UNIVERSIDADE CÂNDIDO MENDES
CAMPOS DOS GOYTACAZES
MESTRADO EM INFORMÁTICA APLICADA

GIOVANNI COLONESE

UMA FERRAMENTA ABERTA DE DESENVOLVIMENTO INTEGRADO DE SISTEMAS DE INFORMAÇÃO PARA PROCESSAMENTO ANALÍTICO E GEOGRÁFICO

CAMPOS DOS GOYTACAZES
2004

GIOVANNI COLONESE

UMA FERRAMENTA ABERTA DE DESENVOLVIMENTO INTEGRADO DE SISTEMAS DE INFORMAÇÃO PARA PROCESSAMENTO ANALÍTICO E GEOGRÁFICO

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade Cândido Mendes, como requisito parcial para a obtenção do grau de Mestre. Área de Concentração: Sistemas de Informação e Apoio à Decisão.

Orientador: Prof. Astério Kiyoshi Tanaka, PhD
Co-Orientador: Prof. Rogério Atem de Carvalho, DSc

Campos dos Goytacazes
2004

GIOVANNI COLONESE

UMA FERRAMENTA ABERTA DE DESENVOLVIMENTO INTEGRADO DE SISTEMAS DE INFORMAÇÃO PARA PROCESSAMENTO ANALÍTICO E GEOGRÁFICO

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade Cândido Mendes, como requisito parcial para a obtenção do grau de Mestre. Área de Concentração: Sistemas de Informação e Apoio à Decisão.

Aprovada em _____

BANCA EXAMINADORA

_____

Prof. Astério Kiyoshi Tanaka, PhD – Orientador
Universidade Cândido Mendes

_____

Prof. Rogério Atem de Carvalho, DSc – Co-Orientador
Universidade Cândido Mendes

_____

Prof. Jugurta Lisboa Filho, DSc
Universidade Federal de Viçosa (UFV)

_____

Prof. Rubens Nascimento Melo, DSc
Pontifícia Universidade Católica (PUC – Rio)

Campos dos Goytacazes
2004

À minha família, pelo apoio sempre seguro e incondicional, pelo amor e pela fé.

Aos meus amigos, companheiros de caminhada e cúmplices nas dificuldades e nos êxitos.

AGRADECIMENTOS

Em primeiro lugar a Deus, sem o qual nada seria possível, pelo inestimável dom da vida e por tudo aquilo que sou e possuo.

À minha esposa Joziane e meus filhos Anna Luisa e Gabriel, pela força necessária nos momentos mais difíceis, por terem compreendido minha ausência ao longo desta jornada, e por terem sido o porto seguro onde eu restabelecia minhas energias.

Aos meus pais, Liberato e Emília, pelo apoio incondicional, pelo carinho e pelo exemplo de vida, e meus irmãos, Daniel e Sandro, pela fraternidade e companheirismo que sempre nos uniu.

Aos meus orientadores, Prof. Tanaka e Prof. Rogério, pela amizade, pelo clima maravilhoso de cumplicidade no conhecimento com que sempre conduziram minhas orientações; pela amizade e dedicação com que sempre iluminaram meus caminhos, tornando-se para mim, exemplos de Mestres, Amigos e Pesquisadores.

Aos meus companheiros do mestrado, pela convivência sempre agradável e por termos nos tornado efetivamente uma "turma", coesa e firme em seus propósitos de busca do conhecimento. Em especial aos outros dois integrantes da turma de Macaé, Alan e Sérgio, amigos inseparáveis e cúmplices em tantas dificuldades e conquistas: a jornada teria sido praticamente impossível sem eles. Uma lembrança ao Alan, que em inúmeros momentos, sem medir fins de semana ou noites de descanso, deixou seu próprio trabalho para ajudar-me nas tarefas mais penosas e desafiantes, tudo em nome da paixão pelo desafio de aprender.

Aos amigos Prof. Nunes e Prof. Thomé, pela amizade sincera, pelo exemplo e pela preocupação com que sempre acompanharam meus passos, pela orientação segura e pelos incentivos.

SUMÁRIO

TABELAS

FIGURAS

LISTA DE ABREVIATURAS

COM – Component Object Model

CORBA – Common Object Request Broker Architecture

CWM – Common Warehouse Metamodel

DCE – Distributed Computing Environment

DCOM – Distributed Component Object Model

DW – Data Warehouse

DED – Diagrama de Estrutura de Dados

DER – Diagrama Entidade-Relacionamento

ETL – Extract, Transform and Load (Extrair, Transformar e Carregar)

ER – Entidade-Relacionamento

GIS – Geographical Information System

HOLAP – Hybrid On-Line Analytical Processing

KDD – Knowledge Discovery in Databases

MOLAP - Multidimensional On-Line Analytical Processing

OGC – Open Gis Consortium

OLAP – On-Line Analytical Processing

OLE – Object Linking and Embedding

OLTP – On-Line Transaction Processing

OMG – Object Management Group

OO – Orientação a Objetos

PDV – Ponto de Venda

ROLAP - Relational On-Line Analytical Processing

SGBD – Sistema de Gerenciamento de Banco de Dados

SGBD-OR – Sistema de Gerenciamento de Banco de Dados–Objeto-Relacional

SIG – Sistema de Informações Geográficas

SQL – Structured Query Language

SRID – System Reference ID (Identificador do Sistema de Referência – Geográfica-)

UML – Unified Modeling Language

WKB – Well-Known Binary

XML – Extensible Markup Language

RESUMO

A integração entre Sistemas Geográficos (GIS) e Analíticos (Data Warehouses e OLAP) em um único Sistema de Suporte à Decisão tem sido o foco de trabalho de muitos pesquisadores ao longo dos últimos anos, de forma a prover a capacidade de analisar todos os aspectos que influenciam uma organização: o próprio negócio, o tempo e o espaço. Diferentemente da maioria dos trabalhos nesta área, que abordam a integração entre dois sistemas pré-existentes: um Geográfico (GIS) e outro Analítico (OLAP), o objetivo do presente trabalho é a apresentação de uma proposta de desenvolvimento de sistemas onde a integração entre os componentes analíticos e geográficos esteja presente desde o nível conceitual até o nível de implementação. A construção de uma ferramenta de desenvolvimento comprova tal integração de funcionalidades analíticas e geográficas. A ferramenta PostGeoOlap é uma aplicação OLAP que trabalha sobre um Banco de Dados Objeto-Relacional Espacial, o PostGreSQL acrescido do PostGIS (extensão Espacial para o PostGreSql), e que executa todas as funcionalidades analíticas e geográficas sobre um Data Warehouse, tornando realidade a modelagem unificada por meio de estereótipos proposta no presente trabalho. A visualização dos dados em mapas é executada pelo componente PlanetGIS, mas poderia ser obtida por meio de qualquer outra ferramenta ou componente gráfico capaz de receber coordenadas e plotá-las, já que nenhuma capacidade de processamento geográfico é requerida do tal componente de visualização, pois todo o processamento, inclusive o geográfico, é realizado pelo SGBD Espacial. A abordagem do presente trabalho para a integração das tecnologias geográficas e analíticas traz como grande vantagem a simplificação de todo o processo de desenvolvimento: da modelagem unificada até a implementação da aplicação final (o que diminui o esforço de modelagem e aumenta a consistência entre os dois vetores do sistema: o analítico e o geográfico). Além disso, a utilização de componentes de código aberto ou gratuitos, viabiliza seu uso por muitas organizações antes limitadas pelos altíssimos custos de aquisição ou licenciamento impostos por tais tipos de sistemas.

ABSTRACT

The integration among analytical (DW/OLAP) and geographical (GIS) components in a Decision Support System has been the focus of many researchers' work, in order to provide to the decision-makers conditions for analyzing the dimensions that influence an organization: business, time and space. The present work aims not to do the integration of two pre-existent systems: a Geographical (GIS) and an Analytical (OLAP), but to propose a development technique for building a whole new system, where the integration of the geographical and analytical components is present since its conceptual level. The implementation of a development tool demonstrates such integration of analytical and geographical functionalities. PostGeoOlap is an OLAP application that works on a Spatial DBMS, PostGreSQL, extended by PostGIS (Spatial extension for PostGreSql), that executes all the analytic and geographical functionalities on a Data Warehouse, making true the unified modeling approach using stereotypes proposed in this work. Data visualization in maps is executed by the component (COM Object) PlanetGIS, but it could be made by any other tool or graphical component capable of receiving and drawing coordinates, since no geographical processing is requested by the visualization component, because the whole processing, including the geographical one, is accomplished by the Spatial DBMS. The present work's approach for the integration of geographical and analytical technologies brings as a great advantage the simplification of the whole development process: from the unified modeling to the implementation of the final application (thus decreasing the modeling effort and leading to a greater consistency among the two dimensions of the system: analytical and geographical). In addition, the use of free software components, makes it feasible the use of such types of systems by many budget-restricted organizations.

1. INTRODUÇÃO

As três últimas décadas assistiram a um fenomenal crescimento na quantidade de dados gerados pelas cada vez mais informatizadas organizações, que os utilizavam, a princípio, apenas para atender às suas necessidades operacionais. Com o aumento da eficiência e precisão nas atividades operacionais proporcionados pelos sistemas transacionais, as organizações viram-se imersas em um mundo ainda mais competitivo, onde "estar informatizada" e "possuir dados a respeito de clientes, produtos e vendas" era mero lugar-comum. Fazia-se necessário não apenas possuir os dados, mas ser capaz de extrair destes mesmos dados, informação e até mesmo conhecimento capaz de gerar um real diferencial competitivo, ou seja, de permitir a tomada de decisões mais acertadas e proporcionar melhores conduções nos negócios. Desta forma, surgiram os Sistemas de Suporte à Decisão, que, segundo D. J. Power (DSSRESOURCE.COM, 2003) são uma classe específica de sistemas de informação computadorizados que visam, de forma interativa, ajudar os decisores a utilizar dados, conhecimento, sistemas de comunicação e modelos na identificação e solução de problemas de forma a apoiar a tomada de decisão.

Os Sistemas de Suporte à Decisão integram diversas tecnologias tais como Data Warehousing, Data Mining, OLAP (On-line Analytical Processing), GIS (Geographical Information Systems) e KDD (Knowledge Discovery in Databases) na produção de um "console de comando" onde o Decisor (normalmente um gerente ou diretor de uma organização) tenha visualização, sob várias perspectivas diferentes, acerca dos negócios da

corporação. A capacidade de analisar dados agregados e integrados a partir de diversas fontes faz do Data Warehouse (DW) aliado a uma aplicação OLAP (On-Line Analytical Processing) ferramentas indispensáveis aos decisores (decision makers) das organizações, provendo medições em tempo real acerca das diversas perspectivas do negócio e do tempo. Outra tecnologia também historicamente ligada ao suporte à decisão é a dos SIG (Sistemas de Informação Geográficos) ou, em Inglês, GIS (Geographical Information Systems), que utilizam mapas para auxiliar seus usuários a analisar dados que possuam como um de seus atributos o posicionamento geográfico, trazendo ao usuário dados acerca do negócio aliados à componente espacial.

Há vários anos, grande esforço vem sendo empreendido pelos pesquisadores no sentido de integrar as tecnologias analíticas (OLAP) e geográficas (SIG) em uma única aplicação de forma a prover um suporte à decisão mais completo e que permita análise sob as perspectivas do negócio, do tempo e do espaço.

Grande parte dos trabalhos que buscam a integração geográfica e analítica foca a junção de uma aplicação GIS com outra OLAP já existentes, fazendo um mapeamento entre seus resultados de forma a prover, em uma terceira aplicação, a almejada integração de resultados.

O presente trabalho busca apresentar uma técnica de modelagem e uma ferramenta de integração (PostGeoOlap) que permita a integração das tecnologias analítica e geográfica desde a concepção inicial de um sistema, ou seja, desde sua modelagem conceitual, até sua implementação, visando atender a sistemas de suporte à decisão onde a coexistência das dimensões tempo e espaço faz-se necessária desde o mais alto nível de abstração e onde se deseje como produto uma única aplicação capaz de integrar características OLAP e SIG.

1.1 MOTIVAÇÃO

Ter a capacidade de “enxergar do alto” todos os fatores que influenciam determinado negócio sempre foi o maior dos objetivos buscados por aqueles responsáveis por decidir o futuro de uma organização. Desta forma, possuir um sistema de informação que permita a mensuração de valores segundo os aspectos do próprio negócio (Produtos, Clientes,

Fornecedores, Rotas de Entrega, etc), aliados ao aspecto Tempo e ao Espaço, traria total entendimento acerca das variáveis que determinam a função de comportamento de uma organização. O uso concomitante das noções de tempo e espaço tem sido obtido historicamente através da integração de duas ferramentas distintas: uma para a componente Analítica do Sistema (OLAP) e outra para a parte geográfica do Sistema (GIS – Geographical Information Systems). Tal abordagem, embora interessante para aqueles que já possuem as citadas aplicações e desejam fazer sua integração, é bastante dispendiosa para os projetos que buscam tal integração desde o mais alto nível de abstração, ou seja, para sistemas que serão construídos "do zero", já que geram redundância de modelagem e de armazenamento de informações (pois teremos dois sistemas sendo mantidos em paralelo, um OLAP e outro Geográfico). Desta forma, a falta de uma metodologia que permitisse, de forma simples, a modelagem e implementação de sistemas que integrem características Analíticas e Geográficas desde sua concepção conceitual foi a grande motivação para o presente trabalho.

1.2 OBJETIVOS DO TRABALHO

O objetivo desta dissertação é propor uma metodologia e uma ferramenta que permitam construir sistemas de suporte à decisão onde, desde a fase de modelagem conceitual, seja possível tratar aspectos analíticos e geográficos, permitindo ainda um mapeamento "direto" e simples (sem a necessidade de transformações ou traduções) do mais alto nível de abstração para a implementação, simplificando com isso a produção de sistemas que integram características OLAP e SIG, principalmente aqueles a serem construídos "do zero", ou seja, ambientes onde não pré-existam uma aplicação OLAP e outra SIG a serem integradas.

Para tanto, este documento inicia apresentando a metodologia que possibilita integrar, em nível conceitual, características analíticas e geográficas em uma única modelagem. Em seguida mostra que tal modelagem teria seu destino perfeito em uma aplicação que suportasse os conceitos OLAP e SIG ali representados e, para mostrar tal simplicidade de mapeamento, uma ferramenta com tais funcionalidades foi desenvolvida (PostGeoOlap). Por fim, um estudo de caso é apresentado para ilustrar todo o ciclo de emprego da metodologia e da ferramenta sobre um caso real.

1.3 ORGANIZAÇÃO DO TRABALHO

A presente dissertação está organizada como se segue: como a principal motivação para este trabalho é a integração de sistemas de informação analíticos e geográficos desde sua mais abstrata concepção, o capítulo 2 trata da modelagem de Data Warehouses utilizando a notação UML, o capítulo 3 fala dos tipos de dados geográficos propostos pelo OpenGis e que serão utilizados como estereótipos em nossa modelagem conceitual, no capítulo 4 é abordado o conseqüente mapeamento do modelo de DW em UML, usando estereótipos geográficos, para o modelo dimensional relacional; o capítulo 5, por sua vez, trata da ferramenta capaz de tornar realidade a modelagem conceitualmente integrada dos conceitos analíticos e Geográficos, o capítulo 6 aborda um estudo de caso ilustrativo do emprego de nossa proposta de integração entre sistemas OLAP e GIS e, finalmente, o capítulo 7 traz as conclusões, as contribuições desta dissertação e sugestões para trabalhos futuros.

2. MODELAGEM CONCEITUAL DE DATA WAREHOUSE USANDO UML

Modelar é representar a realidade por meio de uma simplificação. Um modelo conceitual representa a realidade a ser implementada em um sistema de informação, de forma independente de detalhes tecnológicos como hardware, sistema operacional ou modelo de SGBD (Sistema de Gerenciamento de Banco de Dados). Por seu alto nível de abstração, o modelo conceitual permite visualizar o sistema como se deseja que ele venha a ser quando implementado e ainda facilita a comunicação entre os projetistas e usuários acerca da realidade a ser representada, já que é desenhado sem as restrições impostas pela tecnologia que o implementará.

Ralph Kimball (KIMBALL, 1996), apresenta, como proposta de modelo para data warehouses o esquema estrela que, segundo (SCHRAML, 2002) descreve arranjos dimensionais de tabelas de bancos de dados relacionais usados para análise de dados em sistemas de suporte à decisão. Durante a modelagem de um esquema estrela, os dados são divididos em dois grupos: itens numéricos usados na execução de cálculos e itens não-numéricos utilizados para filtragem e ordenação dos resultados das consultas. Os itens numéricos são chamados de medições e são armazenados como atributos das tabelas “Fato”; os não-numéricos são chamados de descritores e armazenados nas tabelas “Dimensão” (que representam os eixos de análise do sistema, ou as perspectivas segundo as quais os dados numéricos serão analisados). Um aspecto decisivo para um projeto de data warehouse é a granularidade dos itens numéricos a serem representados na tabela Fato, que em verdade

representam os resultados do processo de negócio a ser analisado e que serão restringidos pelos atributos das tabelas dimensão.

O modelo Estrela, proposto por Kimball para a modelagem de Data Warehouses, se tornou praticamente um padrão adotado pela quase totalidade dos desenvolvedores, trazendo como grande benefício um diagrama limpo e objetivo, que consegue dar destaque às entidades mais importantes e que evidencia onde estão os valores numéricos do negócio e onde estão armazenadas as informações descritivas, praticamente estáticas, dos objetos. Tal modelo, apesar de se constituir em um padrão de fato, é um modelo lógico, ou seja, que visa a implementação do sistema dimensional em tecnologia relacional, daí a recomendação de "desnormalização" das dimensões, visando diminuir a quantidade de junções de tabelas e conseqüentemente incrementar o desempenho do sistema como um todo. A desvantagem da utilização do modelo estrela de Kimball como documentação de mais alto nível de abstração para um sistema dimensional, ou seja, usar um modelo lógico ao invés de um conceitual, é que nas dimensões, onde a desnormalização é aplicada, perde-se informação acerca das associações ou relacionamentos entre as diversas entidades do mundo real. Assim, o ideal para se obter o máximo em representatividade de um modelo dimensional seria construí-lo em um nível de abstração conceitual, ou seja, independente ou não restrito em sua formatação pela tecnologia que o implementará. Neste modelo conceitual seria utilizado o esquema estrela de Kimball (no tocante à disposição gráfica do modelo: com uma entidade central contendo valores a mensurar e outras entidades ao redor desta primeira contendo as perspectivas para as análises), devido à já citada simplicidade e evidência dada aos valores relevantes para a análise gerencial, mas a representação das informações contidas nas dimensões seria livre, ou seja, sem a restrição da "desnormalização", de forma a obter máxima expressividade do diagrama. A preocupação com desempenho seria delegada a uma fase posterior quando seria feito o mapeamento do modelo conceitual para o lógico.

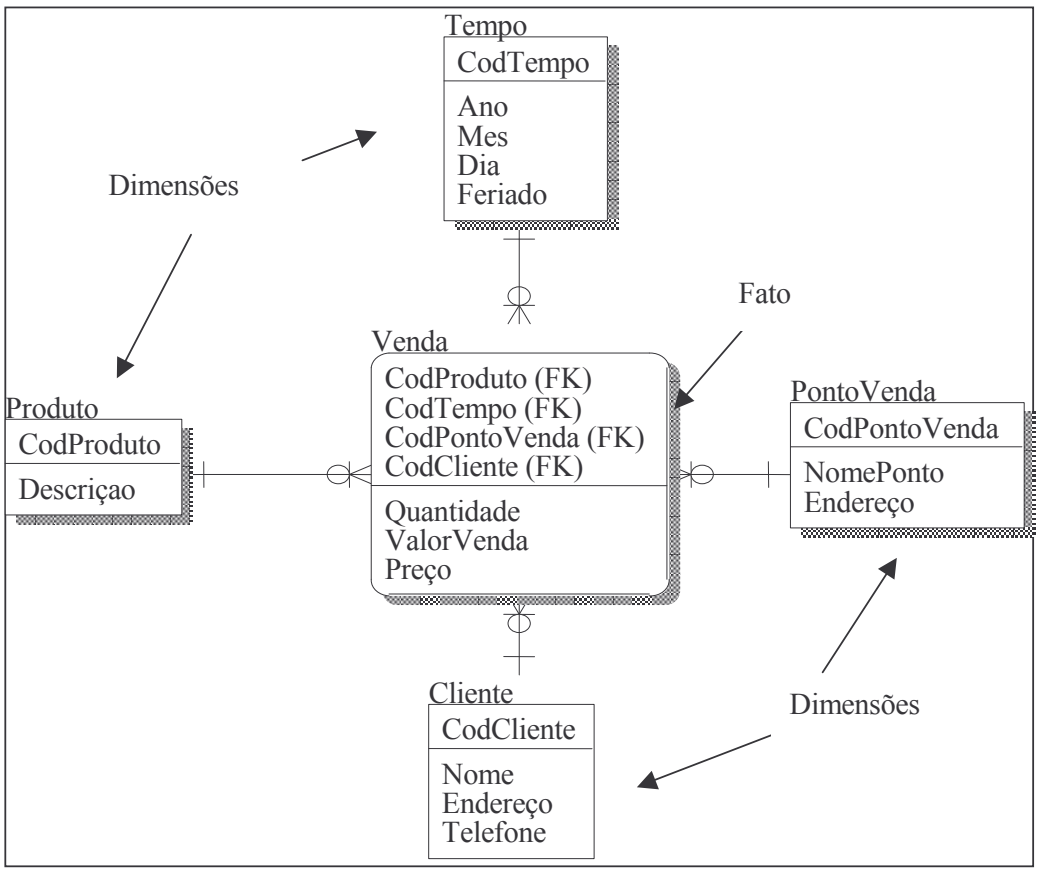


Fig.1 Modelo Estrela proposto por Ralph Kimball

Para a modelagem Conceitual de sistemas dimensionais, diversos modelos vêm sendo utilizados, entre eles o E-R (Entidade-Relacionamento) e o Diagrama de Classes da UML. O uso do diagrama de classes da UML para a modelagem conceitual dimensional, se justifica no fato de que a UML, como linguagem representante do paradigma da Orientação a Objetos, é semanticamente mais rica do que a notação E-R. Segundo (DORSEY, 2002) tudo o que pode ser feito em um diagrama E-R pode também ser feito com UML e as poucas coisas não existentes podem ser facilmente adicionadas através de extensões. Modelar dados através da UML torna também mais elegante o mecanismo de hierarquia (que é inerente ao modelo de Objetos), permitindo a modelagem de conceitos genéricos que podem ser estendidos na medida de sua necessidade. Ainda segundo (TRUJILLO, 2001), o uso de UML para projetos conceituais de Data Warehouses é interessante por sua capacidade de representar as propriedades estruturais e dinâmicas de um sistema de forma mais natural do que abordagens clássicas como a Entidade-Relacionamento. Além disso, a UML possui poderosos

mecanismos como, por exemplo, a OCL (Object Constraint Language) para embutir restrições e requisitos iniciais dos usuários no modelo conceitual.

Segundo (FIRESTONE, 2003) a abordagem atual utilizada nas decisões que orientam a implementação de data warehouses têm sido pragmática e, embora tenha obtido sucesso comercial, torna o acoplamento de objetivos estratégicos aos resultados do data warehouse um mero produto da tecnologia e não de um método ou procedimento explícito. Embora a utilização de uma abordagem Orientada a Objeto não elimine o produto tecnológico ao final do processo, a capacidade de rastreamento dos objetivos estratégicos através dos processos de negócio, dos subprocessos, dos casos de uso e das classes até as decisões-chave para a implementação do Data warehouse, irá, com certeza, incrementar em muito a qualidade da prática na área de modelagem dimensional.

Pelo anteriormente exposto, podemos concluir que utilizar Orientação a Objetos em sua notação UML para modelar data warehouses traz como grandes vantagens a já comprovada riqueza semântica da linguagem (UML), o fato de integrar tal modelo dimensional com todos os demais modelos de objetos de uma organização, trazendo coerência de representação nas diversas áreas do negócio e ainda a facilidade inerente ao modelo de objetos de produzir sistemas genéricos (frameworks) representativos de um grupo de realidades similares mas extensíveis a ponto de poder contemplar, com algumas adições ao modelo original, necessidades específicas.

No exemplo a seguir, apresentamos um modelo dimensional (apenas com alguns atributos nas classes por medida de simplificação de representação) conceitual Orientado a Objeto para uma distribuidora de produtos: A classe fato está representada por VENDA e as dimensões por PRODUTO, PONTOVENDA, TEMPO e CLIENTE.

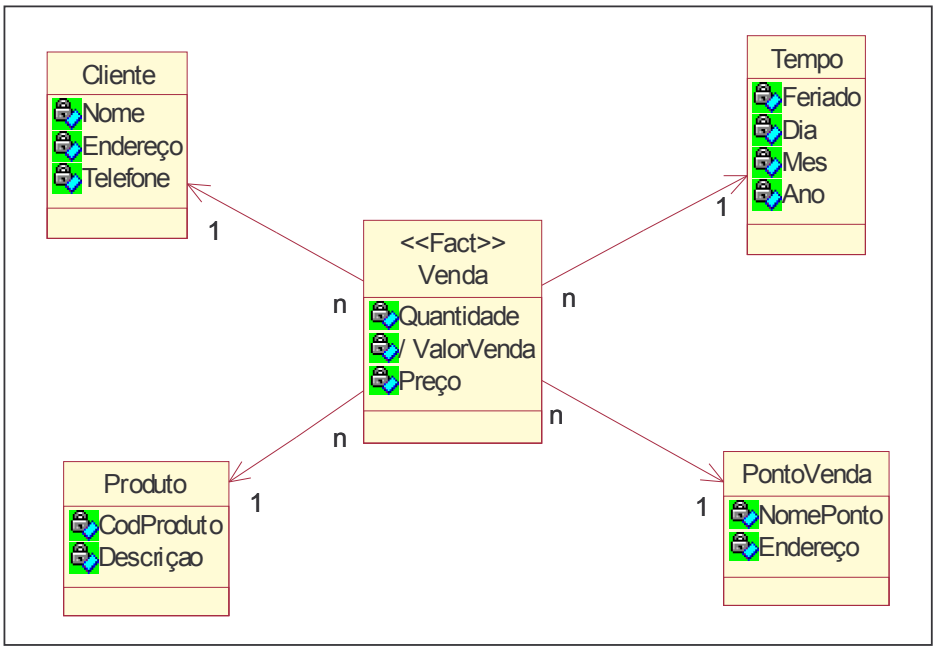


Fig.2 Sistema Dimensional Conceitual OO

No exemplo da figura 2, o sistema dimensional representado destina-se a uma distribuidora de produtos genérica (de forma a atender toda uma família de distribuidoras com características similares, como uma espécie de framework) , cuja granularidade é a venda de quantidades de certo produto, em determinada data, em certa loja e para certo cliente.

Tal modelo pode ser estendido em suas dimensões, de forma a atender uma distribuidora em especial que possua além das características genéricas, particularidades específicas de seu ramo de atuação, como se pode verificar na figura 3. A classe fato, que representa as medições acerca do negócio, não sofrerá extensões, uma vez que estas poderiam alterar a granularidade e as dimensões necessárias a analisar tais valores.

Na Figura 3, a classe produto foi estendida a partir de sua forma inicial genérica, de forma a contemplar informações acerca de categoria e das edições nas quais se desdobram os produtos (no caso, revistas) comercializados pela distribuidora em questão. No exemplo dado, as extensões foram expressas por meio de uma agregação (classe “Categoria”) e de uma composição (classe “Ediçao”), mas outras extensões poderiam ser aplicadas ao modelo, como por exemplo a criação de uma hierarquia de tipos de produtos, como mostrado na figura 4.

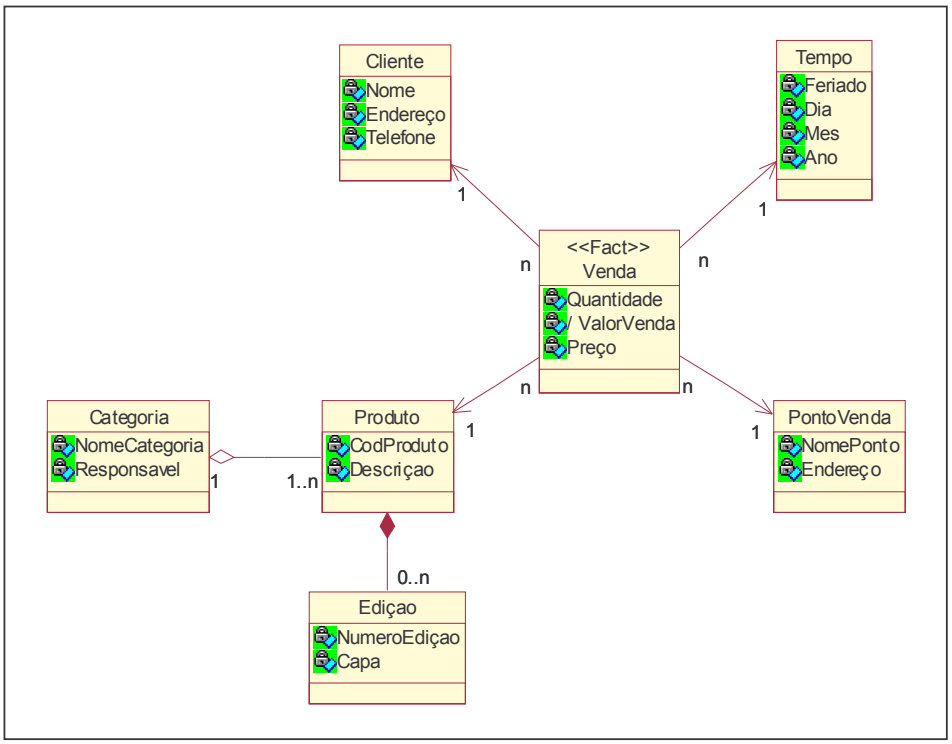


Fig.3 Diagrama dimensional conceitual OO com especializações da classe Produto (por meio do uso de agregação e composição)

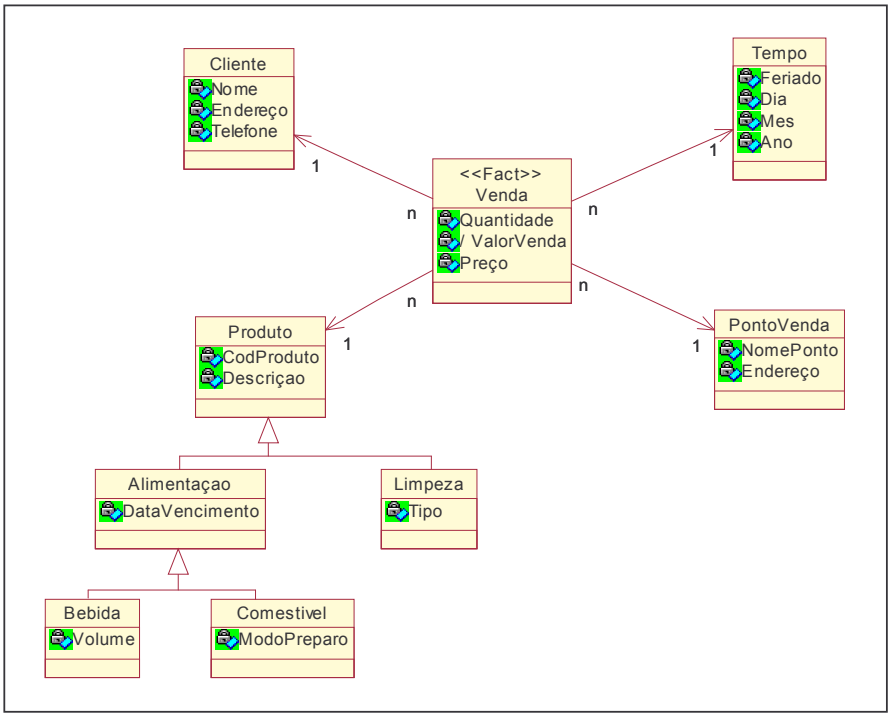


Fig.4 Modelo Conceitual dimensional OO com especializações da classe Produto (uso de hierarquia de tipos de produtos)

Além de todas as vantagens já citadas, o emprego da orientação a objetos na modelagem de Data Warehouses traz também a importante possibilidade do uso de estereótipos, que simplificam sobremaneira a representação de hierarquias extensas de objetos, substituindo a reprodução de uma associação entre uma classe e toda a extensa hierarquia por um símbolo ou ícone representativo da classe com a qual o objeto corrente possui uma relação do estilo: "é representado por".

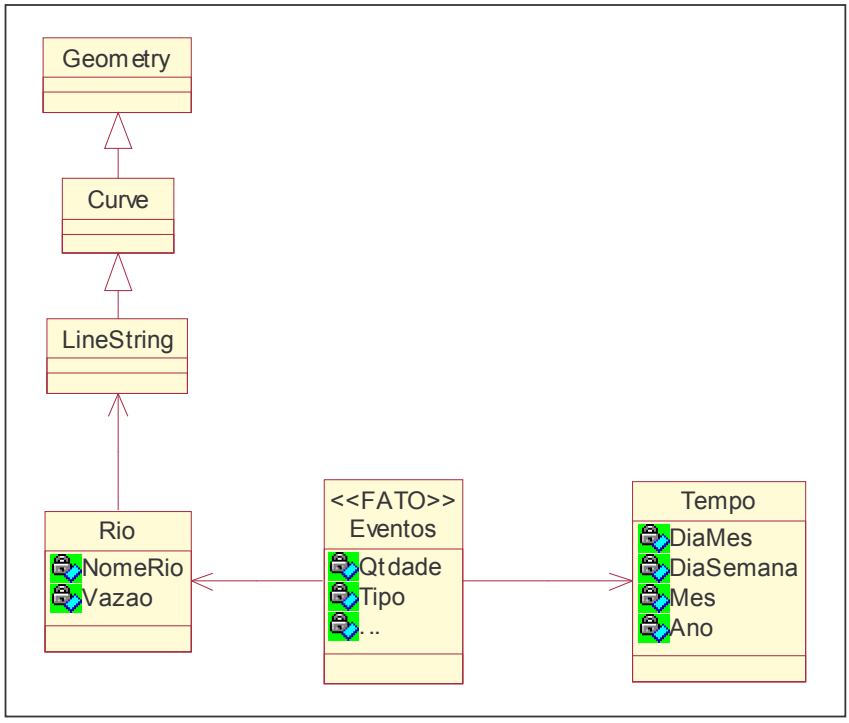


Fig.5 Uso de hierarquia extensa em uma das dimensões – "Rio"

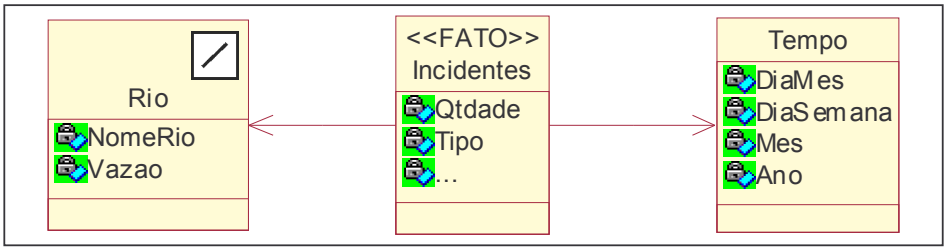


Fig.6 Uso de estereótipo para representar a associação entre a classe Rio e a classe LineString, pela qual pode ser representado

Desta forma, poderíamos substituir a dimensão Rio, composta por uma associação com uma extensa hierarquia de classes (Figura 5), por uma única classe, utilizando um estereótipo significativo da classe (Linha – "LineString") com a qual Rio possui um relacionamento e pela qual pode ser representado (Figura 6). Tal prática conduz a diagramas mais limpos, mantendo a mesma riqueza semântica e possibilitando a coexistência simplificada de conceitos pertencentes a domínios diferentes, como por exemplo, conceitos analíticos e geográficos, temas desta dissertação.

3. EXTENSÕES GEOGRÁFICAS NA MODELAGEM DE DATA WAREHOUSES

3.1 O CONSÓRCIO OPENGIS

O Consórcio Open GIS (OGC – "Open GIS Consortium") pode ser definido como uma organização internacional que está definindo uma especificação que permita a interoperabilidade no processamento de dados geográficos.

O OpenGIS (Open Geodata Interoperability Specification) é uma especificação detalhada de uma arquitetura para acesso distribuído a dados geo-referenciados e a recursos de geoprocessamento. O OpenGIS fornece uma plataforma de trabalho, sobre a qual os desenvolvedores podem implementar software que permita aos seus utilizadores acessar e processar dados geográficos provenientes de várias fontes, através de uma interface de computação genérica, para uma base de informação tecnológica aberta.

Segundo (PROJETO CIMA, 2003) o OpenGIS é uma especificação abstrata, uma vez que, apesar de incluir uma descrição extremamente detalhada do que deve ser implementado, não especifica como é que o software deve ser implementado relativamente a um sistema de computação distribuída particular (OLE/COM/DCOM, CORBA, DCE, Java). Tal especificação consiste em três partes:

-Modelo Aberto de Dados Geo-referenciados: Um modelo para a representação da Terra e de fenômenos terrestres do ponto de vista matemático e conceitual.

Esta parte da especificação define um conjunto geral e comum de tipos de informação geográfica que podem ser usados para modelar as necessidades dos dados geo-referenciados de domínios de aplicação mais específicos, usando métodos de programação convencionais e/ou orientados a objetos.

-Modelo de Serviços OpenGIS: Um modelo para a especificação dos serviços que permitem acessar, gerir, manipular, representar e partilhar os dados geo-referenciados entre diferentes sistemas de informação.

-Modelo de Comunidades de Informação: Uma estrutura que usa o modelo aberto de dados geo-referenciados e o modelo de serviços OpenGIS para resolver não só os problemas técnicos como também os problemas institucionais, econômicos e legais de não-interoperabilidade. Mais precisamente, este modelo integra os dois anteriores num esquema que estabelece:

· Uma forma de uma comunidade de produtores de dados geo-referenciados e de utilizadores que já partilham um conjunto comum de definições de características geográficas, poder manter eficiente e efetivamente estas definições, catalogar e partilhar conjuntos de dados de acordo com as mesmas.

· Uma forma eficiente e precisa para que diferentes comunidades de usuários e produtores de dados geográficos os partilhem, apesar de possuírem conjuntos de características geográficas distintas. Um exemplo, é o caso de engenheiros civis, geólogos e agrônomos desejarem partilhar dados sobre os tipos de solos apesar de os classificarem de um modo diferente, de acordo com os seus objetivos profissionais. O Modelo das Comunidades de Informação define um esquema para tradução automática entre diferentes léxicos de características geográficas.

Para fins do presente trabalho, o foco estará na especificação do modelo de Serviços do OpenGIS, mais particularmente no aspecto da obtenção de um serviço, que significa fazer com que as aplicações que geram, distribuem e processam dados geo-referenciados possam interagir. O OpenGIS define: como fornecer um serviço; como pedir um serviço; e como determinar se se trata de um pedido de dados, de um pedido de operação sobre os dados, ou ambos. Define também um conjunto normalizado de tipos de dados e operações sobre eles, constituindo assim uma plataforma comum de interoperabilidade entre clientes e servidores de dados geográficos. O OpenGIS fornece ainda um conjunto de serviços que facilita a partilha de dados entre diferentes comunidades de informação (conjuntos de utilizadores com

diferentes significados, semântica e sintaxe para os dados geo-referenciados e para o processamento espacial).

Estas capacidades dependem da definição de um modelo comum para transferência de informação geográfica, e da definição do comportamento das operações sobre esses dados. O OpenGIS só não vai mais longe na definição dos métodos de armazenamento e processamento de dados para preservar o vasto investimento em informação geográfica e sistemas de informação geográfica já existentes, e para assegurar a oportunidade de introduzir novos métodos para gestão e manipulação de informação geográfica.

Com vistas a estabelecer algumas especificações abertas em termos de algumas das plataformas comerciais, as seguintes foram definidas: CORBA, OLE/COM/DCOM e SQL. Como o presente documento está focado na definição dos estereótipos geográficos para emprego em Sistemas de Gerenciamento de Bancos de dados Relacionais com extensões espaciais, interessa-nos diretamente a especificação de Feições simples do OpenGIS para a SQL.

Para evitar confusões quanto ao emprego dos termos GEOMÉTRICO e GEOGRÁFICO ao longo da presente dissertação, o primeiro será utilizado para referenciar os tipos de dados definidos pelo OpenGIS, enquanto o segundo será empregado na citação das dimensões que possuam características espaciais.

3.2 ESPECIFICAÇÃO OPENGIS DE FEIÇÕES SIMPLES PARA A SQL

O propósito desta especificação é definir um esquema padrão SQL que suporte o armazenamento, consulta e atualização de coleções de simples feições geo-espaciais usando a API do ODBC. A especificação abstrata do OpenGIS define uma feição simples como possuindo simultaneamente atributos espaciais e não-espaciais. Atributos espaciais simples têm geometria 2D com interpolação linear entre os vértices.

Tais coleções de feições simples são conceitualmente armazenadas como tabelas com colunas de tipo geométrico em um SGBD Relacional, onde cada característica é armazenada como uma linha de uma tabela. Para as feições, cada atributo não-espacial é mapeado nos

tipos de dados normalizados do ODBC/SQL92, e cada atributo espacial é mapeado num valor geométrico.

Uma tabela cujas linhas representem feições OpenGIS denomina-se tabela de feições. Na especificação OpenGIS, descrevem-se implementações de tabelas de feições para dois possíveis ambientes SQL: o SQL92 (SQL92 padrão) e o SQL92 com tipos geométricos (ou seja, para SGBDs Objeto-Relacional, onde seja possível o emprego de tipos de dados definidos pelo usuário). Ambas implementações estendem o esquema de informações do SQL92 de forma a prover consultas padrão sobre os metadados que retornem:

- a lista de tabelas de feições em uma base de dados;
- a lista de colunas geométricas para qualquer tabela de feição da base de dados;
- o sistema de referência espacial para cada coluna geométrica da base de dados.

Assim, a especificação do OpenGIS para a SQL busca padronizar:

- os nomes e definições dos tipos de dados geométricos OpenGIS;
- os nomes, definições e assinaturas para as funções geométricas OpenGIS.

3.3 TIPOS DE DADOS GEOMÉTRICOS PARA A SQL

Os tipos de dados geométricos para SQL são organizados em uma hierarquia de tipos baseada no modelo de Objetos Geométricos do OpenGIS, conforme a figura 7:

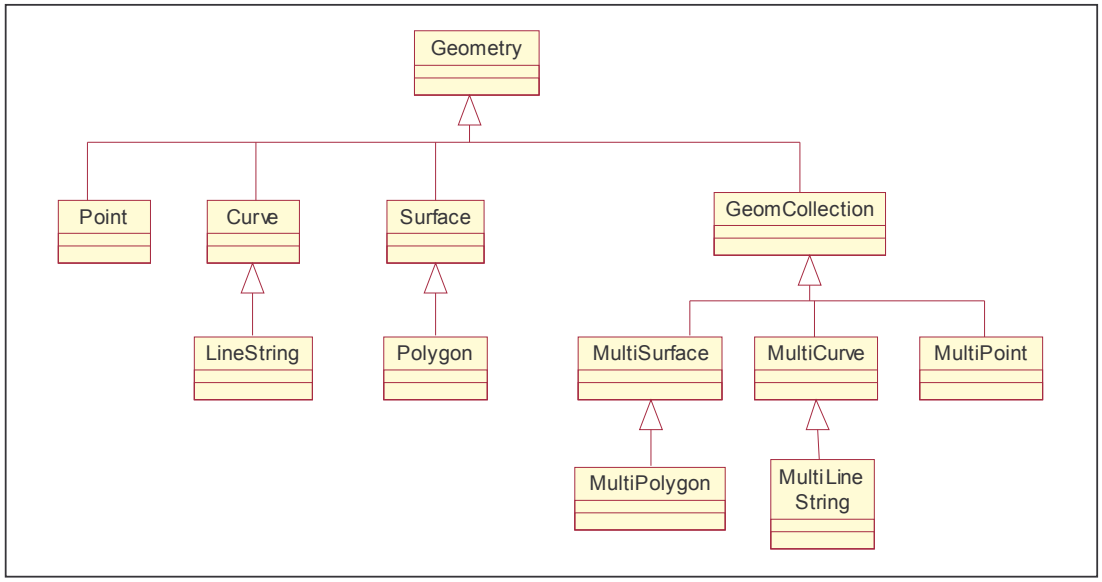


Fig.7 Hierarquia de Tipos Geométricos SQL definidos pelo OpenGIS

O tipo raiz, denominado Geometry (Geométrico) possui subtipos para Point (Ponto), Curve (Curva), Area (Área) e GeometryCollection (Coleção Geométrica). GeometryCollection é uma coleção de objetos geométricos possivelmente heterogêneos. MultiPoint, MultiCurve e MultiSurface são subtipos de GeometryCollection usados para manipular coleções homogêneas de Pontos, Curvas e Superfícies, respectivamente. Os tipos geométricos com zero dimensões são Point (Ponto) e MultiPoint (MultiPonto). Os tipos geométricos de uma dimensão são Curve (Curva) e MultiCurve (MultiCurva), juntamente com suas subclasses. Os tipos geométricos de duas dimensões são Surface (Superfície) e MultiSurface (MultiSuperfície) juntamente com suas subclasses.

Existem funções SQL definidas para a construção de instâncias dos tipos de dados anteriores dadas as representações binárias ou textuais dos tipos.

Os tipos Geometry, Curve, Surface, MultiCurve e MultiSurface são abstratos, ou seja, não-instanciáveis; os demais tipos são instanciáveis e podem ser implementados de forma mais ou menos completa de acordo com a tabela a seguir:

Nível	Tipos Disponíveis	Tipos Instanciáveis
1	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection	Point, LineString, Polygon, GeomCollection
2	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection, MultiPoint, MultiCurve, MultiLineString, MultiSurface, MultiPolygon	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon
3	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection, MultiPoint, MultiCurve, MultiLineString, MultiSurface, MultiPolygon	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeomCollection

Tab. 1 - Possíveis implementações para os tipos de dados geométricos

O ponto representa uma localização única no espaço. A curva é um objeto unidimensional representado por uma seqüência de pontos, enquanto o LineString é uma curva com interpolação linear entre seus pontos. O Polygon (polígono) é uma superfície plana, definida por um limite exterior e um ou mais limites interiores. Cada limite interior define um buraco no polígono (figura 8).

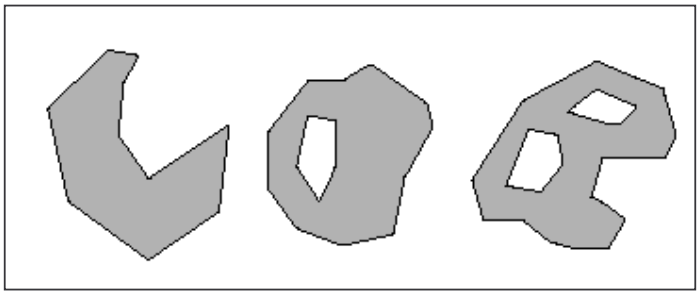


Fig.8 Exemplos de Polígonos

A figura 8 representa exemplos de polígonos. No primeiro exemplo o polígono só possui o limite externo, no segundo existe o limite externo e um limite interno (com um único buraco), e no terceiro exemplo, além do limite externo, existem dois limites internos (dois buracos).

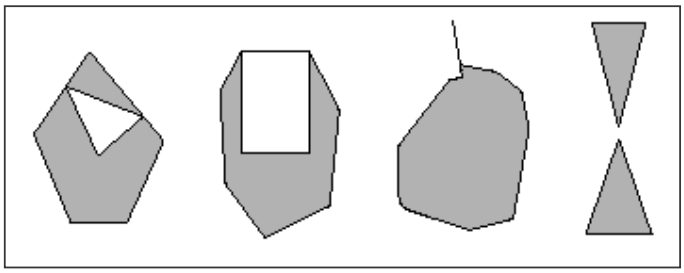


Fig.9 Exemplos de não-polígonos

A figura 9 exibe figuras que não são polígonos simples. O primeiro e o quarto exemplos representam na verdade dois polígonos. O segundo exemplo não consiste de um polígono, uma vez que existe uma ligação externa ao polígono entre os extremos. E, no terceiro exemplo, existem pontos de um limite externo do polígono que não estão presentes no polígono.

O Multipoint é uma coleção geométrica zero-dimensional, restrito à composição por pontos (Point) que não são conectados ou ordenados. O MultiCurve (MultiCurva) é uma coleção unidimensional de curvas. O MultiLineString é uma MultiCurve formada por vários LineString. O MultiSurface (MultiSuperfície) é uma coleção geométrica bidimensional de superfícies. Os interiores de quaisquer duas superfícies em uma coleção MultiSurface podem possuir interseção. O MultiPolygon (MultiPolígono) é um MultiSurface formado por Polygons (polígonos).

Os tipos de dados definidos pelo OpenGIS podem ser representados em um SGBD das seguintes formas, de acordo com o tipo de SGBD:

1) SGBD SQL92 (ou seja, um Banco de dados relacional que suporte SQL 92, mas sem suporte a extensões de objeto, isto é, no qual não é possível a criação de tipos de dados definidos pelo usuário):

-usando tipos numéricos SQL

- usando tipos binários SQL (WKB – Well-Known Binary)

2) SGBD SQL92 com tipos geométricos (ou seja, um SGBD Objeto Relacional, onde a criação de tipos de dados definidos pelo usuário seja possível):

-usando o formato WKB (Well-Known Binary, ou binário bem-conhecido)

-usando o formato WKT (Well-Known Texto, ou texto bem-conhecido)

Em função do SGBD utilizado neste trabalho, PostGreSQL, ser um SGBD Objeto-Relacional (suportando, por isso, tipos abstratos de dados ou tipos de dados definidos pelo usuário), a documentação do OpenGIS de interesse para o presente trabalho é a que se refere a um SGBD SQL92 com tipos geométricos, a qual, em verdade define as funcionalidades providas pelo PostGIS, que é a extensão espacial ao PostGreSql.

Exemplos de representações textuais (WKT) são apresentadas na Tabela 2, a seguir. Embora as coordenadas estejam sendo apresentadas como valores inteiros, poderiam ser qualquer valor de dupla precisão.

Tipo Geométrico	Representação Literal SQL Texto	Observação
Point	'POINT (10 10)'	um Ponto
LineString	'LINESTRING (10 10, 20 20, 30 40)'	um LineString com três Pontos
Polygon	'POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))'	um Polígono com 1 anel exterior e sem anéis interiores
Polygon	'POLYGON ((10 10, 20 10, 25 15, 35 15, 10 30, 10 10), (15 15, 18 15, 20 25, 15 25, 15 15))	um Polígono com 1 anel exterior e um anel interior
Multipoint	'MULTIPOINT (10 10, 20 20)'	um MultiPonto com dois Pontos
MultiLineString	'MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))'	um MultiLineString com 2 LineStrings
MultiPolygon	'MULTIPOLYGON (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))'	um MultiPoligono com dois Poligonos
GeomCollection	'GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))'	uma coleção geométrica composta de 2 Pontos e 1 LineString

Tab. 2 - Representações (WKT) de dados geométricos

3.4 FUNÇÕES GEOMÉTRICAS DEFINIDAS PELO OPENGIS

O OpenGIS define, além dos tipos de dados, uma série de funções geométricas sobre os dados. Algumas são definidas sobre a superclasse Geometry (e assim, são também válidas para todas as suas subclasses), como por exemplo: Equals (umGeometry, outroGeometry), que retorna True ou False caso dois objetos Geometry sejam, respectivamente, iguais ou diferentes; ou GeometryType( ), que retorna uma string com o nome da subclasse à qual pertence o objeto em questão; ou ainda Distance(umGeometry, outroGeometry), que retorna a distância entre os dois objetos Geométricos passados como argumento para a função. São

também definidas algumas funções específicas para as subclasses de Geometry, como por exemplo NumPoints( ), definido para a classe LineString e que retorna a quantidade de pontos contidos no objeto em questão.

A Tabela 3 exibe as funções geométricas definidas pelo OpenGIS julgadas mais relevantes para o contexto desta dissertação.

Função	Definição
AddGeometryColumn (varchar, varchar, varchar, integer, varchar, integer)	Adiciona uma coluna geométrica a uma tabela já existente.
Dimension (geometry)	Retorna a dimensão de um objeto.
IsEmpty (geometry)	Retorna se determinada geometria está ou não vazia.
Equals (geometry)	Compara dois objetos geométricos.
Disjoint (geometry, geometry)	Retorna se dois objetos geométricos estão disjuntos.
Touches(geometry,geometry)	Retorna se um objeto geométrico toca um outro objeto geográfico.
Crosses(geometry,geometry)	Verifica se um objeto geométrico cruza um outro objeto geográfico.
Within(geometry,geometry)	Verifica se um objeto geométrico está dentro de outro objeto geográfico.
Overlaps(geometry,geometry)	Retorna se um objeto geométrico sobrepõe outro objeto geográfico.
Contains(geometry,geometry)	Retorna se um objeto geométrico contém outro objeto geográfico.
Intersects(geometry,geometry)	Retorna se um objeto geométrico intercepta outro objeto geográfico.
Intersection(geometry,geometry)	Retorna uma figura geométrica que representa o conjunto de pontos de uma figura geométrica em relação a outra figura geométrica.
GeomUnion(geometry,geometry)	Retorna uma figura geométrica que representa o conjunto união de pontos entre dois objetos geométricos. Possui duas variações: GeomUnion(geometry,set) e memGeomUnion(geometry, geometry).
Difference(geometry,geometry)	Retorna um conjunto de pontos simetricamente diferentes entre dois objetos geométricos.
X(geometry)	Exibe a coordenada X do primeiro ponto de um objeto geométrico.
Y(geometry)	Exibe a coordenada Y do primeiro ponto de um objeto geométrico.
Z(geometry)	Exibe a coordenada Z do primeiro ponto de um objeto geométrico.
NumPoints(geometry)	Retorna o número de pontos existentes na primeira linha de um objeto geométrico.
PointN(geometry,integer)	Retorna o enésimo ponto da primeira linha de um objeto geométrico.
ExteriorRing(geometry)	Retorna o anel exterior do primeiro polígono de um objeto geométrico.
NumInteriorRings(geometry)	Retorna o número de anéis interiores de um objeto geométrico.
InteriorRingN(geometry,integer)	Retorna o enésimo anel interior do primeiro polígono de um objeto geométrico.
IsClosed(geometry)	Verifica se uma forma geográfica inicia e termina no mesmo ponto.
IsRing(geometry)	Verifica se uma dada curva é fechada ou não.
GeometryN(geometry,int)	Retorna o enésimo objeto geométrico de uma coleção geométrica.
Distance(geometry,geometry)	Retorna a distância cartesiana entre dois objetos geométricos.
AsText(geometry)	Retorna a definição em 'Well-Known Text" de um dado objeto geométrico.
GeometryFromText(varchar, integer)	Converte um objeto definido em "Well-Known Text" em um objeto geométrico.
Centroid(geometry)	Retorna o ponto centróide de uma forma geométrica.

Tab. 3 - Principais funções do PostGIS provenientes do OpenGIS

3.5 EMPREGO DE ESTEREÓTIPOS OPENGIS NA MODELAGEM UML DE DATA WAREHOUSES

Conforme citado no capítulo 2 do presente trabalho, uma das grandes vantagens do uso de UML para a modelagem de data warehouses é a possibilidade do emprego de estereótipos, que substituem a representação de associações com extensas hierarquias de objetos, sem perdas semânticas, permitindo a criação de diagramas ricos em conceitos distintos e pouco "carregados" visualmente.

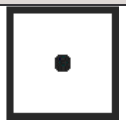
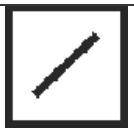
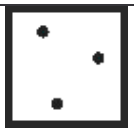
Com o propósito de modelar data warehouses onde conceitos dimensionais e espaciais possam coexistir harmonicamente em um mesmo diagrama, empregaremos estereótipos geográficos para representar as classes geográficas e a ausência de estereótipos para a representação de classes convencionais ou não-geográficas (como por exemplo a classe FATO).

O uso de estereótipos geográficos foi baseado em uma abordagem similar utilizada no GeoFrame. O GeoFrame é um framework conceitual, orientado a objetos e utilizando a notação UML, que provê um conjunto de classes básico (classes abstratas) para auxiliar projetistas na modelagem de dados geográficos (Lisboa, 1999). Em 2001, a hierarquia de classes inicial do GeoFrame foi modificada com vistas à produção de um modelo mais conciso e com menor número de elementos visuais, dando origem à versão 2.0 do GeoFrame (Rocha, 2001).

A abordagem do presente trabalho difere do GeoFrame inicialmente pela finalidade, pois enquanto aquele foi concebido para a modelagem de sistemas OLTP, este possui o foco nos sistemas OLAP; uma segunda diferença está nos tipos e quantidade de estereótipos geográficos: o GeoFrame utiliza dez tipos (Ponto, Linha, Polígono, Complexo, TIN, Grade de Células, Pontos Irregulares, Grade de Pontos, Polígonos Adjacentes e Isolinhas), ao passo que na proposta desta dissertação são empregados sete tipos (Ponto, Linha, Polígono, MultiPonto, MultiLinha, MultiPolígono, Coleção Geométrica), conforme Tabela 4.

Um dos grandes objetivos do presente trabalho é propor uma técnica de modelagem que integre os conceitos espaciais e dimensionais em um único diagrama conceitual e que seja

de simples, praticamente direto mapeamento para o modelo lógico dimensional relacional. Para atender tal requisito de simplicidade ou imediatismo no mapeamento, os tipos de dados utilizados como estereótipos geográficos são aqueles definidos pelo OpenGIS para a SQL. Ressalte-se ainda que, por ter sido baseado na abordagem do GeoFrame, embora com quantidade e nomes de estereótipos diferentes (pelas razões já expostas anteriormente), este trabalho buscou utilizar as mesmas figuras para a representação dos estereótipos geográficos.

Tipo de Dado	Estereótipo
Point (Ponto)	
LineString (Linha)	
Polygon (Polígono)	
MultiPoint (MultiPonto)	
MultiLineString (MultiLinha)	
MultiPolygon (MultiPolígono)	
GeometryCollection (Coleção Geométrica)	

Tab. 4 - Estereótipos representativos dos tipos de dados do OpenGIS

O uso dos tipos de dados definidos para a SQL como estereótipos, apresenta as seguintes vantagens: tais tipos de dados (para a SQL) são um subconjunto do modelo de objetos geométricos do OpenGIS bastante representativo do universo geográfico (ou seja, não são apenas tipos de dados tecnológicos definidos para o nível de implementação, uma vez que

com tais tipos, pode-se modelar e representar qualquer objeto geográfico) e possuem mapeamento direto do modelo conceitual para o lógico e deste para o físico (isto é, o mesmo tipo de dado definido no modelo conceitual será utilizado no modelo lógico e também na implementação), sem qualquer necessidade de transformação ou conversão de conceitos.

Uma vez que o PostGIS (extensão espacial ao SGBD) implementa as definições do OpenGIS para a "SQL92 com tipos geométricos", sendo, em sua versão 0.8 totalmente compatível com o citado documento de especificação do OGC, os seguintes tipos de dados estão disponíveis: Point (Ponto), LineString (Linha), Polygon (Polígono), MultiPoint (MultiPonto), MultiLineString (MultiLinha), MultiPolygon (MultiPolígono) e GeometryCollection (Coleção Geométrica), ou seja, conforme o nível 3 da tabela de possíveis implementações para tipos de dados geométricos.

Desta forma, na modelagem de data warehouses, utiliza-se nenhum estereótipo para as classes convencionais, ou seja, aquelas sem nenhum tipo de atributo geográfico (ex.: Classe TEMPO, classe FATO). Para as classes geográficas, ou seja, aquelas onde um ou mais atributos são do tipo geográfico, utiliza-se um dos estereótipos definidos na Tabela 3, de forma a representar seu subtipo de objeto geométrico. As classes geográficas serão, no modelo lógico, mapeadas para tabelas de feições conforme explicação detalhada do próximo capítulo deste trabalho.

Na figura 10, um simples exemplo de um data warehouse destinado ao registro histórico de incidentes com veículos de determinada organização é apresentado. No diagrama em questão, as classes Veículo, Tempo e Incidente, por não possuírem atributos espaciais, são representadas sem estereótipos (o que significa serem objetos convencionais) e as classes Local (trecho de logradouro onde ocorreu o incidente), Bairro e Logradouro são geográficas e, portanto, utilizam o estereótipo representativo de sua associação com uma classe geométrica (o Logradouro "é representado por" uma Linha, o Bairro "é representado por" um Polígono, etc).

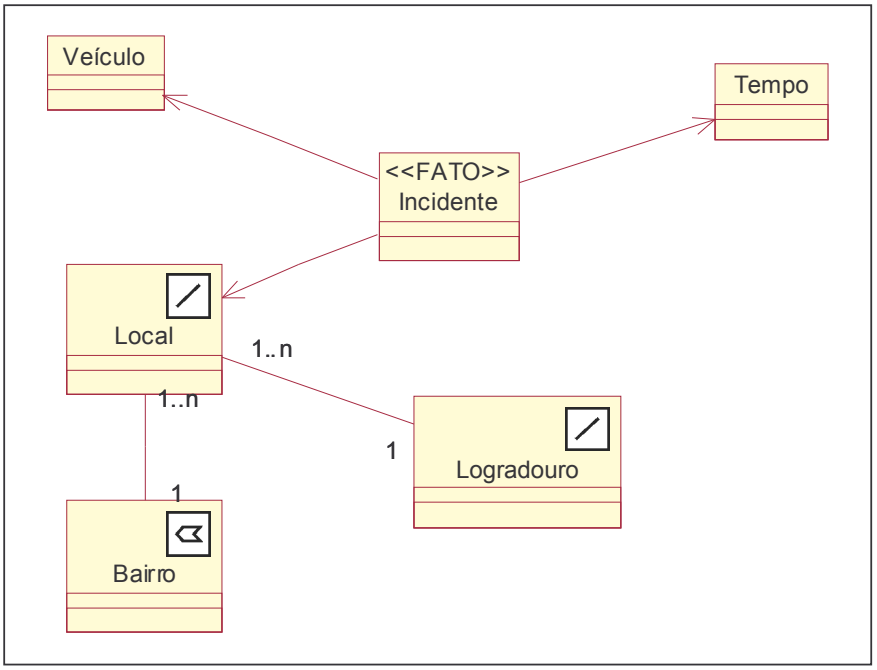


Fig.10 Data Warehouse de Incidentes com veículos – Uso de estereótipos

3.6 O SOFTWARE PLANETGIS

O PlanetGIS é um software para SIG (Sistema de Informações Geográficos), freeware, desenvolvido por uma empresa canadense de mesmo nome, cujas vantagens são: gratuidade, facilidade no uso, velocidade no processamento gráfico e capacidade de manipulação de grandes bases de dados.

O PlanetGIS traz ainda como características:

- Suporte aos principais formatos de arquivo para GIS (SHP, MIF, DXF, DGN, FEA, SDL, E00).
- Aquisição e edição de todos os tipos de entidades geográficas, inclusive no formato raster.

- Produção de mapas com grande potencial estético.
- Criação de mapas temáticos de acordo com os atributos desejados.
- Edição e visualização de atributos vinculados com um banco de dados.
- Capacidade de vinculação das entidades geográficas a tabelas de diversos tipos de SGBD, possibilitando exibir características completas de cada entidade.

Percebe-se diante dessas características, que o PlanetGIS é uma ferramenta SIG bastante versátil. Contudo, por ser freeware e em função de sua empresa desenvolvedora subsistir do suporte e customização do produto, o software não possui instruções detalhadas disponíveis publicamente para sua utilização. O software está disponível como um programa executável (plataforma Windows) ou como um componente COM (arquivo.ocx) a ser integrado em novas aplicações (a exemplo do PostGeoOlap, tema desta dissertação).

Por todas as vantagens acima expostas, o software PlanetGIS (em sua versão componente COM) foi escolhido para a função de componente de visualização da ferramenta PostGeoOlap, tema desta dissertação e que será detalhada no capítulo 5. Outras ferramentas para SIG foram analisadas, como por exemplo: ArcExplorer (ESRI, 2003), MapGuide (AUTODESK, 2002), Spring (INPE, 2003), SpringWeb (INPE, 2003) e MapServer (MAPSERVER, 2003); em todas elas, o fato de não serem um componente (objeto COM), ou trabalharem apenas com o formato proprietário de sua empresa desenvolvedora ou possuírem algum custo de licenciamento tornou-as inapropriadas para o emprego como componente de visualização da ferramenta tema deste trabalho.

Apesar de toda a capacidade da ferramenta PlanetGIS para Sistemas de Informação Geográficos, é importante citar que, neste trabalho, seu emprego restringiu-se a um visualizador dos resultados geográficos retornados por um SGBD espacial; ou seja, nenhum processamento geográfico foi requerido do citado componente.

4. MAPEAMENTO DO MODELO CONCEITUAL DIMENSIONAL PARA O MODELO LÓGICO DIMENSIONAL RELACIONAL

4.1 INTRODUÇÃO

Os sistemas transacionais OLTP visam atender os usuários operacionais das organizações e caracterizam-se por sua grande quantidade de inclusões e alterações de dados e normalmente baixa freqüência de consultas. Em tais sistemas, como a característica transacional é imperiosa, é comum o esvaziamento dos históricos dos dados em favor de uma base de dados atualizada e conseqüentemente de melhor desempenho. Em contraposição, os data warehouses (INMON, 1997) são armazéns de dados orientados a assunto, integrados, não voláteis e variantes no tempo, cuja principal finalidade é atender por meio de consultas a um público-alvo composto de diretores e gerentes.

Assim, o sistema dimensional caracteriza-se por ser do tipo "carregue e use", com cargas de dados em batch (dados em lote) e voltado exclusivamente para as consultas gerenciais. Tais requisitos tornam os data warehouses repositórios imensos em termos de quantidade de dados armazenados (uma vez que nenhum dado é descartado) e criam uma grande pressão por desempenho na execução das requisições de consultas a esta base dimensional. Esta é a razão de perseguirmos, durante o projeto lógico de data warehouses para uso em bancos de dados relacionais, uma efetiva "desnormalização" dentro de cada uma

das dimensões de forma a minimizar a necessidade de junções entre tabelas que depreciariam o desempenho do sistema.

Pelas razões acima expostas, percebem-se várias diferenças de propósito, usuários, volume de informações e forma de utilização entre os sistemas transacionais e os dimensionais que justificam regras diferenciadas de mapeamento do modelo conceitual OO para o lógico Relacional. Nos sistemas transacionais, a redundância de informações é combatida todo o tempo em busca da otimização do espaço de armazenamento e da consistência dos dados, levando a esquemas normalizados; em contraposição, sistemas dimensionais não buscam economia de espaço de armazenamento: objetivam o armazenamento de todos os dados relevantes à organização e, para terem a capacidade de manipular tal massa de dados em tempo aceitável, abrem mão da normalização.

Se a questão do desempenho, traduzida na "desnormalização" das tabelas que compõem as dimensões em um modelo estrela implementado com a tecnologia relacional, já seria uma boa justificativa para a adoção de uma abordagem diferenciada no mapeamento de um modelo dimensional conceitual OO para um modelo dimensional lógico relacional, acrescentemos o caso dos esquemas genéricos dimensionais, que sofrerão extensões para adaptarem-se à novas realidades (adição de informações na forma de novas classes ou atributos) e que já possuem sua associação com a classe Fato (e, conseqüentemente, sua granularidade) pré-definida. Tal mapeamento é, em alguns casos, inviável se implementado pelas técnicas tradicionais.

O propósito deste capítulo é discutir o mapeamento dimensional (nome que damos à técnica de desnormalizar as classes integrantes de uma dimensão – e que já havia sido pregada por Kimball) em substituição ao mapeamento tradicional, para a transformação de modelos conceituais dimensionais OO em modelos lógicos dimensionais relacionais. Para tanto, mostraremos o impacto, em termos de mapeamento de modelo dimensional conceitual OO para lógico relacional, de extensões feitas a modelos OO genéricos e, para os casos em que o método tradicional de transformação OO para relacional seja ineficiente, apresentaremos as vantagens da abordagem dimensional.

O uso do mapeamento aqui proposto visa permitir aos projetistas de sistemas dimensionais usar toda a riqueza semântica de representação da UML para fazer a modelagem Estrela de seus data warehouses, empregando as mesmas classes de negócio presentes em outros modelos da organização, sem se preocupar com os detalhes lógicos ou físicos de implementação em um Banco de dados relacional. Desta forma, sistemas dimensionais podem

ser concebidos com maior riqueza de representação nas classes dimensão, permitindo que esquemas conceituais definidos para uma família de negócios similares sejam reutilizados através de extensões de forma a contemplar especificidades locais de informações. A preocupação com detalhes lógicos é delegada à fase de mapeamento (implementada pelo mapeamento dimensional proposto neste documento).

4.2 REGRAS PARA MAPEAMENTO DO DIAGRAMA CONCEITUAL OO PARA O LÓGICO DIMENSIONAL RELACIONAL

A abordagem dimensional (COLONESE; TANAKA; CARVALHO, 2003) consiste simplesmente no mapeamento de todas as classes com relacionamento de cardinalidade 1..n e presentes em uma mesma dimensão para uma única tabela (que conteria os atributos de todas as classes) representativa desta dimensão e relacionada à tabela Fato (que também foi gerada a partir da classe de mesmo nome) por meio de uma chave estrangeira. A principal diferença da abordagem tradicional para a proposta deste trabalho é que, nesta, buscamos "ferir" as regras normais em busca de uma única tabela (relação) em cada uma das dimensões, ainda que estas sejam compostas por várias classes com relacionamentos 1..n em seu projeto conceitual.

O mapeamento dimensional visa, em verdade, adiar para a fase de projeto lógico, os ditames de Kimball para seu modelo estrela no que dizia respeito à "desnormalização" das relações.

Nos casos a seguir exemplificaremos o emprego do mapeamento dimensional para a transformação de um modelo conceitual dimensional OO em lógico relacional, e para tanto usaremos o mesmo exemplo da distribuidora citado na Figura 2, mas por simplicidade de representação estaremos trabalhando com apenas uma das dimensões e com a classe Fato.

4.2.1 Dimensão representada por uma única classe associada à classe Fato

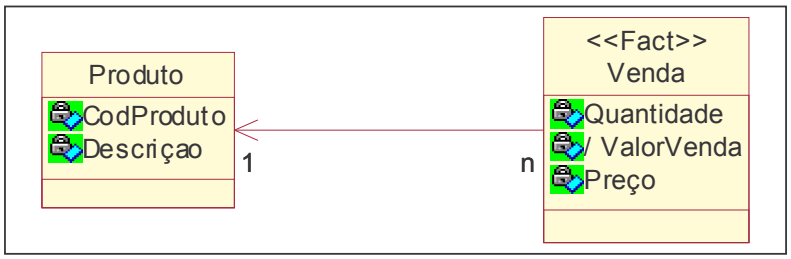


Fig.11 Associação simples entre classes – Modelo Conceitual OO

Neste exemplo, existem poucas diferenças entre o modelo lógico proposto por Kimball e um modelo conceitual (diferenças: a notação – que está em UML, e a ausência da chave estrangeira "CodProduto" em VENDA que só surgiria no momento de se transformar o modelo conceitual em lógico relacional), já que temos como representante de uma dimensão uma única classe. No diagrama acima, o mapeamento do modelo conceitual para o relacional acontecerá da mesma forma quer apliquemos o método tradicional ou o dimensional (apresentado neste trabalho), ou seja: cada classe é transformada em uma tabela e o relacionamento entre as classes é expresso por meio da chave estrangeira (CodProduto) presente na tabela Fato, o que permite a realização de agregações e ordenações baseadas nesta chave.

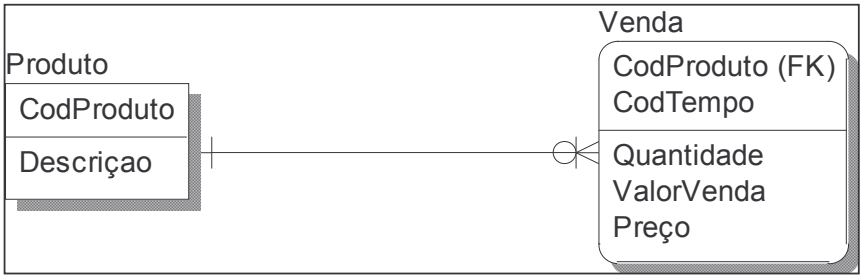


Fig.12 Mapeamento do modelo conceitual para o modelo lógico Relacional. A análise de vendas segundo a perspectiva de produto é possível devido à presença da chave estrangeira CodProduto na tabela Fato (Venda)

Partiremos da versão acima, de uma classe Fato (Venda) e uma outra de dimensão (Produto), para efetuar adições em "Produto" (extensões, de maneira a tornar a classe mais adequada a uma realidade específica) e mostrar como o mapeamento de extensões às dimensões, pelas regras tradicionais, de um modelo conceitual para o modelo lógico, gera um esquema não condizente com os requisitos de performance e simplicidade apregoados por

Kimball. Em seguida estaremos também apresentando, para os casos onde o mapeamento tradicional se mostrou ineficiente, as vantagens do uso do mapeamento dimensional.

4.2.2 Dimensão representada por duas classes com relacionamento entre si do tipo "Agregação" (Compartilhada ou Composta) ou Associação simples, onde a classe diretamente ligada à classe Fato está do lado muitos (n) do relacionamento

Neste caso, a dimensão Produto sofreu extensões através da adição de uma associação (do tipo Agregação Compartilhada) com a classe Categoria, denotando que cada Produto, neste sistema em especial, possui uma categoria.

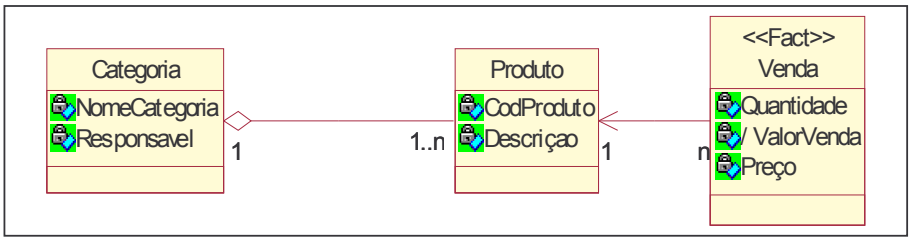


Fig.13 Modelo Conceitual dimensional OO onde a dimensão "Produto" foi adicionada da característica de Categoria (classe Categoria)

O mapeamento (do modelo Conceitual OO para o lógico relacional) de associações do tipo simples ou Agregação (seja ela compartilhada ou composta) entre classes de uma dimensão, onde o lado "muitos" (n) do relacionamento (no exemplo, a classe Produto) é quem está diretamente associado à classe Fato (Venda) acontece como se segue:

4.2.2.1 Segundo o mapeamento tradicional de OO para Relacional

Pelo mapeamento tradicional, cada classe selecionada como persistente (como é o caso de todas as presentes em um data warehouse) será mapeada para uma tabela (para isso deve-se escolher uma atributo identificador de cada classe para chave primária) e o relacionamento entre as classes se dará por meio da existência da chave estrangeira (que é a migração da

chave primária da tabela do lado "1" para o lado "muitos" do relacionamento). Desta forma, o modelo conceitual dimensional OO possuiria a seguinte representação lógica relacional:

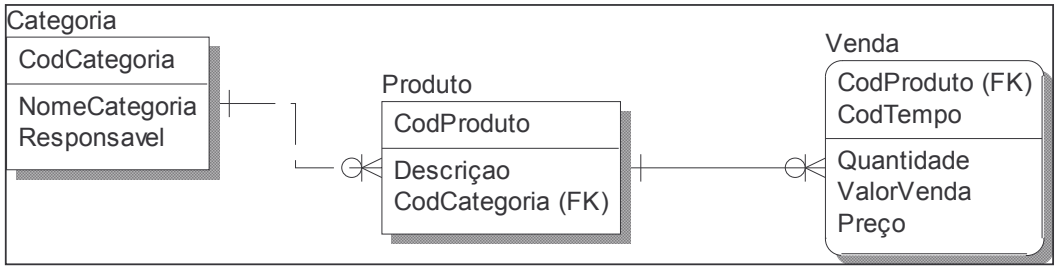


Fig.14 Modelo lógico relacional para o sistema com a extensão de "Categoria" ao modelo

Interessante ainda ressaltar que sumarizações e ordenações podem ser feitas por categoria (que foi a classe adicionada), já que a chave de Categoria (CodCategoria) está presente na tabela Produto, como na seguinte Consulta: "Select Sum(ValorVenda) from Venda, Produto, Categoria Where Venda.CodProduto = Produto.CodProduto and Produto.CodCategoria = Categoria.CodCategoria and NomeCategoria = 'Masculinas'", onde pode-se obter a soma das vendas de todos os produtos pertencentes à categoria 'MASCULINAS'.

4.2.2.2 Segundo o mapeamento dimensional proposto

Na abordagem dimensional de mapeamento, as classes serão mapeadas para uma única tabela e todos os atributos deverão migrar para tal tabela, de forma a produzir o seguinte esquema:

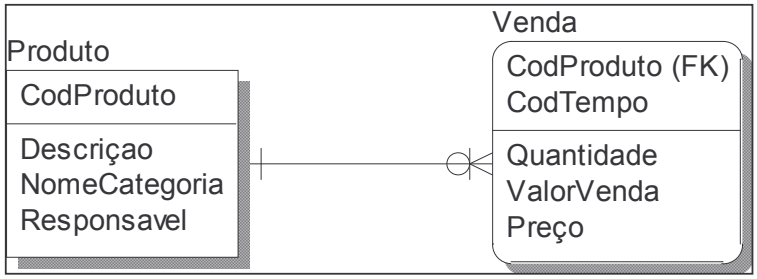


Fig.15 Modelo lógico dimensional relacional (mapeamento dimensional)

A vantagem da presente abordagem é a melhoria de desempenho na execução das consultas em função da eliminação do número de junções e a conseqüente desvantagem é o aumento do espaço de armazenamento, que, conforme (KIMBALL, 1996), possui alta relação custo-benefício (o custo da unidade de armazenamento é baixo em vista do correspondente aumento em desempenho do sistema).

No esquema dimensional, a mesma sumarização por categoria do exemplo anterior seria reescrita da seguinte maneira: "Select Sum(ValorVenda) From Venda, Produto Where Venda. CodProduto = Produto.CodProduto and NomeCategoria = 'Masculinas'". Percebe-se que esta consulta possui uma junção a menos que sua correspondente obtida pelo mapeamento tradicional.

4.2.3 Dimensão representada por duas classes com relacionamento entre si do tipo "Agregação" (Compartilhada ou Composta) ou Associação simples, onde a classe diretamente ligada à classe Fato está do lado um (1) do relacionamento

Neste caso, nosso modelo genérico deve ser estendido de forma a atender um tipo de distribuidora cujos produtos finais para venda ao consumidor são em verdade sequências numeradas de produtos, ou seja, edições dos produtos (suponhamos por exemplo que nossa distribuidora comercialize Revistas que periodicamente lançam novas edições). Importante ressaltar que a granularidade definida para o sistema genérico de distribuidora de produtos permanece o mesmo, ou seja, controlando a venda de certo produto; ocorre que, na situação deste exemplo, os valores numéricos (Quantidade, ValorVenda e Preço) presentes na classe Fato (Venda) referem-se aos itens efetivamente comercializados que são as edições dos produtos (Figura 16).

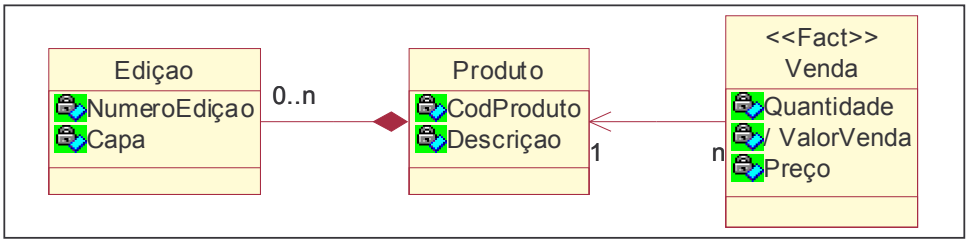


Fig.16 Modelo Conceitual dimensional OO onde a dimensão "Produto" sofreu extensões de forma a atender a existência de edições dos produtos (que são os itens efetivamente comercializados)

4.2.3.1 Segundo o mapeamento tradicional de OO para Relacional

O mapeamento através dos procedimentos tradicionais, neste caso, gera um esquema completamente ineficaz para um data warehouse.

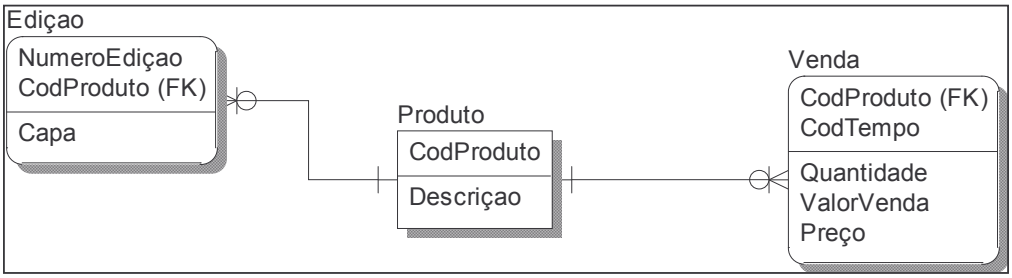


Fig.17 Modelo lógico dimensional Relacional (mapeamento tradicional)

Através do diagrama da figura 17 pode-se perceber que as regras de mapeamento tradicional fazem com que a chave primária de Produto migre como chave estrangeira para a tabela de Edição (Ediçao), tornando impossíveis as sumarizações de Venda por Edição. Para a execução das agregações de valores segundo atributos de certa tabela, é imprescindível a existência da chave estrangeira desta na tabela onde as operações devem ocorrer, ou a obtenção de tal ligação por meio de uma consulta como nos exemplos dos itens 4.2.2.1 e 4.2.2.2. Neste exemplo, a chave de Ediçao não está presente em Produto (cuja chave está por sua vez, em Venda), impossibilitando assim qualquer operação sobre as Edições.

4.2.3.2 Segundo o mapeamento dimensional proposto

Neste caso, devemos mapear todas as classes integrantes da dimensão Produto (Produto e Ediçao) em uma única tabela relacionada com a tabela Fato (Venda), conforme figura a seguir:

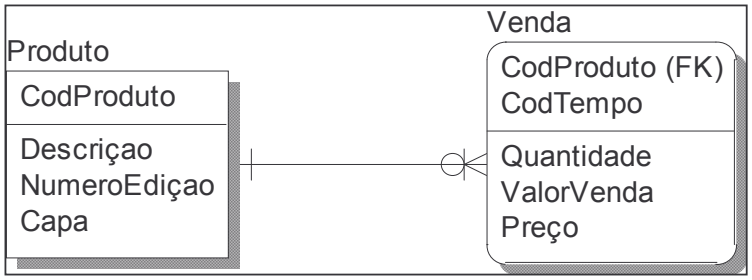


Fig.18 Primeira abordagem para mapeamento dimensional de modelo conceitual OO para lógico relacional

Poderíamos ainda citar uma outra solução intermediária entre o método tradicional e o dimensional, que seria o mapeamento de cada classe para uma tabela, mas com a migração invertida da chave primária de "Ediçao" para "Produto", de forma a permitir as operações sobre Edições na tabela Fato (Venda), conforme figura 19. Tal solução, embora torne viável o projeto do data warehouse, tem pior relação custo-benefício, em função do aumento da quantidade de junções com a conseqüente queda de desempenho.

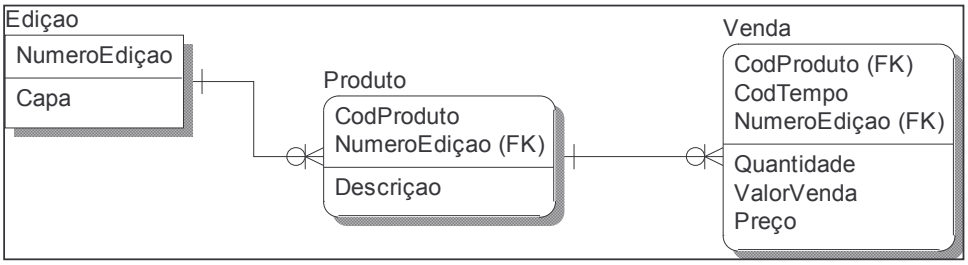


Fig.19 Segunda Abordagem para mapeamento dimensional de modelo Conceitual OO para lógico relacional

Importante comentar que ambas as abordagens conseguem resolver um problema que o mapeamento tradicional de esquemas Conceituais OO para lógicos relacionais não pode solucionar, já que foi concebido para atender sistemas OLTP e não a sistemas dimensionais estendidos, onde já existe uma associação pré-concebida e imutável entre as classes dimensão e a classe Fato.

4.2.4 Dimensão representada por uma hierarquia de Classes

Na presente situação, a dimensão Produto sofrerá extensões de forma a contemplar diferentes tipos de produtos, cada um com um grupo de características específicas. Tal tipo de ambiente, na orientação a objetos, é representado por uma hierarquia, onde as subclasses possuem uma associação "É um tipo de" com a superclasse, como na figura 20:

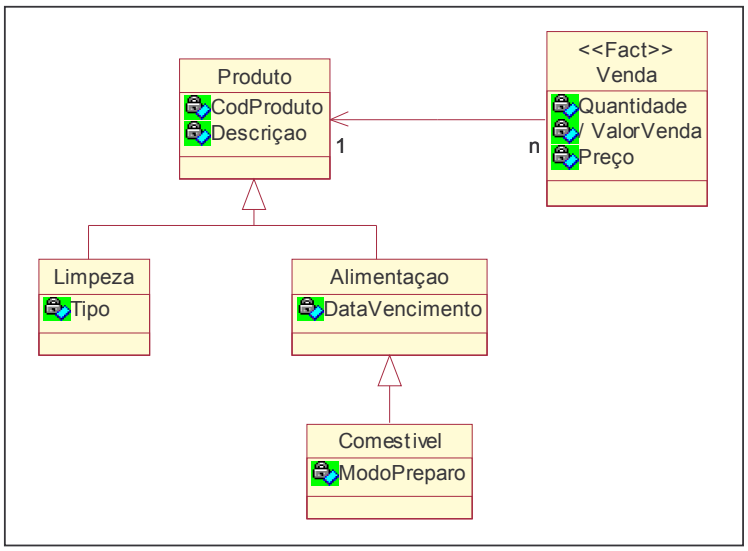


Fig.20 Modelo conceitual dimensional OO, onde a dimensão Produto sofreu extensões de forma a contemplar, com características específicas, certos tipos de Produtos (representados pelas subclasses "Limpeza", "Alimentaçao" e "Comestivel")

Importante ressaltar que, na literatura dimensional, o termo hierarquia muitas vezes é empregado com um significado diferente daquele da Orientação a Objetos. O exemplo mais comum é o emprego do termo hierarquia para designar, por exemplo a relação entre dia, mês e ano dentro de uma dimensão Tempo. A única hierarquia existente entre tais elementos é a de granularidade, ou seja, dias (unidade mais específica ou de menor mensuração) podem ser agrupados em meses que, por sua vez, podem ser agrupados em anos; não se deve entender nesta hierarquia nenhum significado (como proposto pela Orientação a Objetos e utilizado no presente trabalho) da forma: dia "É um tipo de" mês, que por sua vez "É um tipo de" ano. Definitivamente, o significado de hierarquia empregado na literatura dimensional expressa tão somente hierarquia de granularidade de forma a representar para qual termo as operações de roll-up levarão os valores presentes no nível corrente.

Sendo assim, vejamos como mapear este novo caso de distribuidora onde desejamos representar por exemplo que “Limpeza” e “Alimentaçao” são tipos de produtos (“Produto”), mas com características específicas (respectivamente: Tipo e DataVencimento) e ainda que “Comestivel” é um tipo de “Alimentaçao” (e, conseqüentemente, também de “Produto”), mas com outro atributo especial denominado ModoPreparo.

4.2.4.1 Segundo o mapeamento tradicional de OO para Relacional

As hierarquias de classes no modelo conceitual OO podem ser mapeadas para o modelo lógico relacional segundo três abordagens:

4.2.4.1.1 Primeira abordagem

Criação de uma tabela para cada classe e migração da chave primária da superclasse para as subclasses que a receberão como chave estrangeira

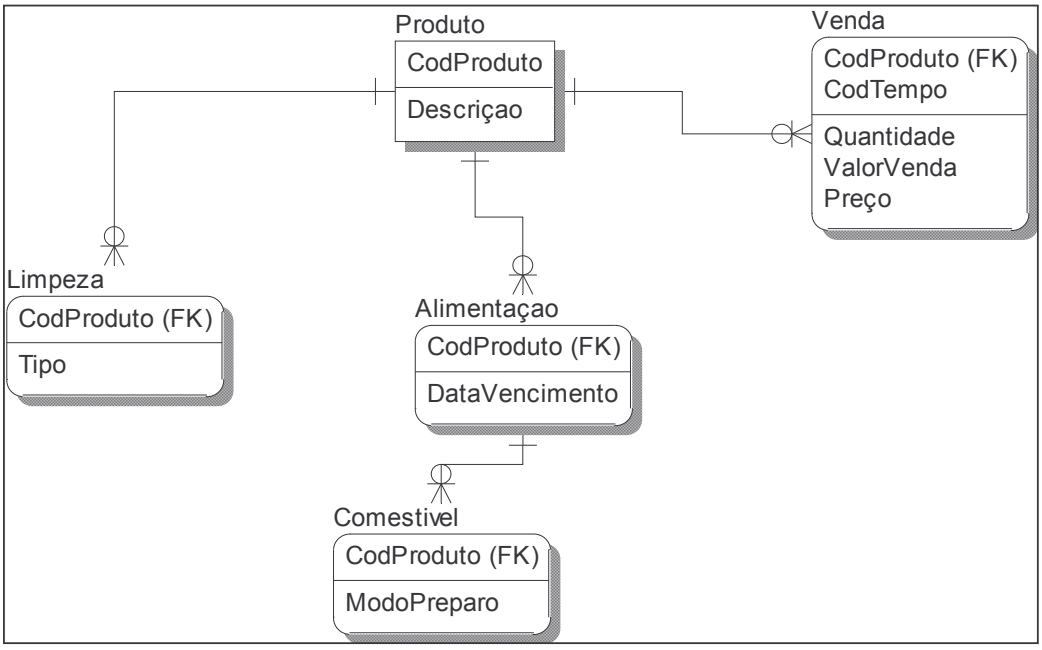


Fig.21 Mapeamento OO para Relacional – Uma tabela para cada classe

Por meio desta abordagem, é possível implementarmos o data warehouse, mas com altíssimo custo em termos de junções; a sumarização de valores da classe fato (Venda) segundo todos os produtos (independentemente de suas sub-hierarquias) deve ser obtida por meio do uso de Outer Joins.

4.2.4.1.2 Segunda abordagem

Espalhamento dos atributos das superclasses nas subclasses, com a conseqüente criação de uma tabela para cada uma das subclasses de nível mais baixo em sua hierarquia.

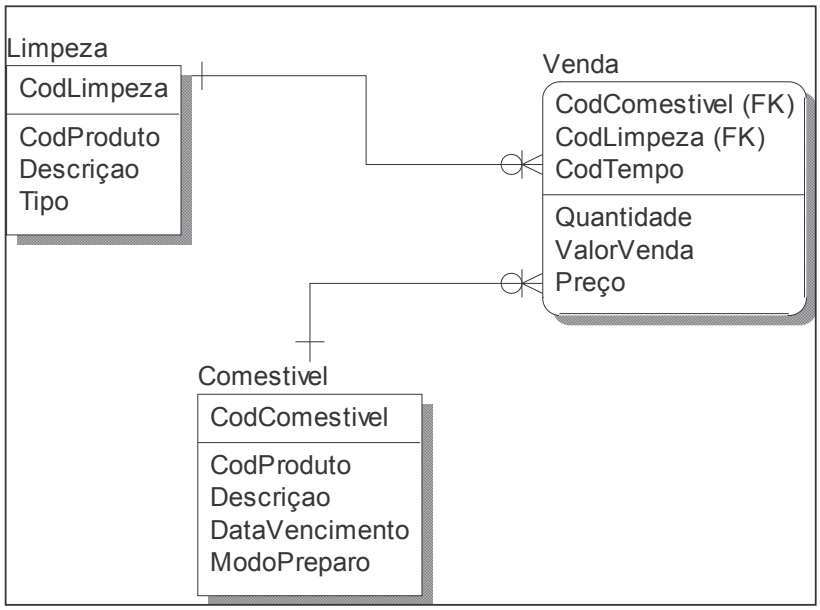


Fig.22 Mapeamento OO para Relacional – Espalhamento dos atributos das superclasses nas subclasses

Esta é a abordagem mais desvantajosa para o data warehouse, embora seja a mais econômica em termos de espaço de armazenamento, visto que ocasionará a implementação de um relacionamento entre cada subclasse (agora transformada em tabela) e a classe Fato, aumentando bastante a carga de gerência sobre as consultas (de forma a identificar com qual parte da dimensão Produto se deseja trabalhar, além de conviver com diversos campos nulos na tabela Fato), e praticamente inviabilizando a filosofia de simplicidade presente no modelo Estrela.

4.2.4.1.3 Terceira abordagem

Criação de uma única tabela contendo todos os atributos, da superclasse e das subclasses, conforme figura a seguir.

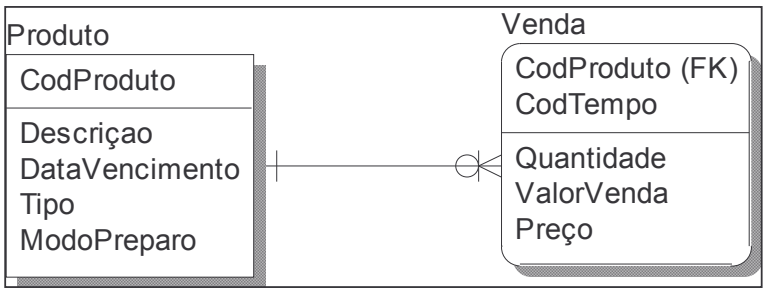


Fig.23 Mapeamento OO para Relacional – Mapeamento de todas as classes e seus atributos em uma única tabela ("Produto")

Esta abordagem é a melhor para o mapeamento de hierarquias de classes em sistemas dimensionais, já que utiliza uma única tabela para toda a dimensão (eliminando a necessidade de junções) e atende às expectativas de simplicidade e performance requeridas por um data warehouse, embora possua o custo do aumento de espaço de armazenamento e manipulação de valores nulos dentro da dimensão ("Produto").

Por todas as suas vantagens, tal abordagem é exatamente a mesma recomendada pelo mapeamento dimensional.

4.2.5 Mapeamento dos estereótipos

Conforme já mencionado anteriormente, uma das grandes vantagens do uso da orientação a objetos para a modelagem de data warehouses é a possibilidade do uso de estereótipos em substituição à representação de associações com extensas hierarquias de classes, permitindo, com grande simplicidade, a coexistência de conceitos diversos em um mesmo diagrama.

Na modelagem de um sistema que contemple funcionalidades analíticas e geográficas, o uso dos estereótipos mostrou-se como a chave para a simplicidade de representação: as classes convencionais ou não geográficas utilizam nenhum estereótipo, enquanto as

geográficas recebem um estereótipo caracterizador de seu tipo de dado (ponto, linha, polígono, multiponto, multilinha, multipolígono ou geométrico).

O mapeamento dos estereótipos geográficos ocorre por meio da adição de um novo campo do tipo de dado especificado pelo próprio estereótipo à tabela em questão. O OpenGIS, em seu documento de especificação para a SQL (OPENGIS, 1999), preconiza a criação de tabelas geográficas (aquelas que possuam ao menos um atributo geográfico) em duas fases: na primeira, a criação da tabela com os dados convencionais, e na segunda fase, o uso de uma função do SGBD para a adição de campos geográficos à relação já existente, já que tal adição implica no registro de tal campo em uma tabela de metadados geográficos.

Assim, para o PostGIS, extensão espacial ao SGBD PostGreSQL (e, em sua versão 0.8 – de novembro/2003 – totalmente aderente às especificações do documento citado) a adição de um campo geográfico ocorre da seguinte forma: inicialmente, a tabela com seus campos convencionais é criada; em seguida, utiliza-se a função: SELECT AddGeometryColumn (<nome_Banco_de_Dados>, <nome_tabela>, <nome_coluna>, <srid>, <tipo_dado_geografico>, <dimensão>), como por exemplo em: "`Create Table ESTRADAS (ID int4, NOME varchar(25))`" (criação da tabela convencional); "`Select AddGeometryColumn('db_Estradas', 'estradasgeo', 'geo', 423, 'LINESTRING', 2)`".

Fig.24 Classe "Rio" utilizando estereótipo geográfico do tipo Linha (LineString)

O mapeamento da classe Rio, presente na figura 24, para o modelo relacional, produziria a seguinte tabela (admitindo-se o uso do atributo NomeRio como chave primária da relação):

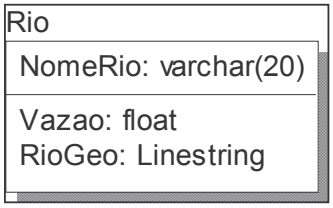


Fig.25 Tabela “Rio”, criada em duas fases, representando os atributos convencionais e o recém-adicionado atributo geográfico “RioGeo”, do tipo LineString

5. FERRAMENTA DE INTEGRAÇÃO OLAP E GIS

5.1 TRABALHOS RELACIONADOS

Diversos outros trabalhos foram conduzidos no sentido de buscar a integração entre sistemas analíticos e geográficos e, por sua relevância, serão aqui citados para estabelecer um paralelo entre o que já foi produzido e o que está sendo proposto nesta dissertação.

GeoMiner é um projeto de datamining espacial (Stefanovic, 1997), onde extensões são propostas ao modelo Estrela de forma a prover ponteiros para objetos espaciais aos itens da tabela Fato e das dimensões. Tal abordagem em verdade não integra um sistema DW com outro GIS, apenas permite operações OLAP sobre um Cubo com dados georeferenciados.

"Materialização seletiva: Um método eficiente para a construção de cubos de dados espaciais" (HAN; STEFANOVIC; KOPERSKI, 1998) foi a primeira proposta de um Data Warehouse espacial, ou seja, um cubo de dados onde as dimensões e as medições poderiam ser atributos convencionais ou espaciais. O trabalho foca justamente nas medições (atributos da tabela Fato) espaciais, propondo uma estratégia para a definição de quais objetos geográficos deveriam ser materializados (pré-calculados).

Em 2001, Papadias et al. (PAPADIAS ET.AL, 2001) focam seu esforço nas dimensões espaciais, onde muitas vezes as hierarquias e agrupamentos não podem ser bem definidos antecipadamente e que, por tal motivo, não podem se beneficiar das técnicas já consagradas

de materialização das visões. O trabalho em questão propõe uma hierarquia de agrupamento baseada no índice espacial em sua menor granularidade.

MapCube (SLTCV, 2000) é um projeto parcialmente patrocinado pelo Exército Americano, cujo principal objetivo é ser uma extensão ao conceito de Cubo de Dados para o domínio espacial. É um operador análogo ao operador CUBE usado em consultas sobre SGBD-R, com a diferença de que enquanto o operador Cube retorna uma tabela (relação) como resultado de suas operações, o MapCube retorna mapas e tabelas, executando operações como Soma e Interseção sobre dados geográficos.

GOAL (Geographical Information On-Line Analysis, ou Análise On-Line de Informação Geográfica) (KMM, 2000) é parte do programa de pesquisa da União Européia INCO-COPERNICUS e seu objetivo é integrar GIS com DW/OLAP de forma a prover os decisores com informações sob as perspectivas dimensionais e espaciais. GOAL utiliza metadados como apoio ao processo de associação entre dados geográficos e dimensionais. O processo de associação é coordenado por um módulo de integração preparado para comunicar com o componente GIS (Software GT Media98 e ArcView) e com o componente OLAP (MSOLAP), provendo uma interface que permite comunicação entre eles.

O projeto SIGOLAP (FTC, 2001) busca a integração entre OLAP e SIG utilizando um modelo de integração em três camadas: Camada de Fontes de Dados, responsável por prover os dados a serem analisados; Camada de Middleware, responsável pela mediação entre a camada de Dados e a de Aplicação e cujo principal componente é o Módulo de Integração que mapeia os conceitos usados pelas ferramentas OLAP e SIG; e Camada de Aplicação, onde estão os aplicativos que devem prover funcionalidades analíticas e geográficas. Em 2002, FERREIRA (2002) propõe um Modelo de Integração baseado em metadados, que atua como um modelo de referência para a tradução de conceitos utilizados por ferramentas OLAP e GIS.

GOLAPA (Geographical On-Line Analytical Processing Architecture) (FTS, 2001), propõe a integração entre DW/OLAP e GIS através do uso de tecnologias abertas e extensíveis como Java, XML e Web Services. Tal arquitetura contempla as fases de ETL (Extração, Transformação e Carga) para um Data Warehouse geográfico.

Do acima exposto, conclui-se que MapCube e GeoMiner são abordagens para processamento analítico sobre dados geográficos, GOAL, SIGOLAP e GOLAPA não abordam um modelo unificado de modelagem que contemple características analíticas e geográficas, e sim tratam as duas tecnologias de forma separada, propondo algum tipo de

módulo de integração que mapeie os dados e as requisições entre os dois componentes (OLAP e SIG) dos sistemas. Os trabalhos de (HAN; STEFANOVIC; KOPERSKI, 1998) e de (PAPADIAS ET AL., 2001), são os que mais se aproximam da proposta desta dissertação, mas, apesar de proporem uma aplicação OLAP que trate dados convencionais e espaciais em um mesmo ambiente, não propõem nenhuma técnica que permita a modelagem do sistema como um todo a partir de seu nível de abstração conceitual.

A principal motivação para a execução do presente trabalho é a possibilidade de tornar mais simples a modelagem e implementação de um sistema de suporte à decisão que integre características analíticas (de um Data Warehouse) e geográficas (de um SIG), fazendo com que tais conceitos possam coexistir no mesmo modelo tanto no nível conceitual (como já mostrado anteriormente por meio do uso de UML e estereótipos geográficos na modelagem de um DW) quanto no nível da implementação e, mais ainda, que possa ser o mais direto possível tal mapeamento entre os citados níveis de abstração. Desta forma, fazia-se necessária uma ferramenta que efetivamente implementasse e fosse tão simples de operar quanto aquilo que era modelado de forma tão trivial em nível conceitual.

Para manipular os dados em um Data Warehouse, de forma a prover consultas ad-hoc on-line, a ferramenta deveria prover funcionalidades de uma aplicação OLAP; além de ser capaz de executar funções geográficas sobre os dados em questão; assim, o aplicativo PostGeoOlap é, em verdade, uma ferramenta OLAP que analisa os dados da tabela Fato (medições, apenas de caráter não-espacial), segundo as perspectivas das dimensões, que podem ser convencionais ou espaciais, obtendo tal processamento geográfico do SGBD-OR Espacial.

5.2 A FERRAMENTA POSTGEOOLAP

Para a implementação da ferramenta de integração, foi utilizado um SGBD (Sistema de Gerenciamento de Banco de Dados) objeto-relacional Espacial, ou seja, que além da capacidade de armazenamento de informações convencionais (não-geográficas), é capaz também de armazenar informações posicionais relativas ao tipo geométrico que está sendo representado e ainda de executar funções geográficas sobre estes dados, como por exemplo,

verificar se um objeto se encontra dentro de outro, ou a 100 metros de distância de um outro referencial.

Utilizando-se um SGBD Espacial, é possível construir uma aplicação OLAP Espacial que, além da manipulação analítica dos dados, permita a análise destes mesmos dados por seus atributos geográficos; e esta é a proposta da ferramenta aqui chamada de PostGeoOlap: substituir a antiga arquitetura de duas aplicações (uma OLAP e outra GIS) com um módulo de integração, por uma única aplicação onde o aplicativo OLAP incorpora também as funcionalidades geográficas (providas, em verdade, pelo SGBD Espacial). A ferramenta implementada trabalha apenas com dados convencionais na tabela Fato (medições) e indistintamente com dados convencionais ou geográficos nas dimensões.

Na construção do PostGeoOlap, foi utilizada a linguagem VB.Net, acrescida de um componente (.ocx, ou seja, Objeto COM, denominado PlanetGis) para a visualização dos dados em um Mapa. O SGBD utilizado é o PostGreSql com o PostGIS, que é uma extensão geográfica ao PostGreSql (em verdade cria tipos de dados definidos pelo usuário – geográficos- e funções geográficas no Banco de Dados).

5.2.1 Tipos de Aplicações OLAP

As aplicações OLAP permitem a execução de consultas ad-hoc definidas on-line pelo usuário sobre um determinado Data Warehouse, que realizam operações tais como sumarização, média e valor máximo sobre os fatos a analisar segundo perspectivas pré-definidas (dimensões). Como tais repositórios de dados (o Data Warehouse) possuem grande quantidade de dados, as execuções das sumarizações ou das outras operações sobre tais dados são procedimentos que consumiriam muito tempo para serem executados realmente “on-line”. Por isso, as aplicações OLAP resolvem a necessidade de performance por meio da pré-agregação das informações do Data Warehouse, de modo que quando o usuário as solicita, o aplicativo busca as informações de uma das agregações pré-processadas que atenda aos critérios de consulta solicitado.

5.2.1.1 Aplicações ROLAP

As aplicações ROLAP (Relational OLAP) – (MICROSOFT., 2000) mantêm os dados originais em sua base relacional e geram as agregações também em relações (tabelas). Possuem como vantagem o fato de buscar os dados para as agregações no próprio modelo e normalmente no mesmo SGBD em que trabalha a aplicação OLAP, não tendo limite de tamanho; em contrapartida, são mais lentos em sua manipulação (na execução das consultas on-line feitas pelo usuário).

5.2.1.2 Aplicações MOLAP

Multi-dimensional OLAP, trabalham com estruturas de dados proprietárias e multidimensionais, que agilizam o processamento das consultas (MICROSOFT., 2000). Exatamente neste ponto está a maior vantagem dos sistemas MOLAP: sua grande velocidade na execução das consultas solicitadas pelo usuário; em contrapartida, a carga dos dados da base original para sua estrutura especial é bastante demorada, o que inviabiliza tal tipo de aplicação dependendo do tamanho da base de dados.

5.2.1.3 Aplicações HOLAP

Hybrid OLAP, tem este nome por utilizar as duas tecnologias citadas anteriormente, ou seja, mantém os dados em seu repositório original relacional e as agregações são mantidas em estrutura multidimensional (MICROSOFT, 2000).

A ferramenta desenvolvida no presente trabalho – PostGeoOlap - utiliza a tecnologia ROLAP (Relational OLAP), apesar da existência de um esforço de padronização na área de metadados para data warehouses (utilizando Orientação a Objetos) conduzido pela OMG

através do CWM (Common Warehouse Metamodel). A razão para a não utilização do padrão proposto no CWM, é que este se destina a aplicações Multidimensionais (MOLAP), e a ferramenta tema desta dissertação deve obrigatoriamente ser desenvolvida segundo a tecnologia ROLAP, visto que as consultas (queries) analíticas ou geográficas são ambas processadas e respondidas por um SGBD objeto-relacional e, para tanto, todos os dados (do nível base do data warehouse até as agregações) devem ser mantidos no modelo físico relacional.

5.2.2 Modelo de Classes da Ferramenta PostGeoOlap

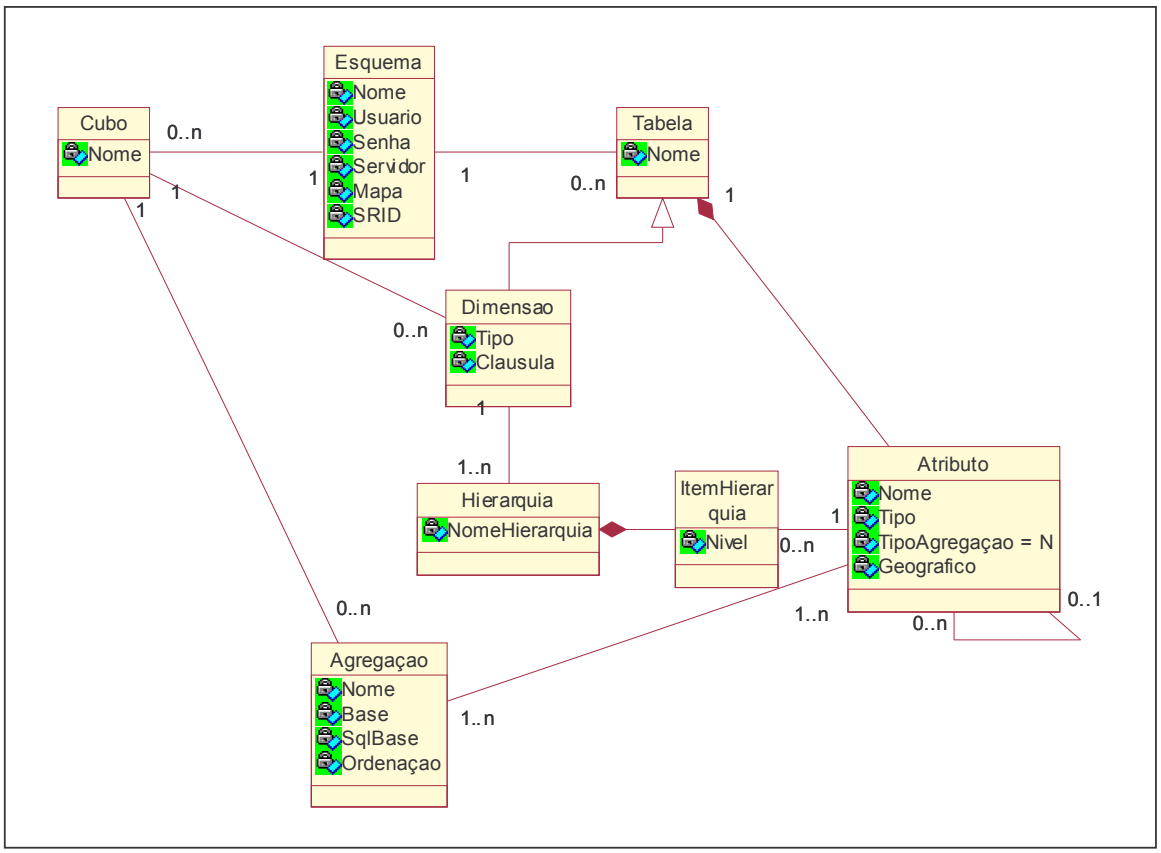


Fig.26 Diagrama de Classes da ferramenta OLAP – PostGeoOlap

O modelo de classes da figura 26 apresenta, em verdade, os metadados necessários à aplicação OLAP para a manipulação dos dados mantidos no repositório do PostGreSQL. Segue a descrição de cada classe:

5.2.2.1 Esquema: A classe Esquema representa o banco de dados (database do PostGreSql) no qual estão as bases de dados (o data warehouse) com o qual se deseja trabalhar. Os atributos de Esquema são:

5.2.2.1.1 Nome (Caractere): É o nome dado ao Esquema (qualquer nome intuitivo para o usuário, de preferência ligado ao data warehouse a ser manipulado).

5.2.2.1.2 Usuario (Caractere): Nome do usuário a ser usado para a conexão com o PostGreSQL.

5.2.2.1.3 Senha (Caractere): Senha do usuário utilizado para a string de conexão com o SGBD.

5.2.2.1.4 Servidor (Caractere): Nome da máquina onde está localizado o servidor de banco de dados PostGreSQL.

5.2.2.1.5 Mapa (Caractere): Nome do arquivo .map (caminho para o arquivo, nome e extensão) – com o mapa e suas camadas ou "layers" componentes a serem utilizados pelo componente de visualização espacial (em verdade uma ocx, ou objeto COM) da ferramenta OLAP.

5.2.2.1.6 SRID (Numérico): Identificador do sistema de referenciamento geográfico para a região do mapa em questão (o mapa a ser usado pela ferramenta de visualização).

5.2.2.2 CUBO: Sobre um data warehouse, representado por um Esquema, pode-se utilizar a ferramenta OLAP para visualizar várias perspectivas ou assuntos do negócio, onde cada assunto de interesse para análise irá constituir um Cubo (utilizando-se os conceitos do modelo estrela de Ralph Kimball, onde um Cubo é composto por uma tabela Fato e por diversas outras tabelas, ligadas à primeira, chamadas Dimensão). O único atributo de Cubo é Nome (caractere), já que sua finalidade é apenas agrupar em um conceito as demais informações acerca de um assunto do negócio que deva ser analisado.

5.2.2.3 TABELA: São as tabelas ou relações existentes no banco de dados ou no Esquema com o qual se vai trabalhar. Das tabelas existentes, algumas (ou todas) serão selecionadas posteriormente para a função de Dimensão ou de tabela fato de cada um dos Cubos. O único atributo de Tabela é Nome (Caractere).

5.2.2.4 DIMENSÃO: Esta classe representa todos os componentes de um Cubo, ou seja, as tabelas Fato e as dimensões (a tabela Fato nada mais é do que uma dimensão cujo atributo "Tipo" é igual a "Fato"). A classe Dimensão é uma subclasse de Tabela, já que toda dimensão é, em última instância, uma tabela comum do banco de dados que foi selecionada pelo usuário com tal finalidade. São atributos de Dimensão:

5.2.2.4.1 Tipo (Caractere): Informa o tipo de dimensão a ser armazenada. Os tipos possíveis são: "NaoAgregavel", que indica que esta é uma dimensão que não será considerada para a agregação de dados em um Cubo. As dimensões não agregáveis são dimensões que possuem algum atributo geográfico e com as quais deseja-se fazer análises comparativas onde os relacionamentos são obtidos por meio do referencial geográfico. "Fato", indica que esta dimensão é na verdade a tabela Fato, ou seja, aquela que contém os valores a serem mensurados. "Dimensao", é uma dimensão no sentido estrito do termo para o modelo estrela de Kimball.

5.2.2.4.2 Clausula (Caractere): armazena a cláusula de junção da tabela dimensão com a tabela Fato.

5.2.2.5 HIERARQUIA: Esta classe representa as várias hierarquias entre os atributos de uma dimensão que podem ser empregadas para análise dos dados. O único atributo desta classe é o Nome da hierarquia.

5.2.2.6 ATRIBUTO: Classe que representa os dados existentes em cada uma das tabelas e, por conseqüente, em cada uma das dimensões. Os atributos são também ligados à classe Hierarquia por meio da classe ItemHierarquia. O auto-relacionamento destina-se à indicação de um atributo como rótulo para outro do tipo geográfico. Atributos da classe ATRIBUTO:

5.2.2.6.1 Nome (Caractere): Nome do atributo

5.2.2.6.2 Tipo (Caractere): É o tipo de dado do atributo para o PostGreSql (Ex.: varchar, int8, boolean, etc). Interessante ressaltar que para tipo de dado armazenado, a ferramenta PostGeoOlap presumirá sempre sua capacidade máxima de armazenamento.

5.2.2.6.3 TipoAgregaçao (Caractere): Indica o tipo de operação de agregação a ser aplicada a cada um dos atributos. Pode ser: N (Não-agregável), S (Soma – Função SUM da SQL), C (Contagem – Função COUNT da SQL), M (Valor Máximo – Função MAX da SQL), I (Valor Mínimo – Função MIN da SQL), A (Média – Função AVG da SQL). Como a maioria dos atributos de um sistema OLAP pertence às dimensões, o valor default para "TipoAgregaçao" é "N" (Não-agregável), já que os atributos que sofrem operações de sumarização ou média ou outra qualquer são aqueles pertencentes à tabela Fato.

5.2.2.6.4 Geografico (Booleano): Indica se determinado atributo é do tipo Geográfico ou não. Apesar de tal informação poder ser obtida a partir do atributo "Tipo", tal decisão tem a finalidade de incrementar o desempenho.

5.2.2.7 ITEMHIERARQUIA: Classe que aloca cada atributo a uma hierarquia, fornecendo a este atributo o seu nível, ou seja, sua posição ou número de ordem dentro da citada hierarquia. seu único atributo é o já citado "NIVEL" (Numérico).

5.2.2.8 AGREGAÇÃO: Classe que encerra as diversas agregações de dados implementadas com vistas a melhorar o desempenho do sistema. Tendo em vista as quantidades de dados serem enormes, a chave para fornecer informações agregadas em tempo razoável está no pré-processamento destes dados com vistas à produção das agregações. Atributos:

5.2.2.8.1 Nome (Caractere): Nome da agregação, tal nome é formado pelo nome do cubo adicionado do nome da hierarquia e da combinação dos níveis que estão sendo agregados em cada uma das dimensões pertencentes ao cubo corrente.

5.2.2.8.2 Base (Boolean): Informa se esta agregação é a base, ou seja, aquela da granularidade mínima e a partir da qual todas as demais agregações se originam.

5.2.2.8.3 SqlBase (Caractere): traz o comando SQL (SELECT) necessário para se fazer uma consulta sobre a base de dados em sua granularidade mínima.

5.2.2.8.4 Ordenação (Numérico): armazena o nr de ordem desta agregação, que seria o "custo" para esta agregação. Desta forma, a agregação mais "custosa", em termos de desempenho para se executar uma consulta seria a da menor granularidade (a base), já que é a que contém maior número de registros e junções; e a de menor custo seria aquela que agregasse os valores da tabela fato segundo os atributos mais agrupadores de cada uma das dimensões. Tal ordenação é utilizada no momento das consultas para que o sistema decida onde inicialmente iniciar sua busca pelos dados.

Apesar do presente trabalho propor o modelo de classes anterior para a modelagem da ferramenta PostGeoOlap, para fins de projeto, foi implementado um modelo com a restrição de cada dimensão possuir uma única hierarquia, o que resultou no modelo da figura 27:

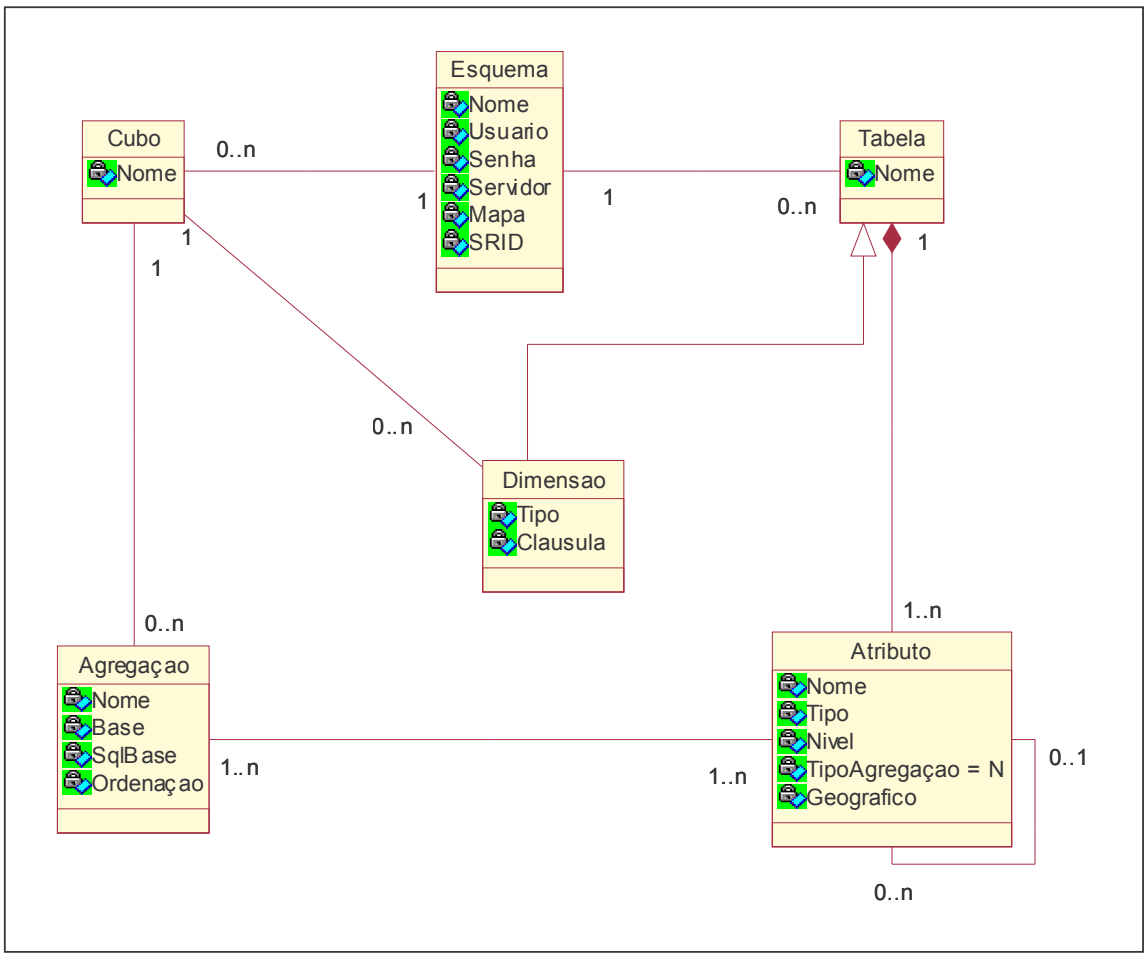


Fig.27 Modelo de Classes (Projeto) da ferramenta PostGeoOlap com uma única hierarquia por dimensão

O modelo de classes da figura 27 (metamodelo) foi mapeado para um arquivo do MS-ACCESS, gerando assim o repositório para os metadados da ferramenta PostGeoOlap.

5.3 FUNCIONAMENTO DA FERRAMENTA POSTGEOOLAP

A seguir serão apresentados os diagramas de seqüência (de alto nível de abstração) para cada um dos Casos de Uso da ferramenta PostGeoOlap, acompanhados de descrição de seu funcionamento

Caso de Uso	Finalidade
Criar Esquema	Cria uma conexão com um certo Banco de Dados do PostGreSQL onde serão posteriormente definidos os Cubos
Criar Cubo	Cria um Cubo dentro de determinado Esquema, selecionando também uma tabela como Fato e determinando seus itens numéricos e as operações desejadas sobre tais itens
Adicionar Dimensão	Cria uma perspectiva para análise dos dados contidos na tabela Fato, por meio da seleção a partir de uma das tabelas do Banco de dados. Determina também a hierarquia dentro da dimensão, atribuindo um nível para cada um de seus atributos
Criar Dimensão não-agregável	Cria uma dimensão que, embora não se constitua em perspectiva para os dados da tabela Fato (e, por tal motivo, não participe da geração das agregações), serve como referência para comparativos geográficos com as demais dimensões que possuam atributos geográficos
Processar Cubo	Verifica a massa de dados armazenados, por meio de seus metadados, buscando inferir sobre a velocidade de execução de consultas. As consultas avaliadas como de baixa performance (em termos de tempo de execução), são otimizadas por meio da criação de agregações, o que deixa o Cubo em condições de ser analisado sob qualquer perspectiva, dentro de tempo aceitável
Analisar Dados	Provê uma interface capaz de permitir a seleção de atributos de uma consulta, sob restrições convencionais ou geográficas e de exibir o resultado de tal consulta sob duas visões: uma tabular para análise dos dados não-espaciais e outra em um mapa para a visualização dos dados espaciais

Tab. 5 - Listagem de Casos de Uso da ferramenta PostGeoOlap

5.3.1 Caso de Uso: Criar Esquema (Conexão com um Banco de Dados e criação do Esquema)

O primeiro passo para a utilização da ferramenta é a criação do ESQUEMA que será utilizado, ou seja, a definição do Banco de Dados (database do PostGreSQL) onde estão os dados do data warehouse que será trabalhado pela aplicação OLAP.

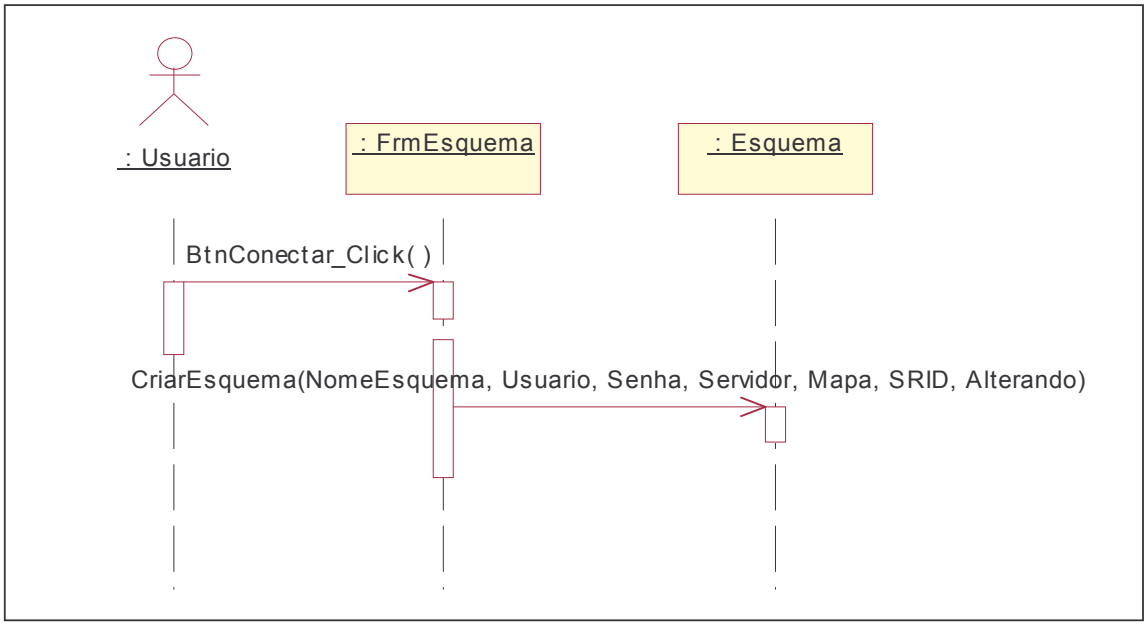


Fig.28 Diagrama de Seqüência: Criar Esquema

Na definição do Esquema, informa-se o nome do Computador onde está localizado o SGBD (PostGreSQL), o nome do Banco de Dados, o nome do usuário e senha necessários à conexão, o caminho e nome com extensão do arquivo do PlanetGIS onde serão visualizados os dados que possuírem alguma componente geográfica, e o SRID (Identificação do Sistema de Referenciamento Geográfico) do mapa em questão.

5.3.2 Caso de Uso: Criar Cubo

Uma vez criado o esquema, o próximo passo é a criação dos diversos cubos nos quais serão realizadas a análise e visualização dos dados e onde estarão definidas as estruturas do modelo estrela a ser manipulado pela aplicação OLAP, ou seja, a tabela Fato e as tabelas dimensão.

Após a criação do Cubo, o programa solicita imediatamente a seleção de uma das tabelas existentes no Banco de Dados para a função de tabela Fato e, para a tabela escolhida, a definição de quais serão os atributos numéricos, ou seja, os atributos da tabela Fato que devem sofrer as operações (Soma, Média, Mínimo, Máximo e Contagem) e quais das operações se deseja sobre tais atributos.

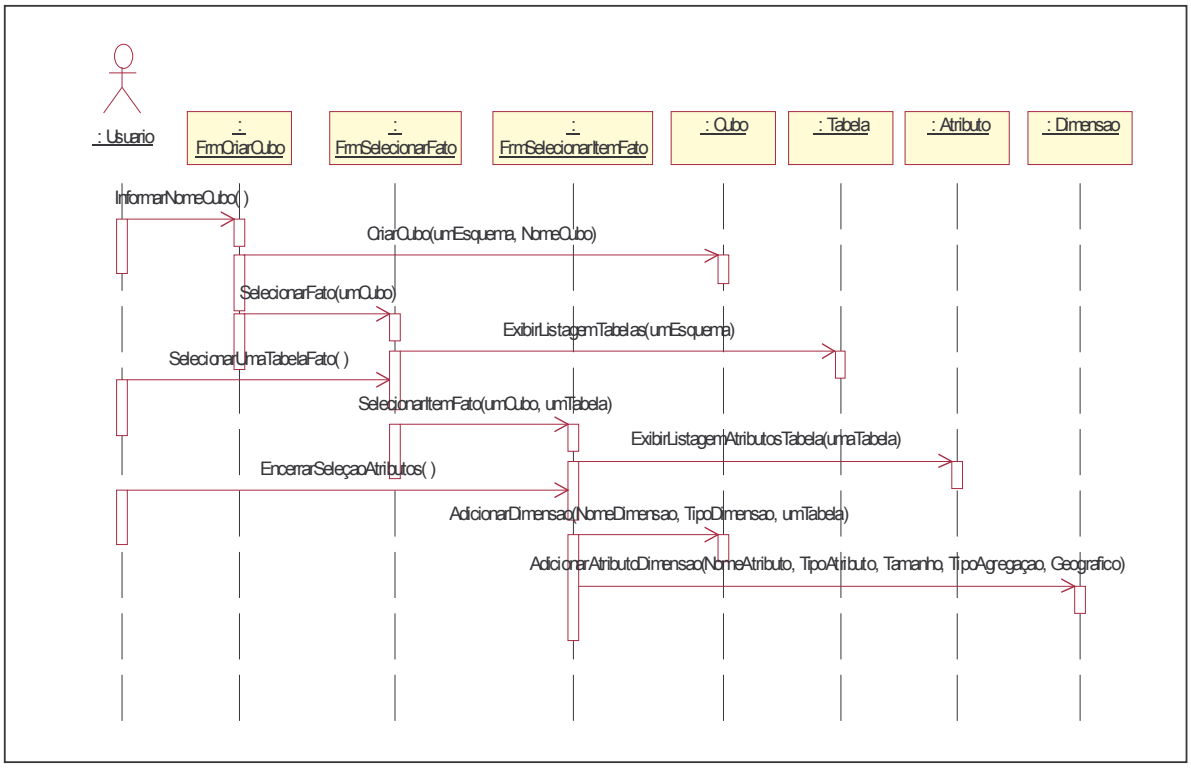


Fig.29 Diagrama de Seqüência: Criar Cubo

5.3.3 Caso de Uso: Adicionar Dimensão

Depois de definida a tabela fato, seus itens numéricos e quais operações se desejam sobre tais atributos, passa-se à definição das tabelas Dimensão, que são as perspectivas segundo as quais serão analisados os itens numéricos da tabela Fato. Nos mesmos moldes da

seleção da tabela Fato, o sistema apresenta uma lista com todas as tabelas do Banco de Dados e o usuário seleciona uma delas como dimensão, definindo, em seguida, a hierarquia dentro desta dimensão. Importante observar que vários atributos podem compartilhar o mesmo nível hierárquico, de forma que os atributos de menor granularidade recebam o nível (número de ordem) 9 (nove), e os atributos de maior granularidade recebam números menores até o limite de 1 (um). Assim, uma hierarquia no PostGeoOlap pode possuir até no máximo 9 níveis.

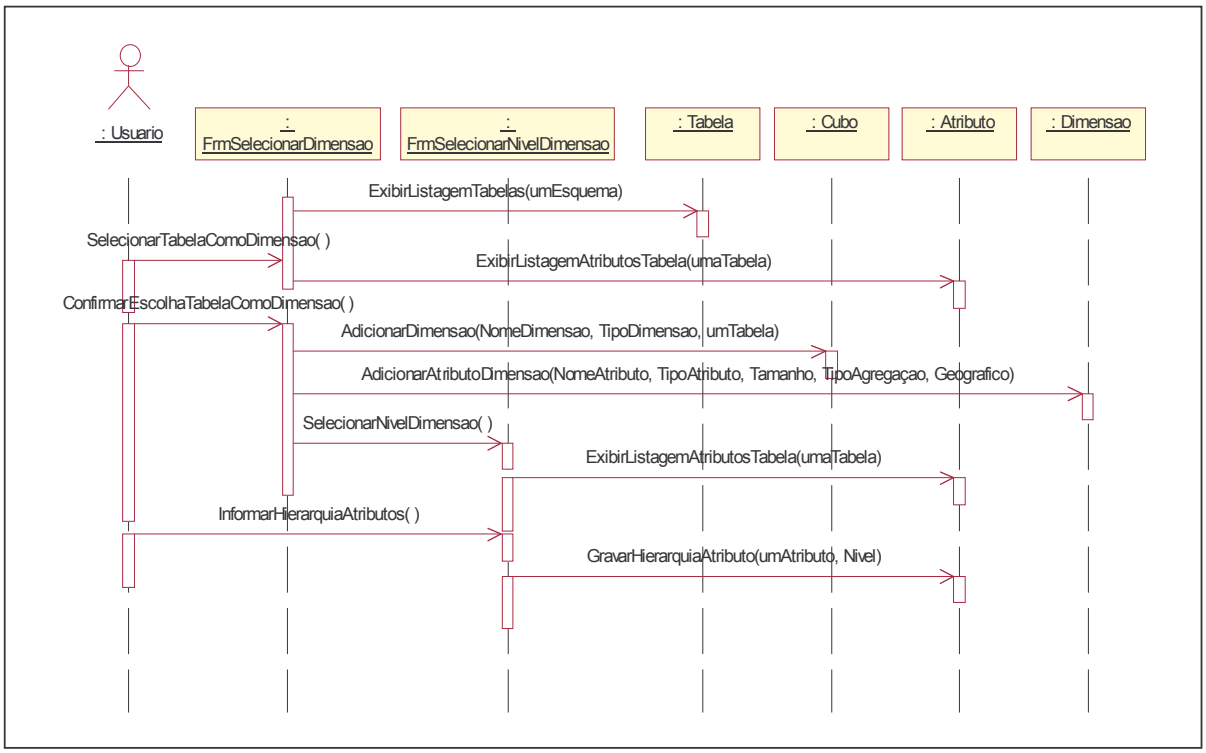


Fig.30 Diagrama de Seqüência: Adicionar Dimensão

Ressalta-se também que, conforme citado anteriormente, a ferramenta implementada contempla apenas uma única hierarquia por dimensão, mas no modelo ideal seria possível criar diferentes hierarquias, de forma que um atributo de uma dimensão pudesse estar em níveis diferentes dentro de cada hierarquia.

5.3.4 Caso de Uso: Criar Dimensão não-agregável

Neste Caso de Uso são criadas as dimensões de natureza geográfica que não participam do processamento do Cubo e conseqüente geração de agregações (daí o nome de "não-agregáveis"). A única finalidade das dimensões não-agregáveis é servir como referência para comparações com as demais dimensões geográficas do Cubo.

Importante mencionar que a criação de dimensões não-agregáveis não é um passo imprescindível para o processamento do Cubo, ou seja, não faz diferença adicionar as dimensões não-agregáveis antes ou após o citado processamento.

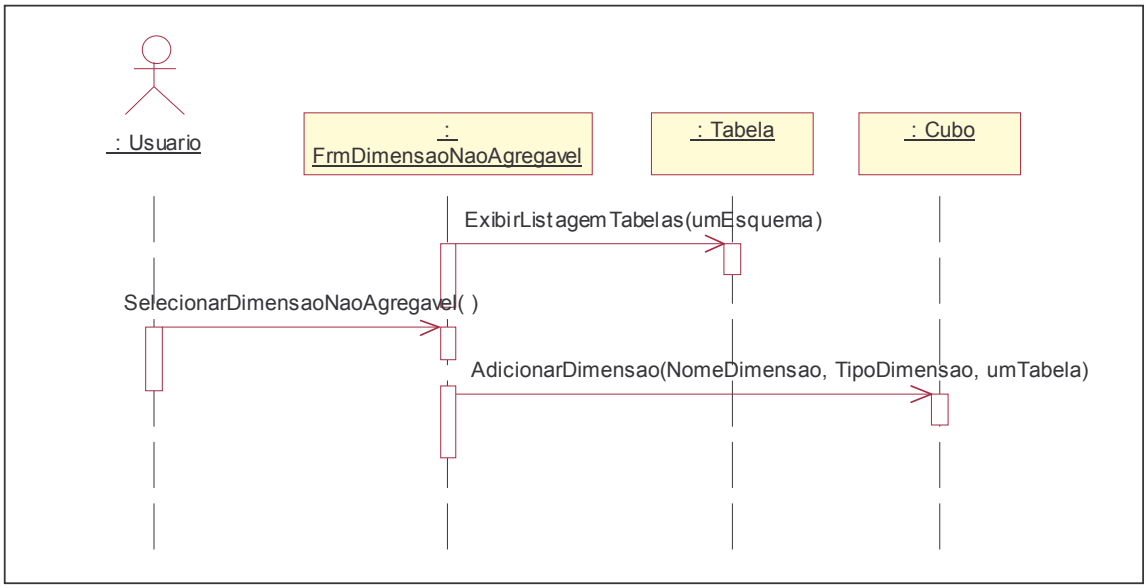


Fig.31 Diagrama de Seqüência: Adicionar Dimensão não-agregável

5.3.5 Caso de Uso: Processar Cubo

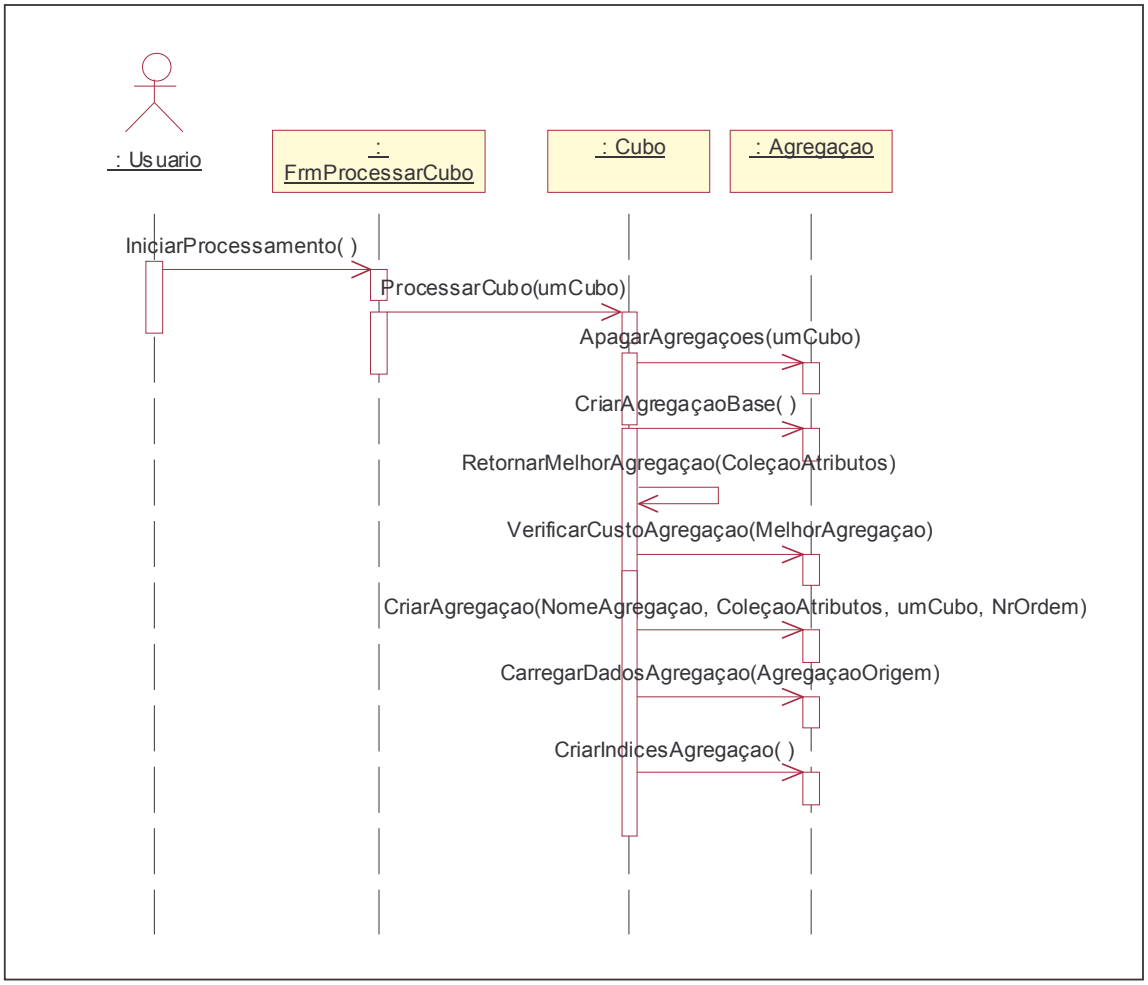


Fig.32 Diagrama de Seqüência: Processar Cubo

Uma vez finalizados os procedimentos de seleção e definição da hierarquia dentro de cada uma das dimensões, os metadados necessários à geração das agregações estão prontos.

Ou seja, o sistema dispõe de todas as informações acerca do Data Warehouse em questão necessárias à geração (ou não) de novas estruturas (tabelas) que auxiliarão a aplicação OLAP a responder às consultas dos usuários dentro de um intervalo de tempo aceitável. Assim, o próximo passo na utilização da ferramenta é o Processamento do Cubo, que significa o sistema avaliar, em função de alguma medida de custo, a necessidade de gerar agregações para melhorar o desempenho das Consultas. Tal processamento inicia pela exclusão de quaisquer agregações pré-existentes, a título de procedimento de administração/manutenção do Cubo (a melhoria deste procedimento é objeto de sugestão para trabalhos futuros).

Seja o seguinte esquema de um Data Warehouse, simplificado para facilitar os exemplos:

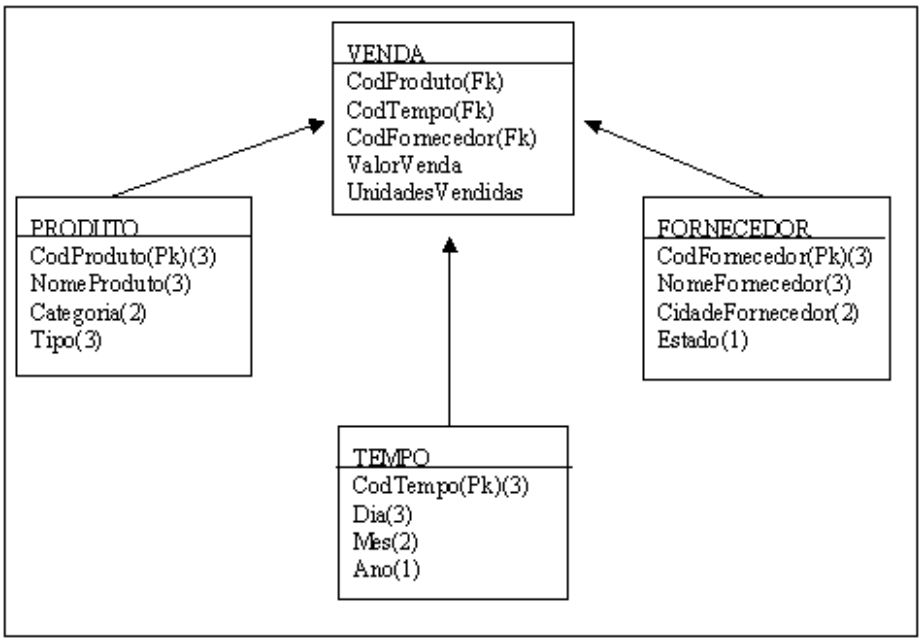


Fig.33 Esquema para um Data Warehouse (DW1)

No esquema em questão, estão representadas as dimensões Produto, Fornecedor e Tempo, cada uma com seus atributos e os níveis hierárquicos de cada atributo. A dimensão Produto possui como atributos: CodProduto (que está no nível da menor granularidade para esta dimensão), NomeProduto (que está no mesmo nível de CodProduto), Categoria (cujo nível hierárquico uma unidade menor que o de NomeProduto indica que uma mesma Categoria pode ser compartilhada por vários NomeProduto e CodProduto) e Tipo (cujo nível

hierárquico, a exemplo do que acontece com “Categoria” em relação a “NomeProduto” e “CodProduto”, é um agrupador de “Categoria”).

Percebe-se pelos níveis hierárquicos das dimensões, admitindo-se que estes estejam organizados hierarquicamente de forma decrescente, onde a menor granularidade recebe o maior valor de nível, que a tabela fato representa, em sua menor granularidade, a venda (ValorVenda e UnidadesVendidas) de um certo Produto (CodProduto e NomeProduto), pertencente a um Fornecedor (CodFornecedor e NomeFornecedor) e ocorrida em determinado Tempo (CodTempo e Dia), já que o nível de maior valor das três dimensões (3), está associado aos citados atributos. Ao nível de agregação inicial do Data Warehouse, ou seja, aquele sem qualquer agregação, já que as informações estão em sua granularidade mínima, denominamos nível base. Tal nível base tem seu script SQL armazenado nos metadados da ferramenta PostGeoOlap no início do processamento do Cubo (método “CriarAgregaçaoBase”).

A grande dificuldade encontrada por uma aplicação OLAP no momento de responder a consultas que necessitem de valores da tabela fato (ValorVenda e UnidadesVendidas) agregados segundo determinados níveis das tabelas dimensão, é o desempenho de tal consulta caso tais tabelas (Fato e Dimensão) possuam grandes quantidades de registros. Seja a seguinte consulta Q1: Exibir a soma de ValorVenda e UnidadesVendidas da Categoria de Produto denominada “Livros”, dos Fornecedores localizados no Estado de “SP” e ocorridas no período (Tempo) do Ano de “2002”. Em função do tamanho da base de dados, tal consulta pode levar muitos segundos, alguns minutos ou mesmo horas para ser executada, por isso, a aplicação OLAP, quando do processamento do Cubo, verifica se o tempo para execução de cada consulta possível (entendendo como consulta possível aquela cujo conjunto de atributos seja subconjunto do produto cartesiano entre cada nível de todas as dimensões) é viável (este termo “viável” é subjetivo e expressa o tempo que um usuário suportaria aguardar pela execução de sua requisição).

Desta forma, utilizando o exemplo anterior das vendas de livros fornecidas por “SP” no decorrer do ano de 2002: a aplicação OLAP verificaria o tempo de execução de tal consulta e, admitindo-se que o tempo entendido como viável fosse de 1 s (um segundo), e tal consulta, aplicada sobre o nível base, consumisse 10 minutos para ser finalizada, o aplicativo criaria então uma agregação (por exemplo, outro conjunto de tabelas, com os dados pré-sumarizados) sobre a qual tal consulta deveria ser executada e que permitisse que tal tempo diminuísse para o suposto patamar aceitável de 1 segundo. Assim, da mesma forma que, para

obter dados agregados da tabela Fato segundo as perspectivas dos níveis 2-1-1 das hierarquias das dimensões (ou seja, do nível 2 da dimensão Produto, nível 1 da dimensão Tempo e nível 1 da dimensão Fornecedor) em um tempo aceitável foi necessária a criação de uma estrutura de agregação, para as demais combinações de níveis hierárquicos das dimensões, tais estruturas de agregação também devem ser consideradas e, caso necessárias, implementadas. Um exemplo seria a agregação dos Fatos segundo o nível 0 de Produto (que significa não considerar dados de Produto na agregação ou tratar de "todos" os produtos), o nível 2 de Tempo – "Mes" e o nível 1 de Fornecedor - "Estado"; criando a perspectiva 0-2-1.

Para a construção das agregações, duas abordagens foram consideradas:

1) A primeira abordagem é sugerida por Kimball (KIMBALL, 1996), e preconiza, para cada perspectiva (ex.: Combinação 3-2-2 dos níveis das hierarquias das dimensões do exemplo de Data Warehouse anterior) a criação de um conjunto de tabelas. Tal conjunto seria composto por uma nova tabela para conter, de forma agregada, os registros da tabela fato e novas tabelas para cada nível hierárquico de cada dimensão, contendo apenas os atributos referentes ao nível em questão.

Desta forma, para a consulta Q1, teríamos (caso a consulta aplicada sobre o nível base dispendesse mais tempo que o "viável") a criação de uma nova tabela para os registros agregados da Fato (Soma de "ValorVenda" e "UnidadesVendidas"), uma nova tabela agregada para a dimensão Produto com todos os atributos de nível igual e abaixo do nível de "Categoria", uma outra tabela agregada para a dimensão Fornecedor, composta pelos atributos de nível menor ou igual a "Estado" e uma última tabela também agregada para a dimensão Tempo, com todos os atributos de nível igual ou menor que "Ano". Importante ressaltar que tal abordagem implica na criação de chaves primárias artificiais nas tabelas dimensão agregadas e, conseqüentemente, também na exportação de tais chaves para a tabela fato agregada.

Ralph Kimball (KIMBALL, 1996), em seu trabalho, cita ainda uma segunda técnica para a construção de agregações baseadas na criação de novos campos nas tabelas já existentes, mas tal técnica, segundo o próprio autor, pode levar a problemas de "dupla contagem", e por isso, não foi aventada como uma técnica viável no presente trabalho.

2) A segunda abordagem cria, para cada perspectiva (ou seja, cada combinação de níveis hierárquicos das dimensões; assim, 2-2-3 seriam os atributos presentes no segundo nível de Produto (2), segundo nível de Tempo (2) e terceiro nível de Fornecedor (3)), uma única tabela contendo os dados agregados da tabela Fato e os atributos de interesse (aqueles

de nível hierárquico igual ou menor ao da perspectiva em questão). Assim, para o exemplo da consulta Q1, teríamos uma única tabela, a título de estrutura de agregação, com os seguintes atributos: Soma de "ValorVenda", Soma de "Unidades Vendidas", "Categoria", "Ano" e "Estado".

Comparando-se as duas abordagens, foi adotado para o presente trabalho a segunda delas porque, apesar da primeira, sugerida por Kimball, gerar um repositório de dados com menor tamanho (já que a tabela Fato de cada agregação permanece normalizada, referenciando suas dimensões agregadas por meio de chaves estrangeiras), a performance da segunda é superior, já que não serão necessárias junções para recuperar informações das agregações (a junção só é necessária para a recuperação de informações do nível base, ou seja, quando todos os dados desejados pertencerem à granularidade mínima de armazenamento do Data Warehouse), e a administração das agregações também é mais simples, permitindo inclusive que tais agregações sejam implementadas por meio de Visões Materializadas ("Materialized Views").

Desta forma, durante o Processamento do Cubo, a aplicação OLAP, para cada uma das perspectivas possíveis (que, no caso do exemplo de DW, seriam: 2-3-1, 0-2-2, 2-0-0, etc), verifica, por meio do método "RetornarMelhorAgregaçao", qual agregação já existente possui todos os atributos desejados, selecionando a de menor esforço computacional, ou seja, a mais agregada. Em seguida verifica, por meio do método "VerificarCustoAgregaçao", o desempenho da execução de uma consulta contendo todos os atributos dos níveis presentes na perspectiva em questão sobre a citada "melhor agregação" obtida anteriormente, como no exemplo a seguir: Q2= "`SELECT SUM(ValorVenda), SUM(UnidadesVendidas), Categoria, Ano, Estado FROM <MelhorAgregaçao> GROUP BY Categoria, Ano, Estado`".

Ressalte-se aqui que, como o PostGeoOlap utiliza a abordagem do emprego de tabelas a título de estruturas de agregação, o método "RetornarMelhorAgregaçao" retornará sempre uma única tabela ou, em último caso, a agregação base, que é a junção de todas as tabelas do Cubo em suas granularidades mínimas.

O exemplo a seguir representa a mesma consulta anterior, Q2, aplicada sobre o nível base (o que ocorreria caso nenhuma outra agregação contivesse os atributos buscados pela Q2): "`SELECT SUM(ValorVenda), SUM(UnidadesVendidas), Categoria, Ano, Estado FROM VENDA, PRODUTO, TEMPO, FORNECEDOR WHERE VENDA.CodProduto = PRODUTO.CodProduto and VENDA.CodTempo =`

TEMPO.CodTempo and VENDA.CodFornecedor = FORNECEDOR.CodFornecedor GROUP BY Categoria, Ano, Estado". Caso o desempenho da consulta anterior esteja abaixo de um limiar pré-estabelecido, a ferramenta cria uma nova estrutura de agregação que, conforme dito anteriormente, é uma nova tabela.

Para a criação da nova tabela de Agregação (método "CriarAgregaçao"), a aplicação OLAP obtém, a partir dos metadados, os atributos (com seus respectivos tipos de dados) das dimensões envolvidas na agregação metadados e, assim, executa um comando SQL (DDL) para a criação da nova relação. Exemplificando tal comando SQL de criação da tabela de agregação sobre o exemplo, teríamos: "CREATE TABLE <NomeAgregação> (Soma_ValorVenda <TipoDeDado>, Soma_UnidadesVendidas <TipoDeDado>, Categoria <TipoDeDado>, Ano <TipoDeDado>, Estado <TipoDeDado>)", onde <TipoDeDado> é o mesmo tipo de dado do atributo em sua dimensão de origem. Importante ressaltar que, caso algum dos atributos de alguma das dimensões fosse do tipo Geográfico (Ponto, Linha, Polígono, MultiPonto, MultiLinha, MultiPolígono ou Geográfico (Complexo)), a criação da tabela deveria ser realizada em duas fases: Na primeira, aconteceria a criação da tabela com seus atributos convencionais (não-geográficos) – conforme exemplo anterior, e na segunda (atendendo às especificações do OpenGIS), a adição dos campos geográficos por meio de uma função, já que tal adição de um campo geográfico implica na criação de um registro em uma tabela de metadados geográficos – geometry_columns.

O passo seguinte é a inserção dos dados agregados nesta nova tabela (método "CarregarDadosAgregaçao") e, para tanto, é usado um comando SQL de inserção, como o seguinte (ainda baseado no exemplo de Data Warehouse deste capítulo): "INSERT INTO <NomeAgregação> (Soma_ValorVenda, Soma_UnidadesVendidas, Categoria, Ano, Estado) VALUES (SELECT SUM(ValorVenda), SUM(UnidadesVendidas), Categoria, Ano, Estado FROM VENDA, PRODUTO, TEMPO, FORNECEDOR WHERE VENDA.CodProduto = PRODUTO.CodProduto and VENDA.CodTempo = TEMPO.CodTempo and VENDA.CodFornecedor = FORNECEDOR.CodFornecedor GROUP BY Categoria, Ano, Estado)".

O último procedimento na geração da tabela de agregação é a geração de um conjunto de índices (método "CriarIndicesAgregaçao") que agilize sua utilização. Para tanto, são criados índices sobre todos os campos não-numéricos (entendendo por numéricos – ou seja,

os que não serão indexados - aqueles obtidos por agregação da tabela Fato, como por exemplo: “Soma_ValorVenda” e “Soma_UnidadesVendidas”). O tipo de índice utilizado para os campos não-geográficos é o B-Tree (Árvore B), que se aplica muito bem a dados que podem ser ordenados segundo um único eixo, como Caractere, data, número, etc; e para os campos geográficos foi utilizado o GiST (Generalized Search Tree, ou Árvore de Busca Generalizada), porque segundo a documentação do PostGIS (POSTGIS, 2004), que é a extensão espacial ao SGBD PostGreSql, a implementação dos índices do tipo R-Tree (que são normalmente usados para indexar campos geográficos em outros SGBD espaciais) no PostGreSql, não é tão robusta quanto a do GiST e, assim, o PostGIS implementa um índice do tipo R-Tree sobre os índices GiST do PostGreSql para indexar os dados geográficos.

Importante ainda frisar que, no processamento do Cubo, o sistema sempre inicia a geração das agregações de maior perspectiva, ou seja, (baseado no exemplo) o sistema inicia pela geração da agregação de perspectiva 3-3-2, em seguida, processa a 3-3-1, e assim prossegue até por último processar a 0-0-0. A razão de tal ordem é permitir que novas agregações possam utilizar, como fonte de seus dados agregados, não as tabelas do nível base, mas uma outra agregação, agilizando assim o processamento. Desta forma, por exemplo, a agregação 0-0-0 pode ser processada a partir de dados já pré-sumarizados na agregação 1-1-1.

5.3.5.1 Algoritmo utilizado para processamento do Cubo

Processar o Cubo significa analisar o custo-benefício da criação de novas estruturas de agregação para cada combinação dos níveis presentes em cada uma das hierarquias, ou seja, significa fazer o processamento de “k” hierarquias, cada uma delas com “n” níveis (lembrando que o “n” máximo para o PostGeoOlap é igual a 9).

Para resolver tal problema de forma genérica, ou seja, permitindo à ferramenta processar Cubos com quaisquer quantidades de dimensões (para fins de projeto, definimos cada dimensão com apenas uma hierarquia, assim, a quantidade de dimensões é a mesma de hierarquias), cada uma delas possuindo qualquer quantidade de níveis (desde que menor que 9), foi utilizado um algoritmo que se utiliza de uma estrutura de dados do tipo pilha, conforme explicado a seguir. Seja o Cubo representado pela Figura 33. Admite-se cada letra como um

nível da hierarquia de cada dimensão (lembrar que cada nível hierárquico em uma dimensão pode conter um ou mais atributos).

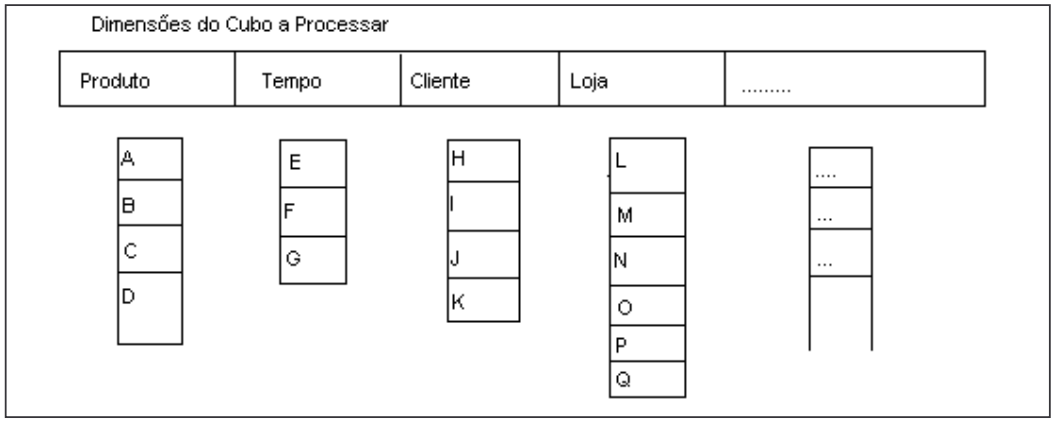


Fig.34 Esquema de Cubo a Processar

O primeiro passo é a criação de uma estrutura do tipo pilha com um elemento para cada nível da primeira dimensão (Produto), conforme a Figura 35.

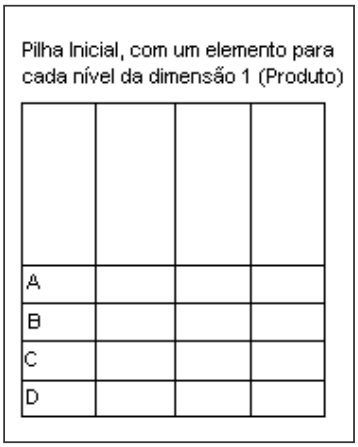


Fig.35 Pilha inicial, onde cada elemento contém um nível da primeira dimensão

Em seguida o algoritmo, implementado por meio de um loop cuja condição de parada é a pilha estar vazia, retira o primeiro elemento do topo da pilha e checa se a quantidade de níveis do elemento é igual à quantidade de dimensões. Caso a quantidade de níveis seja a

mesma de dimensões, significa que tal elemento está pronto para sofrer as análises de custo-benefício e dar origem ou não a uma nova estrutura de agregação, não retornando mais à pilha. Caso a quantidade de níveis do elemento retirado seja menor que a de dimensões, faz-se a combinação do nível deste elemento com cada um dos níveis presentes na dimensão 2 (Tempo), criando assim, três novos elementos que são então adicionados à pilha:

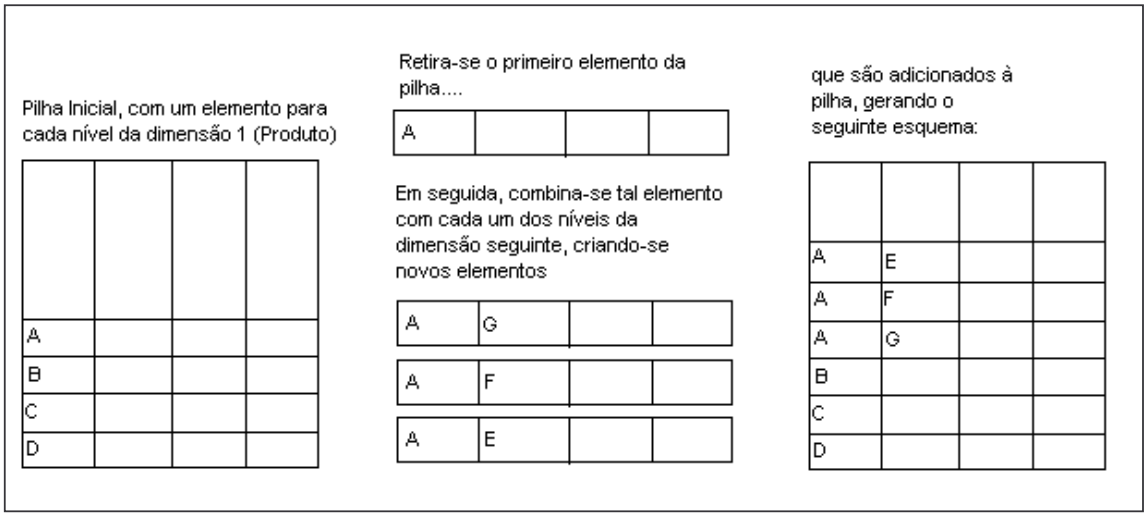


Fig.36 Primeiro ciclo do algoritmo para Processamento do Cubo

A figura 37 ilustra o próximo passo do algoritmo. A sucessão de ciclos do algoritmo se encarrega de fazer com que os elementos contenham todas as combinações possíveis dos níveis das dimensões e, à medida que ficam completos, vão sendo retirados definitivamente da pilha para validação da necessidade de geração de novas estruturas de agregação. O pseudo-código para o algoritmo em questão é apresentado no Apêndice deste trabalho.

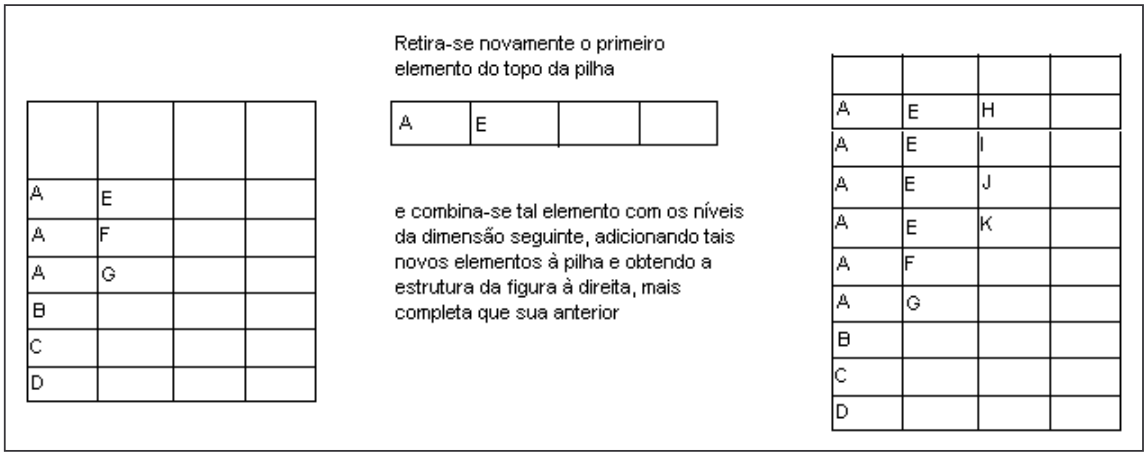


Fig.37 Passo seguinte do algoritmo de Processamento do Cubo

5.3.6 Analisar Dados

Finalizada a geração de índices para a última agregação, o processamento do Cubo está encerrado, e os dados estão prontos para serem analisados pela aplicação OLAP.

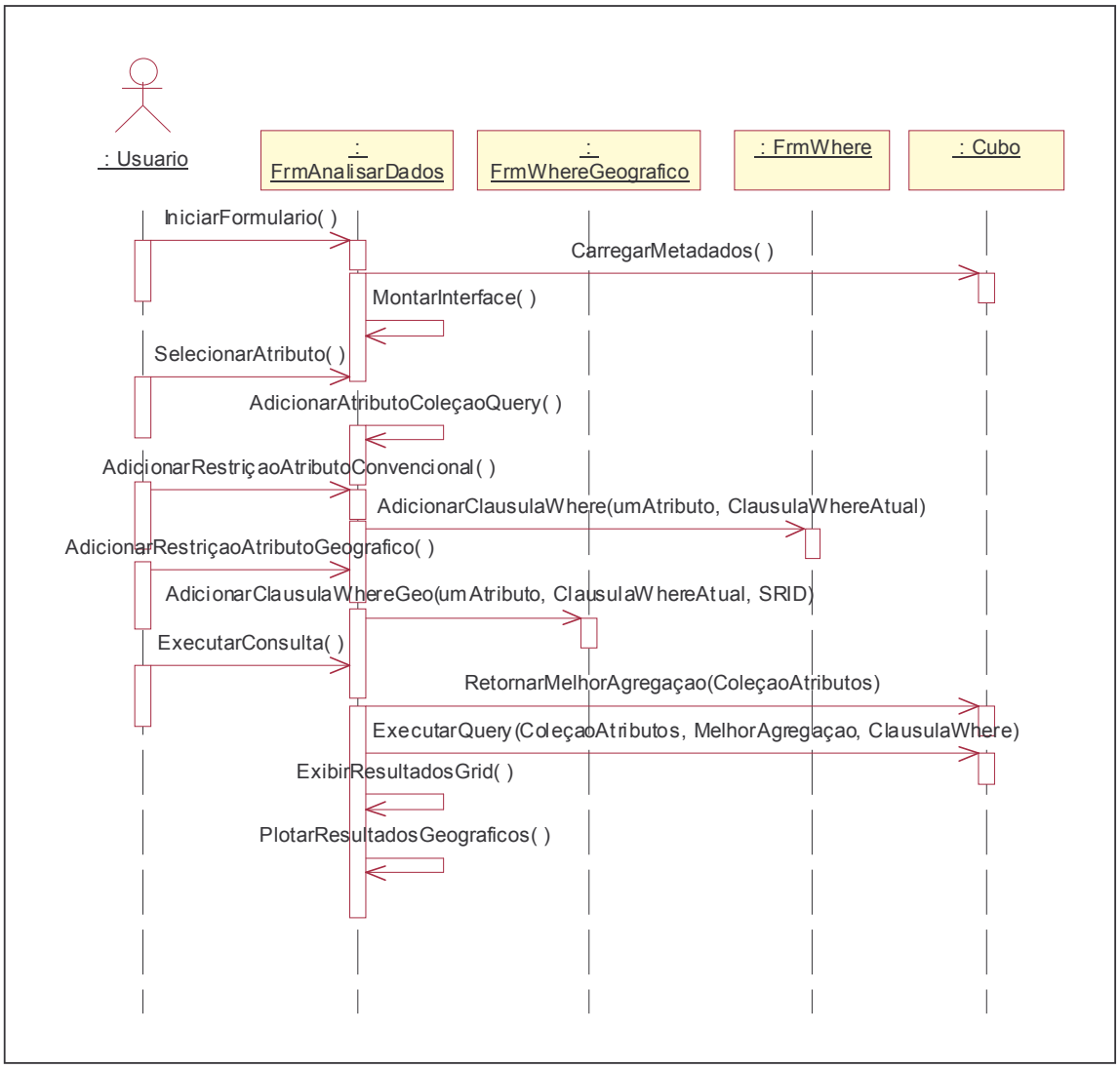


Fig.38 Diagrama de Seqüência: Analisar dados

A ferramenta exibe uma tela para Análise dos dados composta por quatro Quadros: O primeiro quadro localizado na parte superior esquerda da tela exibe todos os atributos pertencentes à Tabela Fato e as Dimensões (obtidos e exibidos por meio dos métodos "CarregarMetadados" e "MontarInterface"), o quadro inferior esquerdo exibe textualmente as

restrições montadas graficamente pelo usuário, o terceiro quadro localizado na parte superior direita da tela, é um componente para a visualização geográfica dos dados (PlanetGIS .ocx), e o quarto quadro, logo abaixo do terceiro, exibe em uma planilha os resultados analíticos das consultas.

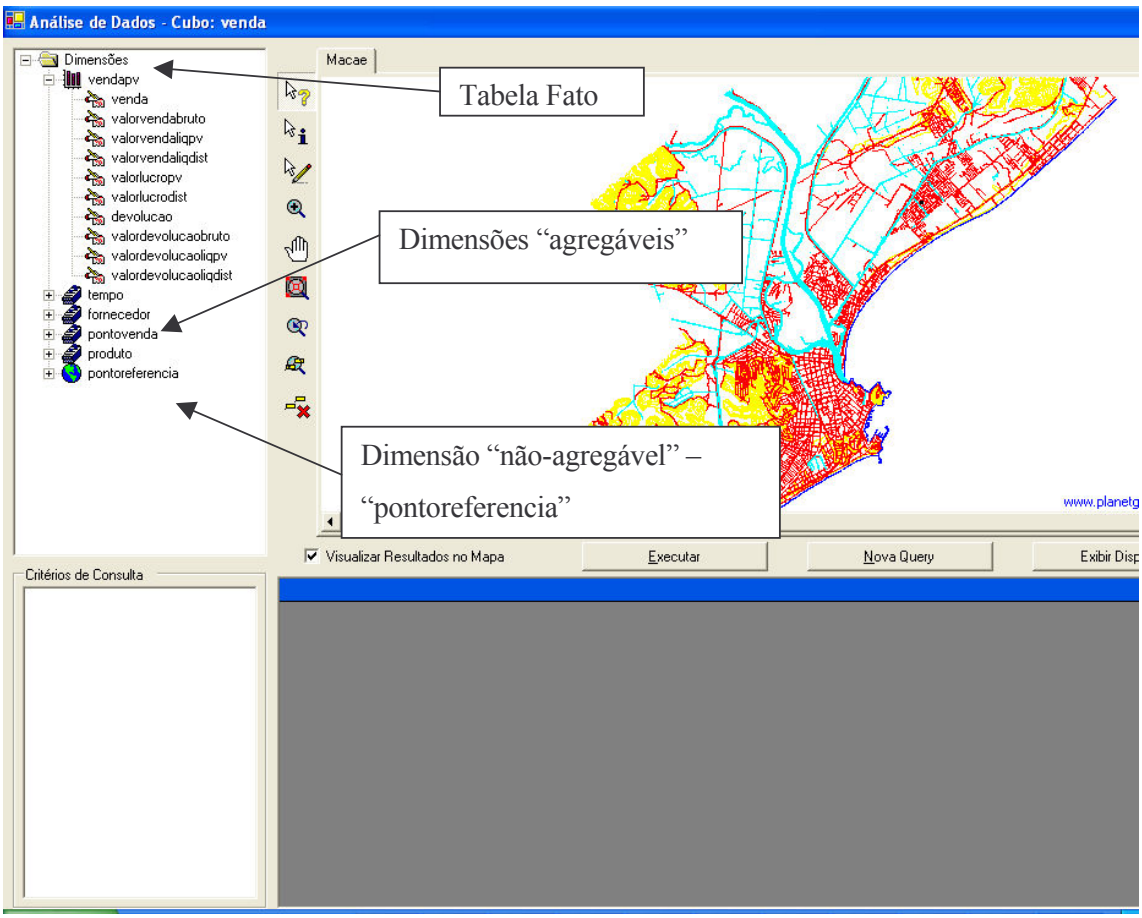


Fig.39 Tela para análise dos dados

A tela de análise dos dados funciona da seguinte forma: o usuário seleciona os atributos de interesse, tanto da tabela Fato, como das tabelas Dimensão, por meio de duplo clique sobre uma árvore de atributos e a aplicação vai armazenando tal seleção em uma coleção – em memória - por meio do método "AdicionarAtributoColeçãoQuery".

O usuário pode adicionar uma restrição a um atributo não-geográfico (Cláusula WHERE) por meio do método "AdicionarRestriçaoAtributoConvencional", que apresenta uma tela com um conjunto de operadores de comparação (igual, menor que, maior que, etc) e exibe uma lista com os valores presentes na base de dados do atributo em questão para escolha do usuário. É possível a concatenação de restrições em uma mesma cláusula do tipo

WHERE, ou seja, pode-se evocar a tela de montagem de restrição sobre atributos convencionais mais de uma vez, e tais subcláusulas serão concatenadas. Exemplo de cláusula (baseado no exemplo de DW do item 4 – Processamento do Cubo): " WHERE Estado = 'RJ'".

Caso o atributo selecionado para montagem de restrição seja do tipo geográfico, o sistema, por meio do método "AdicionarRestriçaoAtributoGeografico" apresenta uma tela com uma lista de funções que podem ser aplicadas sobre o atributo em questão com relação a algum outro atributo (também geográfico) pertencente a uma dimensão aqui denominada de "não-agregável" (porque tal dimensão destina-se apenas a fornecer atributos geográficos com os quais se possa estabelecer comparações, não participando das agregações de um Cubo). A restrição sobre atributos geográficos também pode ser concatenada com restrições não-geográficas ou mesmo com outras restrições de natureza geográfica.

As restrições aplicadas a atributos geográficos acontecem na forma da utilização de funções geográficas envolvendo o atributo geográfico corrente e um outro atributo geográfico pertencente a outra dimensão (aqui denominada de dimensão não-agregável, já que esta não interessa ao cubo como perspectiva para análise dos dados numéricos – ou seja, dos dados da tabela Fato- e, por tal razão, não é processada ou agregada, servindo apenas como perspectiva para comparações com outros atributos geográficos pertencentes às dimensões "agregáveis"). Supondo uma consulta que desejasse a exibição da soma dos valores das vendas de todos os produtos ocorridas num raio de 500 metros da escola "ABC", teríamos: "`Select SUM(ValorVenda) From < NomeAgregação> WHERE distance( <AtributoGEO>, <AtributoGEO correspondente à escola "ABC") <= 500`".

De posse da coleção de atributos desejados pelo usuário, e das restrições impostas a estes mesmos atributos (sejam tais restrições convencionais ou geográficas), o Cubo, por meio do método "RetornarMelhorAgregaçao", busca, dentre as agregações existentes, na ordem da mais agregada para a menos agregada (a menos agregada de todas é a tabela Fato com suas dimensões, ou seja, o nível base) a primeira delas e, conseqüentemente, a de menor custo computacional para consultas, que possua todos os atributos selecionados pelo usuário; obtendo desta forma, a agregação ideal sobre a qual submeter a consulta solicitada, o que será feito pelo método "ExecutarQuery".

Os resultados provenientes da submissão ao SGBD Espacial (PostGreSql + PostGIS) da consulta montada pelo método "ExecutarQuery", são apresentados ao usuário da seguinte

forma: Os dados convencionais são apresentados em uma planilha (“grid”) através do método “ExibirResultadosGrid” e, caso haja, dentre os atributos da consulta, algum do tipo geográfico, suas instâncias são também exibidas no componente de visualização (mapa) por meio da operação “PlotarResultadosGeograficos”.

Convém lembrar que todo o processamento da consulta, seja ele convencional ou geográfico, é processado no SGBD espacial, não necessitando de nenhum outro aplicativo ou componente para a obtenção de tais resultados.

5.3.6.1 Visualização dos resultados

De posse das restrições e da agregação (no caso da ferramenta, uma tabela) sobre a qual submeter a consulta, o programa executa a consulta sobre o SGBD e recebe, em resposta, os resultados (sejam analíticos ou geográficos). Os dados convencionais são então, dispostos em formato tabular dentro de uma grid, e os geográficos são passados para um componente de visualização para representação sobre um mapa. Para tal função, a ferramenta PostGeoOlap utiliza o objeto COM PlanetGIS.

Para fins de análise de dados que possuam alguma componente espacial, deve-se obter um arquivo do PlanetGIS (extensão .map), ou então criá-lo a partir de um arquivo shape (.shp) – é possível converter um arquivo .shp em um .map - que represente os objetos relevantes na região geográfica de interesse, como por exemplo, os polígonos representativos dos distritos de um município, seus bairros, as linhas referentes aos logradouros e pontos de referência como Bancos, Rodoviária e Escolas. Cada um dos tipos de objetos estará representado em um display (ou camada) distinto que pode ser exibido independentemente um do outro, servindo como “cenário” para a exibição dos dados geográficos resultantes da consulta submetida ao SGBD Espacial, que, por sua vez, também serão exibidos em um “display” à parte. Uma vez que todo o processamento, analítico e geográfico, é executado pelo SGBD Espacial, os objetos do tipo geográfico (Ponto, Polígono, Linha, etc) retornados pelas consultas são passados ao componente de visualização (PlanetGIS) para que sejam “desenhados” na tela.

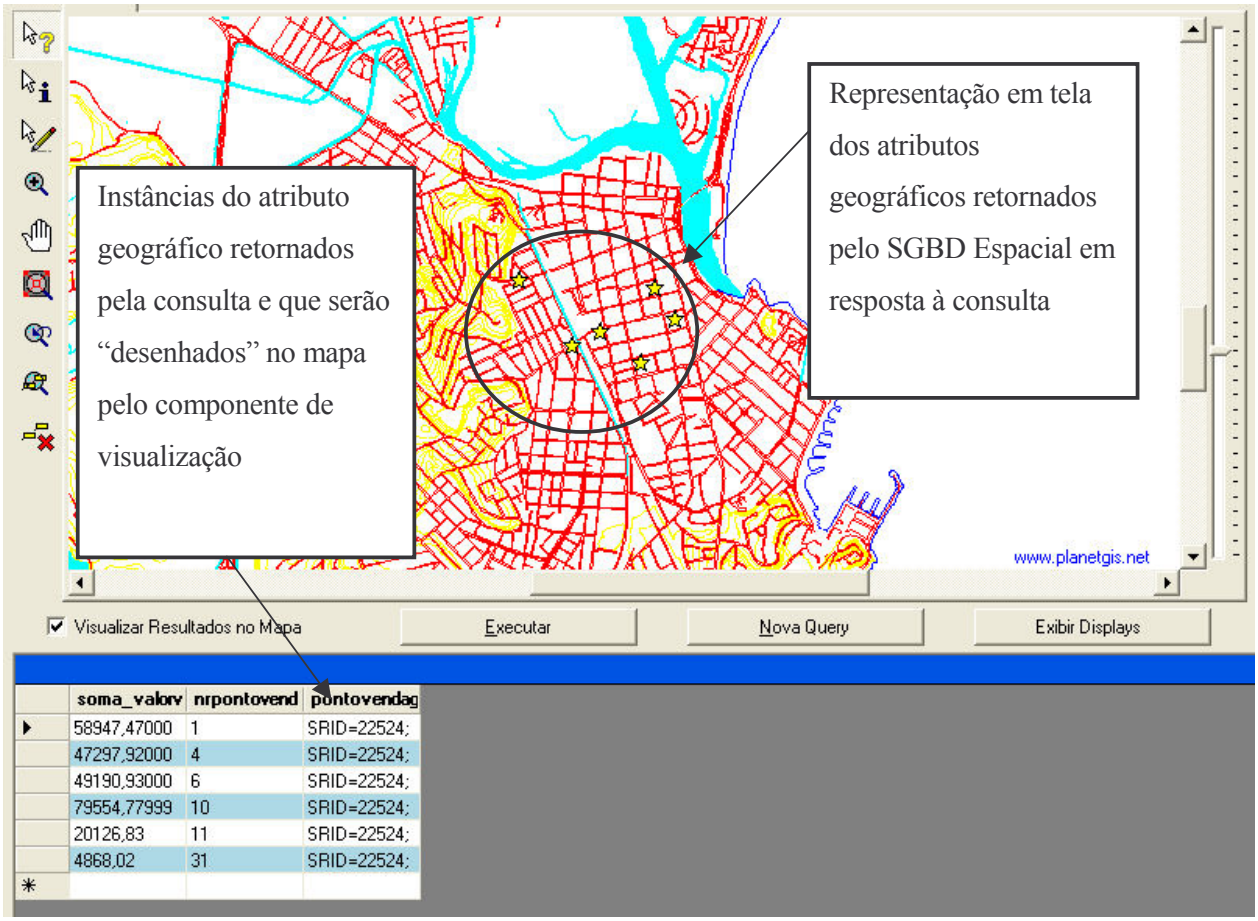


Fig.40 Visualização dos resultados geográficos em um mapa

É também importante ressaltar que um conjunto de formas geométricas sem identificação em uma tela é de pouca valia para o usuário que não pode identificar, por exemplo, dentre os cinco objetos do tipo Linha desenhados na tela como resposta a uma consulta, qual deles corresponde ao Rio Parnaíba e qual corresponde ao Rio Paraíba do Sul. Portanto, faz-se necessário associar aos atributos geográficos (Por exemplo: bairro_geo – Polígono, rio_geo – Linha), outro atributo que permita a identificação da forma geográfica a ser visualizada, e esta é a razão do auto-relacionamento existente na classe Atributo.

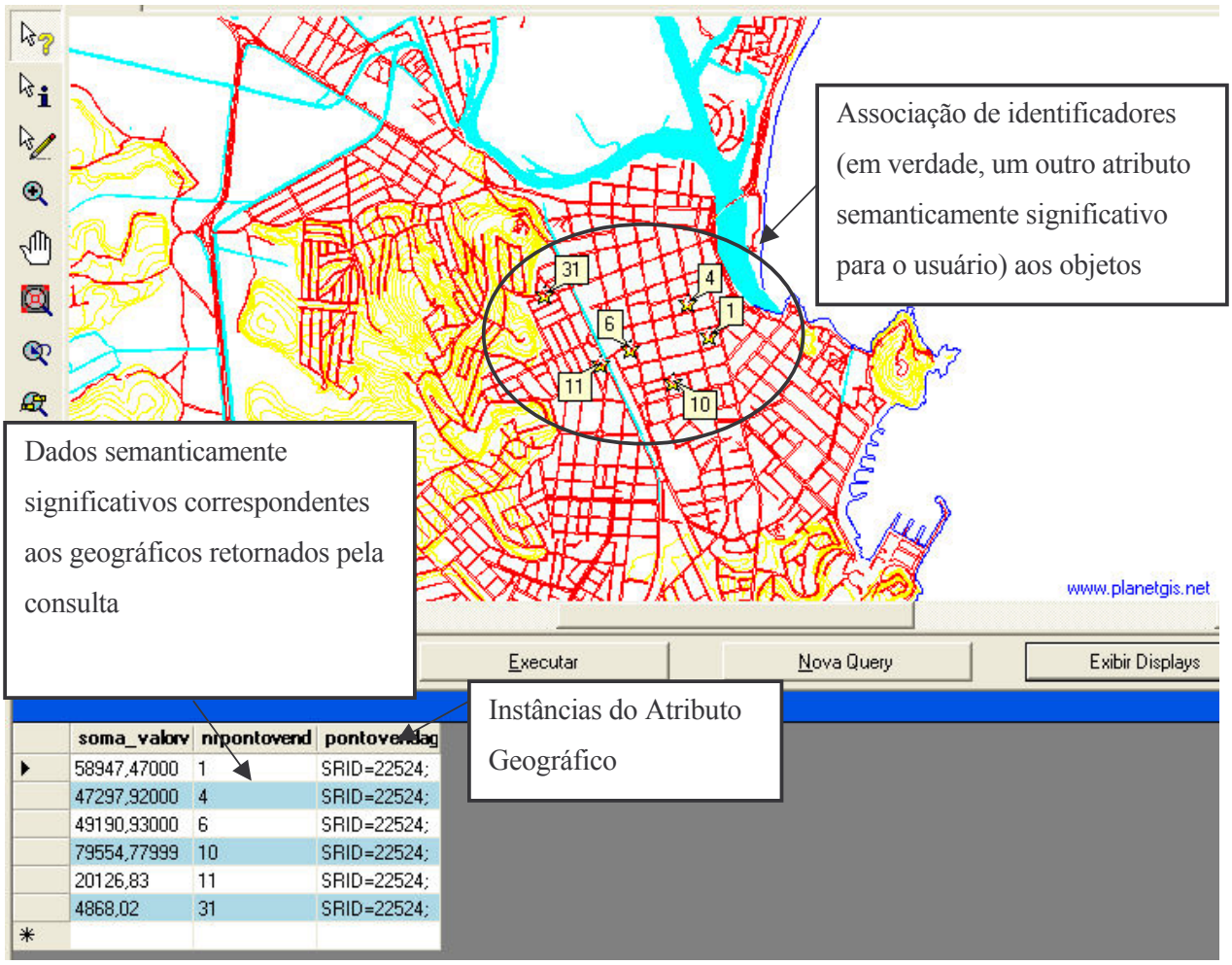


Fig.41 Associação de atributo semanticamente identificador às representações geográficas dos objetos

Para a exibição dos resultados geográficos da consulta, a ferramenta PostGeoOlap necessita de uma ferramenta visual que faça tal representação em um mapa, e este é o papel desempenhado pelo PlanetGIS que, infelizmente, não utiliza os mesmos tipos de dados geográficos propostos pelo OpenGIS, trabalhando somente com: Point (Ponto), Polygon (Polígono), Line (Linha), Text (Texto) e Raster. Por isso, para que o componente de visualização utilizado (PlanetGIS) "desenhe" na tela os objetos geográficos retornados pelo SGBD Espacial, é necessária uma conversão dos tipos de dados retornados pelo SGBD espacial (e aderentes ao OpenGIS) para os utilizados pelo PlanetGIS.

No tocante à visualização dos resultados geográficos, duas abordagens podem ser adotadas: a primeira delas é a de utilizar um arquivo de mapa (arquivo .map, do PlanetGIS)

sem display algum, exibindo em tela apenas os objetos geográficos retornados pelo SGBD Espacial; assim, em uma suposta consulta onde se desejasse a localização e valor médio das notas obtidas no ENEM (Exame Nacional do Ensino Médio) pelas escolas localizadas no bairro 'Centro' e com aproveitamento (ou seja, com grau superior a 5,0), o aplicativo exibiria apenas um conjunto de pontos na tela representativos das escolas, com um nome associado. Não seria possível visualizar nenhum outro elemento, como por exemplo os limites do bairro "Centro", ou os logradouros onde estão localizadas as escolas (a menos que também se incluíssem, explicitamente, tais atributos geográficos na consulta).

Tal abordagem está em perfeita sintonia com os ditames deste trabalho, pois mantém em um único repositório (no SGBD espacial) todos os dados, convencionais e geográficos, necessários à aplicação; mas traz um grande inconveniente: nos casos de consultas que conduzam a mapas muito carregados de informações, a plotagem on-line dos objetos geográficos um a um, torna a resposta do sistema muito demorada. A segunda abordagem é a utilização de um arquivo de mapa (.map, do PlanetGIS) com os displays de interesse (limites dos bairros, logradouros, pontos de referência, etc), a título de "cenário" ou contexto geográfico, o que poupa a inclusão de tais atributos "de contexto" nas consultas e, conseqüentemente, melhora em muito o desempenho das consultas no aplicativo (já que dispensa o SGBD de retornar tais informações e dispensa também o componente de visualização de recebê-los um a um para representação).

As abordagens para visualização anteriormente comentadas são meras decisões de projeto/implementação com vistas a obter ganhos de desempenho, em nada modificando o modelo conceitual de integração analítico-geográfica proposto no presente trabalho. Desta forma, aproveitando o exemplo anterior de consulta, para a visualização dos logradouros nos quais se localizavam as escolas, o fato do usuário explicitar tal atributo geográfico em sua consulta ou encontrar tal camada de informação já representada em um display no PlanetGIS, não dispensaria a existência, no modelo conceitual, das informações relativas aos logradouros em questão.

Na implementação da ferramenta PostGeoOlap, foi utilizada, por questões de desempenho, a abordagem de manter em um arquivo .map (arquivo do PlanetGIS) os displays (ou camadas temáticas) de interesse e buscar os objetos geográficos por meio de algum atributo de associação semântica que também seja chave no SGBD. A associação de um atributo semanticamente significativo a um atributo geográfico foi feita por meio de uma nova tabela de metadados denominada "mapeamentogeo". Tal decisão de projeto visou simplificar

a inclusão, por motivos tecnológicos, do conceito de "feature" (camada) no esquema de mapeamento, ou seja, para dada instância de atributo geográfico retornada pelo SGBD espacial, a aplicação busca, na "feature" indicada, o objeto correspondente e o exibe em tela.

A abordagem da utilização de um arquivo .map para manter os dados de "contexto" geográfico, apesar de duplicar o armazenamento dos objetos geográficos, de forma alguma inviabiliza a proposta do presente trabalho, visto que nenhuma funcionalidade geográfica é requerida do citado componente, ficando a cargo do SGBD Espacial todo o processamento integrado (analítico + geográfico) necessário ao funcionamento da aplicação desenvolvida. Importante ainda ressaltar (de forma a mostrar que a escolha da abordagem é mera decisão de projeto e função apenas do tipo de componente de visualização utilizado, sem frustrar a idéia central de integração deste trabalho) que, segundo contatos com os desenvolvedores do componente PlanetGIS, a futura versão 3.0 do componente tornaria desnecessária a abordagem de duplicação do armazenamento das entidades geográficas, uma vez que nela será possível armazenar todas as entidades geográficas (com bom desempenho) diretamente em um SGBD por meio do uso do formato WKB (Well-Known Binary).

Ainda sobre componente de visualização, a abordagem de manter o citado "contexto geográfico" no arquivo .map traz também outra vantagem: no arquivo .map estão também metadados necessários à visualização dos objetos, como por exemplo, a cor na qual serão exibidos os polígonos da camada "Bairro", ou as linhas da camada "Rio". Para os futuros aperfeiçoamentos da ferramenta PostGeoOlap, no caso do emprego de um único repositório para os dados geográficos, ou seja, a não utilização de um arquivo de apoio (para o "contexto geográfico") pertencente ao componente de visualização, o projeto deverá acrescentar em seus metadados atributos com tal finalidade (provavelmente por meio da adição de novas propriedades à classe ATRIBUTO).

5.4 DIAGRAMAS DE COMPONENTES E DISTRIBUIÇÃO PARA A FERRAMENTA POSTGEOOLAP

O Diagrama de Componentes da ferramenta PostGeoOlap representa a aplicação, escrita em VB.Net, utilizando-se de um componente ActiveX denominado PlanetGIS (ambos compilados em um mesmo executável para a plataforma .Net). A ferramenta PostGeoOlap opera apenas sobre o SGBD PostGreSQL acrescido de sua extensão espacial, o PostGIS, o que explica a representação, no diagrama, dos dois citados componentes (PostGreSQL e PostGIS).

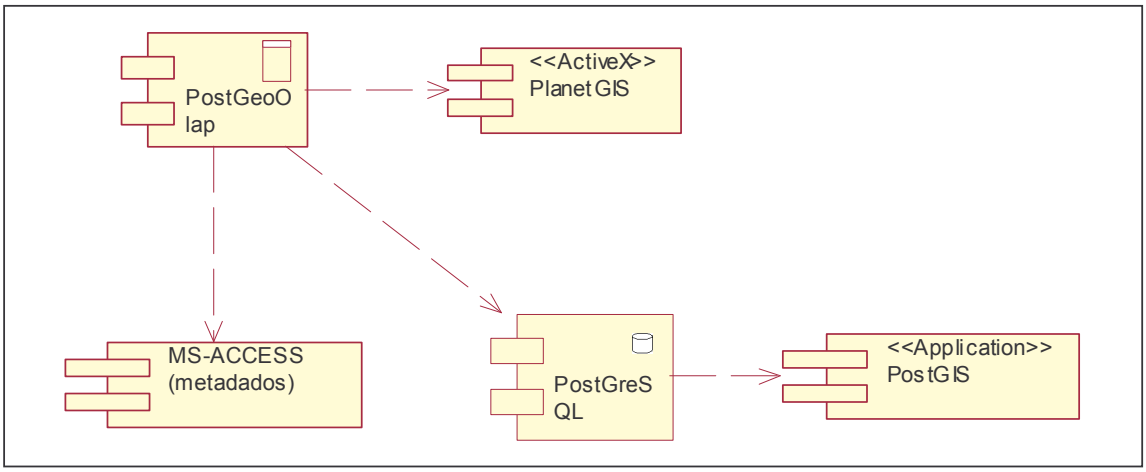


Fig.42 Diagrama de Componentes – PostGeoOlap

O Diagrama de Distribuição representa a arquitetura empregada nos testes com a ferramenta, ou seja, um computador cliente executando a aplicação PostGeoOlap (e acessando em uma base local os seus metadados em MS-ACCESS) e conectado a uma outra máquina com o SGBD Espacial.

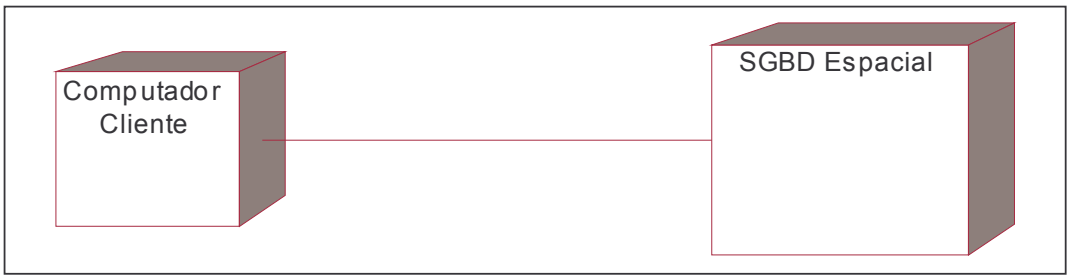


Fig.43 Diagrama de Distribuição – PostGeoOlap

6. ESTUDO DE CASO – DISTRIBUIDORA DE PUBLICAÇÕES

Neste capítulo será apresentado um caso de aplicação do modelo de integração e da ferramenta propostos no presente trabalho, com vistas a produzir "do zero" um novo sistema de suporte à decisão que integre funcionalidades analíticas e geográficas, desde o nível mais alto de concepção, destinado a atender uma empresa distribuidora de publicações real, localizada na cidade de Macaé-RJ, aqui ficticiamente denominada de "SM Distribuidora".

6.1 INTRODUÇÃO

A "SM Distribuidora" é uma organização que adquire, segundo diversas formas de comercialização, produtos (em sua grande maioria – publicações: jornais e revistas) de fornecedores/editores e os distribui, segundo as mesmas ou outras formas de comercialização, a seus pontos de venda – bancas de jornal, lojas de conveniência, hospitais, postos de gasolina, revistarias, etc - para comercialização aos consumidores finais.

Diferentemente da área de entrega de produtos ou transporte, onde a única preocupação do gerente é o translado da mercadoria na quantidade contratada pelo cliente para seu destino final, a atividade de distribuição pressupõe um compromisso do distribuidor com o desempenho de seu cliente, o Ponto de Venda. É o distribuidor quem decide qual a

quantidade e qual o tipo de produto que deve ser enviado a quais pontos de venda, sempre em busca de uma melhor qualidade na distribuição, até porque tal qualidade é fruto de constantes avaliações de performance por parte de seus fornecedores.

A "SM" trabalha com a distribuição de jornais, revistas e cartões telefônicos, o que perfaz uma média de 2100 diferentes títulos de publicações/produtos sendo distribuídos para 150 pontos de venda localizados em quatro municípios, atendendo uma população aproximada de 180.000 habitantes. Para que fiquem claras as necessidades de informação que nortearam o design do sistema de suporte à decisão da distribuidora, segue uma explicação sucinta das regras de funcionamento de cada linha de produto citada.

Na atividade de distribuição de jornais, a SM recebe seus produtos dos fornecedores (as editoras) sem pagar nada por isso e a um preço já previamente definido (impresso na capa do jornal) pelo próprio editor; os jornais são então distribuídos aos pontos de venda normalmente segundo a mesma forma de comercialização (Consignação Total) para venda ao longo daquele dia. Importante salientar o "prazo de validade" do tipo de produto Jornal, que por ser diário, deve estar nos pontos de venda o mais cedo possível para atender seu maior grupo de consumidores: os trabalhadores em seu caminho matinal para o serviço.

Findo o dia, o jornal é devolvido, sua venda calculada e cobrada de acordo com o desconto que cada ponto de venda tenha no preço final de capa do produto. Neste tipo de negócio, a não necessidade de investimento dos pontos de venda na compra do produto para comercialização, dá ao distribuidor uma outra forma de se posicionar: é ele quem decidirá o quanto e quais jornais distribuir a cada ponto (de acordo com potencial de venda, venda anterior, acontecimentos recentes, dia da semana e até condições climáticas) e também é o distribuidor quem deve negociar junto a seus fornecedores a quantidade de produto desejada de forma a atender seus pontos de acordo com a estratégia de distribuição.

A atividade de distribuição de revistas possui similaridades com a de jornais: a forma de comercialização é a consignação total e a receita obtida com a venda das publicações é obtida pela aplicação de descontos ao preço de capa (que também é pré-definido pelo editor), mas possui algumas diferenças relevantes: existe uma quantidade muito maior de títulos de revistas, são voltadas para diversos segmentos (informática, comportamento feminino, culinária, masculinas, pornográficas, etc) e ainda o prazo de permanência dos produtos nos pontos de venda pode variar entre uma semana até um ano.

Assim, depois que a distribuidora entrega seus produtos aos pontos de venda, ela aguarda um pedido de devolução (dita "Chamada de Encalhe") dos fornecedores, onde estão

listados quais produtos devem ser devolvidos e até quando, e repassa tal procedimento operacional de devolução de produtos a seus pontos de venda; a partir da devolução, são computadas as vendas de cada revista e calculados os valores a serem pagos de acordo com o desconto atribuído a cada Ponto de venda.

Interessante ainda mencionar a existência de devoluções parciais, ou seja, o produto é devolvido pelos pontos de venda à distribuidora, esta calcula as vendas e redistribui o montante devolvido como nova distribuição, ajustando desta forma as demandas nos pontos com a distribuição (além de poder com isso, recolher parte dos valores vendidos que estão de posse dos vendedores).

Para a distribuição de cartões telefônicos, a SM adquire seus produtos (cartões) a preço de custo das operadoras de telefonia e os distribui, através de revenda, aos pontos de venda final ao consumidor por um valor de venda qualquer definido pela distribuidora; nesta espécie de negócio, a quantidade e variedade de produtos nos pontos não podem ser impostas pelo distribuidor (embora possam ser trabalhadas através de negociações outras como descontos, formas facilitadas de pagamento e programas de incentivo à aquisição), já que os pontos de venda devem pagar para adquirir os cartões. Este tipo de comercialização, onde não se admite a devolução do produto não comercializado, é denominado venda.

Por meio de uma análise mais detalhada das linhas de produtos distribuídas pela SM: jornais, revistas e cartões telefônicos, percebemos de início a existência de especificidades: tipos de produtos, formas de pagamento, fornecedores, procedimentos de controle, prazos de devolução e formas de comercialização distintas, peculiares a cada tipo de negócio; mas percebemos também se tratarem de uma grande família de negócios similares, com um núcleo de características em comum: o relacionamento com fornecedores, a necessidade de conhecimento acerca de seus produtos e ainda de sua área de atuação geográfica (de forma a bem identificar o perfil de consumo das diversas sub-regiões atendidas), com a finalidade de maximizar as vendas dos produtos nos pontos finais de venda aos consumidores.

Para guiar esforços como por exemplo ações promocionais e alavancar os resultados de venda das diversas linhas de produtos, é imprescindível à SM um bom sistema de suporte à decisão que auxilie, por exemplo, nos seguintes questionamentos: onde abrir ou incentivar a abertura de novos pontos de venda? Quais produtos enviar para cada local em função de seu perfil? Qual a categoria de produtos que mais vende nos pontos localizados no Bairro "Centro"? Ou qual a Categoria de produtos que pode deixar de ser enviada aos pontos de

venda próximos à Escola "ABC"? Que quantidade de produto adquirir para atender a certa região em determinada época do ano?

Para auxiliar os decisores da SM distribuidora, o sistema de suporte à decisão deve contemplar informações agregadas sobre as Vendas sob as perspectivas do Tempo, dos Pontos de Venda (sua localização, vizinhança, capacidade, etc), dos Fornecedores, dos Produtos e suas características. Quanto à dimensão Ponto de Venda, alguns questionamentos como: "Qual a influência exercida pela proximidade deste PDV com outros pontos de referência do município, como por exemplo Escolas, Rodoviária, Zonas Comerciais, Zonas residenciais, etc?" ou "Qual a relação entre a venda de cada ponto e a densidade populacional do bairro no qual está inserido?", denotam que tal sistema de suporte à decisão, além de possuir capacidades analíticas, deverá também contemplar funcionalidades geográficas.

6.2 DIAGRAMA DE CLASSES DO SISTEMA DE SUPORTE À DECISÃO DA SM DISTRIBUIDORA

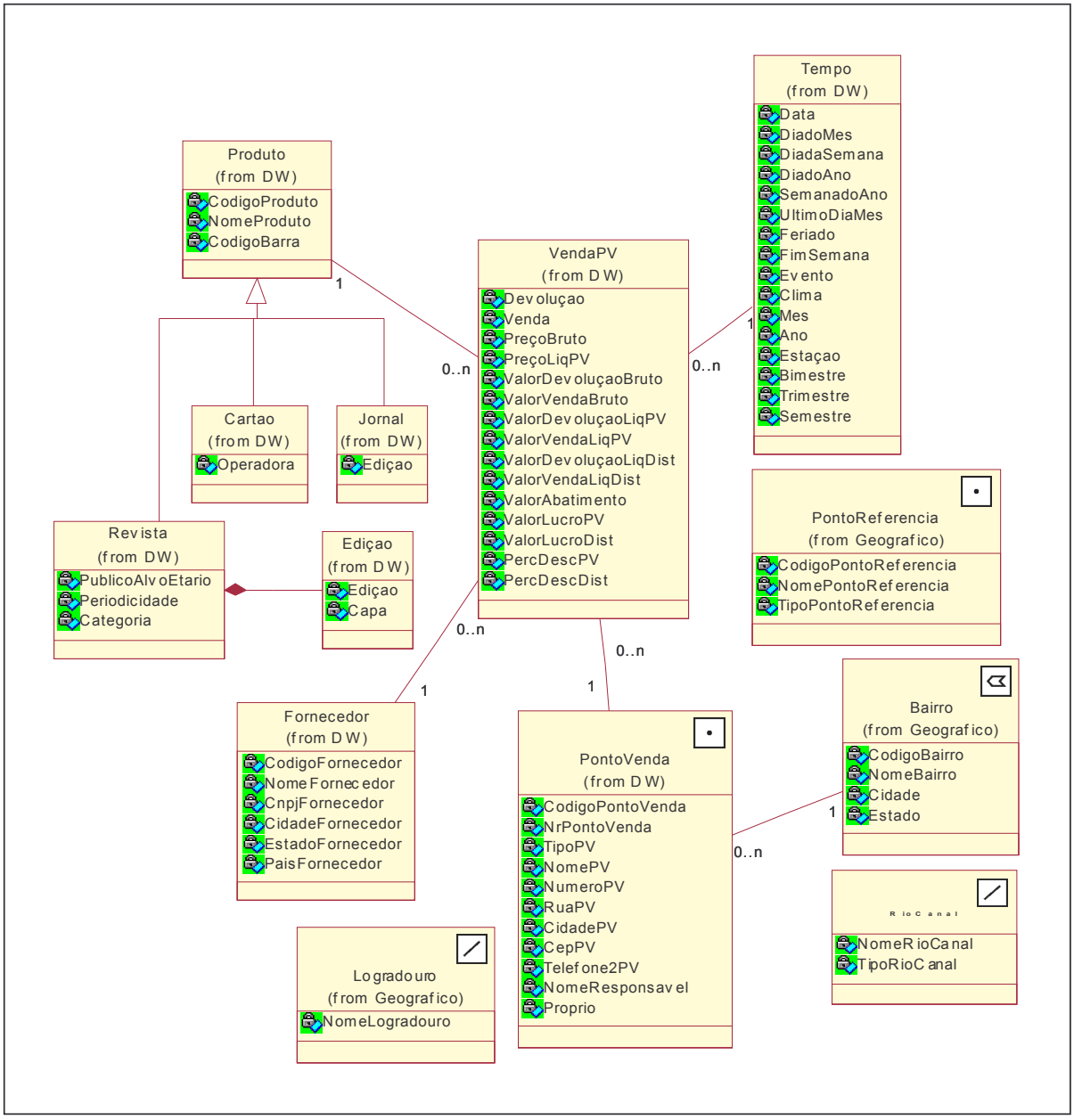


Fig.44 Diagrama de Classes do DW da Distribuidora SM

Apesar de na Distribuidora SM podermos identificar vários Data Marts (análise das vendas dos PDV, para análise das devoluções, etc) para fins do estudo de caso deste trabalho, apenas um deles será utilizado: aquele destinado à análise das vendas dos Pontos, e que dará origem ao Cubo Venda.

A primeira e óbvia vantagem do emprego da Orientação a Objetos para a modelagem do data warehouse da SM foi o reaproveitamento dos objetos já existentes na organização como, por exemplo, Produto, Revista e Edição, o que simplificou a compreensão dos conceitos necessários à implementação do modelo de classes.

Na modelagem das classes geográficas, duas abordagens poderiam ser adotadas: na primeira, seriam criados relacionamentos convencionais (associação, agregação, etc) entre duas classes, e tal abordagem conduziria a um mapeamento dimensional nos moldes do abordado no capítulo 4 do presente trabalho, ou seja, haveria a migração de todos os atributos (inclusive o geográfico) de uma classe para a outra. Em uma segunda abordagem, considerando que todas as classes geográficas já possuem um relacionamento indireto intrínseco umas com as outras, através do referencial geográfico, seria possível manter classes geográficas relacionadas sem o estabelecimento de um relacionamento explícito entre elas. Desta forma, o relacionamento entre as classes geográficas "PontoVenda" (objeto do tipo Ponto) e "Bairro" (objeto do tipo Polígono), onde todo "PontoVenda" está localizado em um e somente um "Bairro", poderia ser implementado através da primeira ou da segunda abordagem, com as seguintes observações: Caso um relacionamento convencional seja estabelecido, os atributos de "Bairro" passariam a integrar a dimensão "PontoVenda" e, conseqüentemente, participariam do processamento do Cubo "Venda" sendo indexados junto com os demais atributos das dimensões, permitindo que consultas sobre dados da tabela Fato abordados sob a perspectiva de "Bairro" pudessem ser respondidas mais rapidamente. A não implementação de um relacionamento convencional qualquer entre as classes "PontoVenda" e "Bairro", embora não impedissem que consultas acerca de dados numéricos da tabela Fato fossem abordados sob a ótica de "Bairro", tornariam tal consulta mais custosa computacionalmente, já que o SGBD teria que, após recuperar os dados relativos aos "PontoVenda", obter, por processamento geográfico (através da função geográfica "geometry_contained", por exemplo), os Bairros que contivessem os "PontoVenda" em questão.

6.3 MAPEAMENTO DO MODELO DE CLASSES PARA O DIMENSIONAL RELACIONAL

A partir do modelo de classes apresentado, foi executado o mapeamento dimensional, de forma a obter o modelo dimensional relacional correspondente para o SGBD PostGreSQL, acrescido do PostGIS (para habilitar espacialmente o SGBD), como mostrado no DED (Diagrama de Estrutura de Dados) a seguir:

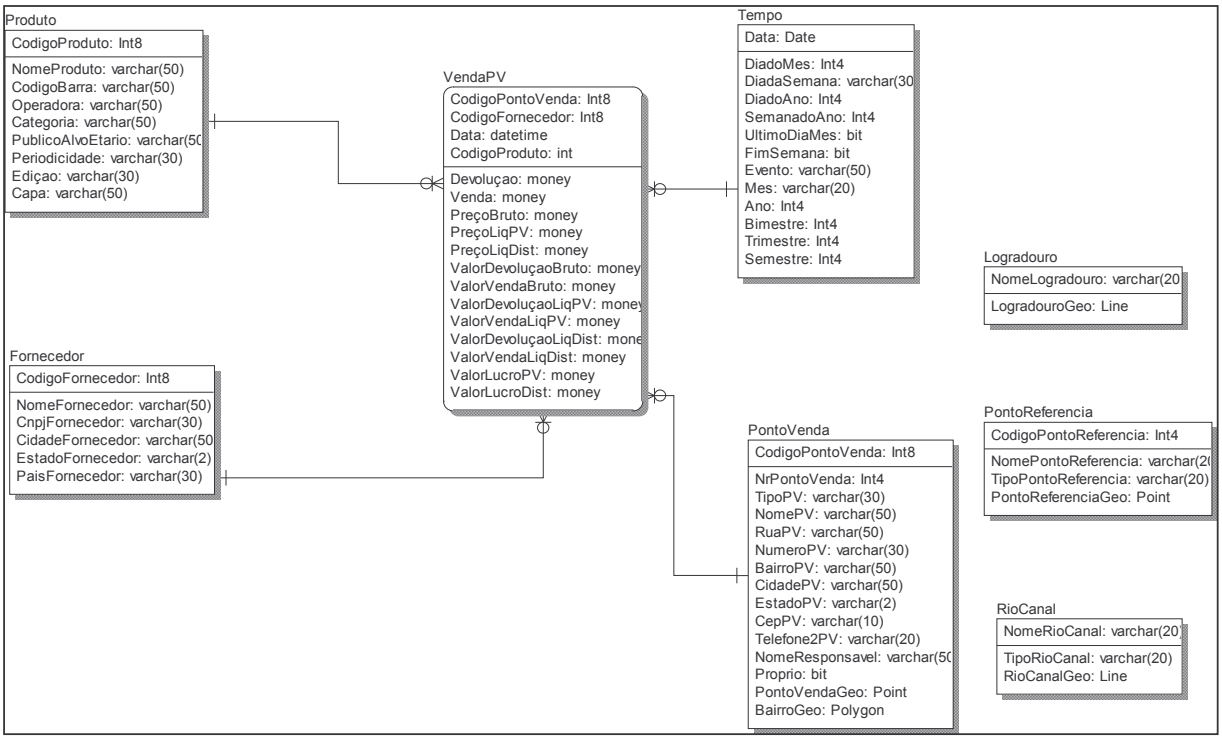


Fig.45 DED Dimensional Relacional para a "Distribuidora SM"

O próximo passo foi a geração dos scripts SQL para a criação das estruturas (tabelas e índices) no PostGreSQL. Tais comandos podem ser executados através da interface "psql" (aplicativo cliente do PostGreSQL, que é operado por meio de linha de comando) ou de alguma interface gráfica, como por exemplo o PgAdmin (atualmente na versão III).

Uma vez que a base de dados da Distribuidora SM estava no MS SQL Server 2000, foi utilizado o "Migration Wizard" do PgAdmin II (um utilitário para cópia de dados de outras bases para o PostGreSQL) para a cópia de dados para o repositório no PostGreSQL. A mesma tarefa de cópia foi tentada através do "Data Transformation Service" do próprio SQL server, por meio de um driver ODBC, mas a primeira abordagem mostrou-se muito mais rápida.

Para a obtenção dos dados geográficos necessários (a "Distribuidora SM" não dispunha de tais dados em sua base de dados) ao Sistema de Suporte à Decisão implementado, foi utilizado o PlanetGIS, um programa para SIG (Sistema de Informações Geográficas) gratuito, bastante flexível e de desempenho superior à média das ferramentas de visualização comerciais (o componente de visualização utilizado na ferramenta OLAP nada mais é que um objeto COM derivado do executável original). Para tanto, a "Distribuidora SM" obteve um arquivo .dxf da região de interesse, com as layers (ou camadas) necessárias (como por exempo: bairros, logradouros e rios) e, através do próprio PlanetGIS, importou tal arquivo .dxf, transformando-o em um arquivo .map (extensão dos arquivos do PlanetGIS) para que, utilizando o já citado PlanetGIS, pudesse capturar, sobre o arquivo em questão, os dados de coordenadas dos Pontos de Venda e Pontos de Referência.

Quando todas as informações geográficas estavam agrupadas em um único repositório, ou seja, no arquivo .map do PlanetGIS, teve início a fase de exportação de cada uma das camadas de interesse para análise (camadas, também denominadas Displays: bairro, logradouro, riocanal, pontovenda, pontoreferencia), em formato .shp (SHAPE FILE) para que um aplicativo do PostGIS (shp2pgsql) pudesse importar tais dados para o SGBD Espacial. Na importação do arquivo Shape para o PostGreSQL, o que ocorre é a transformação dos dados do arquivo .shp em uma nova tabela, que deve ser alterada pelo DBA para que os nomes dos campos fiquem condizentes com a estrutura proposta no DED.

De posse do data warehouse devidamente carregado com dados, utilizou-se a aplicação OLAP para a realização de análise sobre os dados agregados, por meio dos passos descritos na seção seguinte.

6.4 UTILIZAÇÃO DA FERRAMENTA POSTGEOOLAP PARA A PREPARAÇÃO DOS DADOS

6.4.1 Estabelecimento de uma conexão com o SGBD

Fig.46 Conexão com o SGBD

Para o presente estudo de caso, a conexão (Figura 46) foi feita com o PostGreSQL 7.4 + PostGIS 0.8.1 executando em uma máquina Duron 1.3 GHz, com 256 MB de memória Ram e com um HD de 5400 rpm e 40 GBytes de capacidade de armazenamento, utilizando o sistema operacional Conectiva Linux 9. Importante ressaltar que, caso se deseje a instalação de um servidor PostGreSQL em ambiente Windows, deve-se instalar o Cygwin (um emulador de UNIX para o ambiente WINDOWS). Tal camada de emulação compromete o desempenho do SGBD PostGreSql em até 50 % quando comparado ao mesmo Servidor de Banco de Dados instalado em seu ambiente nativo (UNIX, LINUX, BSD, etc).

6.4.2 Criação do Cubo – Definição da tabela Fato e seus itens numéricos

Após a conexão com o SGBD, o próximo passo na utilização da ferramenta PostGeoOlap é a criação do Cubo e definição da tabela FATO e dos itens numéricos, ou seja, aqueles atributos que devem sofrer operações de agregação durante os processamentos do Cubo.

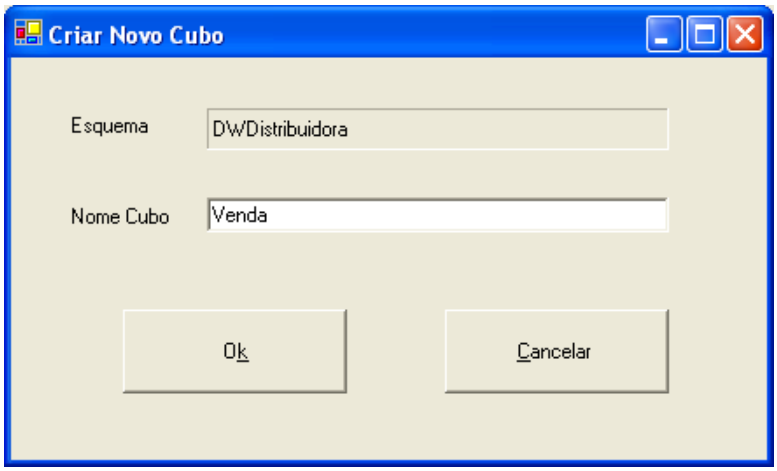


Fig.47 Criação do Cubo

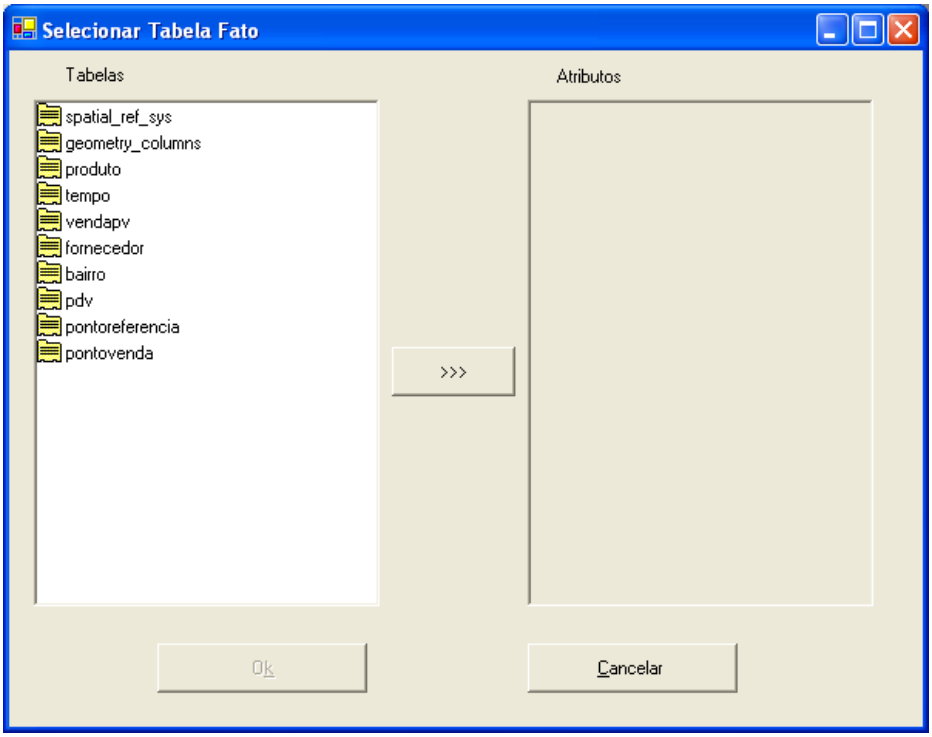


Fig.48 Seleção da tabela Fato

Após selecionar, dentre todas as tabelas existentes no SGBD, aquela que representa a FATO, define-se quais os atributos numéricos a processar e, para tanto, foram escolhidos os atributos que a Distribuidora desejava somar (sobre os quais seria aplicada a operação de agregação SUM) ao longo das demais dimensões de análise (conforme Figura 49).

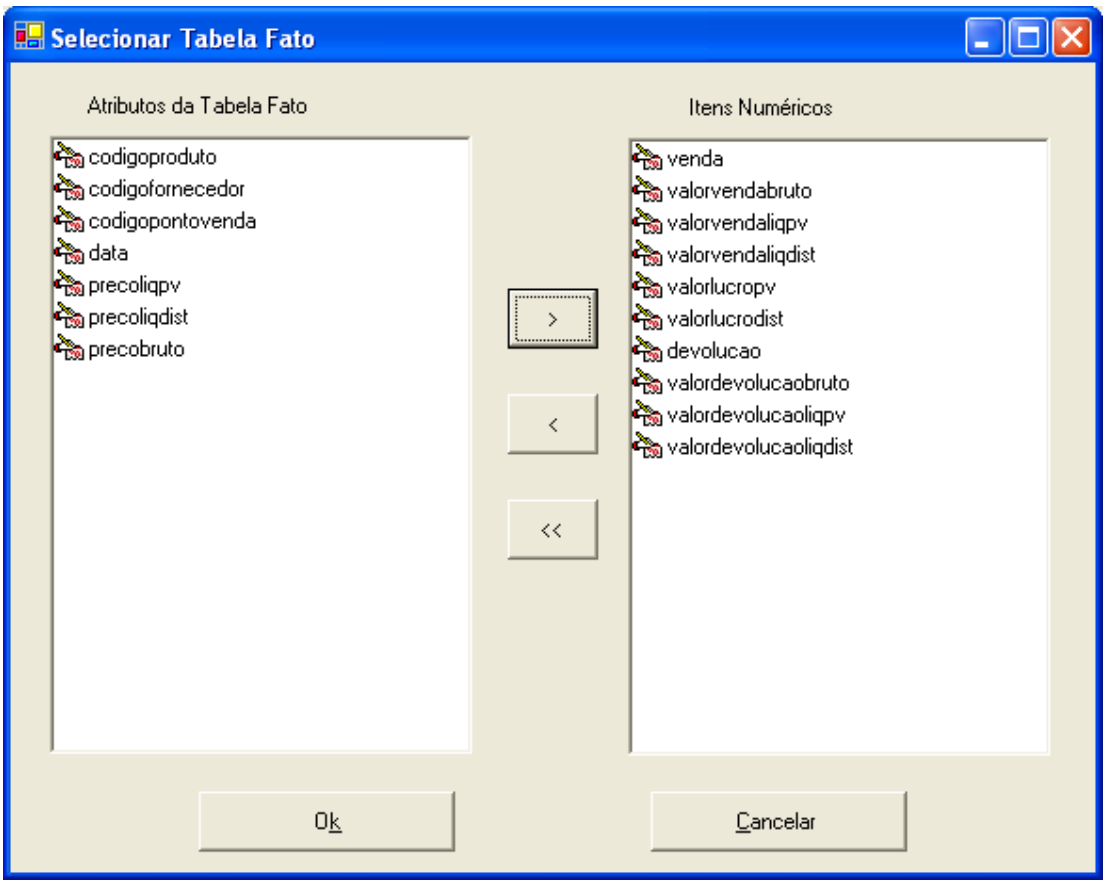


Fig.49 Definição dos itens numéricos da tabela Fato

Após a definição dos itens numéricos, ou seja, daqueles que sofrerão a operação de agregação SOMA (por meio de "Select SUM..") durante o processamento do Cubo, passa-se à definição das tabelas dimensão e dos níveis hierárquicos de seus atributos. A primeira dimensão é a Produto, na qual a hierarquia entre os atributos deve ficar da seguinte forma: dados relativos à edição (codigoproduto, codigobarra, capa, edição) são aqueles de menor granularidade e por isso recebem o maior valor (nove); em um nível acima na escala de agregação, estão os dados relativos ao produto (nomeproduto, operadora, publicoalvoetario, periodicidade), recebendo o nível oito na hierarquia; por último, ocupando o nível mais elevado na escala de agregações, estão os dados acerca da categoria à qual pertencem os

produtos e, assim, recebem o nível sete na hierarquia dos atributos, conforme ilustra a Figura 50.

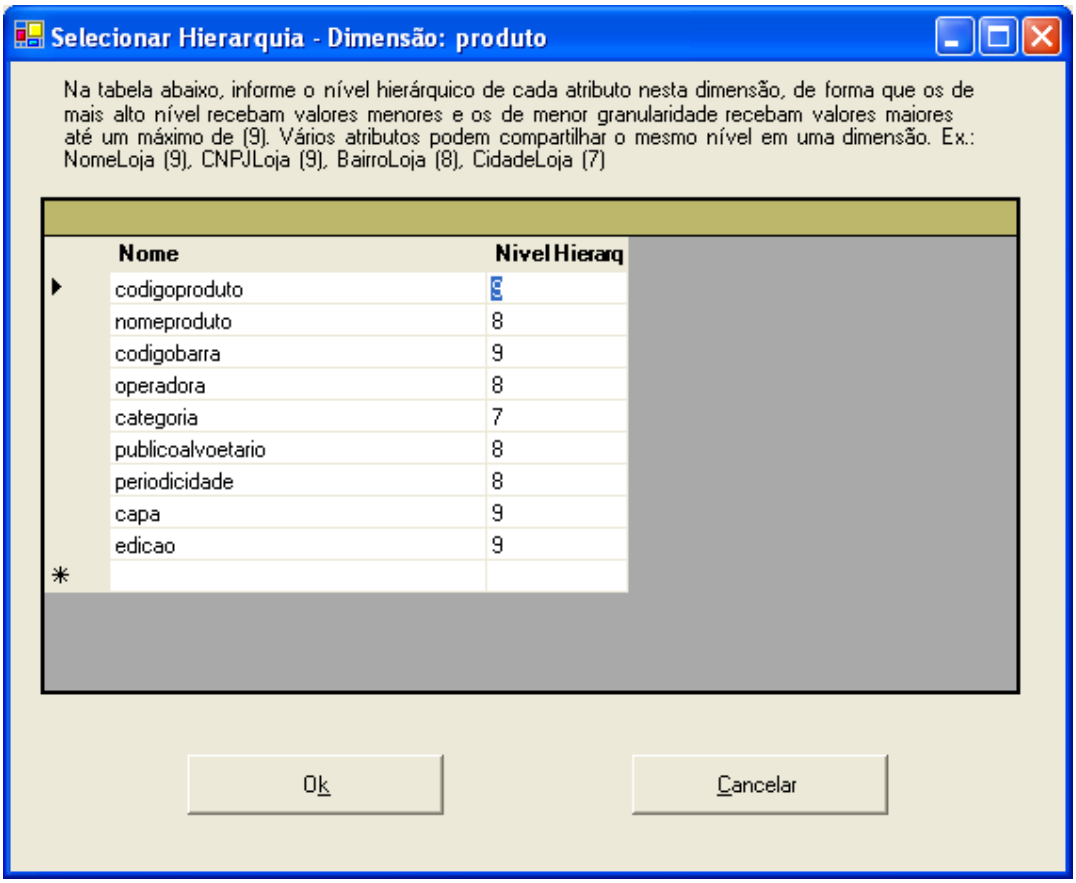


Fig.50 Hierarquia dos atributos na dimensão produto

Na dimensão Tempo (Figura 51), apesar da semântica dos atributos ensejar uma classificação hierárquica com muitos níveis, a hierarquização ficou da seguinte forma: na menor granularidade e com o nível nove, atributos referentes ao dia ou data da venda (data, diadomes, diadasemana, diadoano, ultimodiames, fimsemana, evento); em um nível acima, a semanadoano (que tem grande importância para a distribuidora, para fins de comparativos), com o nível oito e, no nível de agregação mais alto, os dados relativos ao mês (mes, bimestre, trimestre, semestre, ano) ocupando o nível sete.

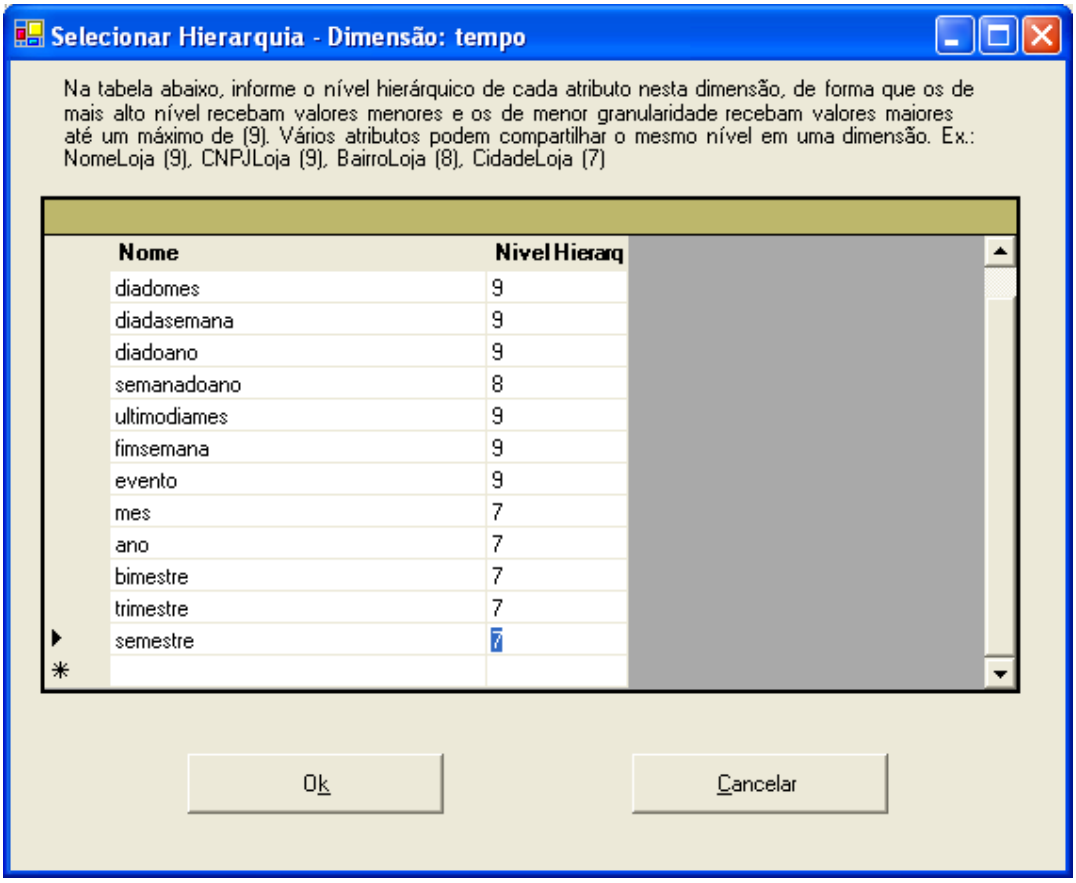


Fig.51 Hierarquia dos atributos da dimensão Tempo

A dimensão Fornecedor (Figura 52) teve seus atributos classificados hierarquicamente da seguinte forma: na menor granularidade, os dados do nome do fornecedor (codigofornecedor, nomefornecedor, cgcfornecedor – todos no nível nove); acima destes, em um nível mais agregado, o nível oito com o atributo cidadefornecedor e, no nível mais alto da dimensão (nível sete), os atributos estadofornecedor e paisfornecedor. Apesar dos nomes dos atributos, a Distribuidora SM não tem interesse na análise geográfica sobre os atributos de fornecedor e, por isso, estes foram tratados como atributos convencionais, destinando-se à simples agregação.

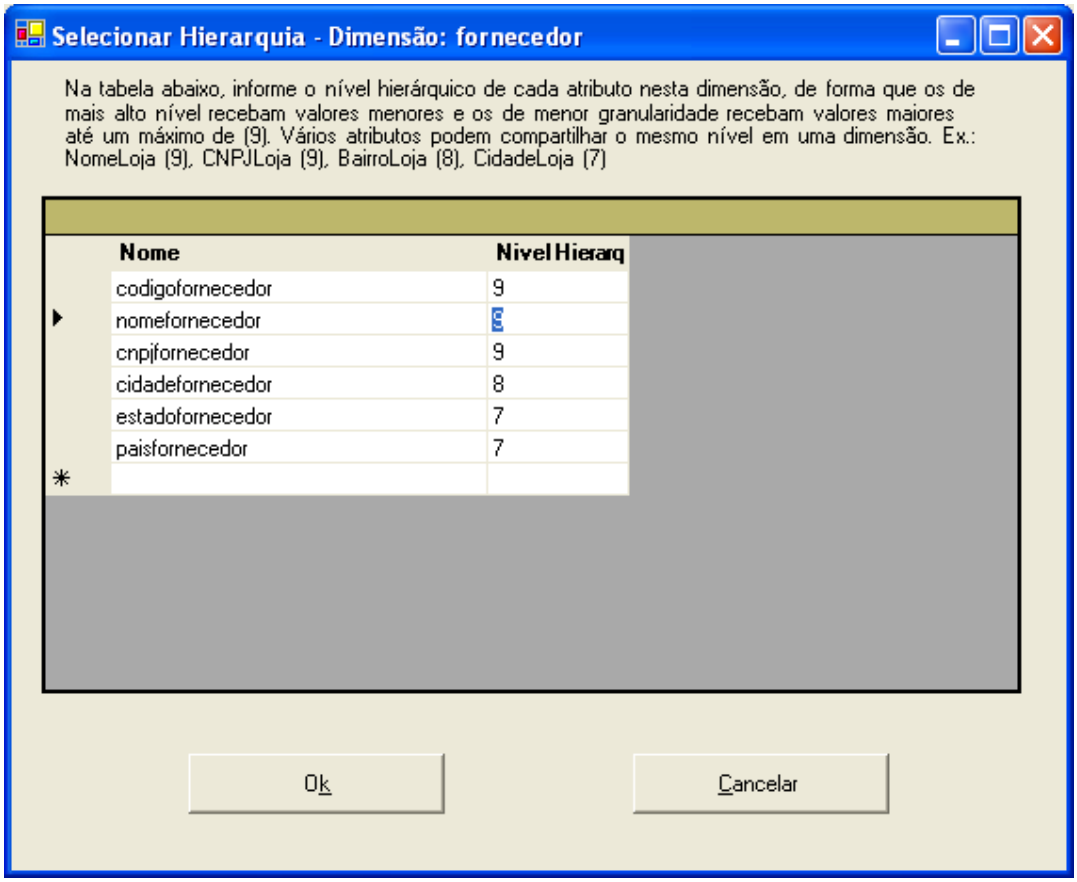


Fig.52 Hierarquia dos atributos da dimensão Fornecedor

Por fim, a dimensão PontoVenda (Figura 53), que foi a única sobre a qual a Distribuidora SM possuía interesses de análise geográfica, teve a classificação hierárquica de seus atributos definida da seguinte forma: no menor nível de granularidade e com o nível 9 (nove), os atributos relativos ao ponto de venda em si (codigopontovenda, nrpontovenda, tipopv, nomepv, ruapv, numeropv, ceppv, telefone2pv, nomeresponsavel, proprio, pontovendageo), em um nível superior de agregação (nível oito), dados referentes ao Bairro onde se localiza o PDV: bairropv, bairrogeo; e no mais alto nível de agregação da dimensão, cidadepv e estadopv (no nível sete).

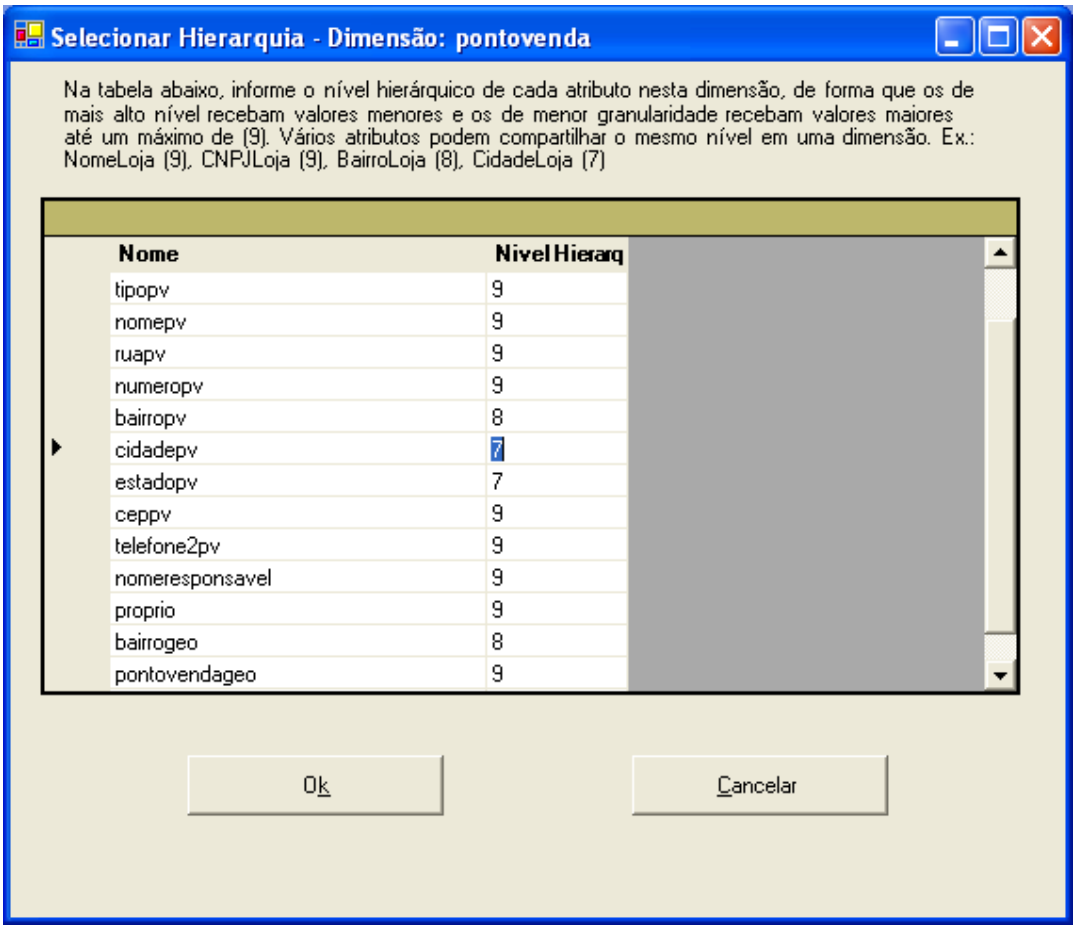


Fig.53 Hierarquia dos atributos da dimensão PontoVenda

6.5 PROCESSAMENTO DO CUBO

Conforme se pode verificar pelo anteriormente exposto, para cada dimensão foram definidos apenas três níveis de agregação, ou seja, para cada dimensão, a aplicação OLAP produzirá, caso necessário (tal necessidade é verificada de acordo com a melhor relação custo-benefício entre desempenho – obtido por meio da criação de novas agregações – e espaço de armazenamento – que será tão menor quanto menos agregações forem criadas), quatro agregações: a primeira agregará os itens numéricos da tabela FATO utilizando dados do próprio nível base (ex.: nomefornecedor), a segunda agrupará por dados de cidadefornecedor, a terceira agregará por estadofornecedor e a terceira fará um único

agrupamento por TODOS os fornecedores (que seria o nível zero da dimensão, ou seja, aquele que agrupa todos os demais).

Após o término das definições das hierarquias dos atributos dentro das dimensões, foi executado o processamento do Cubo, que, em sua pior hipótese (ou seja, na hipótese de haver uma base de dados muito grande e com grande quantidade de distintas instâncias dos atributos integrantes dos diferentes níveis hierárquicos das dimensões), produziria um total de (4 x 4 x 4 x 4) – 1 = 255 agregações, ou seja, cada uma das dimensões contaria com os três níveis definidos pelo usuário acrescido do nível zero, que é o nível "TODOS" da dimensão; a subtração de uma unidade é explicada pelo fato de que o produto do processamento de todas as dimensões em sua granularidade mínima (nível 9 das dimensões Produto, Tempo, Fornecedor e PontoVenda) é exatamente igual ao nível base do data warehouse, e, por isso, não necessita ser recriado (constituindo-se em uma agregação a menos).

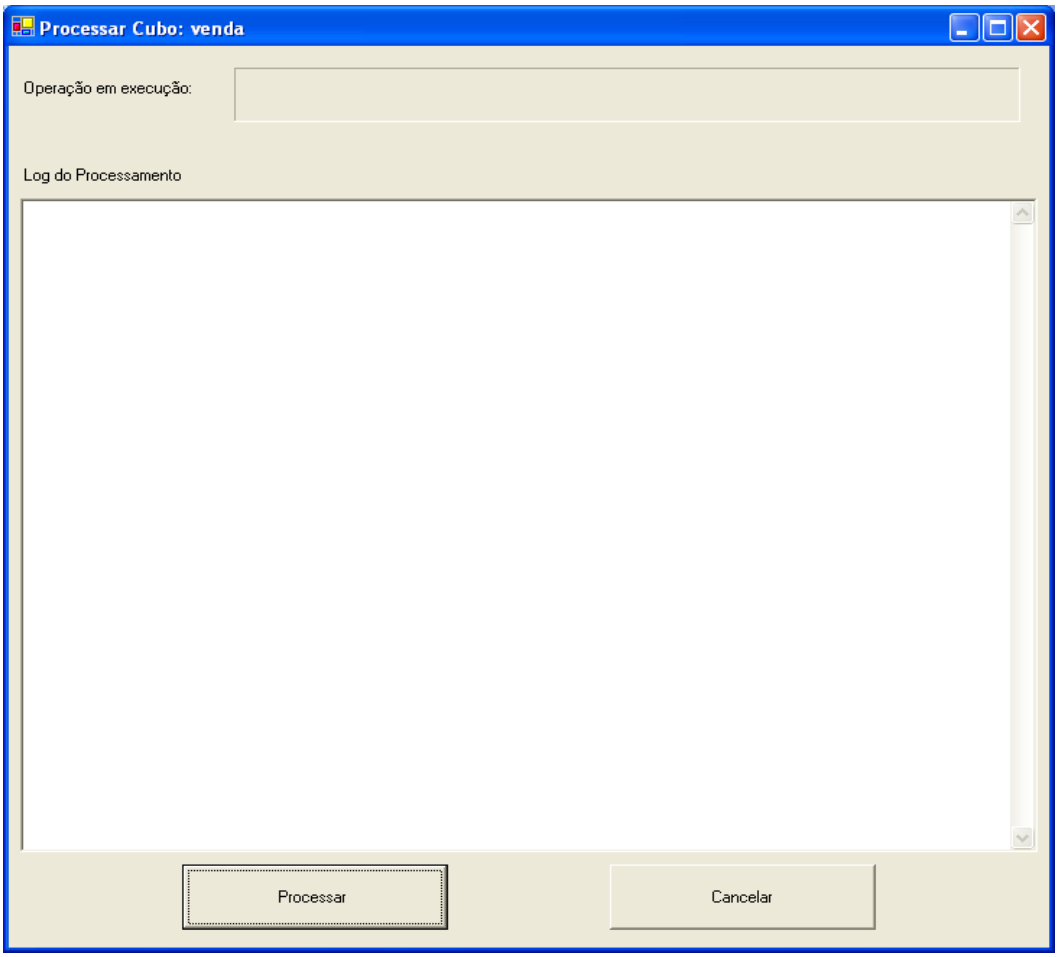


Fig.54 Tela de processamento do Cubo

Para o estudo de caso em questão, foi utilizada uma base de dados com as seguintes características: Tabela FATO: 68577 registros, tabela Fornecedor: 21 registros, tabela Tempo: 59 registros, tabela Produto: 4020 registros e tabela PontoVenda: 88 registros. O processamento (executado na máquina especificada anteriormente neste capítulo) durou 1hora e 28 minutos, gerando 105 agregações.

6.6 SELEÇÃO DAS DIMENSÕES NÃO-AGREGÁVEIS

Conforme mencionado anteriormente, dimensões não-agregáveis são tabelas assim chamadas de dimensões porque, apesar de não serem empregadas como perspectivas para as agregações da tabela FATO, são usadas como perspectivas ou referências para comparações com as dimensões geográficas do Cubo.

A Distribuidora SM tinha a necessidade de analisar a influência de certos pontos de referência (escolas de primeiro e segundo graus, faculdades, zonas comerciais, bancos, rodoviária, etc) sobre a venda de seus PDVs, e tal influência só poderia ser mensurada se o sistema de suporte à decisão tivesse a capacidade de fornecer dados agregados de venda em função de uma dada distância do ponto de venda em relação ao ponto de referência.

Assim, percebe-se que pontoreferencia é uma tabela que deve ser selecionada como uma dimensão não-agregável, ou seja, uma dimensão com dados geográficos cujo único propósito é o de servir como referência para as comparações a partir da dimensão pontovenda (a única dimensão geográfica do Cubo "Venda"). A necessidade de referenciais para comparação com a dimensão geográfica pontovenda, foi a razão que levou à definição das tabelas riocanal e logradouro como dimensões não-agregáveis.

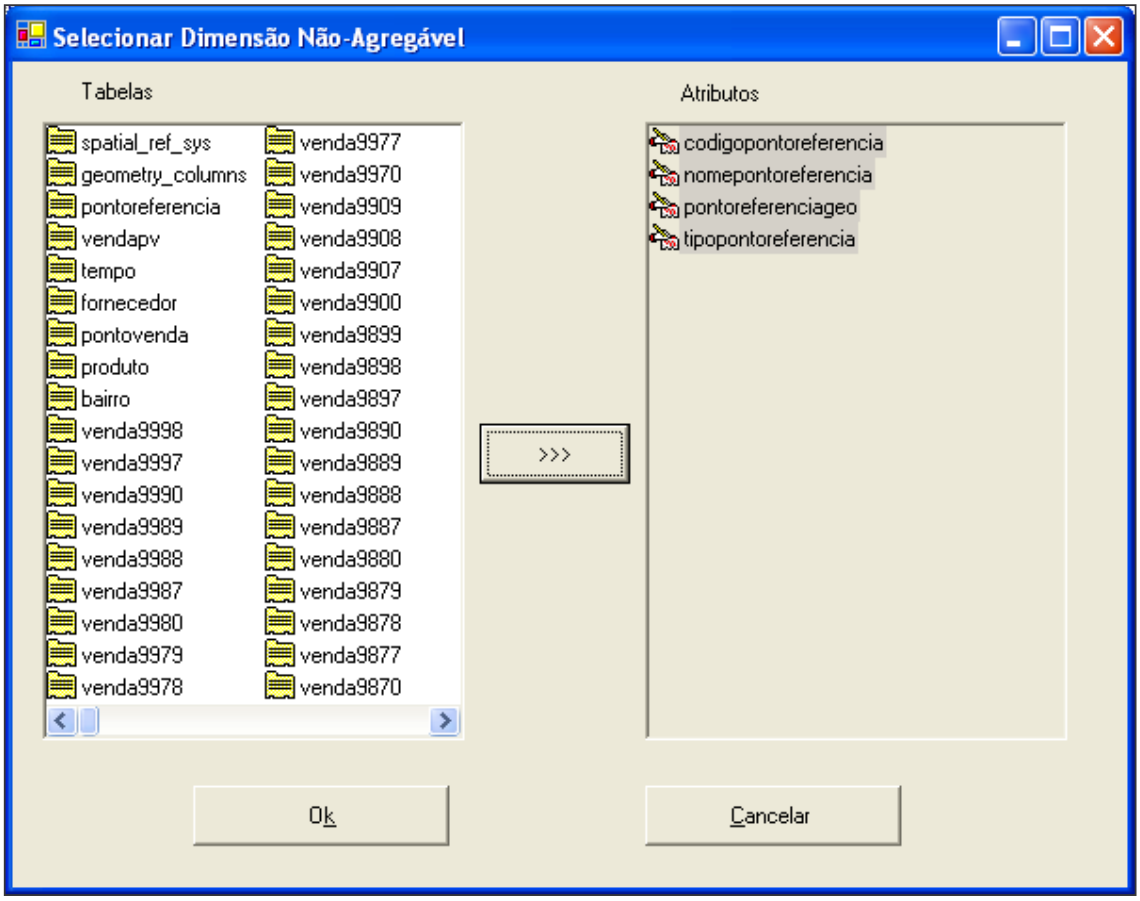


Fig.55 Seleção de Dimensão não-agregável (pontoreferencia)

6.7 ANÁLISE DOS DADOS

Uma vez o Cubo de posse de suas dimensões agregáveis e não agregáveis e executado o seu processamento, o usuário pode abrir a tela de Análise dos dados para a submissão de consultas on-line convencionais e geográficas à aplicação PostGeoOlap.

A primeira das consultas submetidas à aplicação é uma consulta eminentemente convencional, ou seja, sem atributos geográficos:

Q1: “Qual o total de vendas (em valores brutos e líquidos) dos produtos da categoria “Informativas” ocorridos em Janeiro de 2002”

Para a execução da consulta solicitada, o usuário seleciona da tabela Fato os itens valorvendabruto e valorvendaliqpv, da dimensão Produto o item Categoria (sobre o qual será aplicado uma restrição – a categoria deve ser igual a “Informativas”), e da dimensão Tempo os itens mes (restrição: igual a “Janeiro”) e ano (restrição: igual a “2002”).

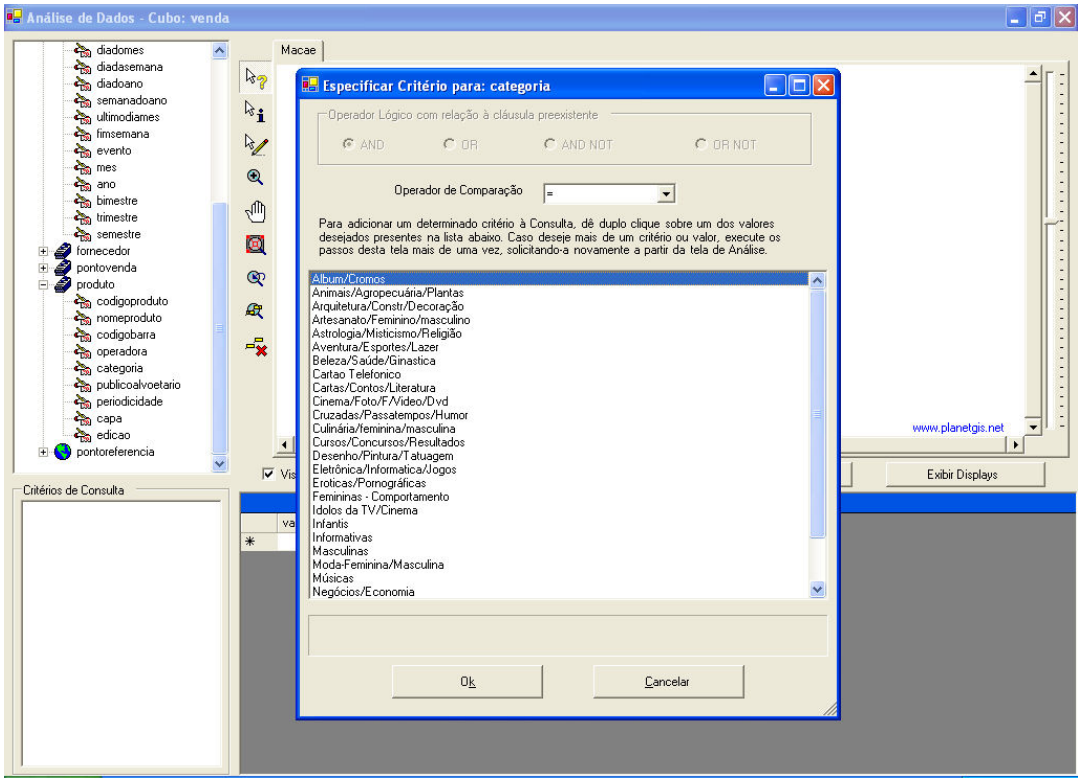


Fig.56 Criação de restrição para o atributo “categoria” da dimensão “produto”

Após a criação de todas as restrições sobre os atributos categoria, mes e ano, o botão “Executar” pode ser clicado para que se obtenha os resultados na grid.

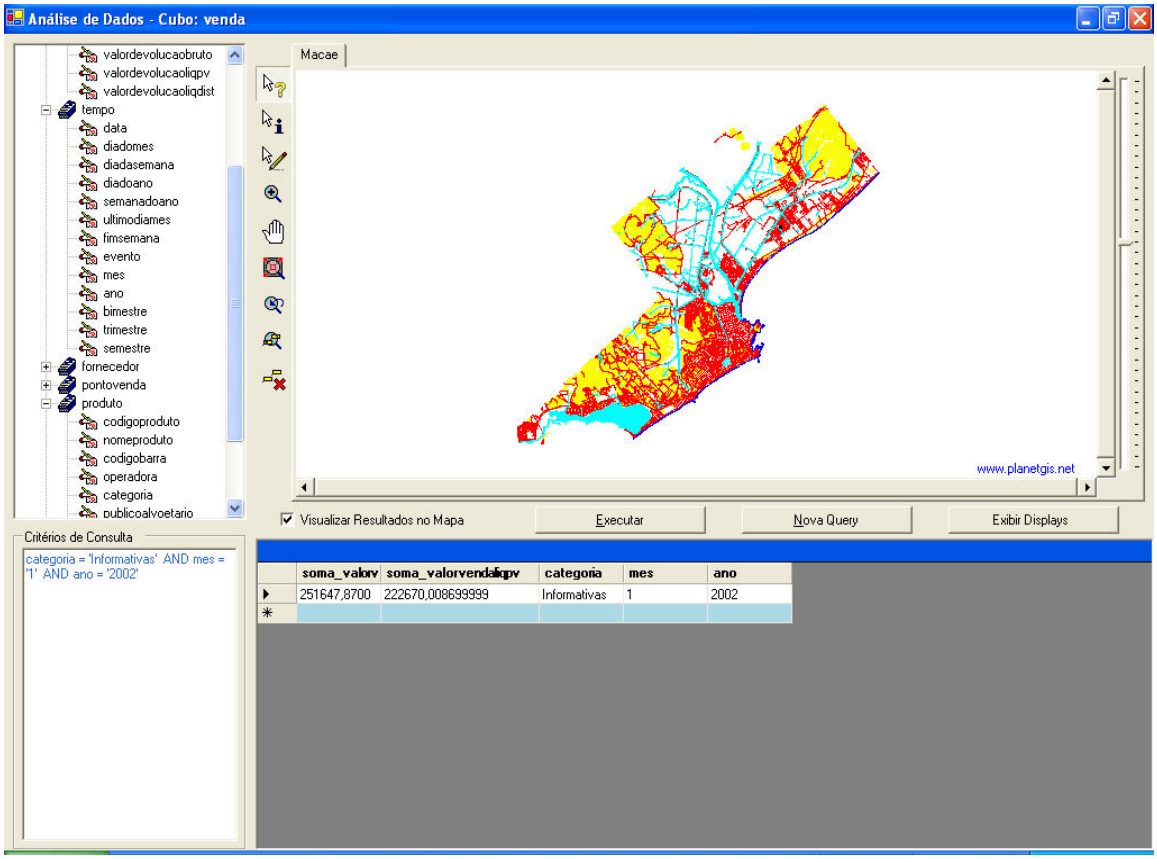


Fig.57 Resultados da consulta "convencional" exibido na grid

Como para a consulta Q1 nenhum atributo do tipo geográfico foi especificado, não houve apresentação de resultados gráficos no mapa da ferramenta de visualização, ou seja, a ferramenta PostGeoOlap comportou-se como uma aplicação OLAP convencional, que busca em suas agregações pré-processadas os resultados para as consultas montadas em tempo real pelo usuário. Ou seja, para o conjunto de atributos solicitado pelo usuário, a aplicação checa, em seus metadados, qual a agregação de mais alto nível (ou seja, a de maior agregação dos dados, onde a quantidade de registros é menor e que, por isso, exige menor esforço computacional para recuperação) que contemple tal conjunto, e sobre esta agregação a consulta é executada, retornando os registros em uma grid.

A segunda consulta submetida à aplicação envolve atributos convencionais e geográficos que buscam atender necessidades de informação reais da Distribuidora SM como, por exemplo, saber da viabilidade ou não da abertura de um ponto de venda alternativo dentro da "Faculdade Fafima" destinado à comercialização de material de pesquisa escolar.

Q2: “Quais os valores de venda bruta (valorvendabruto) e de lucro para a distribuidora (valorlucrodist) obtidos com a venda de produtos da categoria “Pesquisa Escolar” ao longo do ano de 2002, nos pontos de venda próximos (por próximo entenda-se uma distância de até 200 metros) ao ponto de referência “Faculdade Fafima”?”

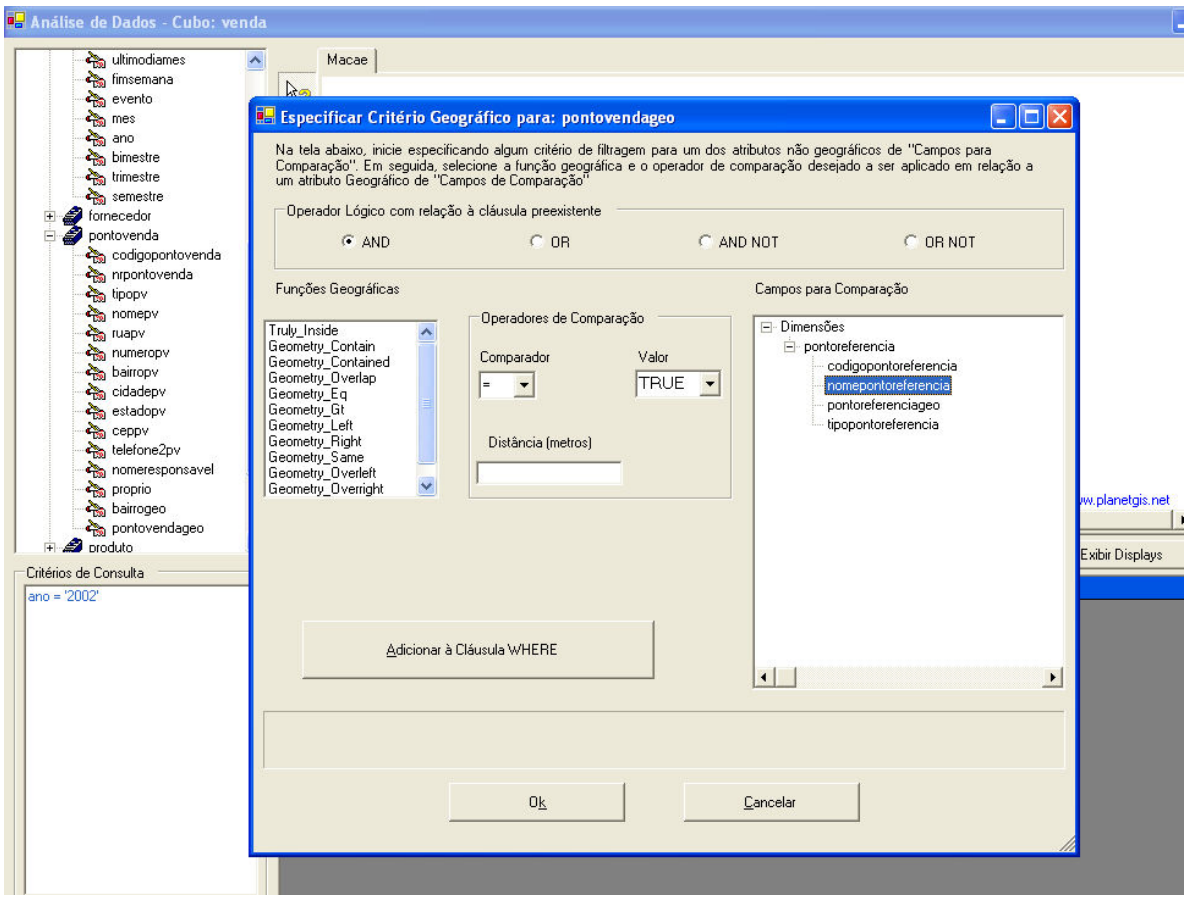


Fig.58 Criação de restrição sobre o atributo geográfico “pontovendageo” (dimensão “pontovenda”), definindo sua distância para o também geográfico atributo “pontoreferenciageo” (dimensão não-agregável “pontoreferencia”)

A restrição geográfica: “...nos Pontos de Venda próximos (por próximo entenda-se uma distância de até 200 metros) ao ponto de referência “Faculdade Fafima””, será aplicada ao atributo “pontovendageo”, (que é o atributo geográfico da dimensão “pontovenda”) buscando aqueles pontos de venda cuja distância seja inferior ou igual ao limite definido de 200 metros em relação a um ponto de referência (tal comparação deverá ser feita com o atributo geográfico “pontoreferenciageo”) de nome igual a “Faculdade Fafima” (a seleção de tal nome para o atributo “nomepontoreferencia” também será fruto da criação de uma restrição do tipo convencional – não geográfica).

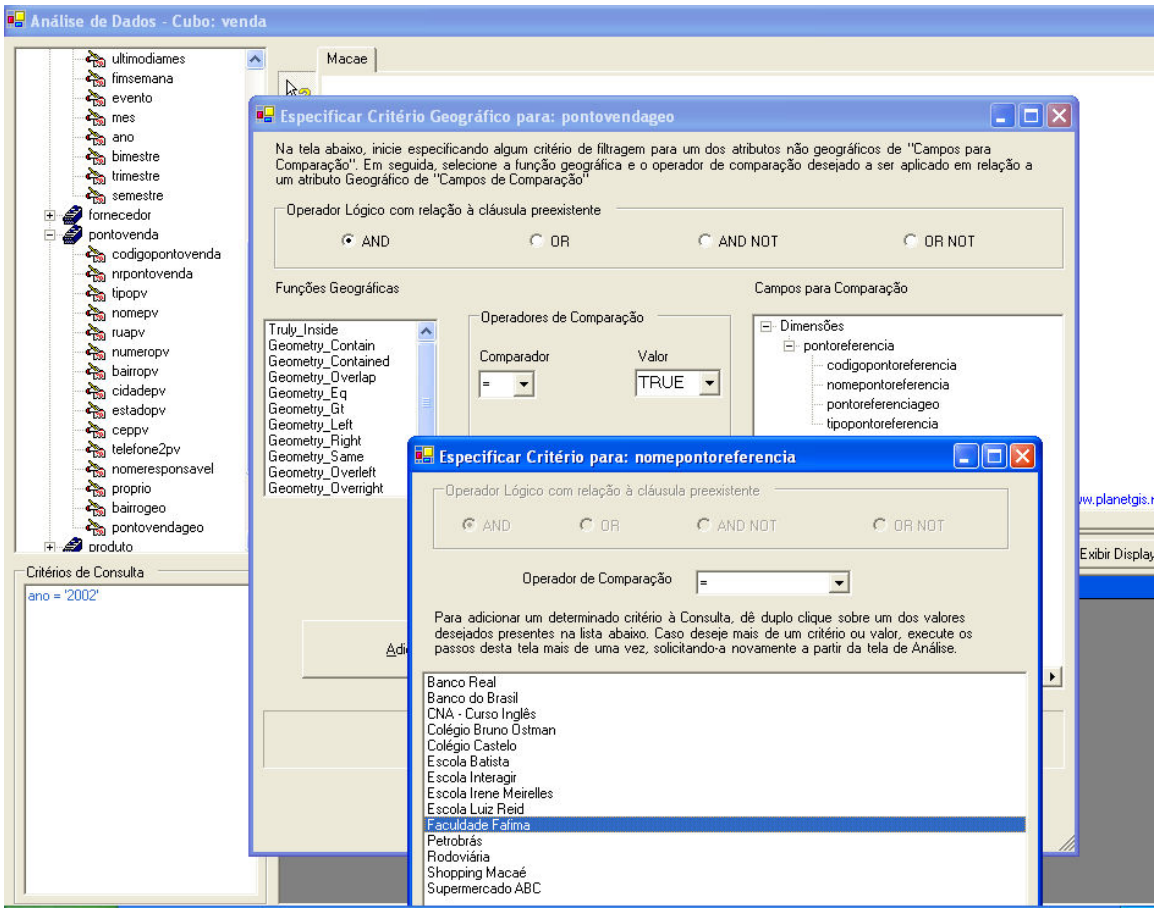


Fig.59 Definição do ponto de referência (“nomepontoreferencia”) sobre o qual se deseja estabelecer a distância para os Pontos de Venda

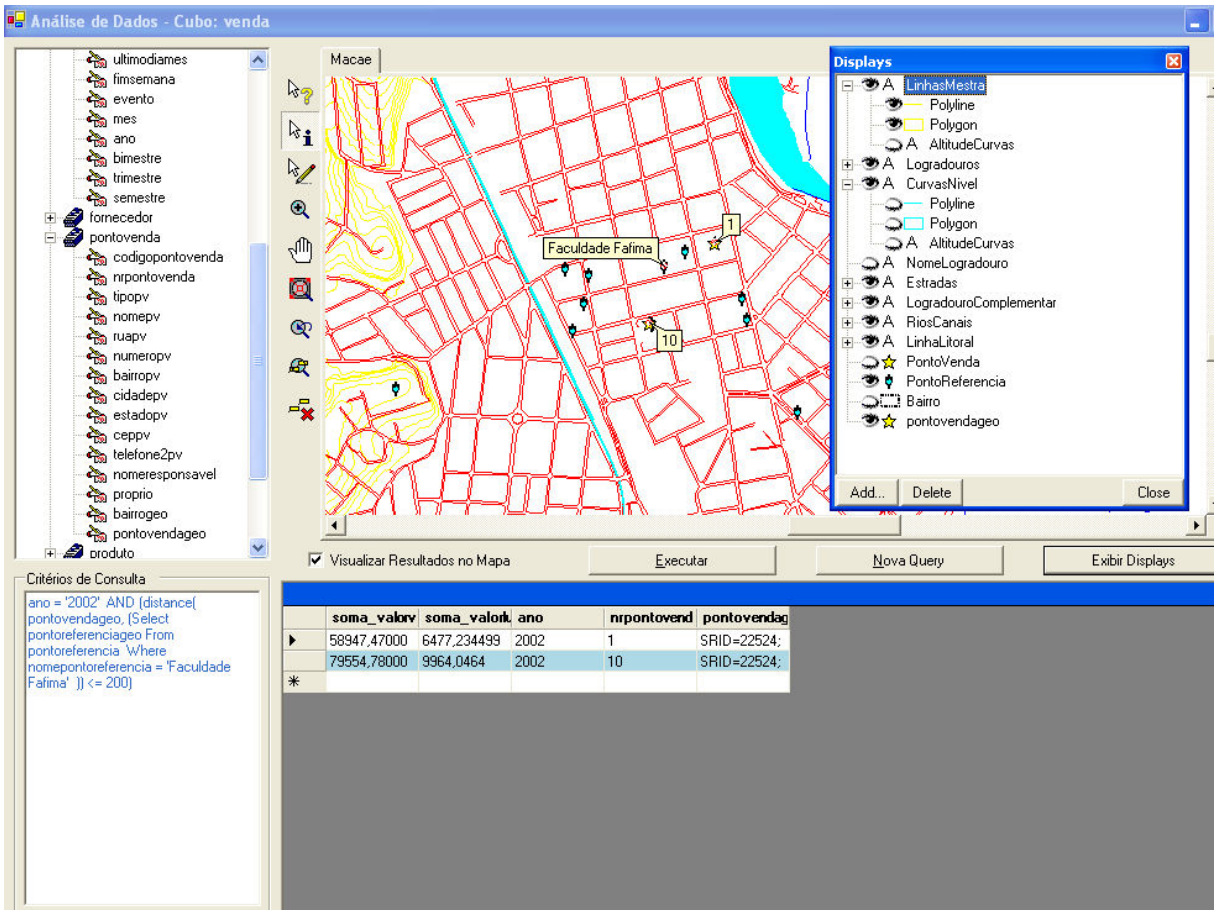


Fig.60 Resultado para a consulta Q2, com resultados analíticos e geográficos

No caso da consulta Q2, como um atributo geográfico foi especificado ("pontovendageo"), a ferramenta exibe no mapa a representação geográfica para suas instâncias (em verdade o simples "desenho" dos dados contidos no campo "pontovendageo").

Os demais objetos presentes no mapa (pontos de referência, logradouros, linhas de litoral, etc) são displays do arquivo .map (arquivo do PlanetGIS) que foram deixados visíveis como forma de proporcionar um "contexto geográfico" para a visualização das respostas geográficas.

Pelo anteriormente exposto, verificamos a simplicidade na modelagem de um sistema de suporte à decisão que integre funcionalidades analíticas e geográficas, e na exeqüibilidade de sua implementação direta, sem mapeamentos, visto que os mesmos tipos de dados empregados nos estereótipos geográficos do modelo conceitual estão presentes na SGBD Espacial no qual serão armazenados.

A ferramenta de integração, em verdade uma aplicação OLAP (ROLAP) com funcionalidades de visualização geográfica, obtém as respostas para as consultas do usuário (sejam elas convencionais ou geográficas) a partir do SGBD Espacial sobre o qual está trabalhando, dispensando o mapeamento entre aplicações.

7. CONCLUSÃO

Com o vertiginoso incremento na produção de dados por parte das organizações ao longo dos últimos anos, intensa tem sido a busca dos pesquisadores por sistemas capazes de tratar tal massa de informações segundo as dimensões que regem o próprio negócio, o tempo e o espaço, de forma a auxiliar gerentes e diretores na tomada de suas decisões.

Apesar da existência de diversos outros trabalhos com o objetivo de integrar em um único sistema de suporte à decisão, funcionalidades analíticas e geográficas, o foco quase sempre tem sido o da criação de um módulo que integrasse duas aplicações: uma OLAP e outra SIG. A necessidade de uma técnica de modelagem que permitisse unir conceitos espaciais e dimensionais em um único diagrama, e que fosse de simples tradução ou mapeamento para uma aplicação onde as funcionalidades de um SIG e de um aplicativo OLAP estivessem presentes, foi a grande motivação para o presente trabalho.

Também a necessidade era distinta daquela vivenciada pelos estudiosos dos trabalhos anteriores: A idéia era a concepção de um sistema a partir do zero, ou seja, não havia duas outras aplicações pré-existentes: uma SIG e outra OLAP, que necessitassem ser integradas. O propósito era o design de uma nova aplicação de suporte à decisão que contemplasse as funcionalidades analíticas e geográficas, de forma a obter, com um único design e um único aplicativo, aquilo que anteriormente só era obtido por meio do uso de duas aplicações distintas, com modelos e tipos de dados distintos e por meio de um terceiro aplicativo que mapeasse requisições e dados entre os dois primeiros.

Para a obtenção da tão buscada simplicidade, esta dissertação abordou o uso da UML com estereótipos geográficos para a modelagem conceitual de Data Warehouses onde uma ou mais dimensões fossem espaciais (embora nada impeça que um DW constituído apenas por dados convencionais seja modelado da mesma maneira). Trabalhando em um nível de abstração mais elevado (conceitual), sem as restrições impostas por determinada tecnologia de implementação, é possível representar com maior riqueza semântica e liberdade, todas as informações relevantes a um sistema de suporte à decisão de tal complexidade, sendo possível ainda melhor reaproveitar o modelo de objetos já existente na organização.

De posse do diagrama de classes acrescido de estereótipos geográficos, executa-se o mapeamento do modelo conceitual OO para o lógico dimensional relacional, conforme abordado no capítulo 4, obtendo-se um DED lógico pronto para ser mapeado para o modelo físico de algum SGBD Espacial aderente aos padrões do OGC (OpenGIS Consortium).

Tal Data Warehouse com características espaciais poderia ser facilmente manipulado por uma aplicação OLAP do tipo ROLAP (para utilizar a motor de consultas convencionais e geográficas do próprio SGBD) que possuísse funcionalidades de visualização dos dados em formato analítico (normalmente em uma tabela, planilha ou grid) e geográfico (em um mapa), e para provar tal premissa, foi construída a ferramenta PostGeoOlap.

A aplicação PostGeoOlap é uma ferramenta que integra funcionalidades de SIG e de OLAP (ROLAP) em um único aplicativo. Escrita em VB.Net com um componente COM para visualização geográfica denominado PlanetGIS e trabalhando sobre o SGBD espacial PostGreSQL, adicionado do PostGIS, a ferramenta se apóia sobre um cubo de dados de um Data Warehouse (que pode possuir dimensões geográficas ou não), gerando para este cubo diversas agregações com a finalidade de acelerar as consultas sobre tais dados. Quando de sua utilização, o PostGeoOlap, provê uma interface gráfica e de seleção da melhor agregação para a submissão das consultas elaboradas pelo usuário e para a visualização dos resultados, quer os dados solicitados sejam convencionais ou geográficos.

7.1 CONTRIBUIÇÕES DO PRESENTE TRABALHO

A presente dissertação apresenta as seguintes contribuições para a área de Sistemas de Informação:

- Uma técnica de modelagem que permite integrar, desde o nível conceitual, a partir do modelo de objetos já existente na empresa, conceitos geográficos e analíticos, permitindo ainda que tais conceitos sejam traduzidos de forma direta para uma aplicação que contemple funcionalidades dimensionais e espaciais;

- O emprego de estereótipos geográficos para a modelagem conceitual de Data Warehouses, que tornam mais simples a tarefa de integrar, em mesmo diagrama DW, conceitos dimensionais e espaciais;

- A ferramenta PostGeoOlap, que é uma aplicação OLAP (ROLAP) capaz de manipular dados espaciais e convencionais, sem a necessidade de módulos de integração entre aplicativos distintos;

- Sistemas de suporte à decisão como aplicativos OLAP ou SIG possuem normalmente custos de aquisição ou licenciamento proibitivos para grande parte das médias ou pequenas empresas. A proposta do presente trabalho foca a utilização de componentes gratuitos ou software livre, tornando acessível a um público muito maior um sistema de suporte à decisão que, utilizado nas empresas, pode contribuir para a redução de custos e incremento dos lucros, trazendo competitividade; e utilizado em órgãos públicos, pode contribuir para a melhoria da qualidade de vida dos cidadãos;

- A existência do projeto PostGeoOlap permitirá também que outros trabalhos, com foco no suporte à decisão, sejam desenvolvidos com base na citada ferramenta.

7.2 SUGESTÕES PARA TRABALHOS FUTUROS

Em função da extensão da proposta do presente trabalho, que apresenta desde uma forma de modelagem a nível conceitual que integra noções dimensionais e espaciais, até uma ferramenta de aplicação para manipulação on-line, passando pelo mapeamento do citado nível conceitual OO para o lógico dimensional relacional, várias são as propostas de trabalhos futuros com vistas a aperfeiçoar o processo em questão:

- Criação de uma ferramenta CASE que permita a modelagem UML das classes com os devidos estereótipos e que, a seguir, gere o correspondente esquema do Data Warehouse no PostGreSQL adicionado do PostGIS;

- Otimização da ferramenta PostGeoOlap, com vistas a melhorar seu desempenho;

- Estudo de outras formas de submissão de restrições geográficas para as consultas;

- Extensão do modelo de forma a incluir metadados geográficos para tratamento de aspectos ligados à visualização dos objetos como, por exemplo, a definição de que os objetos geográficos do tipo "Curva de nível" devam ser representados na cor amarela;

- Emprego de outro componente de visualização para a ferramenta PostGeoOlap. Apesar de toda a capacidade do PlanetGIS, o componente não utiliza os mesmos tipos de dados do OpenGIS;

- Tendo em vista o PostGeoOlap ser um software livre, convertê-lo de VB.Net para Java, tendo em vista a grande quantidade de desenvolvedores na citada linguagem e a sua tradição no ambiente de código livre;

- Utilização de XML para armazenamento dos metadados, em substituição ao MS ACCESS;

- Melhoria da interatividade da ferramenta PostGeoOlap, tornando mais simples sua utilização: um dos pontos seria a tabela de análise dos dados, que deveria permitir maior flexibilidade de manipulação (para as operações de roll-up/drill-down e pivoting);

- Criação de ferramentas para a administração do data warehouse (carga de dados);

- Definição de estratégia para o re-processamento do Cubo, com vistas à atualização das agregações, em face de novas cargas de dados no Data Warehouse. Uma das abordagens sugeridas seria o uso de visões materializadas; como o PostGreSQL ainda não possui tal capacidade, a citada funcionalidade poderia ser implementada por meio de triggers e stored procedures;

- Implementação, na ferramenta PostGeoOlap, da capacidade de definição de múltiplas hierarquias para cada uma das dimensões (conforme diagrama de classes – conceitual - da ferramenta);

- Emprego da aplicação PostGeoOlap como ferramenta em outros trabalhos cujo foco seja o Suporte à Decisão.

8. REFERÊNCIAS

AUTODESK INC. *Autodesk MapGuide Product Information*. Disponível em <http://www.autodesk.com/mapguide>. Acesso em março de 2002.

CAMPOS, Maria Luiza; ROCHA FILHO, Arnaldo V. *Data Warehouse*. Disponível em <http://genesis.nce.ufrj.br/dataware/tutorial/home.html>. Acessado em 13/02/2004.

CAMARA, Gilberto; DAVIS, Clodoveu; PAIVA, João A.; CASANOVA, Marco A. *Geoprocessamento: Teoria e Aplicações*. Disponível em: <http://www.dpi.inpe.br/gilberto/livro/>. Acesso em 13/02/2004.

COLLIAT, George. *Olap, Relational and Multidimensional Database Systems*. SIGMOD Record, Vol 25, No.3, Setembro 1996.

COME, Gilberto de. *Contribuição ao estudo da Implementação de Data Warehousing: um caso no setor de Telecomunicações*. São Paulo: FEA/USP, 2001. (Dissertação de Mestrado).

COLONESE, Giovanni; TANAKA, Astério K.; CARVALHO, Rogério A. *Mapeando Modelos Conceituais Dimensionais OO para Modelos Lógicos Dimensionais Relacionais*. (Artigo aceito para publicação nos anais do CACIC 2003, Argentina).

DATAWAREHOUSE.COM. *Entity Relationship Modeling with UML*. Maksimchuk, Robert A.; Naiburg, Erik J. Disponível em <http://www.datawarehouse.com/iknowledge/articles/article.cfm?ContentID=3444>. Acesso em: 03/01/2003.

DATAWAREHOUSE.COM. *Hidden Complexities in a Simple Star Schema.* Schraml, Todd. Disponível em: <http://www.datawarehouse.com/iknowledge/articles/article.cfm?ContentID=2250>. Acesso em: 13/06/2003.

DATAWAREHOUSE.COM. *UML Class Models as a Data Modeling Tool*. Dorsey, Paul. Disponível em <http://www.datawarehouse.com/iknowledge/articles/article.cfm?ContentID=2685>. Acesso em: 06/06/2002.

DBMS AND INTERNET SYSTEMS. *A Dimensional Modeling Manifesto*. Kimball, R. A. Disponível em <http://www.dbmsmag.com/9708d15.html>. Acesso em: 18/07/2003.

DSSRESOURCES.COM. *A Brief History of Decision Support Systems*. Power, Daniel J. Disponível em <http://www.dssresources.com>. Acesso em: 23/06/2003.

DSSTAR. *Object Oriented Data Warehouse*. Firestone, Joseph M. Disponível em <http://www.tgc.com/dsstar/00/1003/102240.html>. Acesso em: 14/06/2003.

E. F. CODD ASSOCIATES. *Providing Olap to User-Analysts: An IT Mandate*. Codd, E. F.; Codd, S. B.; Salley, C. T. 1993.

ESRI. *Esri ArcExplorer*. Disponível em: http://www.esri.com/software/arcexplorer/overview2.html. Acesso em 22/08/2003.

FERREIRA, A.C.; CAMPOS, M. L.; TANAKA, A. *An Architecture for Spatial and Dimensional Analysis Integration*. In Proceedings of World Multiconference on Systemics, Cibernetics and Informatics (SCI 2001). Vol. XIV Computer Science and Engineering. Part II. p.392-395. Orlando, Florida, EUA. Julho, 2001.

FERREIRA, A. C. F. *Um Modelo para Suporte à Integração de Análises Multidimensionais e Espaciais*. Rio de Janeiro: UFRJ/IM/NCE, 2002. (Dissertação de Mestrado).

FIDALGO, R.N.; TIMES, V. C.; SOUZA, F. F. *GOLAPA: Uma Arquitetura Aberta e extensível para Integração entre SIG e OLAP*. GeoInfo 2001, III Workshop Brasileiro de GeoInformática. p.111-118. Instituto Militar de Engenharia, Rio de Janeiro. 4 e 5 de outubro de 2001.

GOLFARELLI, M.; MAIO, D.; RIZZI, S. *Conceptual Design of Data Warehouses from E/R Schemes*. In Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98), Hawaii, EUA. Janeiro, 1998.

GUPTA, H.; MUMICK, I. *Selection of Views to Materialize Under a Maintenance-Time Constraint*. ICDT, 1999.

HAHN, K; SAPIA, C.; BLASCHKA, M. *Automatically Generating OLAP Schemata from Conceptual Graphical Models*. Relatório Técnico. FORWISS, Munich, Germany, Outubro 2000. Disponível em <http://www.forwiss.tu-muenchen.de/~system42/publications>. Acesso em: 20/09/2003.

HAN, J; STEFANOVIC, N; KOPERSKI, K. *Selective Materialization: An Efficient Method for Spatial Data Cube Construction*. PAKDD, 1998.

INMON, W. H. *Como construir o Data Warehouse*. Campus, 1997.

INPE – INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. *Spring – Sistema de Processamento de Informações Georeferenciadas*. Disponível em <http://www.dpi.inpe.br/spring/portugues/index.html>. Acesso em 10/08/2003.

INPE – INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. *SpringWeb Versão 3.0*. Disponível em <http://www.dpi.inpe.br/spring/portugues/sprweb/springweb.html>. Acesso em 20/08/2003.

KIMBALL, Ralph. *The Datawarehousing Toolkit.* New York: John Wiley & Sons, 1996.

KIMBALL, Ralph. *What's Wrong with SQL*. Disponível em <http://www.almaden.ibm.com/cs/people/ragrawal/papers/md_model_rj.ps>. Acesso em 20/09/2003.

KOUBA, Z.; MATOUSEK, K..; MIKSOVSKY P. *On Data Warehouse and GIS Integration*. In Proceedings of DEXA2000. Greenwich, Inglaterra: DEXA2000, 2000.

LISBOA, J. *Projeto Conceitual de Banco de Dados Geográficos através da Reutilização de esquemas, utilizando padrões de Análise e um Framework Conceitual*. Porto Alegre:UFRGS, 2000. (Tese de Doutorado).

LISBOA, J. F.; IOCHPE, C. *Specifying Analysis Patterns for Geographical Databases on the basis of a Conceptual Framework*. ACM Symposium on Advances in Geographic Information Systems. ACM Press, 1999 p.7-13. Kansas City, EUA: 1999.

MAPSERVER. *MapServer Documentation Project*. Disponível em <http://mapserver.gis.umn.edu/doc.html>. Acesso em 13/08/2003.

MICROSOFT. *Visual Basic.Net*. Disponível em: < http://msdn.microsoft.com/vbasic/>. Acesso em 01/09/2003.

MICROSOFT ANALYSIS SERVICES. Books on Line. Documentação on-line da aplicação MS Analysis Services, distribuída juntamente com MS SQL SERVER 2000. Microsoft, 2000.

NAVATHE, S. B.; ELMASRI, R. *Fundamentals of Database Systems*. Addison-Wesley, 2001.

OGC: *Open GIS Consortium*. Disponível em <http://www.opengis.org>. Acesso em 17/10/2003.

OGC (OPEN GIS CONSORTIUM). *OpenGIS Simple Specifications for SQL Revision 1.1*. OpenGis Project Document 99-049. Publicado em 05/05/1999.

OMG: *CWM - Common Warehouse Metamodel Specification*. Disponível em <http://www.omg.org>. Acesso em 06/09/2003.

OMG. *Unified Modeling Language*. Disponível em <http://www.omg.org/>. Acesso em 11/06/2003.

PAPADIAS, D.; KALNIS, P.; ZHANG, J.; TAO, Y. *Efficient OLAP Operations in Spatial Data Warehouses*. Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases. Pág.: 443 a 459. ACM Records, 2001.

PLANETGIS. *Documentação do software PlanetGIS, versão 2.29*. Disponível em <http://www.planetgis.co.za/>. Acesso em 20/10/2003.

POSTGIS. *Documentação da extensão espacial PostGIS ao PostGreSQL, versão 0.*8. Disponível em <http://postgis.refractions.net/>. Acesso em 05/02/2004.

POSTGRESQL.ORG. *Documentação do SGBD PostGreSQL 7.4*. Disponível em: <http://www.postgresql.org>. Acesso em 05/02/2004.

PROJETO CIMA: INFORMAÇÕES TÉCNICAS EM TECNOLOGIA SIG. *Projeto para especificação de aplicações em tecnologia SIG visando o atendimento dos Municípios da região Norte de Portugal*. Documentos técnicos. Disponível em: <http://simat.inescn.pt/doc/doc_tecnicos/rer101/apendices/v101/openGis.html>. Acesso em 20/01/2004.

ROCHA, L.V.; EDELWEISS, N. *GeoFrame-T: um Framework Conceitual Temporal para Aplicações de Sistemas de Informação Geográfica*. Porto Alegre: UFRGS, 2001. (Dissertação de Mestrado).

ROCHA, L. V.; EDELWEISS, N. *O Framework Conceitual Temporal GeoFrame-T na prática*. Porto Alegre: PPGC - UFRGS, 2001. (Relatório de Pesquisa).

ROCHA, L.V.; IOCHPE, C.; EDELWEISS, N. *O Framework Conceitual GeoFrame – versão 2.0*. Porto Alegre: PPGC –UFRGS, 2001. (Relatório de Pesquisa RP-309).

SHEKHAR, S.; LU, C.T.; TAN, X.; CHAWLA, S.; VATSAVAI, R.R.. *Map Cube: A visualization Tool for Spatial Data Warehouse*. Disponível em <http://www.cs.umn.edu/research/shashi-group/mapcube.htm>. Acesso em 20/01/2004.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistema de Banco de Dados*. São Paulo: Pearson Education do Brasil, 1999.

STEFANOVIC, N.. *Design and Implementation of On-Line Analytical Processing (OLAP) of Spatial Data*. Dissertação de Mestrado. Disponível em <http://gunther.smeal.psu.edu/3070.html>. Universidade de Belgrado, 1997. Acesso em 20/09/2003.

TORRES, Norberto. A Competitividade Empresarial com a Tecnologia de Informação. São Paulo. Makron Books, 1995.

TRUJILLO, Juan; PALOMAR, Manuel; GOMEZ, Jaime et al. *Designing Data Warehouses with OO Conceptual Models*. Computer IEEE, 2001, p.66-75.

TRUJILLO, J.; PALOMAR, M. GÓMEZ, J. *The GOLD Definition Language (GDL): An Object Oriented Formal Specification Language For Multidimensional Databases*. In Proceedings ACM Symposium on Applied Computing (SAC'00). Como, Itália, 2000.

9. APÊNDICE

9.1 PSEUDO-CÓDIGO DO ALGORITMO DE PROCESSAMENTO DO CUBO

```
Classe Pilha
        Inteiro Nivel
fim_classe

Algoritmo ProcessarCubo
Variáveis
        Inteiro NrDimensoes          //Quantidade de dimensoes a participar das agregações

        Coleçao ColAtributosFato     //Coleção para armazenar os atributos da tabela Fato

        Coleçao ColDimen             //Coleção que armazena as dimensões do Cubo a ser
                                     //processado
        Pilha NovaPilha              //Cada Pilha é um tipo de dado composto por uma
                                     //sequência de elementos Nivel, obtidos de Pilha.Nivel
        Pilha PilhaAtiva             //Pilha auxiliar - vide algoritmo
        Coleçao ColPilha             //Coleção de Pilhas.
        Dimensao Dimensao1           //Dimensão número 1 do tipo de dados Dimensão
        Inteiro NrNivel              //Quantidade de níveis
        Coleçao ColAtributosDimen1 //armazena na coleção os atributos da dimensão
                                     //número 1
        Inteiro I                  //contador auxiliar
```

```
        Inteiro j               //contador auxiliar
        Inteiro NrUltPilha      //busca o Topo (último Pilha da estrutura)
        Booleano BlnProcessa    //indica a necessidade ou não de processamento para
                                //determinada perspectiva do Cubo
        Dimensao DimensaoAtiva  //Representa a dimensão do Cubo em questão
        Inteiro NrAtribPilha    //Quantidade de Niveis já existentes na Pilha em
                                //questão
Inicio
        Ler ColDimen
        NrDimensoes = ColDimen.Contagem     //Contagem é um atributo que retorna a
                                            //qtd de elementos da coleção
        SE (NrDimensoes < 2) Entao
                ESCREVER "O Cubo possui menos de 2 dimensões. O processamento será
                abandonado!"
                Sair
        Fim_SE
        Ler ColAtributosFato
        Dimensao1 = ColDimen.Item(1)        //primeira Dimensão da coleção de
                                            //Dimensões
        NrNivel = Ler (nível hierárquico mínimo de) Dimensao1   //chama
                                                                //dimensao1.minimo()
        ColAtributosDimen1 = Ler (Coleção de atributos de) Dimensao1
                //chama dimensao1.atributos()
        //Cria novas pilhas, adicionando a cada uma delas um nível com o valor vindo da
        // hierarquia

       NovaPilha.Niveis.Adiciona(0)         //Cria o nível zero (representativo do
                                            //'All' para esta dimensão) - adicionar
                                            //no fim.
       ColPilha.Adiciona(NovaPilha)         //Adiciona a pilha anterior à coleção.
       For i = NrNivel até 9
                NovaPilha = new Pilha
                NovaPilha.Niveis.Add(i)     //Adiciona o nível com o valor copiado da
                                            // sequência hierárquica da dimensão
                ColPilha.Adiciona(NovaPilha)
       Proximo
       //Agora existe uma coleção de pilhas, cada uma delas com um único nível
       // representativo da dimensao1
       //é iniciar um loop que faz com que tais  lementos possam ir se juntando aos níveis
       //da dimensão2,
       //em seguida aos da dimensão, e assim por diante.

       Faça Enquanto ColPilha.QuantidadeDeItens <> 0
                PilhaAtiva = reinicializa Pilha()
                PilhaAtiva = ColPilha.Item(ColPilha.Contagem) //Copia a pilha do topo da
                                                              // estrut. para a variável
                                                              //'PilhaAtiva'
```

```
NrUltPilha = ColPilha.Contagem
ColPilha.Remove(NrUltPilha)        //Remove a pilha do topo da estrutura,
                                   //para que sofra processamento
//Checa se a pilha está completa, ou seja com tantos niveis quanto dimensões
SE (PilhaAtiva.Niveis.Contagem = NrDimensoes) Entao
        BlnProcessa = Falso        //A princípio não executa o
                                   //processamento, que ocorrerá exceto
                                   //quando todos os níveis possuírem
                                   //valor igual a 9 (ou seja, referente ao
                                   //nível base do data warehouse)

        Para i = 1 até PilhaAtiva.Niveis.Contagem
                SE PilhaAtiva.Niveis.Item(i) <> 9 Então
                        BlnProcessa = Verdadeiro
                Fim_SE
        Proximo
        SE BlnProcessa = Verdadeiro Então
            //Para maiores detalhes vide item 5.3.5 da dissertação
            //Executa a criação da agregação, cópia de dados e indexação da
            //citada agregação, caso o custo-benefício para sua existência
        .....//seja compensador
            PROCESSAR_PERSPECTIVA(PilhaAtiva)
        Fim_SE
SENÃO
        //Faz a combinação dos niveis ja existentes na pilha com os da
        //hierarquia da proxima dimensao, até que o nr de  niveis em uma
        //pilha seja igual ao nr de dimensoes a dimensão ativa (aquela com a
        //qual serão feitas as combinações)
        //é obtida pela contagem de atributos da PilhaAtiva, somando-se um
                NrAtribPilha = PilhaAtiva.Niveis.Count + 1

                DimensaoAtiva = ColDimen.Item(NrAtribPilha)
        //Recupera a Dimensão Ativa pelo índice
                NrNivel = DimensaoAtiva.RetornarGrauMinHierarquia
                //quantidade de niveis da dimensao corrente
                NovaPilha = reinicializa Pilha()       //Limpa o conteúdo da
                                                       // variável NovaPilha

                //copia para NovaPilha o conteúdo de PilhaAtiva
                Para i = 1 Até PilhaAtiva.Niveis.Contagem
                        NovaPilha.Niveis.Adiciona(PilhaAtiva.Niveis.Item(i))
                Próximo
                NovaPilha.Niveis.Adiciona(0)       //já deixa criado o nível 0
        ColPilha.Adiciona(NovaPilha)       //adiciona a pilha à coleção

        //percorre todos os niveis da dimensao corrente até o 9 (o zero já foi
        //inserido de propósito para que as menores granularidades antes
```

```
                    //e novas agregação possam se basear nelas e não no nível base para
                    //serem formadas
                    Para j = NrNivel Até 9
                        //copia para uma nova pilha o conteúdo da pilha
                        //anterior e adiciona os novos atributos (Não usar:
                        //"NovaPilha = PilhaAtiva")
                        //senão a cópia por referência inutiliza a variável
                        NovaPilha = reinicializa Pilha()
                        Para i = 1 Até PilhaAtiva.Niveis.Contagem
                            NovaPilha.Niveis.Adiciona(PilhaAtiva.Niveis.Item(i))
                        Próximo
                        NovaPilha.Niveis.Adiciona(j)
                        ColPilha.Adiciona(NovaPilha)
                    Próximo
                Fim_SE
            Fim_Enquanto
Fim
```