por país, aplicando a técnica novamente visualizaria vendas por estado, sendo possível até chegar ao consumidor que comprou uma mercadoria, tudo isso é claro dependendo da implementação do cubo de dados. A operação de *Roll-up* simplesmente realiza o processo inverso, diminui a granularidade das informações, considerando o exemplo anterior, a operação iria regredir a informação em vendas por estado, vendas por país e finalmente o total de vendas (KIMBALL e ROSS, 2002).

Na Ilustração 11 é exibida a operação de *Drill-down*, onde um indicador de vendas é desmembrado por país, estado e consumidor, na imagem existe uma seta à esquerda indicando o sentido da operação para baixo, ou seja, diminuindo cada vez mais a granularidade. Na imagem também é possível ver a operação de *Roll-up*, o sentido desta operação fica claro através da seta à direita da imagem indicando que ela vai para cima, ou seja, eleva a granularidade da informação.

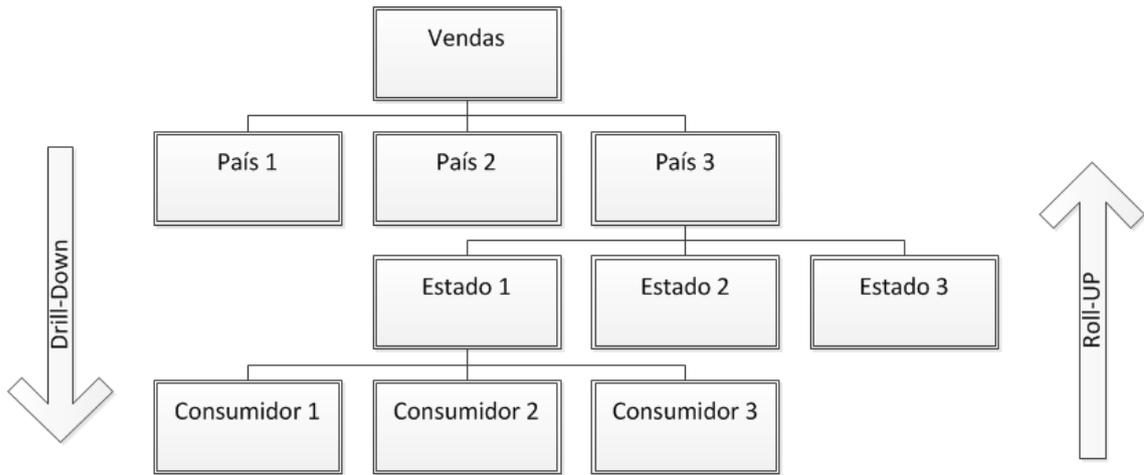


Ilustração 11: Drill-down e Roll-Up

Através das operações de *Slice* e *Dice* o usuário consegue manipular o cubo para ter diversas visões de uma informação, ou seja, diferentes perspectivas (INMON, 2005).

2.2.2 Slice e Dice

Utilizando a operação de *Slice* é possível separar um grupo de informações eliminando da visão as informações desnecessárias, é como retirar uma fatia do cubo para visualizar. A operação de *Dice* consiste em mudar a perspectiva do cubo, para visualizar uma mesma informação de um ângulo diferente (JÚNIOR e BERNARDINO, 2009).

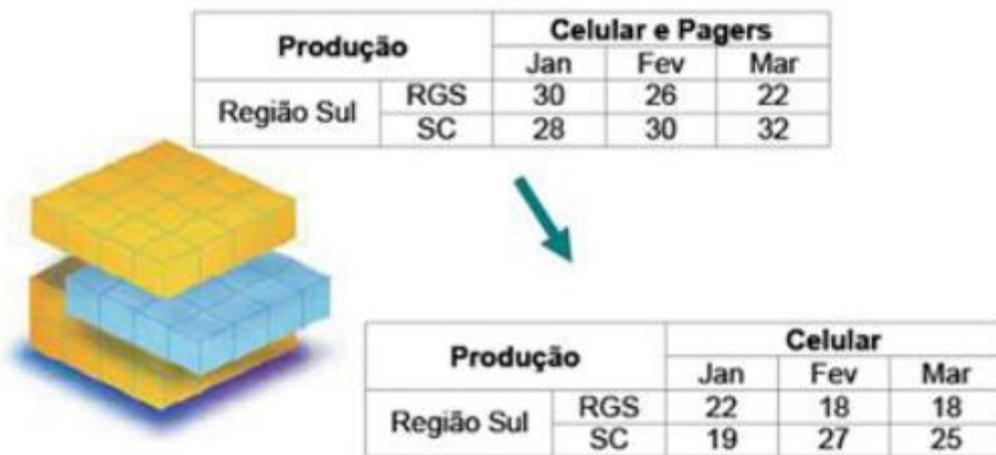


Ilustração 12: Slice

Através da Ilustração 12 é exibida a operação e *Slice*, a ilustração contém a figura de um cubo com a sua parte do meio sendo dividida, esta parte representa a venda de celulares

que foi extraída de um resultado de venda de celulares e pagers, fica claro na figura que a venda de celulares é “cortada” para ser visualizada.

Na operação de *Dice* a perspectiva dos dados é alterada, proporcionando uma visualização diferente dos mesmos dados (JÚNIOR e BERNARDINO, 2009).

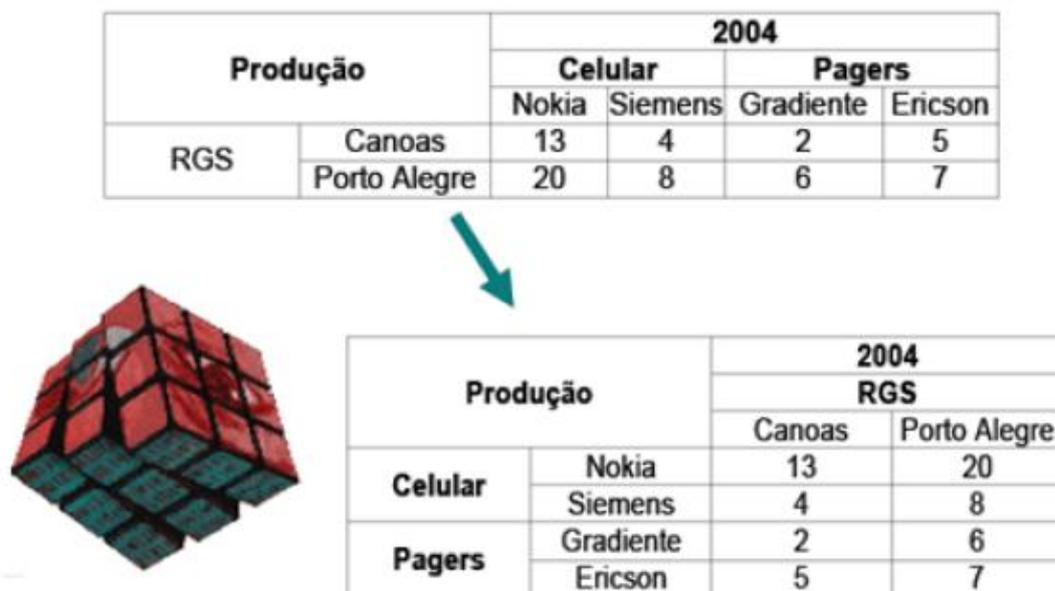


Ilustração 13: Dice

Na Ilustração 13 é exibida a operação de *Dice*, onde existia uma tabela mostrando a produção de celulares e pagers em Canoas e Porto Alegre separados por marca, aplicando a operação de *Dice*, é possível verificar que a perspectiva foi alterada para mostrar a produção de celulares e pagers separados por cidade.

2.3 Considerações Finais

Neste capítulo foram apresentados os principais conceitos necessários ao entendimento de BI, além dos conceitos foram introduzidos modelos e técnicas. Com a explicação sobre *data warehouse* ficou clara a necessidade de utilização de um banco de dados separado e em uma estrutura diferente dos bancos de dados operacionais utilizados pelas empresas.

Apesar de este trabalho ser realizado sobre o produto de uma *software house*, e sendo assim utilizar como fonte de dados somente o banco de dados deste sistema, seria inviável um projeto de BI utilizando as tabelas do banco de dados atual, mesmo que fosse através da utilização da arquitetura ROLAP. O banco de dados transacional possui diversos

processamentos realizados a cada inserção ou atualização de um novo dado. Também, possui mecanismos de bloqueio de dados para quando um determinado dado for consultado não poder ser alterado por outros usuários o que gera uma queda de desempenho deste banco de dados. Para a operação de uma empresa estes mecanismos são vitais, porém para a consulta de informações acabam se tornando vilões.

A vantagem de desenvolver o trabalho sobre o produto de uma *software-house* é de conhecer as fontes de dados de origem, e durante a modelagem do DW já estar ciente de em quais informações será possível chegar. A desvantagem é de que não será possível gerar um modelo que agrade à todos os clientes de uma maneira geral, será possível a definição de um modelo genérico que muito provavelmente sofrerá diversas alterações conforme sua utilização. Porém, como conceituado neste capítulo o DW possui um ciclo de vida incremental, logo estas alterações são esperadas.

Para a definição do modelo genérico, existe a vantagem do acesso aos consultores de produto que estão diariamente acompanhando os processos de diversos clientes, de uma maneira informal, estes consultores são capazes de identificar quais são as informações que os clientes mais utilizam ou questionam sobre a existência no sistema, facilitando assim a criação de um modelo de DW.

Para a modelagem do DW será utilizado o modelo multi-estrela, pois este possui um melhor aproveitamento das tabelas de dimensões, necessitando somente a criação de novos fatos e dimensões extras necessárias aos novos fatos.

As dimensões de degeneração serão utilizadas pois como este trabalho consiste em um produto de gestão de vendas, é necessário agrupar os itens vendidos em cada transação de vendas, logo o número das transações de vendas consistirá em uma informação de dimensão de degeneração.

Chaves substitutas serão de extrema importância neste trabalho, pois o banco de dados atual da *software house* utiliza de chaves literais e compostas, logo, se estas fossem utilizadas no DW aumentariam significativamente o tamanho físico da tabela de fatos. Outro detalhe, é que como este produto poderá ser utilizado por diversos clientes, a utilização de chaves substitutas garantirá que os usuários utilizarão as dimensões como meio de acesso aos dados, logo, será possível que a orientação de acesso aos dados seja distribuída de forma uniforme sem alterações de percurso.

3 FERRAMENTAS

Nesta seção são introduzidas as ferramentas que fazem parte do pacote de *Business Intelligence* da Microsoft. O pacote de BI é composto por três ferramentas:

1. *SQL Server Integration Services*;
2. *SQL Server Analysis Services*;
3. *SQL Server Reporting Services*.

Das três ferramentas citadas, somente as duas primeiras serão utilizadas e portanto conceituadas. O *SQL Server Reporting Services* faz serve como uma ferramenta de interface para o usuário visualizar os dados, ao invés de utilizar esta ferramenta, será utilizado o mecanismo de tabelas dinâmicas do Microsoft Excel.

As duas primeiras ferramentas citadas servem como ferramentas de construção de um ambiente de BI, o Excel com as tabelas dinâmicas serve para visualizar e interagir com os dados contidos neste ambiente. Tudo isso será explicado nas seções a seguir.

3.1 Microsoft SQL Server Integration Services (SSIS)

O Microsoft *SQL Server Integration Services* (SSIS) consiste em um conjunto de tarefas e transformações para integração de dados projetada exclusivamente para integração de dados. É possível obter dados de diversas fontes para executar tarefas com o intuito de distribuir os dados em destinos diferentes. O *Integration Services* pode ser utilizado através de ferramentas gráficas ou linha de comando (MICROSOFT, 2011).

Uma das aplicações do SSIS é o processamento de dados para um criar um *data warehouse* (DW). Muitas empresas utilizam o SSIS como ferramenta para conversão de dados de sistemas legados para novos aplicativos, outras somente para a integração de dados no DW. De uma maneira geral, o SSIS torna tarefas de integração de dados mais simples e eficazes através do uso de um mecanismo de fluxo de controle para conversão de dados (VEERMAN, LANCHEV e SARKA, 2009).

Na Ilustração 14 é possível verificar a tela de projeto do SSIS, onde está sendo realizado um projeto de ETL de dimensões e um fato.

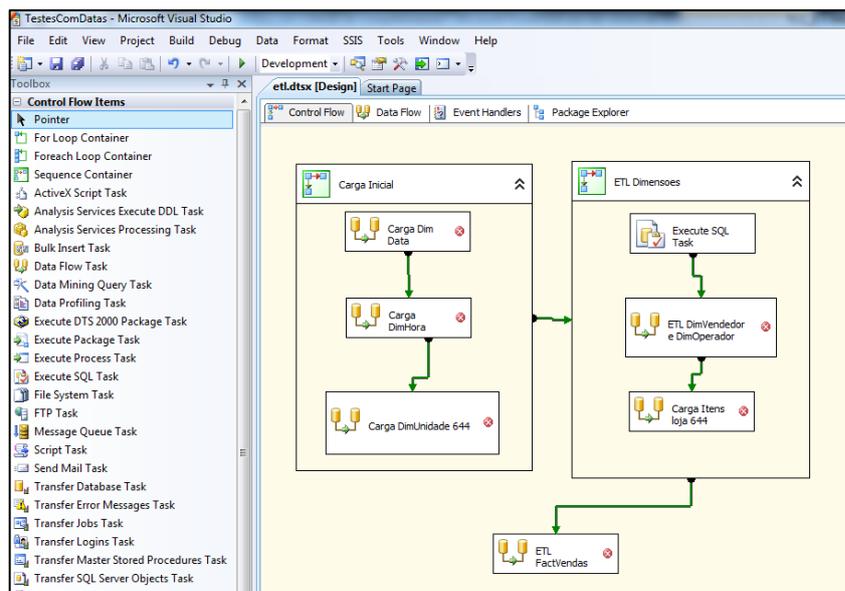


Ilustração 14 Tela de projeto do Microsoft *Integration Services*

A hierarquia básica do SSIS consiste em: pacote (*package*), fluxo de controle (*control flow*), fluxo de dados (*data flow*), dados de origem (*data source*), transformações (*transformations*) e dados de destino (*data destination*) respectivamente. Esta hierarquia pode ser visualizada na Ilustração 15.

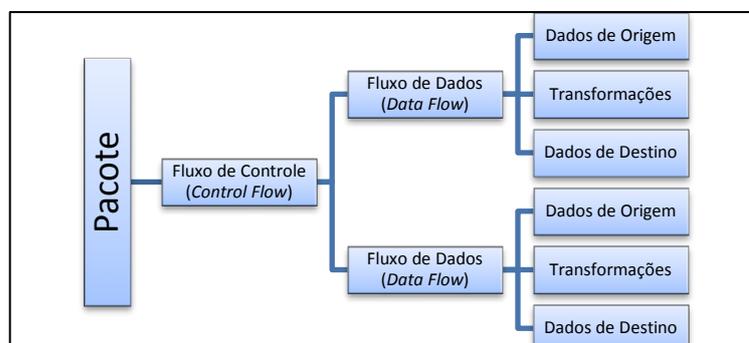


Ilustração 15 Hierarquia de componentes do SSIS

O pacote é o principal objeto do *Integration Services*. Este contém toda a lógica de negócio para o controle e processamento de dados. É utilizado para mover dados de fontes de origem para bases de destino. Todas as ferramentas e mecanismos para extração, transformação e carga de dados, ou seja, operações de ETL estão presentes no pacote. Existem duas formas de criar um pacote, a primeira é utilizar uma ferramenta chamada *SQL Sever Import and Export Wizard* que está presente na IDE (*Integrated Development Environment*) do *SQL Server Management Studio* (SSMS), a segunda maneira é utilizar a ferramenta *SSIS Designer* presente na IDE *Business Intelligence Development Studio* (BIDS).

Na Ilustração 16 é exibido o acesso à ferramenta *SQL Sever Import and Export Wizard* através do SSMS clicando com o botão direito do mouse sobre o banco de dados para o qual se deseja criar o pacote. No menu que aparece acessando a opção *tasks* e depois *import*. Após utilizar estes passos irá aparecer uma tela de boas vindas com breves explicações sobre esta ferramenta.

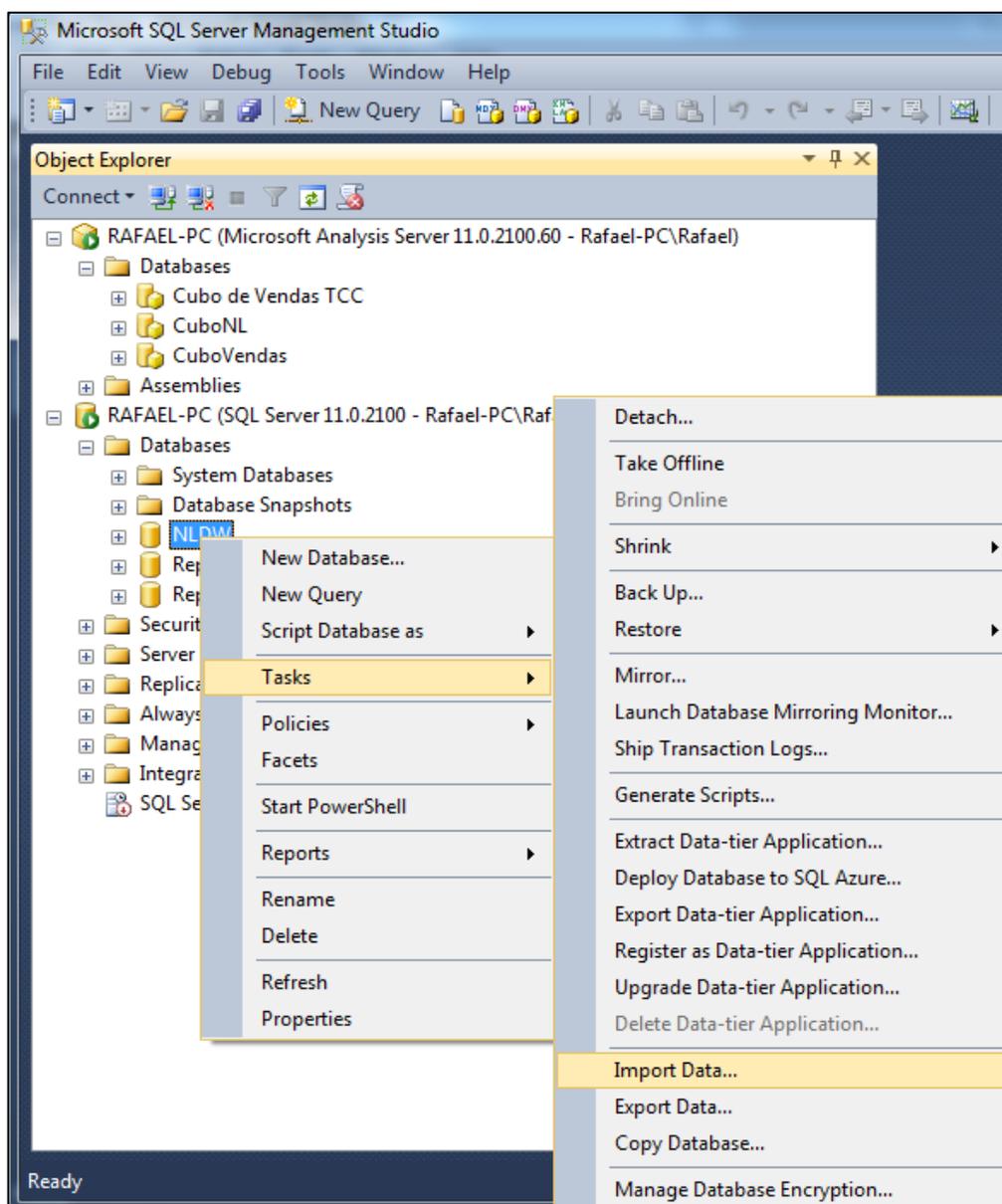


Ilustração 16 Acesso à ferramenta **Import and Export Wizard** no SSMS

Os pacotes para processamentos simples podem ser criados rapidamente pelo *wizard*, porém tarefas mais avançadas não podem ser criadas através desta ferramenta, portanto é necessário criar um projeto do SSIS na ferramenta *BIDS*. Na Ilustração 17 é exibida a tela de criação de um projeto do SSIS na ferramenta de *Business Intelligence*.

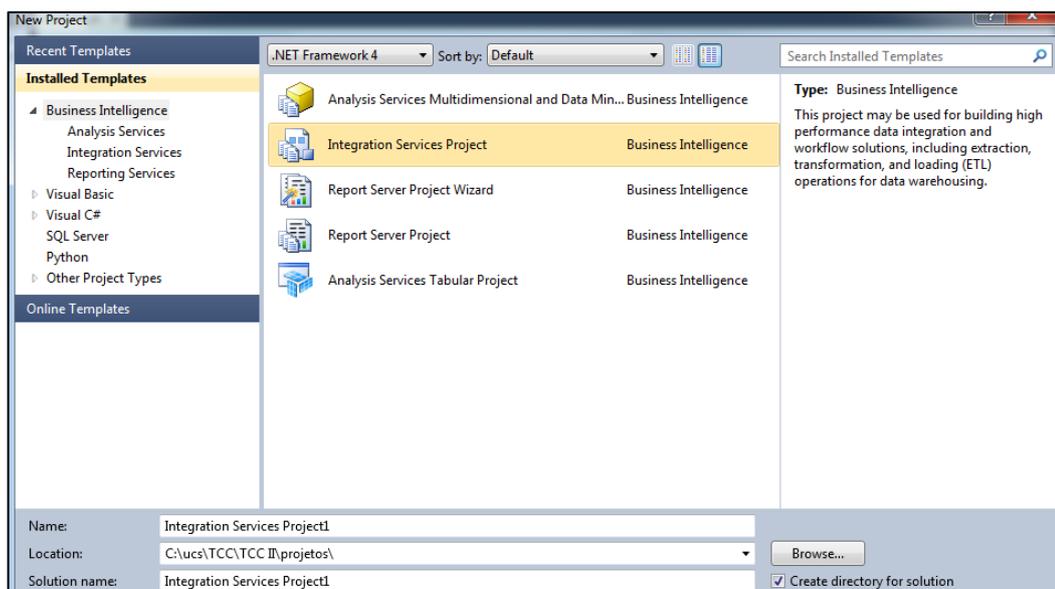


Ilustração 17 Criação de projeto do SSIS no BIDS

Após configurar os dados como nome do projeto, local para gravação e nome da solução, é apresentada a tela principal do BIDS para projetos do SSIS. Esta tela é de simples utilização, à esquerda existe a paleta *SSIS Toolbox* que contém os componentes que poderão ser utilizados dentro do pacote. À direita existe a paleta *Solution Explorer*, nela encontram-se os pacotes e os gerenciadores de conexão do projeto.

No centro existe uma área subdividida em abas, a primeira aba é onde está o fluxo de controle, nele serão colocados diversos componentes de fluxo de dados com ordem de precedência, ou seja, pode ser definida a ordem em que será executado cada fluxo de dados.

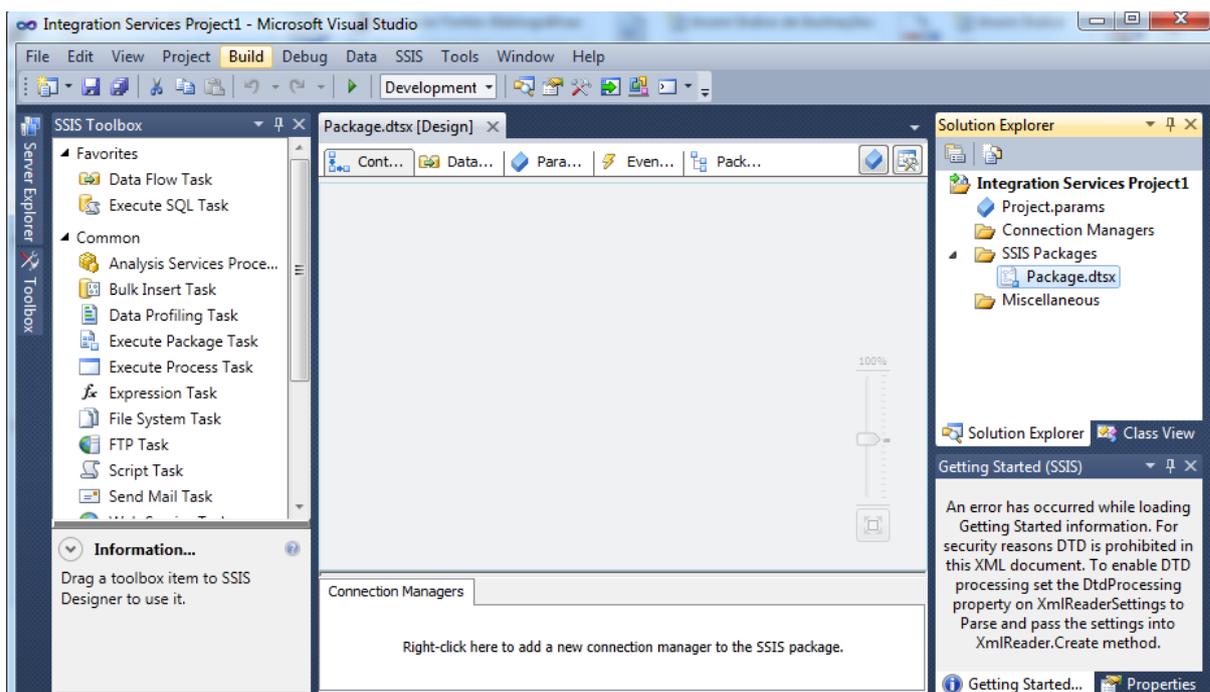


Ilustração 18 Tela principal do BIDS com um projeto SSIS

Dentro do fluxo de controle além de componentes de fluxo de dados, podem ser inseridos componentes para executar operações de envio de e-mail, operações com arquivos, execução de outros programas ou processos no computador, entre outras tarefas do sistema operacional. Também podem ser incluídas operações no cubo de dados, como por exemplo, processar o cubo imediatamente após a carga de dados. Na Ilustração 19 é exibido um exemplo de fluxo de controle carregando três dimensões, uma tabela de fatos, e após todas as cargas a chamada de um processamento do cubo de dados, as setas verdes indicam o fluxo de processamento.

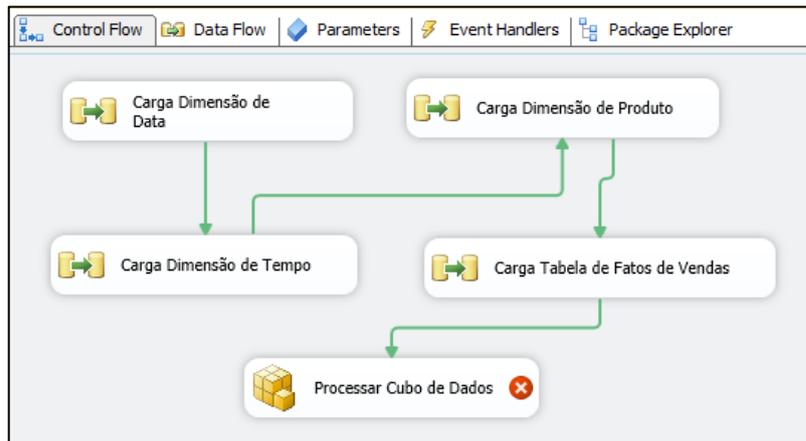


Ilustração 19 Exemplo de Fluxo de Controle

Na Ilustração 20 é exibido um exemplo de fluxo de dados utilizado no fluxo de controle presente na Ilustração 19. Este fluxo de dados utiliza um componente de *data source* para buscar dados de um banco de dados Oracle, um componente de transformação para converter os tipos de dados para tipos aceitos pelo SQL Server e, um componente *data destination* para gravar os dados no SQL Server.

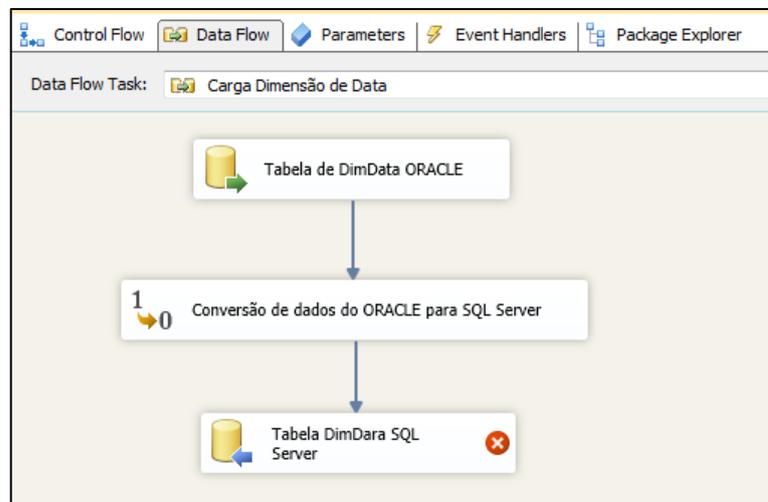


Ilustração 20 Exemplo de fluxo de dados

Os pacotes do SSIS podem ser executados manualmente através do BIDS ou agendados em *Jobs* do SQL Server. Um *job* é um processo que pode ser configurado no gerenciador de banco de dados para ser executado de tempos em tempos, este processo é capaz de executar um pacote, uma *stored procedure* ou outros processamentos do banco de dados. Também é possível executar pacotes através da ferramenta **DTE**xec, esta ferramenta é utilizada através de linhas de comando ou um arquivo batch (arquivo em lotes) (VEERMAN, LANCHEV e SARKA, 2009).

Para executar um pacote através do DTE

`dtexec.exe /file "c:\nome do pacote.dtsx".`

3.2 Microsoft SQL Server Analysis Services (SSAS)

O Microsoft SQL Server Analysis Services (SSAS) é um servidor multidimensional que prove serviços de processamento analítico online e mineração de dados. Com o SSAS é possível projetar, criar e gerenciar estruturas multidimensionais contendo dados detalhados e agregados originados de diversas fontes (VEERMAN, LANCHEV e SARKA, 2009).

Através do SSAS é possível prover um único ponto de acesso aos dados de BI da empresa. Além disto, é possível combinar estruturas OLAP, estruturas de mineração de dados e de indicadores de desempenho (KPI) em um único sistema integrado. Isto facilita o desenvolvimento de soluções de BI e tornando os resultados mais acessíveis para os usuários. Para gerenciar e trabalhar com cubos OLAP pode ser utilizada a ferramenta SSMS. Para projetar e criar novos cubos pode ser utilizada a ferramenta BIDS. As principais funcionalidades do SSAS são (JANUS e FOUNCHÉ, 2010):

1. Processamento Analítico Online – permite criar consultas de dados agregados armazenados em cubos multidimensionais;
2. Mineração de Dados – permite identificar relações nos dados e calcular probabilidades de futuros resultados baseado em ações do passado;
3. Dados de múltiplas fontes – Permite combinar dados de fontes OLAP e OLTP em conjunto;
4. Indicadores de Performance – Possui suporte à criação de *balanced score cards* para avaliar o desempenho dos objetivos da empresa.

Uma instância do SSAS pode conter diversos bancos de dados, e cada um destes bancos de dados pode conter diversos objetos OLAP. As aplicações devem se conectar a uma instância do SSAS e um banco de dados específico para consultar os objetos OLAP. Um computador utilizado como servidor pode hospedar diversas instâncias do SSAS. Estas instâncias são nomeadas utilizando o nome do servidor mais o nome da instância, da seguinte maneira: “<NomeDoServidor>\<NomeDaInstancia>”. Na Ilustração 21 é possível verificar esta estrutura, nota-se na imagem uma aplicação denominada “AMO Application” se conectando a uma instância do SSAS.

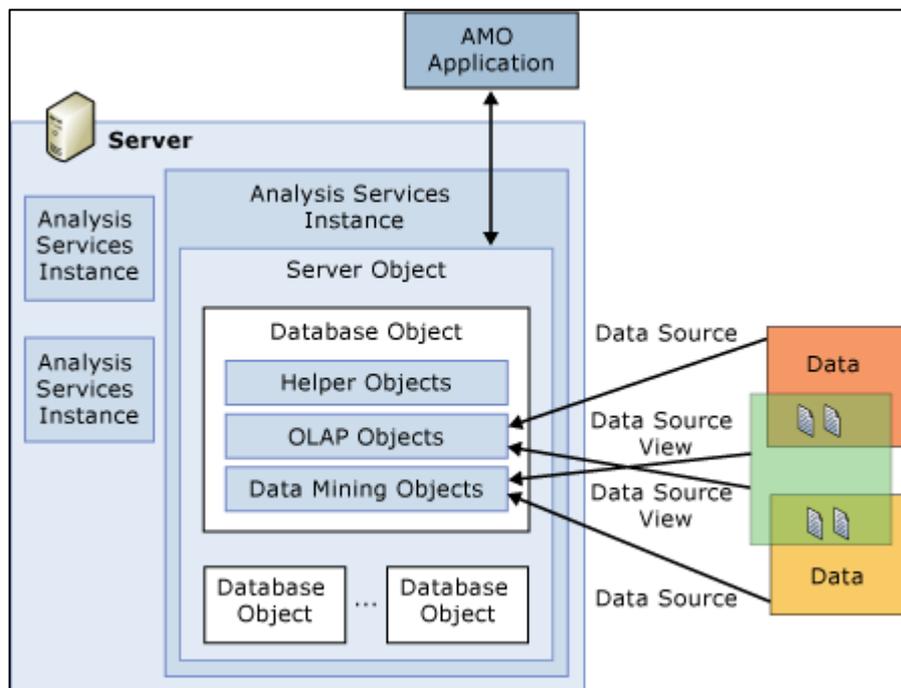


Ilustração 21 Arquitetura do Analysis Services

Para criar um cubo no SSAS são requeridos no mínimo três objetos listados a seguir (JANUS e FOUNCHÉ, 2010):

1. Dimensão – São objetos que representam as dimensões do DW. As dimensões no SSAS possuem seus atributos organizados em hierarquias;
2. Grupo de Medidas – Podem ser considerados como a tabela fato propriamente dita, no grupo de medidas estão os objetos de medidas (*measures*) que nada mais são do que as medidas dos fatos;
3. Partição – É um agrupamento do grupo de medidas para agilizar o processamento de agregações e armazenamento dos grupos.

O objeto de partição é utilizado para o gerenciamento do cubo de dados. Ele permite que um cubo de dados que possua um grande volume de dados possa ser subdividido e armazenar cada uma das subdivisões em locais diferentes.

Se levarmos como exemplo um cubo de dados de vendas, este cubo pode conter milhões de transações de vendas, logo sempre que este cubo for atualizado ele terá que processar vendas de anos anteriores que já foram calculadas anteriormente, este processamento redundante será demorado e de grande custo. Com a utilização de partições, é possível separar os dados de cada ano de vendas em uma partição, logo, será necessário processar somente o ano que sofreu alterações, os demais anos não necessitarão de processamento aumentando o desempenho e diminuindo o custo de processamento.

As partições podem ser armazenadas em diferentes servidores, ou seja, vendas de anos anteriores que talvez sejam pouco consultadas, podem ficar armazenadas em servidores mais lentos, e a partição com as vendas do ano corrente que serão mais frequentemente acessadas, podem ficar no servidor mais rápido e com mais recursos. Na Ilustração 22 é possível verificar um particionamento de dados de vendas em três servidores, cada servidor armazenando os dados de um ano de vendas. Esta separação é transparente para o usuário.

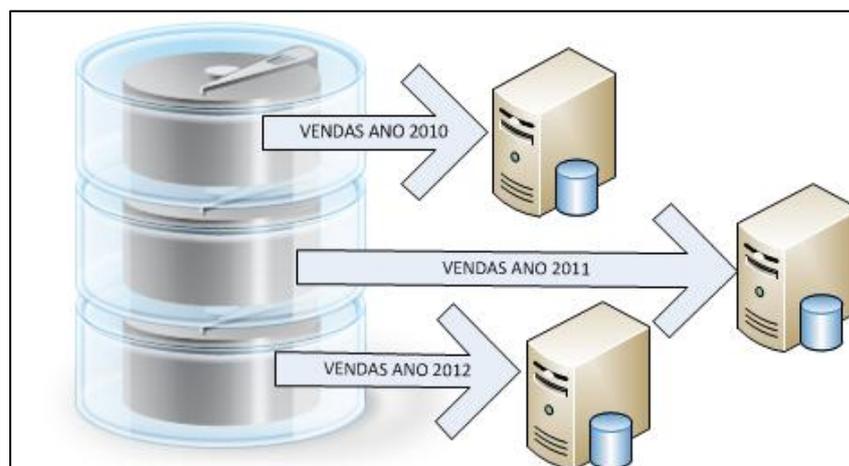


Ilustração 22 Particionamento do cubo de dados em diferentes servidores

É importante salientar que cada grupo de medidas do cubo possui pelo menos uma partição. Nesta é definido o modo de armazenamento dos dados de um cubo, ou seja, se ele será HOLAP, MOLAP ou ROLAP.

Dentro das partições podem ser criados objetos de agregação. Estes objetos são utilizados para manter os dados calculados armazenados no cubo, ou seja, um dado que é acessado com frequência como por exemplo, o valor total de vendas de cada unidade por ano, pode ser calculado durante o processamento do cubo e armazenado para que quando o usuário cruze a dimensão de ano e a dimensão de unidade com o fato de vendas, não seja necessário que este dado seja calculado. O SSAS permite que sejam calculadas todas as agregações possíveis para um grupo de medidas com as dimensões que estão referenciadas ao mesmo.

Para isso deve ser definida uma agregação. A vantagem da agregação está no tempo de resposta que se torna praticamente instantâneo.

Bancos de dados OLAP são definidos por cubos, dimensões e medidas. Os principais objetos do SSAS para definir um bando de dados OLAP são (JANUS e FOUNCHÉ, 2010):

1. *Data Source* – Utilizado para gravar informações da conexão com a fonte de dados em um projeto do SSAS. Mais especificamente ele contém as informações para o SSAS se conectar a uma fonte de dados física. Os *data sources* do SSAS podem se conectar a diversas fontes de dados tais como: Microsoft SQL Server, Oracle e DB2;
1. *Data Source View* – Funciona como uma camada abstrata do *data source*. Ele contém o modelo lógico do esquema de dados que será utilizado no SSAS. O DSV armazena as definições de metadados;
2. *Measures* – São valores numéricos ou fatos que os usuários irão analisar;
3. *Measure Groups* – São grupos lógicos de medidas (pode ser considerado como a tabela fato). Os grupos de medidas possuem a vantagem de poder ser processados individualmente em um cubo;
4. *Dimensions* – provém contexto para as medidas. Organizam os dados relacionando-os com uma área de interesse, como consumidores, empregados, produtos, filiais, etc. As dimensões no SSAS possuem atributos que correspondem às colunas da tabela de dimensão no DW. Os atributos podem ser organizados em hierarquias do tipo pai-filho;
5. *Cubes* – Combinam dimensões e grupos de medidas formando uma estrutura multidimensional contendo agregações de cada medida cruzada com cada atributo de dimensão. Os cubos são os objetos que o usuário irá acessar para analisar os dados.

Na Ilustração 23 é exibida a tela de configurações do *data source* no SSAS. Nestas configurações é possível definir o tempo que o *data source* irá esperar por uma consulta, as informações de *Impersonation*, ou seja, a maneira como ele irá se autenticar no servidor em que está realizando a consulta. Também é possível personalizar o nome e definir uma descrição para este *data source*.

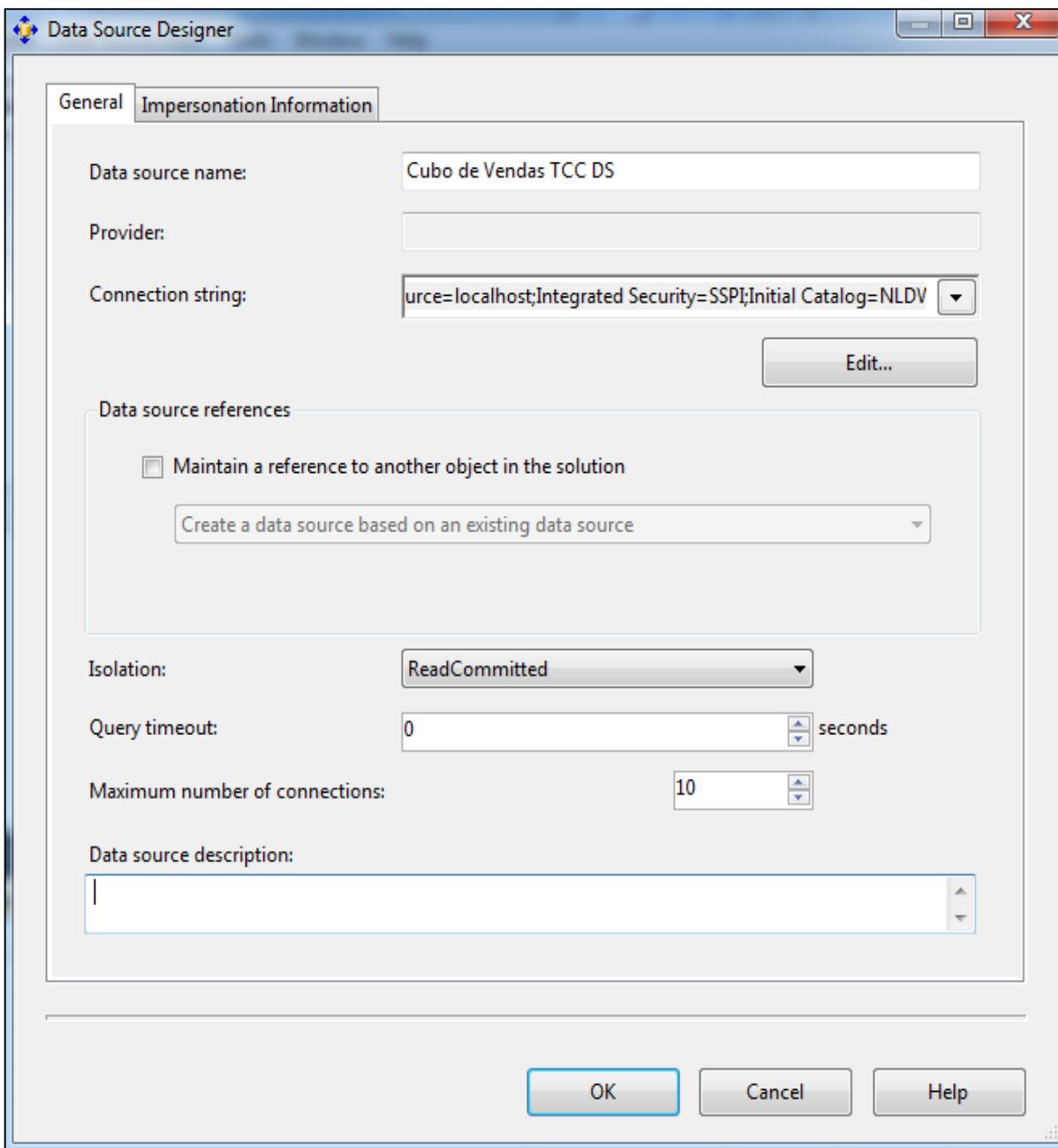


Ilustração 23 Configurações do Data Source no SSAS

Na Ilustração 24 é exibido um *data source view*, onde no centro, encontra-se um diagrama representando as tabelas que este *data source view* contém. No canto inferior direito existe uma caixa de propriedades, onde uma destas propriedades é o *data source* do qual são buscados os dados. No canto inferior esquerdo existe uma caixa que lista todas as tabelas disponíveis no *data source view*. É possível personalizar estas tabelas alterando o nome das tabelas e de seus campos. É importante salientar que as alterações realizadas no *data source view* não irão refletir no banco de dados, pois ele funciona como uma camada de intermédio entre a fonte de dados e o cubo.

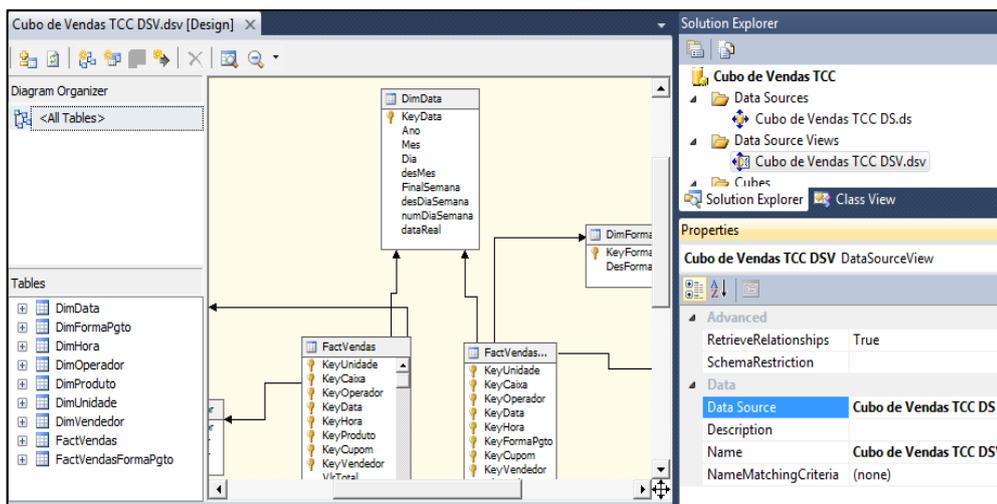


Ilustração 24 Designer de um Data Source View

Na Ilustração 25 é exibido o *cube designer* da ferramenta BIDS, o designer é subdividido em abas onde é possível personalizar a estrutura do cubo (aba selecionada na imagem), personalizar as dimensões, criar campos calculados, criar KPIs, partições, perspectivas, e testar o cubo através do browser. Na imagem fica clara a estrutura de cubo no SSAS, pois no canto superior esquerdo estão os grupos de medidas e no canto inferior esquerdo as dimensões.

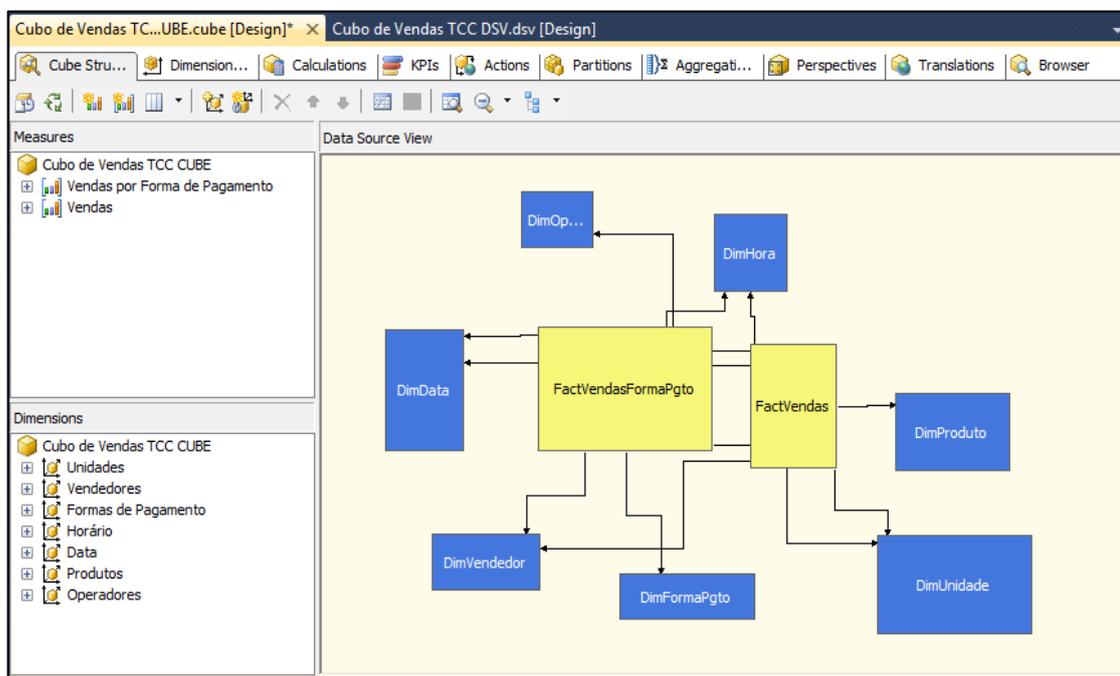


Ilustração 25 Cube Designer no BIDS

Na Ilustração 26 é exibida a relação de cada dimensão com o grupo de medidas, ou seja, a ferramenta de utilização de dimensão serve para visualizar e alterar o barramento de dados do cubo. Nota-se na imagem que no cabeçalho das colunas estão os grupos de medidas,

e nos títulos das linhas as dimensões, os dados são o atributo da dimensão utilizado para ligar a dimensão à tabela de fato, neste caso o grupo de medidas.

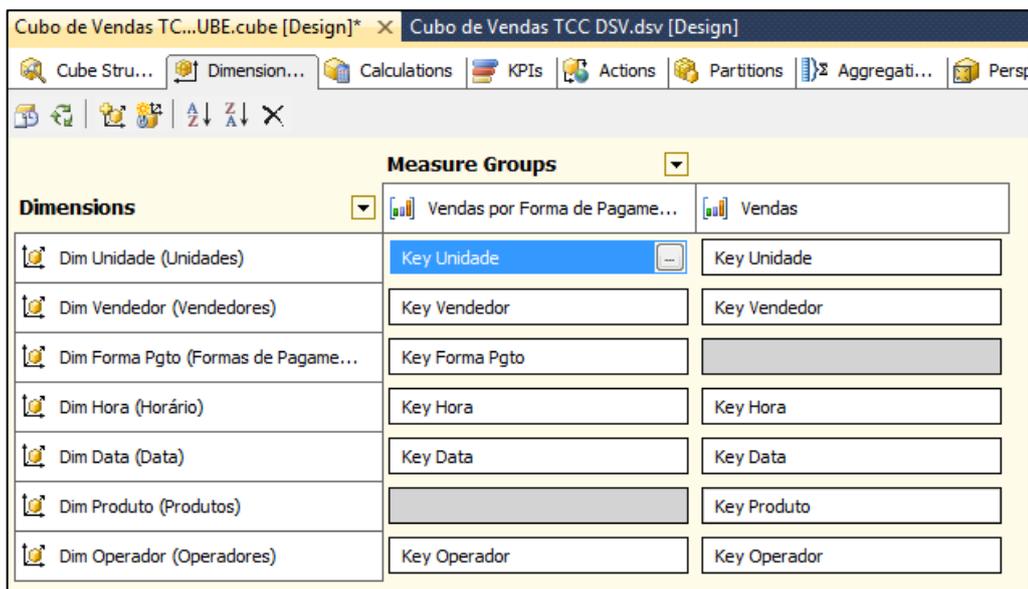


Ilustração 26 Cube Designer - Paleta Dimension Usage

Na Ilustração 27 são exibidas as partições do cubo e o modo de armazenamento de cada uma. Cada grupo de medidas possui pelo menos uma partição. As partições são um recurso importante conforme o volume de dados do cubo vai crescendo. No caso de um cubo de vendas, por exemplo, é possível criar uma partição para cada ano de vendas, logo se torna possível processar as vendas de cada ano de forma individual, considerando um volume de um milhão de registros por ano, não se torna interessante processar dados antigos já processados, somente os dados novos, do ano corrente. O mecanismo de partições aumenta consideravelmente o processamento de um cubo.

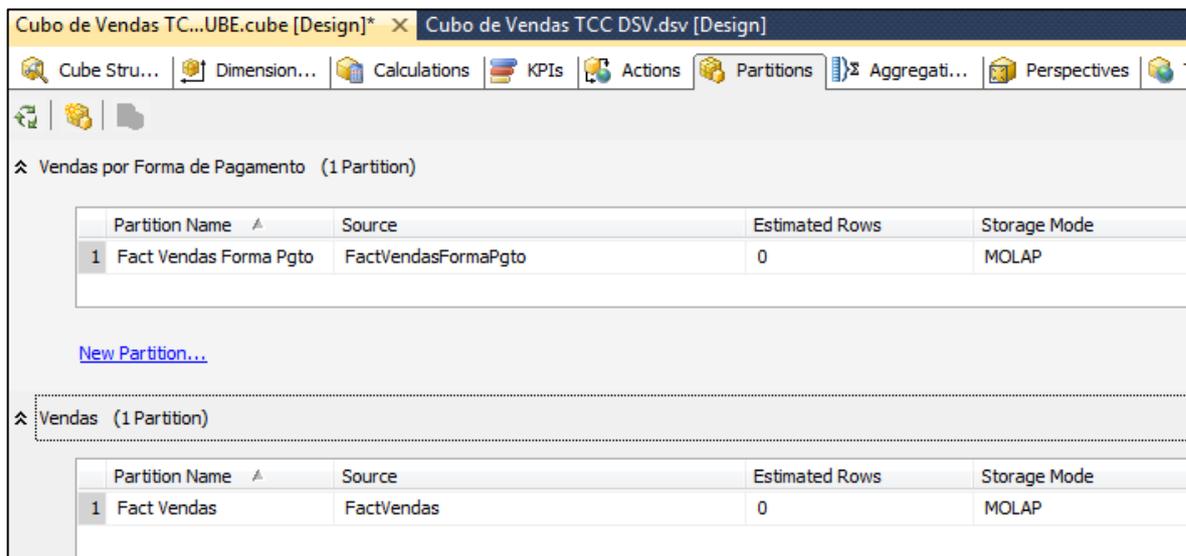


Ilustração 27 Cube Designer - Paleta Partitions

Na Ilustração 28 é exibida a ferramenta *Browser* do cubo, nela é possível testar as estruturas de dados do cubo, no centro existe uma área de dados, onde é possível selecionar medidas e atributos de dimensão do cubo presente na caixa à esquerda da tela, e arrastá-los para dentro da área de dados. Na imagem está sendo utilizada a dimensão de unidade, com o atributo de UF e a dimensão de data com o atributo de mês, a medida que está sendo exibida é o valor total de vendas.

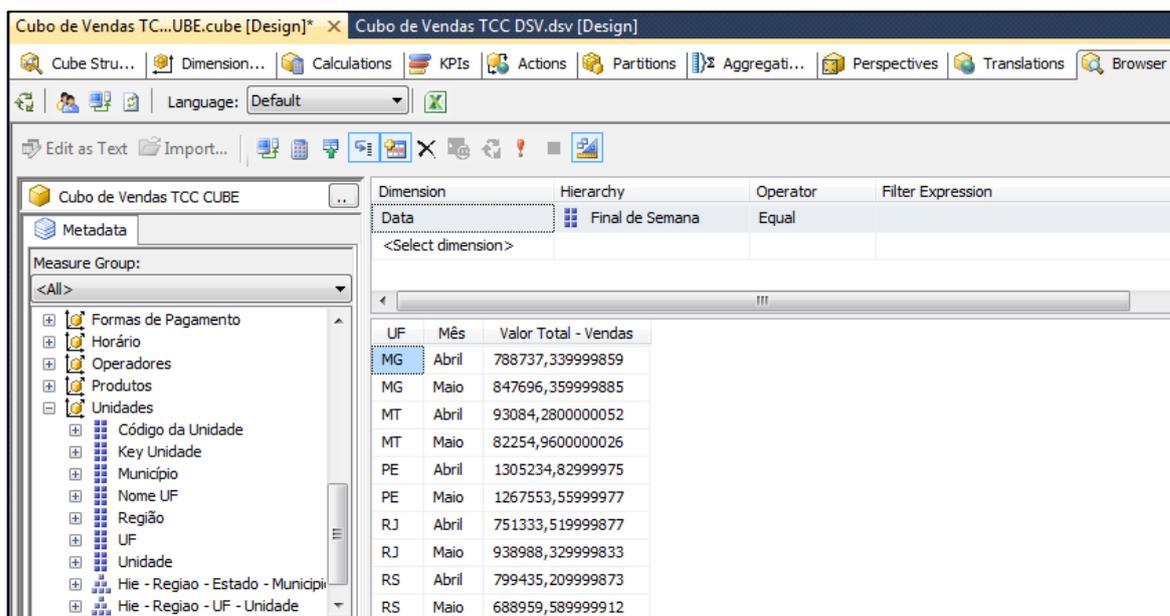


Ilustração 28 Cube Designer - Paleta Browser

No *Browser* é possível visualizar as hierarquias e também testar as políticas de segurança do cubo alterando o usuário que está visualizando o cubo, por exemplo, se é criada uma política de segurança que o gerente de vendas de uma filial não pode ver os dados da outra, é possível logar no browser como gerente de vendas de uma unidade e tentar acessar os dados de outra filial para verificar se os dados são ou não exibidos. Esta ferramenta é de extrema importância durante o desenvolvimento de um cubo, pois permite ao desenvolvedor de BI testar sua solução ainda dentro do ambiente de desenvolvimento sem necessidade de utilizar ferramentas terceiras de consulta.

3.3 Tabelas dinâmicas do Microsoft Excel

A ferramenta BIDS apesar de possuir o *browser* para visualizar os cubos, não é uma ferramenta muito amigável para usuários sem conhecimentos avançados. Então como possibilitar a análise de estruturas multidimensionais para usuários comuns? Uma das

respostas possíveis é através da utilização das *Pivot Tables* e *Pivott Charts* ou tabelas e gráficos dinâmicos do Microsoft Excel. Talvez o recurso analítico mais poderoso do Excel seja a tabela dinâmica, com ela é possível cruzar dados de diferentes listas como produtos e regiões por exemplo (JANUS e FOUNCHÉ, 2010).

O Excel tem a capacidade de se conectar a um cubo do SSAS gerando uma estrutura de tabela dinâmica (*PivotTable*) que permite analisar dimensões, grupos de medidas e medidas de um cubo. Também existe a possibilidade de importação de KPIs do SSAS. A chave para a conexão entre o Excel e o SSAS está na paleta “Dados” do Excel, nela existe a opção “obter dados externos” com o botão “De outras fontes”. Ao clicar neste botão são listadas diversas opções, dentre elas a opção “Dos serviços de análise”. Na Ilustração 29 é exibida esta opção.

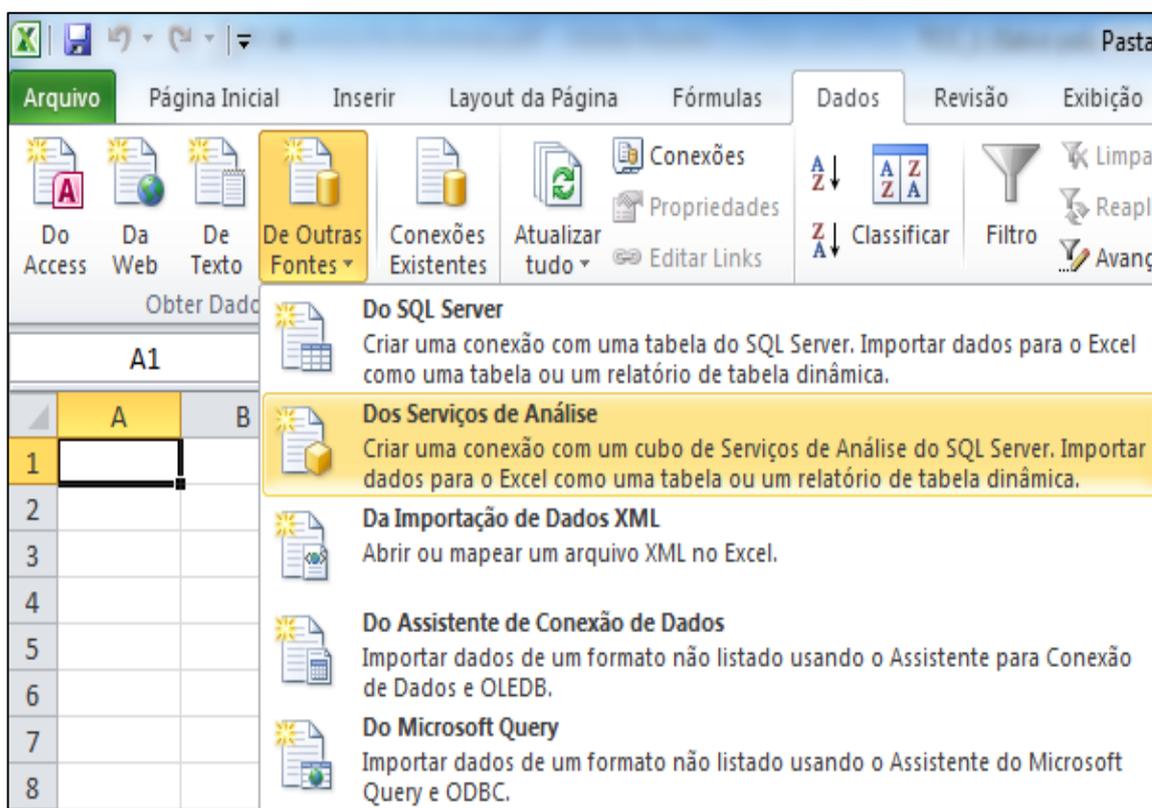


Ilustração 29 Conexão do Excel com os serviços de análise do SSAS

Após selecionar a opção para obter dados dos serviços de análise, o Excel exibe uma janela para configurar o nome do servidor de onde serão obtidos os dados e o usuário que irá se conectar no servidor. Na Ilustração 30 é exibida esta tela.

O mecanismo de perspectivas do SSAS, bem como o mecanismo de segurança, utilizarão o usuário informado na tela de configuração do servidor para definir quais dados poderão ser exibidos.

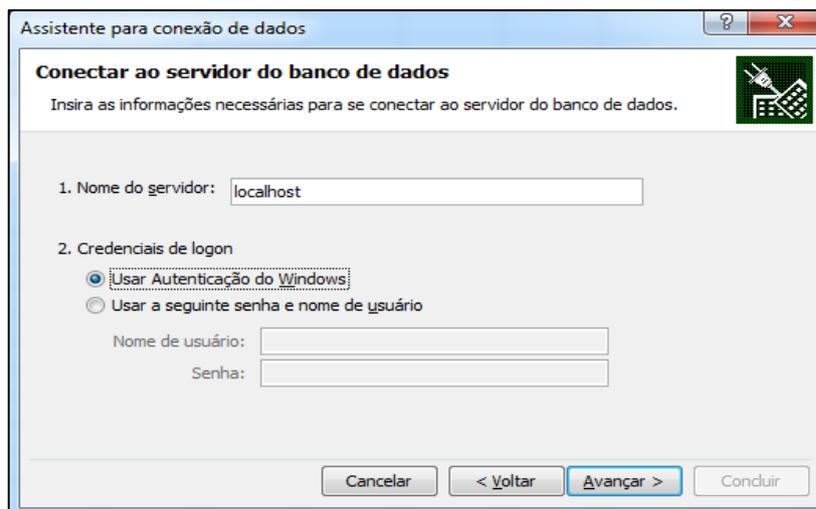


Ilustração 30 Configuração do servidor de dados

Após informar o usuário que irá se conectar ao serviço de análise, o Excel irá exibir uma janela listando os bancos de dados de análise que a instância do SSAS disponibilizou para o usuário autenticado. Deverá ser selecionado o banco de dados que o usuário irá utilizar e após isso o cubo de dados que será utilizado como pode ser visto na Ilustração 31.

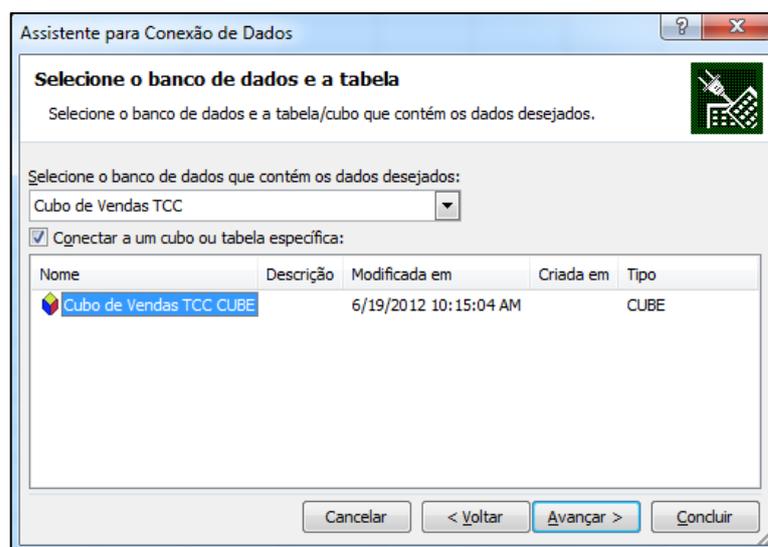


Ilustração 31 Seleção de cubo para análise de dados no Excel

Após a seleção do cubo irá aparecer uma janela onde o usuário poderá personalizar onde deseja gravar as informações de conexão ou simplesmente clicar concluir para finalizar o processo.

Com a conexão configurada o Excel irá questionar sobre como deverão ser importados os dados, deverá ser selecionada a opção de tabela dinâmica e então a tela inicial será exibida. Na Ilustração 32 é exibida a tela inicial para análise dos dados do SSAS, à direita da janela, existe uma caixa onde o usuário pode selecionar os grupos de medidas e as dimensões que deseja utilizar.

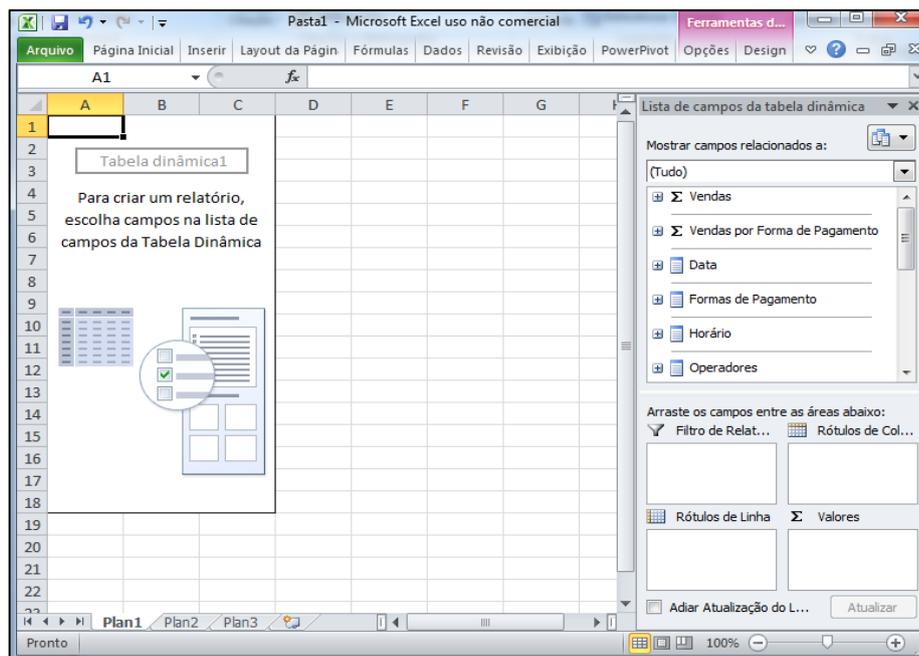


Ilustração 32 Tela inicial da tabela dinâmica

A vantagem na utilização das tabelas dinâmicas é de que além de serem fáceis de utilizar, elas conseguem importar todas as formatações criadas no *Analysis Services*, e permitem que o usuário interaja com os dados.

A criação de gráficos é extremamente simples, ao adicionar um novo gráfico com a tabela dinâmica selecionada ele exibe automaticamente os dados da tabela e sofre alterações conforme o usuário interage com a tabela.

Um recurso muito interessante que complementa as tabelas dinâmicas é a utilização de segmentações de dados. A segmentação de dados funciona como um filtro suspenso, o usuário adiciona uma segmentação de dados clicando na tabela dinâmica, depois a paleta “Opções” e por último clicando no botão “Inserir segmentação de dados” conforme é exibido na Ilustração 33.

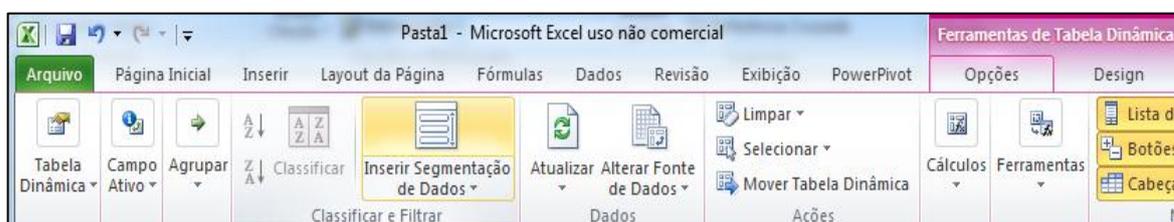


Ilustração 33 Seleção de segmentação de dados

Após selecionar a opção para inserir uma nova segmentação de dados o Excel irá questionar sobre quais atributos de dimensão o usuário irá querer segmentar. Conforme é exibido na Ilustração 34.

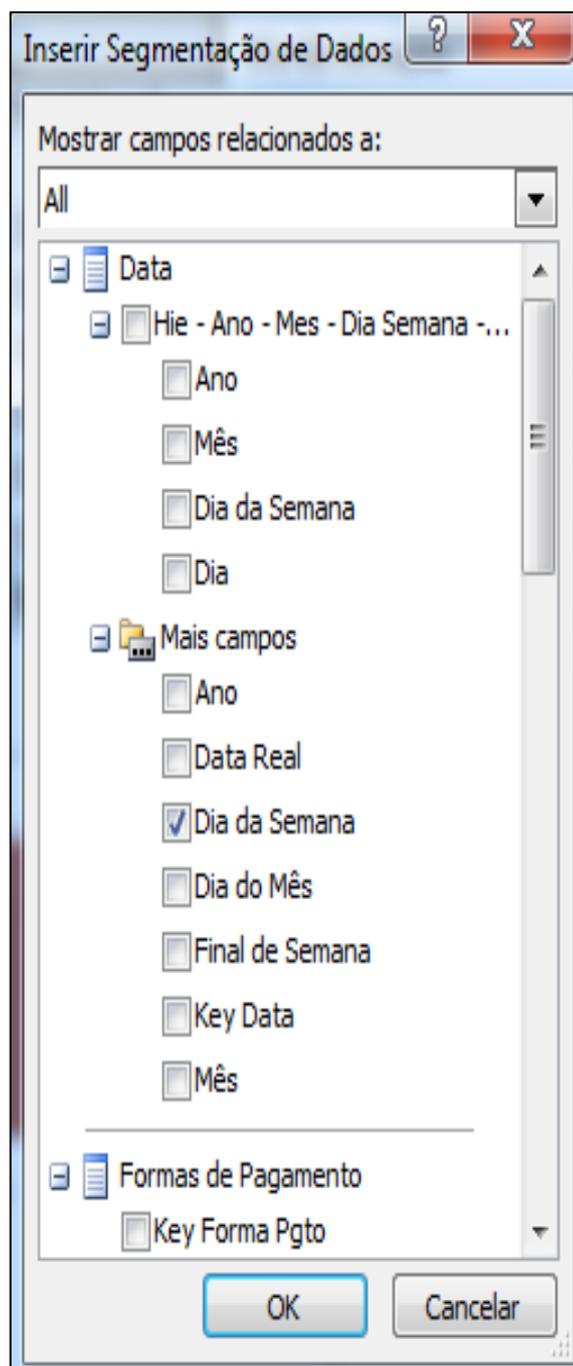


Ilustração 34 Seleção de atributos para segmentação de dados

Quando o usuário selecionar os atributos o Excel irá inserir uma segmentação para cada atributo. Basta então o usuário colocar na posição de tela que preferir e então poderá utilizar a segmentação.

Na Ilustração 35 é exibido um exemplo de tabela dinâmica com um gráfico e uma segmentação de dados por dia da semana. Neste caso conforme forem selecionados os dias da semana na segmentação, eles serão adicionados ou removidos da exibição tanto da tabela quanto do gráfico dinâmico.

A segmentação de dados acaba se tornando um recurso extremamente útil para comparação de dados por período e até mesmo filtragem de dados.

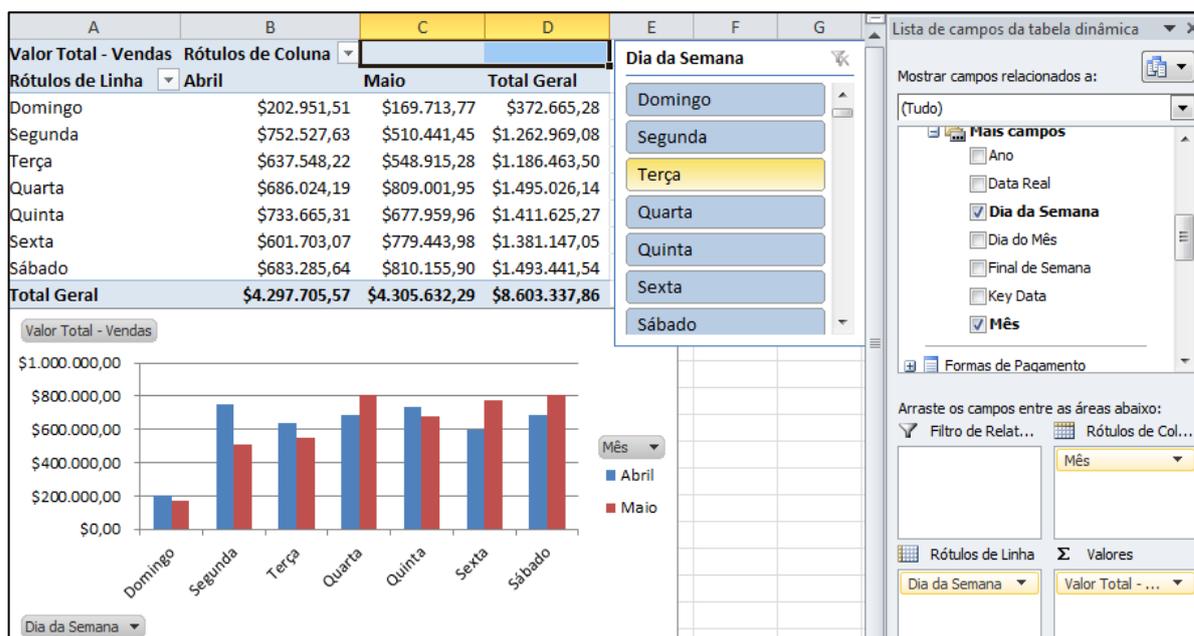


Ilustração 35 Exemplo de tabela dinâmica com segmentação de dados e gráfico

3.4 Considerações Finais

As ferramentas Microsoft para desenvolvimento de BI cobrem todos os principais aspectos no desenvolvimento de uma solução deste porte. Além dos aspectos mencionados neste capítulo, elas possibilitam a integração com ferramentas de gerenciamento de código para criar ambientes colaborativos de desenvolvimento. Também permitem integração de políticas de segurança com o banco de dados.

Durante os testes com estas ferramentas (sem um conhecimento prévio das mesmas), a curva de aprendizado foi extremamente rápida. A ferramenta *Integration Services* pode ser facilmente utilizada por um usuário com prévios conhecimentos de programação e banco de dados. A montagem dos fluxos de controle e dados é consideravelmente rápida e a execução funciona de forma muito satisfatória.

Quanto ao *Analysis Services* para a montagem dos cubos, com os conhecimentos prévios de modelo dimensional e DW, foi extremamente simples a montagem do cubo pois a própria nomenclatura dos objetos já facilita esta montagem.

Para a exibição dos dados podem ser utilizadas outras ferramentas como o *SQL Server Reporting Services (SSRS)* e também o *Microsoft Sharepoint*, porém não houve tempo

suficiente para testar estas soluções. O *Sharepoint* com o lançamento do SQL Server 2012 recebeu complementos interessantíssimos para a visualização de dados, e na continuação deste trabalho deverá ser utilizado.

O Excel com as tabelas dinâmicas mostrou-se eficiente para desenvolver um BI para usuários comuns visto que a maioria destes usuários analisa relatórios gerando pelos seus sistemas de informação dentro do próprio excel através da exportação dos mesmos. Logo com um simples treinamento de como se conectar a um cubo de dados provavelmente não haverá maiores dificuldades em sua utilização.

Durante os testes com a ferramenta *integration Services* no Sistema Operacional Windows 7 em sua versão de 64 bits, houveram problemas na conexão de dados com o servidor de banco de dados Oracle. Ocorre que ao instalar o *Integration Services* em um servidor de 64 bits, ele utiliza um executável de 32 bits, o Windows por sua vez, armazena este executável na pasta “C:\Arquivos de Programas (x86)”. Este é o caminho padrão para aplicativos de 32bits instalados no sistema operacional de 64 bits.

O *driver* para conexão com o banco de dados do Oracle antes da versão do Oracle 11G R2, não conseguia entender diretórios que contivessem caracteres como parênteses, logo, ao receber a conexão provinda de um aplicativo existente na pasta de arquivos 32 bits do Windows, o driver acabava não conseguindo realizar a conexão por não entender a origem do executável que a chamava. Com o lançamento da versão 11G R2 do Oracle, a Oracle desenvolveu um driver de 32 bits para sistemas de 64 bits que consegue entender estes diretórios e então solucionar este problema. É importante salientar que esta explicação não é encontrada em documentações tanto da Oracle quanto da Microsoft, o problema foi corrigido, porém não foi divulgada oficialmente, logo esta solução foi encontrada através da comunicação com especialistas em ETL em diversos fóruns virtuais.

4 MODELAGEM DO BI

Neste capítulo é apresentada a modelagem do BI. Para que esta modelagem seja compreendida, na primeira seção é descrita a arquitetura do sistema de gestão de vendas, como é a estrutura de rede utilizada pelos clientes e o processo de integração dos dados entre matriz e filiais. Sem estes conceitos não é possível entender a maneira pela qual foi modelado o DW e seu processo de ETL. Após explicar a arquitetura do sistema, a seção seguinte descreve como foi definido o processo de negócio e por que este processo foi escolhido.

Com a arquitetura do sistema e o processo de negócio em mente será possível compreender o modelo multidimensional, tabelas de fato e dimensões explicados as seções seguintes.

4.1 Arquitetura do Sistema de Gestão de Vendas

O sistema de gestão de vendas é dividido em dois softwares, sistema de ponto de venda (PDV) e sistema de retaguarda. O primeiro, é o sistema que está presente em cada caixa ou ponto de venda dos clientes, é nele que são realizadas as vendas diárias. O segundo, é um sistema destinado a gerência da loja, serve para visualizar informações agregadas de todos os caixas, como total de vendas diárias, caixas que estão em operação, entre outras.

O sistema do PDV por questões de segurança necessita trabalhar de forma independente do sistema da retaguarda, ou seja, ele deve armazenar os dados no mesmo computador que opera para isso é utilizado um banco de dados chamado Java DB (*database*). Toda transação de venda registrada no PDV é gravada em seu banco de dados local.

Para que o sistema da retaguarda saiba quantas vendas foram realizadas, quais caixas estão operando, e também as demais informações que este sistema possui, é necessário realizar uma integração entre os dados armazenados em todos os PDV e o sistema de retaguarda. Para isso, é realizada de tempos em tempos uma integração de dados através do envio dos dados por parte do PDV para o sistema da retaguarda, ou seja, existe um processo no computador que executa concorrentemente ao processo do PDV que envia as informações do banco de dados Java DB, para o banco de dados Oracle do sistema da retaguarda. O

processo de integração dos dados conecta-se diretamente ao banco de dados do sistema de retaguarda e armazena suas informações.

Concluída a integração de dados entre PDV e retaguarda, inicia a integração dos dados entre filial e matriz. A integração dos dados entre cada filial e a matriz da empresa, ocorre através do acesso do banco de dados da matriz pelo processo de integração de dados através de uma rede VPN (*Virtual Private Network*).

Na Ilustração 36 é exibida a arquitetura de rede descrita anteriormente, onde fica visível a conexão entre o PDV e um banco de dados Java DB, o banco de dados Oracle presente em cada unidade, e o servidor da matriz conectado por VPN a cada unidade.

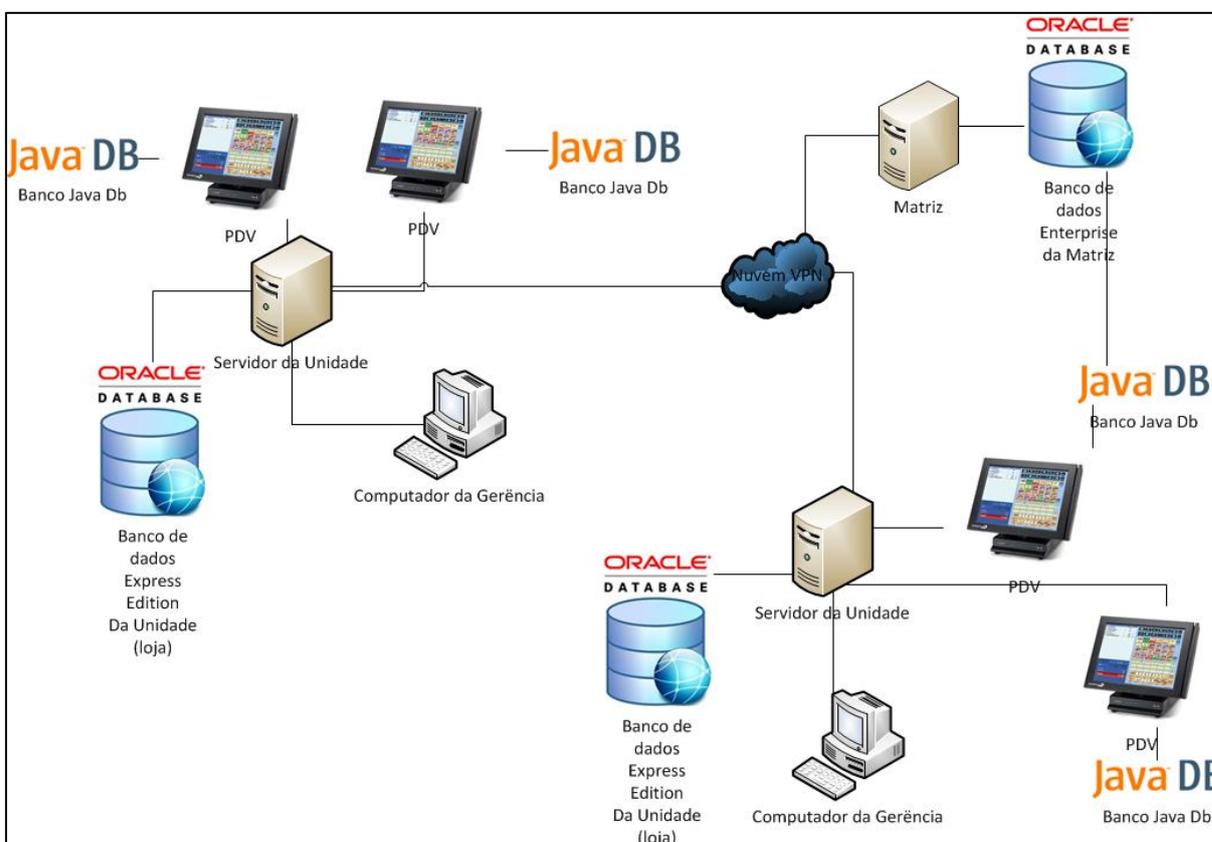


Ilustração 36 Arquitetura de rede de clientes

Fonte: NL Informática

Devido a questões de custo de licenciamento do banco de dados Oracle, os clientes adotam uma estratégia simples que consiste em instalar em suas unidades bancos de dados na versão gratuita com menos recursos, e somente licenciar o banco de dados da matriz.

Como dito anteriormente os bancos de dados gratuitos possuem menos recursos, tais como armazenamento. Como tratamos de um sistema de vendas onde ocorrem milhares de transações diárias por loja, rapidamente o limite de armazenamento do banco de dados chega ao fim. Para continuar a utilizar o banco de dados, é necessário excluir os dados existentes de

forma que possam ser inseridos novos dados. Este processo de exclusão de dados para liberar espaço no banco de dados é denominado de Expurgo.

O Expurgo é configurado em cada cliente de acordo com a necessidade, ou seja, é avaliado um determinado período em que o banco de dados irá chegar ao limite através de uma análise prévia do volume de vendas. Neste ponto verifica-se a importância da integração dos dados entre a unidade e a matriz, pois quando os dados são excluídos da unidade, se não estiverem na matriz, somente é possível recuperá-los através do acesso ao banco de dados local do PDV. Existem mecanismos no processo de expurgo para verificar se os dados já foram integrados na matriz antes de apagá-los.

Na Ilustração 38 é possível verificar este fluxo de dados.

Ilustração 37 Fluxo de dados Gestão de Vendas

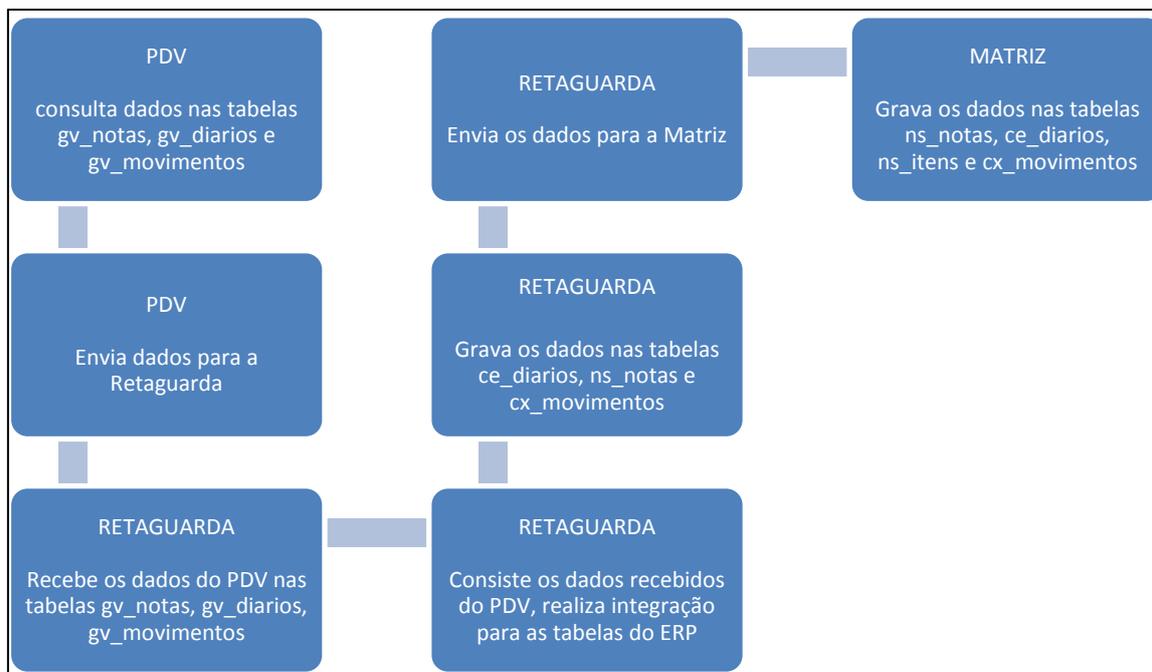


Ilustração 38 Fluxo de dados do Sistema de gestão de vendas

4.2 Definição do Processo de Negócio

Nesta seção é explicado como foi definido o processo de negócio para a criação do BI, a estratégia adotada e os objetivos deste processo. Para definição do processo de negócio a ser coberto pelo BI, foram realizadas reuniões com gerentes e consultores, onde estes identificaram as maiores demandas de análise de dados por parte dos clientes. Foi descoberto

que o processo de negócio que atualmente necessita de um BI para possibilitar uma melhor análise por parte dos usuários dentro do NL Gestão é o processo de Vendas. Através destes questionamentos, foram identificadas as seguintes análises na área de vendas:

1. Horários de picos de vendas;
2. Vendas por forma de pagamento;
3. Vendas por unidade da empresa;
4. Vendas por região;
5. Ciclo de vida de produtos.

4.3 Modelo Multidimensional

Para possibilitar as análises definidas na seção anterior foi criado o cubo de dados presente na Ilustração 39, este cubo possui duas tabelas de fato, vendas e vendas por forma de pagamento.

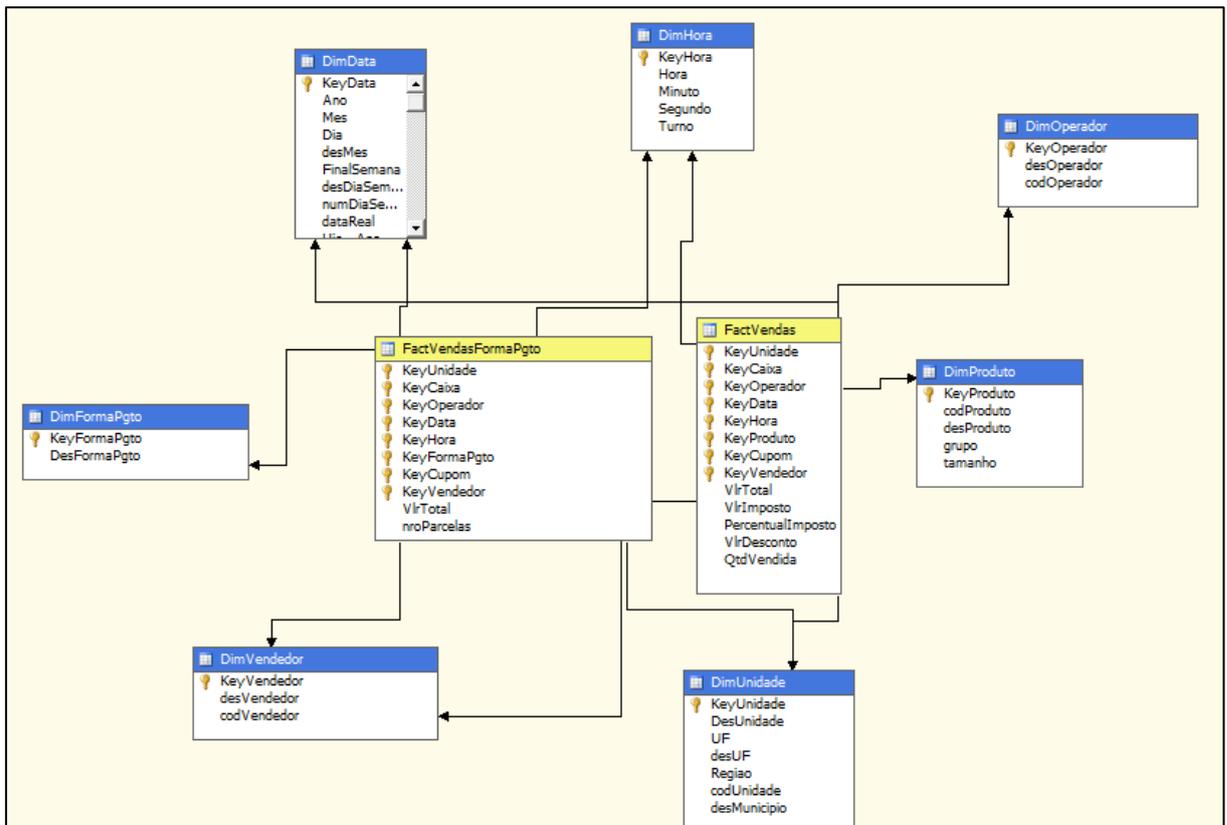


Ilustração 39 Cubo de Vendas

4.4 Tabelas de Fato

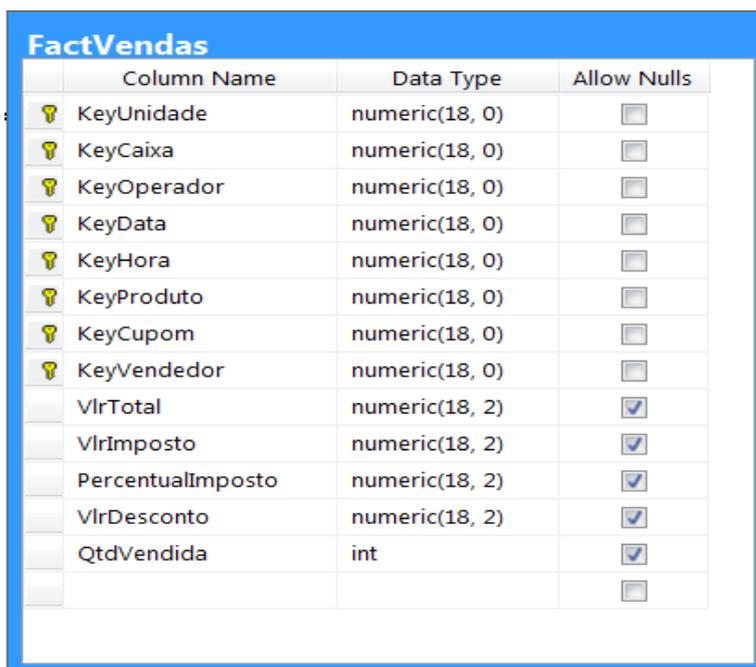
Nesta seção são apresentadas as tabelas de fato definidas para o modelo multidimensional e suas respectivas medidas. Devido ao tempo e complexidade de modelagem para um fato que contemple o ciclo de vida de produtos, este deverá ser realizado posteriormente na empresa.

Considerando o nível de detalhe em produto e forma de pagamento, foi identificada a seguinte situação: Uma venda pode ser paga através de diversas formas de pagamento, logo não é possível definir quais itens foram pagos através de qual forma de pagamento, somente o total que esta forma de pagamento representa em uma venda.

A partir da situação descrita, concluiu-se que a granularidade do DW deve ser em nível de produto vendido, e em nível de forma de pagamento utilizada. Para contemplar esta granularidade, foi necessária a definição de dois fatos: vendas e vendas por forma de pagamento.

A Ilustração 40 exibe o modelo da tabela de fato de vendas com as medidas de:

1. Valor total (vlrTotal);
2. Valor de Imposto (vlrImposto);
3. Percentual de Impostos (PercentualImposto);
4. Desconto concedido (vlrDesconto);
5. Quantidade vendida (qtdVendida).



| Column Name | Data Type | Allow Nulls |
|-------------------|----------------|-------------------------------------|
| KeyUnidade | numeric(18, 0) | <input type="checkbox"/> |
| KeyCaixa | numeric(18, 0) | <input type="checkbox"/> |
| KeyOperador | numeric(18, 0) | <input type="checkbox"/> |
| KeyData | numeric(18, 0) | <input type="checkbox"/> |
| KeyHora | numeric(18, 0) | <input type="checkbox"/> |
| KeyProduto | numeric(18, 0) | <input type="checkbox"/> |
| KeyCupom | numeric(18, 0) | <input type="checkbox"/> |
| KeyVendedor | numeric(18, 0) | <input type="checkbox"/> |
| VlrTotal | numeric(18, 2) | <input checked="" type="checkbox"/> |
| VlrImposto | numeric(18, 2) | <input checked="" type="checkbox"/> |
| PercentualImposto | numeric(18, 2) | <input checked="" type="checkbox"/> |
| VlrDesconto | numeric(18, 2) | <input checked="" type="checkbox"/> |
| QtdVendida | int | <input checked="" type="checkbox"/> |
| | | <input type="checkbox"/> |

Ilustração 40 Fato de Vendas

Na Ilustração 41 é verificado o modelo da tabela de fato de vendas por forma de pagamento, é possível perceber que em relação à tabela de vendas, não existe a chave de produto, esta chave foi substituída pela chave de forma de pagamento.

Também foram removidas as medidas de desconto, quantidade, imposto e percentual de imposto, porém foi adicionada uma nova medida, o número de parcelas em que foi realizada a venda.

| | Column Name | Data Type | Allow Nulls |
|---|--------------|----------------|-------------------------------------|
| 🔑 | KeyUnidade | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyCaixa | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyOperador | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyData | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyHora | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyFormaPgto | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyCupom | numeric(18, 0) | <input type="checkbox"/> |
| 🔑 | KeyVendedor | numeric(18, 0) | <input type="checkbox"/> |
| | VlrTotal | numeric(18, 2) | <input checked="" type="checkbox"/> |
| | QtdVendida | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 41 Fato de Vendas por Forma de Pagamento

4.5 Definição das Dimensões

Nesta seção são apresentadas as dimensões definidas para visualização dos dados contidos nas tabelas de fato. Após definir a granularidade dos dados, foram identificadas as seguintes dimensões para visualização dos dados:

1. Unidade da empresa;
2. Data da venda;
3. Hora da venda;
4. Produto vendido;
5. Operador de caixa;
6. Vendedor;
7. Forma de pagamento;

8. Caixa da unidade;
9. Cupom fiscal da venda.

A dimensão de unidade da empresa representa cada loja da empresa, permitindo que os valores totais de venda sejam analisados loja a loja, bem como pelos demais atributos desta dimensão:

1. Código da unidade;
2. Descrição da unidade;
3. Unidade Federativa (UF);
4. Descrição da unidade federativa;
5. Região do país;
6. Município.

Com estes atributos será possível que sejam analisadas vendas em uma região do país, estado, ou município, identificando produtos vendidos de acordo com a cultura da região.

Na Ilustração 42 é exibida o modelo da dimensão de unidade.

| DimUnidade | | | |
|------------|--|----------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| | DesUnidade | varchar(50) | <input checked="" type="checkbox"/> |
| | UF | varchar(2) | <input checked="" type="checkbox"/> |
| | desUF | varchar(50) | <input checked="" type="checkbox"/> |
| | Regiao | varchar(50) | <input checked="" type="checkbox"/> |
| | codUnidade | numeric(7, 0) | <input type="checkbox"/> |
| | desMunicipio | varchar(50) | <input checked="" type="checkbox"/> |
| |  KeyUnidade | numeric(18, 0) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 42 Dimensão de Unidade

A dimensão de data da venda é a dimensão de tempo do DW, ela garante a visualização dos dados em diferentes períodos, possibilita a análise das vendas em períodos como, por exemplo, final de semana, determinado dia da semana, comparação de períodos entre outras. Esta dimensão possui os seguintes atributos:

1. Ano;
2. Mês;
3. Dia;
4. Descrição do mês;
5. Indicador de final de semana (Sim, Não);

6. Descrição do dia da semana;
7. Número do dia da semana.

Na Ilustração 43 é exibido o modelo da dimensão de data.

| DimData | | | |
|---------|---|----------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| | Ano | int | <input checked="" type="checkbox"/> |
| | Mes | int | <input checked="" type="checkbox"/> |
| | Dia | int | <input checked="" type="checkbox"/> |
| | desMes | varchar(20) | <input checked="" type="checkbox"/> |
| | FinalSemana | varchar(3) | <input checked="" type="checkbox"/> |
| | desDiaSemana | varchar(20) | <input checked="" type="checkbox"/> |
| | numDiaSemana | int | <input checked="" type="checkbox"/> |
| |  KeyData | numeric(18, 0) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 43 Dimensão de Data

A dimensão de hora permitirá identificar horários de picos de vendas e horários em que a loja não precisa de muitos funcionários para operar por exemplo. Para a dimensão de hora, foram definidos os seguintes atributos:

1. Hora;
2. Minuto;
3. Segundo;
4. Turno da venda (Manhã, tarde, noite).

Na Ilustração 44 é exibido o modelo da dimensão de hora.

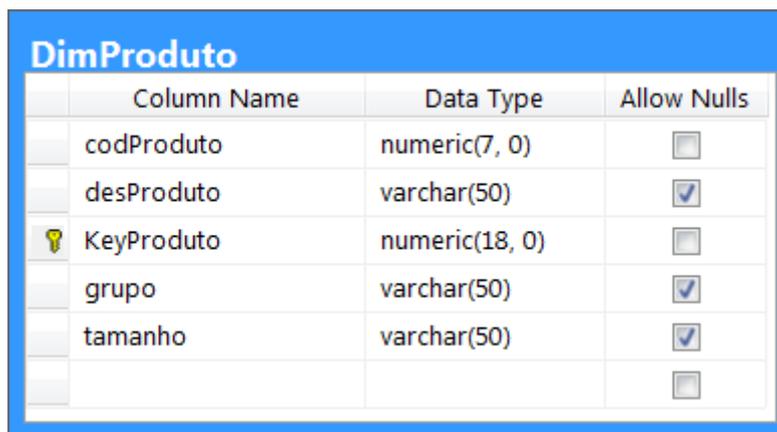
| DimHora | | | |
|---------|---|----------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| |  KeyHora | numeric(18, 0) | <input type="checkbox"/> |
| | Hora | int | <input checked="" type="checkbox"/> |
| | Minuto | int | <input checked="" type="checkbox"/> |
| | Segundo | int | <input checked="" type="checkbox"/> |
| | Turno | varchar(20) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 44 Dimensão de hora

A dimensão de produto vendido tem como objetivo analisar as vendas de acordo com os detalhes de cada produto vendido, para isto foi definido os seguintes atributos:

1. Código do produto;
2. Descrição do produto;
3. Grupo de produtos;
4. Tamanho do produto.

Na Ilustração 45 é exibido o modelo da dimensão de produto.



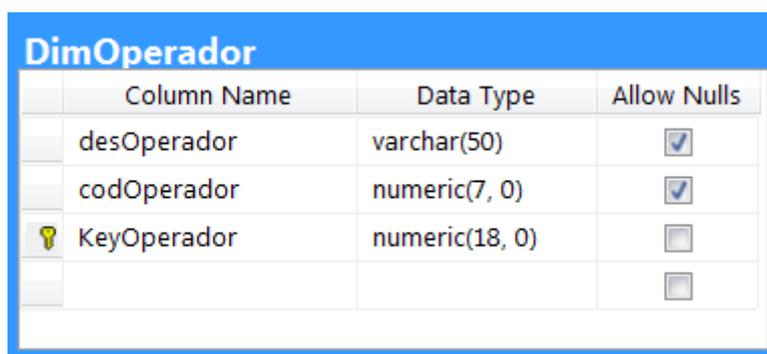
| | Column Name | Data Type | Allow Nulls |
|--|--|----------------|-------------------------------------|
| | codProduto | numeric(7, 0) | <input type="checkbox"/> |
| | desProduto | varchar(50) | <input checked="" type="checkbox"/> |
| |  KeyProduto | numeric(18, 0) | <input type="checkbox"/> |
| | grupo | varchar(50) | <input checked="" type="checkbox"/> |
| | tamanho | varchar(50) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 45 Dimensão de produto

A dimensão de operador de caixa inicialmente serve apenas para em algumas circunstâncias determinar quem foi o operador de caixa responsável pelo registro da venda.

1. Código do operador;
2. Nome do operador.

Na Ilustração 46 é exibido o modelo da dimensão de operador de caixa.



| | Column Name | Data Type | Allow Nulls |
|--|---|----------------|-------------------------------------|
| | desOperador | varchar(50) | <input checked="" type="checkbox"/> |
| | codOperador | numeric(7, 0) | <input checked="" type="checkbox"/> |
| |  KeyOperador | numeric(18, 0) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 46 Dimensão de Operador

Com a dimensão de vendedor é possível analisar as vendas de cada vendedor da unidade, nesta dimensão foi definido os seguintes atributos:

1. Código do vendedor;
2. Nome do vendedor.

Na Ilustração 47 é exibido o modelo de dimensão de vendedor.

| DimVendedor | | | |
|---|-------------|----------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
|  | KeyVendedor | numeric(18, 0) | <input type="checkbox"/> |
| | desVendedor | varchar(50) | <input checked="" type="checkbox"/> |
| | codVendedor | numeric(7, 0) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 47 Dimensão de Vendedor

A dimensão de forma de pagamento permite analisar o total de vendas por cada uma das formas de pagamento utilizadas na empresa. Nesta dimensão existe somente o atributo de descrição da forma de pagamento utilizada.

Na Ilustração 48 é exibido o modelo da dimensão de forma de pagamento.

| DimFormaPgto | | | |
|--|--------------|----------------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| | DesFormaPgto | varchar(50) | <input checked="" type="checkbox"/> |
|  | KeyFormaPgto | numeric(18, 0) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Ilustração 48 Dimensão de Forma de Pagamento

Para as dimensões de caixa da unidade e cupom fiscal foi utilizado o conceito de dimensão de degeneração explicado na seção 2.1.1. O cupom fiscal é um número de seis dígitos que pode repetir diversas vezes durante o ano em uma unidade, no contexto deste trabalho, ele é utilizado somente para agrupar os produtos em uma única venda.

Neste contexto o caixa não irá comportar atributos, pois durante o tempo a impressora do caixa pode ser trocada, invalidando um atributo de marca da impressora, bem como a contagem de cupons e outros dados variarem de acordo com a troca de *hardware*.

Para estas duas dimensões, poderiam ser adicionados atributos de tempo e então controlar as alterações, porém, neste momento, as informações destas dimensões foram julgadas desnecessárias.

4.6 Considerações Finais

Durante o processo de modelagem foram encontrados alguns desafios como, por exemplo, o de chegar a um modelo que pudesse satisfazer a maioria dos clientes. Foram

realizadas diversas reuniões para discutir os dados necessários não só para a área de vendas, mas para também outras áreas que o sistema NL Gestão abrange.

O modelo de cubo de dados de vendas apesar de parecer simples, contemplará diversas consultas e análises necessárias às empresas para identificar comportamentos de seus clientes.

Atualmente muitas das informações presentes no cubo de dados são obtidas nos clientes através de análise manual de cupons emitidos, ou consultas no banco de dados realizadas pelos departamentos de TI.

Foi considerado o objetivo de tornar o modelo simples o suficiente para que possa ser entendido por qualquer pessoa, e que atenda as demandas básicas dos clientes. Conforme este modelo for sendo utilizado, conseqüentemente sofrerá diversas mudanças.

A arquitetura de sistema contribuiu para a facilidade de modelagem, porém a arquitetura de rede dos clientes foi um complicador para o processo de ETL que é descrito no próximo capítulo.

5 CRIAÇÃO DO BI

Neste capítulo é descrito o processo de criação do BI, sua modelagem, carga de dados e apresentação das informações. Este processo foi realizado em cinco etapas, das quais primeiro foram criadas no banco de dados às tabelas definidas durante o processo de modelagem do BI. Após a criação das tabelas, foi desenvolvido o processo de ETL para realizar a carga inicial no banco de dados de *data warehouse*, após ser desenvolvido processo de ETL foi executado para realizar a carga inicial no DW. Posteriormente foi desenvolvido o cubo de dados com os dados já carregados no DW e

Para criar as tabelas no banco de dados do DW foi utilizada a ferramenta SSMS. Em um primeiro momento foi criado um banco de dados nomeado de NLDW. Para este banco de dados, foram criadas as tabelas definidas para o modelo dimensional apresentado na Ilustração 39.

Foi adotado um padrão de nomenclatura para tabelas de dimensão com o prefixo “Dim” e para as tabelas de fato o prefixo “Fact”. Na Ilustração 49 é possível verificar o banco de dados NLDW e suas tabelas no visualizador de objetos da ferramenta SSMS.

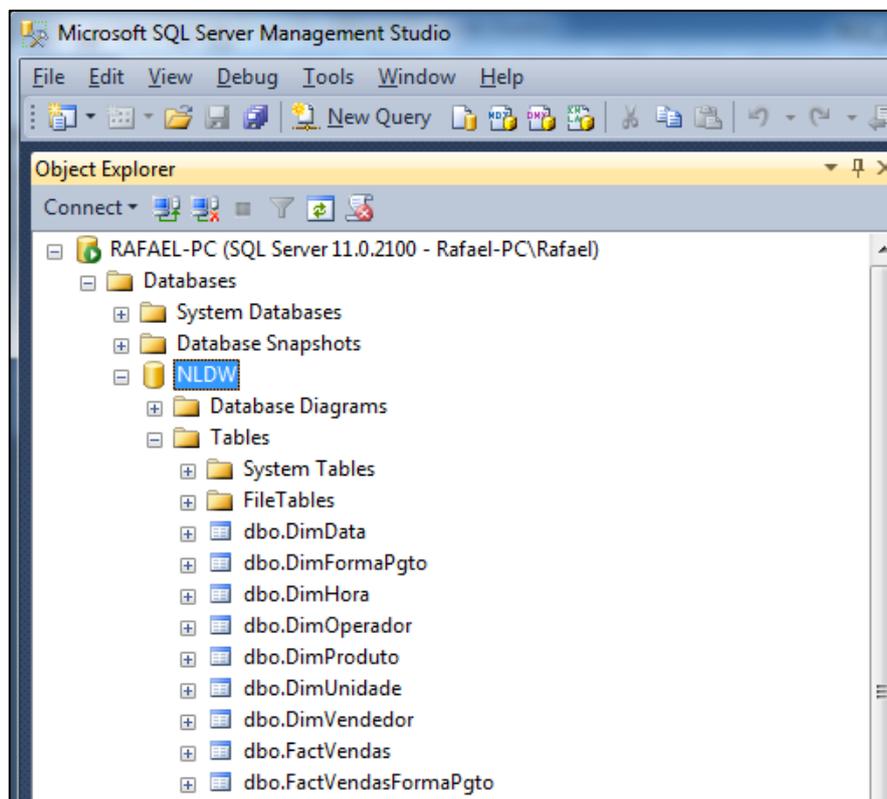


Ilustração 49 Tabelas do banco de dados NLDW

Como as tabelas de dimensão utilizam chaves substitutas, foi adotado um padrão, para que o campo de chave primária de cada dimensão seja um campo do tipo **NUMERIC(18)** e que o nome deste campo contenha o prefixo “Key” seguido do nome da dimensão sem o prefixo “Dim”. O campo deverá também ser configurado como **Identity** esta configuração serve para dizer ao SQL Server que este é um campo de auto incremento.

Na Ilustração 50 é exibido o desenho da tabela de dimensão de *data* na ferramenta SSMS. Nota-se que na imagem o campo KeyData está selecionado, logo abaixo dos campos, existe uma caixa chamada “*Column Properties*”, nesta caixa é possível verificar a marcação do campo KeyData como *is identity* igual a *yes*. Ou seja, a marcação deste campo como de auto-incremento, e para definir que o incremento será realizado de um em um utiliza-se a propriedade *Identity increment* definida com o valor um, neste caso, o valor do incremento.

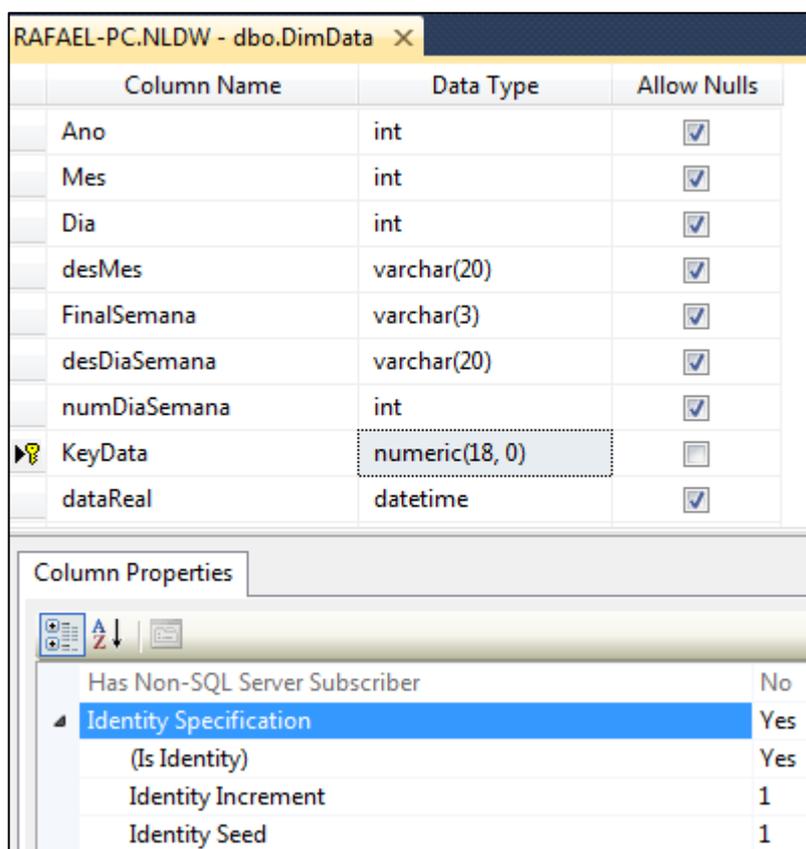


Ilustração 50 Chave da dimensão de Data com auto-incremento

5.1 Criação do Processo de Carga dos Dados

Nesta seção é apresentado o processo de carga dos dados no DW, primeiro é descrito o processo de carga de dimensões em seguida os processos de carga das tabelas fatos. Como

não é o objetivo deste trabalho a criação de um produto final para ser comercializado pela empresa, foi adotada a estratégia de realizar somente a carga inicial no DW. Portanto, possíveis alterações nas dimensões, e possíveis atualizações nos dados dos fatos não são tratadas neste processo.

O processo de ETL foi dividido em duas partes, na primeira é realizada a carga inicial das dimensões, e na segunda, é realizada a carga inicial nas tabelas fatos.

5.1.1 ETL das Dimensões

Nesta seção é descrito o processo de ETL utilizado para as dimensões do DW. Para a maioria dos clientes é mantido um ponto de venda com a retaguarda dentro da NL para funcionar como ambiente de testes e homologação de versões. O banco de dados da retaguarda que fica na NL recebe atualizações da matriz do cliente, porém não envia dados de volta ao cliente, sendo assim ele recebe novos produtos cadastrados, carga de preços, posições de estoque, operadores de caixa cadastrados, etc. Levando em consideração esta situação, o processo de ETL de dimensões foi definido para buscar dados do banco de dados hospedado dentro da NL.

Para criar a dimensão de data e de hora, foi desenvolvida uma rotina no Oracle, para que esta que gerasse estas informações em uma tabela idêntica a tabela de dimensão de *data* no DW, para que estes dados pudessem ser carregados ao DW.

Para as demais dimensões, foi utilizada a estratégia onde as informações são consultadas no banco de dados da retaguarda e em seguida é realizado um processo de consulta na tabela da dimensão pelo código de cada um destes itens, se o código não existir, é inserido o dado na dimensão gerando uma chave nova substituta. Este processo é exibido na Ilustração 51.

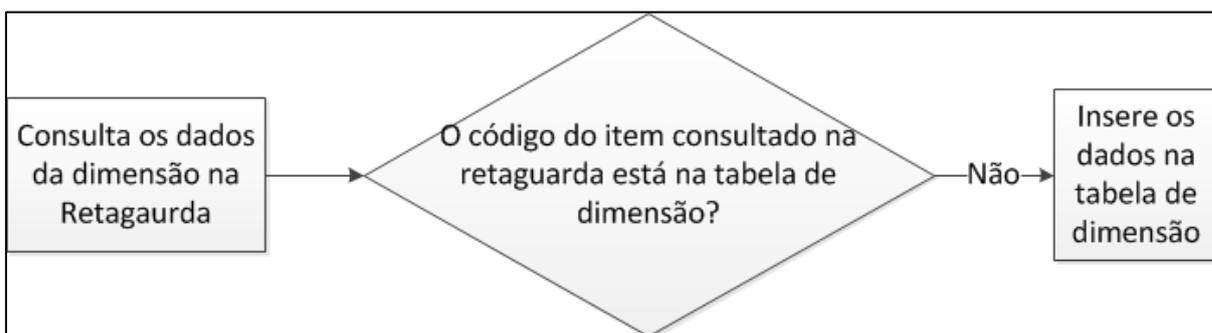


Ilustração 51 Fluxo de carga de dimensão

Na Ilustração 52 é exibido o processo de ETL de dimensão de unidade. Nota-se neste processo a utilização da saída de erros do componente de consulta. Esta saída é utilizada quando um dado não é encontrado na consulta. A linha que estava sendo utilizada para realizar a consulta é redirecionada para a saída de erro, logo só será inserido o dado que fizer a consulta falhar, ou seja, que não existe na dimensão.

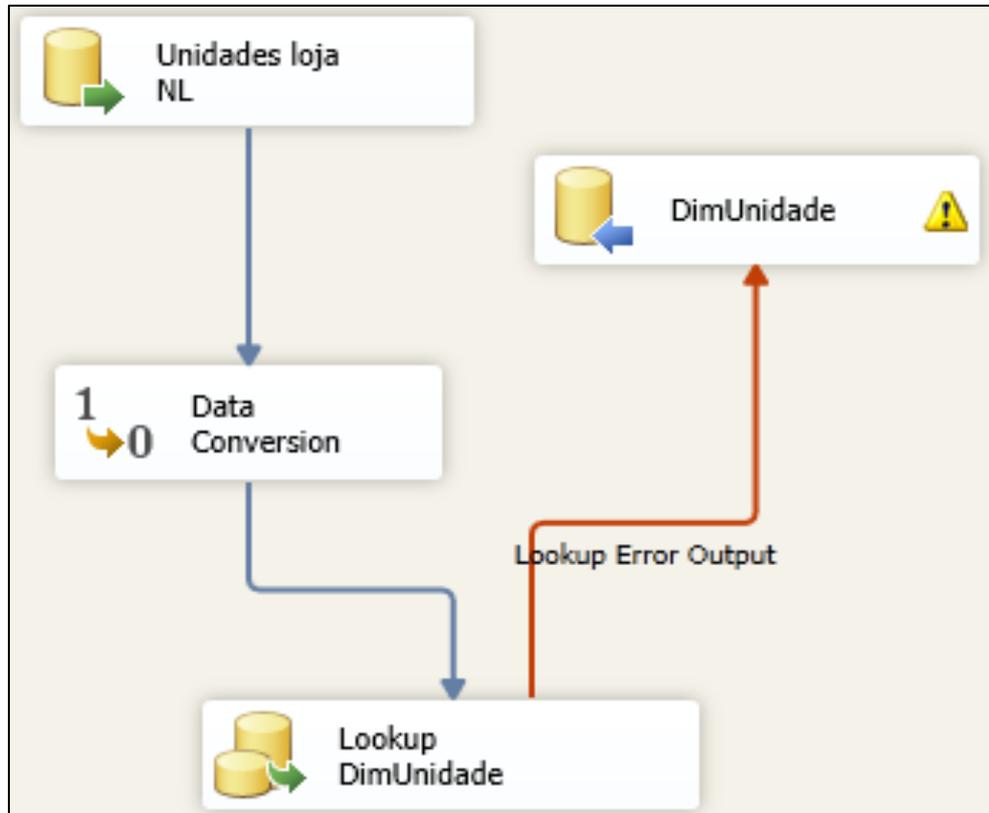


Ilustração 52 Fluxo da ETL de dimensão de unidade

Na Ilustração 53 é exibida a configuração do componente de consulta (“lookup DimUnidade”) onde diz que quando houver um erro de “lookup Match OutPut” ou seja, que o componente não conseguir realizar a consulta, a linha deve ser redirecionada.

| Input or Output | Column | Error | Truncation | Description |
|---------------------|------------|--------------|----------------|-------------|
| Lookup Match Output | | Redirect row | | Lookup |
| | codUnidade | | Fail component | Copy Column |

Ilustração 53 Configuração de saída de erro

5.1.2 ETL Vendas

Nesta seção é apresentado o processo de ETL do fato de vendas que consiste em buscar os dados em seis unidades de um cliente, transformar estes dados, buscar as chaves substitutas das dimensões e por fim inserir os dados no fato de vendas. Na Ilustração 54 é exibido o processo macro de ETL para o fato de vendas na ferramenta SSIS.

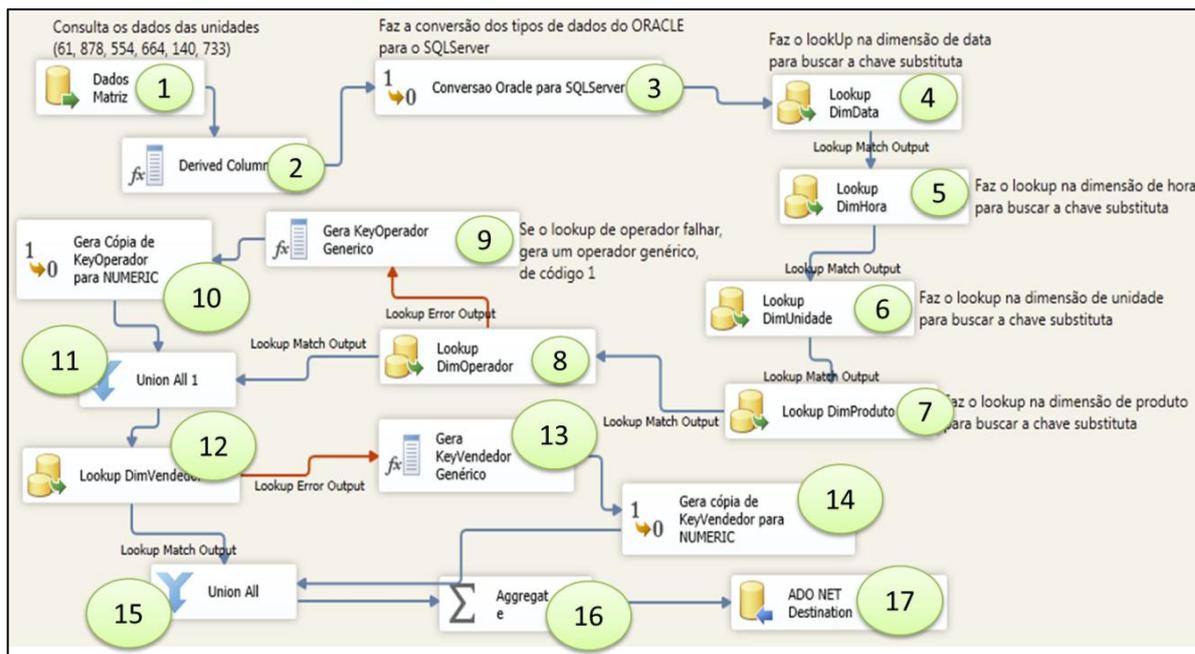


Ilustração 54 ETL para o fato de Vendas

Dentro da arquitetura de banco de dados do PDV e retaguarda, existem alguns detalhes a serem levados em consideração na estrutura de tabelas. Todas as movimentações geradas através do PDV, gravam dados na base local nas tabelas *gv_notas*, *gv_movimentos* e *gv_diarios*. A retaguarda, possui esta mesma estrutura de tabelas, de forma que o PDV para realizar a integração, somente necessita gravar uma cópia do dado na retaguarda.

Os dados analisados pela gerência não são obtidos através destas tabelas, quando a retaguarda recebe os dados do PDV, realiza um processo de consistência para verificar os dados recebidos, um exemplo das consistências é verificar se o valor total da venda corresponde ao valor de venda dos itens, como estão em tabelas diferentes podem ocorrer inconsistências na integração. Após realizar esta e outras consistências, os dados obtidos das tabelas do PDV são gravados nas tabelas *ns_notas*, *ns_itens*, *ce_diarios* e *cx_movimentos*.

As tabelas citadas anteriormente são as tabelas onde são gravados os dados definitivos de vendas na retaguarda, no banco de dados da matriz existem tabelas idênticas a estas, a retaguarda por sua vez, realiza uma cópia destes dados para as respectivas tabelas na matriz.

O processo de ETL inicia com a execução da consulta dos dados do cliente no banco de dados Oracle hospedado na matriz do cliente ver Ilustração 54 (1). Esta consulta utiliza a instrução SQL exibida na Ilustração 55. A consulta consiste em buscar cara item de cupom fiscal vendido na loja que não esteja cancelado.

```

SELECT diario.cod_unidade, nota.cod_caixa AS keycaixa, nota.cod_operador
,nota.dta_emissao, diario.cod_item, nota.num_nota as KeyCupom
,nota.cod_atendente, to_char(nota.dth_ini_venda, 'hh24') AS hora
,to_char(nota.dth_ini_venda, 'mm') AS minuto
,to_char(nota.dth_ini_venda, 'ss') AS segundo
,item.per_icms AS percentual, diario.vlr_total, diario.vlr_icms
,diario.vlr_desconto, diario.qtd_lancamento
FROM ns_notas nota
INNER JOIN ce_diarios diario
ON (nota.cod_emp = diario.cod_emp AND nota.cod_maquina = diario.cod_maquina AND
nota.cod_unidade = diario.cod_unidade AND
nota.num_nota = diario.num_documento AND
nota.cod_serie = diario.des_serie)
INNER JOIN ns_itens item
ON (diario.cod_maquina = item.cod_maq_ce AND
diario.num_seq = item.num_seq_ce)
WHERE nvl(diario.des_obs1, 0) = 0
AND (
(nota.dta_emissao BETWEEN to_date('01/01/2009', 'dd/MM/yyyy') AND
to_date('31/12/2012', 'dd/MM/yyyy')) )
AND nota.cod_unidade IN (69, 61, 878, 554, 664, 140, 733)|

```

Ilustração 55 Script SQL para consulta de dados para o Fato de Vendas

Após executar a consulta dos dados do cliente, o fluxo da ETL segue para o componente “*Derived Column*” exibido na Ilustração 54 (2) onde são adicionadas três novas colunas: Ano, Mês e Dia. Para as novas colunas são definidos dados através da execução de uma expressão de data, que irá extrair do campo de data da venda (DTA_EMISSAO) consultado no banco do cliente, os valores de ano mês e dia respectivamente.

Além das três colunas adicionadas, é executada uma operação lógica nos campos “cod_operador” e “cod_atendente” de forma que se um destes dois campos chegar nulo do cliente será atribuído o valor “1” para o mesmo.

Na Ilustração 56 é verifica-se as três novas colunas adicionadas e as duas colunas que recebem o valor “1” caso sejam nulas. A imagem exibe uma grade, onde na coluna da esquerda com o título “*Derived Column Name*” estão os campos que serão gerados, na coluna com o nome “*Expression*”, está a expressão que define o valor atribuído para este campo, estas expressões são expressões da linguagem *Transact SQL* utilizada no banco de dados SQLServer.

| Derived Column Name | Derived Column | Expression | Data Type |
|---------------------|-------------------------|--|--------------------------------|
| ANO | <add as new column> | YEAR(DTA_EMISSAO) | four-byte signed integer [...] |
| MES | <add as new column> | MONTH(DTA_EMISSAO) | four-byte signed integer [...] |
| DIA | <add as new column> | DAY(DTA_EMISSAO) | four-byte signed integer [...] |
| COD_OPERADOR | Replace 'COD_OPERADOR' | ISNULL(COD_OPERADOR) ? 1 : COD_OPERADOR | numeric [DT_NUMERIC] |
| COD_ATENDENTE | Replace 'COD_ATENDEN... | ISNULL(COD_OPERADOR) ? 1 : COD_ATENDENTE | numeric [DT_NUMERIC] |

Ilustração 56 Configurações de colunas derivadas na ETL de Vendas

Após gerar os novos campos através do componente de colunas derivadas, o processo de ETL entra no componente de conversão de dados exibido Ilustração 54 (3), neste componente são convertidos os tipos de dados do Oracle para os tipos de dados do SQL Server. Como a consulta que busca dados no cliente já retorna colunas com hora, minuto e segundo atribuídos. Não foi necessário executar expressões no componente de colunas derivadas. Porém os dados retornados do Oracle chegam com o tipo caractere e para comparar com os campos da dimensão de hora são necessários campos do tipo inteiro. Neste caso são gerados três novos campos inteiros para receber os dados de hora, minuto e segundo. Segue os nomes dos três campos novos: Copy of HORA, Copy of MINUTO, Copy of SEGUNDO. Na Ilustração 57 é exibida a conversão dos dados.

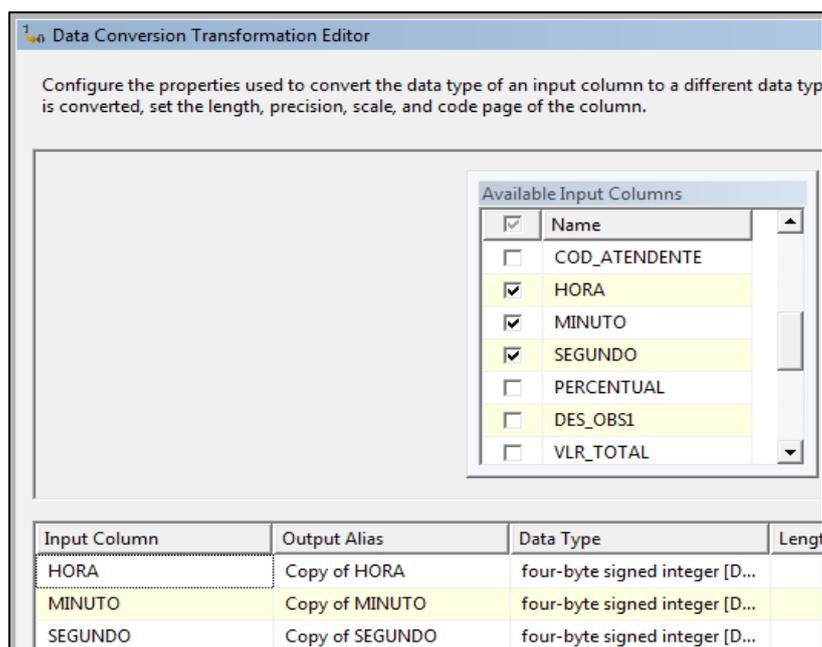


Ilustração 57 Conversão de tipos de dados do Oracle para o SQL Server

Após realizar os processamentos de conversão de dados, o processo de ETL inicia as consultas de chaves substitutas, sendo a primeira consulta realizada na dimensão de data exibida na Ilustração 54 (4). Para realizar esta consulta o processo utiliza as colunas Ano, Mês e Dia geradas no componente de colunas derivadas, para comparar com as colunas ano mês e

dia presentes na dimensão de data e retorna o campo KeyData caso esta comparação esteja correta.

Na Ilustração 58 é exibida a configuração da consulta na dimensão de data. À esquerda, estão os campos originados do campo de dados e as colunas derivadas adicionadas. Na direita estão os campos presentes na dimensão de data. A ligação entre os campos que devem ser comparados é realizada através de setas. O campo KeyHora está selecionado, neste caso significa que este será o campo retornado. Na grade presente na parte inferior da imagem existe a coluna de nome “*Lookup Operation*” onde é possível substituir um campo existente ou adicionar um novo campo ao fluxo de dados da ETL neste caso será adicionado um novo campo.

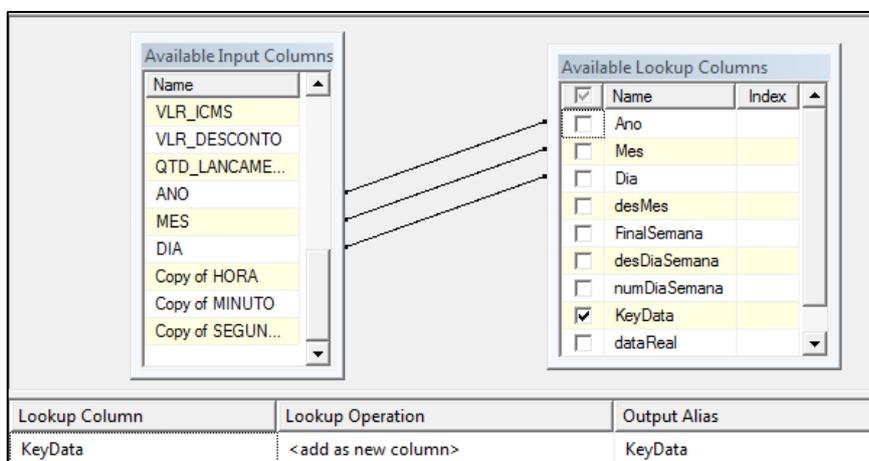


Ilustração 58 Consulta de chave substituta na Dimensão de Data

Após realizar a consulta da chave substituta na dimensão de data, o fluxo segue para a consulta da chave substituta da dimensão de hora exibido na Ilustração 54 (5). A lógica para consulta na dimensão de hora é a mesma da dimensão de data, porém os campos são diferentes. Os campos utilizados para consultar a dimensão de hora são os campos Copy of HORA, Copy of MINUTO, Copy of SEGUNDO, criados anteriormente no componente de transformação de dados. Na Ilustração 59 é exibida a configuração de chaves substitutas na dimensão de hora, é possível perceber na ilustração que o último campo do quadro do lado esquerdo é o campo com o nome “KeyData”, este campo foi adicionado anteriormente pela consulta realizada na dimensão de data.

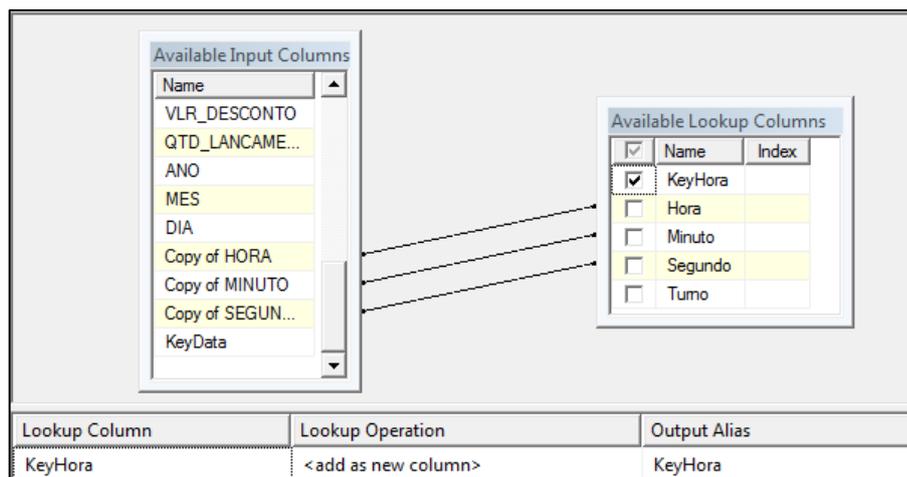


Ilustração 59 Consulta de chave substituta na dimensão de hora

Em seguida, é realizada a consulta da chave substituta na dimensão de unidade exibida na Ilustração 54 (6), novamente segue-se a mesma lógica da dimensão de data, somente alterando os campos. Na Ilustração 60 é possível verificar a comparação do campo “COD_UNIDADE” originado do cliente, com o campo “codUnidade” na tabela de dimensão de unidade.

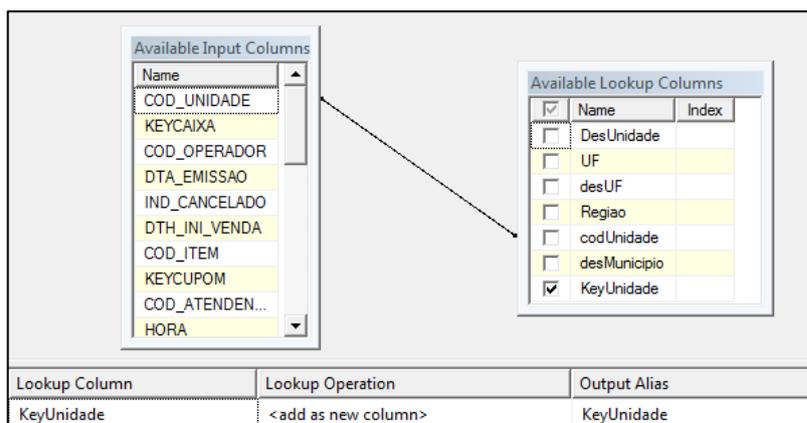


Ilustração 60 Consulta de chave substituta na dimensão de unidade

Após a consulta de chave substituta da dimensão unidade, o fluxo segue para a última consulta de chave substituta que atende a mesma lógica da dimensão de data exibida na Ilustração 54 (7). Neste caso, a dimensão de produto. Na Ilustração 61 é possível verificar a comparação do campo “COD_ITEM” com o campo “codProduto” na dimensão de produto.

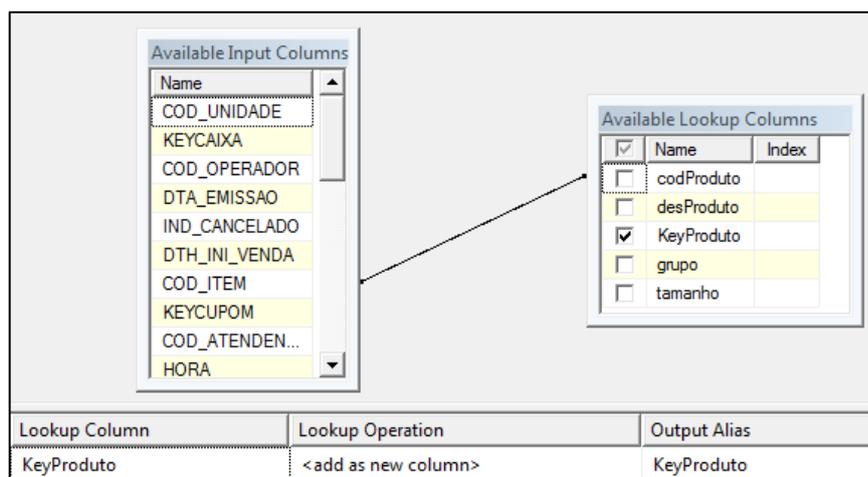


Ilustração 61 Consulta de chave substituta na dimensão de produto

Após passar pela consulta de chave substituta na dimensão de produto, o processo segue para a consulta de chaves substitutas de operador vendedor exibidas na Ilustração 54 (8 e 12). Porém estas últimas duas dimensões possuem algumas particularidades. Onde há muitas vendas o operador ou o vendedor não são registrados, ou então um operador ou vendedor registrado para a venda conforme o tempo não possuem mais informações no banco de dados, pois para estes campos não é mantida integridade referencial no banco de dados operacional. Logo, caso não exista um operador ou um vendedor é necessário informar um valor geral para que não sejam utilizados valores nulos na tabela de fatos (até porque a chave primária desta tabela não iria permitir isto).

Para que o processo de incluir um vendedor ou operador geral seja possível, são necessárias duas providências. Primeiro no processo de carga das dimensões seja definido um valor genérico na dimensão. E segundo utilizar a saída de erros do componente de *lookup*.

Na Ilustração 62 é exibido o processo de consulta de chave substituta para a dimensão de operador realizada em quatro etapas. Primeiro é realizada normalmente a consulta na dimensão de operador (mesma lógica da consulta de chave substituta da dimensão de data). Porém comparando o campo “COD_OPERADOR” do cliente com o campo “codOperador” da dimensão de operador. Caso a consulta encontre o operador a linha é direcionada para um componente de união que será explicado em seguida. Se o operador não for encontrado, a linha será redirecionada para a saída de erro. O componente de coluna derivada simplesmente irá gerar uma nova coluna com o nome “KeyOperador” de valor 1 (Ilustração 63).

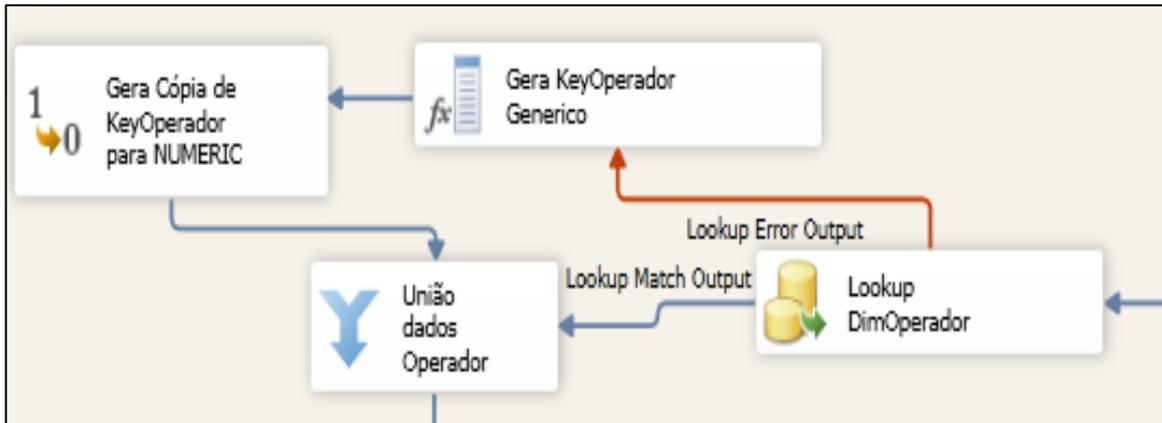


Ilustração 62 Processo de consulta de chave substituta para Dimensão de Operador

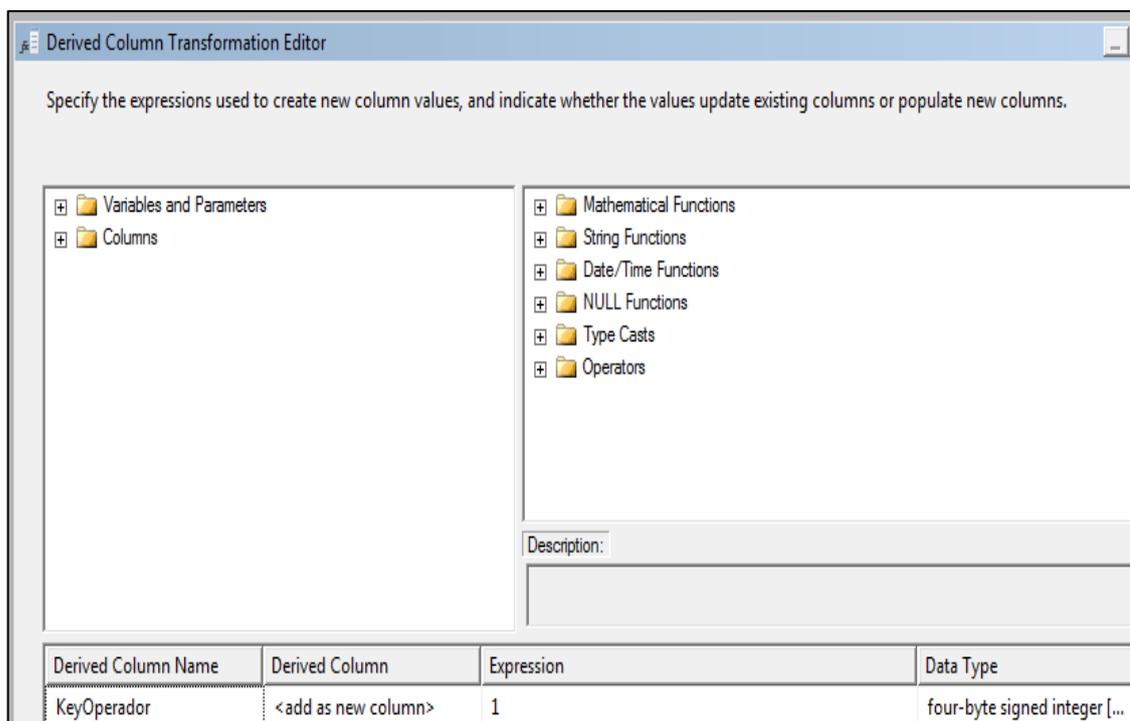


Ilustração 63 Adição da coluna derivada KeyOperator

Após adicionar a nova coluna na linha, o componente de coluna derivada (Ilustração 54 (9)) redireciona a linha para um componente de transformação de dados (Ilustração 54 (10)). Este componente irá transformar o campo inteiro gerado pelo componente de coluna derivada em um campo numérico com tamanho dezoito. Como o componente de transformação de dados não é capaz de substituir campos, ele irá gerar um novo campo de nome Copy of KeyOperator conforme pode ser visto na Ilustração 64. Após gerar o novo campo ele irá redirecionar a linha para o componente de união (Ilustração 54 (11)).

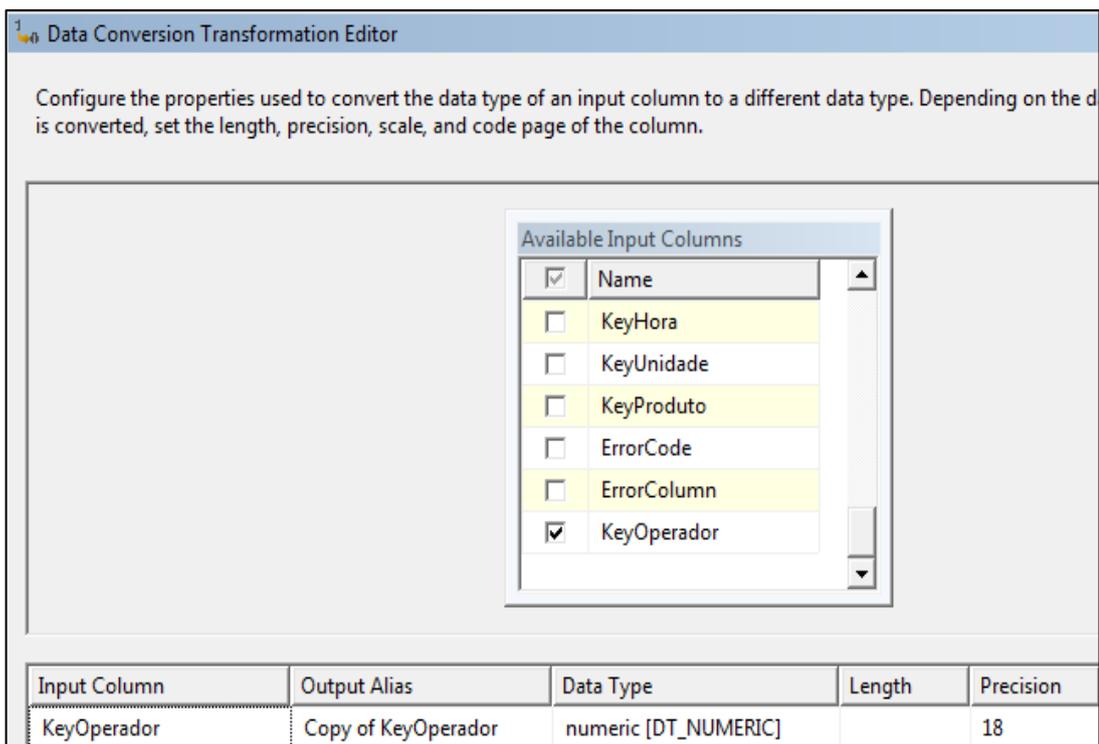


Ilustração 64 Conversão do campo KeyOperador de inteiro para numérico

O componente de união simplesmente une duas linhas vindas de fluxos diferentes de forma que sigam para o mesmo fluxo, ele funciona como um funil. Porém, para que funcione a união é necessário que as duas entradas de dados sejam capazes de gerar a mesma saída, para isso os campos correspondentes devem ser mapeados.

Conforme pode ser observado na Ilustração 65, as colunas são mapeadas de acordo com as correspondentes no processo. Quem define as colunas principais que a união deve conter é o componente conectado por primeiro na união, ou seja, se um componente possui as colunas A, B e C e ele é conectado na união esta identifica que estas colunas são obrigatórias, quando um novo componente é conectado na união ele deve ser capaz de gerar estas três colunas, caso contrário deverá ser utilizado um componente de coluna derivada ou algum outro que gere uma coluna correspondente. No caso da dimensão de operador, a primeira consulta gerou a coluna KeyOperador. Como para a saída de erro da consulta de operador foi gerada uma nova coluna com o nome de Copy of KeyOperador, esta nova coluna deve ser mapeada com a coluna KeyOperador da primeira entrada. Na imagem é possível verificar este mapeamento na última linha da grade exibida.

| Output Column Name | Union All Input 1 | Union All Input 2 |
|--------------------|-------------------|---------------------|
| KEYCAIXA | KEYCAIXA | KEYCAIXA |
| KEYCUPOM | KEYCUPOM | KEYCUPOM |
| COD_ATENDENTE | COD_ATENDENTE | COD_ATENDENTE |
| PERCENTUAL | PERCENTUAL | PERCENTUAL |
| VLR_TOTAL | VLR_TOTAL | VLR_TOTAL |
| VLR_ICMS | VLR_ICMS | VLR_ICMS |
| VLR_DESCONTO | VLR_DESCONTO | VLR_DESCONTO |
| QTD_LANCAMENTO | QTD_LANCAMENTO | QTD_LANCAMENTO |
| KeyData | KeyData | KeyData |
| KeyHora | KeyHora | KeyHora |
| KeyUnidade | KeyUnidade | KeyUnidade |
| KeyProduto | KeyProduto | KeyProduto |
| KeyOperador | KeyOperador | Copy of KeyOperador |

Ilustração 65 Mapeamento de colunas para união do fluxo de dados do operador

Um detalhe a ser verificado na Ilustração 65 é de que as colunas de ano, mês, dia, hora, minuto, segundo, código da unidade e código do produto não estão presentes na união. Como a consulta das chaves substitutas destes campos já foi realizada, para o fluxo de dados que será inserido na tabela de fatos só interessa as colunas de chaves substitutas. Logo na união, as colunas desnecessárias são removidas para evitar o fluxo de dados desnecessários gerando atraso no processo.

O processo de consulta de chave substituta na dimensão de vendedor é exatamente igual ao processo de consulta de chave substituta na dimensão de operador somente alterando o campo de operador para vendedor, por isso este processo não será ilustrado.

Logo após o fluxo consultar todas as chaves substitutas, ele entra no componente mais importante deste fluxo, o componente de agregação (Ilustração 54 (16)). Ocorre que o processo de ETL consulta cada item vendido em cada cupom fiscal, em alguns casos, um mesmo item pode ser vendido duas vezes e, portanto não poderia ser inserido novamente para a mesma transação de vendas. Caso isto ocorra, é necessário somar o valor de venda e a quantidade do item, bem como imposto e desconto, para um único item, evitando assim a duplicidade. Esta função pertence ao componente de agregação, onde são definidas as colunas de agrupamento, e as colunas que devem ser somadas. Na Ilustração 66 é exibida a configuração do componente de agregação, onde existe uma grade com três colunas, *Input Column*, *Output Alias* e *Operation*, onde a primeira possui as colunas recebidas do fluxo de

dados, a segunda possui o nome que será dado para estas colunas quando saírem do componente de agregação, e a terceira a operação a ser executada nestas colunas.

Ainda na Ilustração 66 é possível verificar que para as colunas chaves (que possuem o prefixo Key), a operação é de “*group by*”, ou seja, todo o dado que entrar no componente de agregação será agrupado por estas colunas. Para as colunas de valor de desconto, quantidade de lançamento e valor do imposto, a operação é de “*sum*”, ou seja, os dados agrupados serão somados nestas colunas.

| Input Column | Output Alias | Operation ▲ |
|----------------|----------------|-------------|
| KEYCUPOM | KEYCUPOM | Group by |
| KeyVendedor | KeyVendedor | Group by |
| KeyOperador | KeyOperador | Group by |
| KeyProduto | KeyProduto | Group by |
| KeyUnidade | KeyUnidade | Group by |
| KEYCAIXA | KEYCAIXA | Group by |
| PERCENTUAL | PERCENTUAL | Group by |
| KeyHora | KeyHora | Group by |
| KeyData | KeyData | Group by |
| VLR_DESCONTO | VLR_DESCONTO | Sum |
| QTD_LANCAMENTO | QTD_LANCAMENTO | Sum |
| VLR_TOTAL | VLR_TOTAL | Sum |
| VLR_ICMS | VLR_ICMS | Sum |

Ilustração 66 Configuração da agregação de vendas

Terminado o processamento do componente de agregação, finalmente os dados são direcionados para inserção na tabela de fatos (Ilustração 54 (17)). Para esta inserção é utilizado um componente de *Data Destination*, onde nele é configurada a conexão utilizada para se comunicar com o banco de dados do DW e a tabela onde os dados serão inseridos. Na Ilustração 67 é possível verificar esta configuração.

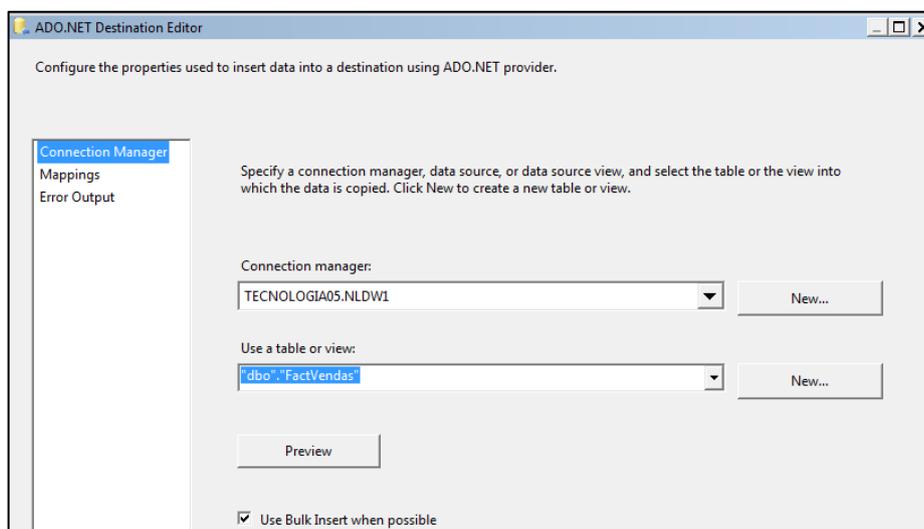


Ilustração 67 Configuração de conexão do Fato de Vendas

Após configurar a conexão com a tabela de fatos, é necessário mapear as colunas da origem que serão inseridas na tabela de fatos, arrastando a origem para a respectiva coluna de destino. Na tela de configuração exibida na Ilustração 68 do destino de dados existem duas colunas “*Input Column*” e “*Destination Column*”, na primeira estão as colunas obtidas do componente de agregação, na segunda as colunas da tabela de fatos mapeadas com as colunas do componente de agregação.

| Input Column | Destination Column |
|----------------|--------------------|
| KeyUnidade | KeyUnidade |
| KEYCAIXA | KeyCaixa |
| KeyOperador | KeyOperador |
| KeyData | KeyData |
| KeyHora | KeyHora |
| KeyProduto | KeyProduto |
| KEYCUPOM | KeyCupom |
| KeyVendedor | KeyVendedor |
| VLR_TOTAL | VlrTotal |
| VLR_ICMS | VlrImposto |
| PERCENTUAL | PercentualImposto |
| VLR_DESCONTO | VlrDesconto |
| QTD_LANCAMENTO | QtdVendida |

Ilustração 68 Mapeamento de colunas para tabela de fato de Vendas

5.1.3 ETL de Vendas por Forma de Pagamento

Nesta seção será explicado o processo de ETL para o fato de vendas por forma de pagamento. Como este processo se assemelha ao processo de ETL para o fato de vendas, exceto pela consulta realizada no banco de dados, troca da dimensão de produto pela dimensão de forma de pagamento, posicionamento da agregação no fluxo de dados e tabela de fatos diferente, serão explicados somente estes itens, o restante do processo pode ser interpretado com base nas informações descritas na seção anterior.

Na Ilustração 69 é exibido o processo macro de ETL de vendas por forma de pagamento. Na imagem é possível verificar que o componente de agregação está logo após o componente de colunas derivadas, diferente do processo de ETL de vendas onde ele se encontra antes da inserção dos dados.

Também é possível verificar na Ilustração 69 que após a consulta de chave substituta da dimensão de unidade, segue a consulta de chave substituta na dimensão de formas de pagamento, diferente do processo anterior onde após a dimensão de unidade se encontra a consulta de chave substituta na dimensão de produto.

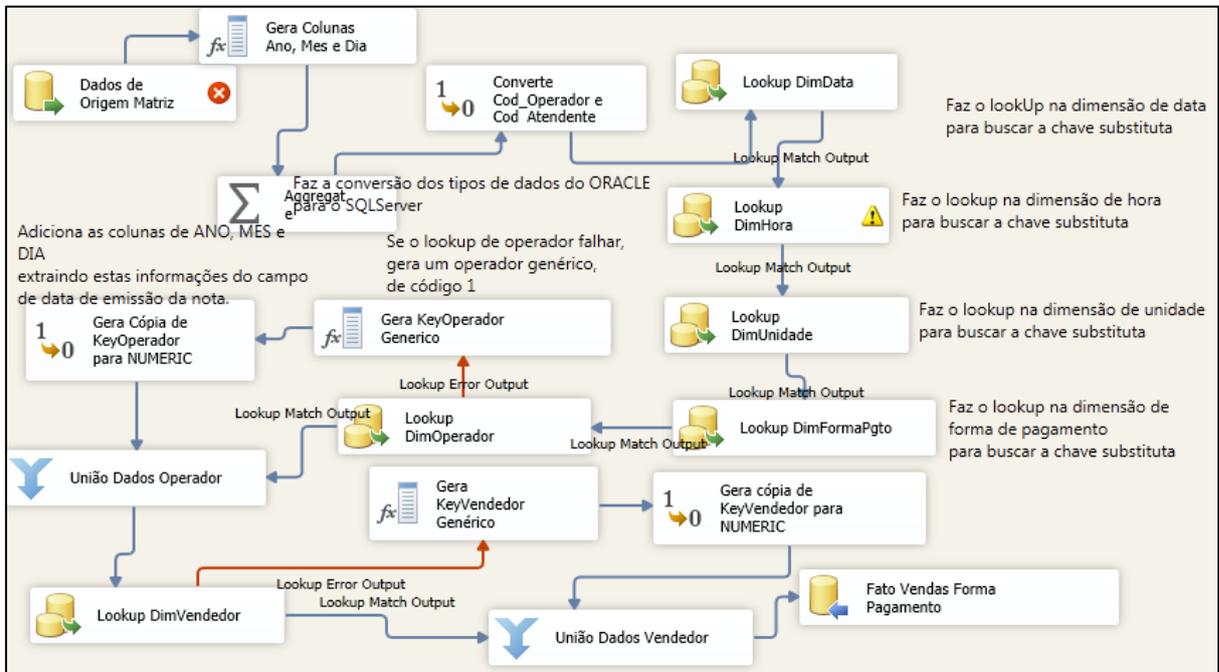


Ilustração 69 Processo macro de ETL de vendas por forma de pagamento

Na Ilustração 70 é exibida a configuração da agregação, a grande diferença entre esta agregação e a agregação do processo anterior é o número de colunas. Como no processo anterior a agregação foi incluída no final do processo de ETL, os agrupamentos são realizados pelos campos chave da tabela de fatos. Neste processo como ainda não foi realizada a consulta de chaves substitutas, é necessário realizar a agregação com todos os campos disponíveis e ainda não transformados, logo o número de colunas se torna maior, os tipos de dados fora de padrão, o que colabora para uma queda no desempenho deste componente.

| Input Column | Output Alias | Operation |
|---------------|---------------|-----------|
| MINUTO | MINUTO | Group by |
| HORA | HORA | Group by |
| DIA | DIA | Group by |
| MES | MES | Group by |
| ANO | ANO | Group by |
| SEGUNDO | SEGUNDO | Group by |
| NUM_ECF | NUM_ECF | Group by |
| COD_UNIDADE | COD_UNIDADE | Group by |
| COD_ATENDENTE | COD_ATENDENTE | Group by |
| COD_DOCUMENTO | COD_DOCUMENTO | Group by |
| NUM_NOTA | NUM_NOTA | Group by |
| COD_OPERADOR | COD_OPERADOR | Group by |
| VLR_ENTRADA | VLR_ENTRADA | Sum |
| NRO_PARCELA | NRO_PARCELA | Sum |

Ilustração 70 Configuração de agregação para o fato de vendas por forma de pagamento

Na Ilustração 71 é exibida a configuração da consulta realizada para buscar a chave substituta na dimensão de formas de pagamento.

| Lookup Column | Lookup Operation | Output Alias |
|---------------|---------------------|--------------|
| KeyFormaPgto | <add as new column> | KeyFormaPgto |

Ilustração 71 Configuração da consulta de chave substituta na dimensão de formas de pagamento

Na Ilustração 72 é exibido o mapeamento de colunas para inserção de dados na dimensão de vendas por forma de pagamento.

| Input Column | Destination Column |
|--------------|--------------------|
| KeyUnidade | KeyUnidade |
| NUM_ECF | KeyCaixa |
| KeyOperador | KeyOperador |
| KeyData | KeyData |
| KeyHora | KeyHora |
| KeyFormaPgto | KeyFormaPgto |
| NUM_NOTA | KeyCupom |
| KeyVendedor | KeyVendedor |
| VLR_ENTRADA | VlrTotal |
| NRO_PARCELA | nroParcelas |

Ilustração 72 Mapeamento de colunas para inserção dos dados no fato de vendas por forma de pagamento

5.2 Carga dos Dados no DW

Nesta seção é descrito como os dados foram carregados para o DW, o volume de dados carregados e o espaço ocupado por estes dados. Antes de explicar o processo de carga dos dados, é importante salientar as configurações do computador que realizou este processo exibidas no Quadro 4.

Quadro 4 Configurações do servidor de ETL

| | |
|----------------------------|---|
| Processador | Intel Core Quad CPU 2 núcleos de 2,66 GHz |
| Memória RAM | 4 GB |
| Disco Rígido | 500 GB |
| Sistema Operacional | Windows 7 Professional 64 Bits |

Mesmo não sendo objetivo deste trabalho, foi identificado como um bom ponto para iniciar um processo de ETL é através da utilização do gatilho do expurgo, os dados podem ser copiados para o *data warehouse* antes de serem apagados de cada unidade.

O processo de ETL iniciou pela execução da carga de dimensões obtendo os resultados de execução exibidos no Quadro 5.

Quadro 5 Resultado da ETL de dimensões

| Dimensão | Número de Linhas Carregadas | Tamanho da Dimensão em MB | Tempo de Carga |
|-----------------|------------------------------------|----------------------------------|-----------------------|
| DimData | 37984 | 4,22 | 1 minuto |
| DimFormaPgto | 14 | 0,008 | Menos de 1 minuto |
| DimHora | 86400 | 3,66 | 1 minuto |
| DimOperador | 29774 | 1,5 | 1 minuto |
| DimProduto | 769096 | 52,50 | 4 minutos |
| DimUnidade | 418 | 0,04 | Menos de 1 minuto |
| DimVendedor | 1 | 0,008 | Menos de 1 minuto |

Após a execução da ETL de dimensões foi executada a ETL para o fato de vendas. Na Ilustração 73 é exibida uma execução de ETL para o fato de vendas, onde foram consultadas as vendas dos anos de 2010 e 2011 para seis unidades de um cliente.

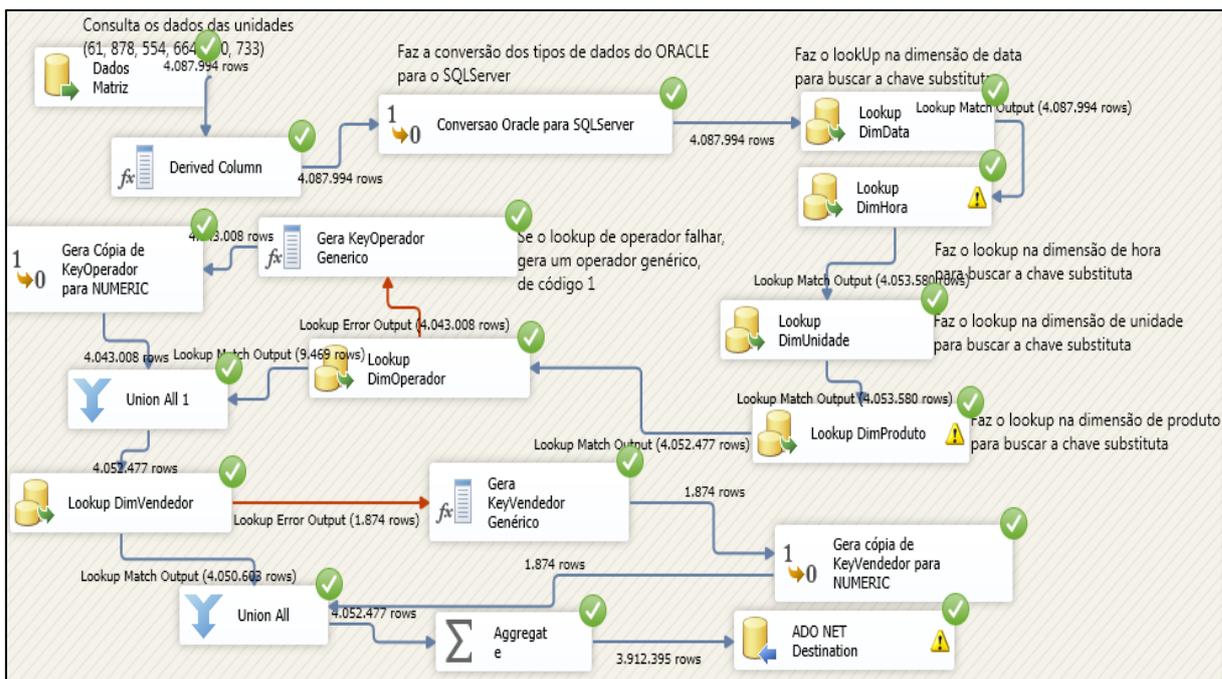


Ilustração 73 Execução da ETL para o fto de vendas

No caso da ETL exibida na Ilustração 73 para a consulta dos mais de quatro milhões de registros, consulta de chaves substitutas, agregação e carga na tabela de fatos, o processo demorou exatamente três horas e vinte e cinco minutos. Este tempo é exibido na Ilustração 74.

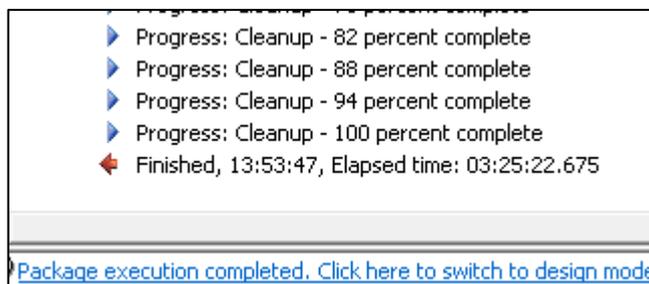


Ilustração 74 Tempo de execução da ETL do fato de vendas

Após esta execução de ETL, o processo foi executado novamente para o ano de 2009 e 2012 incluindo mais uma unidade do cliente. No Quadro 6 é exibido o resultado do processo para a ETL do fato de vendas. Neste fato constam os resultados de sete unidades para os anos de 2009, 2010, 2011 e 2012.

Quadro 6 Resultado da ETL do fato de vendas

| Tabela | Número de Linhas Carregadas | Tamanho da Dimensão em MB | Tempo de Carga |
|------------|-----------------------------|---------------------------|-------------------------|
| FactVendas | 6.608.710 | 1.134,875 | 6 horas aproximadamente |

5.3 Criação do Cubo de Dados

Nesta seção é descrito o processo de criação do cubo de dados na ferramenta BIDS. É apresentada a configuração dos grupos de medidas e dimensões utilizados, bem como, os demais detalhes da implementação do Cubo de dados.

Para criar o Cubo de dados foi criado um novo projeto do *analysis services* na ferramenta BIDS com o nome de Cubo de Vendas TCC. Na Ilustração 76 é possível verificar a criação do novo projeto.

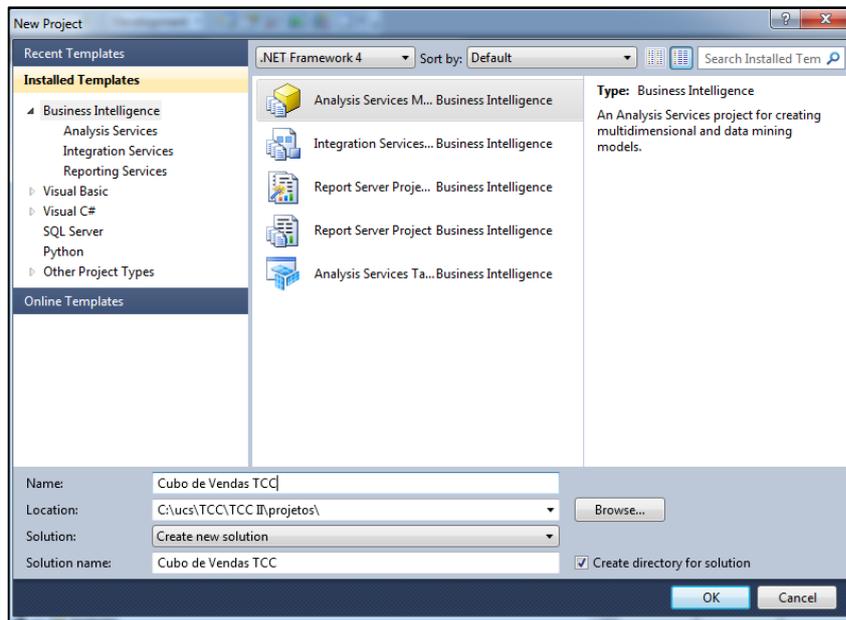


Ilustração 76 Criação do projeto de Cubo de Vendas no BIDS

Neste projeto foi criado um *data source* para buscar os dados no *data warehouse*, e em seguida um *data source view* para carregar os dados deste *data source*. Na Ilustração 77 são exibidos os *data sources* para o SQL Server.



Ilustração 77 Data Sources do projeto

Como não houve a necessidade de alterar os relacionamentos entre as tabelas para uma melhor visualização por parte do servidor OLAP, a estrutura de tabelas permanece a mesma na visualização do *data source view*. Na Ilustração 78 é exibida a estrutura das tabelas do DW no *data source view*. Na imagem é possível verificar as tabelas fato ao centro e as dimensões ao redor dos fatos, caracterizando o modelo multi-estrela.

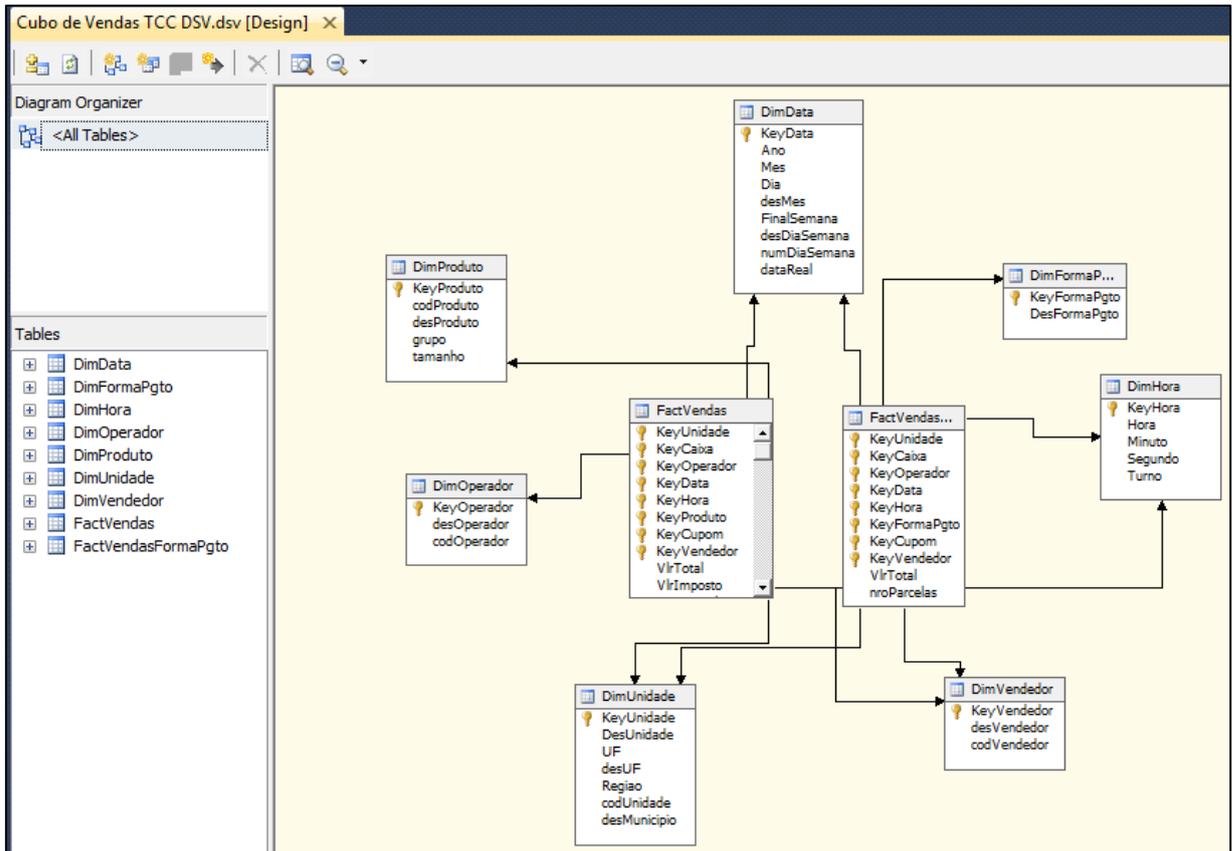


Ilustração 78 Estrutura Data Source View

Após a criação do *data source*, foi criado o cubo de dados propriamente dito. Na Ilustração 79 é exibido à direita, o arquivo do cubo de dados juntamente com os *data sources* e as dimensões. À esquerda, são exibidos os grupos de medidas do cubo juntamente com as dimensões formatadas. Na imagem fica claro que existem dois grupos de medidas, um para vendas por produto, e outro para vendas por forma de pagamento. As dimensões aparecem na esquerda, formatadas de forma diferente que do lado direito, isto porque foram adicionados nomes amigáveis para estas dimensões e também seus atributos foram alterados nas propriedades de máscara, e nome.

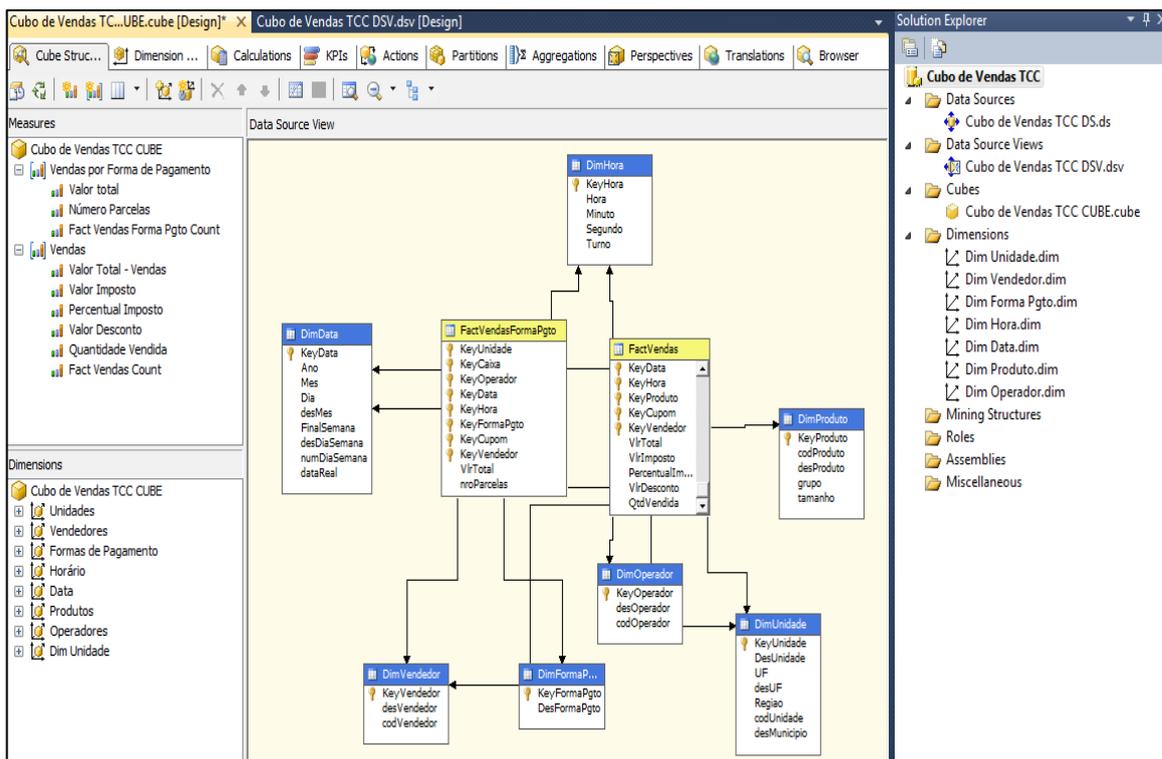


Ilustração 79 Tela Inicial do Cubo de Vendas

| Name | Measure Group | Data Type |
|-------------------------------|--------------------|-----------|
| Valor total - Forma Pagamento | Vendas por Form... | Double |
| Número Parcelas | Vendas por Form... | Integer |
| Fact Vendas Forma Pgto Count | Vendas por Form... | Integer |
| Valor Total - Vendas | Vendas | Double |
| Percentual Imposto | Vendas | Double |
| Valor Imposto | Vendas | Double |
| Valor Desconto | Vendas | Double |

| Property | Value |
|-------------------|-----------|
| AggregateFunction | Sum |
| DataType | Inherited |
| DisplayFolder | |
| MeasureExpression | |
| Visible | True |
| Description | |
| FormatString | Currency |
| ID | |

Ilustração 80 Máscara de valor monetário para medidas

Para ambos os grupos de medidas, as medidas que representam valores monetários foram configuradas com a máscara de *currency*, através da alteração da propriedade “*formatString*”. Na Ilustração 80 é exibida esta configuração.

Após a edição das medidas, foi realizada a edição das dimensões, através da criação de hierarquias e definição dos atributos disponíveis para cada dimensão.

Para a dimensão de unidade foram utilizados todos os atributos disponíveis, conforme é exibido na Ilustração 81. À direita, estão os atributos disponíveis no *data source view*. No centro as hierarquias, e à esquerda os atributos definidos para a dimensão. Estes são originados do DSV, porém com nomenclatura diferente e algumas propriedades alteradas. É possível verificar na imagem duas hierarquias para possibilitar as operações de aumento e diminuição da granularidade dos dados.

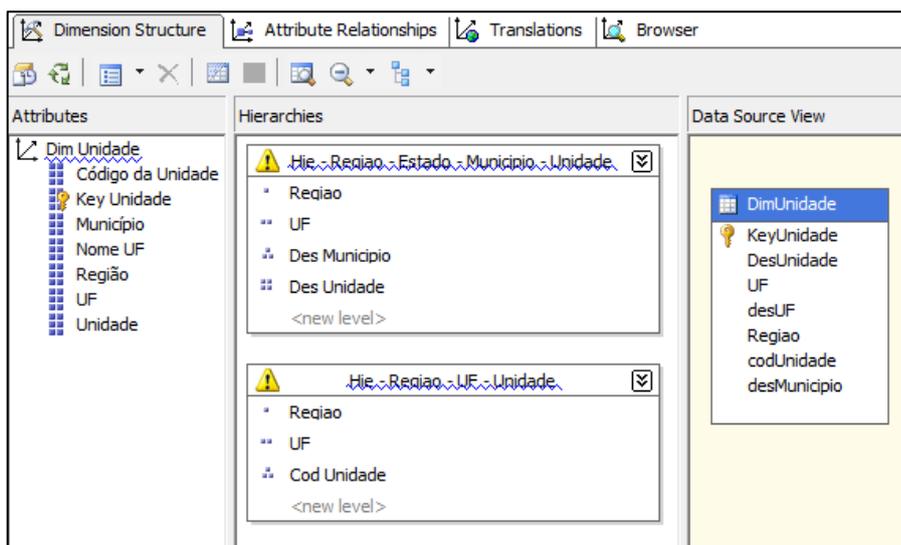


Ilustração 81 Edição da dimensão de Unidade

As hierarquias definidas para a dimensão de unidade permitirão a navegação nos dados de duas maneiras, a primeira permitirá a navegação por Região, Estado, Município e unidade. Na Ilustração 82 são exibidos os dados da dimensão no formato hierárquico definido.

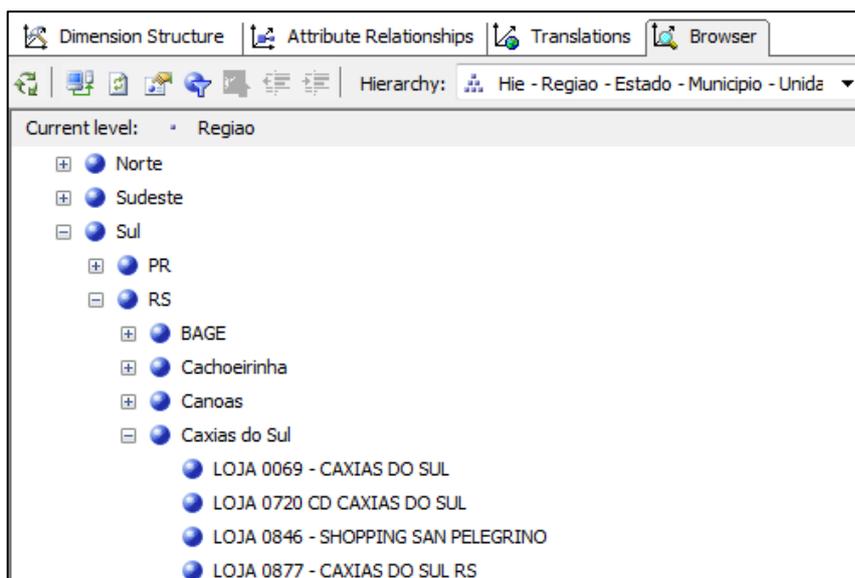


Ilustração 82 Hierarquia da dimensão de Unidade com granularidade por município

A segunda hierarquia criada para a dimensão de unidade, permite a visualização dos dados por Região, Estado e unidade. Na Ilustração 83 são exibidos os dados da dimensão dispostos no formato desta hierarquia.

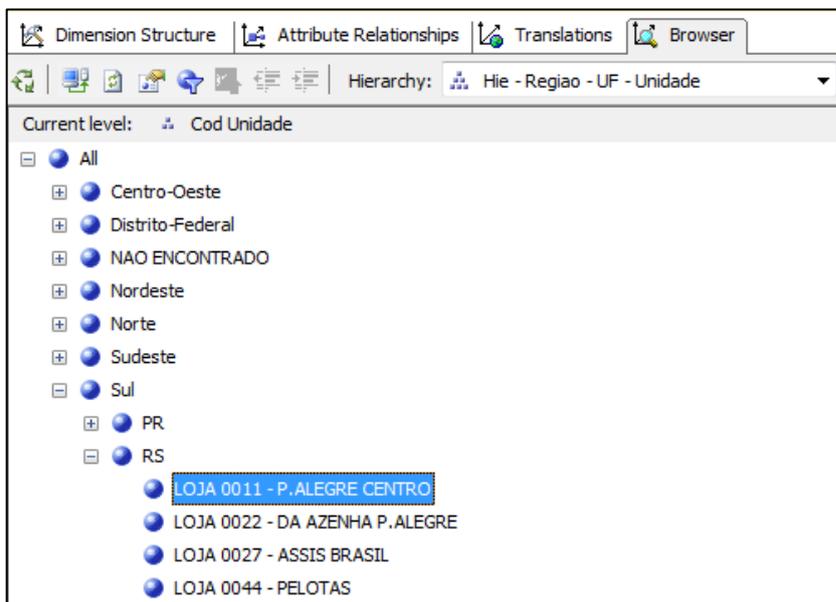


Ilustração 83 Hierarquia de Unidade

Para a dimensão de data foi definida uma hierarquia onde o usuário poderá navegar entre ano, mês, dia da semana e dia do mês. Na Ilustração 84 são exibidos os dados dispostos na hierarquia criada.

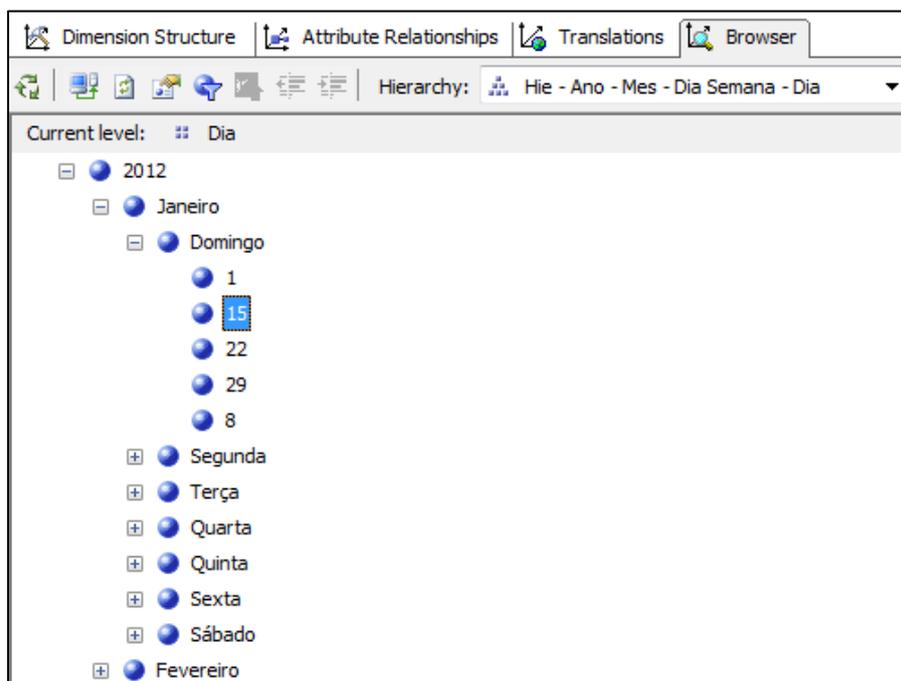


Ilustração 84 Hierarquia de data

Para a dimensão de produto foi definida uma hierarquia que possibilitará a navegação entre grupos de produtos vendidos e tamanho. Na Ilustração 85 são exibidos os dados dispostos da hierarquia.

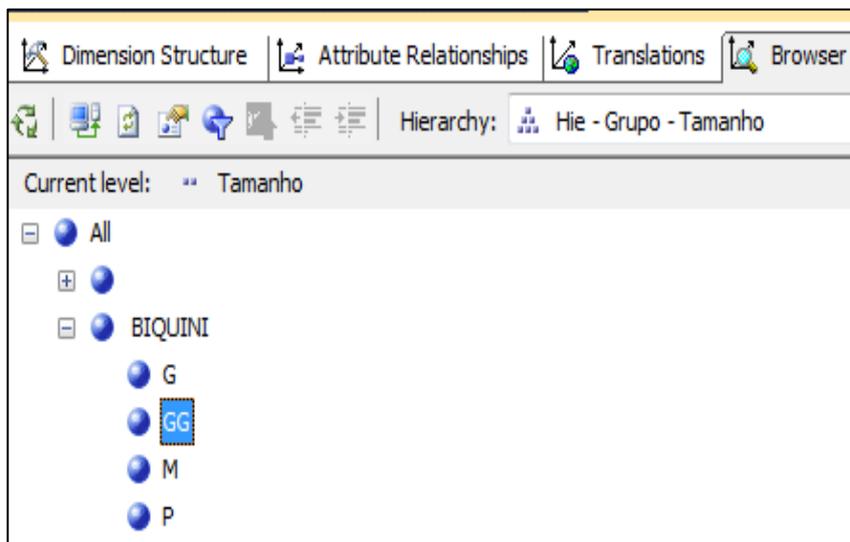


Ilustração 85 Hierarquia de produto

Após as devidas configurações de dimensão realizadas, foram configuradas as partições do cubo de dados. Foi criada apenas uma partição para cada grupo de medidas utilizando o modo de armazenamento MOLAP como pode ser visto na Ilustração 86.

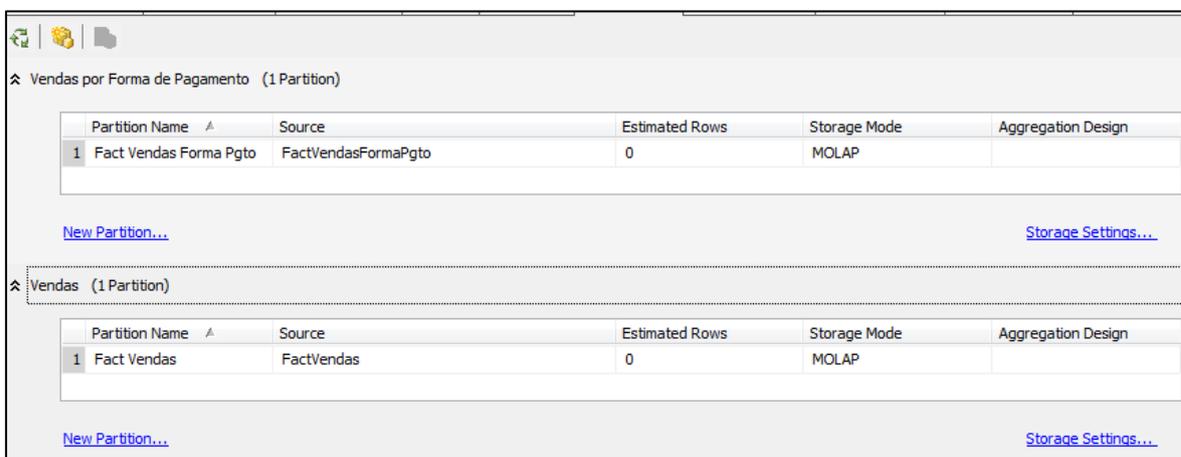


Ilustração 86 Partições do cubo

5.4 Considerações Finais

Durante o desenvolvimento da solução de BI não foram encontradas grandes dificuldades no aprendizado e na utilização das ferramentas devido a facilidade de utilização e grande quantidade de material disponibilizado pelo fornecedor.

Mesmo não havendo grandes dificuldades no aprendizado, inicialmente houve uma grande dificuldade de extração dos dados do cliente. Num primeiro momento, optou-se por extrair os dados direto de cada unidade para o servidor da NL. Para realizar este processo era necessário realizar uma conexão com a matriz do cliente, e em seguida redirecionar uma porta do servidor da matriz para criar uma nova conexão com a filial. Este processo além de lento era muito instável e não permitia a extração de grandes quantias de dados ao mesmo tempo para realizar uma operação de carga inicial.

Para solucionar o problema anterior optou-se por consultar os dados direto do banco de dados do servidor da matriz. As taxas de transferência e estabilidade da conexão se mostraram melhores. Porém, por um detalhe na modelagem do processo de extração no SSIS ainda assim não foi possível a extração de grandes quantidades de dados.

O detalhe que falhou no processo de ETL foi o de inicialmente não ser utilizado o componente de agregação do SSIS. Ocorre que como as transações de vendas podem conter o mesmo produto mais de uma vez, era solicitado ao servidor do banco de dados da matriz que somasse os valores e quantidades de cada produto para unificar os dados do produto antes de encaminhar ao servidor que executava a ETL.

O problema era que o servidor da matriz do cliente, ao invés de somente consultar e encaminhar os dados, necessitava consultar, ordenar, agrupar e somar os dados, gerando uma quantidade de processamento maior no lado do cliente, o que não é o objetivo em um processo de ETL. Por consequência disso o banco de dados do cliente só conseguia encaminhar pacotes com no máximo cinquenta mil registros e demorava cerca de trinta minutos para conseguir somar e encaminhar estes dados.

Após a inclusão do componente de agregação no processo de ETL e a remoção da operação de soma da instrução SQL de consulta no banco de dados da matriz, o desempenho do lado do cliente foi muito maior. O servidor da matriz do cliente foi capaz de enviar pacotes com nove mil registros a cada quarenta segundos ao servidor da ETL. Este por sua vez foi mantendo estes dados em memória até que recebesse todos os dados provindos do servidor do cliente para posteriormente realizar o processo de agregação.

Inicialmente o componente de agregação foi posicionado logo após a consulta dos dados no cliente e antes de realizar as operações de consulta de chaves substitutas. Logo, por realizar agregações com dados redundantes e de tipos diferentes, este processo se mostrou um pouco demorado. Após alguns testes no processo de ETL, o componente de agregação foi movido para o final do processo de ETL o que ocasionou um grande ganho de desempenho no

processo de ETL devido ao processo de agregação ser realizado somente com as chaves substitutas.

Para o processo de modelagem do cubo de dados no SSAS não houveram grandes dificuldades. Conforme o volume do cubo de dados foi crescendo o desempenho do processamento do cubo naturalmente se mostrou mais lento.

O que foi possível perceber é que o SSAS consome bastante memória RAM devido a realização de processamento em tempo real, logo é necessário disponibilizar um servidor somente para este serviço. O interessante é que ele não necessita de um disco rígido muito rápido visto que não realiza processamento de escrita, seu processamento é realizado praticamente todo na memória RAM.

O SSAS também se mostrou eficiente para testes de gerenciamento do cubo de dados, com o mecanismo de partições. Pela experiência obtida na utilização do SSIS e SSAS foi possível concluir que profissionais que possuam conhecimentos de DW e BI, serão capazes de compreender estas ferramentas de forma rápida e portanto conseguir produzir nesta mesma velocidade um ambiente de BI.

6 RESULTADOS OBTIDOS

Neste capítulo são apresentados os resultados obtidos a partir da criação do cubo de dados no SSAS. Os resultados apresentados neste capítulo visam atender os objetivos da modelagem do BI. A intenção é de que os gráficos e tabelas aqui apresentados sejam utilizados como modelos para que os usuários criem novas consultas. Os resultados poderão ser utilizados pelos usuários, porém deverão servir como base para futuras análises. A seguir é apresentada uma seção para cada objetivo apresentado na modelagem do BI (seção 4.2).

6.1 Horários de Picos de Vendas

Com a análise de horários de pico de vendas é possível determinar por cada dia da semana qual horário as lojas vendem mais, e então gerenciar melhor o atendimento destas lojas. Por exemplo, analisando a Ilustração 87 é possível determinar que as lojas não necessitam de muitos funcionários na Região Sul pela manhã, também é possível verificar que o volume das vendas cresce entre quatorze e dezessete horas.

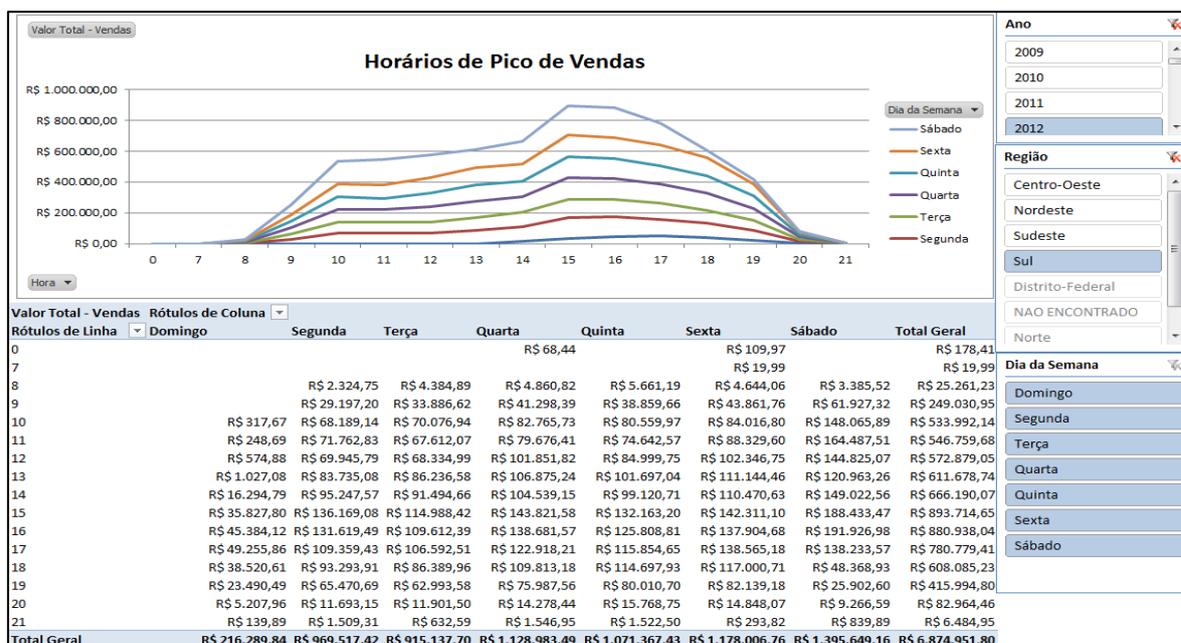


Ilustração 87 Análise de horários de pico de vendas

Utilizando os componentes de segmentação à direita da imagem pode-se comparar os dados com o decorrer dos anos, verificar os valores para outras regiões, bem como filtrar o dia da semana.

Os valores apresentados no gráfico são comprovados na tabela dinâmica abaixo do mesmo. Verifica-se na tabela que os dias da semana estão apresentados como colunas da tabela, logo, são séries do gráfico. As horas do dia, são rótulos de linha, logo, estão no eixo x do gráfico.

6.2 Vendas por Forma de Pagamento

Nesta seção é apresentada a análise de vendas por forma de pagamento. Esta análise permite a identificação das formas de pagamento mais utilizada pelos clientes. Através desta informação é possível direcionar melhor os caixas de uma loja de acordo com o meio de pagamento mais utilizado bem como obter novos equipamentos que atendam as demandas. Estas informações podem ser vistas na Ilustração 88.

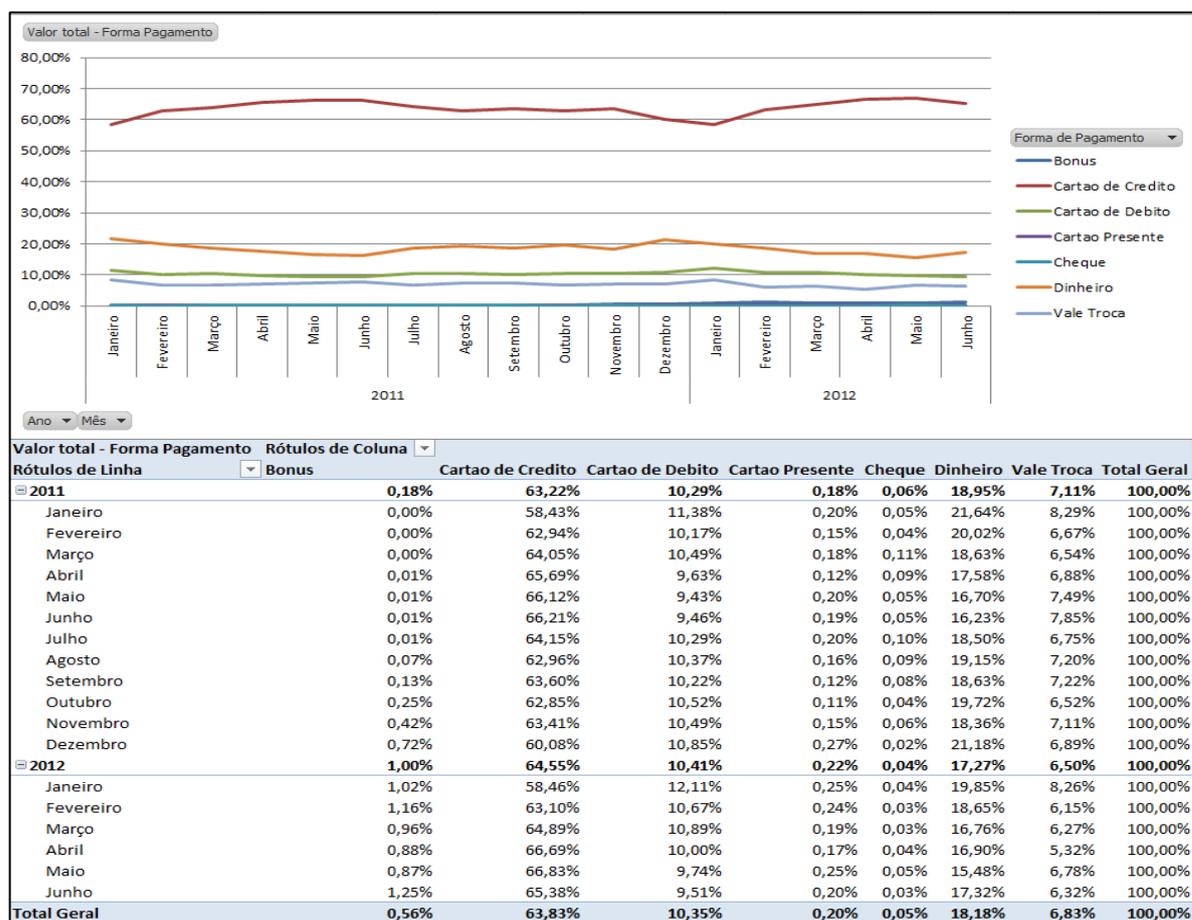


Ilustração 88 Análise de Vendas por forma de pagamento

Também é possível definir promoções que melhor atendam a necessidade dos clientes e os objetivos da loja de acordo com as formas de pagamento utilizadas. Na Ilustração 88 é possível verificar um gráfico e uma tabela de dados de análise de vendas por forma de pagamento. Ainda na imagem é possível verificar que forma de pagamento mais utilizada é o cartão de crédito seguido pelo dinheiro e cartão de débito.

Através do gráfico também é possível perceber que a utilização de vales troca consiste em dez por cento dos pagamentos, portanto pode-se deduzir que ocorre um número relativamente baixo de trocas das mercadorias. Visto o cliente recebe um vale troca no momento que devolve uma mercadoria para empresa esta gera um comprovante informando o valor que o cliente tem disponível para gastar em outras mercadorias no valor da mercadoria devolvida.

6.3 Ciclo de Vida de Produtos

Nesta seção é apresentada a análise de ciclo de vida de produtos. Esta análise visa mostrar ao gestor como estão às vendas de determinado produto em um determinado período de tempo.

Na Ilustração 89 é exibido o gráfico de ciclo de vida de produtos para o ano de 2012. O gráfico demonstra os cinco produtos mais vendidos de março à junho deste ano. No gráfico desta imagem é possível verificar informações interessantes para um gestor como, por exemplo, após o mês de fevereiro quando o verão começa a chegar ao fim, às vendas de calças começam a subir, enquanto as vendas de bermudas e shorts começam a cair.

Outra informação interessante de ser analisada tanto no gráfico quanto na tabela que complementa o gráfico na Ilustração 89 é de que no mês de maio as vendas de todos os itens tem uma leve alta. No mês de junho em contra partida as vendas tendem a cair.

Uma suposição a ser feita sobre estas informações é de que como os dados foram extraídos de um cliente especializado na comercialização de roupas e acessórios femininos, a alta de vendas no mês de maio se deve à comemoração do dia das mães, e a queda no mês de junho vem em consequência das aquisições de presentes para esta comemoração, logo os clientes tendem a economizar ou até mesmo pagar as aquisições realizadas e acabam não realizando novas compras.

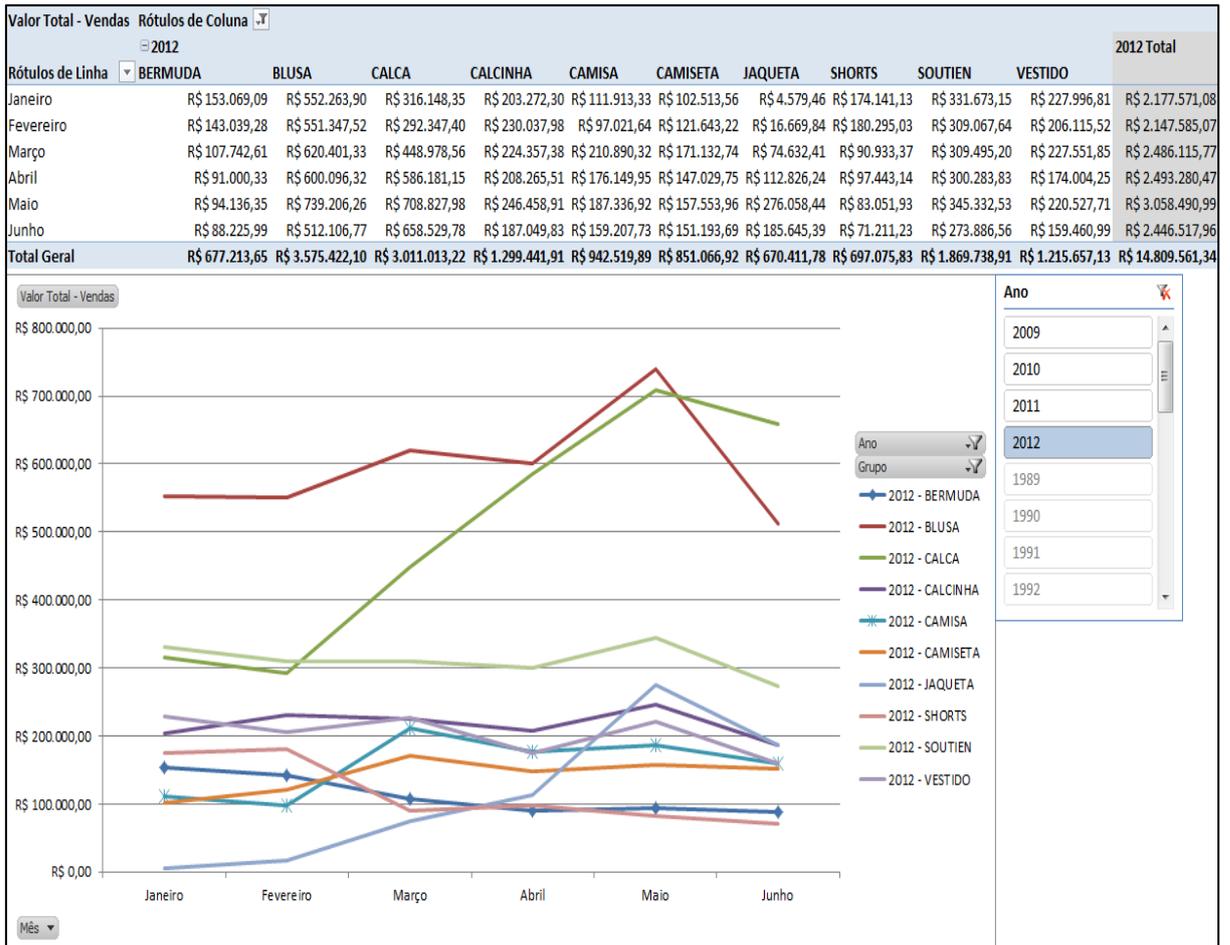


Ilustração 89 Análise de ciclo de vida de produto

6.4 Vendas por Unidade da Empresa

A análise de vendas por unidade da empresa permite verificar o volume de vendas por unidade possibilitando que se identifique a necessidade de criação de promoções ou campanhas publicitárias na região onde esta unidade atua.

Na Ilustração 90 são exibidos dois gráficos, no gráfico de colunas é exibido o volume de vendas por ano em cada unidade da empresa contida no DW. No gráfico de colunas é exibida a proporção que cada unidade representa sobre o volume total de vendas da empresa. Abaixo dos gráficos é exibida a tabela que exibe os dados que geraram estas informações.

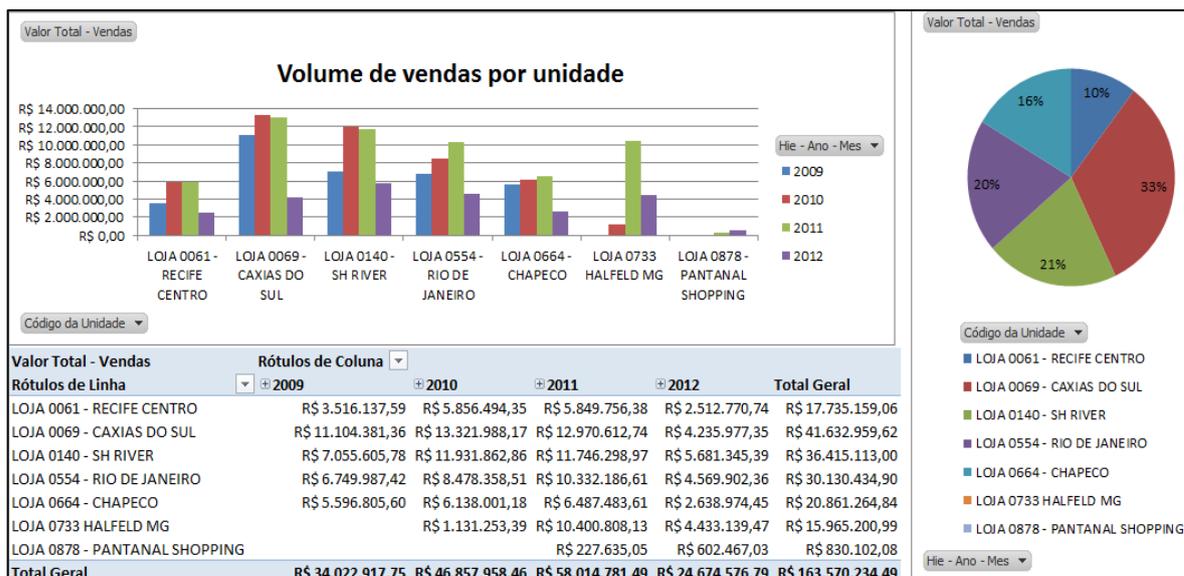


Ilustração 90 Análise de vendas por unidade

Ainda na Ilustração 90 é exibido à esquerda dos rótulos de coluna de ano o sinal de “+”, este sinal identifica que foi utilizada uma hierarquia para mostrar os dados da tabela, logo se o usuário clicar neste símbolo a tabela irá realizar a operação de “drill-down” do ano para o mês. Este comportamento é exibido na Ilustração 91.

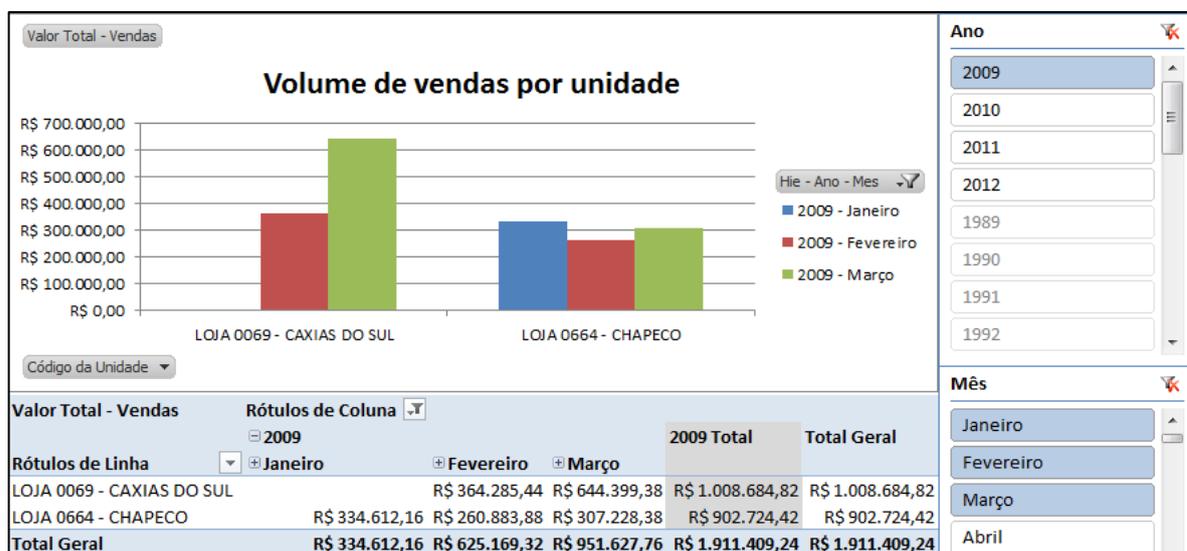


Ilustração 91 Drill-Down na análise de volume de vendas

6.5 Vendas por Região

Nesta seção é apresentado o gráfico e tabela de análise de dados de vendas por ano e região. Na Ilustração 92 é exibido o gráfico e a tabela de dados que complementa este gráfico de vendas por região.

Aparentemente este gráfico pode não representar muitas informações, porém um diretor pode decidir abrir mais unidades da empresa em uma determinada região com base no volume de vendas gerado pelas unidades atuais desta região. O processo contrário também pode acontecer, onde pode ser optado por fechar uma ou outra unidade de uma região devido ao baixo volume de vendas.

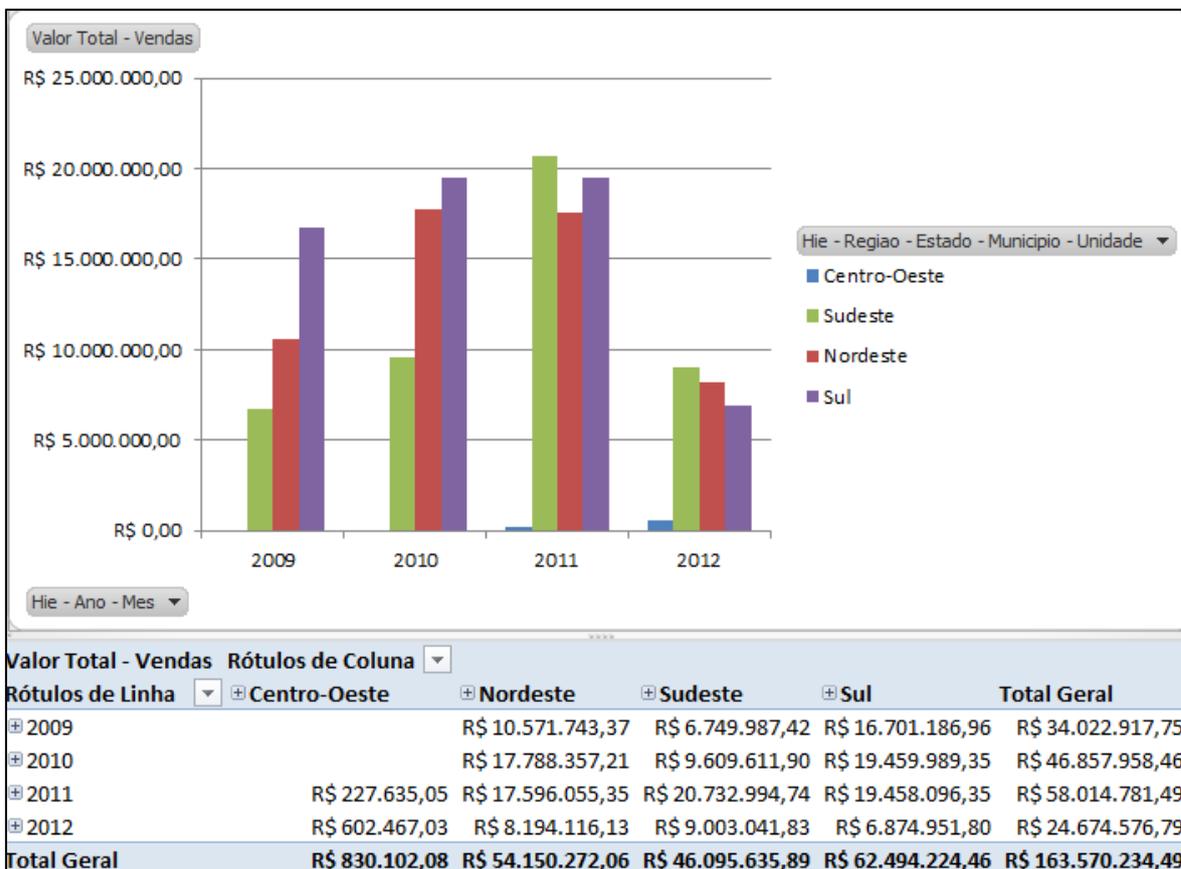


Ilustração 92 Análise de vendas por Região

6.6 Considerações Finais

Os gráficos e tabelas apresentados mostraram-se úteis para as principais análises realizadas por um gestor de vendas. Com estes modelos inicialmente é possível tomar diversas decisões do ponto de vista estratégico para uma empresa com base nestes dados.

As informações disponibilizadas não tem por objetivo serem utilizadas de forma única, a intenção é de que estes modelos tanto de gráficos quanto de tabelas incentivem o usuário a criar novas consultas ou simulações com base no que foi apresentado.

Os objetivos da modelagem do BI foram facilmente atingidos com os gráficos e tabelas apresentados. Além dos objetivos serem atingidos a ferramenta de tabelas dinâmicas

do Excel se mostrou muito eficiente e rápida para realizar as análises. Além destes fatores a criação dos gráficos para os resultados se mostrou uma tarefa divertida e que em poucos minutos trabalhando na mesma foi possível criar várias suposições com base nos dados e imaginar diversas estratégias para o aumento de vendas. Este tipo de comportamento não é facilmente obtido através da utilização de diversos relatórios gerados por um sistema de informação voltado a operação da empresa.

Com o complemento destes gráficos o produto de gestão de vendas da empresa NL acabou mostrando um grande ganho por dois motivos, a necessidade de criação de diversos relatórios e consultas para chegar aos objetivos apresentados se mostrou nula, pois os gráficos e tabelas apresentados são de fácil manipulação e tamanha eficiência que se mostram mais interessantes para um gestor do que a emissão de relatórios pelo sistema.

Através destes resultados fica evidente que a implantação de um sistema de BI sobre o sistema NL Gestão é de extrema importância tanto para os clientes quanto para a estratégia de negócio do fornecedor do sistema, eliminando custos com desenvolvimento e aumentando a possibilidade de venda de serviços de análise de dados.

7 CONCLUSÃO

Os sistemas de informação atuais além de não proverem em sua maioria os dados necessários para auxiliar a tomada de decisão, acabam tornando o usuário desmotivado pela complexidade e grande quantidade de relatórios necessários para obter uma simples informação. Tarefa que no BI acaba se tornando rápida, intuitiva e divertida, o que acaba motivando a utilização do usuário desta ferramenta.

Para diversas pessoas das quais foi mostrado o resultado deste trabalho, a interatividade proporcionada pelas tabelas dinâmicas e gráficos no Excel foi realmente satisfatória. Porém houve uma certa dificuldade para o entendimento de pessoas mais leigas no conceito de sistemas para o entendimento de toda a arquitetura de BI. Esta falta de entendimento acaba dificultando a convencer um usuário ou gestor sobre o custo de implantação deste conceito em sua organização, mesmo demonstrando os benefícios e possíveis ganhos em longo prazo a conversa acaba se tornando complicada. Creio que esta conclusão foi adquirida pelo fato de não conversar sobre o assunto com usuários de grandes organizações. O que leva a acreditar que esta seja uma das dificuldades em visualizar casos de sucesso de BI em empresas de pequeno e médio porte.

Quanto às demonstrações de resultados obtidos para a área de vendas, as pessoas se mostraram extremamente satisfeitas com os resultados que o cubo de dados proporciona. Mesmo não sendo um cubo complexo permite a análise de grandes volumes de dados praticamente de forma instantânea. Conforme o cubo foi visualizado já foram despertadas novas ideias de dimensões a serem incluídas no mesmo, bem como medidas novas a serem analisadas o que irá gerar um novo incremento interessante para este trabalho.

Apesar do fato de o sistema na NL ser desenvolvido e armazenado em tecnologias da empresa Oracle, a utilização de ferramentas da Microsoft para o BI não foi mal vista pelos usuários. Usuários mais técnicos concordaram que o potencial das ferramentas utilizadas é muito bom e que com elas é possível criar com facilidade uma solução de BI robusta e confiável.

De uma maneira geral o trabalho apresentado atendeu seus objetivos de forma satisfatória. Foi possível demonstrar aos gestores da empresa como funciona o conceito de BI, quais ferramentas são utilizadas, e foi comprovada a necessidade de implantação deste

conceito no sistema NL Gestão. Através da demonstração dos resultados do trabalho, ficou comprovado que uma ferramenta de BI irá diminuir significativamente a necessidade de customizações em relatórios do sistema e telas de consulta para atender as necessidades dos usuários, com a interatividade que a ferramenta de BI proporciona, conseqüentemente irá aumentar o nível de satisfação dos clientes com o produto.

7.1 Trabalhos Futuros

Este trabalho mostrou o caminho inicial para implantação de BI no sistema NL Gestão. Poderá ser utilizado como um guia tanto para a criação de cubos de dados, quanto para utilização das ferramentas de BI. Contudo, o trabalho abre um leque de oportunidades e necessidades conforme descrito a seguir:

1. Necessidade de uma ferramenta Web para visualização do cubo de dados. Uma boa opção é a complementação deste trabalho com a ferramenta Sharepoint da Microsoft utilizando os recursos da ferramenta Power View do SQL Server na versão 2012;
2. Criação de estruturas de mineração de dados através da inclusão de dados de consumidores no DW e possível aplicação das já existentes ferramentas de mineração do *Analysis Services*;
3. Definição de um padrão de desenvolvimento de tabelas fato e dimensões, ou seja, definir um barramento de dados para o DW;
4. Criação de métricas que sejam capazes de definir o retorno sobre o investimento em uma ferramenta de BI para que a comercialização de um produto neste sentido seja melhor trabalhada e justificada aos usuários;
5. Definição de um processo de desenvolvimento e papéis para equipes de desenvolvimento do BI.

8 BIBLIOGRAFIA

BALLARD, C.; ABDEL-HAMID, A.; FRANKUS, R.; HASEGAWA, F.; LARRECHART, J. **Improving Business Performance Insight. With Business Intelligence and Business Process Management.** [S.l.]: IBM REDBOOKS, 2006.

BALLARD, C.; GUPTA, A.; MAZUELA, C.; VOHNIK, S.; FARREL, D. M. **Dimensional Modeling: In a Business Intelligence Environment.** [S.l.]: IBM REDBOOKS, 2006.

INMON, W. H. **Building the Data Warehouse.** Indianapolis: WILEY, 2005.

JANUS, P.; FOUNCHÉ, G. **Pro SQL Server 2008 Analysis Services.** [S.l.]: [s.n.], 2010.

JIawei, H.; KAMBER, M.; PEI, J. **Data Mining Concepts and Techniques.** [S.l.]: [s.n.], 2012.

JÚNIOR, A. G. D. S.; BERNARDINO, C. C. **Proposta de um Sistema de Business Intelligence para exploração de indicadores de gerência de redes.** Brasília. 2009.

KIMBALL, R.; CASERTA, J. **The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data.** [S.l.]: WILEY, 2004.

KIMBALL, R.; ROSS, M. **The data warehouse toolkit: the complete guide to dimensional modeling.** [S.l.]: WILEY, 2002.

MICROSOFT. SQL Server Integration Services. **MSDN Microsoft**, 2011. Disponível em: <<http://msdn.microsoft.com/pt-br/library/ms141026.aspx>>. Acesso em: 19 maio 2011.

SPERLEY, E. **Enterprise Data Warehouse: planning, building and implementation.** [S.l.]: Prentice Hall, 1999.

TURBAN, E.; SHARDA, R.; ARONSON, J. E.; KING, D. **BUSINESS INTELLIGENCE Um enfoque gerencial para a Inteligência do Negócio.** Tradução de Fabiano Bruno Gonçalves. Porto Alegre: Bookman, 2009.

VEERMAN, E.; LANCHEV, T.; SARKA, D. **Microsoft SQL Server 2008 - Business Intelligence Development and Maintenance.** [S.l.]: Microsoft Press, 2009.