

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

CRIAÇÃO DE *CUSTOMER RELATIONSHIP MANAGEMENT*
(CRM) APLICÁVEL A SISTEMAS COMERCIAIS PARA O
SEGMENTO DE TURISMO

PETERSON MARTINS GONÇALVES

BLUMENAU
2004

2004/2-40

PETERSON MARTINS GONÇALVES

**CRIAÇÃO DE *CUSTOMER RELATIONSHIP MANAGEMENT*
(CRM) APLICÁVEL A SISTEMAS COMERCIAIS PARA O
SEGMENTO DE TURISMO**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho de
Conclusão de Curso II do curso de Ciência da
Computação — Bacharelado.

Prof. Alexander Roberto Valdameri - Orientador

**BLUMENAU
2004**

2004/2-40

**CRIAÇÃO DE *CUSTOMER RELATIONSHIP MANAGEMENT*
(CRM) APLICÁVEL A SISTEMAS COMERCIAIS PARA O
SEGMENTO DE TURISMO**

Por

PETERSON MARTINS GONÇALVES

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Alexander Roberto Valdameri – Orientador, FURB

Membro: _____
Prof. Carlos Eduardo Negrão Bizzotto

Membro: _____
Prof. Marcel Hugo

Blumenau, 15 de novembro de 2004

Dedico este trabalho a todos os amigos,
especialmente aqueles que me ajudaram
diretamente na realização deste.

“As pessoas aprendem melhor aquilo que é significativo para elas”

Lev Vygotsky

AGRADECIMENTOS

A Deus, pelo seu imenso amor e por estar sempre presente em minha vida.

À minha família, que mesmo longe, sempre esteve presente e que não mediu esforços para ajudar-me e apoiar-me durante minha formação acadêmica.

Aos meus amigos, pelos momentos de descontração e companheirismo.

À empresa Senior Sistemas, na qual trabalho, por disponibilizarem recursos e apoiarem a realização deste trabalho.

Ao meu orientador, Alexander Roberto Valdameri, por ter acreditado na conclusão deste trabalho.

À minha namorada, Andresa Jaqueline Toassi, pelo incentivo e compreensão nos momentos de ausência devido à realização deste trabalho.

RESUMO

Na sociedade atual tornam-se cada vez maiores as disputas comerciais entre concorrentes de um determinado segmento de negócio, fazendo com que as empresas que possuam um melhor conhecimento de seus clientes e da situação do mercado possam obter um retorno mais significativo de seus investimentos nas diversas áreas de vendas. Nesse contexto, o presente relatório visa apresentar a implementação de um protótipo de software utilizando conceitos de *Customer Relationship Management* (CRM) ou gerenciamento de relacionamento com clientes baseado em *Data Warehouse* (DW), com o objetivo de orientar o esforço de vendas de uma agência de turismo. Para isso, a base de dados utilizada para dar suporte à construção do modelo de dados foi obtida a partir do Sistema de Informação Transacional denominado Regente, da empresa Senior Sistemas, o qual tem como principal intuito a automação comercial no segmento de turismo. A partir da extração de dados da base operacional realizou-se o desenvolvimento de um modelo dimensional no qual foram realizadas consultas dinâmicas e relatórios, disponibilizando-se informações gerenciais sobre as vendas para clientes da agência de turismo. Tais informações estando em posse dos responsáveis pela mesma, possibilitarão melhores condições para tomar decisões estratégicas com o intuito de obter melhores resultados em vendas.

Palavras chaves: Clientes; CRM; *Data Warehouse*.

ABSTRACT

In the current society the commercial disputes between competitors of a one determined business segment, making each time bigger with the companies who possess one better knowledge of its customers and by the market's situation they can get a more significant return of their investments in the diverse areas of sales. In this context, the present report aims to present the implementation of a software archetype being used concepts of Customer Relationship Management (CRM) or management of relationship with customers based in Data Warehouse (DW), with the purpose of guiding the effort of sales in a travel agency. For this, the database used to give support for the construction of the data's model was gotten from the Transaccional Information System called Regent, of the company Senior Systems, which has as the main objective the commercial automation in the tourism segment. From the data's extraction of the operational base the development of a dimensional model in which was become fulfilled had been realized in dynamic consultations and reports, giving itself a managerial information on sales for the customers of the travel agency. This information in ownership the responsible for this, that'll make possible better conditions to take strategic decisions with the intention of getting a better result in sales.

Key-Words: Customer; CRM; Data Warehouse

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Contexto	22
Figura 2 – Diagramas de Fluxo de Dados	22
Figura 3 – Diagramas de Fluxo de Dados	23
Figura 4 – MER lógico do modelo multidimensional	24
Figura 5 – Interface do aplicativo <i>Enterprise Manager</i>	25
Quadro 1 – Exemplo de código utilizando <i>tQuery</i>	27
Figura 6 – <i>Pallet</i> de componentes <i>Decision Cube</i>	28
Quadro 2 – Código gerado automaticamente pelo componente <i>DecisionQuery</i>	28
Figura 7 – Menu principal do protótipo.....	29
Figura 8 – Interface de definição de período para importação dos dados.	30
Figura 9 – Interface para realização de consultas dinâmicas.....	31
Figura 10 – Interface para visualização dos gráficos de consultas dinâmicas.....	32
Figura 11 – Interface para salvar consultas dinâmicas.....	33
Figura 12 – Interface para recuperação de consultas dinâmicas.	33
Figura 13 – Exemplo de relatório emitido através do protótipo de software.	34

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 OBJETIVOS DO TRABALHO	11
1.2 ESTRUTURA DO TRABALHO	12
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 AGÊNCIAS DE TURISMO.....	13
2.2 BANCO DE DADOS RELACIONAIS	13
2.3 MODELAGEM MULTIDIMENSIONAL.....	15
2.4 <i>DATA MART, DATA MINING E DATA WAREHOUSE</i>	17
2.5 CRM – <i>CUSTOMER RELATIONSHIP MANAGEMENT</i>	19
3 DESENVOLVIMENTO DO TRABALHO	21
3.1 REQUISITOS PRINCIPAIS	21
3.2 ESPECIFICAÇÃO	21
3.3 IMPLEMENTAÇÃO	24
3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	25
3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	29
3.4 RESULTADOS E DISCUSSÃO	35
4 CONCLUSÕES	36
4.1 EXTENSÕES	37
REFERÊNCIAS BIBLIOGRÁFICAS	38
APÊNDICE A – MER Lógico da base de dados operacional.....	39
APÊNDICE B – Nomenclatura das tabelas.....	40
APÊNDICE C – Script para extração dos dados da base operacional	41
APÊNDICE D – Dicionários de dados das tabelas do modelo dimensional.....	44

1 INTRODUÇÃO

No atual mundo globalizado, onde a concorrência entre as empresas fica a cada dia mais acirrada, é fundamental que qualquer organização com fins lucrativos, seja de pequeno, médio ou grande porte, consiga administrar seus recursos e investimentos de maneira tal, que possa obter o maior retorno possível de todo capital despendido.

Uma estratégia freqüentemente adotada pelo mercado atual para atingir esse objetivo é não só conquistar novos clientes, mas também estreitar seus relacionamentos com os já existentes. Para isso faz-se necessário um conhecimento mais profundo de suas principais características, suas preferências, seu poder de compra, enfim, de seu perfil para os negócios.

Com o desenvolvimento dos recursos de informática tornou-se realidade a possibilidade de armazenar informações referentes aos contatos e negócios mantidos entre as empresas e seus clientes. Porém, com o decorrer do tempo, o volume desses dados tomou proporções enormes, dificultando e exigindo maior concentração de trabalho por parte dos especialistas de negócio para obter uma boa análise do relacionamento com sua clientela. Para superação dessas dificuldades, tornou-se muito interessante desenvolver ferramentas que auxiliem nas tomadas de decisões a partir do conhecimento e análise desses dados.

Segundo Kimball e Ross (2002, p. 3) é muito comum ocorrer situações onde duas pessoas apresentem os mesmos indicadores utilizando números diferentes dentro de uma mesma organização. Atualmente, a maioria das empresas não possui sistemas que tratem seus dados com a finalidade de levantar informações gerenciais, as quais estando em poder de pessoas capacitadas, podem direcionar a estratégia de *marketing* da organização.

Uma abordagem largamente empregada para atingir esse fim é o *Customer Relationship Management* (CRM) ou gerenciamento de relacionamento com clientes, uma técnica que envolve conceitos, metodologias e objetivos que serão apresentados posteriormente.

O presente trabalho embasa, fundamenta e descreve a implementação de um protótipo de software, utilizando conceitos de CRM, o qual faz o levantamento de informações gerenciais, a partir de uma base de dados populada de uma agência de turismo. O mesmo foi desenvolvido utilizando recursos de *Data Warehouse* (DW) tais como *data mart* e *data*

mining, implementados através da ferramenta de desenvolvimento *Delphi* utilizando a linguagem de programação *Object Pascal* e o banco de dados Microsoft SQL Server. Objetiva-se, com isso, auxiliar e direcionar esforços de vendas e estratégias de *marketing* a serem adotadas pela agência de turismo, evitando assim, investimentos que possam não apresentar os resultados esperados comercialmente.

A utilização de um DW foi relevante para o desenvolvimento deste trabalho em consequência das características de um CRM, o qual faz uso de dados históricos, bem representados nesta filosofia (DW) de armazenamento de dados.

Cabe ressaltar que a base de dados, a qual serviu de suporte à construção do modelo de dados, foi obtida a partir do Sistema de Informação Transacional Regente da Empresa Senior Sistemas, o qual é voltado para a automação comercial do segmento de turismo.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um protótipo de CRM aplicável a sistemas comerciais utilizados em agências de turismo, possibilitando consultas dinâmicas e emissão de relatórios com informações gerenciais.

Os objetivos específicos do trabalho são:

- a) analisar aspectos de normalização da base de dados;
- b) disponibilizar modelo dimensional para a base de dados;
- c) disponibilizar DW para dar suporte ao CRM na análise dos dados referentes aos processos comerciais da agência de turismo;
- d) disponibilizar interface para consulta às informações geradas;
- e) disponibilizar estudos de casos em base de dados pré-definida (em ambiente de produção, disponibilizada e com utilização autorizada pela empresa Senior Sistemas em Turismo).

1.2 ESTRUTURA DO TRABALHO

Este trabalho apresenta-se em capítulos conforme segue:

No primeiro capítulo é feita uma introdução com o objetivo de apresentar aos leitores qual assunto o trabalho refere-se e seus principais objetivos.

No segundo capítulo são apresentadas as fundamentações teóricas procurando disponibilizar ao leitor uma breve referência bibliográfica sobre os principais tópicos do trabalho desenvolvido.

No terceiro capítulo é explanado como ocorreu o desenvolvimento do trabalho, demonstrando os principais requisitos do protótipo de software desenvolvido, assim como sua especificação e de que forma foi realizada sua implementação. Apresentam-se, também, quais técnicas e ferramentas foram utilizadas para o desenvolvimento do mesmo e as principais operacionalidades do protótipo implementado, demonstrando quais foram os resultados obtidos.

No quarto capítulo são apresentadas as conclusões alcançadas, as limitações do protótipo desenvolvido e sugestões de extensões para dar continuidade em pesquisas na área do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Para facilitar a compreensão da finalidade deste trabalho, foram elencados alguns conceitos que são apresentados em formas de tópicos nesta seção, sendo eles: agências de turismo, banco de dados relacionais; modelagem dimensional; *data mart*; *data mining*; DW e CRM.

2.1 AGÊNCIAS DE TURISMO

As agências de turismo podem ser definidas, resumidamente, como sendo uma mediadora nas negociações entre os consumidores (viajantes, hóspedes, etc) e os fornecedores de produtos e serviços (companhias aéreas, hotéis, etc) dos segmentos de turismo. Seu papel vem sofrendo constantes mudanças devido às adaptações que esse mercado necessitou efetuar para manter-se ativo. Há cerca de duas décadas, as agências de turismo apresentavam-se como apenas emissoras de documentos para os clientes após terem sido realizadas as negociações com os fornecedores.

Atualmente, atuam como consultoras de turismo, sugerindo alternativas e opções que se assemelhem mais com o perfil de cada consumidor e estabelecendo várias fontes de comunicação com seus fornecedores. Suas atividades podem ser descritas, resumidamente, como sendo o atendimento a clientes (verificando quais são seus interesses), partindo em seguida para a demonstração das opções que podem atender essas necessidades e efetivando a reserva com os fornecedores dos segmentos de turismo, emitindo finalmente, documentos de cobrança para o cliente.

Com o aumento da complexidade para a administração dessas informações, surgiram os softwares para auxiliarem nessas tarefas, os quais, em sua maioria, armazenam seus dados em banco de dados relacionais.

2.2 BANCO DE DADOS RELACIONAIS

Para o desenvolvimento de uma ferramenta de apoio à tomada de decisões, baseada em DW, faz-se necessária a utilização de um banco de dados relacional, o qual pode ser definido como um conjunto de tabelas (as quais relacionam-se entre si) compostas de linhas e colunas,

onde as linhas contêm os registros e as colunas representam os campos das tabelas a serem preenchidos.

Um banco de dados relacional é percebido pelo usuário como um conjunto de tabelas. Uma tabela ou relação é composta por linhas conhecidas como tuplas (lê-se “tuplas”) e colunas. Cada coluna tem um conjunto de valores possíveis chamado domínio. Na linguagem da computação, tabela ou relação equivale a arquivo, t-upla equivale a registro e coluna equivale a campo. (KERN, 1994, p. 77).

De acordo com Inmon (1997, p. IX), surgiu um novo conceito de banco de dados nos últimos anos, o qual distingue-os em duas áreas, a primeira responsável em atender às necessidades operacionais e a segunda com a finalidade de suprir as necessidades informacionais ou analíticas. Tal divisão ocorre devido a diversas razões, dentre elas, pode-se citar como principais: os dados operacionais são fisicamente diferentes dos dados informacionais ou analíticos; dados operacionais são utilizados por comunidade de usuários diferente da que é atendida pelos dados analíticos e o processamento do ambiente operacional e informacional são, fundamentalmente, diferentes.

Conforme Kimball e Ross (2002, p. 02), os usuários de dados operacionais determinam o rumo da empresa através das anotações de pedidos, inscrições de novos clientes e registros de reclamações enquanto que os usuários de dados analíticos observam o rumo que a empresa toma contabilizando os novos pedidos, comparando-os com os da semana anterior e analisando por que novos clientes se associaram e por que clientes fizeram reclamações.

Caso os relacionamentos entre as tabelas do banco de dados não sejam bem estudados, analisados e projetados, pode-se gerar redundância de dados o que, invariavelmente, afeta o desempenho do sistema de gerenciamento de banco de dados (SGBD). Para evitar esse tipo de problema foi definido um processo de ajustes de tabelas (denominado normalização), seguindo regras práticas as quais são conhecidas como formas normais, para simplificação das mesmas.

Segundo Kimball e Ross (2002, p. 14) a modelagem normalizada é útil para o desempenho do processamento operacional, pois uma transação de atualização ou inserção necessita atingir um banco de dados em apenas um local, porém prejudica toda a finalidade do DW, devido à sua complexidade, caso seja utilizada em sua área de apresentação, principalmente a recuperação intuitiva e de alto desempenho dos dados. A partir dessa

constatação foi desenvolvida uma técnica para projetar a base de dados, a ser utilizada em um aplicativo desse gênero, denominada de modelagem dimensional.

2.3 MODELAGEM MULTIDIMENSIONAL

Conforme Serra (2002, p. 43), trata-se de uma técnica para projetar banco de dados bastante utilizada para DW, onde os dados são organizados em cubos, sendo que as informações a serem recuperadas residem na interseção de suas arestas.

Existe uma terminologia específica para tratar a modelagem dimensional que, segundo Kimball e Ross (2002, p. 20), teve origem em um projeto de pesquisa em conjunto entre a empresa americana General Mills e pela Dartmouth University na década de 1960. Contidos nessa terminologia encontram-se os termos: tabela de fatos, tabela de dimensões e granularidade.

De acordo com Kimball e Ross (2002, p. 21) pode-se definir tabela de fatos como sendo a principal tabela de um modelo dimensional, na qual são armazenadas as medições numéricas de desempenho da empresa, sendo que, cada medição de negócio corresponde a uma linha da tabela de fatos. Normalmente a chave primária de uma tabela de fatos é composta por um subconjunto de chaves estrangeiras, o que geralmente é denominada de chave composta ou concatenada. Em um modelo dimensional toda tabela de fatos possui uma chave composta e pode-se afirmar que tabelas que expressem a relação de muitos para muitos devem ser tabelas de fatos, sendo que as demais deverão ser tabelas de dimensão na modelagem dimensional.

Conforme Kimball e Ross (2002, p. 24) as tabelas de dimensão sempre acompanham uma tabela de fatos e possuem descritores textuais da empresa. Cada dimensão é definida por sua chave primária, a qual serve para a integridade referencial para a tabela de fatos a que esteja associada e, quando a tabela de dimensão encontra-se em modelos dimensionais bem projetados, apresenta diversas colunas e atributos. Esses, denominados de atributos dimensionais são a origem de praticamente todas as restrições e rótulos de relatórios interessantes. “As tabelas de dimensão são pontos de entrada para a tabela de fatos. Atributos de dimensões eficazes produzem recursos de separação e combinação (*slicing and dicing*) analíticos eficazes. As dimensões implementam a interface de usuário para o DW.” (KIMBALL; ROSS, 2002, p. 25)

Inmon (1997, p. 45) classifica a granularidade como sendo o principal aspecto do projeto de um DW, pois está diretamente relacionada com o nível de detalhe do mesmo, ou seja, quanto mais detalhe (dados históricos resumidos), mais baixo o nível de granularidade e quanto menos detalhe (dados levemente resumidos), mais alto o nível de granularidade. Tal classificação baseia-se no fato de que a granularidade afeta profundamente o volume de dados do DW, assim como o tipo de consulta a ser atendida, fazendo com que seja feito um balanceamento desse volume de acordo com o nível de detalhe de cada consulta.

Pode-se afirmar que, se o espaço para armazenamento dos dados é um problema a ser resolvido, um alto nível de granularidade representa uma forma mais eficiente de representar os dados do que uma representação de nível baixo, pois serão necessários menos *bytes* e índices. Porém, volume de dados e espaço bruto não são as únicas questões relevantes, deve-se considerar também a força de processamento necessária para acessar aos dados, assim como o detalhamento das consultas a serem realizadas, pois “à medida que o nível de granularidade se eleva, há uma correspondente diminuição da possibilidade de utilização dos dados para atender as consultas” (Inmon 1997, p. 48).

Enfim, pode-se resumir que, com um nível muito baixo de granularidade é possível responder a qualquer consulta utilizando, talvez, de uma grande quantidade de recursos para varrer uma grande quantidade de registros. Já com um alto nível o número de questões a que os dados podem satisfazer é limitado. Partindo-se dessa constatação e como sempre há uma grande demanda por eficiência no armazenamento de dados e no acesso a eles, bem como a exigência da possibilidade de análise detalhada dos dados, recomenda-se a utilização de dois ou mais níveis de granularidade na parte detalhada do DW, o que segundo Inmon (1997, p. 50) é chamado de nível “dual” de granularidade.

Para a criação de modelos dimensionais deve-se partir da determinação da granularidade, em seguida detalhar a tabela de fatos e como passo seguinte detalhar as tabelas de dimensões. Para concluir a modelagem dimensional deve-se, então, reunir essas tabelas através da associação da tabela de fatos a um conjunto de tabelas de dimensões resultando em um esquema simples e simétrico, geralmente utilizado em *data mart*, *data mining* ou DW.

2.4 DATA MART, DATA MINING E DATA WAREHOUSE

“Em sua forma mais simples, um *data mart* representa dados de um único processo de negócio” (KIMBALL; ROSS, 2002, p. 12). Podem ser definidos ainda como um pequeno DW com a finalidade de fornecer suporte à tomada de decisões de um pequeno grupo de uma organização.

Algumas organizações optam pela utilização de *data mart* por apresentarem um custo mais baixo e requererem um tempo menor de implementação servindo, muitas vezes, de ferramenta de testes para empresas que desejam explorar os benefícios do DW. Segundo Serra (2002, p. 136) há um consenso no mercado no que diz respeito à idéia de começar pequeno para depois crescer aos poucos, visto que as diferenças entre *data mart* e DW são apenas com relação ao tamanho e escopo a ser tratado, ou seja, enquanto um *data mart* trata de um problema departamental ou local, um DW foca toda a empresa para que o suporte de decisões atue em todos os níveis da organização.

Quanto a *data mining* ou mineração de dados, Serra (2002, p. 206) define como sendo a exploração e análise de grande quantidade de dados com o objetivo de possibilitar, a uma organização, a improvisação de suas operações de *marketing*, vendas e suporte ao cliente.

Segundo Serra (2002, p. 207) é um processo o qual descende, fundamentalmente, de práticas de probabilidade e estatística combinadas com princípios de inteligência artificial com o objetivo de pesquisar, em bases de dados de grande porte, informações que descrevam o comportamento do consumidor. Ainda pode ser classificado como um processo de extração de informações válidas, previamente desconhecidas e de máxima abrangência a partir de grandes bases de dados, utilizando-as para tomada de decisões importantes para a organização. O ponto de partida para a mineração é a organização da base de dados, geralmente obtida pela montagem de DW, o qual tem a finalidade de armazenar todas as informações que se encontram distribuídas em diversos sistemas, em um grande banco de dados.

Finalmente, DW trata-se de um banco de dados composto de informações de outros bancos operacionais com o intuito de dar suporte à decisão aos usuários que observam o caminho que a empresa toma. Segundo Serra (2002, p. 140) é um conjunto de diversas

tecnologias, como ferramentas de extração e conversão de dados voltados para consultas complexas.

Deve ser orientado por assuntos, ou seja, toda modelagem de dados deve ser voltada para os principais assuntos (áreas estratégicas) da empresa, seguindo em direção contrária aos sistemas transacionais os quais focam processos e aplicações específicas.

A integração do DW é a característica mais importante, pois através dela será padronizada a representação única para os dados de diversos sistemas transacionais, os quais geralmente encontram-se armazenados em vários padrões de codificação devido ao fato de terem sido implementados por diferentes programadores e analistas. Um exemplo a ser citado, que ocorre com certa frequência, é o caso das utilizações de diferentes unidades de medida para mensurar uma mesma característica, fato que jamais poderá ocorrer em um DW. No caso específico do desenvolvimento desse trabalho, esse aspecto não se apresenta como crítico pelo fato de que os dados foram extraídos de um único sistema transacional.

Outro aspecto a ser considerado é o fato do DW ser variável em relação ao tempo, ou seja, são armazenados os dados por um período de tempo maior do que nos sistemas transacionais. Enquanto em um ambiente operacional mantemos os dados com o intuito de fornecer as informações em um momento exato, em um DW o objetivo é analisar as mesmas e seus comportamentos durante um determinado período de tempo. “A dimensão tempo sempre estará presente em qualquer fato de um DW” (SERRA, 2002 p. 145).

Uma característica importante a ser citada é a não volatilidade dos dados em um DW, pois são realizadas apenas duas operações: a carga inicial (leitura dos dados de origem e gravação no banco modelado multi-dimensionalmente) e as consultas dos dados. O que difere de um sistema transacional onde são realizadas, constantemente, atualizações dos mesmos.

Quanto à localização dos dados, esses podem ser armazenados de três maneiras: em um único local centralizando o banco de dados em um DW integrado, objetivando assim, maximizar o poder de processamento e agilizando a busca dos dados; de forma distribuída armazenando os dados em *data marts* por área de interesse; e por níveis de detalhes onde os dados são armazenados em servidores diferentes de acordo com seu nível de detalhamento.

Todas essas formas de apresentação e análise de dados servem como suporte para a efetiva implantação de um CRM.

2.5 CRM – *CUSTOMER RELATIONSHIP MANAGEMENT*

Apesar das inúmeras definições e os diversos conceitos definidos para CRM, essa metodologia orienta-se em um princípio básico: conhecer bem as necessidades de cada cliente para que possa oferecer-lhes o que cada um realmente deseja.

“O CRM baseia-se na simples noção de que quanto melhor você conhecer seus clientes, mais facilmente conseguirá manter relacionamentos duradouros e importantes com eles” (KIMBALL; ROSS, 2002, p. 162).

Já Serra (2002, p. 110) define CRM como sendo a integração de conceitos, métricas, processos, soluções, gestão de canais, estratégias e ferramentas das áreas de *marketing*, vendas e serviços.

De acordo com Greenberg (2001, p. 65), CRM é uma estratégia disciplinada de negócios, a qual utiliza a tecnologia de informação como um veículo para atingir seus objetivos.

O CRM pode ser dividido em três tipos de estratégias: operacional, analítico e colaborativo. O CRM operacional, segundo Greenberg (2001, p. 66-67), trata das funções empresariais típicas, as quais envolve os serviços ao consumidor, gerenciamento de pedidos, faturamento ou automação de *marketing* e vendas.

Segundo Kimball e Ross (2002, p. 164) o CRM operacional procura integrar todas as atividades relacionadas aos clientes, caracterizando cada contato com o mesmo como sendo uma oportunidade de coletar mais medidas e características, aumentando assim, os dados existentes para usufruir melhor do relacionamento. Considera a relação com o cliente desde o contato inicial de prospecção, passando pela informação dos preços, a transação da compra, a concretização, a realização do pagamento e o acompanhamento até o serviço de atendimento posterior ao cliente.

O CRM analítico é responsável pela captação, armazenagem, acesso, processamento, interpretação e transmissão de dados dos clientes para o usuário do programa. Segundo Kimball e Ross (2002, p. 165), este tipo de CRM tem como finalidade avaliar a eficiência das decisões tomadas anteriormente para otimizar interações futuras. Os dados de clientes são utilizados para identificar melhores oportunidades, apontar ineficiências, gerar demanda e

melhorar a retenção. Serra (2002, p. 111), por sua vez, define CRM analítico como a área que permite identificar e acompanhar diferentes tipos de clientes (os de maior valor, de valor potencial ou de valor zero) dentro da carteira de clientes de uma empresa. Diante dessas informações seriam determinadas as estratégias para atender às diferentes necessidades dos clientes identificados. O protótipo desenvolvido nesse trabalho enquadra-se nessa classificação de CRM.

Por fim, o CRM colaborativo apresenta-se como um centro de comunicações ou uma rede de coordenação que oferece os principais caminhos entre os clientes e seus fornecedores. Pode significar como sendo um ponto de interação entre o consumidor e o canal de comunicação. Segundo Serra (2002, p. 111) é a aplicação da tecnologia da informação, permitindo a automação e a integração entre os pontos de contato do cliente com a empresa. Esses pontos devem estar preparados para interagir com o cliente e distribuir as informações capturadas para os sistemas do CRM operacional. Enquadram-se nessas estratégias canais de comunicação como a *Internet*, canal de voz ou correio convencional.

Convém salientar que não foram encontrados trabalhos nessa instituição com foco em CRM. Foram apresentados anteriormente trabalhos como o de Dal Pozzo (2002), o qual desenvolveu um trabalho com o objetivo de implementação de um sistema de informação, baseado em DW, para auxiliar profissionais ligados à área de venda, carteira e faturamento, o qual visou a especificação de um modelo multidimensional, possibilitando a implementação de conceitos de CRM ao segmento.

Constatou-se, também, que Dalfovo (2001) aplicou um modelo de Sistema de Informação Estratégico para o Gerenciamento Operacional (SIEGO) baseado em DW disponibilizando consultas e acessos a Universidade e Sociedade, através da rede interna ou acesso de dados via *Internet* sobre a área de gestão ambiental. Este trabalho também relacionou conceitos relativos a CRM em sua transcrição para o desenvolvimento, todavia, não descreve sua real utilização diante do cenário criado.

Além desses citados anteriormente, convém referenciar o trabalho de Schatzmann (2003) o qual visou a construção de um Sistema de Informação onde, aplicando-se a filosofia de *Data Warehouse* e definições de granularidade, disponibilizou informações da área comercial auxiliando executivos de qualquer empresa a tomarem suas decisões estratégicas.

3 DESENVOLVIMENTO DO TRABALHO

Neste capítulo é apresentada a metodologia utilizada para o desenvolvimento deste trabalho. Como ponto inicial, são apresentados os requisitos do protótipo de software desenvolvido.

3.1 REQUISITOS PRINCIPAIS

O protótipo de software apresenta como requisitos funcionais:

- a) possibilidade de efetuar consultas dinâmicas;
- b) possibilidade de emissão de relatórios com informações gerenciais;
- c) possibilidade de emissão de gráficos referentes às informações gerenciais;
- d) possibilidade de salvar as consultas dinâmicas realizadas;
- e) possibilidade de importação de dados da base operacional pré-definida que contribuam para a análise gerencial dos dados.

Como requisitos não funcionais podem ser citados:

- a) operação em ambiente Cliente / Servidor;
- b) utilização em ambiente Windows;
- c) operação sobre o SGBD Microsoft SQL Server;

3.2 ESPECIFICAÇÃO

Para a especificação deste trabalho optou-se pela técnica de análise estruturada, a qual foi desenvolvida através da utilização do software *Power Designer* versão 9.5. Foram utilizados: diagrama de contexto, diagramas de fluxo de dados e modelo entidade-relacionamento (MER).

A seguir é demonstrado o diagrama de contexto (figura 1), com o objetivo de representar as entidades externas que se relacionam com o protótipo desenvolvido.

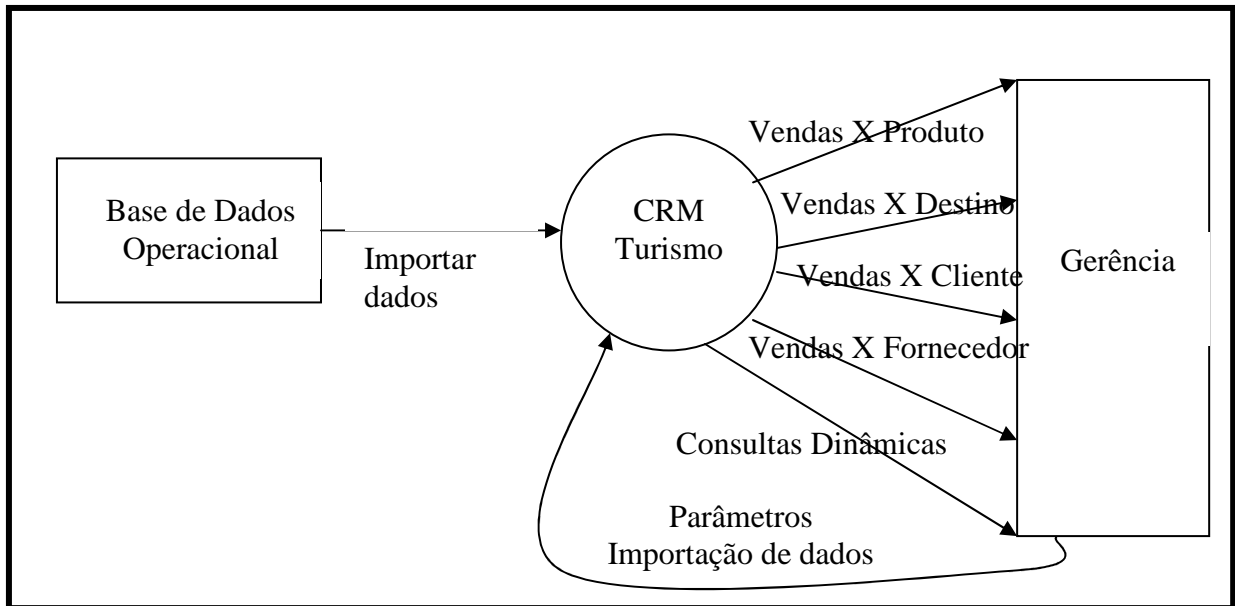


Figura 1 – Diagrama de Contexto

Na seqüência são apresentados os diagramas de fluxo de dados - DFD (figuras 2 e 3), os quais têm o intuito de demonstrar as diferentes funcionalidades desenvolvidas no protótipo de software, assim como onde são armazenadas as informações tratadas pelo mesmo.

Na figura 2 são demonstrados os DFDs referentes às rotinas de: importação de dados, a qual realiza a extração dos dados da base operacional para o modelo dimensional e geração de relatório de vendas por produto.

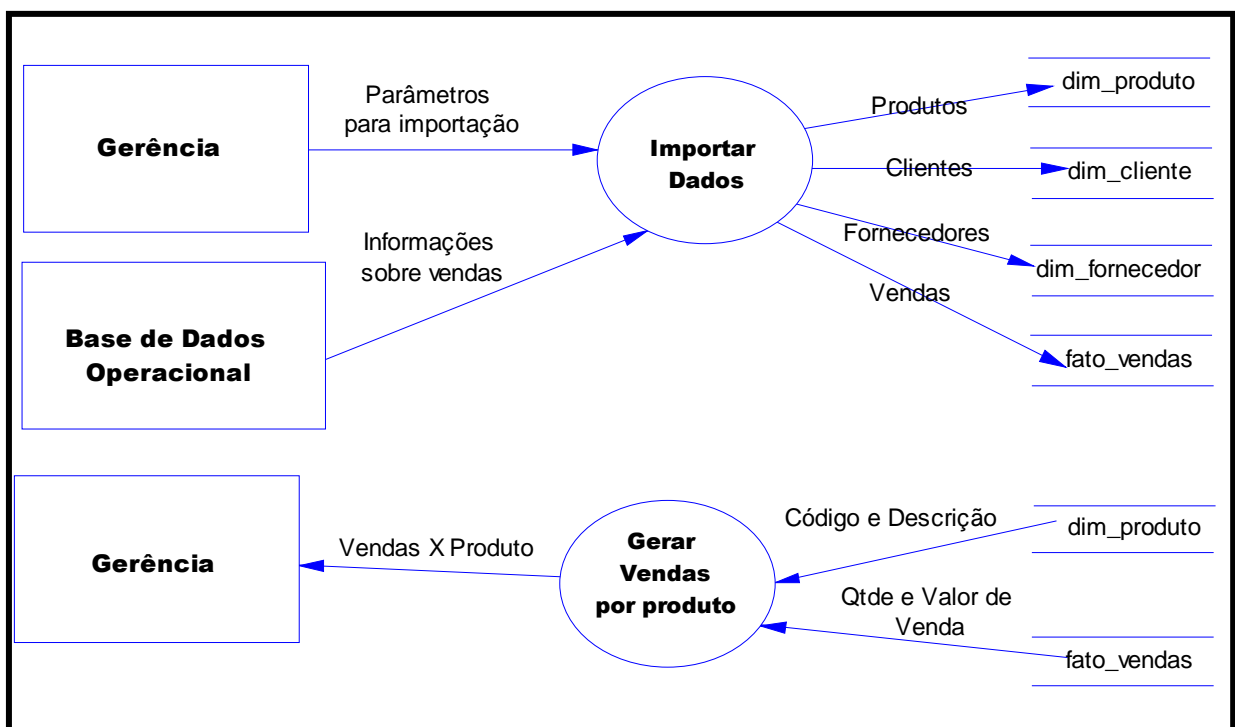


Figura 2 – Diagramas de Fluxo de Dados

Na figura 3 são apresentados os DFDs das rotinas de: geração dos relatórios de vendas por destino, vendas por cliente, vendas por fornecedor e geração de consultas dinâmicas.

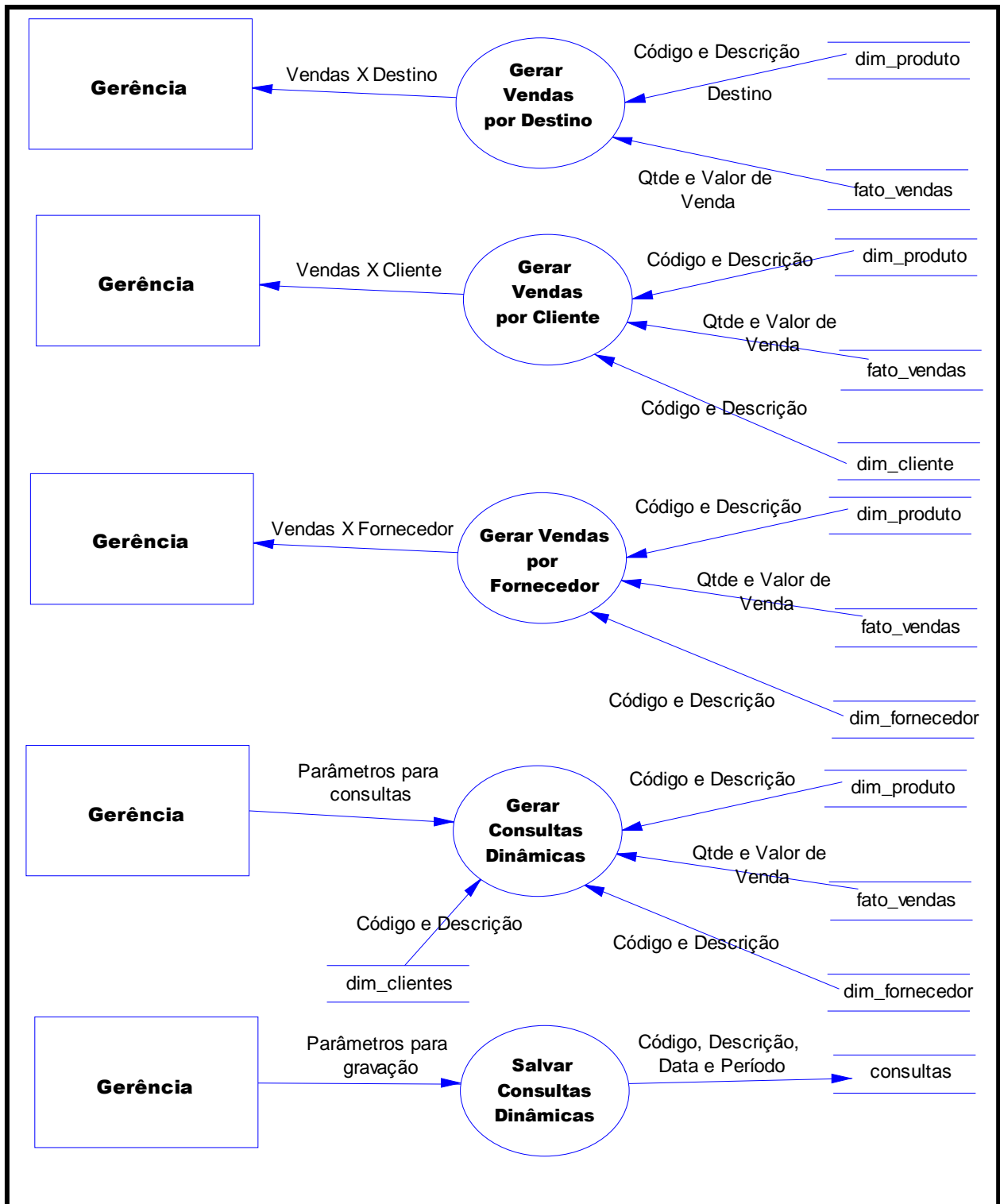


Figura 3 – Diagramas de Fluxo de Dados

Outra representação importante a ser detalhada é o modelo de entidade-relacionamento (MER). A figura 4 apresenta o MER referente ao modelo multidimensional da base de dados, a qual é gerada através da execução da rotina de extração dos dados. Convém ressaltar que nos apêndices A e B são representadas as nomenclaturas das tabelas da base operacional para facilitar o entendimento.

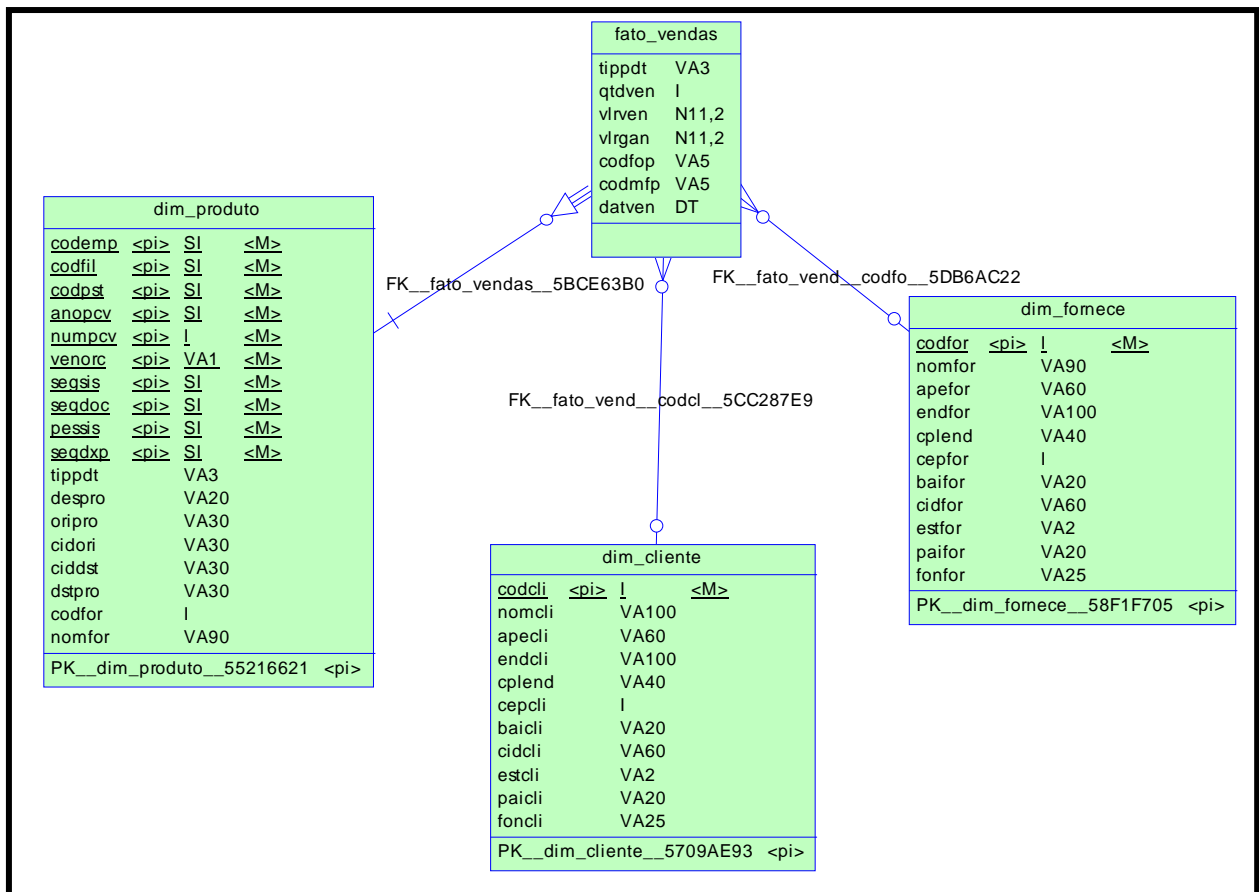


Figura 4 – MER lógico do modelo multidimensional

3.3 IMPLEMENTAÇÃO

Nesta seção do trabalho são demonstradas as ferramentas e técnicas utilizadas para implementação do protótipo de software proposto, assim como sua operacionalidade. No primeiro subitem tem-se como objetivo citar de que maneira foi realizada a implementação detalhando tecnicamente os recursos empregados e no segundo momento é representado como o usuário deverá interagir com o protótipo para obter os resultados desejados.

3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

O ponto inicial para o desenvolvimento do protótipo foi obter a base de dados do sistema transacional. Tal procedimento foi realizado utilizando o aplicativo *Enterprise Manager*, o qual faz parte do pacote de instalação do SGBD Microsoft SQL Server, conforme representa a figura 5. A exportação dos dados do servidor que continha a base operacional, assim como a recuperação da base na máquina onde foi desenvolvido o protótipo foram realizadas utilizando esse aplicativo. Tal procedimento fez-se necessário apenas pelo fato do protótipo ser desenvolvido em um computador o qual não tinha acesso ao servidor com a base operacional, caso seja executado o mesmo em uma estação com acesso à base de dados operacional tal procedimento é dispensável.

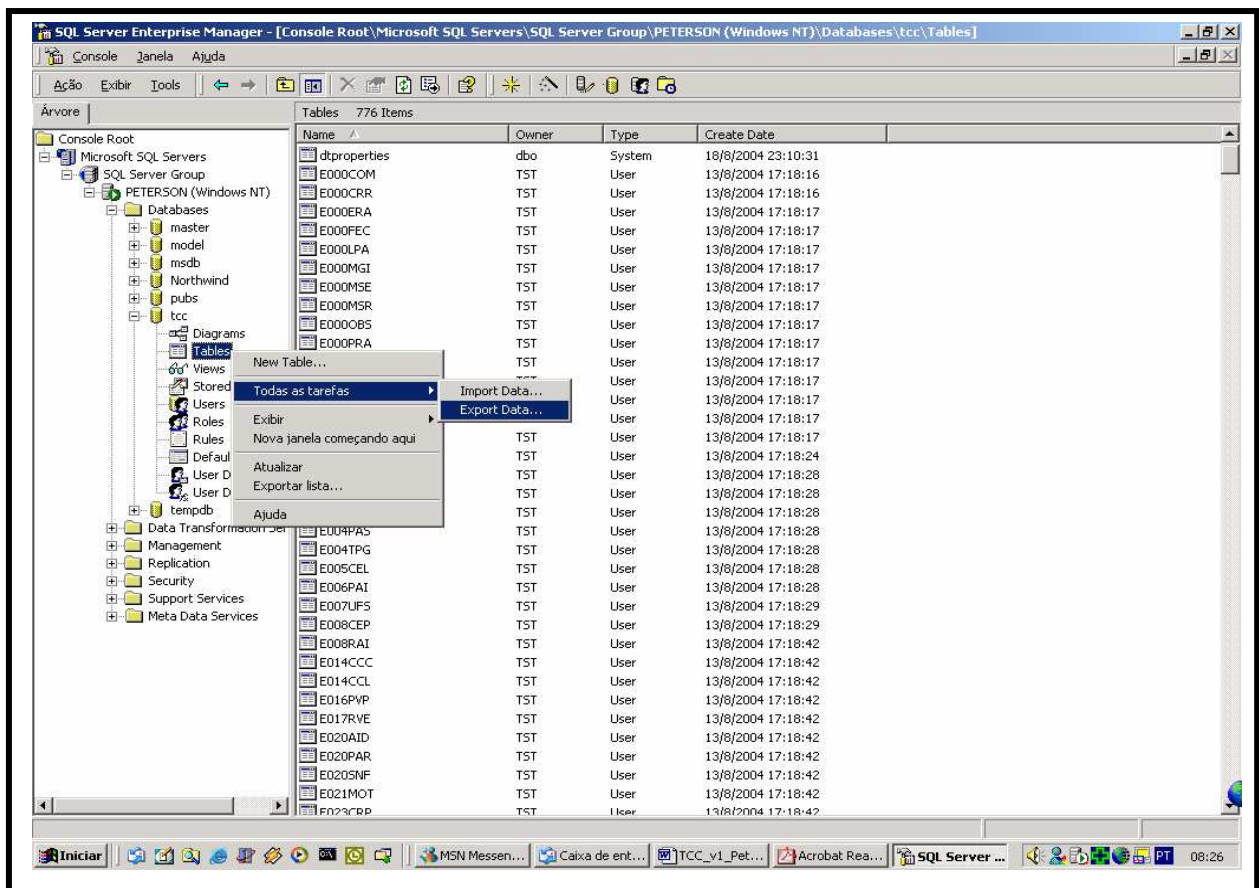


Figura 5 – Interface do aplicativo *Enterprise Manager*

Convém salientar que, por tratar-se de uma base de dados de uma agência de turismo, cliente do sistema Regente da empresa Senior Sistemas, foi necessária a execução, na base de dados, de um *script* criado internamente na empresa, o qual tem a finalidade de omitir dados que poderiam disponibilizar informações sigilosas pertinentes ao negócio.

A partir do momento em que a base de dados foi restaurada, foi realizada análise dos aspectos de normalização da mesma e constatado que, devido à complexidade do sistema transacional, no qual são utilizadas aproximadamente setecentas e quinze tabelas, não são respeitadas as regras de normalização de dados no que diz respeito à redundância de dados. Tal característica justifica-se para evitar que durante operações dentro do sistema transacional, seja necessário acessar diversas tabelas para buscar determinadas informações. Conforme citado na fundamentação teórica deste trabalho, para a modelagem dimensional não necessariamente deve-se normalizar a base de dados, pois essa operação pode afetar o desempenho do DW no momento das consultas ao modelo dimensional.

Seguindo orientação de Kimball e Ross (2002, p. 36-38), pôde-se criar um banco de dados dimensional considerando sistematicamente quatro etapas em uma ordem específica:

- a) selecionar o processo de negócio a ser modelado. Neste trabalho foi definido que o processo a ser modelado seria o processo comercial de uma agência de turismo com o objetivo de analisar as medições de desempenho da área de vendas;
- b) declarar a granularidade do processo de negócio. Foi elaborado o *script* para a extração dos dados da base operacional para a criação do modelo multidimensional considerando a granularidade definida junto aos consultores de negócio do sistema transacional, os quais ressaltaram que se deveria ter um nível de detalhamento muito alto, ou seja, um baixo nível de granularidade possibilitando consultas por vendas. Esse *script* pode ser visualizado no apêndice C, sendo que o mesmo faz parte de uma rotina interna do processo de carga de dados do DW, sendo apresentado no apêndice apenas para facilitar sua visualização;
- c) escolher as dimensões que se aplicam a cada linha da tabela de fatos. Foi definido que seriam tratadas três dimensões: produtos, clientes e fornecedores sendo que a dimensão produto ainda permitiria consultas por destino do segmento de turismo;
- d) identificar os fatos numéricos que preencherão cada linha da tabela de fatos. Neste trabalho foi estipulado que seriam levantadas informações de quantidade vendida, valor de venda e valor de ganho para cada venda realizada.

A partir dessas definições iniciou-se a implementação do protótipo utilizando a linguagem de programação *Object Pascal* através da ferramenta de desenvolvimento *Delphi*, sendo que toda interação entre essa ferramenta e o SGBD Microsoft SQL Server foi realizada através de comandos SQL definidos em componentes do tipo *tQuery* conforme pode ser

visualizado no quadro 1. Para comunicação com o SGBD, foi utilizado um *drive* nativo para *Microsoft SQL Server* disponibilizado pela Borland Database Engine (BDE).

```

try
  qImporta := tQuery.Create(Self);
  try
    qImporta.DatabaseName := 'TCC';
    qImporta.SQL.Clear;
    qImporta.SQL.Add('create table dim_fornece (codfor integer,
nomfor varchar(100), apefor varchar(60), endfor varchar(100), cplend
varchar(40), '+
                                'cepfor integer, baifor varchar(20), cidfor
varchar(60), estfor varchar(2), paifor varchar(20), '+
                                'fonfor varchar(25), '+
                                'primary key (codfor)');
    qImporta.Prepare;
    qImporta.ExecSQL;
  finally
    qImporta.Free;
    qImporta := Nil;
  end;
except
  ShowMessage('Erro ao tentar criar tabela de fornecedores !');
end;

```

Quadro 1 – Exemplo de código utilizando *tQuery*

Um ponto importante a ser ressaltado dentro da implementação do protótipo é que todas funcionalidades do mesmo dependem da rotina de importação de dados, a qual, devido a essa característica, deve ser executada, no mínimo uma vez, ao iniciar a utilização do protótipo e anterior a qualquer outra operação. Depois de executada essa funcionalidade, é gerado o modelo dimensional, criando a tabela de fatos e as tabelas de dimensão, as quais têm seus dicionários de dados apresentado no apêndice D. Essas tabelas ficam gravadas no banco de dados mesmo tendo sido encerrada a execução do protótipo, sendo excluídas e recriadas a cada nova execução da rotina de importação de dados.

A principal funcionalidade do protótipo (consultas dinâmicas) foi desenvolvida utilizando um grupo de componentes da *pallet* “*Decision Cube*” do *Delphi* conforme pode ser visualizado na figura 6, os quais realizam sumarizações e tratamentos de dados que, caso fossem implementados sem a utilização dos mesmos, exigiria uma grande quantidade de código e de horas de trabalho para sua conclusão.

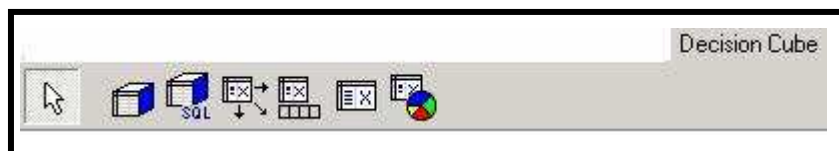


Figura 6 – *Pallet de componentes Decision Cube.*

Observando-se a figura e acompanhando da esquerda para a direita têm-se os componentes e quais suas funções detalhadas a seguir:

- a) *DecisionCube*: como citado anteriormente é um componente que deve ser utilizado quando deseja-se realizar o tratamento de dados sumarizados;
- b) *DecisionQuery*: esse componente é um descendente de *TQuery* com características específicas para preparar dados da *query* no formato suportado pelo componente *DecisionCube*. Deve-se editar um comando SQL nesse componente com o intuito de buscar as informações desejadas na tabela de fatos, informar quais serão as colunas de dimensões e quais valores serão sumarizados, fazendo com que ele crie automaticamente o comando SQL, conforme demonstrado no quadro 2, o qual interferirá diretamente na apresentação dos componentes visuais *DecisionGrid* e *DecisionGraph*;
- c) *DecisionSource*: utiliza dos dados preparados pelo componente *DecisionQuery* para disponibilizá-los aos componentes *DecisionGrid* e *DecisionGraph*;
- d) *Decision Pivot*: componente responsável em permitir ao usuário controlar, através de botões, as dimensões ativas de um componente *DecisionSource* e que serão apresentados em componentes visuais *DecisionGrid* ou *DecisionGraph*;
- e) *DecisionGrid*: componente visual similar ao *DBGrid* que apresenta a funcionalidade de apresentar ou ocultar colunas conforme opção do usuário;
- f) *DecisionGraph*: componente visual que tem como principal finalidade apresentar gráficos dinâmicos ao usuário.

```
SELECT despro, nomcli, nomfor, datven, SUM( qtdven ),
SUM( vlrven ), SUM( vlrgan )
FROM fato_vendas
GROUP BY despro, nomcli, nomfor, datven
```

Quadro 2 – Código gerado automaticamente pelo componente *DecisionQuery*.

Para implementação dos relatórios foram utilizados os componentes da *pallet QReport*, a qual permite que sejam desenvolvidos relatórios baseados em *queries* e que sejam realizadas quebras de acordo com a configuração das seções dos relatórios. Convém ressaltar que foram implementadas as opções de exportação desses relatórios para formato texto e HTML.

3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Nesta seção são apresentadas as operacionalidades do protótipo de software desenvolvido, servindo como um manual de usuário, com o intuito de auxiliar os usuários na utilização do mesmo.

O protótipo, ao ser executado, apresenta um menu como tela inicial (figura 7), disponibilizando ao usuário opções que serão detalhadas abaixo:

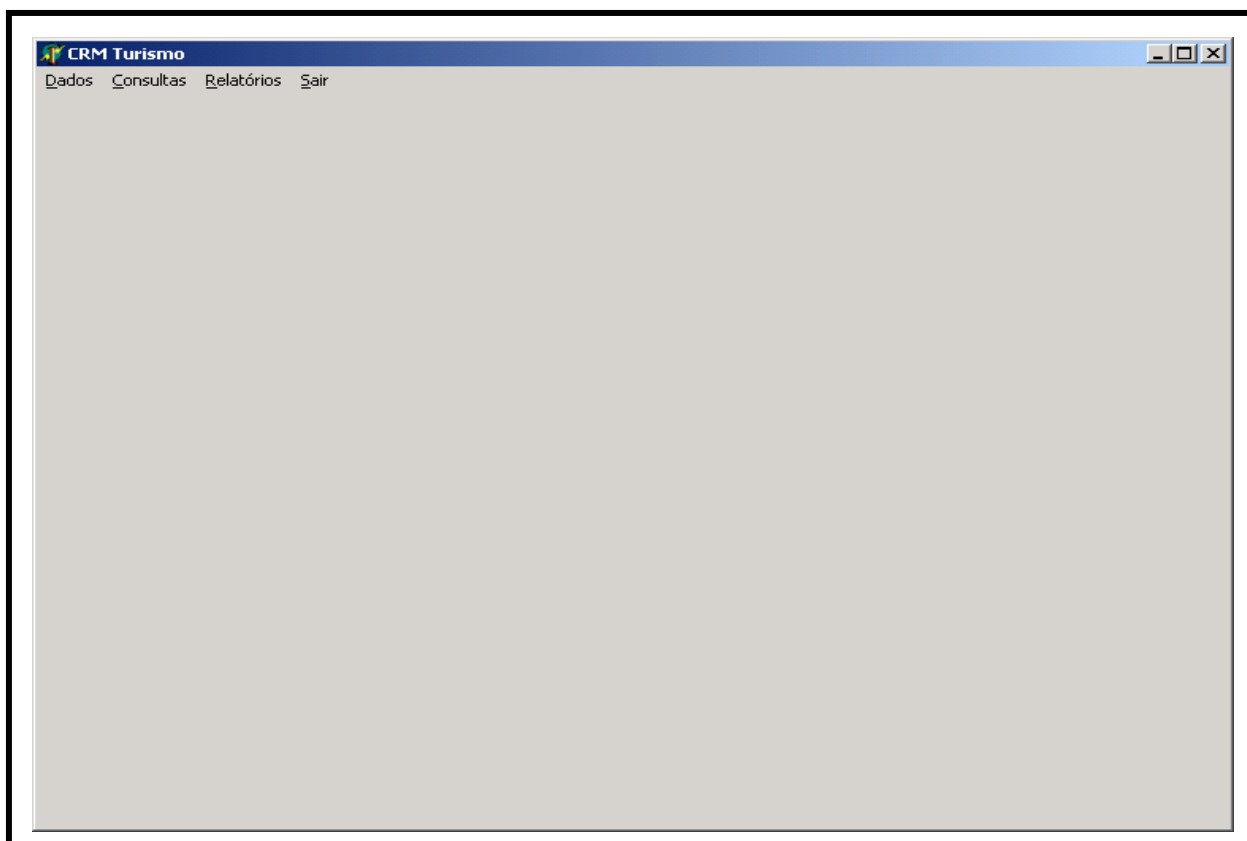


Figura 7 – Menu principal do protótipo

Ao acessar a opção de menu “Dados”, é apresentada ao usuário apenas uma opção denominada “Importação”. Essa rotina é responsável em questionar qual período será utilizado para filtrar os registros para criação do modelo dimensional, através da interface

apresentada na figura 8. Convém salientar que esse filtro afetará apenas a tabela de fatos, visto que as tabelas de dimensões possuem os dados descritivos, os quais são cadastrados na base operacional não necessariamente no mesmo período em que os dados referentes às vendas são lançados.



Figura 8 – Interface de definição de período para importação dos dados.

Ao clicar em executar, o protótipo de software apaga as tabelas de fato e dimensões existentes para o usuário no banco de dados e gera as mesmas novamente, populando-as com os dados referentes ao período informado. No término da execução dessa rotina é apresentada uma mensagem ao usuário informando que está concluído o processo através de uma caixa de diálogo com apenas um botão “OK”, o qual sendo pressionado finaliza a interface de seleção de período, retornando ao menu principal do protótipo.

Como item seguinte, no menu principal do protótipo, é apresentada a possibilidade de realizar consultas, a qual disponibiliza os subitens: “Dinâmicas” e “Recuperar”. Quando acessada a opção “Dinâmicas” é exibida a interface conforme figura 9, a qual permite ao usuário consultar as sumarizações dos dados contidos no modelo dimensional, sendo que na parte superior da tela estão disponíveis botões para selecionar quais colunas (referentes às tabelas de dimensões) serão visíveis na *grid*, com exceção do primeiro da esquerda para a direita, o qual possibilita ao usuário selecionar quais dados serão demonstrados no interior da *grid*, tendo como opções (apresentadas em um menu *popup* quando clicado sobre o mesmo): “Valor de Venda”, “Valor Ganho” e “Qtde Vendida”.

Cliente	Data de venda	Aéreo	Carro	Cruzeiro	Hotel
Css BC_NomY_CoYc	2000	R\$ 272316,00	R\$ 1227,00	R\$ 5300,00	R\$ 51808,93
	Sum	R\$ 272316,00	R\$ 1227,00	R\$ 5300,00	R\$ 51808,93
FontY's YngYnhp Yt	2000	R\$ 18649,05	R\$ 1700,00		R\$ 845,00
	Sum	R\$ 18649,05	R\$ 1700,00		R\$ 845,00
Ponto ComYrcpY Ytd	2000	R\$ 257,00			
	Sum	R\$ 257,00			
PpYOT SmPYR GRp	2000	R\$ 1507,20			
	Sum	R\$ 1507,20			
PrYFptmr Mmnpopp	2000	R\$ 100,00			
	Sum	R\$ 100,00			
PYtros Qmpmpc Ytd	2000				R\$ 340,00
	Sum				R\$ 340,00
SoYmr TYxtpY S..	2000	R\$ 53002,75			R\$ 4705,74
	Sum	R\$ 53002,75			R\$ 4705,74

Figura 9 – Interface para realização de consultas dinâmicas.

Na região inferior, ainda da mesma interface, são disponibilizados mais três botões:

- “Gráfico”: o qual tem a função de apresentar uma outra interface contendo os gráficos referentes às consultas dinâmicas;
- “Salvar”: tem como finalidade exibir interface para que o usuário possa salvar as consultas dinâmicas realizadas de acordo com o que é demonstrado na figura 11;
- “Sair”: sai da tela de consultas para retornar ao menu inicial do protótipo.

Nessa interface, pode-se analisar que o primeiro cliente listado consumiu uma variedade maior de tipos de produto do que os demais clientes e que suas maiores compras foram para os produtos: aéreo e hotel. A partir desses dados, especialistas de negócio podem consultar, nos relatórios disponibilizados pelo protótipo, quais destinos o mesmo costuma viajar e, portanto oferecerem produtos que mais se enquadram ao seu perfil.

Da mesma maneira pode-se averiguar que o segundo cliente consumiu também produtos do tipo aéreo e hotel, porém em menor quantidade. Através dessa constatação e

recuperando informações mais detalhadas obtidas através dos relatórios, pode-se avaliar quais melhores ofertas a propor para esse cliente para que seu consumo aumente gerando melhores rendimentos à agência de turismo.

Conforme citado anteriormente, o protótipo disponibiliza interface para visualização de gráficos referentes às consultas dinâmicas conforme representado na figura 10.

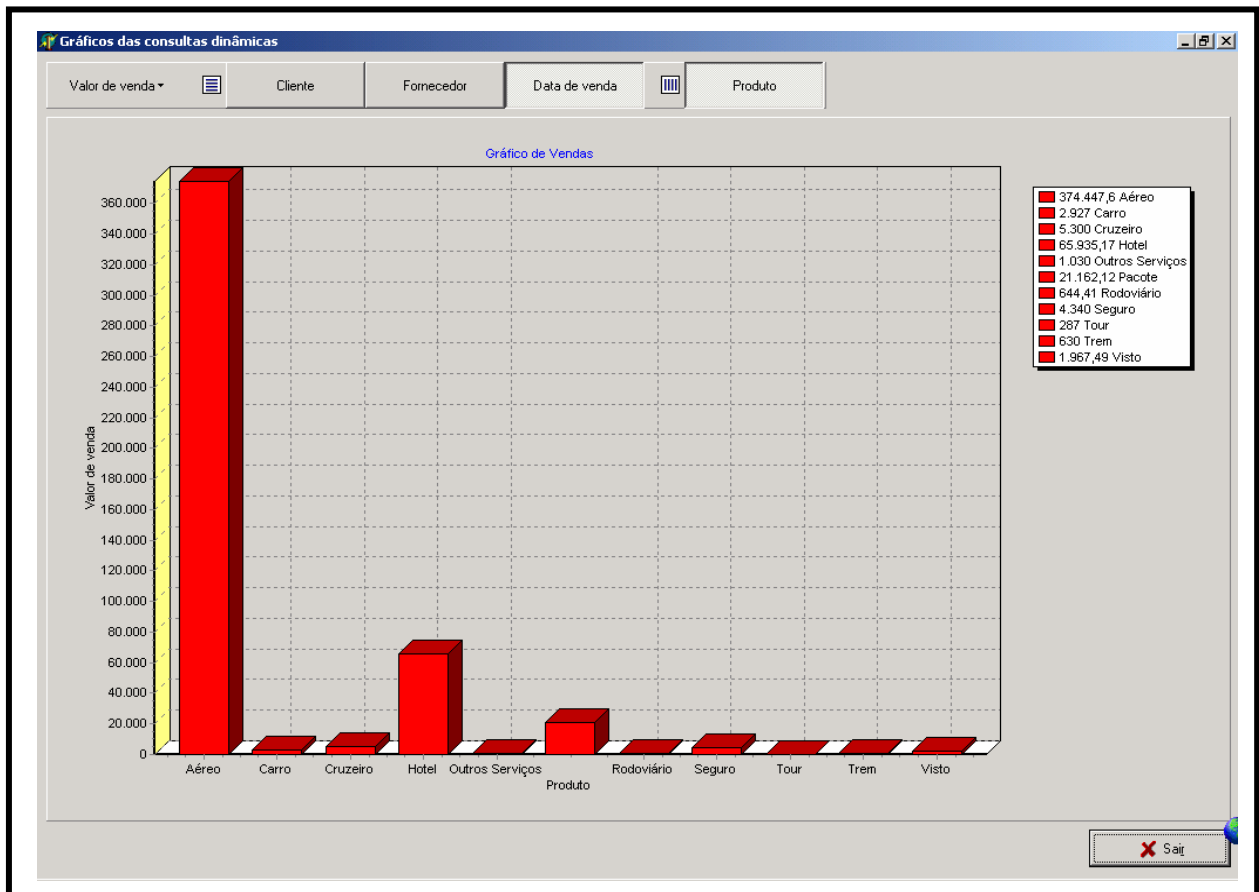


Figura 10 – Interface para visualização dos gráficos de consultas dinâmicas.

Nessa interface, apresentada na figura acima, é possível visualizar os gráficos referentes às consultas dinâmicas realizadas, disponibilizando ao usuário a interação com as colunas através da seleção dos botões posicionados na parte superior da tela. Convém salientar que ainda é disponibilizada a opção de seleção de quais serão os valores que comporão o eixo esquerdo do gráfico através do primeiro botão no canto superior esquerdo.

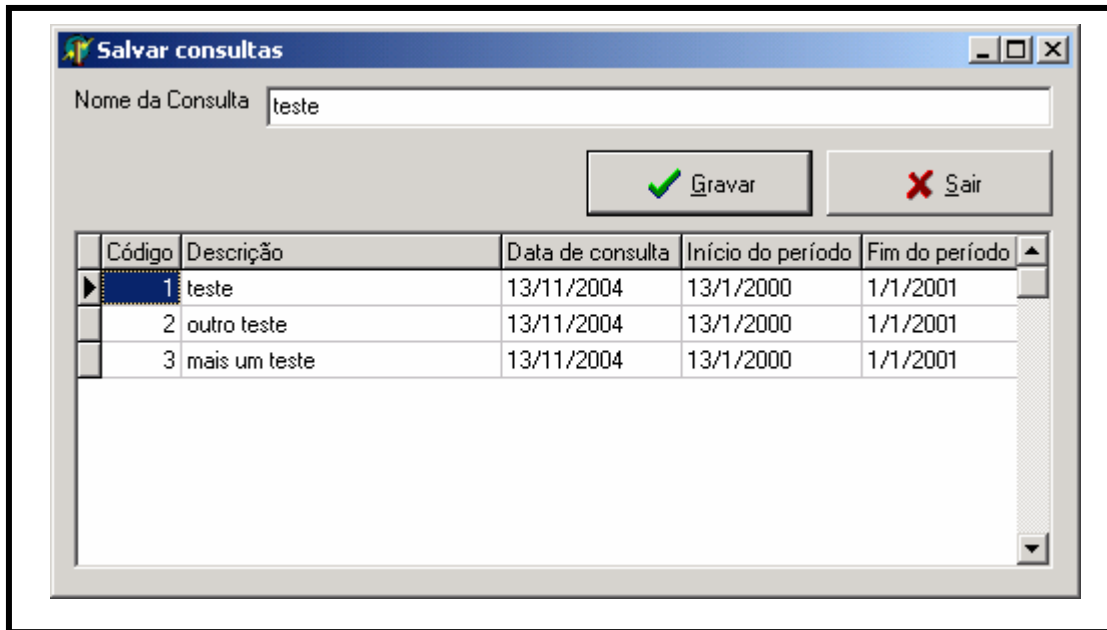


Figura 11 – Interface para salvar consultas dinâmicas.

Através da interface apresentada acima, o usuário pode informar um nome qualquer para atribuir o mesmo à consulta dinâmica realizada, possibilitando assim, que seja recuperada a mesma através da opção “Consultas” “Recuperar” no menu principal do protótipo, o qual apresenta uma tela para interação do usuário conforme representado na figura 12.

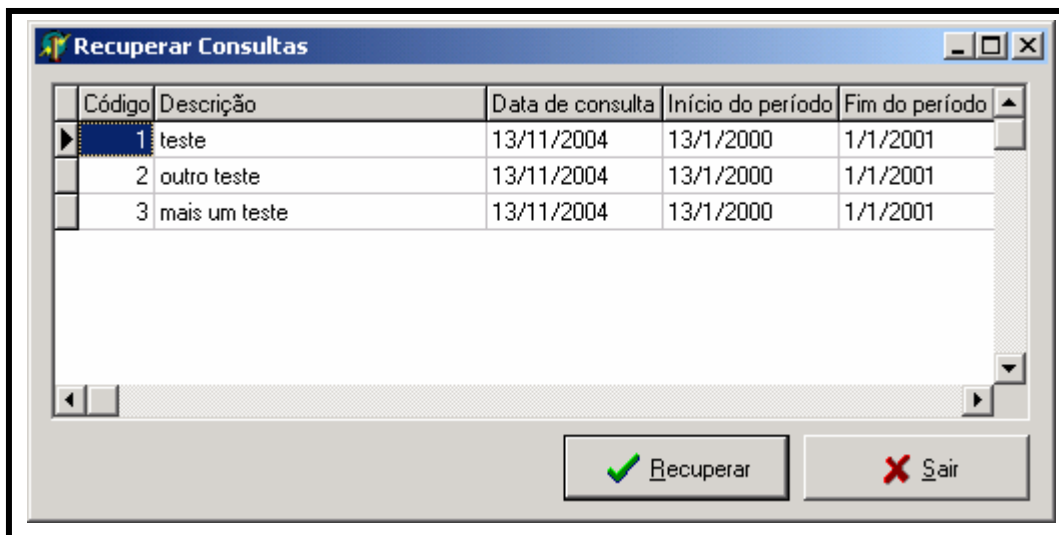


Figura 12 – Interface para recuperação de consultas dinâmicas.

Ao selecionar um dos registros da *grid* e clicar no botão “Recuperar” dessa tela, o usuário define que todas consultas e gráficos visualizados estejam operando sobre os dados

gerados através da importação de dados realizada naquele período. Ao clicar no botão sair, retorna ao menu principal do protótipo.

A próxima opção do menu principal do protótipo de software disponibiliza emissões de relatórios de vendas de determinado período informado pelo usuário com as seguintes opções:

- “Vendas por produto”: apresenta relatório com quebras por tipo de produto;
- “Vendas por cliente”: apresenta relatório com quebras por cliente;
- “Vendas por fornecedor”: apresenta relatório com quebras por fornecedor;
- “Vendas por destino”: apresenta relatório com quebras por destino;
- “Vendas por data”: apresenta relatório com quebras por data.

Os *lay-outs* de todos relatórios assemelham-se com o que é apresentado na figura 13, diferenciando apenas no que diz respeito ao cabeçalho para as quebras de acordo com cada título de relatório.

13/11/2004 09:02:27

Relatório de Vendas por Clientes 1

Cliente 2 - Css BC_NomY_CoYocda_Pr_o_Máxpmo PossivYY_Pr_mm_CYpYntY Pr_QTYst_Y_Omtrst_Y's

Data da Venda	Produto	Qtde Vendida	Valor de Venda	Valor Ganho
23/1/2000	Aéreo	2	1008,01	937,2
23/1/2000	Aéreo	3	850,32	808
4/3/2000	Aéreo	1	510	428,89
8/4/2000	Aéreo	1	1000	887,78
8/4/2000	Aéreo	3	107,89	91,2
9/4/2000	Aéreo	1	600	568,89
9/4/2000	Aéreo	2	0	-15,87
9/4/2000	Aéreo	2	110,05	93
15/4/2000	Aéreo	1	83,3	-16,11
15/4/2000	Aéreo	1	133,28	-25,78
16/4/2000	Hotel	1	115	115
16/4/2000	Hotel	1	132	132
16/4/2000	Hotel	1	142	142
16/4/2000	Hotel	1	220	200
26/4/2000	Aéreo	1	590	568,89
27/4/2000	Aéreo	1	200	151,56
27/4/2000	Aéreo	1	1000	757,78
27/4/2000	Aéreo	1	1000	837,78

85% Page 1 of 15

Figura 13 – Exemplo de relatório emitido através do protótipo de software.

E, finalmente, como último item do menu principal do protótipo de software, é apresentada a opção “Sair”, a qual tem como finalidade encerrar a execução do protótipo.

3.4 RESULTADOS E DISCUSSÃO

Após o término da confecção deste trabalho, tem-se como resultado uma ferramenta de apoio para que os responsáveis em gerenciar os recursos para investimentos na área de vendas de uma agência de turismo possam tomar suas decisões baseados em valores e resultados demonstrados através das funcionalidades do protótipo.

Caso a gerência queira ter uma visão geral sobre as vendas poderá utilizar o recurso de consultas dinâmicas, podendo interagir com a mesma para obter as informações da maneira que lhe seja mais clara. Se a necessidade do usuário for uma visão detalhada das vendas, desejando visualizar item a item, ele poderá utilizar os relatórios disponibilizados no protótipo de software com as diferentes opções de quebra.

Quanto à performance do protótipo, pode-se avaliar que após o tratamento dos dados na rotina de extração conseguiu-se obter resultado positivo, pois houve uma redução significativa na massa de dados, passando da ordem de aproximadamente vinte mil registros da base de dados operacional para aproximadamente três mil registros no modelo dimensional.

4 CONCLUSÕES

Atualmente, é cada vez mais comum que as empresas adotem a utilização de ferramentas que subsidiem informações que sejam úteis para um melhor gerenciamento dos investimentos e recursos da mesma. Geralmente esses tipos de soluções apresentam um custo muito elevado e demandam um tempo razoável de projeto e implantação, fatos que contribuem para que surja uma certa resistência à implantação das mesmas. Porém, é visto que quando realizado um projeto bem elaborado, o retorno desses investimentos geralmente ocorre, o que faz com que a empresa busque novas soluções para aprofundar ainda mais seus conhecimentos em relação aos seus clientes.

Com o desenvolvimento desse trabalho pode-se averiguar que as técnicas de *Data Warehouse* são perfeitamente aplicáveis para obter êxito no que diz respeito ao levantamento e tratamento de dados de clientes com o intuito de estabelecer com o mesmo um melhor relacionamento de negócios.

As ferramentas utilizadas para o desenvolvimento deste trabalho e da implementação do protótipo de software atenderam as expectativas, fazendo com que os objetivos propostos fossem alcançados em sua totalidade.

Avalia-se que, como principal vantagem de desenvolvimento e utilização desse protótipo de software é a integração que ele oferece com a base de dados operacional, visto que uma vez configurado o acesso ao banco de dados, o usuário pode fazer as importações nos momentos que julgar necessário apenas selecionando uma opção de menu do mesmo.

Do ponto de vista pessoal, pode-se concluir que o desenvolvimento desse trabalho fez com que o autor obtivesse conhecimento sobre a área de gerenciamento de relacionamento com clientes, a qual desperta muito interesse e que ainda há muito que ser estudado e desenvolvido.

Quanto aos objetivos apresentados, pode-se afirmar que foram alcançados plenamente pois, ao final do desenvolvimento deste trabalho chegou-se a um protótipo que permite aos gerentes de agências de turismo terem acesso à informações que realmente possibilitem tomadas de decisões coerentes com que o mercado consome.

4.1 EXTENSÕES

A partir da detecção da limitação do protótipo no que se refere ao controle de versões dos dados importados, surge então a sugestão de extensão para esse trabalho de desenvolver uma rotina que permita ao usuário controlar quando realizou as importações de dados e que possa trabalhar sobre diferentes modelos dimensionais simultaneamente.

REFERÊNCIAS BIBLIOGRÁFICAS

DAL POZZO, Marco Aurélio. **Implementação do cubo de decisão em um *data warehouse* extraído de um sistema de gerenciamento empresarial**. 2002. 42 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DALFOVO, Oscar. **Metodologia sistema de informação estratégico para o gerenciamento operacional (SIEGO)**. Um modelo SIEGO para a universidade com aplicação na gestão ambiental baseado em *data warehouse*. 2001. 291 f. Tese (Doutorado em Ciências da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

GREENBERG, Paul. **CRM, *customer relationship management* na velocidade da luz: conquista e lealdade de clientes em tempo real na Internet**. Tradução Reinaldo Marcondes. Rio de Janeiro: Campus, 2001. 409 p.

INMON, William H. **Como construir o *data warehouse***. Tradução Ana Maria Netto Guz. Rio de Janeiro: Campus, 1997. 388 p.

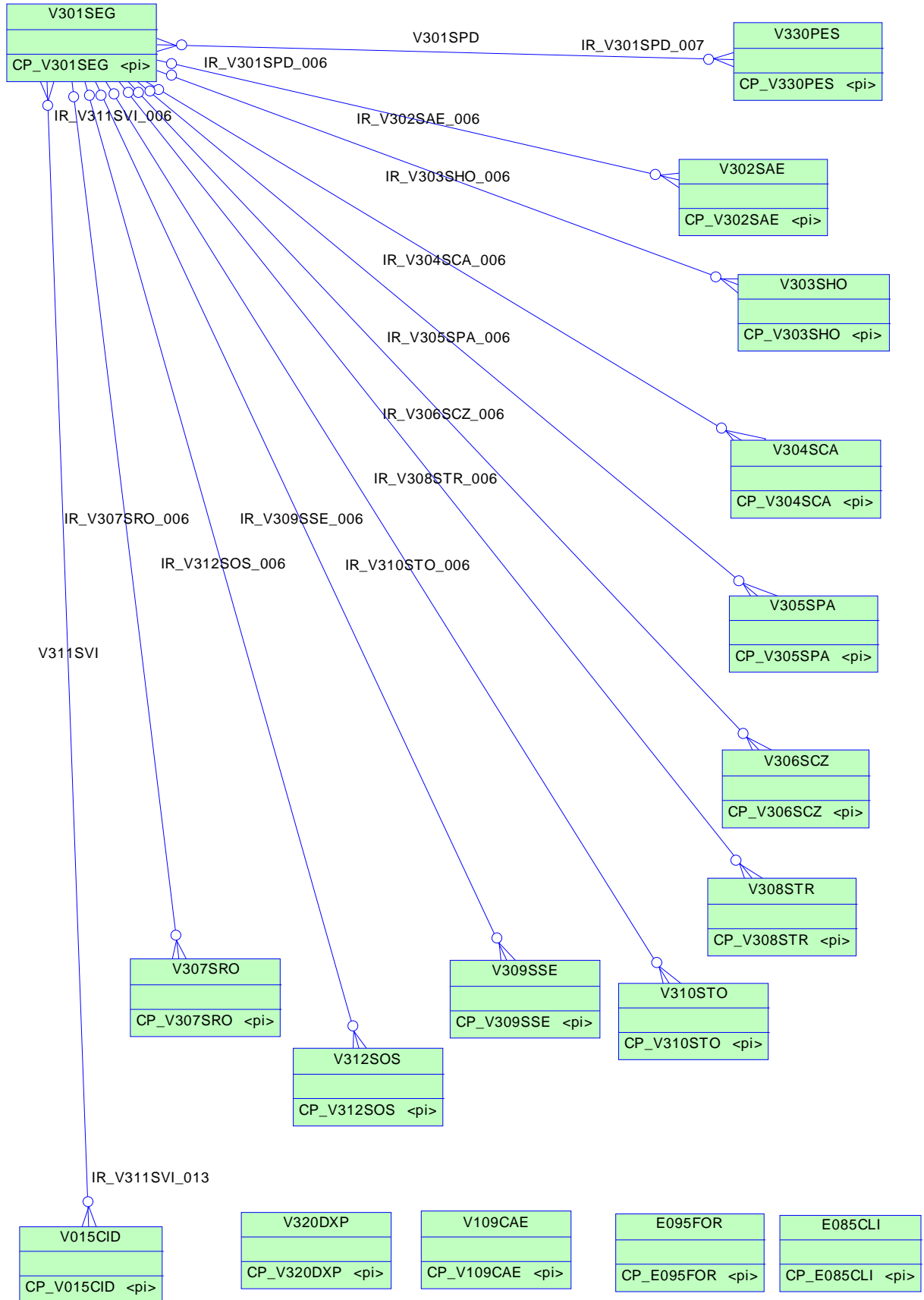
KERN, Vinícius Medina. **Banco de dados relacionais: teoria e prática de projeto**. São Paulo: Érica. 1994. 228 p.

KIMBALL, Ralph; ROSS, Margy ***The data warehouse toolkit*; o guia completo para modelagem multidimensional**. Tradução Ana Beatriz Tavares e Daniela Lacerda. Rio de Janeiro: Campus, 2002. 494 p.

SCHATZMANN, Eduardo. **Sistema de informações gerenciais da área comercial de uma empresa aplicando *Data Warehouse***. 2003. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SERRA, Laércio. **A essência do *business intelligence***. São Paulo: Berkeley Brasil, 2002. 288 p.

APÊNDICE A – MER Lógico da base de dados operacional



APÊNDICE B – Nomenclatura das tabelas

E085CLI – Cadastros – Clientes;

E095FOR – Cadastros – Fornecedores;

V015CID – Cadastros – Cidades;

V109CAE – Cadastros – Aeroportos;

V301SEG – Vendas – Segmentos;

V301SPD – Vendas – Segmentos X Passageiros X Documentos;

V302SAE – Vendas – Aéreo – Segmentos;

V303SHO - Vendas – Hotel – Segmentos;

V304SCA - Vendas – Carro – Segmentos;

V305SPA - Vendas – Pacotes – Segmentos;

V306SCZ - Vendas – Cruzeiros – Segmentos;

V307SRO - Vendas – Rodoviários – Segmentos;

V308STR – Vendas – Trem – Segmentos;

V309SSE – Vendas – Seguro de Viagem – Segmentos;

V310STO – Vendas – Tours – Segmentos;

V311SVI – Vendas – Vistos – Segmentos;

V312SOS - Vendas – Outros Serviços – Segmentos;

V320DXP – Vendas – Documento X Passageiro;

V330PES – Vendas – Pessoas do Processo Comercial.

APÊNDICE C – Script para extração dos dados da base operacional

```

drop table fato_vendas
go
drop table dim_produto
go
drop table dim_cliente
go
drop table dim_fornece
go
create table dim_produto (codemp smallint, codfil smallint, codpst smallint, anopcv smallint,
numpcv integer,
                venorc varchar(1), segsis smallint, seqdoc smallint, pesis smallint,
seqdpx smallint, tippdt varchar(3),
                despro varchar(20), oripro varchar(30), cidori varchar(30), ciddst
varchar(30), dstpro varchar(30),
                codfor integer, nomfor varchar(90),
primary key (codemp, codfil, codpst, anopcv, numpcv, venorc, segsis, seqdoc, pesis,
seqdpx))
go
create table dim_cliente (codcli integer, nomcli varchar(100), apecli varchar(60), endcli
varchar(100), cplend varchar(40),
                cepcli integer, baicli varchar(20), cidcli varchar(60), estcli varchar(2),
paicli varchar (20),
                foncli varchar(25),
primary key (codcli))
go
create table dim_fornece (codfor integer, nomfor varchar(100), apefor varchar(60), endfor
varchar(100), cplend varchar(40),
                cepfor integer, baifor varchar(20), cidfor varchar(60), estfor
varchar(2), paifor varchar (20),
                fonfor varchar(25),
primary key (codfor))
go
create table fato_vendas (codemp smallint, codfil smallint, codpst smallint, anopcv smallint,
numpcv integer,
                venorc varchar(1), segsis smallint, seqdoc smallint, pesis smallint,
seqdpx smallint, tippdt varchar(3),
                despro varchar(20), qtdven integer, vlrven numeric(11,2), vlrgan
numeric (11,2), codcli integer, nomcli varchar(100),
                codfor integer, nomfor varchar(100), codfop varchar(5), codmfp
varchar(5), datven datetime,
primary key (codemp, codfil, codpst, anopcv, numpcv, venorc, segsis, seqdoc, pesis,
seqdpx),
foreign key (codemp, codfil, codpst, anopcv, numpcv, venorc, segsis, seqdoc, pesis, seqdpx)
references dim_produto(codemp, codfil, codpst, anopcv, numpcv, venorc, segsis, seqdoc,
pesis, seqdpx),
foreign key (codcli) references dim_cliente(codcli),
foreign key (codfor) references dim_fornece(codfor))
go

```

```

insert into dim_produto select s.codemp, s.codfil, s.codpst, s.anopcv, s.numpcv, s.venorc,
s.segsis, p.seqdoc, p.pessis, d.seqdpx, s.tippdt,
        despro = case s.tippdt when 'AE' then 'Aéreo'
                    when 'HO' then 'Hotel'
                    when 'RO' then 'Rodoviário'
                    when 'CA' then 'Carro'
                    when 'SE' then 'Seguro'
                    when 'OS' then 'Outros Serviços'
                    when 'TR' then 'Trem'
                    when 'VI' then 'Visto'
                    when 'CZ' then 'Cruzeiro'
                    when 'TO' then 'Tour'
                    else 'Pacote' end,
        s.aecori,
        cidori = isnull(case s.tippdt when 'AE' then (select c.nomcid from
v015cid c, v109cae a where s.aecori=a.codaer and c.codcid=a.codcid)
                    else (select c.nomcid from v015cid c where
c.codcid=s.aecori) end,' '),
        s.aecdst,
        ciddst = isnull(case s.tippdt when 'AE' then (select c.nomcid from
v015cid c, v109cae a where s.aecdst=a.codaer and c.codcid=a.codcid)
                    else (select c.nomcid from v015cid c where
c.codcid=s.aecdst) end,' '),
        s.codfor, f.nomfor
from v301seg s, v301spd p, v320dpx d, e095for f
where s.codfor=f.codfor
and s.codemp=p.codemp
and s.codfil=p.codfil
and s.codpst=p.codpst
and s.anopcv=p.anopcv
and s.numpcv=p.numpcv
and s.venorc=p.venorc
and s.segsis=p.segsis
and d.codemp=p.codemp
and d.codfil=p.codfil
and d.codpst=p.codpst
and d.anopcv=p.anopcv
and d.numpcv=p.numpcv
and d.venorc=p.venorc
and d.seqdoc=p.seqdoc
and d.pessis=p.pessis

go
insert into dim_cliente select c.codcli, c.nomcli, c.apecli, c.endcli, c.cplend, c.ceplici, c.baicli,
c.cidcli,
        c.sigufs, c.codpai, c.foncli
from e085cli c

go
insert into dim_fornece select f.codfor, f.nomfor, f.apefor, f.endfor, f.cplend, f.cepfor, f.baifor,
f.cidfor,

```

```

        f.sigufs, f.codpai, f.fonfor
    from e095for f

go
insert into fato_vendas select d.codemp, d.codfil, d.codpst, d.anopcv, d.numpcv, d.venorc,
s.segsis, d.seqdoc, d.pessis,
        d.seqdpx, s.tippdt, despro = case s.tippdt when 'AE' then 'Aéreo'
        when 'HO' then 'Hotel'
        when 'RO' then 'Rodoviário'
        when 'CA' then 'Carro'
        when 'SE' then 'Seguro'
        when 'OS' then 'Outros Serviços'
        when 'TR' then 'Trem'
        when 'VI' then 'Visto'
        when 'CZ' then 'Cruzeiro'
        when 'TO' then 'Tour'
        else 'Pacote' end,
        s.qtdite, (d.vlrtab + d.vlrtxb - d.vlrdna + d.vlrren) as vlrven,
        (d.vlrtab + d.vlrtxb - d.vlrcfn - d.vlrifn) as vlrgran,
        d.codcli, e.nomcli, s.codfor, f.nomfor, d.codfop, d.codmfp, d.dataalt
from v320dpx d, v301spd p, v301seg s, e085cli e, e095for f
where d.codemp=p.codemp
and d.codfil=p.codfil
and d.codpst=p.codpst
and d.anopcv=p.anopcv
and d.numpcv=p.numpcv
and d.venorc=p.venorc
and d.seqdoc=p.seqdoc
and d.pessis=p.pessis
and s.codemp=p.codemp
and s.codfil=p.codfil
and s.codpst=p.codpst
and s.anopcv=p.anopcv
and s.numpcv=p.numpcv
and s.venorc=p.venorc
and s.segsis=p.segsis
and d.codcli=e.codcli
and s.codfor=f.codfor

go

```

APÊNDICE D – Dicionários de dados das tabelas do modelo dimensional

Tabela de fatos:

- Tabela fato_vendas (valores referentes às vendas):

tippdt - tipo de produto - varchar(3);

qtdven - quantidade vendida - integer;

vlrven - valor de venda - numeric(11,2);

vlrgan - valor de ganho - numeric(11,2);

codfop - código de forma de pagamento - varchar(5);;

codmfp - código de modalidade da forma de pagamento - varchar(5);

datven - data da venda - datetime.

Tabelas de dimensões:

- Tabela dim_produto (características dos produtos cadastrados)

codemp - código da empresa - smallint;

codfil - código da filial - smallint;

codpst - código do posto - smallint;

anopcv - ano do processo comercial - smallint;

numpcv - nro do processo comercial - integer;

venorc - flag para venda ou orçamento - varchar(1);

segsis - segmento de sistema - smallint - ;

seqdoc - sequência de documento - smallint;

pessis - pessoa do sistema - smallint;

seqdpx - sequência de documento x passageiro - smallint;

tippdt - tipo de produto - varchar(3);

despro - descrição de produto - varchar(20);

oripro - código da origem do produto - varchar(30);

cidori - cidade de origem do produto - varchar(30);

ciddst - código da cidade de destino do produto - varchar(30);

dstpro - descrição da cidade de destino do produto - varchar(30);

codfor - código do fornecedor - integer;

nomfor - nome do fornecedor - varchar(90).

- Tabela dim_fornece (características dos fornecedores cadastrados)

codfor - código do fornecedor - integer;

nomfor - nome do fornecedor - varchar(90);

apefor - apelido do fornecedor - varchar(60);

endfor - endereço do fornecedor - varchar(100);

cplend - complemento do endereço - varchar(40);

cepfor - cep do fornecedor - integer;

baifor - bairro do fornecedor - varchar(20);

cidfor - cidade do fornecedor - varchar(60);

estfor - estado do fornecedor - varchar(2);

paifor - país do fornecedor - varchar(20);

fonfor - telefone do fornecedor - varchar(25).

- Tabela dim_cliente (características dos clientes cadastrados)

codcli - código do cliente - integer;

nomcli - nome do cliente - varchar(100);

apecli - apelido do cliente - varchar(60);

endcli - endereço do cliente - varchar(100);

cplend - complemento do endereço - varchar(40);

cepcli - cep do cliente - integer;

baicli - bairro do cliente - varchar(20);

cidcli - cidade do cliente - varchar(60);

estcli - estado do cliente - varchar(2);

paicli - país do cliente - varchar(20);

foncli - telefone do cliente - varchar(25).