

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Raphael das Chagas Campos

**PROCESSO DE MIGRAÇÃO DE APLICAÇÃO JAVA PARA
PROGRAMAS DE ETL COM ÊNFASE NO CONCEITO DE
BUSINESS INTELLIGENCE**

Niterói

2009

Raphael das Chagas Campos

**PROCESSO DE MIGRAÇÃO DE APLICAÇÃO JAVA PARA
PROGRAMAS DE ETL COM ÊNFASE NO CONCEITO DE
BUSINESS INTELLIGENCE**

**Monografia apresentada ao Departamento de Ciência
da Computação da Universidade Federal Fluminense
como parte dos requisitos para obtenção do Grau de
Bacharel em Ciência da Computação.**

Orientadora: Bianca Zadrozny

Niterói

2009

Raphael das Chagas Campos

**PROCESSO DE MIGRAÇÃO DE APLICAÇÃO JAVA PARA
PROGRAMAS DE ETL COM ÊNFASE NO CONCEITO DE
BUSINESS INTELLIGENCE**

**Monografia apresentada ao Departamento de Ciência
da Computação da Universidade Federal Fluminense
como parte dos requisitos para obtenção do Grau de
Bacharel em Ciência da Computação.**

Aprovado em dezembro de 2009.

BANCA EXAMINADORA

Profa. Bianca Zadrozny, Ph.D.
Orientadora
UFF

Prof. Leonardo Cruz da Costa, M.Sc.
UFF

Profa. Luciana Ferraz Thomé, D.Sc.
UFF

Niterói

2009

RESUMO

Esta monografia tem como objetivo apresentar um estudo sobre o conceito de Business Intelligence, mostrando sua importância dentro das empresas para ajudar os executivos na tomada de decisão. Um sistema de Business Intelligence envolve uma gama de tecnologias que serão estudadas ao longo desse trabalho. Além disso, é apresentado um estudo de caso mostrando a migração de uma aplicação feita em Java para o ambiente de Business Intelligence, abordando na prática como funciona esse conceito tão pouco difundido nas universidades.

Palavras Chave:

Business Intelligence, estudo de caso, migração de aplicações

ABSTRACT

The goal of this work is to present a study about the concept of Business Intelligence, showing its importance to the companies in helping executives in decision-making. A Business Intelligence system involves a range of technologies that will be studied throughout this work. Moreover, a case study showing the migration of an application written in Java to the Business Intelligence environment is presented, demonstrating in practice how a concept that is so little known in universities functions.

Keywords:

Business Intelligence, case study, application migration

LISTA DE ACRÔNIMOS

BI: Business Intelligence

ERP: Enterprise Resource Planning

DW: Data Warehouse

DM: Data Mart

COBOL: COmmon Business Oriented Language

FORTRAN: FORmula TRANslation System

DASD: Direct Access Storage Device

SGBD: Sistema Gerenciador de Banco de Dados

EIS: Enterprise Information Systems

CPD: Centro de Processamento de Dados

DSS: Decision Support System

OLAP: On-Line Analytical Processing

ETL: Extract Transform Load

OLTP: Online Transaction Processing

SAP: Sistemas, Aplicações e Produtos para Processamento de Dados

DOLAP: Desktop Online Analytical Processing

ROLAP: Relational Online Analytical Processes

MOLAP: Multidimensional Online Analytical Processes

HOLAP: Hibrid Online Analytical Processing

DDL: Data Definition Language

DBA: DataBase Administrator

A Deus.

Reconhecemos que todos somos senhores do saber. O saber ensoberbece, mas o amor edifica. Se alguém julga saber alguma coisa, com efeito, não aprendeu ainda como convém saber. (1 Coríntios 8;1-2)

AGRADECIMENTOS

À Universidade Federal Fluminense,

A Professora Bianca Zadrozny, pela sua orientação.

A todos os professores do Instituto de Computação,
dos quais tive o privilégio de ser aluno.

A toda equipe de Business Intelligence da Ipiranga.

Ao João Henrique Romano Arienti e Igor Araújo pela ajuda para
realização deste trabalho.

A Marcella Torres, por sua paciência e companhia.

Aos meus pais, por sua infinita dedicação e por tudo o que me
proveram.

SUMÁRIO

1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	13
1.2 OBJETIVOS	13
1.3. ORGANIZAÇÃO	14
2 O QUE É BUSINESS INTELLIGENCE?	15
2.1 CONCEITO	15
2.2 HISTÓRICO	16
2.3 DIMENSÕES ENVOLVIDAS	17
2.4 BENEFÍCIOS DO BUSINESS INTELLIGENCE	17
3 COMPONENTES DE UM SISTEMA DE BUSINESS INTELLIGENCE	23
3.1 DATA WAREHOUSE	23
3.1.1 INTRODUÇÃO	23
3.1.2 O QUE É DATA WAREHOUSE?	24
3.1.3 O QUE DATA WAREHOUSE <u>NÃO</u> É	27
3.1.4 OS COMPONENTES DO DATA WAREHOUSE	28
3.1.5 MODELAGEM MULTIDIMENSIONAL	30
3.1.6 DATA MARTS	38
3.1.7 METADADOS	40
3.1.8 TABELAS AGREGADAS	42
3.2 FERRAMENTAS DE ETL	43
3.2.1 CONCEITOS	43
3.2.2 EXTRAÇÃO	43
3.2.3 TRANSFORMAÇÃO	44
3.2.4 CARGA	45

3.3 FERRAMENTAS OLAP	46
3.3.1 CONCEITO	46
3.3.2 TIPOS DE FERRAMENTAS	47
4 PROCESSO DE MIGRAÇÃO	50
4.1 INTRODUÇÃO	50
4.2 FASES DO PROCESSO	52
4.3 PRIMEIRA FASE	52
4.4 SEGUNDA FASE	53
4.5 TERCEIRA FASE	54
4.6 QUARTA FASE	80
5 CONCLUSÃO	87
6 REFERÊNCIAS	88
APÊNDICE A	90
APÊNDICE B	109

LISTA DE FIGURAS

Figura 2.1: Quadro Evolutivo	16
Figura 2.2: Quadro Evolutivo	16
Figura 2.3: Dimensões Envolvidas	17
Figura 2.4: Estatísticas do B.I	19
Figura 2.5: Estatísticas do B.I	20
Figura 2.6: Estatísticas do B.I	21
Figura 2.7: Estatísticas do B.I	22
Figura 3.1: Informações de Vendas orientadas para o assunto	25
Figura 3.2: Integração de esquemas de codificação	26
Figura 3.3: OLTP para leitura/gravação versus data warehouse read-only	26
Figura 3.4: A dimensão tempo no sistema OLTP versus no data warehouse	27
Figura 3.5: Componentes envolvidos no ambiente de Data Warehouse	29
Figura 3.6: Exemplo de Pesquisa Multidimensional	30
Figura 3.7: Esquema Estrela (Star)	32
Figura 3.8: Exemplo de um Modelo Multidimensional implementado com o Esquema Estrela	33
Figura 3.9: Exemplo do Esquema Flocos de Neve	34
Figura 3.10: Representação de um cubo com dimensões de Produto, Região e Tempo	35
Figura 3.11: Operadores Dimensionais Drill-down e Drill-Up	37
Figura 3.12– Operadores Dimensionais Drill Across e Drill Through	38
Figura 3.13 – Estratégia de implementação gradativa de Data Marts	39
Figura 3.14 – Exemplo de um modelo de repositório de metadados	40
Figura 3.15 – Extração dos dados de múltiplas fontes	44
Figura 3.16 – Características OLAP	47
Figura 4.1 – Layout da aplicação Java	50
Figura 4.2 – Geração do Relatório em Excel	51
Figura 4.3 – Relatório gerado pela aplicação	52
Figura 4.4 – Modelagem utilizada no processo de migração	53
Figura 4.5 – Seleciona os clientes que cancelaram sua participação	54
Figura 4.6 – Seleciona os clientes que não estão aptos para participarem	55
Figura 4.7 – Seleciona os clientes que estão aptos, porém o assessor ainda não realizou a negociação	56

Figura 4.8 – Selecciona os clientes que estão com o termo pendente de assinatura	57
Figura 4.9 – Selecciona os clientes participantes com termos já assinados	58
Figura 4.10 – Programas de ETL	59
Figura 4.11 – Programa ETL – Fluxo de Delete	60
Figura 4.12 – Programa ETL – Update Strategy	61
Figura 4.13 – Programa ETL – Fluxo de Delete Ampliado	62
Figura 4.14 – Programa ETL – Fluxo de Delete – Edit	62
Figura 4.15 – Programa ETL – Fluxo de Delete – SQL	63
Figura 4.16 – Programa ETL – Fluxo de Saída	64
Figura 4.17 – Programa ETL - Expression	66
Figura 4.18 – Programa ETL – Fluxo de Saída – Ampliado	67
Figura 4.19 – Programa ETL – Fluxo dos Aptos – Mapplet	67
Figura 4.20 – Programa ETL – Lookup	68
Figura 4.21 – Programa ETL – Router	69
Figura 4.22 – Programa ETL – Router – Condição	69
Figura 4.23 – Programa ETL – Fluxo dos Aptos	70
Figura 4.24 – Programa ETL – Fluxo dos Aptos - SQL	71
Figura 4.25 – Programa ETL – Fluxo dos Não Aptos	72
Figura 4.26 – Programa ETL – Fluxo dos Não Aptos – Ampliado	73
Figura 4.27 – Programa ETL – Fluxo dos Não Aptos – Mapplet	74
Figura 4.28 – Programa ETL – Fluxo Termo Gerado	75
Figura 4.29 – Programa ETL – Fluxo Termo Aceito	76
Figura 4.30 – Sessões	78
Figura 4.31 – Sessão – Conexões	79
Figura 4.32 – Workflow	80
Figura 4.33 – Pastas dos Relatórios	80
Figura 4.34 – Relatórios	81
Figura 4.35 – Lista as situações da participação dos Clientes no Plano de Marketing.....	81
Figura 4.36 – Análise da negociação da adesão do cliente ao Plano de Marketing.....	82
Figura 4.37 – Análise da emissão e do aceite do Termo de Adesão do Plano de Mark.....	83
Figura 4.38 – Análise da participação do cliente no Plano de Marketing.....	84
Figura 4.39 – Analisa o cliente e como está a sua participação no Plano de Marketing.....	85
Figura 4.40 – Evolução dos clientes que aderiram ao Plano de Marketing ao longo do ano.....	86

1 INTRODUÇÃO

1.1 Motivação

Este trabalho aborda um conjunto de diferentes tecnologias que unidas agilizam o processo de geração de informação para o apoio na tomada de decisão conhecida como Business Intelligence.

A escolha deste tema foi motivada pelo fato do conceito de Business Intelligence ser pouco difundido no meio acadêmico e, ao mesmo tempo, cada vez mais presente nas empresas e no mundo dos negócios.

As ferramentas utilizadas nesse processo são capazes de armazenar os dados de forma histórica e consistente em um grande repositório de dados, que chamamos de Data Warehouse. É possível também, a partir dessas ferramentas, visualizarmos os dados de forma flexível e rápida, identificando as novas necessidades de negócio e gerando vantagens competitivas para a organização.

Em uma arquitetura de Business Intelligence, os dados são levados dos sistemas transacionais para o Data Warehouse, que nada mais é do que um banco de dados criado especialmente para dar suporte à decisão, carregando os dados históricos da empresa e permitindo que os executivos analisem as informações conforme acharem melhor.

Assim, o Business Intelligence traz um diferencial competitivo para as organizações onde os executivos poderão antecipar-se e prever o mercado antes do seu concorrente.

Espera-se que este projeto forneça os conhecimentos básicos para o entendimento desse processo e como o mesmo pode ser aplicado.

1.2 Objetivos

O objetivo principal desse projeto é apresentar um estudo de caso mostrando a migração de uma aplicação Java para ETL e identificando os benefícios dessa migração. Além disso, disponibilizar um estudo acerca do conceito de Business Intelligence (BI), abordando cada componente de um processo de BI, desde a extração das informações dos sistemas transacionais, passando pela modelagem multidimensional desses dados, o processo de limpeza e transformação dos dados e o carregamento das informações dentro do Data Warehouse, que a partir daí serão consultadas pelos executivos para análises e ajudando na tomada de decisão das organizações.

1.3 Organização

O trabalho está organizado em cinco capítulos, como descrito a seguir. O Capítulo 2 aborda os conceitos básicos, o histórico, as dimensões envolvidas e os benefícios que o processo de BI traz para as organizações. O Capítulo 3 apresenta os componentes deste processo, como a construção do Data Warehouse que é o núcleo principal de qualquer projeto de BI, as ferramentas de ETL utilizadas, que realizam o tratamento e a limpeza dos dados transformando-os em inteligência, e as ferramentas OLAP que realizam a análise dos dados para auxiliar no momento da decisão por parte dos executivos. O Capítulo 4 apresenta um estudo de caso que tem como propósito demonstrar na prática a migração uma aplicação que roda em Java para o ambiente de Business Intelligence, explorando os benefícios dessa migração. No Capítulo 5, é apresentada uma breve conclusão do trabalho feito.

2 O QUE É BUSINESS INTELLIGENCE?

2.1 Conceito

O conceito de Business Intelligence (BI) está ligado à extração e consolidação de informações contidas nos diversos sistemas que uma determinada empresa possui, como os famosos ERP (Enterprise Resource Planning), sistemas integrados de gestão cujo objetivo é facilitar o aspecto operacional das empresas. O intuito do BI é reunir essas informações em um único local, a fim de que elas fiquem facilmente disponíveis para os usuários.

Segundo Tyson [15], BI é um processo que envolve a coleta, análise e validação de informações sobre concorrentes, clientes, fornecedores, candidatos potenciais à aquisição e alianças estratégicas. O processo de BI analisa e valida todas essas informações e as transforma em conhecimento estratégico. Dessa forma, os usuários poderão fazer análises que os permitam tomar decisões estratégicas, com o objetivo de manter a empresa sempre à frente dos seus concorrentes. [15]

O sistema que armazena os dados provenientes de diferentes fontes é conhecido como Data Warehouse (DW), termo que pode ser traduzido como Armazém de Dados [2]. No DW, as informações da empresa devem estar acessíveis, integradas (limpas e consistentes), sincronizadas no tempo e organizadas de forma multidimensional para análises. Geralmente, o DW é subdividido em conjuntos menores chamados Data Marts (DM) que agregam dados referentes a um assunto específico dentro da empresa, como por exemplo, Vendas ou Estoque. Note que não se tem um tamanho fixo para um DW, já que ele pode crescer ao longo do tempo à medida que os dados são atualizados, e pelo fato de se criar novos Data Marts. Assim, pode-se extrair dados atuais desse depósito estruturado de informações com maior rapidez e maior segurança para ajudar na análise e tomada de decisões.

2.2 Histórico

As figuras 2.1 e 2.2 mostram um quadro evolutivo do BI ao longo do tempo, tendo início na Idade Antiga evoluindo até o Século XVI [1].

QUADRO EVOLUTIVO DO B.I.	
Período	O que aconteceu ?
Idade Antiga	<ul style="list-style-type: none"> • Povos do Oriente Médio cruzavam informações da natureza, marés, etc. para analisarem a viabilidade do cultivo de determinadas espécies, períodos de pesca abundante, etc.
Século XVI	<ul style="list-style-type: none"> • Rainha Elizabeth I, determinou que a base da força inglesa fosse "informação e comércio" e ordenou então ao filósofo Francis Bacon que inventasse um sistema dinâmico de informação, o qual foi amplamente aplicado pelos ingleses.
Década de 60	<ul style="list-style-type: none"> • Evolução dos computadores que deixaram de ocupar salas enormes, na medida em que diminuíram de tamanho. • As empresas passaram a perceber os dados como uma possível e importante fonte geradora de informações decisórias e que renderiam eventuais lucros. • Época dos cartões perfurados. • Predominância das linguagens de programação COBOL, ASSEMBLY e FORTRAN.

Figura 2.1 – Quadro Evolutivo (Extraído de [1])

Década de 70	<ul style="list-style-type: none"> • Evolução das formas de armazenamento e acesso a dados - DASD e SGBD (Sistema Gerenciador de Banco de Dados). • Desenvolvido o EIS.
Década de 80	<ul style="list-style-type: none"> • Desenvolvimento e evolução das linguagens de programação CLIPPER e PASCAL • Início da aplicação do termo Business Intelligence
Década de 90	<ul style="list-style-type: none"> • Início do termo CPD nas empresas • Desenvolvimento do Data Warehouse • O setor corporativo passou a se interessar pelas soluções de BI
Década de 2000	<ul style="list-style-type: none"> • Evolução dos conceitos de DSS (Decision Support System - sistema de suporte à decisão), Planilhas Eletrônicas, Geradores de Consultas e de Relatórios, Data Marts, Data Mining, Ferramentas OLAP, entre outras • Estreitamento das conexões entre o BI e o ERP • Surgimento do termo E-business

Figura 2.2 – Quadro Evolutivo (Extraído de [1])

2.3 Dimensões Envolvidas

A figura 2.3 representa as dimensões envolvidas em um processo de Business Intelligence [16].



Figura 2.3 – Dimensões Envolvidas (Extraído de [16])

Processos – Inclui os processos de extração de dados a partir dos sistemas fonte, de tratamento, de transformação e preparação dos dados para análises, envolvendo também o processo de distribuição das informações para a organização. (Arquitetura de Extract-Transform-Load ou ETL).

Organização/Cultura - Estabelecimento das principais funções/papéis envolvidos na manutenção da arquitetura de Business Intelligence. Está relacionada também a segurança/control de acesso do ambiente de Data Warehouse. (Arquitetura de Pessoas).

Aplicações - Tipos de análises a serem suportados pela arquitetura de Business Intelligence. Envolve as ferramentas OLAP's, com a disponibilização de relatórios para consultas através de portais executivos, aplicações Web. (Arquitetura de Distribuição de Informações).

Dados – Questões relacionadas à falta de qualidade de dados e de falta das informações necessárias para se implementar os novos conceitos de negócio requeridos no DW. (Arquitetura de Dados).

Tecnologia – Infra-estrutura tecnológica necessária para suportar as aplicações, dados e processos envolvidos. Como por exemplo: Rede, Servidores, Banco de Dados. (Arquitetura Tecnológica).

2.4 Benefícios do Business Intelligence.

Os principais benefícios alcançados por uma organização após a implantação de um sistema de Business Intelligence são: [17]

- Ter o acesso rápido, facilitado e confiável às informações da organização sem gerar impacto negativo às suas operações, ou seja, permitir que os profissionais das áreas de negócio tenham acesso a essas informações sem depender da intervenção direta da área de Tecnologia da Informação.

- Responder de maneira completa, rápida e segura a algumas questões simples como: Como estamos? Como deveríamos estar? Porque estamos assim? Ou ainda: Quanto capacitamos, produzimos, faturamos, gastamos ou lucramos? O quê? Quando? Como? E onde?

- Apontar tendências e tomar decisões eficientes e adequadas em tempo hábil.

- Corrigir os defeitos e potencializar os acertos, gerando um diferencial competitivo com o Mercado.

- Conhecer melhor a realidade da empresa, internamente. Perceber suas relações com o mercado, tanto interno, quanto externo, melhorando assim, o desempenho da sua atuação e inovações necessárias ao seu crescimento.

- Detectar as fraudes, erros e custos excessivos da empresa.

- Aperfeiçoar a relação com os clientes.

- Permitir o direcionamento do capital intelectual da empresa para a execução das funções mais nobres: o pensar e o agir.

Dando continuidade, aos benefícios que o BI pode trazer para dentro de uma empresa, foi realizada uma entrevista com diversas empresas de diferentes portes em todo o Brasil. Dentre algumas perguntas abordadas nesta entrevista, foi selecionada quatro que podem dar uma boa base sobre os resultados eficientes que o Business Intelligence traz para as empresas que o adotam.

As perguntas foram dirigidas aos clientes dessas empresas e os resultados foram esses: [1]

1. Está satisfeito com as políticas de tratamento individual?

	Antes B.I. (%)	Após B.I. (%)
Ótimo	38	78
Bom	40	20
Razoável	10	2
Péssimo	12	0

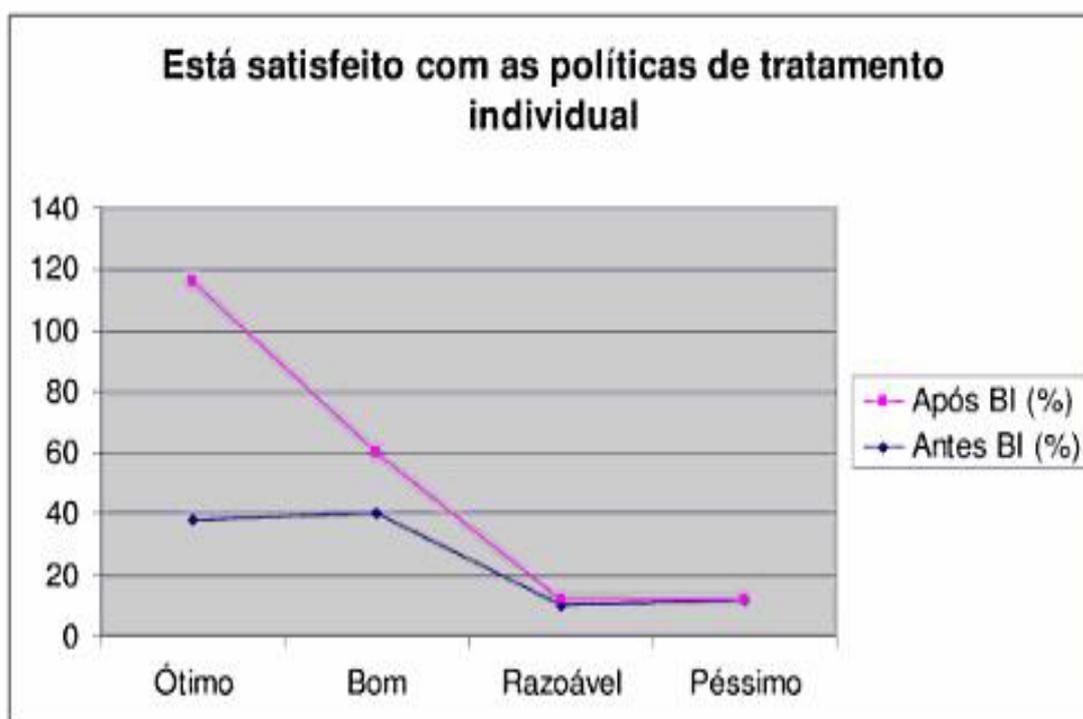


Figura 2.4 – Estatísticas do B.I (Extraído de [1])

2. Nossos produtos e serviços estão sendo bem aceitos pelo mercado consumidor?

	Antes B.I. (%)	Após B.I. (%)
Ótimo	49	84
Bom	35	12
Razoável	8	4
Péssimo	8	0

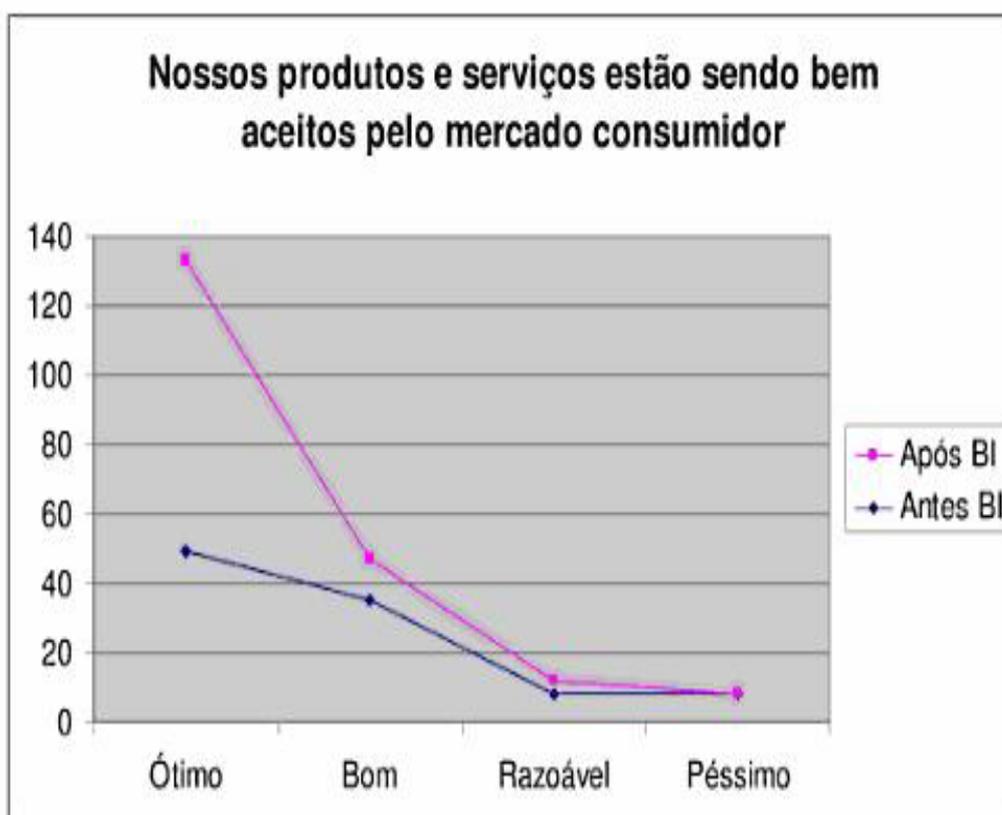


Figura 2.5 – Estatísticas do BI (Extraído de [1])

3. O tempo de atendimento aos novos prospects reduziu?

	Antes B.I. (%)	Após B.I. (%)
Ótimo	15	57
Bom	47	40
Razoável	22	2
Péssimo	16	1

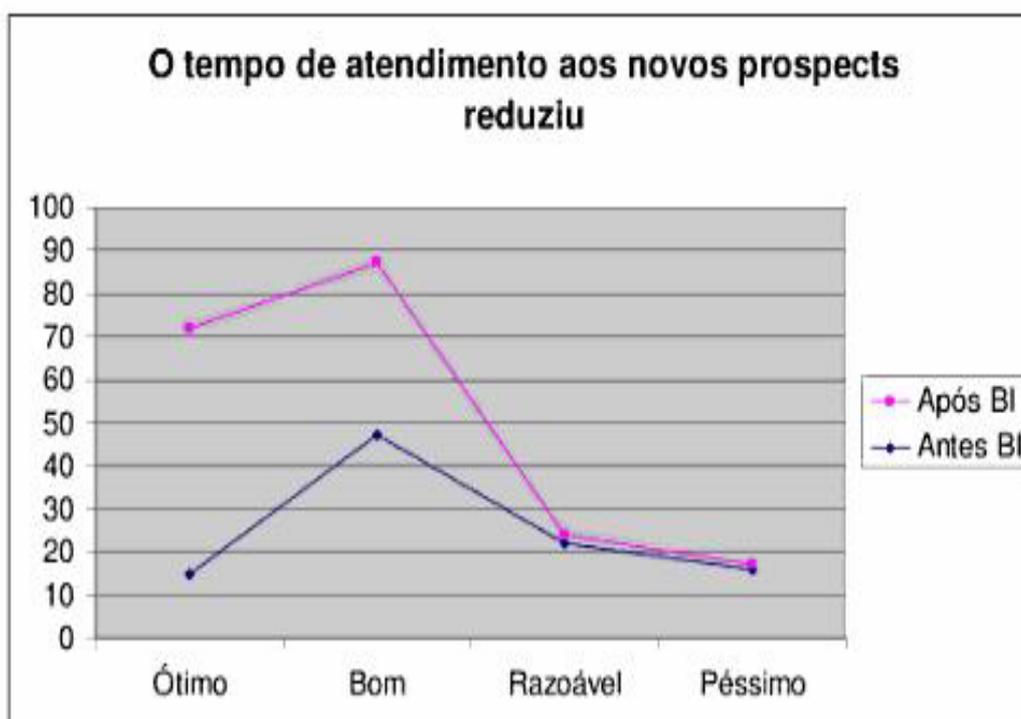


Figura 2.6 – Estatísticas do BI (Extraído de [1])

4. O tempo de resposta para questionamentos de clientes está dentro do previsto?

	Antes B.I. (%)	Após B.I. (%)
Ótimo	13	69
Bom	65	25
Razoável	16	6
Péssimo	6	0

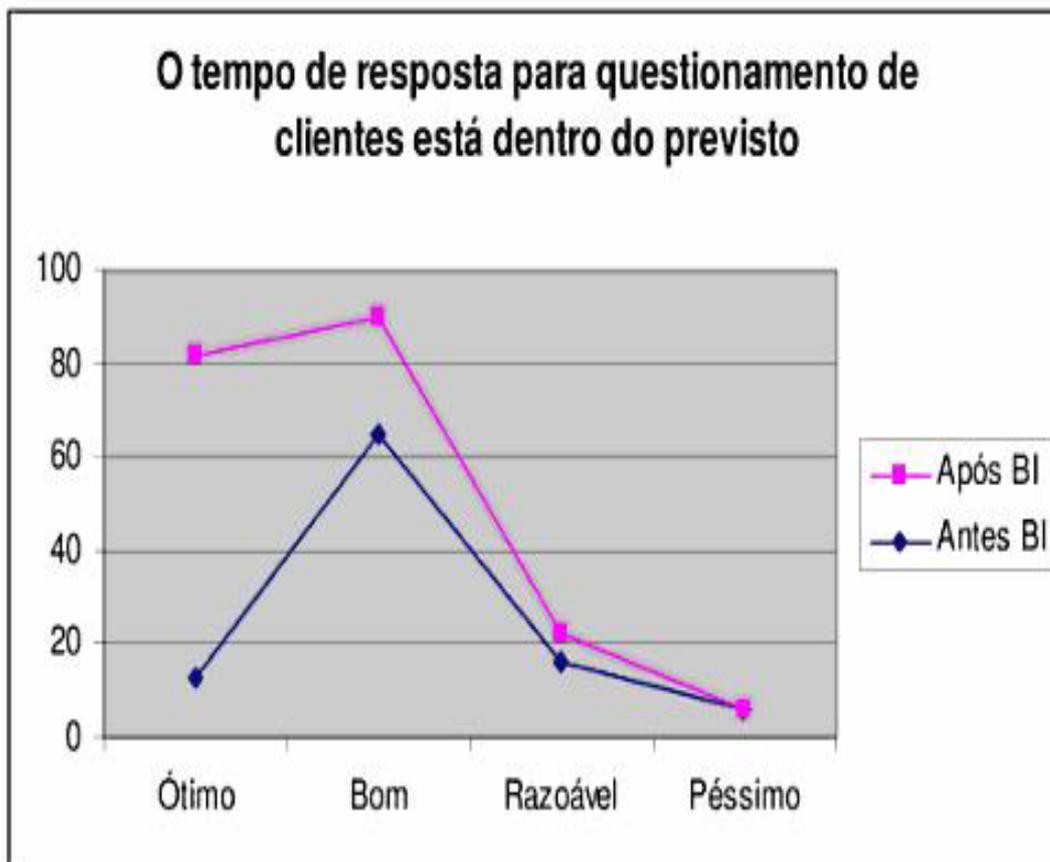


Figura 2.7 – Estatísticas do BI (Extraído de [1])

É possível verificar pelos números e gráficos acima, que a maioria dos entrevistados estão satisfeitos com o retorno que o Business Intelligence tem trazido para as grandes e médias empresas, contribuindo para o crescimento e sucesso dessa nova tecnologia. [1].

3 COMPONENTES DE UM SISTEMA DE BUSINESS INTELLIGENCE

3.1 Data Warehouse

3.1.1 Introdução

Sabe-se que os bancos de dados são de vital importância para as empresas e também que sempre foi difícil analisar os dados neles existentes. Tudo isso porque geralmente as grandes empresas detêm um volume enorme de dados e estes estão em diversos sistemas diferentes espalhados por ela. Desta forma, é difícil buscar informações que permitam tomar decisões embasadas num histórico dos dados. Por outro lado, com base nesse histórico, seria possível identificar tendências e posicionar a empresa estrategicamente para ser mais competitiva e conseqüentemente maximizar os lucros diminuindo o índice de erros na tomada de decisão. Desta forma, introduziu-se um novo conceito no mercado, o Data Warehouse (DW) – uma arquitetura de Banco de Dados com informações de caráter gerencial voltado para: suporte à decisão, planejamento estratégico, análise de comportamento de clientes e análise de vendas. Funciona como um provedor de informações de uma organização, pois concentra todas as informações estratégicas e históricas, extraídas dos sistemas transacionais relativos a clientes e produtos. [3]

Um Data Warehouse permite a geração de dados integrados e históricos auxiliando os executivos das empresas decidirem embasados em fatos e não em intuições ou especulações, o que reduz a probabilidade de erros e aumenta a velocidade na hora da decisão. Cerca de 88% dos diretores admitem que dedicam quase 75% do tempo às tomadas de decisão apoiadas em análises subjetivas (Aspect International Consulting [3]), menosprezando o fato de que por volta de 100% deles tem acesso a computadores [3]. Conhecer mais sobre essa tecnologia permite aos administradores descobrir novas maneiras de diferenciar sua empresa numa economia globalizada, deixando-os mais seguros para definirem as metas e adotarem diferentes estratégias em sua organização, conseguindo assim visualizar, antes de seus concorrentes, novos mercados e oportunidades, atuando de maneira diferente conforme o perfil de seus consumidores [3].

3.1.2 O que é um Data Warehouse?

Para entender melhor o que é um Data Warehouse, é interessante fazer uma comparação com o conceito tradicional de banco de dados.

“Um Banco de Dados é uma coleção de dados operacionais armazenados e utilizados pelo sistema de aplicações de uma empresa específica”. Em que os dados utilizados por uma empresa são chamados de ‘operacionais’ ou ‘primitivos’. [1]

A partir dessa definição sobre dados operacionais, podemos dizer que o DW é um conjunto de dados que foi extraído dos sistemas operacionais (OLTP), onde esses dados passaram por um processo de refinamento para que possam ser chamados de dados “gerenciais”, “informativos” ou “analíticos” [1] Em seu livro, *Building the Data Warehouse* [18], Inmon definiu o warehouse como um conjunto de dados orientado por assuntos, integrados, não-volátil, variante no tempo, no apoio de decisões gerenciais. Com isso, pode-se aprofundar em cada uma dessas definições:

Orientado a Assunto: Os bancos de dados OLTP normalmente são destinados a conter informações a respeito de pequenos assuntos da empresa. Por exemplo, um varejista poderia ter sistemas de entrada de pedidos separados e bancos de dados para vendas no varejo, por catálogo e no atacado. Cada sistema oferecerá suporte para consultas a respeito das informações que ele captura. Mas, suponha que um usuário deseja fazer uma consulta sobre todas as vendas e não simplesmente as vendas capturadas por um sistema em particular. Em um determinado momento, supõe-se que o usuário, deseja um relatório que contenha somente as vendas de um determinado produto, independente do sistema responsável por essa venda. Para tratar desse tipo de situação, o Data Warehouse deve estar orientado por assunto, ou seja, organizado por áreas de assunto, como vendas, em vez de em torno de origens de dados operacionais [3] [4]. A figura 3.1 a seguir, mostra um exemplo clássico da migração de dados de vários sistemas operacionais relacionados a Vendas se unem em um único, englobando todas essas informações dentro do data warehouse.

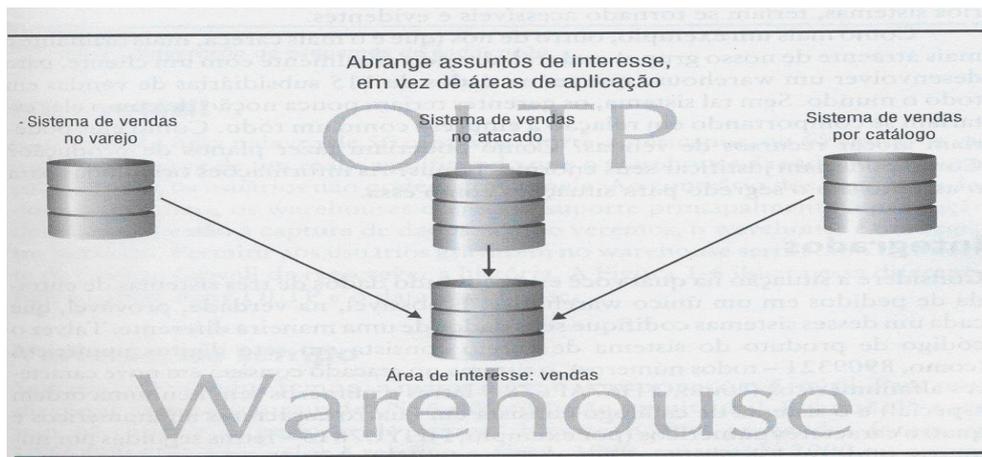


Figura 3.1 – Informações de Vendas orientadas para o assunto (Extraído de [4])

Imagine que uma determinada empresa deseja reunir dados de 115 filiais espalhadas por todo o Brasil. Caso, essa empresa não tivesse um Data Warehouse, os gerentes teriam pouca noção de como elas estariam se comportando em relação à empresa como um todo. Como eles poderiam alocar recursos de vendas? Como poderiam fazer planos de produção? Como poderiam justificar seus bônus? As informações orientadas para o assunto são o segredo para situações como essa.

Integrado: Considere a situação em que você tem dados de três sistemas diferentes referentes a clientes em que o campo sexo de um determinado cliente está codificado de diferentes maneiras nos três sistemas. No primeiro sistema o campo sexo está definido como H/M, no segundo sistema como 1/0 e no terceiro sistema temos M/F. Quando a integração for realizada para dentro do Data Warehouse, um padrão único precisa ser estabelecido para o campo sexo [3] [4]. A figura 3.2 a seguir, mostra um exemplo similar, para a situação na qual está reunindo dados de três sistemas de entrada de pedidos em um único Warehouse.

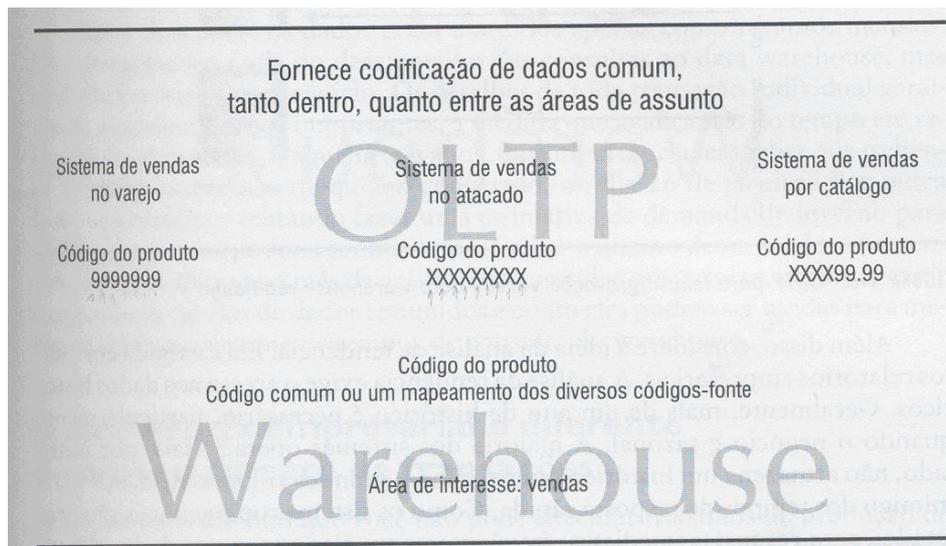


Figura 3.2 – Integração de esquemas de codificação (Extraído de [4])

Não Volátil: Significa que os dados no warehouse nunca mudam, somente quando é necessário fazer algum acerto em um dado que já tinha sido carregado, porém de forma errada. Os usuários não conseguem gravar dados dentro de um Warehouse, apenas ler esses dados. Ao contrário, dos sistemas operacionais em que o usuário pode ler e gravar os dados, os warehouses oferecem suporte principalmente à produção de relatórios e não à captura de dados [3] [4]. A figura 3.3 mostra a diferença entre os dados dentro de um Warehouse e os dados em um sistema operacional (OLTP).

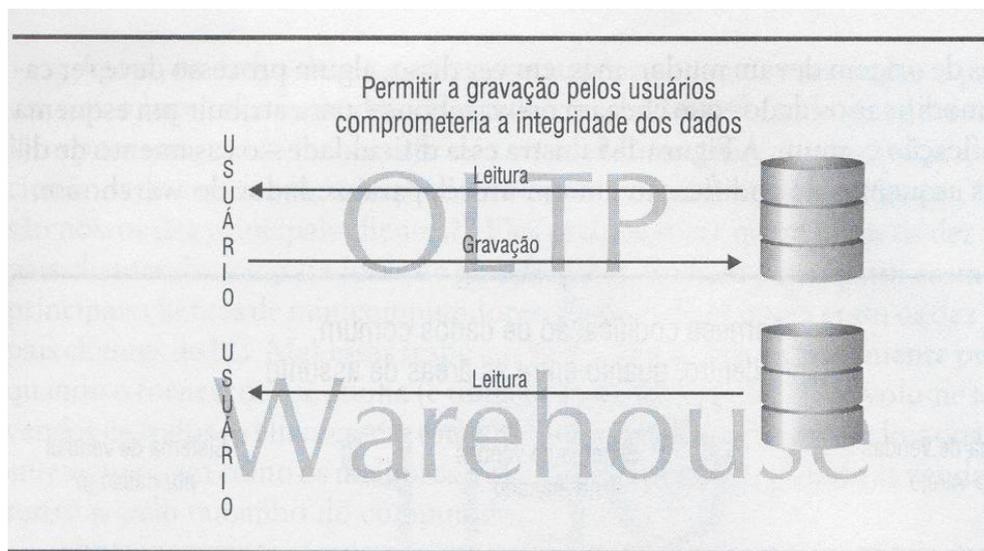


Figura 3.3 – OLTP para leitura/gravação versus data warehouse read-only (Extraído de [4])

Variante no Tempo: O tempo é sem dúvida um dos componentes chave para o entendimento de um Data Warehouse. Isso porque o Data warehouse é um registro histórico em que possui posições históricas das atividades no tempo. Pois sem o

componente tempo, a maioria das informações, se tornam inúteis. Dizer que as vendas foram de R\$ 10.000.000 não significa nada, pois não tenho a informação do intervalo de tempo exato no qual as vendas aconteceram. Esse valor é relativo a 2005, 2006, janeiro de 2007 ou ao ano corrente até o presente momento? O tempo é de fundamental importância porque permite uma melhor análise de eventos passados, oferecendo suporte a tomada de decisões presentes e a previsão de eventos futuros. Os Data Warehouses são bancos de dados enormes que armazenam informações ao longo do tempo, diferentes dos sistemas operacionais que foram criados para armazenar as informações diárias, não guardando informações históricas por muito tempo. [3] [4] A figura 3.4 ilustra esse componente tempo dentro do DW e dos sistemas operacionais (OLTP).

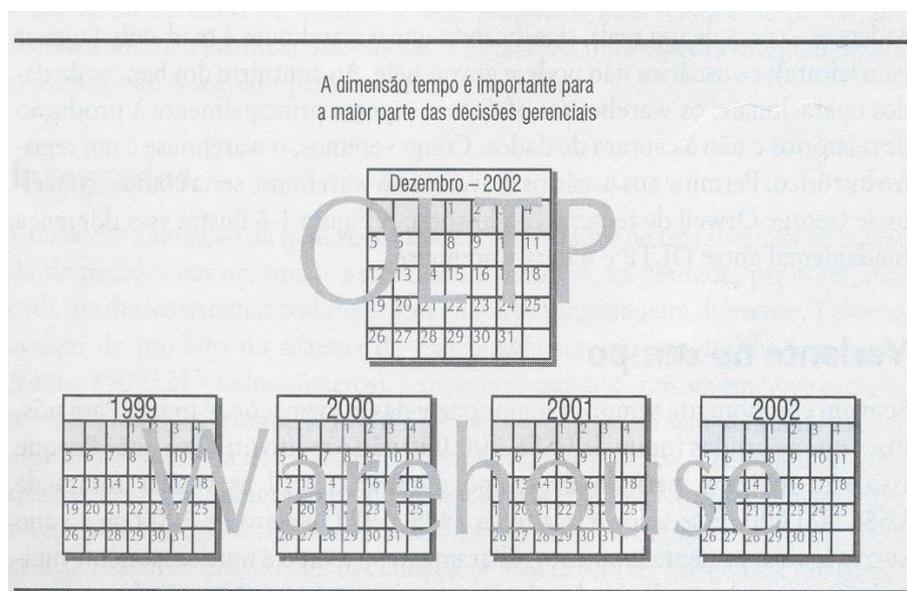


Figura 3.4 – A dimensão tempo no sistema OLTP versus no data warehouse (Extraído de [4])

3.1.3 O que Data Warehouse não é

Produto: O Data Warehouse é um sistema e não um produto. Para se construir um Data Warehouse é preciso realizar um levantamento das necessidades do negócio, ou seja, um entendimento do que o cliente necessita. A partir dessas análises, o sistema de Data Warehouse é desenvolvido. Assim, o Data Warehouse não é um produto pronto que pode ser comprado como um software qualquer de banco de dados [3].

A linguagem: Pelo fato de existirem diversos componentes e fases que compõem um Data Warehouse, ele não pode ser definido com uma linguagem de programação. O que acontece, é que o Data Warehouse dá suporte a um conjunto de linguagens de

programação que tem a função de extrair os dados, limpar esses dados e apresentar as informações para os usuários [3].

Projeto: O sistema de Data Warehouse é mais bem definido como um processo do que um projeto. Visto que projeto tem a idéia de início-meio-fim, enquanto que um sistema de DW não possui essa característica, pois ele é definido como um processo que está em constante evolução ao longo do tempo não tendo um fim estabelecido [3].

Modelagem: O sistema de Data Warehouse não é somente um modelo de banco de dados. Ele envolve todo o processo do DW que inclui todos os procedimentos de ETL, limpeza e apresentação dos dados [3] que serão apresentados nas próximas seções.

Cópia do sistema OLTP: Como já vimos anteriormente, não podemos afirmar que o DW é simplesmente uma cópia de um sistema transacional. Existem ferramentas que são utilizadas para realizar essa migração dos dados de um ambiente para o outro. Nesse processo há uma transformação dos dados, além do armazenamento dessas informações dentro do DW, que ocorre de forma diferente dos sistemas transacionais. Um exemplo poderia ser quanto ao histórico dos dados, que não existe nos sistemas OLTP com tanta frequência como dentro do DW [3].

3.1.4 Os componentes do Data Warehouse

O ambiente de Data Warehouse é formado por alguns componentes. Cada componente em um Warehouse possui uma função específica. Os quatro principais componentes podem ser classificados como:

- Operational Source Systems (Sistemas Operacionais de Origem)
- Data Staging Area (Área de Staging)
- Data Presentation Area (Área de Apresentação dos Dados)
- Data Access Tools (Ferramentas de Acesso aos Dados)

A figura 3.5 a seguir, ilustra os componentes envolvidos.

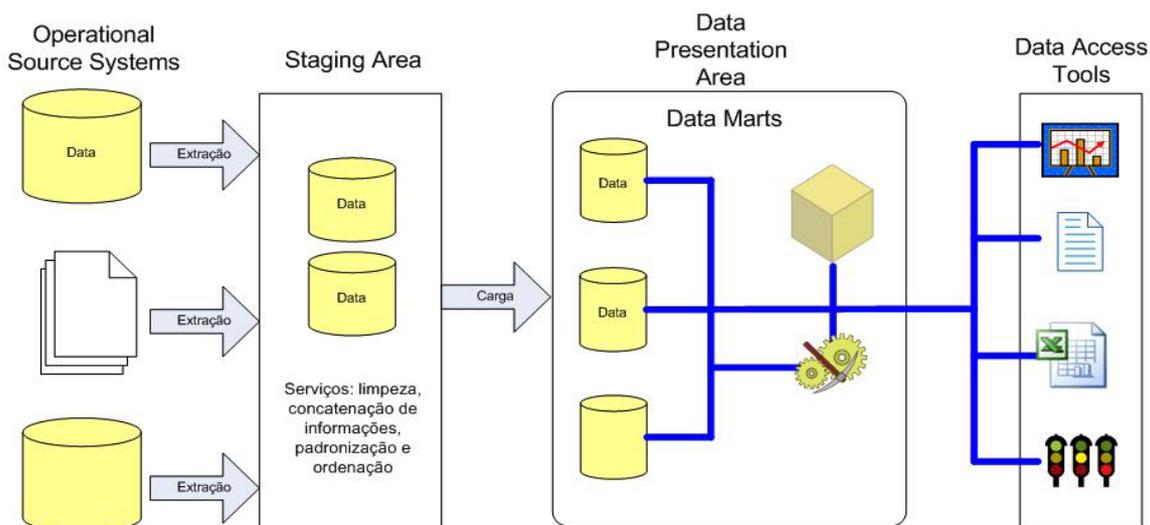


Figura 3.5 – Componentes envolvidos no ambiente de Data Warehouse (Extraído de [6]).

Operational Source Systems – São os sistemas conhecidos como transacionais, que capturam as transações diárias de uma determinada organização. Exemplos desses sistemas são aqueles relacionados a Vendas, Contabilidade, etc. Sistemas de Origem podem ser também os pacotes de sistemas integrados de gestão (ERP), como SAP, Peoplesoft entre outros. Podem até ser fontes externas da empresa, como por exemplo, os dados adquiridos de fornecedores externos [3] [4] [5] [6].

Data Staging Area – É uma área composta de um espaço para armazenamento de dados e de um conjunto de processos com a função de extrair os dados de sua origem, transformando esses dados no padrão desejado pelo usuário e carregando essas informações dentro do Data Warehouse. Esses processos são conhecidos como ETL (Extract – Transformation – Load), que realiza o refinamento dos dados. Ou seja, poderíamos pensar que essa área de staging é similar à cozinha de um restaurante, onde os alimentos crus são transformados em uma refeição, que é degustada pelo cliente que a solicitou. Dessa forma, como na cozinha de um restaurante o cliente não pode ver o alimento sendo preparado, numa área de staging o usuário não tem acesso aos dados ali armazenados [3] [4] [5] [6].

Data Presentation Area – É nesta área em que se encontram os Data Marts que armazenam todas os dados que foram extraídos e transformados na área de staging. É aqui o local em que os dados ficam organizados, armazenados e acessíveis para os usuários consultarem e extraírem os relatórios necessários para análise. Essa área é

também onde se desenvolvem aplicações analíticas e onde são utilizadas ferramentas de acesso a dados [3] [4] [5] [6].

Data Access Tools – O último componente principal do ambiente de Data Warehouse são as ferramentas de acesso a dados. Essas ferramentas fazem consulta aos dados na área de apresentação de dados (Data Presentation Area) buscando analisar os dados da melhor forma ajudando os usuários a tomarem decisões corretas. Podem ser utilizadas ferramentas que apenas tenham o objetivo de analisar os dados de forma simples, como também com o objetivo de realizar um estudo mais profundo das informações com a utilização de ferramentas de mineração de dados [3] [4] [5] [6].

3.1.5 Modelagem Multidimensional

3.1.5.1 Introdução

Um dos fatores que levaram ao surgimento da Modelagem Dimensional foi o aumento significativo do volume de dados dentro das corporações. Este aumento estava trazendo problemas para os usuários, já que o tempo de resposta a uma consulta estava cada vez mais lento. Com a criação dessas bases de dados multidimensionais, as consultas obtiveram um ganho significativo de performance. Além disso, elas permitem que o usuário analise de forma mais clara as informações do negócio e também visualizem os dados de várias formas possíveis [2]. Um exemplo de consulta em uma base de dados multidimensional pode ser visualizado na figura 3.6:

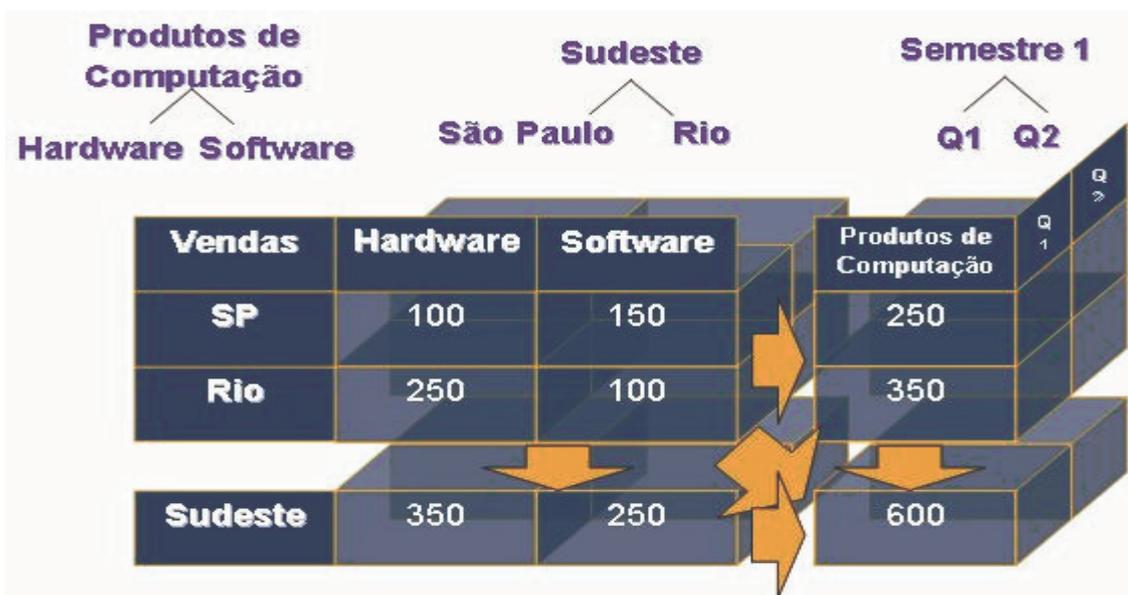


Figura 3.6 – Exemplo de Pesquisa Multidimensional (Extraído de [10])

Quando comparada com os conceitos tradicionais de Banco de Dados, essa técnica difere em diversos aspectos. Por exemplo, o conceito de normalização muito aplicado no Modelo Entidade-Relacionamento criado por Peter Chen, na maioria das vezes não é aplicado na Modelagem Dimensional. Isso ocorre, com o objetivo de obter um melhor desempenho nas consultas feitas pelos usuários, reduzindo o número de junções que teria que ser feito caso as tabelas estivessem normalizadas [2].

O modelo dimensional é formado por tabelas Fato e tabelas Dimensão. As tabelas Dimensão possuem uma probabilidade maior de não sofrerem normalização do que as tabelas Fato. Existem duas abordagens distintas quanto ao conceito de normalização de tabelas Dimensão. A primeira delas é a Star Schema (Esquema Estrela) que apoia a não normalização das tabelas Dimensão e a Snowflake Schema (Esquema de Flocos de Neve) que apoia a normalização [2]. Esses conceitos serão discutidos detalhadamente nas seções à frente.

Os Fatos e as Dimensões são tabelas como quaisquer outras em um banco de dados, apenas adquirem esse nome na modelagem multidimensional, devido à função de cada tabela delas [11].

As tabelas Dimensões representam entidades de negócios, armazenando informações de tempo, geografia, produto, cliente, etc.[2] Geralmente elas possuem uma quantidade muito grande de atributos, dentre os quais alguns representam sua hierarquia. Por exemplo, na dimensão tempo, pode existir a seguinte hierarquia: ANO - > MÊS -> DIA. Essas tabelas têm um número bem inferior de registros quando comparadas aos registros das tabelas Fato [2].

As tabelas Fato geralmente possuem relacionamento com as tabelas Dimensão [2] [7]. Elas armazenam um volume muito grande de dados, girando em torno de Gigabytes nos primeiros meses de um Data Warehouse [11].

A maioria das Tabelas Fato tem associado a elas, pelo menos quatro dimensões básicas [7].

As Dimensões Básicas são:

- Quem
- Quando
- Onde
- O Quê

A Dimensão Quem, indica que entidades participam do fato. Por exemplo, no caso do fato ser uma venda, a dimensão Quem neste caso estaria fazendo o papel de Vendedor. Toda venda tem um Vendedor.

A Dimensão Quando, é a própria dimensão tempo. Por exemplo, no caso do fato ser uma venda, a dimensão Quando neste caso estaria fazendo o papel do tempo. Toda venda é feita num período de tempo.

A Dimensão Onde, indica o local onde o fato ocorreu. Pensa-se como fato ser uma venda, o local em que aquela venda ocorreu, faria o papel da Dimensão Onde.

A Dimensão O Quê, indica qual é o objeto do fato. Neste exemplo, como o fato é uma venda, a dimensão O Quê, faria o papel do produto que foi vendido. Toda venda possui um objeto a ser vendido.

Deriva-se disso o conceito de esquema estrela (Star). A figura 3.7 mostra que pode ser construído uma estrela com dimensões e um fato.

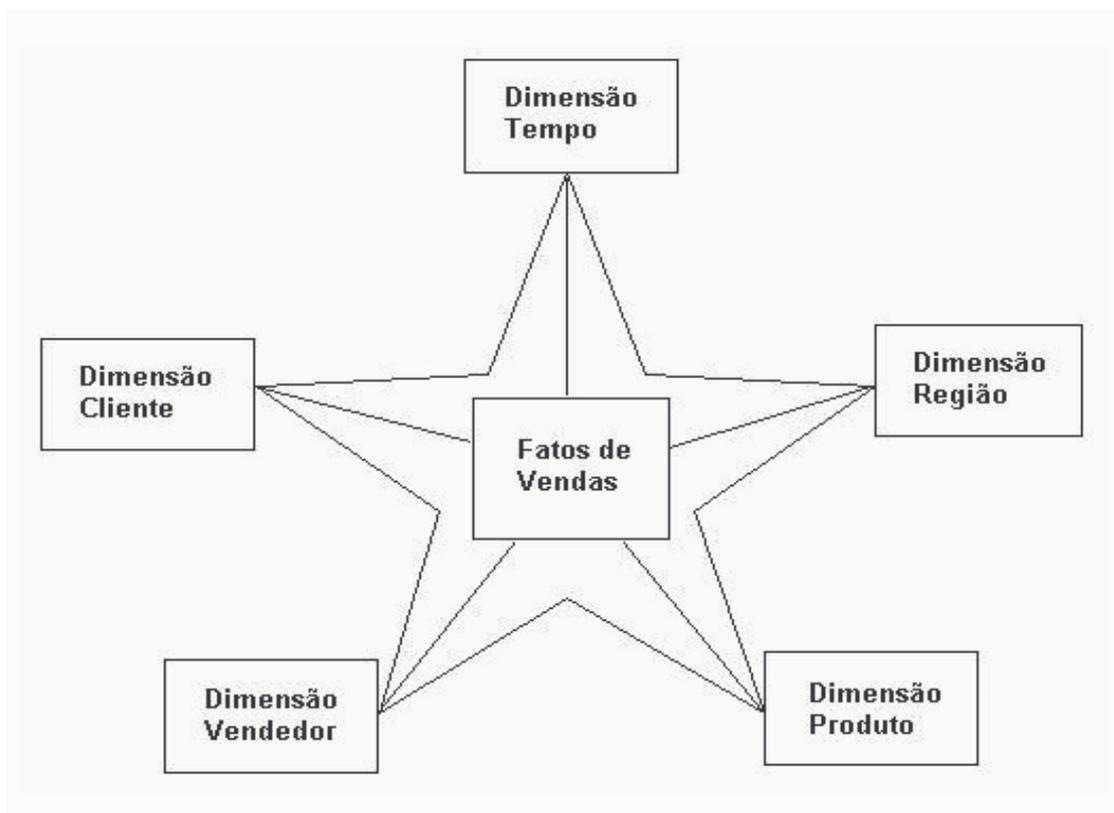


Figura 3.7 – Esquema Estrela (Star) (Extraído de [7])

3.1.5.2 Star Schema (Esquema Estrela)

Como já foi visto no final da seção anterior uma sucinta idéia a respeito do Esquema Estrela, nessa seção, será aprofundando os conceitos dessa abordagem utilizada em modelagem multidimensional e apresentada por Ralph Kimball.

Como visto, esse modelo é relativamente simples e eficiente. Possui uma única tabela Fato e contém chaves simples nas tabelas Dimensões. Cada dimensão é representada por uma única tabela. [10]

A figura ilustra um modelo de dados com base no Esquema Estrela.

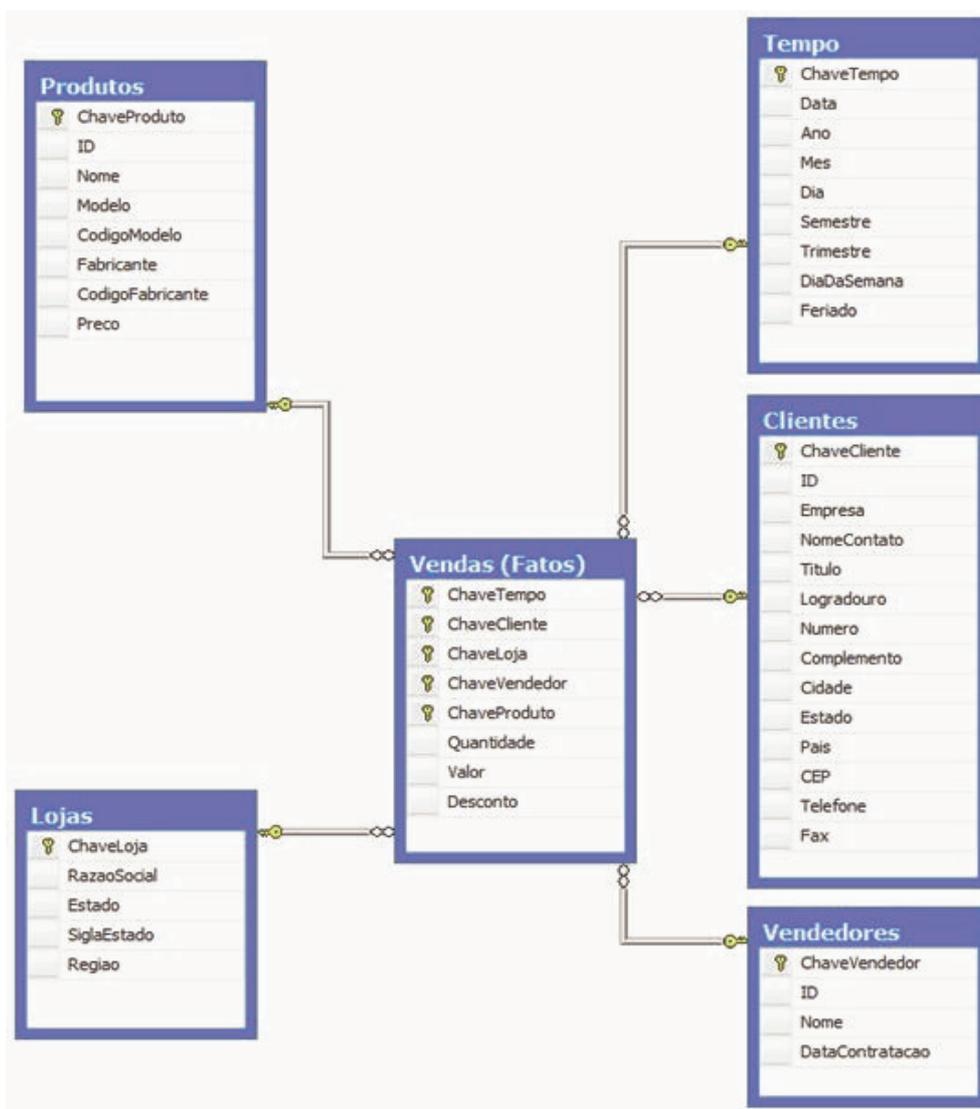


Figura 3.8 – Exemplo de um Modelo Multidimensional implementado com o Esquema Estrela (Extraído de [10])

A vantagem de se utilizar o Esquema Estrela está ligada ao fato de ser obter maior eficiência no momento de consultar os dados, pois não é necessário fazer várias junções (join) no Data Warehouse, visto que as tabelas estão desnormalizadas [2] [11].

A desvantagem é que com as tabelas desnormalizadas, os dados ficam redundantes [2].

3.1.5.3 Snowflake Schema (Esquema de Flocos de Neve)

No modelo Snowflake, o relacionamento entre as tabelas não ocorre apenas entre a tabela Fato e as dimensões. [11] Nesse modelo, pode ocorrer de existirem relacionamentos entre as dimensões. Isso ocorre pelo fato de neste modelo as tabelas serem normalizadas. [2] A figura 3.9 ilustra o modelo de dados que foi apresentado na figura 3.8, porém agora transformado para o Esquema de Flocos de Neve. É interessante analisar que as dimensões Produtos e Lojas foram normalizadas, surgindo novas tabelas no modelo.

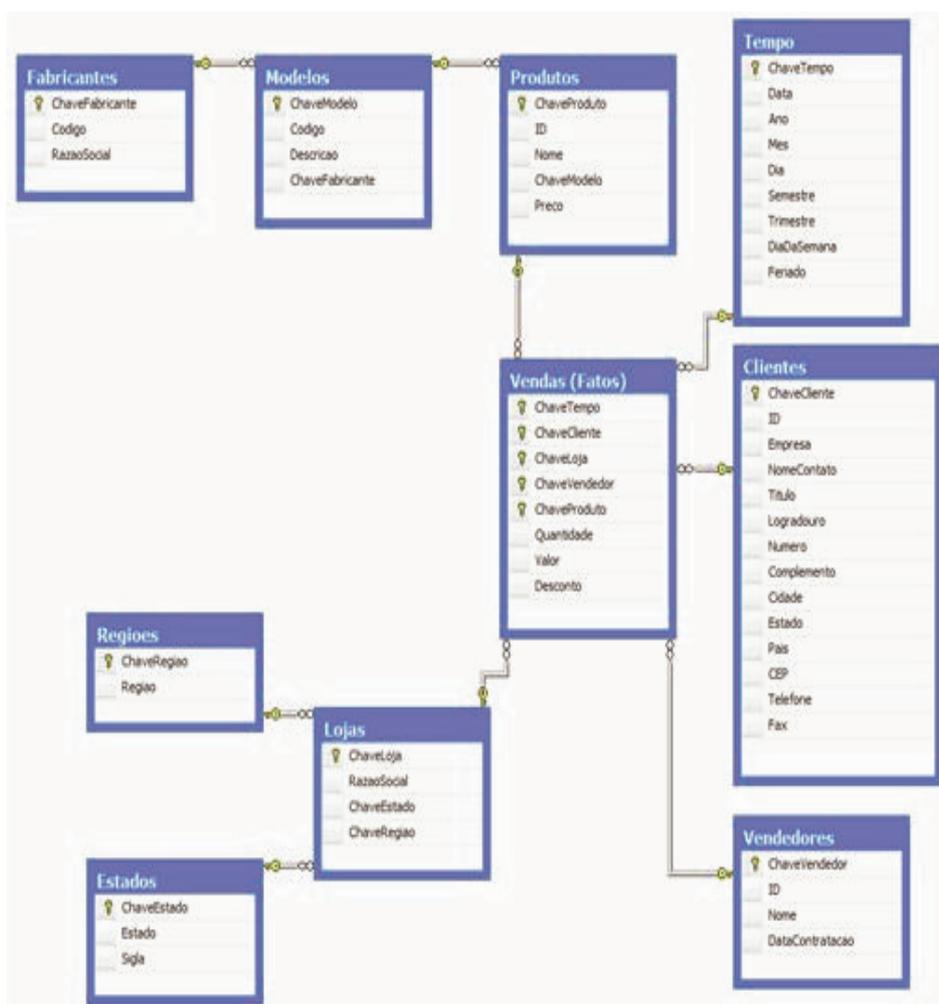


Figura 3.9 – Exemplo do Esquema Flocos de Neve.

Fonte: [10]

A desvantagem desse modelo, é que com a normalização das tabelas Dimensões, o número de tabelas aumenta no modelo [11]. Assim, o modelo fica mais difícil de compreender, impactando no entendimento dos dados por parte dos usuários. Além disso, torna-se necessário utilizar mais tabelas quando for feita uma consulta e como consequência disso, o número de junções aumenta, gerando uma queda no desempenho das consultas quando comparada ao Esquema Estrela [2] [11].

Esse nome Snowflake ou Flocos de Neve surgiu, pois com a normalização das tabelas elas se dividem em várias outras, que ficam organizadas em forma de camadas, que de certa forma lembram um floco de neve [2] [11].

3.1.5.4 Cubos

Matematicamente um cubo é constituído de três dimensões: altura, largura e profundidade. Entretanto, no modelo dimensional a metáfora do cubo pode assumir quantas dimensões forem necessárias para representar um fato [7]. Cada dimensão no cubo equivale ao armazenamento de um tipo específico de informações. Por exemplo, uma dimensão do cubo do tipo Produto, armazenando dados referentes aos produtos que uma empresa vende. O uso do cubo nesse contexto tem o objetivo de melhorar o processo de consulta dos dados para o usuário, ajuda na criação de relatórios, e também no momento de efetuar análises corporativas [9]. Por exemplo, um cubo pode conter informações das vendas de produtos em diversas regiões, como ilustra a figura 3.10.

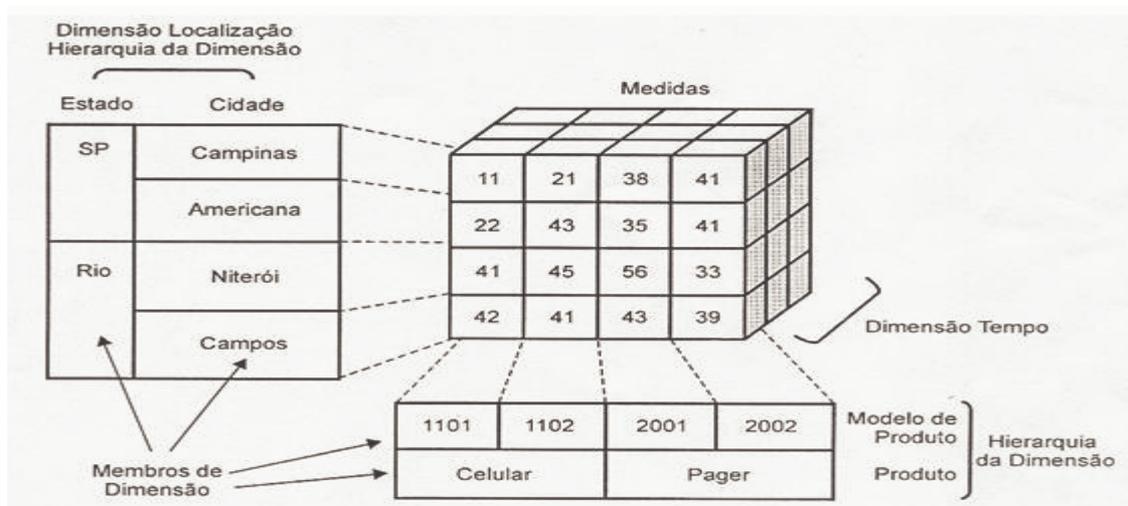


Figura 3.10 – Representação de um cubo com dimensões de Produto, Região e Tempo (Extraído de [7])

Essa figura mostra que é possível saber, através das coordenadas dimensionais do cubo, a quantidade de um produto específico que foi vendida em uma determinada cidade em um período de tempo.

Dentro desse contexto, surge o conceito de operadores dimensionais que estão diretamente ligados com o nível de granularidade das informações armazenadas em um Data Warehouse.

Os principais operadores dimensionais são:

- Drill-Down
- Drill-Up
- Drill-Across
- Drill-Through

Na figura 3.10, é possível observar que as informações relacionadas à dimensão Geográfica, estão estruturadas em dois níveis: Estado -> Cidade e as relacionadas à dimensão Produto também estão estruturadas em dois níveis segundo a hierarquia Produto -> Modelo do Produto.

O conceito de drill down está relacionado ao fato de sair de um nível mais alto na hierarquia e descer para um nível mais granular, ou seja, mais específico [2]. Por exemplo, analisar as vendas de um determinado produto por Estado e desejar uma análise mais específica por Cidade, estaria realizando um drill down.

Por outro lado, o conceito de drill up é o inverso, ou seja, caso esteja-se analisando as informações no nível de Modelo do Produto e deseje-se obter uma informação mais consolidada por Produto, basta realizar um drill up.

A figura 3.11 exemplifica esses dois conceitos.

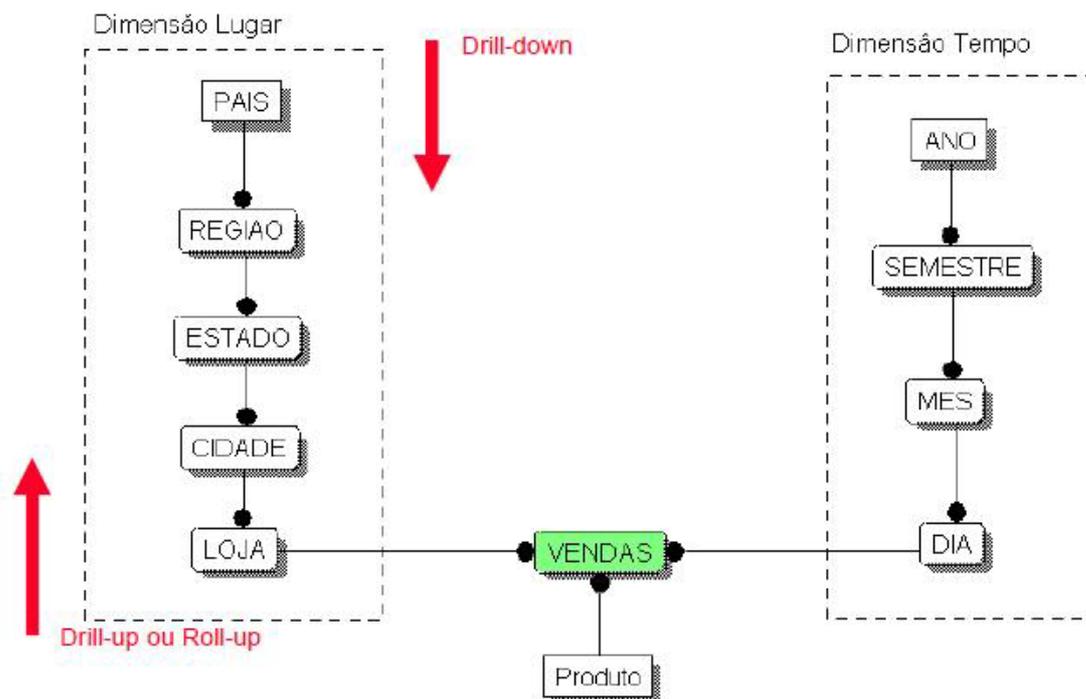


Figura 3.11 – Operadores Dimensionais Drill-down e Drill-Up (Adaptado de [8])

O drill-across tem a função de comparar dados em duas estruturas separadas que possuem dimensões em comum [2]. Suponha-se que para o nosso exemplo da figura 3.10 em vez de analisar-se o cubo com a idéia de Vendas de Produtos, que faria o papel da nossa tabela Fato, ele fosse analisado pelo fato Entrega de Produtos. Uma análise que poderia ser realizada utilizando esse novo fato Entrega, seria a quantidade de produtos que foi entregue em uma determinada região em um determinado mês. Caso fosse comparar informações referentes a essas duas entidades bastaria solicitar um drill-across, que na realidade não passa de um join dimensional, entre estruturas relacionadas [2]. A figura 3.12 a seguir, ilustra essa idéia.

E por fim, o drill-through está vinculado à necessidade que um usuário tem de realizar uma análise em um nível de granularidade muito baixo que uma tabela Fato não pode solicitar. Suponha que no exemplo ilustrado da figura 3.12, a tabela Fato sobre Vendas tenha armazenado dados num nível de produto por dia e por loja e suponha que o usuário deseja analisar essas informações por nível de Nota Fiscal. Porém, não é possível, pois o limite de detalhamento da análise é feita por Produto. Mas, o usuário sabe que essa informação mais granular pode ser encontrada em um ambiente fora do DW. O drill-through tem essa função de busca a informação além do nível de granularidade

existente na estrutura dimensional [2]. Nesse exemplo, o usuário estaria consultando as Notas Fiscais, que deram origem ao fato armazenado em cada linha da tabela Fato. É similar ao drill-down, só que buscando a informação mais detalhada em outra estrutura.

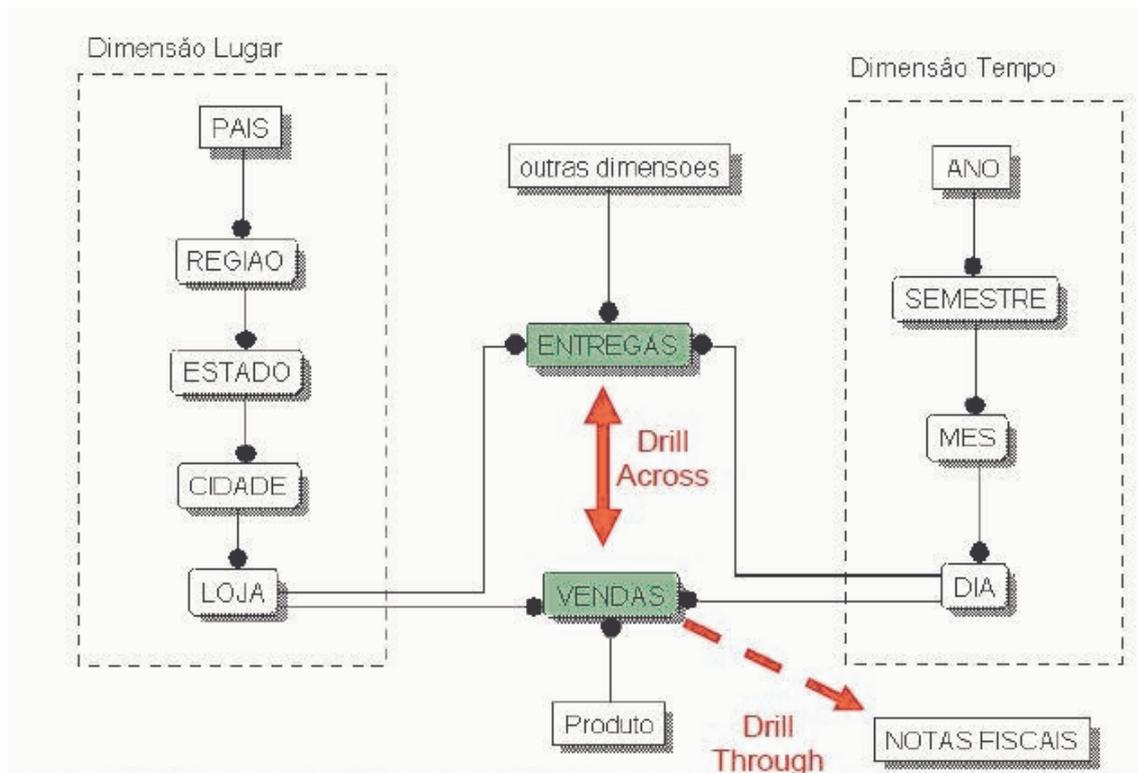


Figura 3.12– Operadores Dimensionais Drill Across e Drill Through (Adaptado de [8]).

3.1.6 Data Marts

O conceito de Data Marts foi criado com o objetivo de oferecer uma solução rápida e que atenda a grupos específicos de negócios ao invés de se concentrar nas necessidades da empresa inteira [1] [3].

Os Data Marts podem ser entendidos como uma espécie de Data Warehouse em menor escala e que possuem um escopo mais definido [2].

Esse conceito inventado por Ralph Kimball trouxe uma forma mais simples e incremental de se criar projetos de DW [2]. A figura 3.13 ilustra um esquema baseado na abordagem de Kimball.

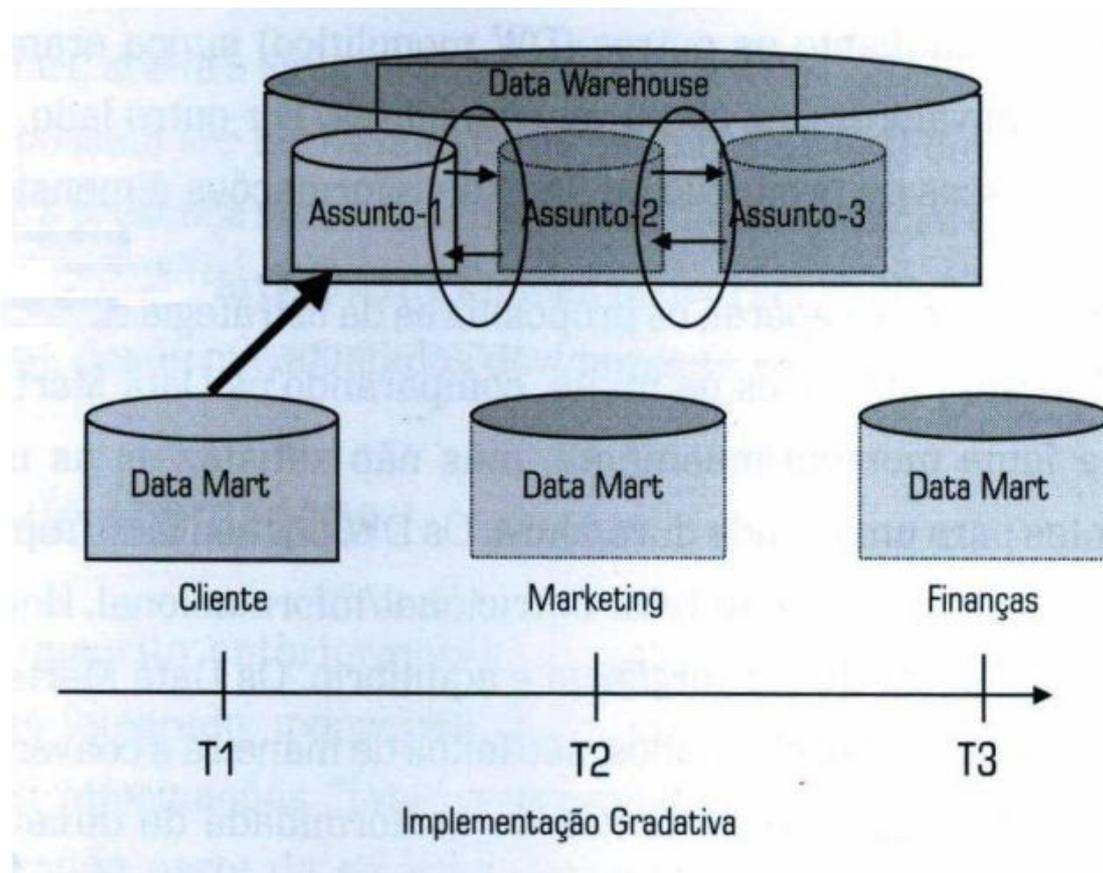


Figura 3.13– Estratégia de implementação gradativa de Data Marts (Adaptado de [2])

A abordagem segundo Bill Inmon, baseada em um estilo mais conservador de construção de Data Warehouse, não obteve muito sucesso quando comparada a metodologia de Kimball [2]. Pois para Inmon a ênfase está em construir um grande depósito central de dados, tratados e integrados [2]. Com base nisso, esse depósito seria dividido em depósitos menores, os Data Marts, que seriam por fim, separados por assuntos específicos, atendendo às necessidades de um departamento ou função empresarial em particular [3]. Essa técnica utilizada por Inmon, conhecida como top-down [10], demanda tempo, dinheiro e considerável esforço gerencial [1].

Muitas empresas têm optado por iniciar um projeto de Data Warehouse através da metodologia criada por Kimball. Uma vez que essas empresas são bem sucedidas, com a criação de alguns Data Marts, elas passam a integrar esses Marts, construindo um warehouse global [4]. Essa técnica de construção de Data Warehouses a partir de pequenos Data Marts ficou conhecida como bottom-up [10].

Com isso, as corporações tem se beneficiado com as vantagens que essa metodologia fornece. Um dos fatores positivos dessa abordagem foi a diminuição do

custo de implementação e manutenção dos sistemas de apoio à decisão [1]. Outra vantagem é que os Data Marts podem ser prototipados muito mais rápido, com alguns pilotos sendo construídos em 30 a 120 dias e sistemas completos sendo desenvolvidos entre 3 a 6 meses, sendo bem inferior à implantação de um DW utilizando a técnica de Inmon [1][3].

3.1.7 Metadados

Os metadados, que são dados sobre os dados, são essenciais em um projeto de Data Warehouse [1]. Como um projeto de DW possui muitas fases, documentar cada uma dessas fases é fundamental. Dentro de um repositório de metadados, devem existir informações sobre de onde vêm os dados que são carregados no DW, o formato dos dados, os processos de extração, limpeza e agrupamento dos dados, as mudanças sofridas pelos dados ao longo do tempo, a frequência de acesso aos dados, além das regras de negócio que envolve a empresa. [3]

Os metadados têm a função de orientar os usuários e analistas envolvidos no projeto, como por exemplo, identificar o local exato em que um dado se encontra. Os metadados possuem muito mais do que simplesmente informações de colunas e tabelas. Eles podem ser comparados a um dicionário, que contém informações que adicionem valor aos dados. [3]

A figura 3.14 foi utilizada por Carlos Barbieri em seu livro, com o intuito de exemplificar um modelo de metadados para um ambiente de Data Warehouse.

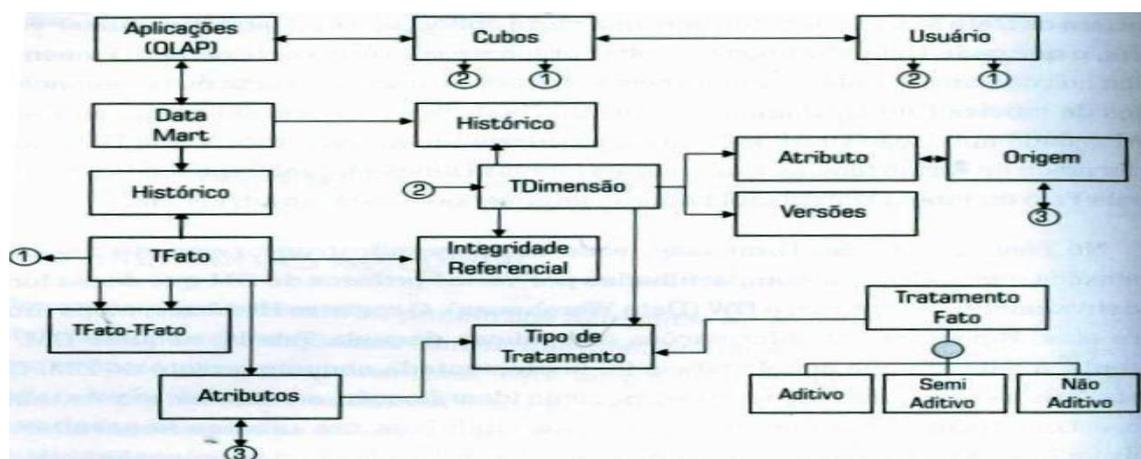


Figura 3.14 – Exemplo de um modelo de repositório de metadados (Adaptado de [2])

O repositório foi feito com base na modelo entidade-relacionamento, que é muito utilizado em projetos de banco de dados.

Num primeiro momento, foi criada uma entidade para armazenar os dados referentes a uma determinada aplicação OLAP, contendo informações como a data de implementação, o assunto-base relacionado à aplicação, etc. Num segundo momento, teríamos a entidade Data Mart (DM), que faria um relacionamento com as aplicações OLAP, indicando que certa aplicação OLAP pode ter vários Data Marts relacionados, e que um DM pode estar presente em várias aplicações [2].

O mesmo pode ser aplicado para a entidade Cubo, em que um cubo pode existir em várias aplicações e cada aplicação OLAP pode ter vários cubos. Tendo em mente, que um cubo é parte ou combinação de dados de tabelas Fato e Dimensão como visto no modelo [2].

Outro ponto abordado, é que cada Data Mart poderia ser formado a partir de várias tabelas Fatos e várias tabelas Dimensões, enquanto que cada tabela Fato e Dimensão poderia estar sendo utilizada em vários Data Marts [2].

A entidade Histórico representa as informações, por exemplo, da data em que uma dimensão foi utilizada em um determinado projeto de DM. Cada tabela teria guardadas as suas informações básicas, como nome, quantidade de atributos, etc. A entidade Integridade Referencial armazena as informações sobre a integridade referencial entre as tabelas Fato e Dimensão. A entidade Tratamento Fato armazenaria as informações referentes às formas de tratamento de uma métrica, como sendo aditiva (dados manipulados por soma, média, etc) numa dimensão, mas não aditiva em outra. É importante também ressaltar o armazenamento das informações relativas à extração, transformação e carga de cada atributo registrado dentro das tabelas Fato ou Dimensão. Para isso, foi criada a entidade Tipo de Tratamento. A entidade Origem poderia armazenar as informações de onde vêm os dados que são armazenados nas tabelas Dimensão e Fato. O auto-relacionamento da entidade TFato mostraria a junção possível entre elas, via comandos de drill across (espécie de join dimensional, entre estruturas relacionadas). A entidade Versões tem o papel de armazenar as versões de tabelas Dimensão, visando o controle das mudanças de estado dos atributos. Por fim, a entidade Usuário guarda os usuários que tem acesso aos cubos ou aqueles que possuem permissão para acessar tabelas Fato ou Dimensão, enriquecendo assim o modelo [2].

3.1.8 Tabelas Agregadas

Em um ambiente de Data Warehouse, tabelas agregadas são tabelas sumarizadas que armazenam dados em níveis mais elevados do que quando foram inicialmente capturados e armazenados [13]. O gerenciamento dessas tabelas não é uma tarefa trivial. Tabelas Agregadas são criadas quando há uma necessidade de melhorar o desempenho de consultas que são frequentemente solicitadas pelos usuários [12].

Suponha o projeto de DW abaixo com as seguintes tabelas Dimensão e Fato [2]:

TdLoja (chave-loja, nome-loja, endereço-loja, cidade, estado, cod-região)

TdProduto(chave-produto,marca,categoria,tipo-embalagem, departamento)

TdDia (chave-dia, mês, ano, período-fiscal, estação)

TfVendas (chave-loja, chave-produto, chave-dia, valor-vendido-real, custo-real, lucro, qtd-vendida)

Um bom exemplo para definição de tabelas agregadas, utilizando o projeto acima, é quando um usuário está interessado apenas nos resultados mensais. Ele deseja analisar as tendências das vendas por mês e comparar um mês do ano corrente com o mesmo mês do ano passado. Embora esse usuário possa executar sua consulta pela tabela TfVendas, as consultas serão feitas muito mais rapidamente, se puderem ser executadas em dados já agregados, através de uma tabela agregada sumarizada por mês [4]. Assim, poderíamos criar uma tabela contendo resumos dos registros da tabela fato TfVendas, mas apenas com um nível mensal de granularidade. O código SQL utilizado para criar essa tabela agregada poderia ser parecido com o seguinte, equivalente a agregação por loja, por mês, para todos os produtos [2].

```
INSERT INTO TAG_POR_LOJA_MÊS AS
SELECT NOME-LOJA,
      MÊS,
      SUM (VALOR-VENDIDO-REAL),
      SUM (CUSTO-REAL)
FROM TdLoja, TfVendas, TdDia
WHERE TdLoja.Chave-loja = TfVendas.Chave-loja and
```

TdDia.Chave-dia = TfVendas.Chave-dia

GROUPY BY NOME-LOJA, MÊS

Com essa tabela criada, os usuários que precisam de dados mensais podem consultar através dessa tabela agregada e com certeza o tempo de resposta dos dados será muito menor em comparação com a consulta à tabela TfVendas diretamente [4]. Porém um aspecto negativo da criação de tabelas agregadas está no grande aumento do espaço necessário para seu armazenamento.

O projeto e a criação das tabelas agregadas devem ter alguns cuidados operacionais [2]:

- ✓ As tabelas agregadas devem ter um modelo separado, com base nas tabelas Fato e nas tabelas Dimensão que compõem o modelo. Isso evita contenções mútuas no momento da carga ou atualização dos dados.
- ✓ É aconselhável realizar a carga/atualização dessas tabelas agregadas na parte da noite, pois o volume geralmente é muito grande.

3.2 Ferramentas de ETL

3.2.1 Conceito

O processo de ETL (Extract, Transform and Load) tem o objetivo de extrair, transformar e carregar os dados de fontes múltiplas e complexas para dentro do Data Warehouse [14]. Essas fontes podem ser os sistemas operacionais da empresa, arquivos textos e outras estruturas que são utilizadas para armazenar informações em uma organização.

A etapa de ETL é uma das fases mais críticas e demoradas em um projeto de Data Warehouse [1].

Esse processo é muito trabalhoso, complexo e necessita de muito cuidado para ser executado de forma adequada e correta, pois, caso uma informação seja carregada de forma errônea, isso trará consequências imprevisíveis para a organização [1] [14].

3.2.2 Extração

A extração dos dados tem a função de retirar os dados de um sistema de origem de maneira eficiente e carregar esses dados em um ambiente de Data Warehouse [4]. Isso nem sempre é tão fácil quanto parece à primeira vista e a complexidade começa a surgir

quando entra-se um pouco mais a fundo nos sistemas de origem. Em alguns casos é necessário selecionar vários campos de uma base de dados operacional para compor um campo no ambiente de DW [4]. Geralmente as tecnologias utilizadas nos ambientes operacionais são diferentes dos ambientes de Data Warehouse, tendo que adaptá-los para que possa ocorrer a extração dos dados de um ambiente para o outro [1]. Geralmente os dados estão em formatos diferentes dentro desses ambientes. Um exemplo seria a de um campo data de um sistema operacional que é do tipo DD/MM/YYYY, e deve ser passado para o outro ambiente no formato ano e mês como YYYYMM [1].

A figura 3.15 ilustra a fase de extração dos dados de múltiplas fontes realizada pelo processo de ETL e sendo gravado em um banco de dados de destino que no nosso caso seria o Data Warehouse [4].

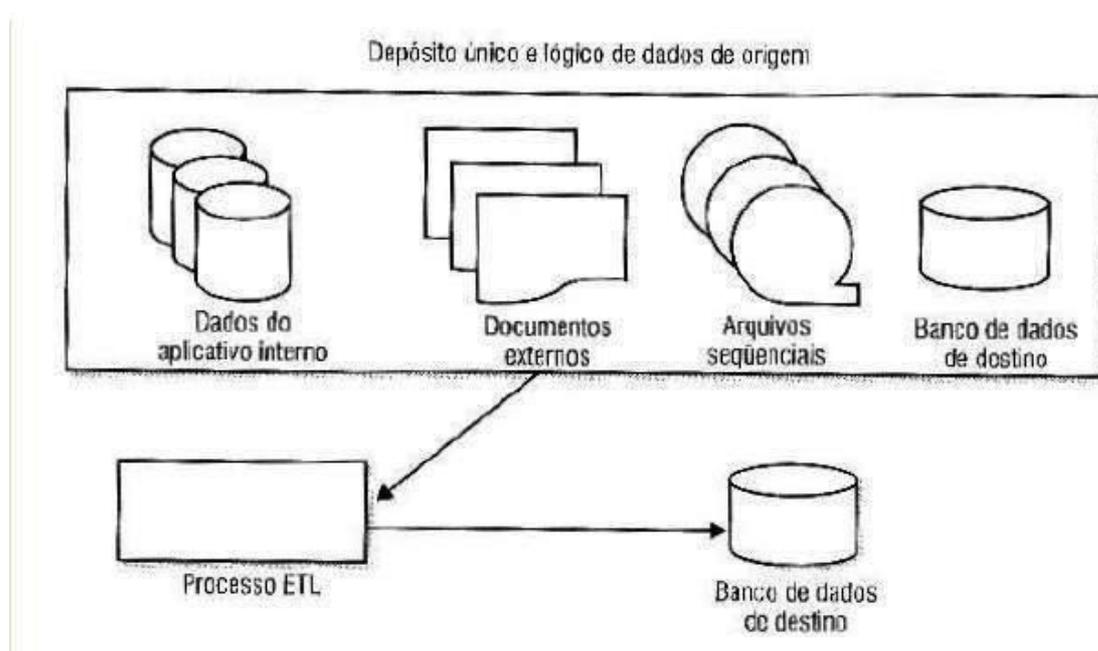


Figura 3.15 – Extração dos dados de múltiplas fontes (Adaptado de [4])

3.2.3 Transformação

O fase de transformação dos dados é um momento em que os dados são manipulados, partindo de como eles eram antes vindos dos sistemas fonte sendo transformados em outra forma para ser utilizada no Data Warehouse ou Data Mart, de maneira que melhore ou simplifique seu significado [4]. Dessa forma, os dados são transformados em informações estratégicas.

É neste momento que ocorre a correção de erros de digitação, a descoberta de violações de integridade, a substituição de caracteres desconhecidos, a padronização de abreviações, etc. [14].

Um bom exemplo, que demonstra a importância dessa fase, seria o de uma empresa que vende cartões de créditos, em que vendedor está mais preocupado em vender o produto (cartão) do que com a qualidade dos dados que ele está inserindo para documentar as informações do cliente. Então, se por acaso o cliente não tiver o número do CPF no momento do cadastro, o vendedor simplesmente digita um número qualquer que o sistema aceite como, por exemplo, 999999999-99. Agora imagine, caso uma consulta venha ser feita por um executivo desse banco, para verificar quais são os seus maiores clientes e aparecer em primeiro lugar o cliente cujo número do CPF é igual a 999999999-99? Por isso que é muito importante existir essa fase de transformação e limpeza dos dados dentro do ETL, pois ela garante a consistência das informações [1].

Segundo Kimball, as características mais relevantes para garantir a qualidade dos dados são [14]:

- Unicidade: Evitando assim duplicações de informação.
- Precisão: Os dados não podem perder suas características originais assim que são carregados para o DW.
- Completude, não gerando dados parciais de todo o conjunto relevante às análises.
- Consistência, ou seja, os fatos devem apresentar consistência com as dimensões que os compõem.

Desta forma, quando o usuário for analisar os dados dentro do Data Warehouse, ele encontrará informações limpas e precisas que o ajudarão no processo de tomada de decisões.

3.2.4 Carga

Após as fases de extração e transformação dos dados, a etapa de carga dos dados pode ser planejada. É nesse momento, que é definida a ordem de execução das cargas. Essa ordem de execução depende de diversos fatores que incluem a programação de execução de processos externos, as expectativas do usuário com relação à disponibilidade dos dados, e a execução de processos internos [4].

Num primeiro momento, é preciso analisar quando os processos externos começam a executar e quando eles terminam para que a partir daí os dados de origem estejam disponíveis para que os processos de ETL que extraem informações desses sistemas possam iniciar [4]. Após o término das cargas dos sistemas de origem, precisamos verificar a necessidade que os usuários possuem de analisar as informações [4], ou seja, com que frequência eles desejam ter essas informações atualizadas. É necessário rodar as cargas diariamente, semanalmente, mensalmente? Uma vez que as necessidades do usuário tenham sido satisfeitas, é preciso verificar a execução dos processos internos. Isso envolve as dependências existentes entre esses processos, como no caso de certas tabelas precisarem ser carregadas antes de outras de modo que a integridade referencial seja garantida [4].

Feita essa análise, um módulo escalonadora (scheduler), disponível nas ferramentas de ETL, tem a função de gerenciar todo o processamento das cargas, que foram programadas por um analista de acordo com as premissas vistas acima.

3.3 Ferramentas OLAP

3.3.1 Conceito

As empresas começaram a utilizar bancos de dados relacionais pela primeira vez com o intuito de ajudar no processo de produção dos dados [4]. A solução encontrada foi utilizar as ferramentas OLTP (Online Transaction Processing – Processamento de transações on-line) para executar seus sistemas operacionais. À medida que o volume de dados que atende aos usuários cresceu, a complexidade de análise dessas informações por parte dos usuários também aumentou [4]. Com isso, as ferramentas relacionais padrão existentes não conseguiam mais atender satisfatoriamente aos pedidos solicitados pelos usuários. Essa limitação impulsionou a criação de novas ferramentas.

As ferramentas OLAP, surgiram como uma resposta para suprir essas necessidades existentes. O termo “OLAP” que traduzindo para o português quer dizer Processamento Analítico On-line foi inventado por E.F.Codd em 1992, quando em seu artigo publicou 12 regras que definem uma ferramenta OLAP [1]. A figura 3.16 a seguir cita essas doze regras.

Características das Ferramentas OLAP
1. Visão conceitual multi-dimensional
2. Transparência
3. Acessibilidade
4. Informações de performance consistente
5. Arquitetura cliente-servidor
6. Dimensionalidade genérica
7. Manipulação de matrizes dinamicamente
8. Suporte a multi-usuários
9. Operações ilimitadas em referências cruzadas
10. Manipulação de dados intuitivamente
11. Consultas flexíveis
12. Níveis de dimensões e agregações ilimitadas

Figura 3.16– Características OLAP (Extraído de [1])

O OLAP proporciona aos usuários a capacidade de ter idéias sobre os dados, o que anteriormente não era possível, proporcionando modos de visualização rápidos, coerentes, fáceis de usar e interativos para uma variedade de informações. Além de ser possível analisar os dados de diferentes ângulos, como nunca visto antes, fornecendo o conhecimento necessário para tomar decisões seguras [4].

3.3.2 Tipos de Ferramentas

A ferramenta OLAP é caracterizada pela análise multidimensional dinâmica dos dados, permitindo consultas ad-hoc, realização de slice-and-dice, realização de drill down/up, geração de queries pelos usuários, etc. [1].

- Consultas ad-hoc: São consultas feitas pelos usuários de acordo com as análises que eles desejam fazer, cruzando diversas informações através do conjunto de dimensões disponíveis no sistema unidas em forma de cubos, que proporcionam informações de uma forma nunca vista e de métodos que os levam a encontrar o que procuram [1].
- Slice-and-dice: Ela é de suma importância para os usuários. É uma técnica criada que possibilita analisar as informações de diferentes formas, o que não era possível de se analisar antes da criação do DW e das ferramentas OLAP [1].
- Drill down/up: Ele permite que você analise as informações por diferentes níveis de granularidade. É possível utilizando o comando drill down analisar os dados em um nível bem detalhado, como por exemplo, uma análise diária das vendas de um produto. Já a utilização do drill up permite que a informação

seja analisada por um prisma mais consolidado, por exemplo, as vendas anuais de um produto [1] [2].

- Geração de Queries: Essa técnica exige que o usuário final tenha um conhecimento mínimo de informática. Ela é fornecida de uma maneira simples e fácil, como por exemplo, utilizando uma linguagem SQL.

Alguns termos muito utilizados na arquitetura OLAP são conhecidos. Dentre eles podemos citar: DOLAP, ROLAP, MOLAP, HOLAP. O que há de diferente em cada uma dessas arquiteturas é a forma como os dados são armazenados fisicamente. [4]

No DOLAP, é criado um conjunto de dados multidimensionais, podendo ser representado por um cubo de dados, quando uma consulta SQL é feita por um usuário. Quando isso ocorre, o sistema busca essas informações no servidor e as transfere para o desktop [1].

A vantagem dessa arquitetura é que há uma redução do tráfego de dados na rede, além do servidor de banco de dados não ficar sobrecarregado, visto que todo processamento OLAP, acontece na máquina do cliente [1]. Com isso, ela também fornece portabilidade para os usuários, que não podem estar a todo instante no escritório sem acesso direto ao servidor, e que precisam das informações atualizadas. Esse acesso é feito através dos aparelhos portáteis utilizado pelos usuários [4].

Uma desvantagem está no fato de que o volume de informações a ser acessado não pode ser muito grande, caso contrário o processamento pode demorar muito tempo [1].

O ROLAP, conhecido como Relational On line Analytical Processing, funciona de maneira diferente da realizada pela arquitetura DOLAP. Neste caso, quando uma consulta é solicitada ao servidor de banco de dados relacional pelo usuário, o conjunto de dados que será retornado dessa consulta permanece no servidor [1].

A vantagem dessa arquitetura está no fato de que é possível analisar enormes volumes de dados. Um ponto negativo é que, como os dados são acessados diretamente no servidor, se existir uma quantidade muito grande de usuários acessando ao mesmo tempo as informações no servidor, a performance no banco pode cair, além de até mesmo poder congelar o sistema [1].

Já a arquitetura MOLAP, conhecida como Multidimensional On line Analytical Processing, possui propriedades especiais de armazenamento dos dados de forma multidimensional para que possam ser visualizados de diferentes aspectos pelos usuários. O acesso ocorre diretamente no banco, onde o usuário, com os dados geralmente na formado de cubo, pode fazer análises, manipular as informações e criar novos cubos. Isso traz diversos benefícios como um alto grau de performance e a possibilidade de se dividir os dados em conjuntos menores. Quanto menores os conjuntos, mais rápido será o tempo de processamento. Por exemplo, uma aplicação financeira pode ser uma forte candidata a utilizar arquitetura MOLAP, pois cada departamento normalmente divide os conjuntos de dados. Como os conjuntos de dados são pequenos, é possível executar os dados rapidamente nas estruturas multidimensionais, caso queira atualizar o modelo. [4]

A última arquitetura, que muitos dizem ser uma mistura das arquiteturas MOLAP e ROLAP, é conhecida como HOLAP (Híbrido On line Analytical Processing) ou simplesmente processamento híbrido [1]. Nessa arquitetura as estruturas relacionais são geralmente utilizadas para os dados de maior granularidade e as estruturas dimensionais são utilizadas para armazenar os dados de menor granularidade [1]. Essa arquitetura é beneficiada pela alta performance que a arquitetura MOLAP proporciona, além da escalabilidade vinda da arquitetura ROLAP.

A aplicação de ferramentas OLAP é fundamental para as empresas atualmente, pelo fato delas proporcionarem um bom gerenciamento, trazendo produtividade para as empresas e beneficiando na tomada de decisões e nas análises feitas pelos gerentes [9].

4 PROCESSO DE MIGRAÇÃO

4.1 Introdução

Este capítulo aborda na prática como ocorre o funcionamento de um processo de Business Intelligence. Para isso, adaptou-se uma aplicação feita em Java já existente que aborda um determinado assunto dentro da minha empresa. Essa adaptação foi necessária pelo fato do programa em Java não conseguir dar o suporte necessário para os usuários, devido ao crescente volume de informações que a aplicação estava processando. Como as ferramentas de ETL possuem essa característica de manipular grandes volumes de dados, achou-se melhor migrar essa aplicação Java para programas de ETL.

O assunto abordado para a criação dessa aplicação tem a função de atender uma gama de usuários da área de marketing, com o objetivo de identificar os clientes que aderiram ao Plano de Marketing oferecido pela companhia. Como a empresa é voltada para o ramo de distribuição de combustíveis, os clientes são os postos de gasolina espalhados pelo Brasil. Essa adesão proporciona aos clientes todas as vantagens que o Plano de Marketing possui, possibilitando a esses clientes receberem treinamentos de capacitação de seus funcionários e gerentes para uma melhor administração do posto, além de poderem participar de todas as promoções feitas pela companhia.

Com isso, essa aplicação feita em Java disponibiliza via intranet uma interface em que os usuários fazem pesquisas com o intuito de saber a situação dos clientes quanto à sua posição em relação à adesão do Plano de Marketing, semelhante à figura 4.1 abaixo:

The screenshot shows the 'Plano de Marketing 2009' application interface. On the left is a vertical menu with items: Cargas, Manter Pontos de Vendas, Cobrança, Relatórios, and Sair. The main area is titled 'Arquivo de Saída' and contains a search form. The form has the following fields and options:

- PY Corporativo:** Text input field.
- Nome do Posto:** Text input field.
- Empresa:** Dropdown menu with '... Todas ...' selected.
- GC:** Dropdown menu with '... Todas ...' selected.
- GV:** Dropdown menu with '... Todas ...' selected.
- ZV:** Dropdown menu with '... Todas ...' selected.

Below the search fields are five checkboxes, all of which are checked:

- Saídas
- Não Apto
- Apto
- Termo Gerado
- Termo Aceito

At the bottom right of the form area is a button labeled 'Exportar Arquivo de Saída'.

Figura 4.1– Layout da aplicação Java

Foram criadas cinco situações em que o cliente pode se encontrar quanto a sua participação no Plano de Marketing. São elas:

1. **Saídas:** Clientes que cancelaram sua participação.
2. **Não Apto:** Clientes não participantes.
3. **Apto:** Clientes Aptos, mas assessor não realizou negociação.
4. **Termo Gerado:** Clientes com Termos de Aceite pendente de assinatura.
5. **Termo Aceito:** Clientes participantes com termos já assinados.

Com esta interface, o usuário pode realizar diversos tipos de pesquisas, verificando a situação de seus clientes através da digitação do código do Cliente (PV Corporativo) e do nome do Posto que deseja analisar, além de poder realizar uma consulta de forma macro, buscando todos os clientes que estão localizados em uma determinada Zona de Venda (ZV) ou que pertençam a uma Gerência de Venda (GV) específica. Além disso, o usuário pode realizar uma pesquisa pela situação em que os clientes se encontram em um determinado instante. Por exemplo, ele pode obter a listagem dos clientes que não estão participando do plano, marcando a opção Não Apto da interface. Após o usuário decidir a visão que ele deseja obter dos clientes, ele clica no botão Exportar Arquivo de Saída e a aplicação gera um relatório no formato de planilha em Excel com os dados dos clientes, como ilustra a figura 4.2 abaixo:

The screenshot shows the 'Plano de Marketing 2009' web interface. On the left is a navigation menu with options: Cargas, Manter Pontos de Vendas, Cobrança, Relatórios, and Sair. The main area is titled 'Arquivo de Saída' and contains a search form with the following fields: 'PV Corporativo' (text input with '14943'), 'Nome do Posto' (text input), 'Empresa' (dropdown menu with 'Todas'), 'GC' (dropdown menu with 'Todas'), 'GV' (dropdown menu with 'Todas'), and 'ZV' (dropdown menu with 'Todas'). Below these fields are five checkboxes, all of which are checked: 'Saídas', 'Não Apto', 'Apto', 'Termo Gerado', and 'Termo Aceito'. At the bottom right of the form area is a button labeled 'Exportar Arquivo de Saída', which is highlighted with a red arrow. Below the button is a small Excel icon and the text 'Clique para fazer o download da Planilha...'.

Figura 4.2– Geração do Relatório em Excel

Após o clique para fazer o download da planilha, ela pode ser visualizada pelo usuário como mostra a figura 4.3.

	D	G	J	K	S	T	W	X	Y
	Nome Posto	Pessoa de Contato	Endereço	Bairro	Participação	Valor Arrecadado	Termo Adesão	Data Emissão Termo	Hora Emissão Termo
1									
2	A Abast Love Story Ltda	Claudio/Antonio	Est Da Barra Da Tijuca, 2770	Itanhangá	SIM	6800,00	ACEITO	05/03/2008	06:38:34
3	A P Casa E Pesca Ltda	Fabiano	Av Semambetiba, 4270	Barra Da Tijuca	SIM	5300,00	ACEITO	04/03/2008	12:48:14
4	A P De Gas E Serv L & Z Ltda	Jaqueline / Riu	Est Bandeirantes, 23679	Vargem Grande	SIM	8000,00	ACEITO	25/02/2008	18:27:34
5	A P De Serv Marisol Ltda	Flavia / Pires	Av Semambetiba, 16500	Recreio Bandeirant	SIM	6000,00	ACEITO	05/03/2008	06:44:38
6	A P Map Barra Ltda	Marcelo/Alessan	Av Das Americas, 1510 - Posto Ipiranga	Barra Da Tijuca	SIM	6000,00	ACEITO	25/02/2008	18:31:18
7	A P Sao Furtuoso Ltda	Franklin	Estrada Dos Bandeirantes, 130	Taquara	NÃO		PENDENTE		
8	A P Senna Ltda	Marcos Fontes	Av Ayrton Senna, S/N - Cebolaol/Canteir	Barra Da Tijuca	SIM	9800,00	ACEITO	03/03/2008	23:00:04
9	A P Xavante Ltda	Antonio/Malve	Est De Jacarepagua, 7744	Jacarepagua	SIM	3000,00	ACEITO	05/03/2008	06:40:28
10	Auto Posto Valdevez Ltda	Ribeiro/Rogério	Estr Pau Ferro, 1128	Jacarepagua	SIM	6000,00	ACEITO	04/03/2008	12:41:19
11	C A Ancora Ltda	Alberto / Franc	R Cel Tedim, 40	Jacarepagua	SIM	8000,00	ACEITO	23/02/2008	18:33:36
12	C A Esquilos Ltda	Alberto	Est De Jacarepagua, 4735	Anil	SIM	8900,00	ACEITO	29/02/2008	18:30:48
13	Dist Way De Combs Ltda	Hernany	Av Das Americas, 2100	Barra Da Tijuca	SIM	6800,00	ACEITO	27/02/2008	07:48:44
14	Mattar E Mattar A P Ltda	Hernany	Av Das Americas, 10495	Barra Da Tijuca	SIM	9000,00	ACEITO	25/02/2008	18:28:23
15	Mill Barragas	Marco	Av Ayrton Sena, 9201	Barra Da Tijuca	SIM		ACEITO	27/02/2008	07:49:16
16	Mill Diamond	Marcoio	Av Das Americas, 3201	Barra Da Tijuca	SIM		ACEITO	27/02/2008	07:49:37
17	P Barra Linda Ltda	Claudio	Av Das Americas, 7000	Barra Da Tijuca	SIM	8000,00	ACEITO	25/02/2008	18:34:57
18	P Gas E Bar Garoa Ltda	Antonio	Est Da Gavea, 636	Sao Conrado	SIM	8000,00	ACEITO	28/02/2008	18:06:07
19	P Gastotal Ltda	Silvio / S	Av Geremario Dantas, 88	Jacarepagua	SIM	6000,00	ACEITO	03/03/2008	23:00:53
20	P Mega Verao Ltda	Antonio	Av Armando Lombardi, 1100	Barra Da Tijuca	NÃO		PENDENTE		
21	P Via 11 Ltda	Claudio	Av Ayrton Sena, 2400	Barra Da Tijuca	SIM	5500,00	ACEITO	25/02/2008	18:33:01
22	Pologaz 10 P De Abast Ltda	Francisco / Joa	Po Prof Camisao, 5	Jacarepagua	SIM	2600,00	ACEITO	29/02/2008	18:36:15
23	De Comb Mega Engenho Nov	Vania	R Sousa Barros, 636	Engenho Novo	NÃO		PENDENTE		
24	Ventura Com E Serv Ltda	Eduardo / Ana	R Luis Carlos Prestes, S/N	Barra Da Tijuca	SIM	7900,00	ACEITO	29/02/2008	18:35:12

Figura 4.3– Relatório gerado pela aplicação.

Como dito anteriormente, o volume de dados começou a aumentar significativamente devido ao aumento do número de usuários acessando essa aplicação. Isso fez com que o programa passasse a gerar arquivos na faixa de Megabytes (MB), muito superior ao que era gerado no início quando a aplicação foi criada, com arquivos em torno de Quilobytes (KB). Com isso, a performance da aplicação diminuiu, o que contribuiu ainda mais para que se buscasse uma solução que pudesse atender a demanda.

4.2 Fases do Processo

Esta seção aborda de forma sucinta as fases da migração da aplicação Java para o ambiente de Business Intelligence e de suas ferramentas.

4.3 Primeira fase

A primeira fase se dá através da construção de um modelo de dados multidimensional que se adapte a esse novo ambiente do BI, como mostra a figura 4.4 abaixo.

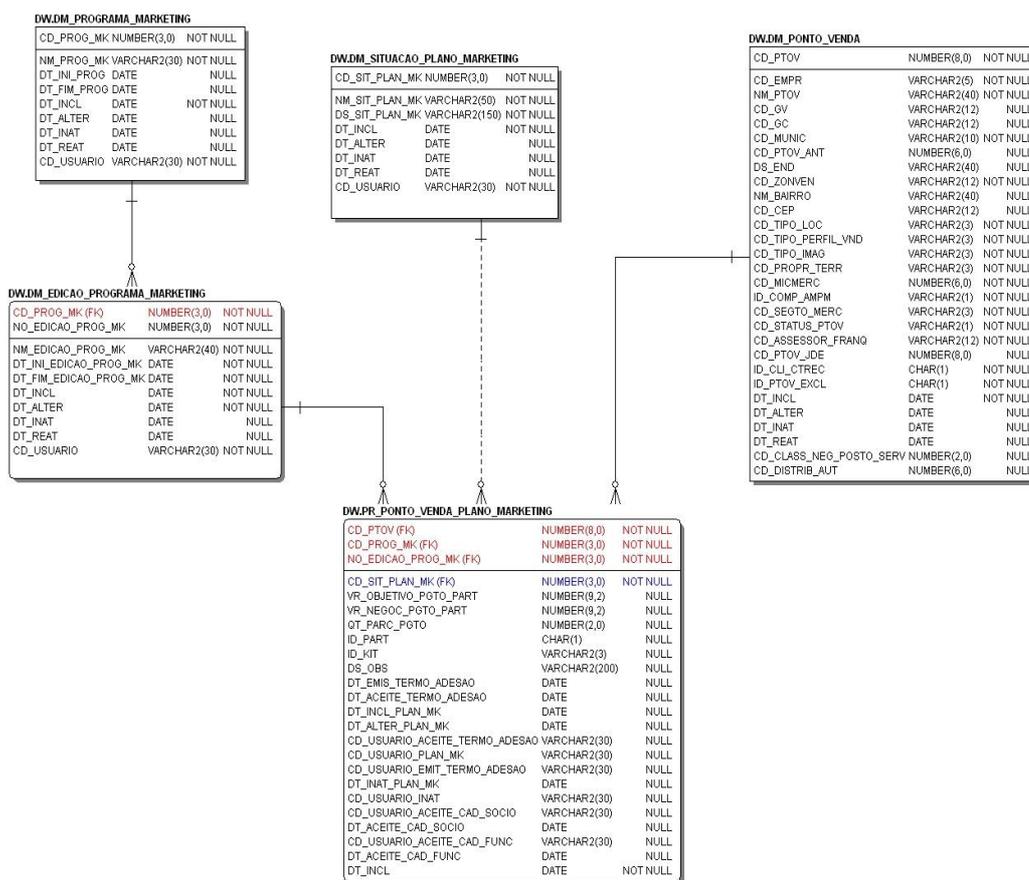


Figura 4.4 – Modelagem utilizada no processo de migração.

Esse modelo foi construído utilizando a ferramenta de modelagem de dados E/R Studio. Ele é formado por cinco tabelas sendo quatro tabelas Dimensão e uma tabela Fato. No Apêndice A, seguem as definições detalhadas de cada tabela.

4.4 Segunda fase

A segunda fase tem o objetivo de criar essas tabelas no banco de dados e popular a dimensão DM_SITUACAO_PLANO_MARKETING.

É importante observar que algumas tabelas foram reaproveitadas de outros modelos de dados que já existiam no Data Warehouse da empresa. Foi criada especificamente para esse projeto uma tabela Dimensão e a tabela Fato que no modelo acima são respectivamente a DM_SITUACAO_PLANO_MARKETING e a PR_PONTO_VENDA_PLANO_MARKETING. Os comandos de DDL, de criação dessas tabelas no banco podem ser visualizados no Apêndice B. Esses comandos são executados pelos DBAs que tem os privilégios suficientes para realizar essa tarefa.

4.5 Terceira fase

É nesta terceira fase que acontece a criação dos programas de ETL. Para isso foi utilizada a query abaixo, que era utilizada na aplicação Java para buscar os dados no banco, gerando a planilha para o usuário. Essa query pode ser dividida em cinco partes, mostradas nas figuras 4.5 a 4.9 a seguir.

```

select
  p.CD_PTOV,
  p.CD_PTOV_ANT,
  p.CD_GC,
  p.NM_GC,
  ppi.CD_COMP_JDE,
  p.CD_GV,
  p.NM_GV,
  p.CD_ZONVEN,
  p.NM_ZONVEN,
  p.CD_EMPR,
  p.NM_PTOV,
  p.CD_PESS,
  p.CD_DV,
  p.DS_END,
  p.NM_BAIRRO,
  p.CD_CEP,
  p.NM_MUNIC,
  p.SG_UF,
  p.CD_INSCRE,
  p.ID_RODOREDE,
  p.ID_CARTAO_IPIR,
  p.DS_TPNEG,
  p.CD_TIPO_PERFIL_VND,
  decode(p.ID_AMPM, 'S', 'AM/PM')
  ampm,
  decode(p.ID_AMPM_EXPRESS, 'S',
'AM/PM Express') express,
  decode(p.ID_LUB, 'S', 'Jet Oil') lub,
  decode(p.ID_LUB_SERVICE, 'S', 'Lub
Service') service,
  pp.VR_OBJETIVO_PGTO_PART,
  pp.VR_NEGOC_PGTO_PART,
  pp.DT_EMIS_TERMO_ADESAO,
  pp.DT_ACEITE_TERMO_ADESAO,
  pp.DT_INAT,
  pp.QT_PARC_PGTO,
  pp.DS_OBS,
  pp.ID_KIT,
  f.cd_ddd1,
  f.no_tel_direto1
from V_PTOV_REV_IPIRANGA      p,
     V_COMPONENTE_IPIRANGA    ppi,
     PARTICIPANTE_ADESAO_UNICA pp,
     V_FUNCIONARIO            f,
     V_HIERARQUIA_VENDA_IPIRANGA h
where f.CD_USUARIO(+) = h.CD_USUARIO
and p.cd_comp = ppi.cd_comp
and h.CD_ZONVEN = p.cd_zonven
and pp.cd_prog_mk(+) = 14
and pp.no_edicao_prog_mk(+) = 2
and p.CD_EMPR = ?
and p.CD_GC = ?
and p.CD_GV = ?
and p.CD_ZONVEN = ?
and pp.cd_ptov(+) = p.cd_ptov
and pp.cd_empr(+) = p.cd_empr
and pp.dt_inat is not null

UNION

```

Figura 4.5 – Seleciona os clientes que cancelaram sua participação

```

select
  p.CD_PTOV,
  p.CD_PTOV_ANT,
  p.CD_GC,
  p.NM_GC,
  ppi.CD_COMP_JDE,
  p.CD_GV,
  p.NM_GV,
  p.CD_ZONVEN,
  p.NM_ZONVEN,
  p.CD_EMPR,
  p.NM_PTOV,
  p.CD_PESS,
  p.CD_DV,
  p.DS_END,
  p.NM_BAIRRO,
  p.CD_CEP,
  p.NM_MUNIC,
  p.SG_UF,
  p.CD_INSCRE,
  p.ID_RODOREDE,
  p.ID_CARTAO_IPIR,
  p.DS_TPNEG,
  p.CD_TIPO_PERFIL_VND,
  decode(p.ID_AMPM, 'S', 'AM/PM')
  ampm,
  decode(p.ID_AMPM_EXPRESS, 'S',
'AM/PM Express') express,
  decode(p.ID_LUB, 'S', 'Jet Oil') lub,
  decode(p.ID_LUB_SERVICE, 'S', 'Lub
Service') service,
  pp.VR_OBJETIVO_PGTO_PART,
  pp.VR_NEGOC_PGTO_PART,
  pp.DT_EMIS_TERMOS_ADESAO,
  pp.DT_ACEITE_TERMOS_ADESAO,
  pp.DT_INAT,
  pp.QUANT_PARC_PGTO,
  pp.DS_OBS,
  pp.ID_KIT,
  f.cd_ddd1,
  f.no_tel_direto1
from V_PTOV_REV_IPIRANGA p,
     V_COMPONENTE_IPIRANGA ppi,
     PARTICIPANTE_ADESAO_UNICA pp,
     V_FUNCIONARIO f,
     V_HIERARQUIA_VENDA_IPIRANGA h
where f.CD_USUARIO(+) = h.CD_USUARIO
and p.cd_comp = ppi.cd_comp
and h.CD_ZONVEN = p.cd_zonven
and pp.cd_prog_mk(+) = 14
and pp.no_edicao_prog_mk(+) = 2
and p.CD_EMPR = ?
and p.CD_GC = ?
and p.CD_GV = ?
and p.CD_ZONVEN = ?
and pp.cd_ptov(+) = p.cd_ptov
and pp.cd_empr(+) = p.cd_empr
and pp.vr_objetivo_pgto_part is null
UNION

```

Figura 4.6 – Seleciona os clientes que não estão aptos para participarem

```

select
  p.CD_PTOV,
  p.CD_PTOV_ANT,
  p.CD_GC,
  p.NM_GC,
  ppi.CD_COMP_JDE,
  p.CD_GV,
  p.NM_GV,
  p.CD_ZONVEN,
  p.NM_ZONVEN,
  p.CD_EMPR,
  p.NM_PTOV,
  p.CD_PESS,
  p.CD_DV,
  p.DS_END,
  p.NM_BAIRRO,
  p.CD_CEP,
  p.NM_MUNIC,
  p.SG_UF,
  p.CD_INSCRE,
  p.ID_RODOREDE,
  p.ID_CARTAO_IPIR,
  p.DS_TPNEG,
  p.CD_TIPO_PERFIL_VND,
  decode(p.ID_AMPM, 'S', 'AM/PM')
  ampm,
  decode(p.ID_AMPM_EXPRESS, 'S',
'AM/PM Express') express,
  decode(p.ID_LUB, 'S', 'Jet Oil') lub,
  decode(p.ID_LUB_SERVICE, 'S', 'Lub
Service') service,
  pp.VR_OBJETIVO_PGTO_PART,
  pp.VR_NEGOC_PGTO_PART,
  pp.DT_EMIS_TERMOS_ADESAO,
  pp.DT_ACEITE_TERMOS_ADESAO,
  pp.DT_INAT,
  pp.QT_PARC_PGTO,
  pp.DS_OBS,
  pp.ID_KIT,
  f.cd_ddd1,
  f.no_tel_direto1
from V_PTOV_REV_IPIRANGA p,
     V_COMPONENTE_IPIRANGA ppi,
     PARTICIPANTE_ADESAO_UNICA pp,
     V_FUNCIONARIO f,
     V_HIERARQUIA_VENDA_IPIRANGA h
where f.CD_USUARIO(+) = h.CD_USUARIO
and p.cd_comp = ppi.cd_comp
and h.CD_ZONVEN = p.cd_zonven
and pp.cd_prog_mk(+) = 14
and pp.no_edicao_prog_mk(+) = 2
and p.CD_EMPR = ?
and p.CD_GC = ?
and p.CD_GV = ?
and p.CD_ZONVEN = ?
and pp.cd_ptov(+) = p.cd_ptov
and pp.cd_empr(+) = p.cd_empr
and pp.vr_objetivo_pgto_part is not null
and pp.dt_emis_termos_adesao is null
UNION

```

Figura 4.7 – Seleciona os clientes que estão aptos, porém o assessor ainda não realizou a negociação

```

select
  p.CD_PTOV,
  p.CD_PTOV_ANT,
  p.CD_GC,
  p.NM_GC,
  ppi.CD_COMP_JDE,
  p.CD_GV,
  p.NM_GV,
  p.CD_ZONVEN,
  p.NM_ZONVEN,
  p.CD_EMPR,
  p.NM_PTOV,
  p.CD_PESS,
  p.CD_DV,
  p.DS_END,
  p.NM_BAIRRO,
  p.CD_CEP,
  p.NM_MUNIC,
  p.SG_UF,
  p.CD_INSCRE,
  p.ID_RODOREDE,
  p.ID_CARTAO_IPIR,
  p.DS_TPNEG,
  p.CD_TIPO_PERFIL_VND,
  decode(p.ID_AMPM, 'S', 'AM/PM')
  ampm,
  decode(p.ID_AMPM_EXPRESS, 'S',
'AM/PM Express') express,
  decode(p.ID_LUB, 'S', 'Jet Oil') lub,
  decode(p.ID_LUB_SERVICE, 'S', 'Lub
Service') service,
  pp.VR_OBJETIVO_PGTO_PART,
  pp.VR_NEGOC_PGTO_PART,
  pp.DT_EMIS_TERMOS_ADESAO,
  pp.DT_ACEITE_TERMOS_ADESAO,
  pp.DT_INAT,
  pp.QT_PARC_PGTO,
  pp.DS_OBS,
  pp.ID_KIT,
  f.cd_ddd1,
  f.no_tel_direto1
from V_PTOV_REV_IPIRANGA p,
     V_COMPONENTE_IPIRANGA ppi,
     PARTICIPANTE_ADESAO_UNICA pp,
     V_FUNCIONARIO f,
     V_HIERARQUIA_VENDA_IPIRANGA h
where f.CD_USUARIO(+) = h.CD_USUARIO
and p.cd_comp = ppi.cd_comp
and h.CD_ZONVEN = p.cd_zonven
and pp.cd_prog_mk(+) = 14
and pp.no_edicao_prog_mk(+) = 2
and p.CD_EMPR = ?
and p.CD_GC = ?
and p.CD_GV = ?
and p.CD_ZONVEN = ?
and pp.cd_ptov(+) = p.cd_ptov
and pp.cd_empr(+) = p.cd_empr
and pp.dt_emis_termos_adesao is not null
and pp.dt_aceite_termos_adesao is null
UNION

```

Figura 4.8 – Seleciona os clientes que estão com o termo pendente de assinatura

```

select
  p.CD_PTOV,
  p.CD_PTOV_ANT,
  p.CD_GC,
  p.NM_GC,
  ppi.CD_COMP_JDE,
  p.CD_GV,
  p.NM_GV,
  p.CD_ZONVEN,
  p.NM_ZONVEN,
  p.CD_EMPR,
  p.NM_PTOV,
  p.CD_PESS,
  p.CD_DV,
  p.DS_END,
  p.NM_BAIRRO,
  p.CD_CEP,
  p.NM_MUNIC,
  p.SG_UF,
  p.CD_INSCRE,
  p.ID_RODOREDE,
  p.ID_CARTAO_IPIR,
  p.DS_TPNEG,
  p.CD_TIPO_PERFIL_VND,
  decode(p.ID_AMPM, 'S', 'AM/PM')
  ampm,
  decode(p.ID_AMPM_EXPRESS, 'S',
'AM/PM Express') express,
  decode(p.ID_LUB, 'S', 'Jet Oil') lub,
  decode(p.ID_LUB_SERVICE, 'S', 'Lub
Service') service,
  pp.VR_OBJETIVO_PGTO_PART,
  pp.VR_NEGOC_PGTO_PART,
  pp.DT_EMIS_TERMOS_ADESAO,
  pp.DT_ACEITE_TERMOS_ADESAO,
  pp.DT_INAT,
  pp.QT_PARC_PGTO,
  pp.DS_OBS,
  pp.ID_KIT,
  f.cd_ddd1,
  f.no_tel_direto1
from V_PTOV_REV_IPIRANGA p,
     V_COMPONENTE_IPIRANGA ppi,
     PARTICIPANTE_ADESAO_UNICA pp,
     V_FUNCIONARIO f,
     V_HIERARQUIA_VENDA_IPIRANGA h
where f.CD_USUARIO(+) = h.CD_USUARIO
and p.cd_comp = ppi.cd_comp
and h.CD_ZONVEN = p.cd_zonven
and pp.cd_prog_mk(+) = 14
and pp.no_edicao_prog_mk(+) = 2
and p.CD_EMPR = ?
and p.CD_GC = ?
and p.CD_GV = ?
and p.CD_ZONVEN = ?
and pp.cd_ptov(+) = p.cd_ptov
and pp.cd_empr(+) = p.cd_empr
and pp.dt_aceite_termos_adesao is not null
order by NM_PTOV

```

Figura 4.9 – Seleciona os clientes participantes com termos já assinados

Com a quebra dessa query em cinco partes, podemos criar um programa de ETL para cada uma dessas partes como mostra a figura 4.10.

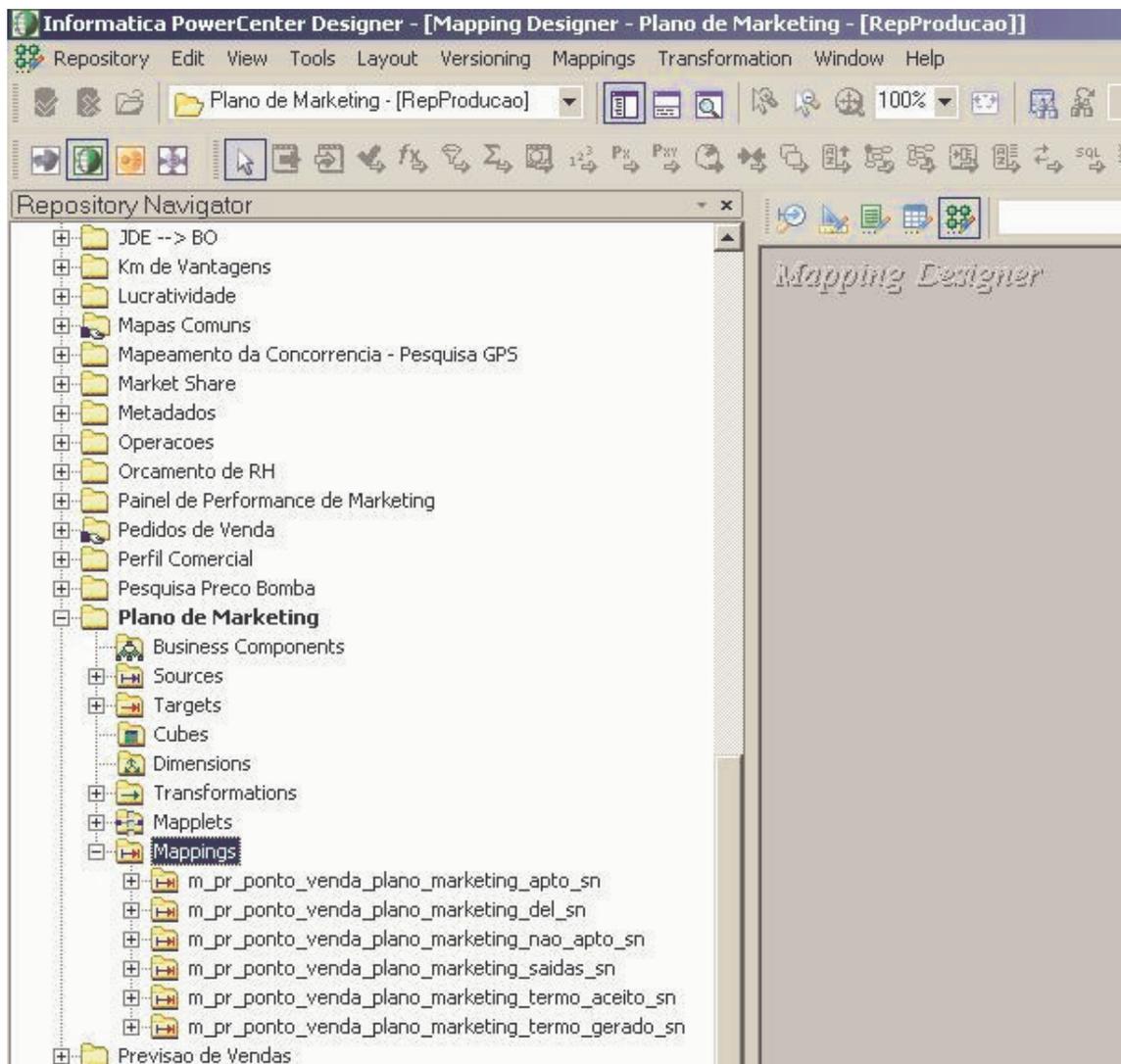


Figura 4.10 – Programas de ETL.

Foi utilizada a ferramenta *Informatica PowerCenter* para criar os programas de carga. Uma pasta com o nome de Plano de Marketing foi criada exclusivamente para esse projeto como pode ser visto na figura 4.10. Dentro dessa pasta além dos programas de ETL, temos armazenados todos os objetos utilizados para criação desses programas.

Cada ETL deve tratar de uma situação, implementada por uma regra de negócio específica. Cada programa de ETL possui como origem uma tabela chamada de PARTICIPANTE_ADESAO_UNICA que foi utilizada no SQL da aplicação Java. Essa tabela é utilizada para armazenar as informações dos clientes segundo a sua situação no

Plano de Marketing no ambiente de Sistemas de Negócios da empresa, onde a aplicação Java era executada.

O Fato PR_PONTO_VENDA_PLANO_MARKETING foi criado no ambiente de BI com base na tabela PARTICIPANTE_ADESAO_UNICA.

4.5.1 Fluxo de Deleção

O objetivo dessa carga é deletar os Pontos de Venda (Clientes) do Plano de Marketing para as duas últimas edições independente da situação. Isso ocorre para que não haja problemas de integridade no momento de executar as outras cargas. A figura 4.11 mostra o programa de ETL responsável por esse procedimento.

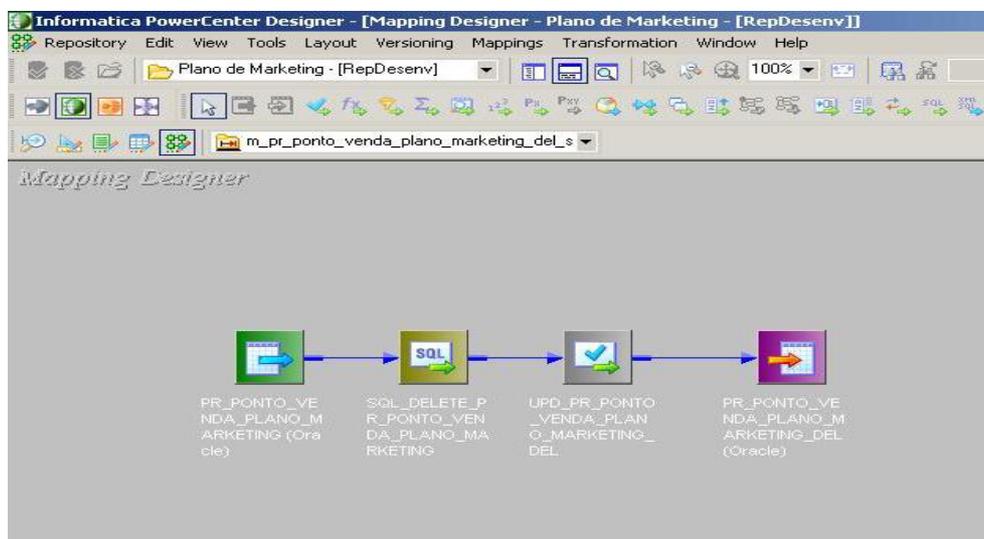


Figura 4.11 – Programa ETL – Fluxo de Deleção

A figura 4.12 que representa o primeiro programa ETL, possui quatro objetos diferentes que serão explicados abaixo:

Source Definition: é o primeiro objeto de cor verde que representa a origem dos dados.

Source Qualifier: é o segundo objeto de cor amarela que mostra o SQL que a carga está fazendo no banco de dados onde a tabela fonte se encontra.

Update Strategy: é o terceiro objeto de cor cinza que indica a operação que será executada no banco de destino. Ex: Insert, Update, Delete. Uma versão ampliada desse objetivo pode ser visto na figura 4.12.

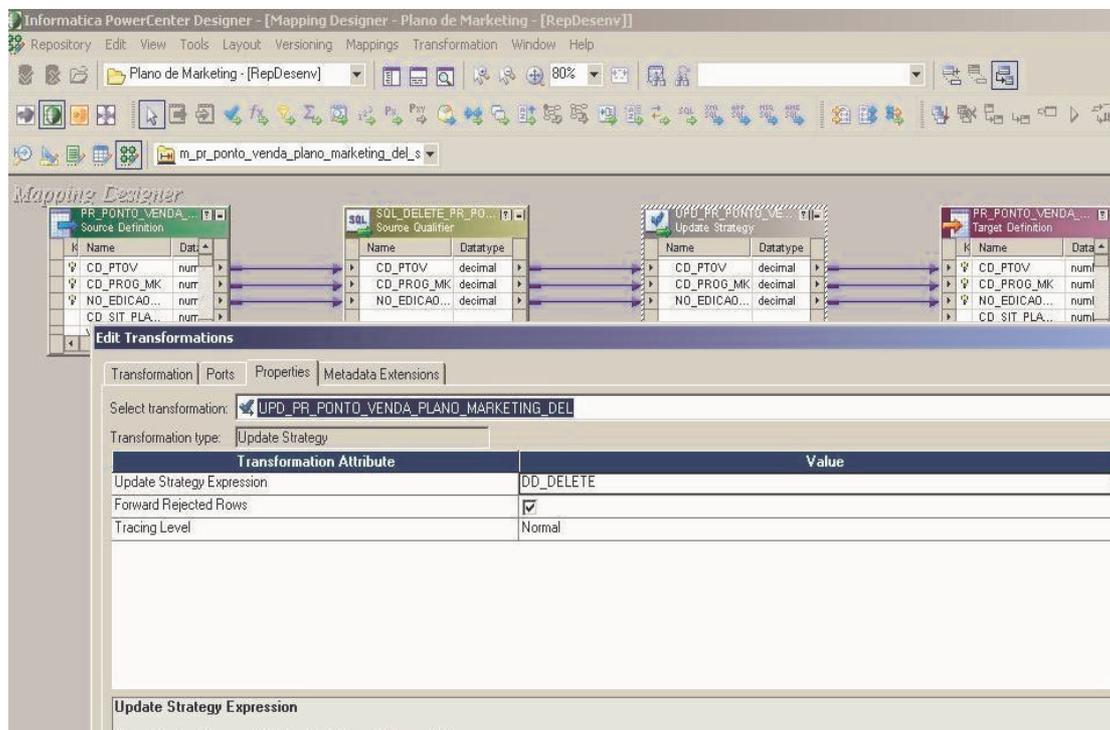


Figura 4.12 – Programa ETL – Update Strategy

No exemplo da figura 4.12, o objeto está executando uma operação de deleção na base de dados de destino. Isso é identificado pelo comando DD_DELETE gerado dentro do objeto na aba Properties.

Target Definition: é o último objeto de cor roxa que representa o destino dos dados. Ou seja, a tabela que estará armazenando os dados da carga.

Esses conceitos são importantes, pois ao longo dessa seção, eles serão vistos com frequência nas imagens dos processos de carga.

A seguir mostramos a imagem do mesmo programa da figura 4.13 de forma ampliada.

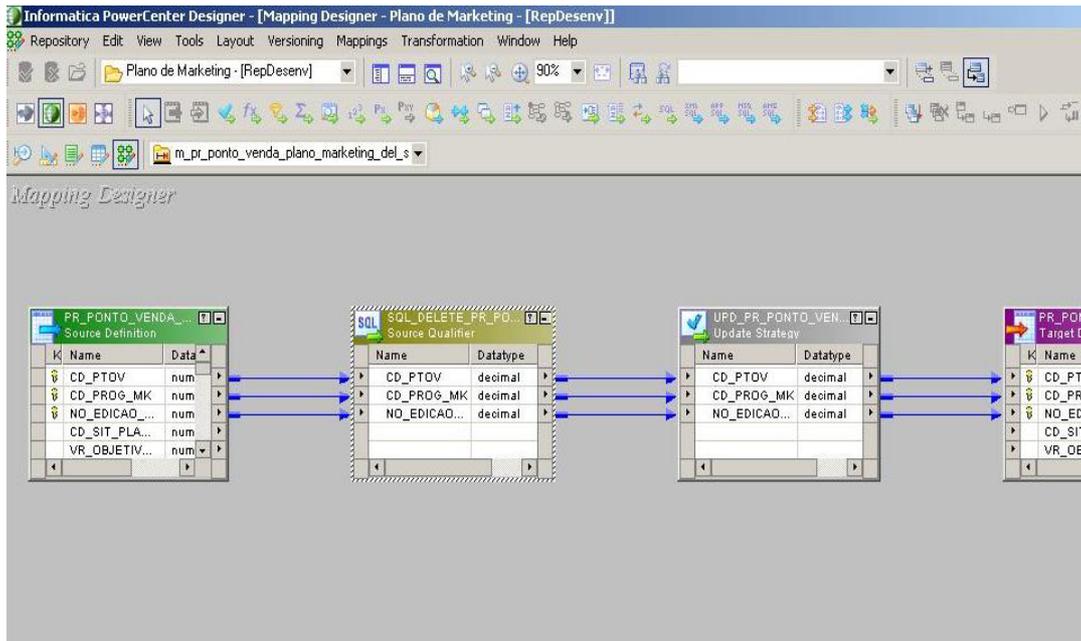


Figura 4.13 – Programa ETL – Fluxo de Delete Ampliado

Além disso, quando se clica no Source Qualifier, a seguinte janela é aberta como mostra a figura 4.14.

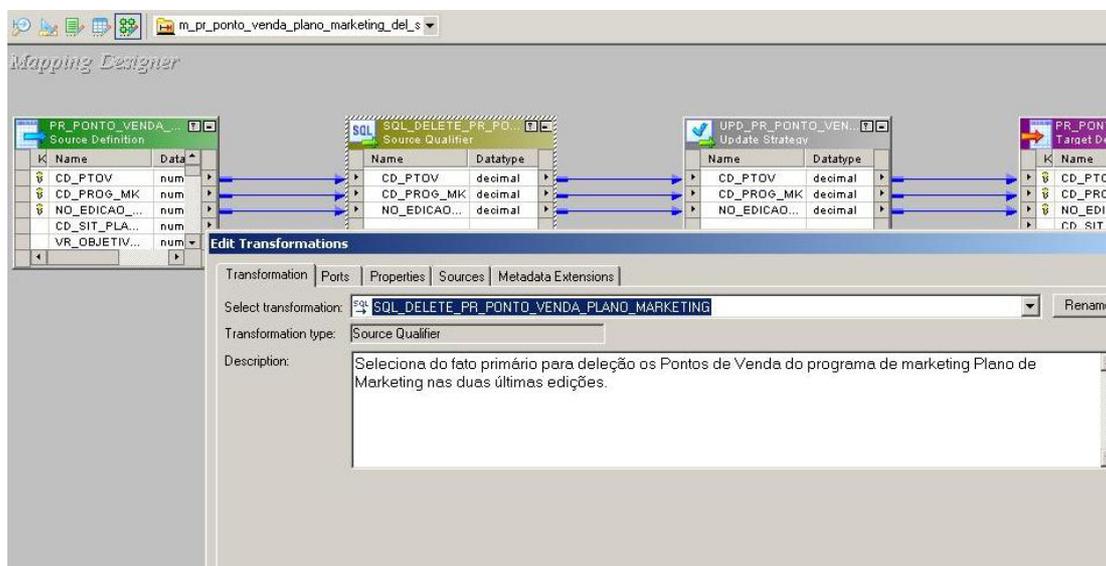


Figura 4.14 – Programa ETL – Fluxo de Delete - Edit

Para visualizar o SQL que o mapa está fazendo, basta clicar na aba Properties como mostra a figura 4.14 e clicar em Sql Query como mostra a figura 4.15 a seguir.

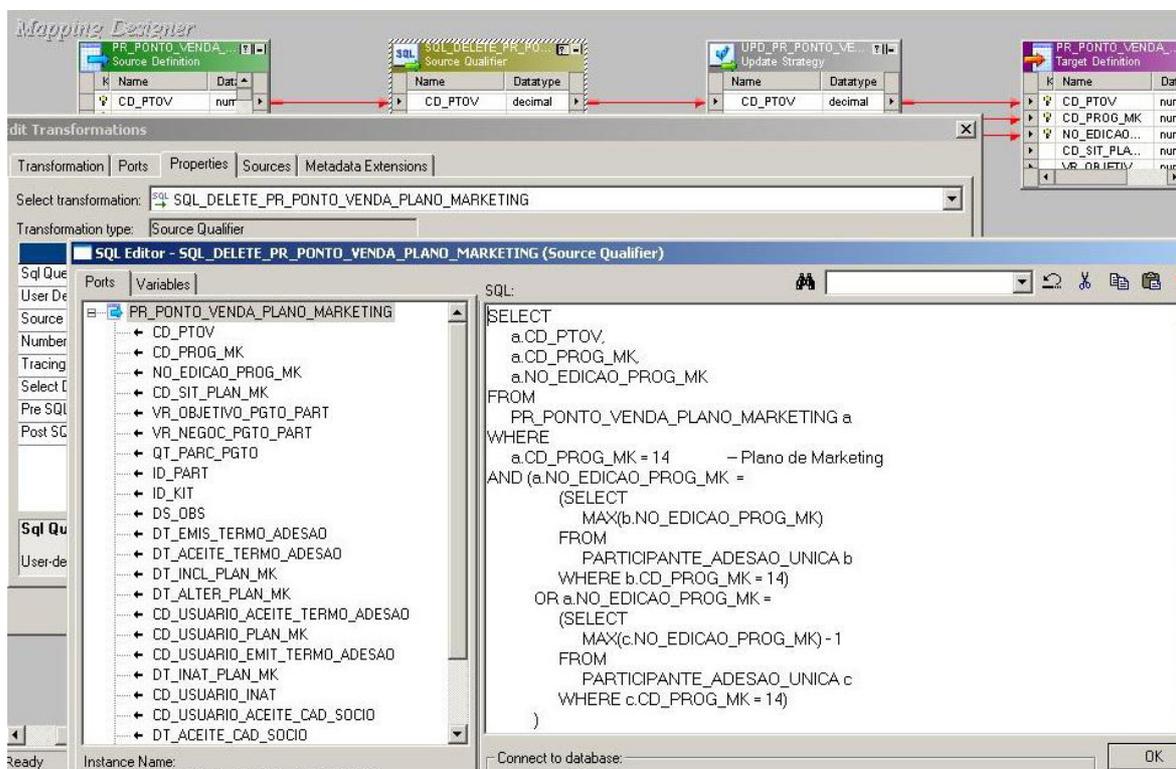


Figura 4.15 – Programa ETL – Fluxo de Delete – SQL

4.5.2 Fluxo das Saídas

A segunda carga é responsável pelo processo que carrega os clientes que aderiram ao Plano de Marketing, porém cancelaram sua participação nas duas últimas edições. Dentro do Source Qualifier, a query busca somente os clientes que estão com a Data de Inativação (DT_INAT) diferente de nulo. Ou seja, os clientes que cancelaram sua participação. A figura 4.16 a seguir detalha esse processo.

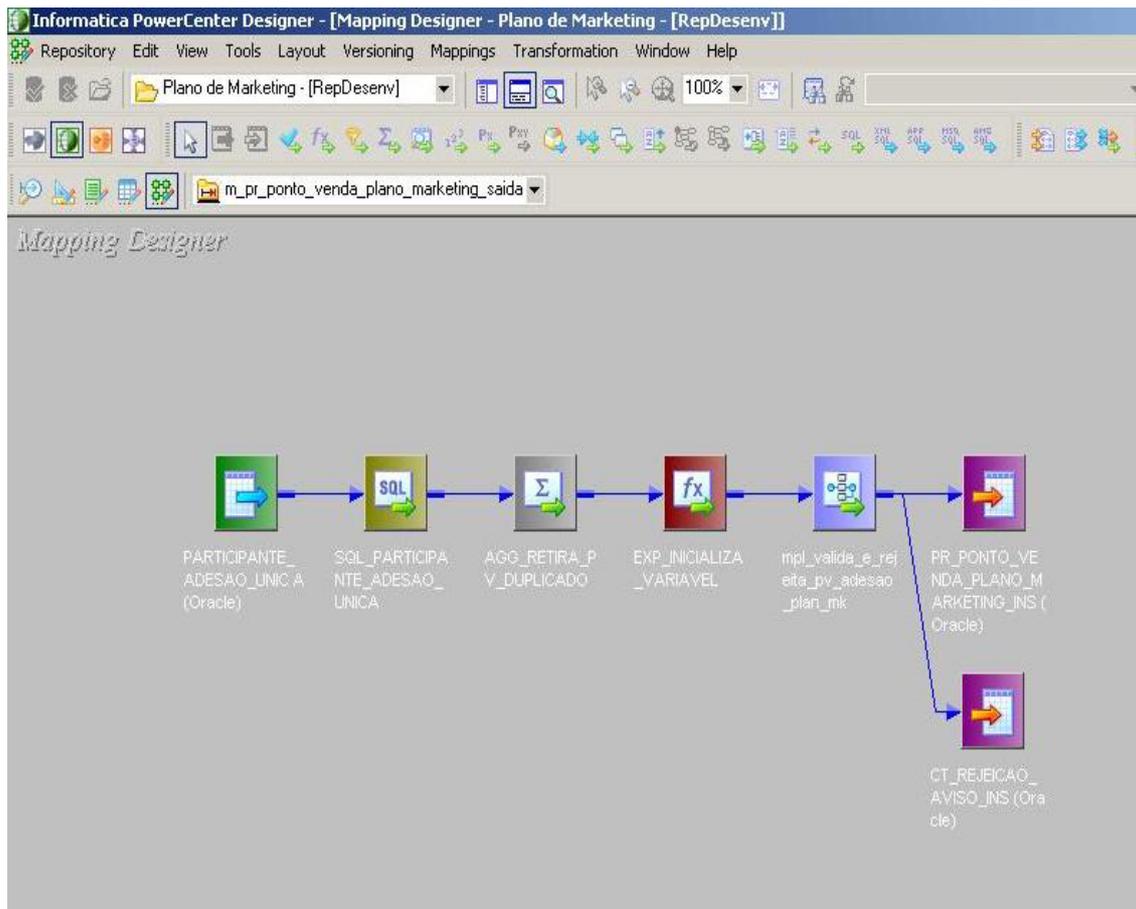


Figura 4.16 – Programa ETL – Fluxo de Saída

Os objetos envolvidos nesse processo são:

Source Definition: representado pela tabela participante adesão única (PARTICIPANTE_ADESAO_UNICA).

Source Qualifier: possui o SQL representado pela query a seguir.

```

SELECT
    a.CD_PTOV,
    a.CD_PROG_MK,
    a.NO_EDICAO_PROG_MK,
    a.VR_OBJETIVO_PGTO_PART,
    a.VR_NEGOC_PGTO_PART,
    a.QT_PARC_PGTO,
    a.ID_PART,
    a.ID_KIT,
    a.DS_OBS,
    a.DT_EMIS_TERMOS_ADESAO,
    a.DT_ACEITE_TERMOS_ADESAO,
    a.DT_INCL,
    a.DT_ALTER,
    a.CD_USUARIO_ACEITE_TERMOS_ADESAO,
    a.CD_USUARIO,
    a.CD_USUARIO_EMIT_TERMOS_ADESAO,
    a.DT_INAT,
    a.CD_USUARIO_INAT,
    a.CD_USUARIO_ACEITE_CAD_SOCIO,
    a.CD_USUARIO_ACEITE_CAD_FUNC,
    a.DT_ACEITE_CAD_SOCIO,
    a.DT_ACEITE_CAD_FUNC
FROM
    PARTICIPANTE_ADESAO_UNICA a
WHERE
    a.DT_INAT IS NOT NULL -- Saídas
AND
    a.CD_PROG_MK = 14 -- Plano de Marketing
AND (a.NO_EDICAO_PROG_MK =
      (SELECT
MAX(b.NO_EDICAO_PROG_MK)
      FROM
PARTICIPANTE_ADESAO_UNICA b
      WHERE b.CD_PROG_MK = 14)
OR
    a.NO_EDICAO_PROG_MK =
      (SELECT
MAX(c.NO_EDICAO_PROG_MK) - 1
      FROM
PARTICIPANTE_ADESAO_UNICA c
      WHERE c.CD_PROG_MK = 14))

```

Aggregator: Neste caso tem a função de retirar os registros duplicados para um mesmo cliente.

Expression: é o quarto objeto de cor vermelha que realiza o tratamento dos campos que são necessários, cria novos campos, etc. A figura 4.17 mostra uma expression com mais detalhes. Note que foi criado um campo para a situação do Plano de Marketing no objeto com pode ser visto em vermelho. Cada programa possui o seu número relacionado à sua situação.

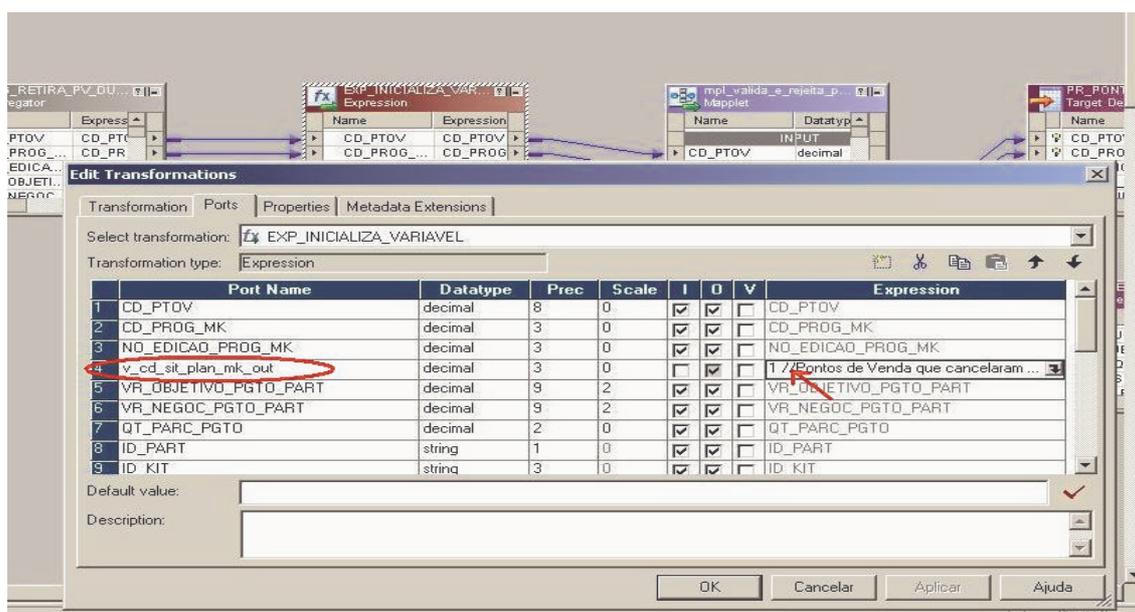


Figura 4.17 – Programa ETL - Expression

Mapplet: é um objeto que será reutilizado em diversos programas e tem a função de encapsular um trecho do fluxo. Será estudado com mais detalhes ao longo do trabalho.

Target Definition: Para esse mapa temos dois targets. O primeiro que recebe os dados válidos e o segundo que recebe os dados inválidos.

Na figura 4.18 é possível visualizar o processo ampliado.

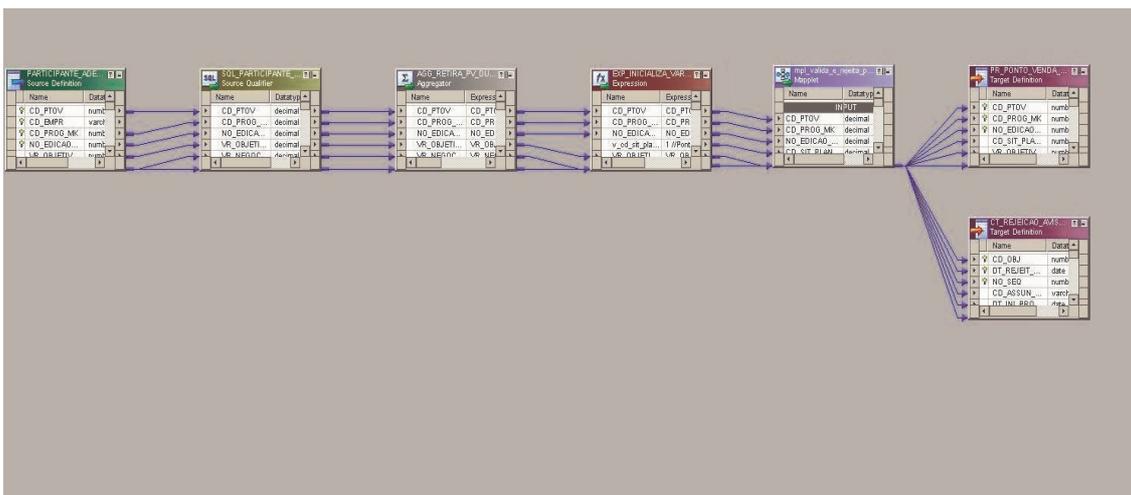


Figura 4.18 – Programa ETL – Fluxo de Saída – Ampliado

O Maplet deste mapa pode ser visualizado na figura 4.19 abaixo.

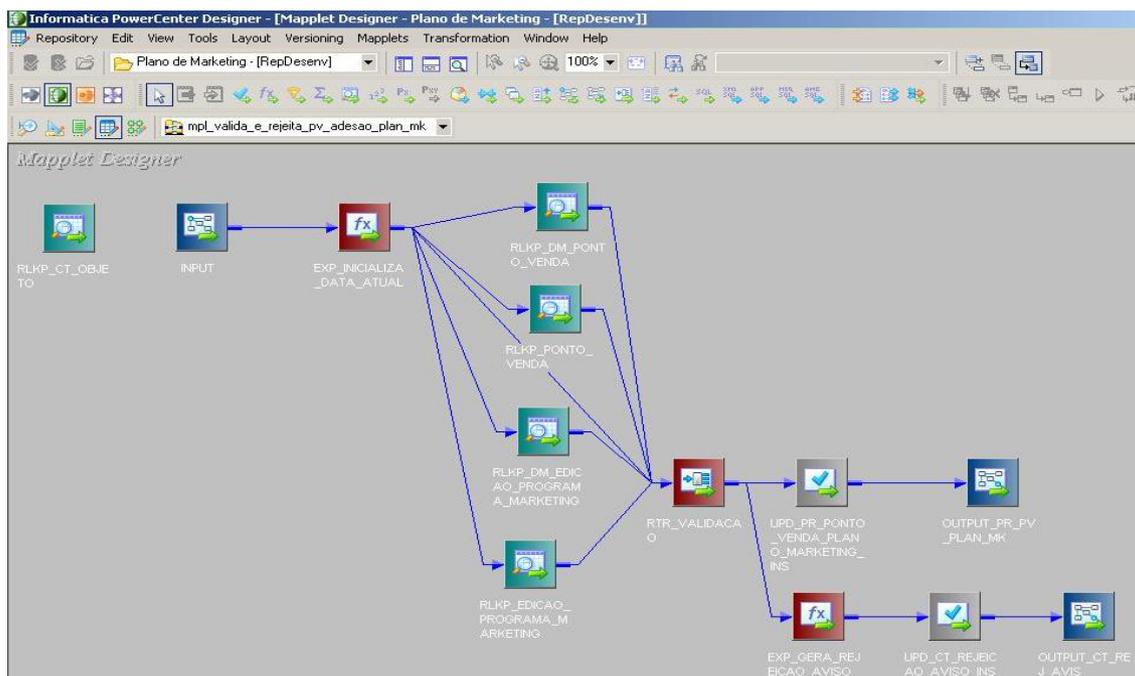


Figura 4.19 – Programa ETL – Fluxo dos Apts – Mapplet

Os objetos do Mapplet são:

Lookup: é a transformação que tem a função de procurar um determinado registro em uma tabela, através de um ou mais parâmetros de entrada. Pode ser identificado na figura 4.19 pelo objeto de cor verde. A Lookup pode ser melhor visualizada pela figura 4.20 a seguir.

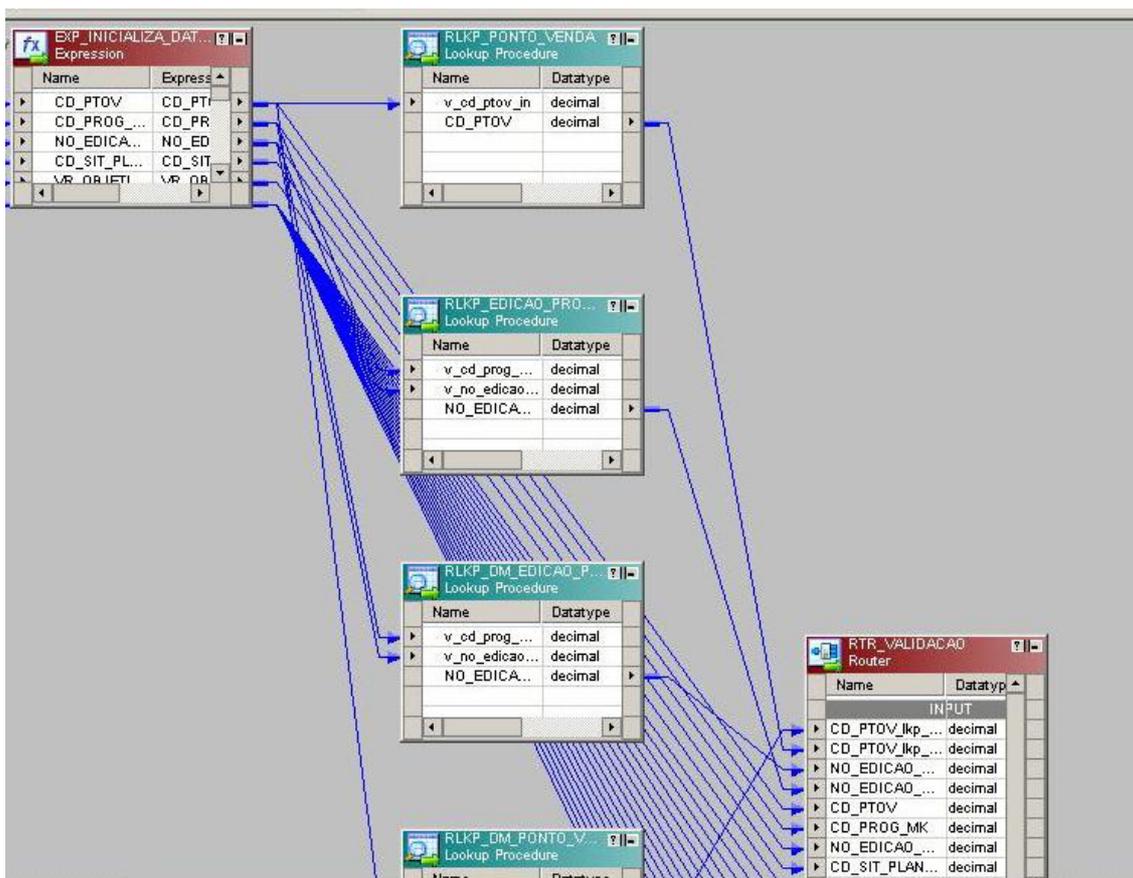


Figura 4.20 – Programa ETL – Lookup

Neste exemplo, a Lookup PONTO_VENDA tem a função de verificar se o Código do Cliente (CD_PTOV) existe no ambiente do BI para não ocorrer erro de integridade, no momento da inserção ou atualização dos dados.

Input: é a porta de entrada de todos os registros no Mapplet.

Output: é a porta de saída de todos os registros do Mapplet.

Router: tem o objetivo de dividir o fluxo em registros válidos e inválidos, segundo uma determinada condição. As figuras 4.21 e 4.22 a seguir, ilustram o Router.

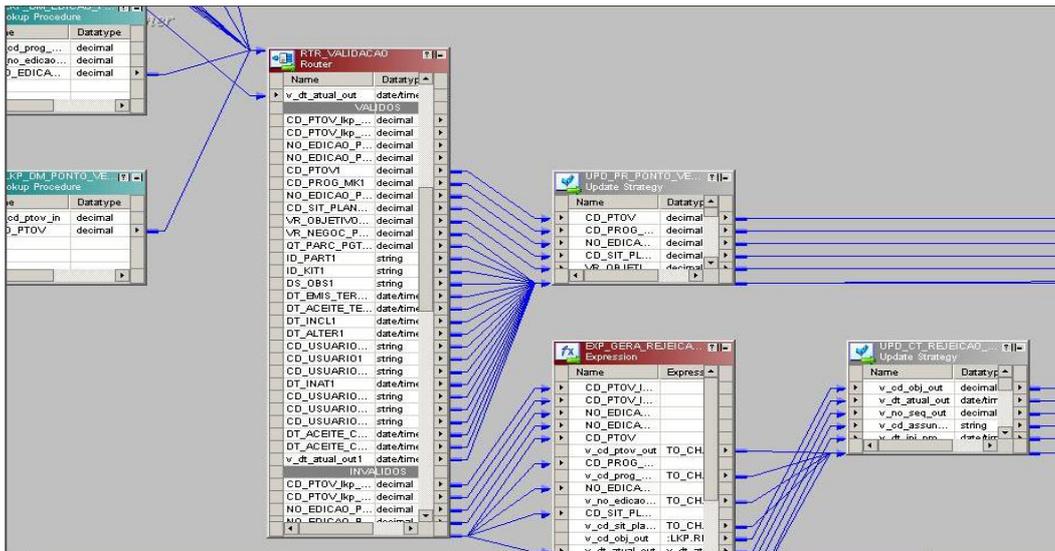


Figura 4.21 – Programa ETL – Router

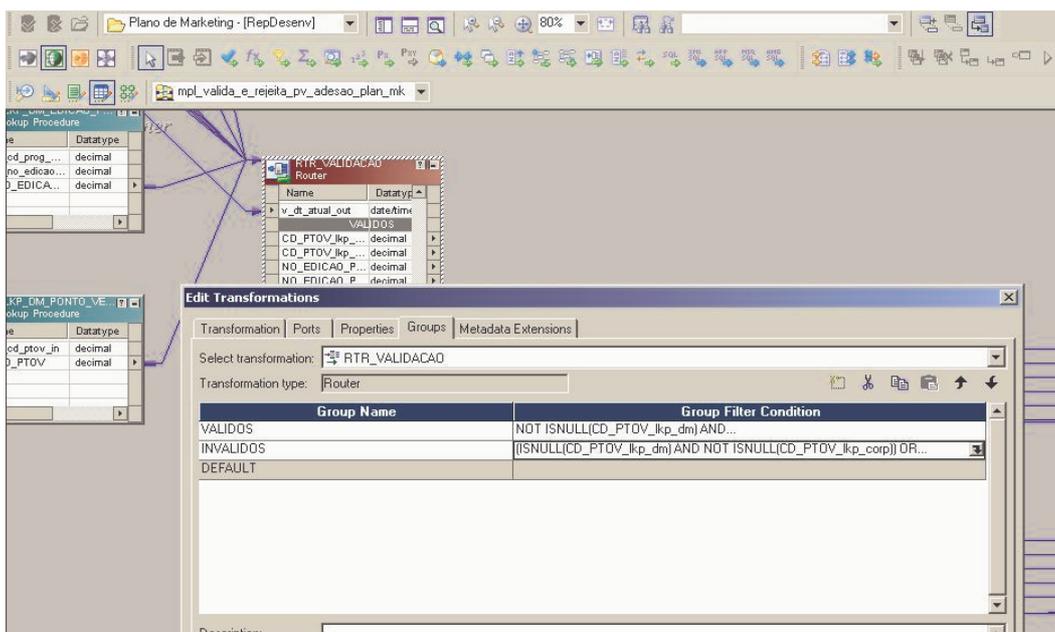


Figura 4.22 – Programa ETL – Router - Condição

4.5.3 Fluxo dos Aptos

O terceiro programa carrega os clientes que estão aptos a participar do Plano de Marketing, porém o assessor ainda não realizou a negociação. A carga considera as duas últimas edições. É bem semelhante ao fluxo de saídas visto na seção 4.5.2. O processo de carga é ilustrado na figura 4.23.

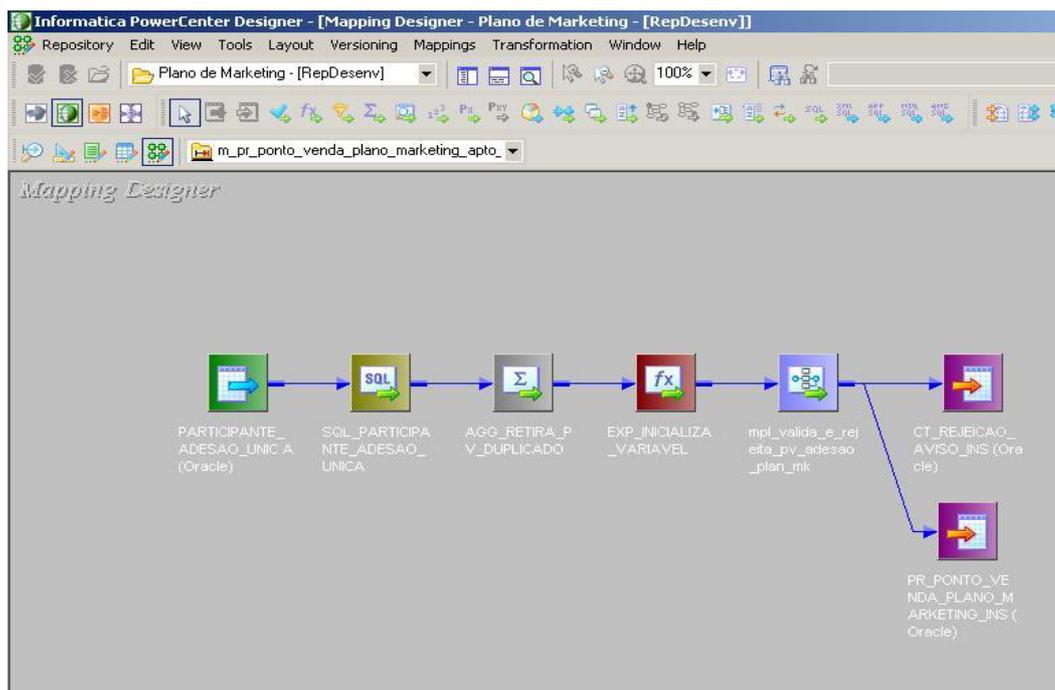


Figura 4.23 – Programa ETL – Fluxo dos Aptos

A diferença ocorre dentro do Source Qualifier em que o SQL é alterado para filtrar somente os clientes que estão com o valor objetivado para participação do Plano de Marketing pago e a data de inativação e data emissão do termo de adesão nula. A figura 4.24 a seguir mostra a query dentro da transformação Source Qualifier.

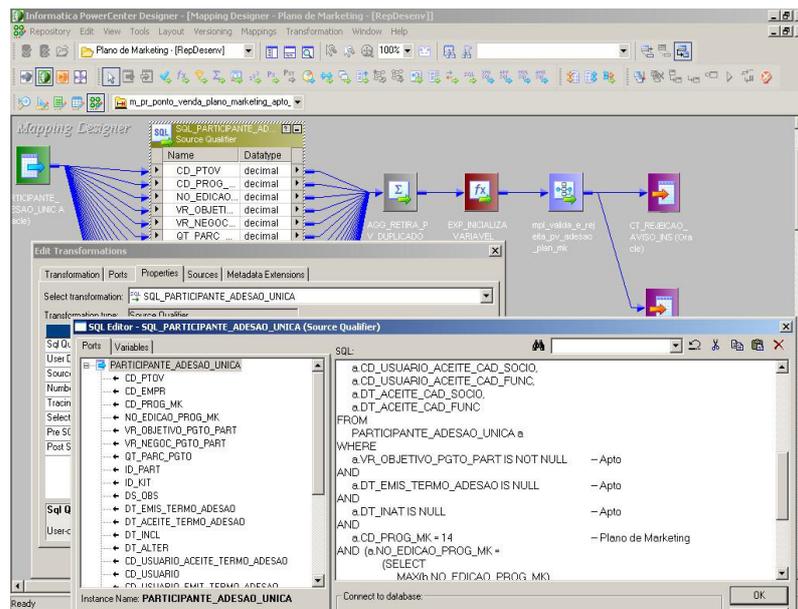


Figura 4.24 – Programa ETL – Fluxo dos Aptos - SQL

O SQL deste mapa pode ser visualizado melhor através da query a seguir:

```

SELECT
    a.CD_PTOV,
    a.CD_PROG_MK,
    a.NO_EDICAO_PROG_MK,
    a.VR_OBJETIVO_PGTO_PART,
    a.VR_NEGOC_PGTO_PART,
    a.QT_PARC_PGTO,
    a.ID_PART,
    a.ID_KIT,
    a.DS_OBS,
    a.DT_EMIT_TERM_ADESAD,
    a.DT_ACEITE_TERM_ADESAD,
    a.DT_INCL,
    a.DT_ALTER,
    a.CD_USUARIO_ACEITE_TERM_ADESAD,
    a.CD_USUARIO,
    a.CD_USUARIO_EMIT_TERM_ADESAD,
    a.DT_INAT,
    a.CD_USUARIO_INAT,
    a.CD_USUARIO_ACEITE_CAD_SOCIO,
    a.CD_USUARIO_ACEITE_CAD_FUNC,
    a.DT_ACEITE_CAD_SOCIO,
    a.DT_ACEITE_CAD_FUNC
FROM
    PARTICIPANTE_ADESAD_UNICA a
WHERE
    a.VR_OBJETIVO_PGTO_PART IS NOT NULL -- Apto
AND
    a.DT_EMIT_TERM_ADESAD IS NULL -- Apto
AND
    a.DT_INAT IS NULL -- Apto
AND
    a.CD_PROG_MK = 14 -- Plano de Marketing
AND (a.NO_EDICAO_PROG_MK =
    (SELECT
        MAX(b.NO_EDICAO_PROG_MK)
        FROM
            PARTICIPANTE_ADESAD_UNICA b
            WHERE b.CD_PROG_MK = 14)
OR
    a.NO_EDICAO_PROG_MK =
    (SELECT
        MAX(c.NO_EDICAO_PROG_MK) - 1
        FROM
            PARTICIPANTE_ADESAD_UNICA c
            WHERE c.CD_PROG_MK = 14))

```

4.5.4 Fluxo dos Não Aptos

O quarto processo que carrega os Pontos de Venda (Clientes) que não estão aptos ao Plano de Marketing nas duas últimas edições.

A figura 4.25 mostra o programa de ETL para essa situação.

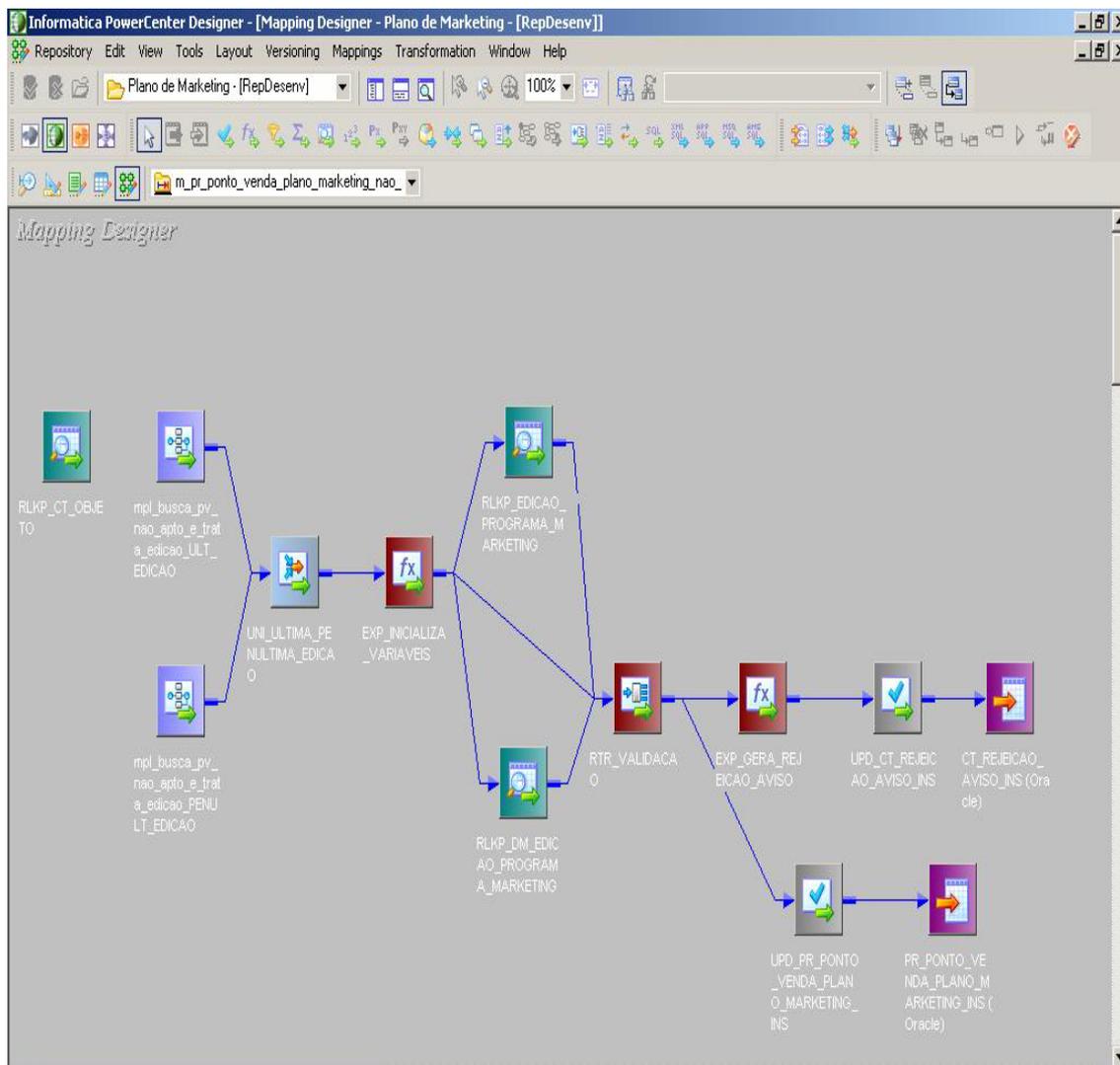


Figura 4.25 – Programa ETL – Fluxo dos Não Aptos

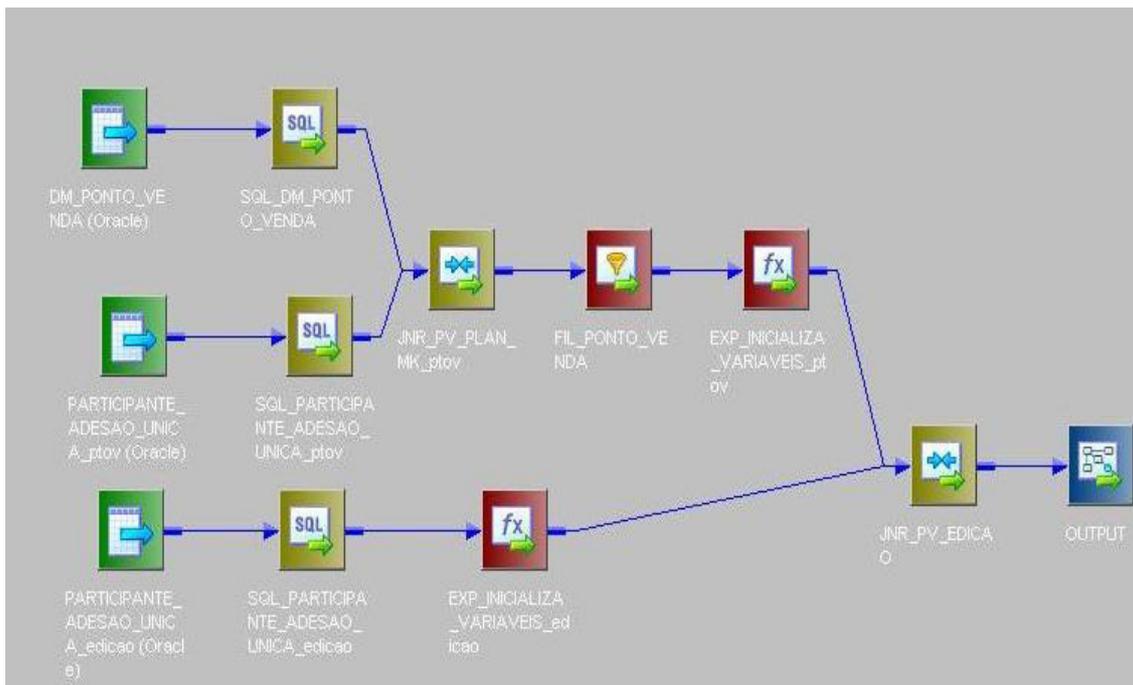


Figura 4.27 – Programa ETL – Fluxo dos Não Aptos - Maplet

Union Transformation: é o objeto de cor azul na figura 4.27. Ele une os clientes da penúltima e última edição.

Joiner: Essa transformação é utilizada dentro do maplet da figura 4.27 com o objetivo de realizar um join entre as tabelas.

Os demais objetos foram definidos anteriormente, não sendo mencionados neste exemplo novamente.

4.5.5 Fluxo dos Termos Gerados

O quinto programa tem o objetivo de processar os Pontos de Venda que estão com Termos de Aceite pendente de assinatura no Plano de Marketing. O processo que carrega esses clientes é mostrado através da figura 4.28 a seguir, muito semelhante ao visto nas seções 4.5.2 e 4.5.3.

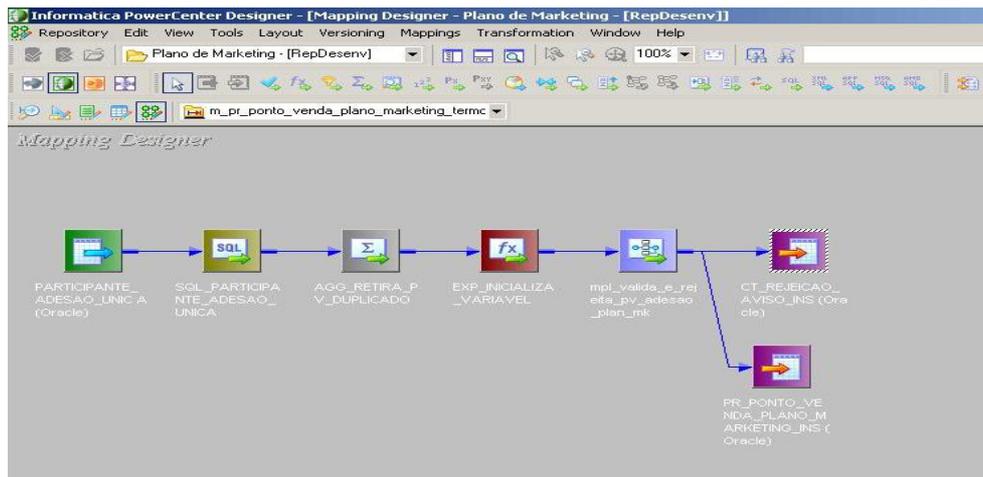


Figura 4.28 – Programa ETL – Fluxo Termo Gerado

Percebe-se que este mapa está utilizando o mesmo mapplet do programa que carrega os clientes que cancelaram sua participação e os que estão aptos para participarem.

A query utilizada no Source Qualifier deste mapa é expressa abaixo.

```

SELECT
    a.CD_PTOV,
    a.CD_PROG_MK,
    a.NO_EDICAO_PROG_MK,
    a.VR_OBJETIVO_PGTO_PART,
    a.VR_NEGOC_PGTO_PART,
    a. QT_PARC_PGTO,
    a.ID_PART,
    a.ID_KIT,
    a.DS_OBS,
    a.DT_EMIS_TERMOS_ADESAO,
    a.DT_ACEITE_TERMOS_ADESAO,
    a.DT_INCL,
    a.DT_ALTER,
    a.CD_USUARIO_ACEITE_TERMOS_ADESAO,
    a.CD_USUARIO,
    a.CD_USUARIO_EMIT_TERMOS_ADESAO,
    a.DT_INAT,
    a.CD_USUARIO_INAT,
    a.CD_USUARIO_ACEITE_CAD_SOCIO,
    a.CD_USUARIO_ACEITE_CAD_FUNC,
    a.DT_ACEITE_CAD_SOCIO,
    a.DT_ACEITE_CAD_FUNC
FROM
    PARTICIPANTE_ADESAO_UNICA a
WHERE
    a.DT_EMIS_TERMOS_ADESAO IS NOT NULL -- Termo Gerado
AND
    a.DT_ACEITE_TERMOS_ADESAO IS NULL -- Termo Gerado
AND
    a.DT_INAT IS NULL
AND
    a.CD_PROG_MK = 14 -- Plano de Marketing
AND (a.NO_EDICAO_PROG_MK =
      (SELECT
MAX(b.NO_EDICAO_PROG_MK)
FROM
PARTICIPANTE_ADESAO_UNICA b
WHERE b.CD_PROG_MK = 14)
OR
a.NO_EDICAO_PROG_MK =
(SELECT
MAX(c.NO_EDICAO_PROG_MK) - 1
FROM
PARTICIPANTE_ADESAO_UNICA c
WHERE c.CD_PROG_MK = 14))

```

4.5.6 Fluxo dos Termos de Aceite

O sexto fluxo tem o objetivo de carregar todos os clientes que possuem o contrato assinado (termo aceite) de participação do Plano de Marketing. Abaixo segue a imagem 4.29 ilustrando essa carga.

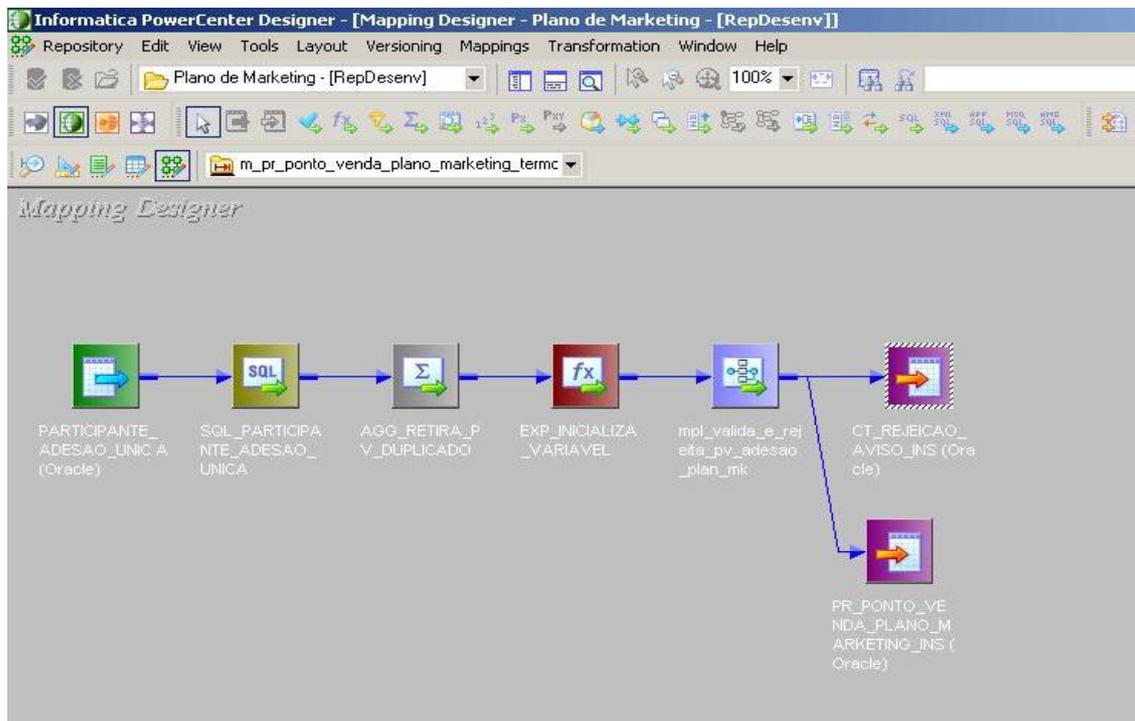


Figura 4.29 – Programa ETL – Fluxo Termo Aceito

Com a estrutura semelhante aos fluxos 4.5.2, 4.5.3 e 4.5.5 o SQL para essa carga pode ser visualizado a seguir. Nele ocorre um filtro pela data de inativação que deve ser nula e o campo DT_ACEITE_TERMO_ADESAO diferente de nulo, indicando a data em que o cliente fechou o contrato com a empresa para participar da promoção.

```

SELECT
    a.CD_PTOV,
    a.CD_PROG_MK,
    a.NO_EDICAO_PROG_MK,
    a.VR_OBJETIVO_PGTO_PART,
    a.VR_NEGOC_PGTO_PART,
    a.QT_PARC_PGTO,
    a.ID_PART,
    a.ID_KIT,
    a.DS_OBS,
    a.DT_EMIS_TERMOS_ADESAO,
    a.DT_ACEITE_TERMOS_ADESAO,
    a.DT_INCL,
    a.DT_ALTER,
    a.CD_USUARIO_ACEITE_TERMOS_ADESAO,
    a.CD_USUARIO,
    a.CD_USUARIO_EMIT_TERMOS_ADESAO,
    a.DT_INAT,
    a.CD_USUARIO_INAT,
    a.CD_USUARIO_ACEITE_CAD_SOCIO,
    a.CD_USUARIO_ACEITE_CAD_FUNC,
    a.DT_ACEITE_CAD_SOCIO,
    a.DT_ACEITE_CAD_FUNC
FROM
    PARTICIPANTE_ADESAO_UNICA a
WHERE
    a.DT_ACEITE_TERMOS_ADESAO IS NOT NULL -- Termo Aceito
AND
    a.DT_INAT IS NULL -- Termo Aceito
AND
    a.CD_PROG_MK = 14 -- Plano de Marketing
AND (a.NO_EDICAO_PROG_MK =
    (SELECT
        MAX(b.NO_EDICAO_PROG_MK)
        FROM
            PARTICIPANTE_ADESAO_UNICA b
            WHERE b.CD_PROG_MK = 14)
OR
    a.NO_EDICAO_PROG_MK =
    (SELECT
        MAX(c.NO_EDICAO_PROG_MK) - 1
        FROM
            PARTICIPANTE_ADESAO_UNICA c
            WHERE c.CD_PROG_MK = 14))

```

4.5.7 Sessões e Workflow

Depois que todos os mapas foram criados, é necessário escolher as conexões desses mapas para se comunicarem com o banco de dados. Para isso, é utilizado o conceito de sessão. Uma sessão está relacionada a um e somente um processo de ETL. É nela, que serão configuradas todas as conexões que o mapa precisa para ser executado. A figura 4.30 ilustra as sessões dos programas criados anteriormente.

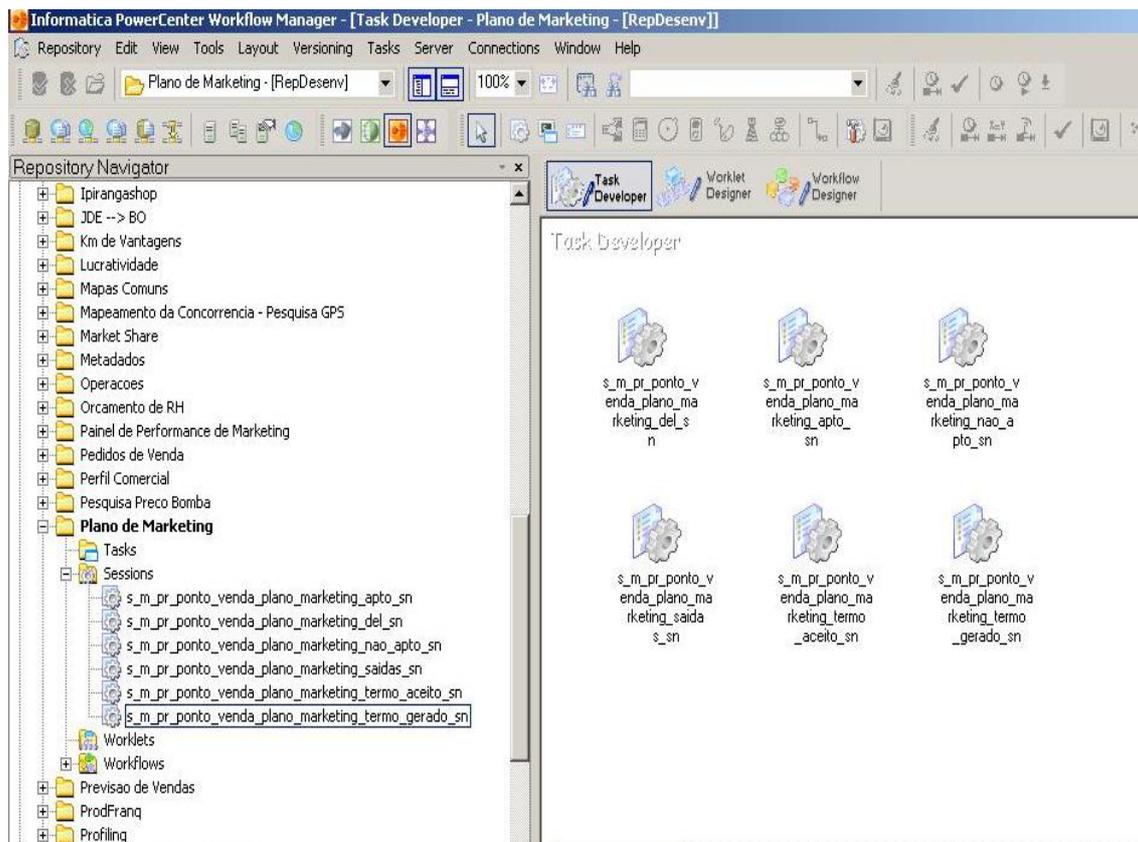


Figura 4.30 – Sessões

A figura 4.31 mostra as conexões dentro de uma sessão.

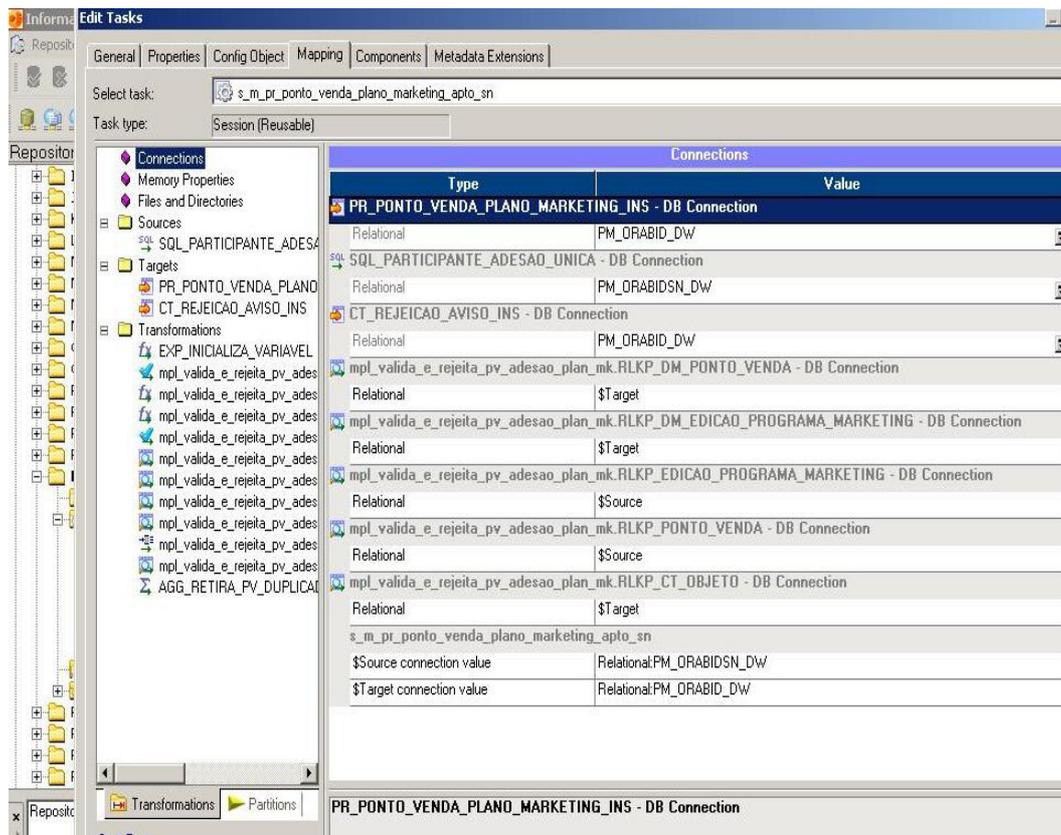


Figura 4.31 – Sessão - Conexões

Após a criação de todas as sessões, cria-se o objeto workflow. Esse objeto tem a função de realizar o controle das sessões e determinar ordem de dependência entre os fluxos. Quais cargas podem rodar em paralelo e quais podem rodar em série? Todas essas análises são feitas e monitoradas pelo Workflow. As sessões são instanciadas dentro do Workflow, que pode conter uma ou mais sessões dentro dele.

A figura 4.32 a seguir mostra um exemplo de workflow para o projeto de Plano de Marketing.

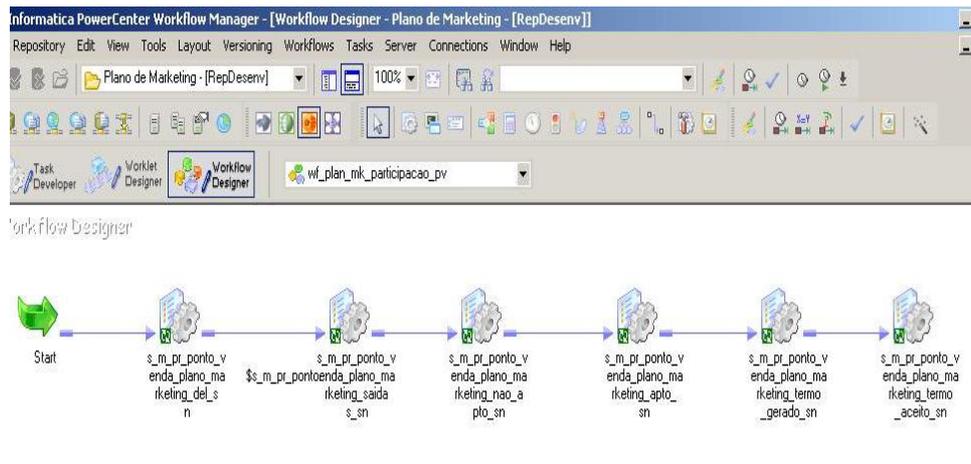


Figura 4.32 – Workflow

4.6 Quarta Fase

Com a conclusão da etapa de construção dos programas de ETL, os dados já estão armazenados dentro do Data Warehouse para serem consultados. Nesta quarta fase, utiliza-se a ferramenta OLAP chamada MicroStrategy para criar os relatórios no BI. Esses relatórios são publicados na intranet da empresa para consulta por parte dos usuários. Neste projeto foram criados alguns relatórios com o objetivo de fornecer ao usuário o conhecimento necessário para análise e tomada de decisão. Esses relatórios podem ser vistos de diferentes formas por parte do usuário. A figura 4.33 mostra um exemplo dos relatórios publicados no BI disponíveis para o usuário.

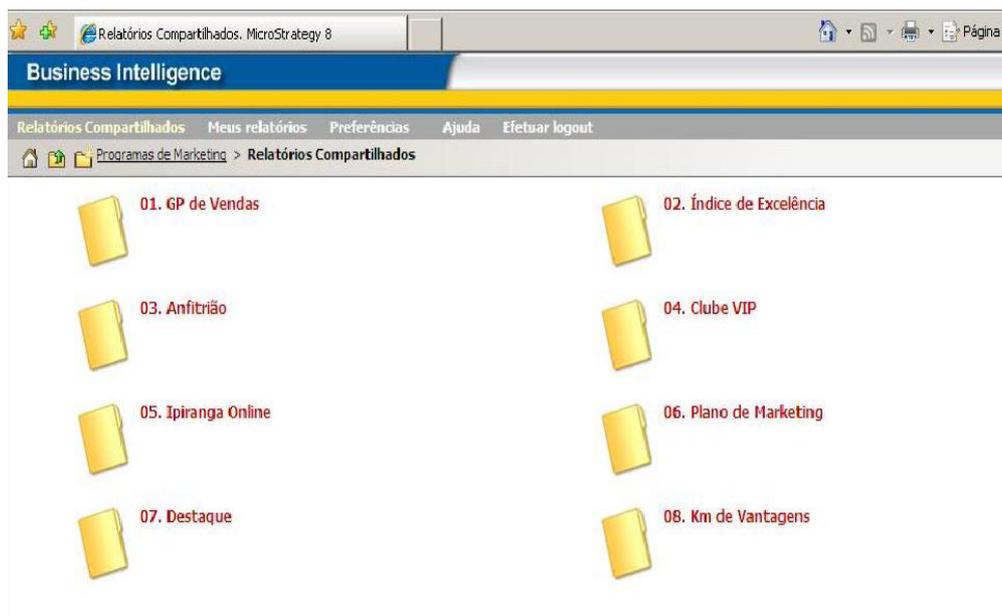


Figura 4.33 – Pastas dos Relatórios

O usuário tem a opção de escolher o relatório do assunto de seu interesse dando um clique em uma das pastas acima. Os relatórios do Plano de Marketing podem ser encontrados na pasta 06.

Após o usuário escolher a pasta 06. *Plano de Marketing* os seguintes relatórios são disponibilizados para análise como mostra a figura 4.34.

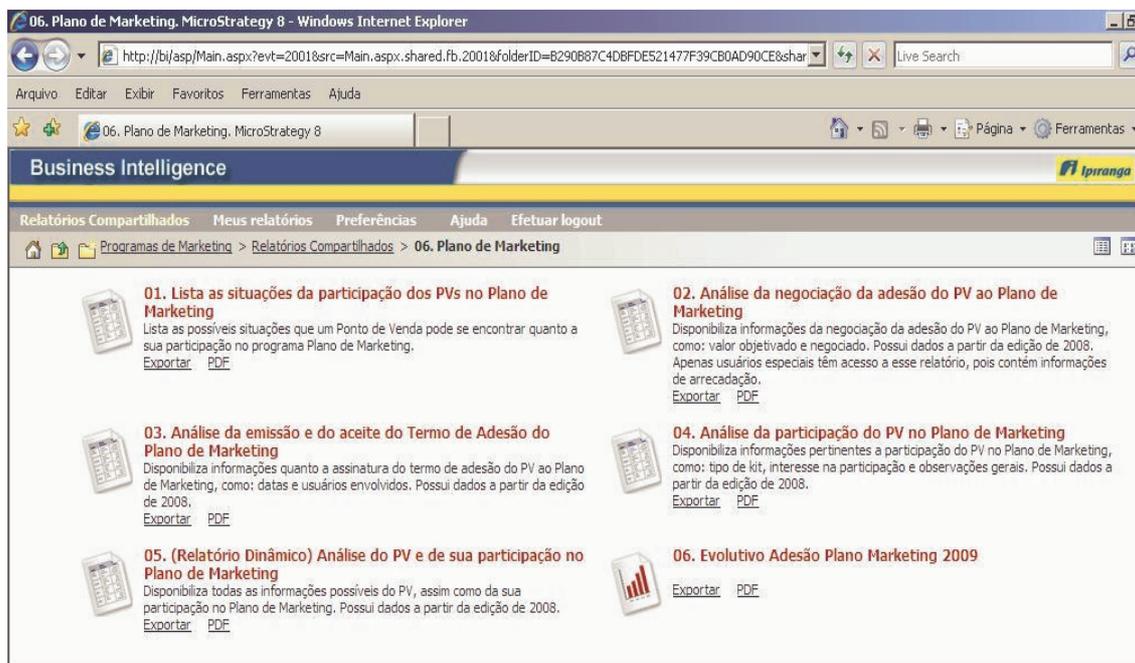


Figura 4.34 – Relatórios

O primeiro relatório lista as situações da participação dos Clientes no Plano de Marketing como mostra a figura 4.35.

The screenshot shows a web browser window titled "01. Lista as situações da participação dos PVs no Plano de Marketing, MicroStrategy 8 - Windows Internet Explorer". The address bar shows a URL starting with "http://bi/asp/Main.aspx?". The page content displays a table with the following data:

Código da Situação	Nome da Situação	Descrição da Situação
1	Saídas	Pontos de Venda que cancelaram sua participação.
2	Não Apto	Pontos de Venda não participantes e que não cancelaram sua participação.
3	Apto	Pontos de Venda aptos, mas o assessor ainda não realizou a negociação.
4	Termo Gerado	Pontos de Venda com o Termo de Aceite pendente de assinatura.
5	Termo Aceito	Pontos de Venda participantes com o Termo de Aceite já assinado.

Figura 4.35 – Lista as situações da participação dos Clientes no Plano de Marketing

O segundo relatório faz a análise da negociação da adesão do cliente ao Plano de Marketing.

The screenshot displays a web application interface for Business Intelligence. The main content area shows a report titled "02. Análise da negociação da adesão do PV ao Plano de Marketing". The report is presented as a table with two columns: "Ponto de Venda" and "Qtde. de Parcelas". The table lists 20 different points of sale (Ponto de Venda) and their corresponding number of parcels (Qtde. de Parcelas). The interface also includes a navigation menu at the top, a search bar, and a status bar at the bottom indicating "Linhas de dados: 28" and "Colunas de dados: 0".

Ponto de Venda ▲	Qtde. de Parcelas ▲
A P Kellner Ltda	10
Abast Combs Sepiranga Ltda	10
Ana M Groberio	10
Auto Abast Machado Ltda	10
Auto Posto Aguas Frias Ltda	10
Auto Posto Avenida Constantina Ltda	10
Auto Posto Jjf Ltda	10
Auto Posto Midas Ltda	10
Auto Posto Natal Ltda	10
Auto Posto Sao Lucas Ltda	10
Auto Posto Viaduto Independencia Ltda	9
Auto Posto Xiru Ltda	10
Cgg Combs E Servicos Ltda	10
Com Combs Rio Preto Ltda	10
Com De Combs Stocchero Ltda	10
Com Maqs Combs Pitangueira Ltda	10
Coop Trit Mista Campo Novo Ltda	10
Garoupa Der De Petr Ltda	10
Joao C Rossi	10
Lcs Calixto	10
P Analise Ltda	9
P De Gas Floresta Ltda	10

Figura 4.36 – Análise da negociação da adesão do cliente ao Plano de Marketing

O terceiro relatório faz a análise da emissão e do aceite do Termo de Adesão do Plano de Marketing.

Business Intelligence

Relatórios Compartilhados Meus relatórios Preferências Ajuda Efetuar logout

Programas de Marketing > Relatórios Compartilhados > 06. Plano de Marketing > 03. Análise da emissão e do aceite do Termo de Adesão do Plano de Marketing

Arquivo Exibir Dados Formato Última atualização: 16/11/09 15

PAGINAR POR: Edição Programa de Marketing: Plano De Marketing 2008 Situação: Termo Aceito

Saídas
Não Apto
Apto

Linhas de dados: 1 - 400 de 2679 Colunas de de

Ponto de Venda		Data de Emissão do Termo	Usuário que Gerou o Termo	Data de Aceite do Termo	Usuário que Aceitou o Termo
3a P De Serv E Conv Ltda	80058	25/2/2008 09:28:45	amauriv	25/2/2008 14:52:31	ps3a
7 Postos Cabreuva Ltda	1278682	16/2/2008 00:07:10	acapozzi	3/3/2008 12:41:18	ps7postos
A Abast Love Story Ltda	7762	5/3/2008 06:38:34	analv	5/3/2008 14:03:32	love
A Alves Com De Deriv De Petroleo Ltda	1293132	23/5/2008 15:53:37	leopoldo	23/5/2008 15:54:14	psaalves
A B O Distr De Der De Petr Ltda	24354	20/2/2008 15:59:15	leilar	22/2/2008 17:01:38	postoesper
A Center Galina Ltda	55677	25/2/2008 12:26:45	fbarros	25/2/2008 13:34:42	galina
A Centro P Aguiar Ltda	84941	27/2/2008 10:22:38	algayer	5/3/2008 14:38:38	apaguiar
A De Oliveira Posto De Gasolina - Me	1291694	28/2/2008 18:33:11	alsergio	29/2/2008 17:58:23	psadeoliveira
A Diesel Miradouro Ltda	55704	25/2/2008 16:15:25	ebouhid	25/2/2008 16:29:11	psmiradouro
A F Torres & Cia Ltda	11676	27/2/2008 13:50:05	maurino	29/2/2008 15:47:25	torres
A I Azrak & Cia Ltda - Filial	9308	22/2/2008 17:23:57	lmoragas	23/2/2008 12:04:35	azrak
A I Azrak & Cia Ltda - Matriz	77258	22/2/2008 17:24:32	lmoragas	23/2/2008 11:34:09	azrak
A J D'Agostini & Cia Ltda	10987	11/2/2008 00:06:43	falves	5/3/2008 18:34:58	agostini
A J Koch E Cia Ltda	1275252	20/2/2008 18:02:27	mbegrow	5/3/2008 09:52:08	koch
A Lubrif Trimiense Ltda	29691	3/3/2008 16:08:36	guilhermec	3/3/2008 17:17:36	trimiense
A P 10 De Dezembro Ltda	22247	14/2/2008 13:38:48	avolpini	18/2/2008 10:35:58	dez
A P 1010 Ltda	9696	5/3/2008 17:57:00	ivanf	5/3/2008 18:12:43	ap1010
A P 151 Ltda	33712	5/3/2008 15:30:10	jtubino	5/3/2008 16:56:14	ap151
A P 18 Irmaos Bogaz Ltda	107863	17/4/2008 10:01:44	jfeitoza	17/4/2008 10:04:43	ap18ibogaz
A P 188 Ltda	14551	15/2/2008 12:09:36	femandof	28/2/2008 17:42:50	ps188

Figura 4.37 – Análise da emissão e do aceite do Termo de Adesão do Plano de Marketing

O quarto relatório faz a análise da participação do cliente no Plano de Marketing.

The screenshot displays a web application interface for 'Business Intelligence' by 'Ipiranga'. The main report is titled '04. Análise da participação do PV no Plano de Marketing'. The interface includes a navigation menu with options like 'Relatórios Compartilhados', 'Meus relatórios', 'Preferências', 'Ajuda', and 'Efetuar logout'. The report is part of a series: 'Programas de Marketing > Relatórios Compartilhados > 06. Plano de Marketing > 04. Análise da participação do PV no Plano de Marketing'. The report is dated '16/11/09 15:59:16'. The data is presented in a table with the following columns: 'Ponto de Venda', 'Kit', 'Participação', and 'Observação'. The table contains 24 rows of data, each representing a different point of sale (Ponto de Venda) and its participation in the marketing plan.

Ponto de Venda	Kit	Participação	Observação
A P Atalaia Ltda	1288572 URB	Sim	
A P Bateias Ltda	50397 URB	Sim	
A P Conilon Ltda	7305 URB	Sim	
A P Cruz De Malta Ltda	85858 ROD	Sim	
A P De Serv Cabralia Ltda	76717 ROD	Sim	
A P E Com Darolt Ltda	86174 URB	Sim	
A P Florao Ltda	85565 ROD	Sim	
A P Formigao Ltda	84919 ROD	Sim	
A P Kak Com De Combs E Lubes Ltda	21258 ROD	Sim	
A P Munaretto Ltda	11962 URB	Sim	
A P Seyboth Ltda	85617 URB	Sim	
Abast Combs Alberton Ltda	1275325 URB	Sim	
Abast Combs Faccin E Faccin Ltda	1275135 URB	Sim	
Abast Combs Frantel Ltda	1275193 ROD	Sim	
Abast Combs Sepiranga Ltda	1279776 ROD	Sim	
Abast Combs Servicentro Ltda/Me	1284457 URB	Sim	
Abast Combs Viel Ltda	1275078 URB	Sim	
Abast Picada Cafe Ltda	1275591 URB	Sim	
Abast Talisma Ltda	1275109 ROD	Sim	
Abast Teutonia Ltda	1275395 ROD	Sim	
Adonatto Comercio De Combs Ltda	1275424 URB	Sim	
Alvino Pianzoli E Irmao Ltda	64652 URB	Sim	

Figura 4.38 – Análise da participação do cliente no Plano de Marketing

O quinto relatório é um relatório dinâmico que analisa o cliente e como está a sua participação no Plano de Marketing.

The screenshot shows a web-based Business Intelligence application. The browser title is "05. (Relatório Dinâmico) Análise do PV e de sua partici...". The application header includes "Business Intelligence" and navigation links like "Relatórios Compartilhados", "Meus relatórios", "Preferências", "Ajuda", and "Efetuar logout". The breadcrumb trail is "Programas de Marketing > Relatórios Compartilhados > 06. Plano de Marketing > 05. (Relatório Dinâmico) Análise do PV e de sua participação no Plano de Marketing". The interface includes a menu bar (Arquivo, Exibir, Dados, Formato) and a toolbar with various data analysis icons. Below the toolbar, there are filters for "Edição Programa de Marketing: Plano De Marketing 2009" and "Situação no Plano de Marketing: Não Apto". The main content area shows a table with 4 columns: "Ponto de Venda", "Participação", "Data de Emissão do Termo", and "Data de Aceite do Termo". The table contains 20 rows of data for various sales points.

Ponto de Venda ▲	Participação ▲	Data de Emissão do Termo ▲	Data de Aceite do Termo ▲
2008 Pecas E Acessorios Ltda - Me	53090		
5 Avenida Posto De Servicos Ltda	1304249		
A Abast Recreio Com E Serv Ltda	319999		
A Abast Recreio Com E Serv Ltda	320000		
A Aguiar Neto Comb Lubrif	1313057		
A B Caceres Epp	1309141		
A B Lessa Neto & Cia Ltda	1310699		
A B Ribeiro	1312199		
A C Cambraia	1311626		
A De M Viana Posto	1310567		
A F M Albuquerque Combustiveis	1309747		
A F Macedo & Cia Ltda	1311552		
A F Moraes Melo	1313283		
A G Amaral Apiai	1285895		
A J Luchetta E Filho Ltda	52799		
A J Santos Cia Ltda	1312582		
A L & M Auto Posto Ltda	40967		
A L & M Auto Posto Ltda	1278182		
A L C Galvao	1306532		
A M Bittencourt	1313574		
A M Bittencourt	1313598		
A M Diesel Ltda	1304517		

Figura 4.39 – Analisa o cliente e como está a sua participação no Plano de Marketing

O sexto e último relatório é um gráfico com a evolução dos clientes que aderiram ao Plano de Marketing ao longo do ano.

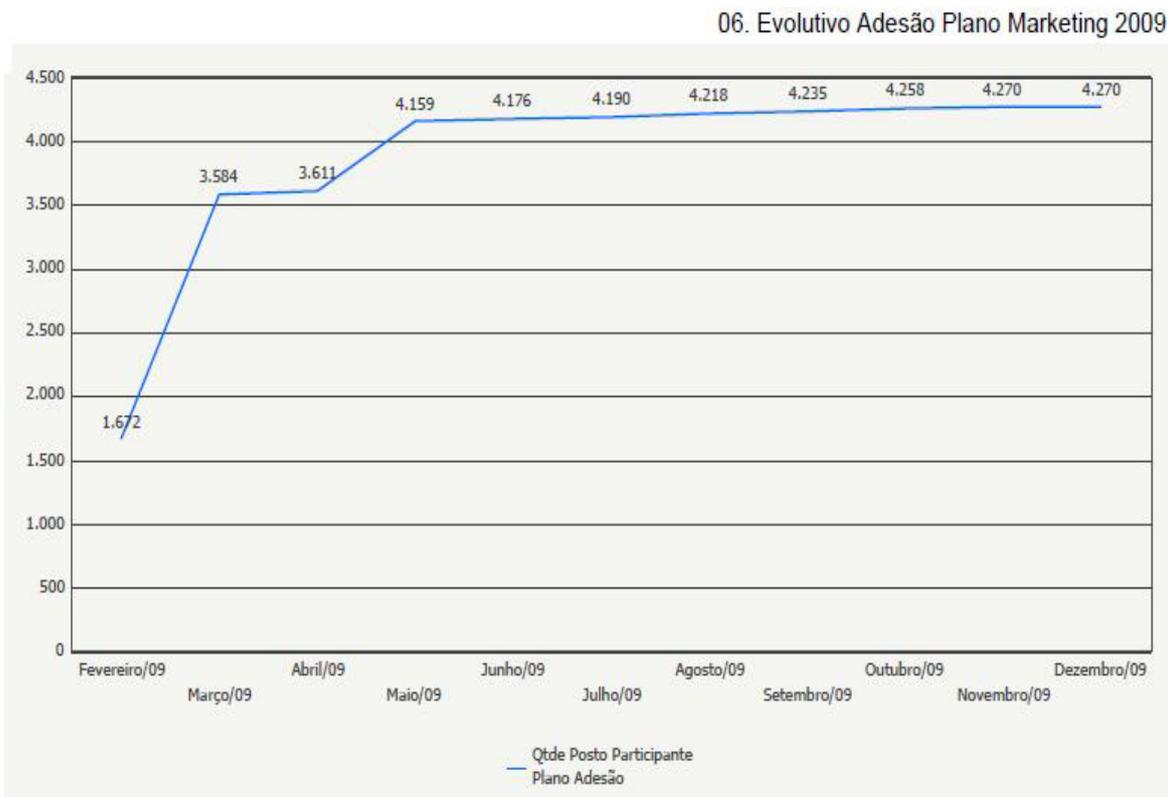


Figura 4.40 – Evolução dos clientes que aderiram ao Plano de Marketing ao longo do ano

São notórias as vantagens e os benefícios que o processo de migração da aplicação Java para o ambiente de Business Intelligence traz para os usuários.

Dentre os benefícios encontrados é possível citar:

- O desempenho da execução dos programas de ETL quando comparada com a aplicação em Java é bem melhor.
- A facilidade em gerar diferentes tipos de relatórios com as ferramentas do BI enquanto que na aplicação Java era simplesmente gerado uma planilha em Excel para análise.
- O problema de sobrecarga no servidor que estava ocorrendo devido à grande quantidade de usuários utilizando a aplicação foi solucionado.

5 CONCLUSÃO

A implantação de um processo de Business Intelligence dentro das organizações é de suma importância nos dias atuais. Como visto ao longo dessa monografia, um sistema de BI tem trazido grandes benefícios por parte daqueles que aderem a essa tecnologia. Para isso, o investimento em um sistema desse porte precisa ser feito com muito cuidado. É necessário um bom planejamento para que não haja falhas dos componentes e entre esses componentes que se relacionam entre si dentro do sistema de Business Intelligence.

Neste trabalho procurou-se discutir acerca do conceito de Business Intelligence abordando cada componente de um sistema de BI, dentre os quais podemos citar o Data Warehouse, as ferramentas de ETL e OLAP, e a forma como eles se relacionam uns com os outros.

Além deste estudo, o aspecto de implementação também foi abordado com a criação de tabelas dentro do Data Warehouse, além da implementação de programas de ETL e a criação de relatórios explorando o conceito de OLAP. Em particular, foi realizada a migração de uma aplicação Java para ETL com o intuito de demonstrar na prática como funciona um sistema de Business Intelligence.

Em termos de resultados, procurou-se demonstrar principalmente a facilidade que é encontrada por parte dos processos de ETL em carregar grandes volumes de dados em um curto intervalo de tempo. Além disso, há vantagens em se ter uma ferramenta voltada para geração de relatórios atendendo de forma satisfatória os requisitos dos usuários em substituição à planilha que era gerada pela aplicação em Java.

6 REFERÊNCIAS

- [1] PRIMAK, FÁBIO, DECISÕES COM B.I / FÁBIO PRIMAK
- [2] BARBIERI, CARLOS, BUSINESS INTELLIGENCE/CARLOS BARBIERI
- [3] Equipe DW, site de profissionais colocados à disposição das comunidades acadêmica e empresarial seus conhecimentos e competência na área de Data Warehouse, disponível na Internet via protocolo HTTP, no endereço <http://www.datawarehouse.inf.br>, data do último acesso 20/10/2009.
- [4] COREY, Michael, Oracle 8i Data Warehouse/Michael Corey, Ian Abramson, Michael Abbey, Ben Taub: Oracle Press.
- [5] Kimball, Ralph, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling - Ralph Kimball, Margy Ross; SECOND EDITION
- [6] Schulz, Alex, Blog contendo informações relacionadas à Tecnologia da Informação, disponível na Internet via protocolo HTTP, no endereço <http://blogs.msdn.com/alexschulz/archive/2008/10/04/overview-data-war.spx>, data do último acesso em 21/10/2009.
- [7] Site da disciplina Sistema de Apoio à Decisão - Negócios Inteligentes na Web, para alunos da graduação do curso de Ciência da Computação da UNISUL – Univ. do Sul de Santa Catarina, disponível na Internet via protocolo HTTP, no endereço <http://www2.stela.ufsc.br/aran/sad> com último acesso em 24/10/2009.
- [8] Baião, Fernanda, Site da disciplina Tópicos Avançados em Banco de Dados do curso de Ciência da Computação da UNIRIO, disponível na Internet via protocolo HTTP, no endereço <http://www.uniriotec.br/~fernanda.baiao/SAIN/> com último acesso em 24/10/2009.
- [9] SISNEMA INFORMATICA, Site de empresa destinada a proporcionar treinamentos profissionais na área de TI, disponível na Internet via protocolo HTTP, no endereço <http://sisnema.com.br/Materias/idmat014812.htm> com último acesso em 24/10/2009.
- [10] Microsoft Developer Network - MSDN, Site utilizado por profissionais e estudantes da área de TI com o objetivo de adquirir e trocar conhecimentos de novas tecnologias criadas pela Microsoft, disponível na Internet via protocolo HTTP, no endereço <http://msdn.microsoft.com/pt-br/library/cc518031.aspx> com último acesso em 24/10/2009.

[11] iMaster, Portal que congrega uma comunidade extremamente segmentada e fidelizada, de profissionais e estudantes na área de internet/tecnologia, formadores de opinião, apto a assimilar informações técnicas e comerciais capazes de estimular o interesse ao produto/serviço anunciado, disponível na Internet via protocolo HTTP, no endereço http://imasters.uol.com.br/artigo/3836/modelo_dimensional_para_data_warehouse com último acesso em 24/10/2009.

[12] Microsoft Developer Network - MSDN, Site utilizado por profissionais e estudantes da área de TI com o objetivo de adquirir e trocar conhecimentos de novas tecnologias criadas pela Microsoft, disponível na Internet via protocolo HTTP, no endereço.

<http://msdn.microsoft.com/pt-br/library/ms166713.aspx> com último acesso em 31/10/2009.

[13] MicroStrategy, Líder mundial na tecnologia de business intelligence, a MicroStrategy proporciona a criação de relatórios, análises e monitoramento integrados, auxiliando as empresas a tomar melhores decisões de negócios, disponível na Internet via protocolo HTTP, no endereço

<http://www.microstrategy.com/Events/symposium/detail.asp?eventid=1794> com último acesso em 31/10/2009.

[14] Silva, Flavio, artigo sobre Desmistificando o Conceito de ETL, aluno da faculdade Salesiana Maria Auxiliadora, disponível na Internet via protocolo HTTP, no endereço www.fsma.edu.br/si/Artigos/V2_Artigos1.pdf com último acesso em 31/10/2009.

[15] LigRede, é um site que reúne empresas, pesquisadores, consultores e idéias inovadoras para criar um ambiente favorável à realização de negócios em rede, disponível na Internet via protocolo http, no endereço <http://www.ligrede.com.br>, data do último acesso 14/10/2009.

[16] Apresentação Comercial da Empresa PriceWaterHouseCoopers sobre Business Intelligence.

[17] Abacus Soluções, site de uma empresa especializada em sistema de gestão, disponível na internet via protocolo http, no endereço <http://abacussolucoes.com/bi.html>, data do último acesso 10/09/2009.

[18] Inmon, William, Building the Data Warehouse – William H. Inmon; (3rd Edition), 2002.

APÊNDICE A

DW.DM_PROGRAMA_MARKETING		
CD_PROG_MK	NUMBER(3,0)	NOT NULL
NM_PROG_MK	VARCHAR2(30)	NOT NULL
DT_INI_PROG	DATE	NULL
DT_FIM_PROG	DATE	NULL
DT_INCL	DATE	NOT NULL
DT_ALTER	DATE	NULL
DT_INAT	DATE	NULL
DT_REAT	DATE	NULL
CD_USUARIO	VARCHAR2(30)	NOT NULL

DW.DM_PROGRAMA_MARKETING				
Nome da Entidade Lógica	Dimensão Programa de Marketing			
Nome da Tabela	DM_PROGRAMA_MARKETING			
Owner	DW			
Definição	Esta tabela contém os programas de Marketing. Ex.: 14 – PLANO DE MARKETING			
DW.DM_PROGRAMA_MARKETING Colunas				
Nome da Coluna	Domínio	Tipo	NULL	Definição
CD_PROG_MK 		NUMBER(3, 0)	NO	Código do Programa de Marketing. O código será um sequencial e servirá como chave da tabela. Ex.: 14 – Plano de Marketing
NM_PROG_MK		VARCHAR2(30)	NO	Nome do Programa de Marketing. O nome deve ser preenchido obrigatoriamente. Ex.: Plano de Marketing
DT_INI_PROG		DATE	YES	Data do Início do Programa. Contém a data em que o Programa de Marketing teve início.

<u>DT_FIM_PROG</u>		DATE	YES	Data do Fim do Programa. Contém a data em que o Programa de Marketing terminou.
<u>DT_INCL</u>		DATE	NO	Data de Inclusão. Contém a data em que o registro foi inserido na tabela e deve ser preenchido obrigatoriamente.
<u>DT_ALTER</u>		DATE	YES	Data de Alteração. Contém a data em que o registro foi alterado na tabela e só será preenchido caso o registro sofra mudanças.
<u>DT_INAT</u>		DATE	YES	Data de Inativação. Contém a data em que o registro foi inativado. Só deve estar preenchido caso o registro esteja inativo. Se o registro estiver ativo ou tiver sido reativado, não deve estar preenchido.
<u>DT_REAT</u>		DATE	YES	Data de Reativação. Contém a data em que o registro foi reativado. Só deve estar preenchido caso o registro tenha sido reativado. Se o registro estiver inativo ou ativo sem ter sido reativado, não deve estar preenchido.
<u>CD_USUARIO</u>		VARCHAR2(30)	NO	Código do Usuário. Contém o mnemônico do usuário que fez o cadastro do registro e deve ser preenchido obrigatoriamente.

DW.DM_EDICAO_PROGRAMA_MARKETING

<u>CD_PROG_MK (FK)</u>	NUMBER(3,0)	NOT NULL
<u>NO_EDICAO_PROG_MK</u>	NUMBER(3,0)	NOT NULL
<u>NM_EDICAO_PROG_MK</u>	VARCHAR2(40)	NOT NULL
<u>DT_INI_EDICAO_PROG_MK</u>	DATE	NOT NULL
<u>DT_FIM_EDICAO_PROG_MK</u>	DATE	NOT NULL
<u>DT_INCL</u>	DATE	NOT NULL
<u>DT_ALTER</u>	DATE	NOT NULL
<u>DT_INAT</u>	DATE	NULL
<u>DT_REAT</u>	DATE	NULL
<u>CD_USUARIO</u>	VARCHAR2(30)	NOT NULL

DW.DM_EDICAO_PROGRAMA_MARKETING

Nome da
Entidade
Lógica

Dimensão Edição Programa de Marketing

Nome da Tabela	DM_EDICAO_PROGRAMA_MARKETING
Owner	DW
Definição	Esta tabela contém as edições dos Programas de Marketing. Ex.: 1. Plano de Marketing 2007 2. Plano de Marketing 2008 3. Plano de Marketing 2009

DW.DM_PROGRAMA_MARKETING Colunas

Nome da Coluna	Domínio	Tipo	NULL	Definição
<u>CD_PROG_MK</u> 		NUMBER(3, 0)	NO	Código do Programa de Marketing. O código será um sequencial e servirá como chave da tabela. Ex.: 14 – Plano de Marketing
<u>NO_EDICAO_PROG_MK</u> 		NUMBER(3, 0)	NO	Número da Edição do Programa de Marketing. O número será um sequencial, faz parte da chave e contém as edições dos Programas de Marketing. Ex.: 1. Clube do Milhão 2005 2. Clube do Milhão 2006 3. Clube do Milhão 2007
<u>NM_EDICAO_PROG_MK</u>		VARCHAR(40)	NO	Nome Edição Programa de Marketing. O nome deve ser preenchido obrigatoriamente. Ex.: Clube do Milhão 2008, Índice de Excelência 2008, Ipiranga Shop 2008, etc.
<u>DT_INI_EDICAO_PROG_MK</u>		DATE	NO	Data do Início da Edição Programa de Marketing. Contém a data em que teve início a Edição do Programa de Marketing.
<u>DT_FIM_EDICAO_PROG_MK</u>		DATE	NO	Data do Fim Edição Programa de Marketing. Contém a data em que a Edição do Programa de Marketing terminou.
<u>DT_INCL</u>		DATE	NO	Data de Inclusão. Contém a data em que o

				registro foi inserido na tabela e deve ser preenchido obrigatoriamente.
<u>DT ALTER</u>		DATE	NO	Data de Alteração. Contém a data em que o registro foi alterado na tabela e só será preenchido caso o registro sofra mudanças.
<u>DT INAT</u>		DATE	YES	Data de Inativação. Contém a data em que o registro foi inativado. Só deve estar preenchido caso o registro esteja inativo. Se o registro estiver ativo ou tiver sido reativado, não deve estar preenchido.
<u>DT REAT</u>		DATE	NO	Data de Reativação. Contém a data em que o registro foi reativado. Só deve estar preenchido caso o registro tenha sido reativado. Se o registro estiver inativo ou ativo sem ter sido reativado, não deve estar preenchido.

<u>CD USUARIO</u>		VARCHAR2(30)	NO	Código do Usuário. Contém o mnemônico do usuário que fez o cadastro do registro e deve ser preenchido obrigatoriamente.
-------------------	--	--------------	----	---

DW.DM_SITUACAO_PLANO_MARKETING

CD_SIT_PLAN_MK	NUMBER(3,0)	NOT NULL
NM_SIT_PLAN_MK	VARCHAR2(50)	NOT NULL
DS_SIT_PLAN_MK	VARCHAR2(150)	NOT NULL
DT_INCL	DATE	NOT NULL
DT_ALTER	DATE	NULL
DT_INAT	DATE	NULL
DT_REAT	DATE	NULL
CD_USUARIO	VARCHAR2(30)	NOT NULL

DW.DM_SITUACAO_PLANO_MARKETING

Nome da Entidade Lógica	Dimensão Situação Plano de Marketing
Nome da Tabela	DM_SITUACAO_PLANO_MARKETING
Owner	DW
Definição	<p>Esta tabela contém as possíveis situações na qual o cliente pode se encontrar quanto a sua participação no programa Plano de Marketing (Adesão Única).</p> <p>Ex.: 1. Saídas (Pontos de Venda que cancelaram sua participação);</p> <p>2. Não Apto (Pontos de Venda não participantes);</p> <p>3. Apto (Pontos de Venda aptos, mas o assessor ainda não realizou negociação);</p> <p>4. Termo Gerado (Pontos de Venda com o termo de aceite pendente de assinatura);</p> <p>5. Termo Aceito (Pontos de Venda participantes com termos já assinados).</p> <p>Obs.: A carga desta tabela será feita inicialmente por script.</p>

DW.DM_SITUACAO_PLANO_MARKETING Colunas

Nome da Coluna	Domínio	Tipo	NULL	Definição
<u>CD_SIT_PLAN_MK</u>		NUMBER(3, 0)	NO	Código da situação do cliente no Plano de Marketing. O código será um sequencial e servirá como chave da tabela. Ex.: 1 - Saídas, 2 - Não Apto, etc.
<u>NM_SIT_PLAN_MK</u>		VARCHAR2(50)	NO	Nome da situação do cliente no Plano de Marketing. O nome deve ser preenchido obrigatoriamente. Ex.: Saídas, Não Apto, etc.
<u>DS_SIT_PLAN_MK</u>		VARCHAR2(150)	NO	Descrição da situação do cliente no Plano de Marketing. A descrição deve ser preenchida obrigatoriamente. Ex.: Pontos de Venda que cancelaram sua participação, Pontos de Venda não participantes, etc.
<u>DT_INCL</u>		DATE	NO	Data de Inclusão. Contém a data em que o registro foi inserido na tabela e deve ser preenchido

				obrigatoriamente.
<u>DT ALTER</u>		DATE	YES	Data de Alteração. Contém a data em que o registro foi alterado na tabela e só será preenchido caso o registro sofra mudanças.
<u>DT INAT</u>		DATE	YES	Data de Inativação. Contém a data em que o registro foi inativado. Só deve estar preenchido caso o registro esteja inativo. Se o registro estiver ativo ou tiver sido reativado, não deve estar preenchido.
<u>DT REAT</u>		DATE	YES	Data de Reativação. Contém a data em que o registro foi reativado. Só deve estar preenchido caso o registro tenha sido reativado. Se o registro estiver inativo ou ativo sem ter sido reativado, não deve estar preenchido.
<u>CD USUARIO</u>		VARCHAR2(30)	NO	Código do Usuário. Contém o mnemônico do usuário que fez o cadastro do registro e deve ser preenchido obrigatoriamente.

DW.DM_PONTO_VENDA

CD_PTOV	NUMBER(8,0)	NOT NULL
CD_EMPR	VARCHAR2(5)	NOT NULL
NM_PTOV	VARCHAR2(40)	NOT NULL
CD_GV	VARCHAR2(12)	NULL
CD_GC	VARCHAR2(12)	NULL
CD_MUNIC	VARCHAR2(10)	NOT NULL
CD_PTOV_ANT	NUMBER(8,0)	NULL
DS_END	VARCHAR2(40)	NULL
CD_ZONVEN	VARCHAR2(12)	NOT NULL
NM_BAIRRO	VARCHAR2(40)	NULL
CD_CEP	VARCHAR2(12)	NULL
CD_TIPO_LOC	VARCHAR2(3)	NOT NULL
CD_TIPO_PERFIL_VND	VARCHAR2(3)	NOT NULL
CD_TIPO_IMAG	VARCHAR2(3)	NOT NULL
CD_PROPR_TERR	VARCHAR2(3)	NOT NULL
CD_MICMERC	NUMBER(8,0)	NOT NULL
ID_COMP_AMPM	VARCHAR2(1)	NOT NULL
CD_SEGTO_MERC	VARCHAR2(3)	NOT NULL
CD_STATUS_PTOV	VARCHAR2(1)	NOT NULL
CD_ASSESSOR_FRANQ	VARCHAR2(12)	NOT NULL
CD_PTOV_JDE	NUMBER(8,0)	NULL
ID_CLI_CTREC	CHAR(1)	NOT NULL
ID_PTOV_EXCL	CHAR(1)	NOT NULL
DT_INCL	DATE	NOT NULL
DT_ALTER	DATE	NULL
DT_INAT	DATE	NULL
DT_REAT	DATE	NULL
CD_CLASS_NEG_POSTO_SERV	NUMBER(2,0)	NULL
CD_DISTRIB_AUT	NUMBER(8,0)	NULL

DW.DM_PONTO_VENDA

Nome da Entidade Lógica	Dimensão Ponto de Venda
Nome da Tabela	DM_PONTO_VENDA
Owner	DW
Definição	<p>É um espaço físico gerador de receita para Ipiranga. O Ponto de Venda é formado por diversos Componentes. O Ponto de Venda está localizado em um endereço fixo. Assim, todos os componentes que pertencem ao mesmo ponto de venda, deverão então possuir este mesmo endereço.</p> <p>Esta tabela contém os clientes da empresa.</p> <p>Exemplo: Os componentes podem ser: Área de Abastecimento, Loja AmPm, Jet Oil, Banco 24 Horas, Borracharia, etc.</p>

DW.DM_PONTO_VENDA Colunas

Nome da Coluna	Domínio	Tipo	NULL	Definição
<u>CD_PTOV</u> 		NUMBER(8, 0)	NO	Código do ponto de venda é um sequencial.
<u>CD_EMPR</u>		VARCHAR2(5)	NO	Código da empresa.
<u>NM_PTOV</u>		VARCHAR2(40)	NO	Nome genérico do ponto de venda. Pode ser o nome do principal componente do ponto de venda ou outro nome pelo qual o ponto de venda é mais conhecido pela empresa.
<u>CD_GV</u>		VARCHAR2(12)	YES	Código da gerência de venda.
<u>CD_GC</u>		VARCHAR2(12)	YES	Código da gerência comercial.
<u>CD_MUNIC</u>		VARCHAR2(10)	NO	Código do município.
<u>CD_PTOV_ANT</u>		NUMBER(6, 0)	YES	Código do ponto de venda antigo.
<u>DS_END</u>		VARCHAR2(40)	YES	Descrição do endereço do ponto de venda.
<u>CD_ZONVEN</u>		VARCHAR2(12)	NO	Código da Zona de Venda.
<u>NM_BAIRRO</u>		VARCHAR2(40)	YES	Nome do bairro onde o ponto de venda está localizado.
<u>CD_CEP</u>		VARCHAR2(12)	YES	Código do CEP onde o ponto de venda está localizado.
<u>CD_TIPO_LOC</u>		VARCHAR2(3)	NO	Código do tipo de localização do ponto de venda. Valores válidos: C - Cidade E - Estrada '-' - Não Aplicável
<u>CD_TIPO_PERFIL_VND</u>		VARCHAR2(3)	NO	Código do tipo de perfil de vendas. Valores válidos : U - Urbano R - Rodovia M - Misto '-' - Não Aplicável

				É referente ao componente do tipo Pista.
<u>CD TIPO IMAG</u>		VARCHAR2(3)	NO	Código do tipo de imagem do ponto de venda Valores válidos: I - Ipiranga A - Atlantic NI - Nova Imagem '-' - Não Aplicável
<u>CD PROPR TERR</u>		VARCHAR2(3)	NO	Código de propriedade do terreno onde o ponto de venda está localizado (P - Próprio, T - Terceiros, NA - Não Aplicável).

<u>CD MICMERC</u>	NUMBER(6, 0)	NO	Código do micro-mercado utilizado pelo JDE.
<u>ID COMP AMPM</u>	VARCHAR2(1)	NO	Identifica se o ponto de venda possui alguma loja franqueada do tipo AMPM FULL (código tipo componente = 2) ou AMPM EXPRESS (código tipo componente = 3). S = Sim possui Loja N = Não possui Loja
<u>CD SEGTO MERC</u>	VARCHAR2(3)	NO	Código do segmento de mercado. REV - Revenda CON - Consumo TRR - Rodovia CNG - Congênera

<u>CD_STATUS_PTOV</u>	VARCHAR2(1)	NO	Código do status do componente. Valores válidos : A - Ativo I - Inativo EX - Ex-cliente
<u>CD_ASSESSOR_FRANQ</u>	VARCHAR2(12)	NO	Código do Assessor de franquia
<u>CD_PTOV_JDE</u>	NUMBER(8, 0)	YES	Código do ponto de venda do JDE utilizado após a implantação do JDE.
<u>ID_CLI_CTREC</u>	CHAR(1)	NO	Identifica se o cliente Contas a Receber Este Identificador é provisório até construirmos uma tabela de Clientes para o Contas a Receber
<u>ID_PTOV_EXCL</u>	CHAR(1)	NO	Identifica o Ponto de Venda a ser excluído. Se esta coluna estiver preenchida com 'N', significa que o ponto de venda não será excluído. Se esta coluna estiver preenchida com 'S' ou 'P', significa que o ponto de venda será excluído.
<u>DT_INCL</u>	DATE	NO	Data da inclusão do registro.
<u>DT_ALTER</u>	DATE	YES	Data da última alteração efetuada no registro (inclusão, alteração, inativação, etc)
<u>DT_INAT</u>	DATE	YES	Data da inativação do registro.
<u>DT_REAT</u>	DATE	YES	Data de reativação do registro no DW.
<u>CD_CLASS_NEG_POSTO_SERV</u>	NUMBER(2, 0)	YES	
<u>CD_DISTRIB_AUT</u>	NUMBER(6, 0)	YES	

DW.PR_PONTO_VENDA_PLANO_MARKETING

CD_PTOV (FK)	NUMBER(8,0)	NOT NULL
CD_PROG_MK (FK)	NUMBER(3,0)	NOT NULL
NO_EDICAO_PROG_MK (FK)	NUMBER(3,0)	NOT NULL
CD_SIT_PLAN_MK (FK)	NUMBER(3,0)	NOT NULL
VR_OBJETIVO_PGTO_PART	NUMBER(9,2)	NULL
VR_NEGOC_PGTO_PART	NUMBER(9,2)	NULL
QT_PARC_PGTO	NUMBER(2,0)	NULL
ID_PART	CHAR(1)	NULL
ID_KIT	VARCHAR2(3)	NULL
DS_OBS	VARCHAR2(200)	NULL
DT_EMIS_TERMOS_ADESAO	DATE	NULL
DT_ACEITE_TERMOS_ADESAO	DATE	NULL
DT_INCL_PLAN_MK	DATE	NULL
DT_ALTER_PLAN_MK	DATE	NULL
CD_USUARIO_ACEITE_TERMOS_ADESAO	VARCHAR2(30)	NULL
CD_USUARIO_PLAN_MK	VARCHAR2(30)	NULL
CD_USUARIO_EMIT_TERMOS_ADESAO	VARCHAR2(30)	NULL
DT_INAT_PLAN_MK	DATE	NULL
CD_USUARIO_INAT	VARCHAR2(30)	NULL
CD_USUARIO_ACEITE_CAD_SOCIO	VARCHAR2(30)	NULL
DT_ACEITE_CAD_SOCIO	DATE	NULL
CD_USUARIO_ACEITE_CAD_FUNC	VARCHAR2(30)	NULL
DT_ACEITE_CAD_FUNC	DATE	NULL
DT_INCL	DATE	NOT NULL

DW.PR_PONTO_VENDA_PLANO_MARKETING

Nome da Entidade Lógica	Pr Ponto de Venda Plano de Marketing
Nome da Tabela	PR_PONTO_VENDA_PLANO_MARKETING
Owner	DW
Definição	<p>Contém os pontos de venda aptos a participar do programa de marketing Plano de Marketing (Adesão Unica).</p> <p>O programa de marketing Plano de Marketing (Adesão Unica) é o programa onde o cliente escolhe participar de todas ou nenhuma promoção.</p> <p>Obs.: A carga desta tabela será feita por ETL's desenvolvidos no PowerCenter. As duas últimas edições do Programa de Marketing Plano de Marketing (Adesão Única) serão atualizadas diariamente. Isso se faz necessário, pois, apesar dos usuários não poderem atualizar uma edição passada, há períodos no ano em que apesar de já existir uma nova edição cadastrada, ainda estamos utilizando a anterior.</p>

DW.PR_PONTO_VENDA_PLANO_MARKETING Colunas

Nome da Coluna	Domínio	Tipo	NULL	Definição
<u>CD_PTOV</u> 		NUMBER(8, 0)	NO	<p>Código do Ponto de Venda. O código será um sequencial.</p> <p>Estratégia de carga: Cópia simples.</p>
<u>CD_PROG_MK</u> 		NUMBER(3, 0)	NO	<p>Código do Programa de Marketing. O código será um sequencial e servirá como chave da tabela.</p> <p>Obs.: Como a tabela contempla apenas o programa Plano de Marketing, o valor será fixo.</p>

				<p>Ex.: 14 - Plano de Marketing (Adesão Única)</p> <p>Estratégia de carga: Cópia simples.</p>
<u>NO EDICAO PROG MK</u> 		NUMBER(3, 0)	NO	<p>Número da Edição do Programa de Marketing. O número será um seqüencial que faz parte da chave e contém as edições dos Programas de Marketing.</p> <p>Ex.: 1 – Plano de Marketing 2007 2 – Plano de Marketing 2008 3 – Plano de Marketing 2009</p> <p>Estratégia de carga: Cópia simples.</p>
<u>CD SIT PLAN MK</u> 		NUMBER(3, 0)	NO	<p>Código da situação do cliente no Plano de Marketing. O código será um seqüencial.</p> <p>Ex.: 1 - Saídas, 2 - Não Apto, etc.</p> <p>Estratégia de carga: Valor fixo. Cada ETL deve tratar de uma situação, implementando uma regra de negócio específica.</p>
<u>VR OBJETIVO PGTO PART</u>		NUMBER(9, 2)	YES	<p>Valor objetivado para o pagamento pelo ponto de venda para sua participação no programa de Marketing Plano de Marketing</p>

				(Adesão Única). Estratégia de carga: Cópia simples.
<u>VR NEGOC PGTO PART</u>		NUMBER(9, 2)	YES	Valor negociado pelo assessor de vendas para o pagamento pelo ponto de venda para sua participação no programa de Marketing Plano de Marketing (Adesão Única). Pode ser igual, maior e menor que o objetivado. Estratégia de carga: Cópia simples.

<u>QT_PARC_PGTO</u>	NUMBER(2, 0)	YES	Quantidade de parcelas do pagamento para participação do Ponto de venda no programa Plano de Marketing (Adesão Única). Estratégia de carga: Cópia simples.
<u>ID_PART</u>	CHAR(1)	YES	Identifica participação pode ser: S = Sim N = Não O Ponto de Venda pode estar apto a participar do programa Plano de Marketing (Adesão Única), mas até o momento não manifestou seu interesse. Estratégia de carga: Cópia simples.
<u>ID_KIT</u>	VARCHAR2(3)	YES	Identifica o kit do programa Adesão Única. Pode ser: 'ROD' Rodovia ou 'URB' = Urbano Estratégia de carga: Cópia simples.

<u>DS_OBS</u>	VARCHAR2(200)	YES	Descrição da observação sobre o Programa de Marketing para aquele Ponto de Venda. Estratégia de carga: Cópia simples.
<u>DT_EMIS_TERMOS_ADESAO</u>	DATE	YES	Data de Emissão do termo de adesão do ponto de venda ao programa de marketing Plano de Marketing (Adesão Única). Estratégia de carga: Cópia simples.
<u>DT_ACEITE_TERMOS_ADESAO</u>	DATE	YES	Data do aceite da adesão do ponto de venda ao programa de marketing Plano de Marketing (Adesão Única). Estratégia de carga: Cópia simples.
<u>DT_INCL_PLAN_MK</u>	DATE	YES	Data de Inclusão. Contém a data em que o registro foi inserido na tabela de Plano de Marketing de Sistemas de Negócio e deve ser preenchido obrigatoriamente. Estratégia de carga: Cópia simples.
<u>DT_ALTER_PLAN_MK</u>	DATE	YES	Data de Alteração. Contém a data em que o registro foi alterado na tabela de Plano de Marketing de Sistemas de Negócio e só será preenchido caso o registro sofra mudanças. Estratégia de carga: Cópia simples.

<u>CD_USUARIO_ACEITE_TERMOS_ADESAO</u>	VARCHAR2(30)	YES	Código do Usuário que aceitou o Termo de Adesão. Estratégia de carga: Cópia simples.
<u>CD_USUARIO_PLAN_MK</u>	VARCHAR2(30)	YES	Código do Usuário. Contém o mnemônico do usuário que fez o cadastro do registro na tabela de Plano de Marketing de Sistemas de Negócios e deve ser preenchido obrigatoriamente. Estratégia de carga: Cópia simples.
<u>CD_USUARIO_EMIT_TERMOS_ADESAO</u>	VARCHAR2(30)	YES	Código do Usuário que emitiu o Termo de Adesão. Estratégia de carga: Cópia simples.
<u>DT_INAT_PLAN_MK</u>	DATE	YES	Data de Inativação. Contém a data em que o registro foi inativado na tabela de Plano de Marketing de Sistemas de Negócios. Só deve estar preenchido caso o registro esteja inativo. Se o registro estiver ativo ou tiver sido reativado, não deve estar preenchido. Estratégia de carga: Cópia simples.

<u>CD_USUARIO_INAT</u>	VARCHAR2(30)	YES	Código do Usuário que efetuou mudanças no registro. Estratégia de carga: Cópia simples.
<u>CD_USUARIO_ACEITE_CAD_SOCIO</u>	VARCHAR2(30)	YES	Código do Usuário que cadastrou o sócio para o Termo de Aceite. Estratégia de carga: Cópia simples.
<u>DT_ACEITE_CAD_SOCIO</u>	DATE	YES	Data em que o sócio foi cadastrado para o Termo de Aceite. Estratégia de carga: Cópia simples.
<u>CD_USUARIO_ACEITE_CAD_FUNC</u>	VARCHAR2(30)	YES	Código do Usuário que cadastrou o funcionário para o Termo de Aceite. Estratégia de carga: Cópia simples.
<u>DT_ACEITE_CAD_FUNC</u>	DATE	YES	Data em que o funcionário foi cadastrado para o Termo de Aceite. Estratégia de carga: Cópia simples.

<u>DT INCL</u>		DATE	NO	<p>Data de Inclusão. Contém a data em que o registro foi inserido na tabela e deve ser preenchido obrigatoriamente.</p> <p>Estratégia de carga: Data do sistema no momento da carga. Deve ser igual para todos os registros que estão sendo carregados no mesmo ETL.</p>
----------------	--	------	----	--

APÊNDICE B

PR_PONTO_VENDA_PLANO_MARKETING

```

-- Create table
create table DW.PR_PONTO_VENDA_PLANO_MARKETING
(
  CD_PTOV                NUMBER(8) not null,
  CD_PROG_MK             NUMBER(3) not null,
  NO_EDICAO_PROG_MK     NUMBER(3) not null,
  CD_SIT_PLAN_MK        NUMBER(3) not null,
  VR_OBJETIVO_PGTO_PART NUMBER(9,2),
  VR_NEGOC_PGTO_PART    NUMBER(9,2),
  QT_PARC_PGTO          NUMBER(2),
  ID_PART               CHAR(1),
  ID_KIT                VARCHAR2(3),
  DS_OBS                VARCHAR2(200),
  DT_EMIS_TERMOS_ADESAO DATE,
  DT_ACEITE_TERMOS_ADESAO DATE,
  DT_INCL_PLAN_MK      DATE,
  DT_ALTER_PLAN_MK     DATE,
  CD_USUARIO_ACEITE_TERMOS_ADESAO VARCHAR2(30),
  CD_USUARIO_PLAN_MK   VARCHAR2(30),
  CD_USUARIO_EMIT_TERMOS_ADESAO VARCHAR2(30),
  DT_INAT_PLAN_MK     DATE,
  CD_USUARIO_INAT     VARCHAR2(30),
  CD_USUARIO_ACEITE_CAD_SOCIO VARCHAR2(30),
  DT_ACEITE_CAD_SOCIO DATE,
  CD_USUARIO_ACEITE_CAD_FUNC VARCHAR2(30),
  DT_ACEITE_CAD_FUNC  DATE,
  DT_INCL              DATE not null
)
)
tablespace TABELA
pctfree 20
initrans 4
maxtrans 255
storage
(
  initial 128K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DW.PR_PONTO_VENDA_PLANO_MARKETING
  add constraint PK_PRPOVEPM primary key (CD_PTOV, CD_PROG_MK,
NO_EDICAO_PROG_MK)
  using index
  tablespace INDICE
  pctfree 20
  initrans 8
  maxtrans 255
  storage
  (
    initial 128K
    minextents 1
    maxextents unlimited
  );
alter table DW.PR_PONTO_VENDA_PLANO_MARKETING
  add constraint FK_PRPOVEPM_DMEDPRMA_01 foreign key (CD_PROG_MK,
NO_EDICAO_PROG_MK)

```

```

references DW.DM_EDICAO_PROGRAMA_MARKETING (CD_PROG_MK,
NO_EDICAO_PROG_MK);
alter table DW.PR_PONTO_VENDA_PLANO_MARKETING
add constraint FK_PRPOVEPM_DMPONVEN_01 foreign key (CD_PTOV)
references DW.DM_PONTO_VENDA (CD_PTOV);
alter table DW.PR_PONTO_VENDA_PLANO_MARKETING
add constraint FK_PRPOVEPM_DMSIPLMA_01 foreign key (CD_SIT_PLAN_MK)
references DW.DM_SITUACAO_PLANO_MARKETING (CD_SIT_PLAN_MK);
-- Create/Recreate check constraints
alter table DW.PR_PONTO_VENDA_PLANO_MARKETING
add constraint CK_ID_PART_01
check (ID_PART IN ('S', 'N'));
-- Create/Recreate indexes
create index DW.IF_PRPOVEPM_01 on DW.PR_PONTO_VENDA_PLANO_MARKETING
(CD_PROG_MK, NO_EDICAO_PROG_MK)
tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
initial 128K
minextents 1
maxextents unlimited
);
create index DW.IF_PRPOVEPM_02 on DW.PR_PONTO_VENDA_PLANO_MARKETING
(CD_SIT_PLAN_MK)
tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
initial 128K
minextents 1
maxextents unlimited
);
create index DW.IF_PRPOVEPM_03 on DW.PR_PONTO_VENDA_PLANO_MARKETING
(CD_PTOV)
tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
initial 128K
minextents 1
maxextents unlimited
);
);

```

DM_SITUACAO_PLANO_MARKETING

```

-- Create table
create table DW.DM_SITUACAO_PLANO_MARKETING
(
CD_SIT_PLAN_MK NUMBER(3) not null,
NM_SIT_PLAN_MK VARCHAR2(50) not null,
DS_SIT_PLAN_MK VARCHAR2(150) not null,
DT_INCL DATE not null,

```

```

DT_ALTER      DATE,
DT_INAT       DATE,
DT_REAT       DATE,
CD_USUARIO    VARCHAR2(30) not null
)
)
tablespace TABELA
pctfree 20
initrans 4
maxtrans 255
storage
(
  initial 16K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DW.DM_SITUACAO_PLANO_MARKETING
add constraint PK_DMSIPLMK primary key (CD_SIT_PLAN_MK)
using index
tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
  initial 16K
  minextents 1
  maxextents unlimited
);
);

```

DM_PROGRAMA_MARKETING

```

-- Create table
create table DW.DM_PROGRAMA_MARKETING
(
  CD_PROG_MK  NUMBER(3) not null,
  NM_PROG_MK  VARCHAR2(30) not null,
  DT_INI_PROG DATE,
  DT_FIM_PROG DATE,
  DT_INCL     DATE not null,
  DT_ALTER    DATE,
  DT_INAT     DATE,
  DT_REAT     DATE,
  CD_USUARIO  CHAR(8)
)
)
tablespace TABELA
pctfree 1
initrans 4
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DW.DM_PROGRAMA_MARKETING
add constraint PK_DMPROGMA primary key (CD_PROG_MK)
using index
tablespace INDICE

```

```

pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

DM_EDICAO_PROGRAMA_MARKETING

```

-- Create table
create table DW.DM_EDICAO_PROGRAMA_MARKETING
(
  CD_PROG_MK          NUMBER(3) not null,
  NO_EDICAO_PROG_MK  NUMBER(3) not null,
  NM_EDICAO_PROG_MK  VARCHAR2(40) not null,
  DT_INI_EDICAO_PROG_MK DATE not null,
  DT_FIM_EDICAO_PROG_MK DATE,
  DT_INCL            DATE not null,
  DT_ALTER           DATE,
  DT_INAT           DATE,
  DT_REAT           DATE,
  CD_USUARIO         VARCHAR2(30) not null
)
tablespace TABELA
pctfree 1
initrans 4
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DW.DM_EDICAO_PROGRAMA_MARKETING
  add constraint PK_DMEDPRMA primary key (CD_PROG_MK,
NO_EDICAO_PROG_MK)
  using index
  tablespace INDICE
  pctfree 99
  initrans 8
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
alter table DW.DM_EDICAO_PROGRAMA_MARKETING
  add constraint FK_DMEDPRMA_DMDIA_01 foreign key
(DT_FIM_EDICAO_PROG_MK)
  references DW.DM_DIA (DT_REF);
alter table DW.DM_EDICAO_PROGRAMA_MARKETING
  add constraint FK_DMEDPRMA_DMDIA_02 foreign key
(DT_INI_EDICAO_PROG_MK)
  references DW.DM_DIA (DT_REF);
alter table DW.DM_EDICAO_PROGRAMA_MARKETING

```

```

add constraint FK_DMEDPRMA_DMPROMAR_01 foreign key (CD_PROG_MK)
references DW.DM_PROGRAMA_MARKETING (CD_PROG_MK);
-- Create/Recreate indexes
create index DW.IF_DMEDPRMA_01 on DW.DM_EDICAO_PROGRAMA_MARKETING
(CD_PROG_MK)
  tablespace INDICE
  pctfree 99
  initrans 8
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
create index DW.IF_DMEDPRMA_02 on DW.DM_EDICAO_PROGRAMA_MARKETING
(DT_FIM_EDICAO_PROG_MK)
  tablespace INDICE
  pctfree 99
  initrans 8
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
create index DW.IF_DMEDPRMA_03 on DW.DM_EDICAO_PROGRAMA_MARKETING
(DT_INI_EDICAO_PROG_MK)
  tablespace INDICE
  pctfree 99
  initrans 8
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );

```

DM_PONTO_VENDA

```

-- Create table
create table DW.DM_PONTO_VENDA
(
  CD_EMPR          VARCHAR2(5) not null,
  CD_PTOV          NUMBER(8) not null,
  NM_PTOV          VARCHAR2(40) not null,
  CD_PTOV_ANT      NUMBER(6),
  CD_MUNIC         VARCHAR2(10) not null,
  DS_END           VARCHAR2(40),
  CD_ZONVEN        VARCHAR2(12) not null,
  NM_BAIRRO        VARCHAR2(40),
  CD_CEP           VARCHAR2(12),
  CD_TIPO_LOC      VARCHAR2(3) not null,
  CD_TIPO_PERFIL_VND VARCHAR2(3) not null,
  CD_TIPO_IMAG     VARCHAR2(3) not null,
  CD_PROPR_TERR    VARCHAR2(3) not null,
  ID_COMP_AMPM     VARCHAR2(1) not null,
  CD_SEGTO_MERC    VARCHAR2(3) not null,
  CD_STATUS_PTOV   VARCHAR2(1) not null,

```

```

CD_GC          VARCHAR2(12),
CD_GV          VARCHAR2(12),
DT_INCL       DATE not null,
DT_ALTER      DATE,
DT_INAT       DATE,
DT_REAT       DATE,
CD_ASSESSOR_FRANQ VARCHAR2(12) default '-1' not null,
ID_CLI_CTREC  CHAR(1) default 'N' not null,
CD_PTOV_JDE   NUMBER(8),
ID_PTOV_EXCL  CHAR(1) default 'N' not null,
CD_MICMERC    NUMBER(6) default '-1' not null,
CD_CLASS_NEG_POSTO_SERV NUMBER(2),
CD_DISTRIB_AUT NUMBER(6),
ID_COMP_JETOIL CHAR(1) default 'N' not null,
CD_PTOV_DPPI  NUMBER(8),
CD_TIPO_LOGR  VARCHAR2(36),
DS_LOGR       VARCHAR2(100),
NO_LOGR       VARCHAR2(20),
DS_COMPM_LOGR VARCHAR2(50),
CD_EMPR_ORIG  NUMBER(3)
)
)
tablespace TABELA
pctfree 1
initrans 4
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table DW.DM_PONTO_VENDA
add constraint FK_DMPONVEN primary key (CD_PTOV)
using index
tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
alter table DW.DM_PONTO_VENDA
add constraint FK_DMPONVEN_DMASSFRA_01 foreign key
(CD_ASSESSOR_FRANQ)
references DMFRANQ.DM_ASSESSORIA_FRANQUIA (CD_ASSESSOR_FRANQ);
alter table DW.DM_PONTO_VENDA
add constraint FK_DMPONVEN_DMCLNEPS_01 foreign key
(CD_CLASS_NEG_POSTO_SERV)
references DW.DM_CLASS_NEGOCIO_POSTO_SERVICO
(CD_CLASS_NEG_POSTO_SERV);
alter table DW.DM_PONTO_VENDA
add constraint FK_DMPONVEN_DMEMPORI_01 foreign key (CD_EMPR_ORIG)
references DW.DM_EMPRESA_ORIGEM (CD_EMPR_ORIG);
alter table DW.DM_PONTO_VENDA
add constraint FK_DMPONVEN_DMEMPRES_01 foreign key (CD_EMPR)
references DW.DM_EMPRESA (CD_EMPR);
alter table DW.DM_PONTO_VENDA

```

```

    add constraint FK_DMPONVEN_DMGERCOM_01 foreign key (CD_GC)
    references DW.DM_GERENCIA_COMERCIAL (CD_GC);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMGERVEN_01 foreign key (CD_GV)
    references DW.DM_GERENCIA_VENDA (CD_GV);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMMICMER_01 foreign key (CD_MICMERC)
    references DW.DM_MICRO_MERCADO (CD_MICMERC);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMMUNICI_01 foreign key (CD_MUNIC)
    references DW.DM_MUNICIPIO (CD_MUNIC);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMPROTER_01 foreign key (CD_PROPR_TERR)
    references DW.DM_PROPRIEDADE_TERRENO (CD_PROPR_TERR);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMSEGMER_01 foreign key (CD_SEGTO_MERC)
    references DW.DM_SEGMENTO_MERCADO (CD_SEGTO_MERC);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMTIPEVE_01 foreign key
(CD_TIPO_PERFIL_VND)
    references DW.DM_TIPO_PERFIL_VENDA (CD_TIPO_PERFIL_VND);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMTIPIMA_01 foreign key (CD_TIPO_IMAG)
    references DW.DM_TIPO_IMAGEM (CD_TIPO_IMAG);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMTIPLIC_01 foreign key (CD_TIPO_LOC)
    references DW.DM_TIPO_LOCALIZACAO (CD_TIPO_LOC);
alter table DW.DM_PONTO_VENDA
    add constraint FK_DMPONVEN_DMZONVEN_01 foreign key (CD_ZONVEN)
    references DW.DM_ZONA_VENDA (CD_ZONVEN);
-- Create/Recreate indexes
create bitmap index DW.IF_DMPONVEN_01 on DW.DM_PONTO_VENDA
(CD_MICMERC)
    tablespace INDICE
    pctfree 99
    initrans 4
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_02 on DW.DM_PONTO_VENDA (CD_ZONVEN)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_03 on DW.DM_PONTO_VENDA (CD_EMPR)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (

```

```

        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_04 on DW.DM_PONTO_VENDA
(CD_SEGTO_MERC)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_05 on DW.DM_PONTO_VENDA
(CD_TIPO_LOC)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_06 on DW.DM_PONTO_VENDA
(CD_TIPO_IMAG)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_07 on DW.DM_PONTO_VENDA
(CD_PROPR_TERR)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_08 on DW.DM_PONTO_VENDA
(CD_TIPO_PERFIL_VND)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K

```

```

        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_09 on DW.DM_PONTO_VENDA (CD_MUNIC)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_10 on DW.DM_PONTO_VENDA (CD_GC)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_11 on DW.DM_PONTO_VENDA (CD_GV)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_12 on DW.DM_PONTO_VENDA
(CD_ASSESSOR_FRANQ)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create bitmap index DW.IF_DMPONVEN_13 on DW.DM_PONTO_VENDA
(CD_CLASS_NEG_POSTO_SERV)
    tablespace INDICE
    pctfree 99
    initrans 8
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
create index DW.IF_DMPONVEN_14 on DW.DM_PONTO_VENDA (CD_EMPR_ORIG)

```

```

tablespace INDICE
pctfree 20
initrans 8
maxtrans 255
storage
(
  initial 336K
  minextents 1
  maxextents unlimited
);
create bitmap index DW.IS_DMPONVEN_01 on DW.DM_PONTO_VENDA
(CD_PTOV_ANT)
tablespace INDICE
pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
create bitmap index DW.IS_DMPONVEN_02 on DW.DM_PONTO_VENDA
(CD_PTOV_JDE)
tablespace INDICE
pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
create index DW.IS_DMPONVEN_03 on DW.DM_PONTO_VENDA (CD_EMPR, CD_PTOV,
CD_SEGTO_MERC)
tablespace INDICE
pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
create bitmap index DW.IS_DMPONVEN_04 on DW.DM_PONTO_VENDA
(ID_PTOV_EXCL)
tablespace INDICE
pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
create index DW.IS_DMPONVEN_05 on DW.DM_PONTO_VENDA (CD_MUNIC,
CD_PTOV, CD_EMPR)
tablespace INDICE

```

```
pctfree 99
initrans 8
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
```