UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ALEXANDRE OGRODOVSKI

CONSTRUÇÃO DE SPATIAL DATA WAREHOUSES UTILIZANDO SOFTWARES LIVRES

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2013

ALEXANDRE OGRODOVSKI

CONSTRUÇÃO DE SPATIAL DATA WAREHOUSES UTILIZANDO SOFTWARES LIVRES

Trabalho de diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas - CSTADS - da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Dr. Claudio Leones Bazzi.

Co-orientador: Msc. Alan Gavioli.

MEDIANEIRA

2013

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Coordenação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

TERMO DE APROVAÇÃO

CONSTRUÇÃO DE SPATIAL DATA WAREHOUSES UTILIZANDO SOFTWARES LIVRES

Por

ALEXANDRE OGRODOVSKI

Este Trabalho de Diplomação (TD) foi apresentado às 13:00 h do dia 26 de agosto de 2013 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. (Dr. Claudio Leones Bazzi)
UTFPR – *Campus* Medianeira
(Orientador)

Prof. (Msc. Hamilton Pereira da Silva)
UTFPR – *Campus* Medianeira
(Convidado)

Prof. (Msc. Ricardo Sobjak)
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

A folha de aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

OGRODOVSKI, Alexandre. CONSTRUÇÃO DE SPATIAL DATA WAREHOUSES UTILIZANDO SOFTWARES LIVRES. 77 f. TRABALHO DE DIPLOMAÇÃO – Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Medianeira, 2013.

O *Spatial Data Warehouse* é uma base de dados de nível corporativo, capaz de fornecer de dados à Sistemas de Suporte a Decisão com um tempo de resposta aceitável para ambientes estratégicos. Os dados que possui, na estrutura que se encontram, permitem ser explorados e apresentados de forma adequada em soluções baseadas em Sistemas de Informações Geográficas. Neste trabalho são avaliados *softwares* livres para construção de *Spatial Data Warehouses* através do desenvolvimento de um projeto experimental. Para tanto são confrontados os requisitos de extração, transformação e carga de dados espaciais e as funcionalidades oferecidas pelas ferramentas. Também são apresentados conceitos de *Data Warehouses*, *Spatial Data Warehouses*, Sistemas de Suporte a Decisão, Sistemas Gerenciadores de Banco de Dados e ferramentas para ETL. É possível verificar que o sistema de banco de dados utilizado, ao ser complementado por um *software* que lhe adicionasse suporte a espacialidade, pôde acomodar todo conjunto de dados espaciais. Também foi verificado que ao satisfazer todos requisitos de extração, transformações e carga de de dados, o sistema utilizado para tais funções mostrou-se apto para utilização em projetos de *Spatial Data Warehouse*.

Palavras-chave: Armazém de Dados Espaciais, Armazém de Dados, Sistema de Suporte a Decisão, Ferramenta de ETL

ABSTRACT

OGRODOVSKI, Alexandre. CONSTRUCTION OF SPATIAL DATA WAREHOUSES USING FREE SOFTWARES. 77 f. TRABALHO DE DIPLOMAÇÃO – Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Medianeira, 2013.

The Spatial Data Warehouse is a corporate level database, capable of providing data to Decision Support Systems with an acceptable response time for strategic environments. It's data in the structure they are encountered, can be explored and presented appropriately in Geographic Information System's based solutions. This paper aims to evaluate free softwares for building Spatial Data Warehouses by means of developing an experimental project. The evaluation is done by confronting the requirements of extraction, transformation and load of spatial data and the features offered by the tools. Also are presented concepts of Data Warehouses, Spatial Data Warehouses, Decision Support Systems, Data Base Management Systems and ETL softwares. It's possible to verify that the used database system, beeing complemented by a software that added support for spatiality, was able to accommodate the entire spatial dataset. Also was verified that satisfying all the requirements of extraction, transformations and load of data, the used system for these operations could be used in Spatial Data Warehouse's projects.

Keywords: Data Warehouse, Decision Support System, ETL tool, Free software

LISTA DE FIGURAS

LISTA DE SIGLAS

CSV	Comma-separated values
CSW	Compressed Square Wave
DBA	Database Administrator
DBF	Arquivo de banco de dados
DM	Data Mart
DW	Data Warehouse
ETL	Extract, Transform, Load
FTP	File Transfer Protocol
JSON	JavaScript Object Notation
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
SDW	Spatial Data Warehouse
SFTP	Secure File Transfer Protocol
SGBD	Sistema Gerenciador de Banco de Dados
SIG	Sistema de Informacao Geográfica
SK	Surrogate Key
SOLAP	Spatial Online Analytical Processing
SQL	Structured Query Language
SSD	Sistema de Suporte a Decisão
SSH2	Secure Shell 2
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Muitas empresas encontram dificuldades para obter informações que apresentem uma visão mais ampla do negócio, útil para tomada de decisões estratégicas (INMON, 2002). Os dados que possuem, no formato que se encontram nos sistemas transacionais, suportam bem as decisões operacionais do dia-a-dia. Mas quando necessitam de dados integrados e sumarizados, se deparam com um processo difícil de ser implementado.

Em cenários onde decisões de nível corporativo são baseadas em dados operacionais de fontes dispersas, são recorrentes os problemas relacionados a consistência, inteligibilidade e confiabilidade das informações. Nesta arquitetura, o processo de integração destes dados se faz necessário sempre que houver a necessidade de informações que representem a organização como um todo.

Uma solução alternativa que facilitaria o acesso a estas informações, seria a construção de uma base de dados operacionais centralizada, eliminado o retrabalho de integração. Porém, Bédard et al. (2001) alerta que tentativas de substuição de sistemas independentes já estabelecidos por uma única solução integrada tendem a falhar devido a transtornos e dificuldades políticas internas.

O conceito de *Data Warehouse* (DW) surgiu na década de 80, apresentando-se como a solução unificada para integração dos dados de bases dispersas e heterogêneas. Em sua arquitetura, sistemas legados não são substituídos, mas sim complementados com a inclusão de um sistema preparado para manter dados de toda corporação.

Esta solução traz consigo benefícios como consistência, confiabilidade e segurança dos dados, bem como flexibilidade a mudanças arquiteturais e alta disponibilidade das informações. Outras vantagens conseguidas com seu uso, como inteligibilidade das informações, dados de qualidade, estrutura de dados adequada e informações relevantes para tomada de decisões fazem desta tecnologia a base ideal para Sistemas de Suporte a Decisões (SSD).

Com a crescente necessidade de decisões baseadas em fatos, o *Data Warehouse* é considerado cada vez mais um forte aliado em ambientes estratégicos. Com características de prontidão e simplicidade arquitetural, fornecem informações instantâneas e de baixo custo, aumentando a vantagem competitiva da empresa.

Ao unir conceitos de *Data Warehouse* com bases de dados espaciais, as possibilidades

de análise, bem como a qualidade das decisões são aumentadas. Da união destas tecnologias origina-se o *Spatial Data Warehouse*, capaz de gerenciar dados georreferenciados e podendo assim servir de base para descoberta de conhecimentos geográficos.

Segundo Karabegovic e Ponjavic (2012), a forma de referenciamento geográfico utilizada por *Spatial Data Warehouses* abre novas possibilidades de análises para Sistemas de Suporte a Decisão. Novos relacionamentos e padrões podem ser descobertos sobre os dados de negócio através de ferramentas para mineração de dados espaciais. Outra forma de exploração é com uso de ferramentas *Spatial Online Analytical Processing* (SOLAP), capazes de representar gráficamente a espacialidade dos dados.

Apesar dos benefícios oferecidos pelo *Spatial Data Warehouse*, sua implementação é muitas vezes desencorajada devido a custos relacionados as tecnologias utilizadas. Entre estes custos estão o licenciamento dos *softwares* para processos de extração, transformação e carga (ETL) de dados e o Sistema Gerenciador de Banco de Dados (SGBD), sobre o qual será construída a solução. Mesmo ao se optar por soluções mais simples, que é o caso dos *Spatial Data Marts* que exigem menor investimento financeiro, os custos com *software* podem levar o projeto a não ser realizado (WIXOM, 2007).

Neste trabalho serão utilizadas ferramentas gratuitas para a realização de um projeto experimental de *Spatial Data Warehouse*, afim de julgar a aptidão das soluções de *softwares* escolhidas para desenvolvimento de um projeto real.

1.1 OBJETIVO

Estudar a viabilidade técnica para construção de *Spatial Data Warehouses* utilizando somente *softwares* livres.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar um levantamento teórico sobre os principais conceitos relacionados a *Spatial Data Warehouses*.

- Selecionar tecnologias gratuitas destinadas a construção de *Spatial Data Warehouses*.
- Desenvolver um projeto experimental, executando operações de extração, transformações e carga de dados espaciais utilizando apenas *softwares* livres.
- Avaliar o desempenho quanto a riqueza de recursos das ferramentas utilizadas no projeto experimental.

1.3 JUSTIFICATIVA

Tomadores de decisões esperam que Sistemas de Suporte a Decisões forneçam respostas rápidas e alto nível de flexibilidade para exploração de dados em diferentes níveis de agregação (BÉDARD et al., 2001). Em cenários onde decisões devem ser tomadas com base em informações geográficas, espera-se que seja possível uma visualização adequada dos dados espaciais, como através de mapas temáticos. Tais requisitos são satisfeitos com a utilização de *Data Warehouses* ou *Spatial Data Warehouses*, os quais atuam como base para SSDs.

Segundo Inmon (2002), o uso de *Data Warehouses* diminui o custo das informações em duas ordens de magnitude. Se com este sistema um relatório custasse um dólar, sem este sistema custaria cem dólares. Para uma empresa que não tenha implementado este sistema, cada consulta levaria a execução das tarefas de encontrar os dados necessários, acessá-los, integrá-los e apresentá-los de forma adequada, justificando assim o custo elevado da informação.

Apesar de oferecer benefícios como redução do custo de informações e fornecimento de informações de negócio baseadas em fatos, o projeto de *Spatial Data Warehouses* acaba muitas vezes por não ser realizado por empresas de baixo poder aquisitivo, devido seu alto custo de contrução. Este valor elevado é fortemente influenciado pelas licenças dos *softwares* necessários para sua construção. Segundo Sodré (2013), ”o preço de uma licença, dependendo da ferramenta pode chegar a 40% do custo total de um projeto de ETL”.

Busca-se uma resposta quanto a viabilidade de se utilizar *softwares* livres para construção de *Spatial Data Warehouses*, realizado por meio de um projeto experimental explorando os requisitos de funcionalidades das ferramentas envolvidas. A aptidão dos *softwares* para desenvolvimento de projetos reais pode ser provada caso os requisitos identificados sejam satisfeitos.

1.4 ORGANIZAÇÃO

Este trabalho é dividido em seis capítulos, sendo apresentado no primeiro a problemática envolvendo o tema estudado, os objetivos, a motivação para o desenvolvimento e uma breve descrição dos resultados obtidos.

No segundo capítulo encontra-se a fundamentação teórica, sendo abordados conceitos de *Data Warehouse*, *Spatial Data Warehouse* e tecnologias para desenvolvimento e exploração das soluções.

No terceiro é descrita a proposta de desenvolvimento do estudo de caso que comprova se é possível ou não construir o *Spatial Data Warehouse* utilizando apenas de *softwares* livres.

No quarto capítulo são apresentados os *softwares* utilizados no experimento, bem como suas principais características. Também é apresentado o conjunto de dados espaciais manipulado no projeto de ETL.

No quinto capítulo é apresentado e explicado o desenvolvimento do experimento, bem como os requisitos de ETL identificados para o conjunto de dados utilizado. Neste capítulo também são apresentados alguns recursos oferecidos pelos softwares.

No último capítulo são feitas considerações finais, apresentando as dificuldades e facilitades encontradas na utilização das ferramentas. No mesmo capítulo são feitas sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 DATA WAREHOUSE

Data Warehouse é a fonte de dados que alimenta os Sistemas de Suporte a Decisões (SSD) (KIMBALL et al., 1998). Ele está para tais sistemas assim como bases de dados operacionais estão para sistemas Online Transaction Processing (OLTP). Kimball et al. (1998) reforça a importância desta solução ao afirmar que ”um dos maiores ativos de uma empresa é a informacão”, uma vez que esta riqueza reside no DW.

O *Data Warehouse* é o coração de um Sistema de Suporte a Decisão e sua história inicia a partir da evolução desta tecnologia. Para Inmon (2002), DW é a definição para Sistema de Informação moderno, servindo como base sólida para informações confiáveis em um ambiente estratégico.

Em ambientes estratégicos, decisões são tomadas com base em informações, e das disponíveis espera-se que sejam baseadas em fatos e construídas a partir de dados consistentes. Para Kimball e Ross (2002), o DW fornece informações com a qualidade necessária e garante a consistência[1] dos dados nele contidos. Sistemas de Suporte a Decisões baseados em *Data Warehouses* têm apenas disponíveis dados de qualidade, já que esta é validada no abastecimento da sua fonte de dados.

Segundo Kimball et al. (1998), o DW é preparado para suportar grandes quantidades de dados. Por isso, é fundamental que haja a navegabilidade requerida não tornando sua exploração uma atividade dificultosa. Para atender este requisito, os dados alí presentes encontram-se estruturados de forma adequada e compatível com o modelo requerido pelos SSDs.

Para Kimball et al. (1998), a acessibilidade é um dos benefícios oferecidos pelo *Data Warehouse*. Este se dispõem como uma fonte centralizada de dados, facilitando o suprimento das informações. Em um cenário onde esta solução não está presente, a recuperação dos dados tende a ser trabalhosa visto que estes podem estar em sistemas legados dispersos, de difícil acesso. A rapidez de alcance aos dados é significativamente aumentada com a utilização deste sistema.

[1] Consistência de dados está relacionada a qualidade e completeza. Em *Data Warehouses*, implica em definições comuns para todo seu conteúdo (KIMBALL; ROSS, 2002).

Outra vantagem da utilização desta tecnologia é a inteligibilidade dos dados. Em seu processo de abastecimento os dados são devidamente rotulados, facilitando o entendimento por seus utilizadores. Recursos como metadados são cuidadosamente implementados na construção do *Data Warehouse* tornando seu conteúdo mais compreensível (KIMBALL et al., 1998).

Segundo Kimball et al. (1998), um dos objetivos do DW é garantir a consistência das informações. Ele explica que tal qualidade é conseguida quando dados que se encontram armazenados em diferentes locais e são igualmente rotulados, significam exatamente a mesma coisa. O cuidado quanto aos rótulos é tomado antes do processo de carga, garantindo a consistência requerida.

Por conta da estrutura que os dados se encontram em *Data Warehouses* sua exploração é altamente flexível, suportando consultas dinamicamente construídas para dados de diferentes granularidades. Este modelo de dados permite que o DW sejá escalável, podendo ser realizadas novas inserções de grandes conjuntos de dados, sem que os já presentes sofram alterações (KIMBALL et al., 1998).

Kimball e Ross (2002) afirmam que estes sistemas devem ser projetados para serem adaptáveis. Explica que ao mudar tecnologias, condições de negócio ou necessidades dos usuários, dados e aplicações não devem ser invalidados.

Segundo Kimball e Ross (2002), os problemas recorrentes que levam gerentes de negócio a desejar DWs são:

- A necessidade de analisar dados sob várias perspectivas, sendo possível utilizar um grande número de variáveis como filtros em consultas, e sendo estas consultas realizadas de acordo com a realidade da empresa, não se limitando a funções pré-programadas;
- A dificuldade de acessar os dados espalhados por diversas fontes entre os departamentos da empresa ou mesmo em outras filiais;
- A necessidade de um sistema que apresente apenas o que é relevante para auxílio na tomada de decisões;
- A ausência de Sistemas de Suporte a Decisões que se baseie em fatos.

2.1.1 CARACTERÍSTICAS

Há um consenso entre os principais estudiosos desta tecnologia quanto as suas principais características. São elas:

- Orientado a organização;
- Integrado;
- Não volátil;
- Apenas para leitura;
- Possui fontes heterogêneas;
- Dados com diferêntes níveis de granularidade;
- Construído para ser a base de Sistemas de Suporte a Decisão.

Diferente de bases de dados operacionais, *Data Warehouses* não são orientados a aplicações. Estes são projetados para atender a organização e possui dados que dizem respeito a várias atividades da empresa. Não são construídos para suprir sistemas OLTP, utilizados em atividades do dia-a-dia, mas sim para atender a sistemas que suportem ações estratégias para o negócio como um todo.

Seu conjunto único de dados não possui a granularidade existente em ambientes operacionais, mas possui dados sumarizados e preparados para o entendimento do negócio como um todo (BÉDARD et al., 2001), por isso, é caracterizado como sendo orientado a organização.

Segundo Kimball et al. (1998), o *Data Warehouse* integra dados de todos os departamentos, possibilitando a construção de consultas considerando fatores relacionados a várias áreas da empresa. A integração que este sistema oferece, facilita o alcance dos dados necessários para tomada de decisões estratégicas. Em uma realidade onde um sistema com esta função não está presente, a recuperação de dados dispersos deve ser feita sempre que houver necessidade de análise dos dados de toda organização.

Nenhuma decisão de remoção de dados neste ambiente deve ser de natureza transacional. De fato, são raras as situações em que dados são removidos. Um clássico exemplo

é a exclusão de dados de uma filial não mais existente. *Data Warehouses* são considerados não-voláteis pois matém dados desatualizados para análises históricas, descoberta de tendências e predições (BÉDARD et al., 2001).

Os dados contidos neste sistema são destinados a exploração e consultas por meio de ferramentas especializadas. A inclusão de novos dados pode ser feita regularmente, mas nunca a substutuição. Esta limitação de operações o caracteriza como sendo de ”somente leitura”. Inmon (2002) defente que ao contrário do que acontece em ambientes operacionais, as únicas atualização feitas sobre os dados destinados para análise são corretivas. Exemplos de ações corretivas são mudanças de rótulos, hierarquias, status ou permissões de acesso, e estas devem ser feitas sempre que houver necessidade.

O *Data Warehouse* pode ser alimetado por dados de diversas fontes, havendo entre elas diferentes semânticas, constantes, formatos e codificações (BÉDARD et al., 2001). Para que haja correta integração dos dados, procedimentos como reformatação de campos, calibração de constantes e mudanças de unidades e escalas são necessários. Segundo Aguiar (2010), este trabalho de transformação geralmente é difícil e demorado, principalmente se a codificação for manual. Para Bédard et al. (2001), a adesão de padrões e conceitos de interoperabilidade na implementação de sistemas minimizam problemas de integração de dados.

Os dados implantados no DW estão dispostos em diferentes níveis de detalhamento. Para este detalhamento dá-se o nome de granularidade. Segundo Bédard et al. (2001), diferentes níveis de granularidade permitem que analistas de suporte a decisões façam observações sob várias perspectivas, descobrindo inclusive a origem de cada fenômeno observado.

O principal objetivo do *Data Warehouse* é servir de base para SSDs, oferecendo facilidades na obtenção de seus dados e fornecendo-os em tempo hábil. Esta tecnologia é projetada para ser compatível com as principais classes de software para descoberta de conhecimento.

2.1.2 COMPONENTES

O *Data Warehouse* é composto por um conjunto de componentes com funções bem definidas dentro de uma só arquitetura. Estes são programados para funcionar em harmonia, conseguindo-se assim o correto funcionamento da tecnologia em questão.

Em um primeiro momento fica a impressão que esta solução pode ser conseguida

com um número reduzido de componentes, programando-os para desempenharem mais de uma função ao mesmo tempo. Este é o caso da base de dados operacionais, que em uma arquitetura erroneamente planejada manteria os dados destinados a análises.

Como confirma Bédard et al. (2001), devido a limitações tecnológicas seria inviável manter uma estrutura hibrida (modelo relacional e multidimensional) em um mesmo sistema, pois resultaria em tempo de resposta insatisfatório para ambas operações transacionais e de análise.

A solução para este problema adotada em arquiteturas de *Data Warehouse* é manter sistemas distindos: o primeiro orientado a transações para suprir necessidades do dia-a-dia; o segundo para apoiar operações de análise, sendo este alimentado pelos dados do primeiro.

2.1.2.1 BASE DE DADOS OPERACIONAIS

Um dos componentes do *Data Warehouse* é a base de dados operacionais. Sua função básica é registrar as transacões de négocio (KIMBALL et al., 1998). Dentro da arquitetura de um DW, sua função é de alimentar a base de dados para análises com seus dados operacionais.

Bédard et al. (2001) define a base de dados operacionais como sendo destinada ao acesso, armazenamento, atualização, checagem de integridade, segurança e consultas simples sobre dados. Segundo Kimball et al. (1998), este componente difere do DW em suas necessidades, clientes, estruturas e ritmos.

Para este sistema o foco reside em mínima redundância e máxima integridade. Em sua estrutura normalizada, a cada consulta um processamento relativamente grande é feito para uma pequena quantidade de dados, mas ainda assim apresenta desempenho aceitável para operações do dia-a-dia (BÉDARD et al., 2001).

Kimball et al. (1998) define bases de dados operacionais como sistemas voláteis, pois não matém dados históricos, mas sim dados atualizados que suportem operações cotidianas. Bédard et al. (2001) explica que conforme estes ficam desatualizados, são substituídos, destruídos ou arquivados.

Como sua função não é servir de base para análises, mantém apenas dados de baixa granularidade de acordo com o que foi registrado pelos sistemas que o serve. Estes sistemas são denominados OLTP, e realizam operações transacionais sobre esta base de dados.

Como já mencionado, o foco deste sistema não é servir de base para decisões estratégicas, e geralmente mantém dados de um escopo reduzido, pertinente a departamentos ou divisões em uma empresa. Portanto não são orientados a organização, mas sim orientados a aplicação, uma vez que são projetados para atender a aplicações específicas.

2.1.2.2 STAGING AREA

Staging Area é a área na qual os dados provenientes das bases de dados operacionais são processados, a fim de se obter uma estrutura e qualidade adequada exigida pelo *Data Warehouse*. Para Kimball et al. (1998), *Staging Area* também pode ser definida como um conjuto de processos, como de integração, limpeza, transformação, combinacão, resolucão de duplicidade, arquivamento e preparacão dos dados que serão depositados no DW.

Os dados ao passar por estes procedimentos podem ser armazenados sobre várias tecnologias. Kimball et al. (1998) confirma ao dizer que ”a *Staging Area* nem sempre baseia-se em tecnologia relacional”, mas deixa claro a preferência da construção sobre bancos de dados relacionais ao afirmar que ”em alguns casos, os responsáveis pela *Staging Area* sentem-se mais confortáveis realizando os processos de limpeza, transformação e combinacão sobre estruturas normalizadas”.

O armazenamento dos dados nesta etapa não necessariamente é feito de forma centralizada, mas pode ser distribuído em vários repositórios fisicamente dispersos. Kimball et al. (1998) comenta que a prática de armazenamento distribuído é bastante comum, ainda que na etapa de preparação dos dados.

As operações realizadas nesta etapa dividem-se em extração, transformação e carga de dados. A extração de dados compreende em entender os dados contidos nas bases de dados operacionais, extrair destes o que for de interesse e implantá-los no ambiente de *Staging Area* (KIMBALL et al., 1998). Antes da etapa de extração, análises sobre os dados operacionais devem ser realizadas a fim de identificar as necessidades de transformações.

A obtenção dos dados operacionais pode ser dificultada caso não haja documentações que explanem seus significados. Tais documentações possuem informações que correspondem aos metadados, comumente utilizados na construção de *Data Warehouses*.

Uma vez carregados e integrados, diversas transformações podem ser realizadas sobre eles. Segundo Kimball et al. (1998), algumas das possíveis operações desta etapa são:

- Remoção de dados de pouca ou nenhuma utilidade para decisões estratégicas. A decisão dos quais serão mantidos no *Data Warehouse* é feita por especialistas no negócio, e não por profissionais técnicos, responsáveis pela implementação da tecnologia;

- Resolução de conflitos de domínio. Esta transformação aplica convenções sobre os dados e é bastante comum em situações onde sistemas OLTP são construidos desconsiderando normas ou padrões. Um tradicional exemplo desta transformação é a definição dos possíveis valores ”masculino”e ”feminino”para os dados que representam o sexo de uma pessoa;

- Tratamento de dados faltantes. Sistemas especialistas em predição, conhecidos como ferramentas para mineração de dados, podem ser utilizados para completar os dados não preenchidos no conjunto;

- Correção de erros de digitação. Erros ortográficos são corrigidos em massa, garantindo uniformidade e correta representação dos dados. A qualidade das informações fornecidas pelo DW a Sistemas de Suporte a Decisões está diretamente ligada a qualidade dos dados que as compõem, e a qualidade destes é negativamente influenciada por erros ortográficos;

- Combinação de dados provenientes de diversas fontes: Os dados pertencentes a um departamento são completados pelos de outros departamentos, formando assim um conjunto completo que representa a empresa em sua totalidade;

- Manutenção de integridade referencial: São executadas estratégias que garantem a integridade dos dados, como uso de chaves *Surrogate Key* (SK) que substituem os atributos de chave primária originalmente utilizados em modelos relacionais.

Após a etapa de transformação ser concluída, os dados são carregados no *Data Warehouse* em um processo conhecido por carga de dados (KIMBALL et al., 1998). Esta é a última operação sobre os dados da *Staging Area*, mas vale notar que neste ponto os dados já devem estar estruturados de acordo com um modelo adequado para o DW.

2.1.2.3 SERVIDOR DE APRESENTAÇÃO

Entende-se como servidor de apresentacão a máquina ou sistema sobre o qual é construído o *Data Warehouse* (KIMBALL et al., 1998). Juntamente com as bases de dados

operacionais e a *Staging Area*, forma o sistema necessário para o funcionamento do DW.

2.1.2.4 DATA MART

Para Kimball et al. (1998), *Data Warehouses* são formados pela união de *Data Marts* (DM), cada um destes representando um processo de negócio em uma organização. Esta solução pode inclusive representar, não um, mas vários processos de negócio aglomerados em um só lugar, atendendo a um departamento ou grupo específico.

Kimball et al. (1998) defendem a abordagem de construção de DWs a partir de *Data Marts*. Desta forma, a arquitetura completa seria construída de forma incremental, de acordo com as necessidades e possibilidades da empresa. De acordo com Bédard et al. (2001), a construção de *Data Marts* requer menor investimento e esforços que de DWs. Kimball et al. (1998) explica que tais vantagens existem pois nesta solução o escopo dos dados é limitado, uma vez que não é projetada para atender a toda organização.

Na etapa de transformação que ocorre antes da carga deste componente, os dados são estruturados de acordo com o modelo multidimensional, o mesmo encontrado em *Data Warehouses*. Isto acontece pois este sistema é igualmente explorado por Sistemas de Suporte a Decisões que exigem os benefícios oferecidos por este modelo.

O conceito de *Buss Arquitecture* é introduzido por Kimball et al. (1998), e diz respeito a normas adotadas na construção de DMs. Pode ser entendido como boas práticas de implementação, onde a construção acontece a partir de padrões arquiteturais e são adotadas convensões para nomes de tabelas fato e dimensões. Kimball et al. (1998) complementa que quando adotada a estratégia de construção incremental do DW, a adoção de convensões em seus *Data Marts* é fundamental.

Segundo Inmon (2000), as principais áreas para as quais são construídos *Data Marts* são a financeira, vendas e relacionamento com o cliente.

2.1.3 ARQUITETURAS

Não existe uma regra quanto a disposição dos componentes em uma arquitetura para suporte de Sistemas de Suporte a Decisões. A presença ou não de *Data Marts* ou de um *Data Warehouse* que atenda a toda organização será definida de acordo com os recursos financeiros, o prazo disponível para implementação, a capacidade técnica da equipe de construção e outros fatores que podem variar de empresa para empresa.

Um dos possíveis arranjos de componentes é apresentado por Bédard et al. (2001) em uma arquitetura conhecida por corporativa ou genérica. Nesta há um único servidor que é acessado diretamente pelos clientes através de ferramentas especializadas. Na arquitetura corporativa os dados são extraídos das fontes heterogêneas, transformados na *Staging Area* e então carregados no *Data Warehouse*.

Na arquitetura denominada federada (BÉDARD et al., 2001), os dados operacionais são integrados e processados na *Staging Area*, carregados no DW e então distribuídos entre *Data Marts*. Consultas que envolvem dados de toda organização podem ser feitas no *Data Warehouse*, já os *Data Marts* atendem setores individualmente. Nesta arquitetura, a granularidade dos dados dos DMs é necessariamente igual ou maior que do *Data Warehouse*.

Vale notar que na arquitetura federada, não pode-se dizer que o *Data Warehouse* é composto e montado a partir de vários *Data Marts*, mas é correto afirmar que estes foram populados por subconjuntos de dados do servidor principal.

A arquitetura de várias camadas proposta por Bédard et al. (2001) possui dois DWs, sendo o primeiro composto por dados de mesma granularidade que das bases de dados operacionais, e o segundo com dados mais agregados. Assim como na arquitetura federada, os *Data Marts* são populados a partir dos dados do *Data Warehouse*, neste caso o segundo.

É possível ter um conjunto de DMs independentes, cada um atendendo a departamentos ou grupos específicos. Nesta arquitetura não há um *Data Warehouse* como intermediário, sendo desta forma os dados carregados diretamente nos servidores departamentais após um processo de transformação (BÉDARD et al., 2001).

2.1.4 MODELO MULTIDIMENSIONAL

O modelo de dados multidimensional é uma estrutura que visa o ganho de performance em consultas sobre grandes conjuntos de dados. A oferta de respostas rápidas somado ao suporte de consultas dinamicamente construídas, faz deste um modelo mais adequado para *Data Warehouses* e *Data Marts* que o modelo relacional [2] (KIMBALL et al., 1998).

Outro benefício conseguido com sua utilização é a simplicidade estrutural que seus dados se encontram. Diferente do modelo relacional que devido a sua normalização pode resultar em enormes quantidades de tabelas, possui uma estrutura pequena, simples e de fácil compreensão. Kimball et al. (1998) comenta que a estruturação de DWs de acordo com o modelo multidimensional, trás benefícios de inteligibilidade dos dados e flexibilidade a mudanças.

Sua denormalização replica em redundância nas tabelas dimensão. Para Inmon (2000), estes dados redundantes não representam impácto em *Data Warehouses* ou *Data Marts*, pois correspondem a menos de 1% do volume total ocupado no sistema. Inmon (2000) considera a denormalização das tabelas como o principal responsável pela performance oferecida.

No modelo multidimensional os dados de negócio são representados por um tipo de cubo de dados (KIMBALL et al., 1998). Nesta estrutura, as células representam medidas, enquanto que as arestas do cubo representam as dimensões nas quais as medidas se enquadram.

2.1.4.1 FATO

O fato é um elemento importante no modelo multidimensional. Para Kimball e Ross (2002), o termo fato representa uma medida de negócio. Já Kimball et al. (1998) conceitua os fatos como sendo algo desconhecido ou a ser observado.

Os fatos são os objetos de atenção das análises realizadas por qualquer Sistema de Suporte a Decisões, e estes podem se representados por valores numéricos, textuais ou típos específicos de dados.

Diferente dos atributos presentes nas tabelas dimensão, os fatos não são utilizados

[2] Modelo relacional é um modelo de dados amplamente utilizado em bases operacionais. Este introduz conceitos de atributo, chave primária e estrangeira e relações entre tabelas.

para descrever algo, mas sim para representar uma medida de de negócio. Kimball et al. (1998) defende que um valor deve ser tratado como fato caso varie com certa frequência, o que geralmente ocorre com dados numéricos.

Segundo Kimball e Ross (2002), não se deve haver redundância nas informações textuais das tabelas fato. Caso alguma medida textual repita, é provável que pertença a alguma tabela dimensão e sirva para descrever um fato. Bédard et al. (2001) complementa que fatos são valores imprevisíveis, por isso raramente são do tipo textual.

A chave primária em uma tabela fato é composta a partir de um conjunto de atributos cujos valores referenciam as tabelas dimensão. Tais valores são tipicamente números inteiros, e garantem a integridade referencial no modelo multidimensional (KIMBALL et al., 1998).

2.1.4.2 DIMENSÃO

As tabelas dimensão e seus atributos ditam as possíveis perspectivas de visualização dos dados. Seus atributos podem passar de 50 ou 100, sendo textuais ou numéricos e tendo como função descrever atividades de negócio (KIMBALL; ROSS, 2002; KIMBALL et al., 1998).

Kimball e Ross (2002) considera as tabelas dimensão como o ponto de entrada para a tabela fato e a interface para os usuários do *Data Warehouse*. Para ele, a capacidade de exploração do conjunto de dados está relacionada a robustez destas tabelas.

Tabelas dimensão são tipicamente denormalizadas, havendo assim redundância em seus dados. Mas aínda assim seus conjuntos de dados são consideravalmente menores que de tabelas fato. Kimball e Ross (2002) relata que o volume da união dos conjuntos de todas dimensões geralmente não representa 10% do volume total dos dados de um *Data Mart*.

Segundo Kimball et al. (1998), não existem restrições quanto ao número de dimensões em um modelo multidimensional. Este número está diretamente ligado as necessidades de análise e realidade do negócio.

Ao contrário dos fatos e geralmente sendo do tipo texto, os atributos das tabelas dimensão são valores conhecidos que tem como função descrever características de algum evento (KIMBALL et al., 1998). Outras funções incluem servir como filtros de pesquisa, meios de agrupamento e rótulos das informações. Kimball e Ross (2002) deixa claro a importância de se ter atributos de qualidade ao afirmar que o poder do *Data Warehouse* é proporcional a

qualidade destes atributos descritivos.

A garantia da integridade nas tabelas dimensão é feita por meios de um atributo que serve como chave. Este é conhecido por *Surrogate Key* (SK), e é referenciado em tabelas fato ou mesmo em outros níveis hierarquicos caso o esquema *Snowflake* seja adotado (KIMBALL; ROSS, 2002). Kimball et al. (1998) comenta que a chave original de um registro nunca deve ser utilizada, mas sim o SK.

Quando o esquema *Snowflake* é utilizado, estruturas hierarquicas são formadas nas dimensões. No relacionamento entre dois níveis em uma hierarquia, o que representa menor granularidade em seus dados é chamado *child*, já o outro é denominado *parent* (ZIMÁNYI; MALINOWSKI, 2004).

O nível hierárquico diretamente ligado a tabela fato é referido como folha, enquanto que o nível da outra extremidade, o qual representa o mais alto nível de agregação de dados, é chamado raiz (ZIMÁNYI; MALINOWSKI, 2004).

Segundo Zimányi e Malinowski (2004), os nível das dimensões são compostos por membros, representados fisicamente por tuplas em tabelas. A cardinalidade representada em um modelo conceitual apresenta o número mínimo e máximo de relações entre membros de diferentes níveis.

Para as ligações entre tabelas fato e folhas de diversas hierarquias dá-se o nome de *fact relationships*. Zimányi e Malinowski (2004) explica que em consultas SQL tais relacionamentos correspondem aos operadores *join*.

2.1.4.3 INTEGRIDADE REFERENCIAL

A integridade referencial existe no momento em que as chaves das tabelas fato referenciam corretamente o conteúdo das tabelas dimensão (KIMBALL; ROSS, 2002). *Surrogate Key* é o atributo chave utilizado para unir fatos e dimensões. Também conhecido como *meaningless key*, *integer key*, *nonnatural key*, *artificial key* e *synthetic key*, é do tipo inteiro e atribuído de forma sequencial.

Segundo Kimball e Ross (2002), chaves primárias utilizadas em bases de dados operacionais não devem ser utilizados no modelo multidimensional. Além disso, não deve ser possível afirmar algo sobre uma medida observando apenas sua chave.

Um dos benefícios do uso de SKs é a garantia de consistência dos dados. Caso SKs não sejam utilizados, mudanças em dados operacionais, como o reaproveitamento de atributos identificadores podem comprometer a integridade do *Data Warehouse* (KIMBALL; ROSS, 2002). Outro benefício é a eliminação de inconsistências na integração dos dados de diferentes fontes, caso hajam atributos identificadores duplicados (KIMBALL; ROSS, 2002).

Segundo Kimball e Ross (2002), o tipo de dado ideal para SKs é o integer, pois é econômico em espaço em disco ocupando apenas 4 bytes e pode representar aproximadamente 2 bilhões de números positivos.

2.1.4.4 ESQUEMAS

Segundo (KIMBALL et al., 1998), o esquema *Star* consiste em uma única tabela fato que se relaciona com tabelas dimensão através de chaves SK. Nesta estrutura, a identificaçao dos fatos é feita pela união de seus atributos SK. A principal vantagem oferecida é o ganho de performance na exploração do conjunto de dados. Este benefíco é conseguido pois poucas tabelas são utilizadas simplificando consideravelmente as consultas.

A simplicidade do esquema *Star* é o resultado de sua denormalização. Para Kimball e Ross (2002), tal benefício é perdido conforme novas tabelas são encadeadas nas dimensões, formando hierarquias. Seguindo o caminho inverso do modelo relacional, sempre há redundância no dados das tabelas dimensão.

Quando há em um mesmo esquema mais de uma tabela fato, dá-se o nome de *Constellation*. Neste caso, dimensões podem ser compartilhadas entre eles formando uma arquitetura poderosa e robusta.

O esquema *Snowflake* pode ser considerado uma extensão do esquema *Star*, pois em seu centro uma tabela fato também está presente, e esta é interligada em suas dimensões por atributos identificadores.

Sua principal característica é a normalização de suas tabelas, formando assim estruturas hierarquicas de pelo menos dois níveis em suas dimensões. O principal benefício conseguido com a normalização é a baixa redundância dos dados. Embora haja um ganho de espaço em disco com este esquema, em *Data Warehouses* ou *Data Marts* este ganho é insignificante. Kimball e Ross (2002) explica que o conjunto de dados redundantes em uma dimensão chega a ser inferior a 1% do espaço ocupado por todo esquema (KIMBALL; ROSS,

2002).

Embora um ganho de espaço seja conseguido, o esquema *Snowflake* deve ser utilizado com cautela, uma vez que aumenta o uso das custosas operações SQL[3] *join*. Kimball e Ross (2002) alertam que a análise realizada sobre um modelo normalizado tende a ser complexa, e completa afirmando que o principal prejuízo resultante da normalização é a baixa performance das consultas.

2.1.5 GRANULARIDADE DOS DADOS

Um dos aspectos mais importantes ao se considerar em desenhos de *Data Warehouse* e *Data Mart* é o nível de granularidade dos dados (INMON, 2002). Granularidade é o nível de detalhamento ou sumarização dos dados. Quantos mais detalhes, menor a granularidade.

A correta escolha do nível de granularidade dos dados que serão persistidos no *Data Warehouse* é fundamental, visto que afeta diretamente no volume total persistido e o tipo de consultas que poderão ser efetuadas sobre este sistema. Inmon (2002) reforça a importância deste conceito ao afirmar que a chave da reusabilidade do *Data Warehouse* é o correto nível de granularidade nele presente, pois permite que a solução seja utilizada de diferentes formas por diferentes pessoas.

O nível adequado de granularidade influencia na aceitabilidade a mudanças pelo sistema. Para Inmon (2002), o acerto neste quesito permite que novos requisitos quanto a consultas sejam satisfeitos com maior facilidade.

Inmon (2002) afirma que o volume de dados não é o único fator a se considerar na definição de granularidade. Quanto maior o conjunto de dados, maior a capacidade de processamento necessária para atender a consultas em um tempo aceitável. Inmon (2000) comenta que a granularidade pode ser ajustada caso haja problemas quanto ao volume.

Os dados provenientes de sistemas transacionais são altamente detalhados. Pelos motivos acima citados, é fundamental que estes sejam processados afim de se obter um nível de sumarização adequado antes de serem implantados em um *Data Warehouse*.

[3] SQL é uma linguagem multiparadigma utilizada para manipulação de dados em banco de dados relacionais.

2.1.6 METADADOS

Metadados são dados sobre os dados. São de grande importância para *Data Warehouses* e *Data Marts*, pois ajudam os analistas de suporte a decisões a explorar toda potencialidade do sistema (INMON, 2002). Com o auxílio de metadados, o analista pode navegar entre todas as possibilidades que o conjunto de dados oferece. Inmon (2002) complementa que seu uso agiliza as consultas, pois facilita a interpretação dos dados de negócio.

Kimball e Ross (2002) define metadados, no contexto de ambientes para suporte de decisões, como informações que não representam dados de negócio. Complementa afirmando que funcionam como uma enciclopédia para *Data Warehouses*.

Inmon (2002) explica que com ajuda de metadados, o analista de suporte a decisões pode compreender quais dados estão ou não alí presentes, além de quais são as possibilidades de consultas neste sistema.

Em *Data Warehouses* geralmente são utilizados para apresentar:

- A estrutura dos dados para o programador;
- A estrutura dos dados para o analista de suporte a decisões;
- Informações sobre as fontes dos dados;
- As transformações realizadas sobre os dados;
- Informações sobre o modelo de dados;
- O histórico das extrações;

Segundo Kimball e Ross (2002), a construção dos metadados é muitas vezes negligenciada em um projeto de *Data Warehouse*. Isto acontece pela falta de conhecimento da equipe técnica sobre da sua importância, influenciando negativamente nas análises e manutenção do sistema.

Os metadados atendem os usuários do *Data Warehouse* para finalidades técnicas, administrativas e de negócio. Uma vez que os dados foram integrados e se encontram na *Staging Area*, facilitam os processos de carga e transformação, apresentando normas sobre fatos e dimensões, regras de transformação, limpesa de dados e definições de agregação (KIMBALL; ROSS, 2002).

De acordo com Kimball e Ross (2002), em *Data Warehouses* metadados também mantém informações quanto a permissões de acesso e privilégios, índices, tabelas do sistema e configurações de particionamento.

2.1.7 CONSTRUÇÃO

A construção do *Data Warehouse* é custosa em valores e dificuldade de implementação, porém vantajosa pois é feita do início uma única vez, e uma vez feita tráz benefícios como flexibilidade e reusabilidade (INMON, 2002).

Para Kimball et al. (1998), Inmon (2000), a construção do *Data Warehouse* deve ser realizada em um processo iterativo e incremental, iniciando com poucos requisitos. Inmon (2002) explica que tal abordagem de desenvolvimento permite que o analista de suporte a decisões tenha seu primeiro contato com a ferramenta mais cedo, antes de um maior investimento, inclusive financeiro.

Em uma abordagem de desenvolvimento iterativo e incremental, é importante atentar-se ao escopo do projeto na primeira iteração. A solução a ser construída deve ser pequena e fácil o suficiente para ser realizada, mas grande a ponto de ser significante (INMON, 2002).

Segundo Inmon (2000), o processo de construcão de um *Data Warehouses* difere da construção de bases de dados operacionais. Este sistema deve ser projetado para suportar contínuas mudanças mantendo a confiabilidade exigida em ambientes de suporte a decisões.

Inmon (2002) afirma que a metodologia de desenvolvimento cascata[4] não pode ser utilizada em projetos de *Data Warehouse*, visto nem todos requisitos podem ser identificados em um momento inicial. Em contrapartida, um método de construção ”tentativa e erro”geralmente é adotado.

Kimball et al. (1998) apresenta duas estratégias de construção de *Data Warehouses*, sendo elas a *bottom-up* e *top-down*. A qual será escolhida vai de acordo com as necessidades e possibilidades da empresa.

Em uma estratégia de construcão *bottom-up*, o DW é construído pela união dos *Data Marts* já implementados, estando eles dispersos e não relacionados entre sí (KIMBALL et al., 1998). Nesta forma de implementacão, a união dos DMs acontece através de suas dimensões.

[4]Na metodologia de desenvolvimento cascata, todos os requisitos são coletados antes da etapa de implementação (INMON, 2002)

Kimball et al. (1998) afirma que isto só é possível quando há uma conformidade quanto ao significado destas dimensões nas estruturas que elas se encontram.

Os setores que geralmente são priorizados com a primeira versão do *Data Warehouse* são o financeiro, marketing, vendas e ocasionalmente os setores de fábrica e de interesses atuariais.

Em uma estratégia de construcão *top-down*, antes do conjunto total de dados ser distribuído entre os vários *Data Marts*, é concluída a construcão do DW que terá todos os dados da organizacão (KIMBALL et al., 1998).

Bédard et al. (2001) conclui que independente da forma de construção ou estratégia adotada, as bases de dados operacionais não devem sofrer alterações ao longo do processo de *data warehousing*[5].

2.2 SPATIAL DATA WAREHOUSE

Segundo Karabegovic e Ponjavic (2012) a maioria dos eventos de negócio registrados em *Data Warehouses* tradicionais possuem uma referência para onde aconteceram. Malinowski e Zimányi (2004) explicam que estas referências encontram-se em dimensões e são representadas por dados alfanuméricos.

Acontece que esta forma de referenciar localizações não permite que o *Data Warehouse* tradicional ofereça respostas para perguntas como ”o quão longe os trabalhadores iriam viajar de seu trabalho a suas residências?”, ”como escolher a melhor localização para um novo negócio?”ou ”quais edifícios estão em potenciais áreas de alagamento?”(MALINOWSKI; ZIMÁNYI, 2004).

Neste sentido, o *Data Warehouse* tradicional permite que Sistemas de Suporte a Decisão respondam perguntas onde o ”quem”, ”o que”, ”quando”e ”por que”estão presentes. Mas tal sistema não é preparado para responder com precisão perguntas onde a localização é o grande ponto de interrogação (KARABEGOVIC; PONJAVIC, 2012).

A forma de referenciamento geográfico presente em *Spatial Data Warehouses* - SDW abre um novo leque de possibilidades para Sistemas de Suporte a Decisão. A análise de

[5] Processo de construção de uma arquitetura de *Data Warehouse*, estando o servidor principal presente ou apenas *Data Marts* independentes.

negócio é potencializada com respostas mais precisas, dando ao analista de suporte a decisões uma visão mais clara do meio em que a organização se encontra. Ao se utilizar de *Spatial Data Warehouses*, novos relacionamentos e padrões podem ser descobertos sobre os dados (KARABEGOVIC; PONJAVIC, 2012).

Segundo Bédard et al. (2001), Sistemas de Informação Geográfica (SIG) por sí só atendem bem as operações transacionais, mas quando utilizados para descoberta de conhecimentos geográficos, devem ser substituídas por Sistemas de Suporte a Decisão baseados em *Spatial Data Warehouses*.

O SDW surgiu da união de conceitos de *Data Warehouse* com base de dados espaciais (GARG; MITHAL, 2010). Ao unir dados georreferenciados com dados tradicionais de negócio, são conseguidas novas percepções e decisões melhor embasadas podem ser tomadas. Para Bédard et al. (2001), a fusão destas tecnologias resultou em uma solução sofisticada, capaz de atender satisfatoriamente softwares para descoberta de conhecimentos geográficos.

Esta solução oferece uma visão unificada dos dados provenientes de bases geográficas (BÉDARD et al., 2001). Suas fontes de dados podem ser heterogêneas, havendo assim a necessidade de transformações em seus dados para garantia de integridade e adequação em sua estrutura.

De acordo com Karabegovic e Ponjavic (2012), os dois fatores que estimularam o uso de *Spatial Data Warehouses* nas empresas foram a crescente disponibilidade de dados georreferenciados relacionados ao negócio e a necessidade de melhores percepções analíticas. Entre os dominios do SDW se encontram a logística, previsão do tempo, segurança, detecção de mudanças climáticas e monitoramento de saúde (GARG; MITHAL, 2010).

Segundo Bédard et al. (2001), na construção desta solução, quando comparado ao *Data Warehouse* tradicional, novos cuidados devem ser tomados. Um exemplo é a verificação quanto a integridade espacial dos dados, e se estes estão topologicamente corretos. Outros cuidados incluem verificar se o sistema de referências espaciais dos dados a serem carregados é suportado pela tecnologia a ser utilizada, se a geometria dos objetos é apropriada para vários níveis de granularidade e se a sobreposição dos mapas importados está topológicamente correta.

2.2.1 DADOS

Shneider (1997) define dados espaciais ou geométricos como sendo relacionados ao

espaço. Estes podem ser armazenados, consultados e manipulados em sistemas como *Spatial Data Warehouses* e Bases de Dados Espaciais.

Em sistemas de *Spatial Data Warehouse* é comum operações de agregação de dados espaciais, na construção de medidas espaciais ou níveis hierárquicos de dimensões espaciais (GARG; MITHAL, 2010). Entre as possíveis operações de agregação sobre estes estão a caixa delimitadora ortogonal, mínimo, união geométrica, intersecção geométrica, centróide, centro de gravidade, centro de massa, índice *Nearest Neighbor* e *Equi Partition* (SHEKHAR et al., 2002).

Dados espaciais, também conhecidos por objetos espaciais, tem como função representar geometrias, sendo estas pontos, linhas ou polígonos (SHNEIDER, 1997). Tais dados são capazes de representar estruturas, extensões e formas de objetos no espaço.

Entre os três tipos de dados, o ponto é o mais simples sendo formado por um par de coordenadas x e y em um plano. Este tipo de objeto espacial pode representar em mapas incidências de mortalidades, acidentes de trânsito ou assaltos. Em caso de generalização dos objetos espacialmente referenciados onde a representação do fenômeno é simplificada, pontos podem ser utilizados para representar cidades ou outros objetos complexos.

Segundo Shneider (1997), um objeto espacial do tipo linha é composto por um ou mais segmentos de linha, onde tais segmentos correspondem a conexão entre dois pontos em um plano. Este tipo de objeto é caracterizado por sempre constituir uma cadeia aberta de linhas. O objeto espacial polígono é uma cadeia de linhas, onde o primeiro e último segmento encontram-se unidos através de um ponto.

Na literatura (SHNEIDER, 1997), o termo objeto espacialmente referenciado é utilizado para descrever um objeto ou fenômeno no mundo real. Tais fenômenos podem ser relacionados com um ou mais objetos espaciais em um sistema. Um exemplo de objeto espacialmente referenciado é uma cidade, a qual seria referenciada por um objeto espacial do tipo polígono que representaria seus limites geográficos. Esta cidade poderia ser referenciada ao mesmo tempo por objetos não espaciais, de formato alfanumérico, que representariam seu nome e população.

Objetos espacialmente referenciados são suscetíveis a mudanças ao longo do tempo. Exemplos de mudanças dos fenômenos representados são o desvio do curso de um rio ou a mudança do nome de um bairro. As propriedades destes objetos correspondem às características dos fenômenos. A primeira classe destas propriedades é conhecida por não geométrica, constituindo de valores alfanuméricos para representar nomes e quantidades. A

segunda é conhecida por geométrica, constituindo de dados espaciais para representar formas, localizações, extensões e medidas (SHNEIDER, 1997).

Um conjunto de objetos espaciais ou geográficos onde haja relação topológica entre eles, é denominado objeto de estrutura (SHNEIDER, 1997). Um exemplo deste é a malha rodoviária brasileira, composta por estradas representadas por objetos espaciais do tipo linha.

Diferentemente dos dados alfanuméricos, a estrutura dos dados espaciais é complexa, possibilitando que estes sejam interpretados como objetos complexos, compostos por vários componentes (SHNEIDER, 1997). Um exemplo é um dado espacial do tipo polígono, formado por um conjunto de linhas, capaz de representar um território ou outro objeto complexo.

2.2.2 MODELO MULTIDIMENSIONAL

Assim como no *Data Warehouse* convencional, a estrutura multidimensional também é utilizada em *Spatial Data Warehouses* (GARG; MITHAL, 2010). Sistemas de Suporte a Decisões que se apoiam nesta tecnologia, também têm como requisitos baixo tempo de resposta, alta adaptabilidade e flexibilidade em suas consultas, naturalmente satisfeitos com o uso do modelo multidimensional.

No que diz respeito a diferenças para o modelo multidimensional utilizado em *Data Warehouses* convencionais, tabelas dimensão e fato passam a suportar novos tipos de dados, estes denominados geométricos, comuns em *Spatial Data Warehouses* (GARG; MITHAL, 2010).

Bédard et al. (2001) explica que não necessariamente devem existir nesta solução ao mesmo tempo dimensões e fatos com medidas espaciais. A espacialidade neste caso será representada de acordo com as necessidades do negócio. Malinowski e Zimányi (2004) comentam que quando possível, a adição de dimensões espaciais a um modelo multidimensional deve ser feita, uma vez que extende as possíveis formas de visualização dos dados georreferenciados.

Os esquemas *Star*, *Constellation* e *Snowflake* também são utilizados em *Spatial Data Warehouses*. O último mencionado aparece com mais frequência nesta tecnologia do que em *Data Warehouses* convencionais, pois sua estrutura normalizada possibilita mais clara representação entre as relações topológicas dos dados persistidos.

Para Malinowski e Zimányi (2004), modelos multidimensionais para *Spatial Data Warehouses* devem atender aos seguintes requisitos:

- Ser simples e inteligível;
- Independente de implementação;
- Suportar multiplas hierarquias espaciais, estando estas explícitas;
- Ser capaz de acomodar dados de diferentes granularidades;
- Suportar hierarquias espaciais irregulares;
- Suportar agregações geométricas e temáticas;
- Suportar operações SOLAP.

2.2.2.1 FATOS

Em tabelas fato de *Spatial Data Warehouses*, o conceito de medidas espaciais é introduzido. Medidas espaciais podem conter valores numéricos calculados por operadores espaciais ou topológicos, ou então valores espaciais, representando relacionamentos entre as dimensões espaciais (BÉDARD et al., 2001).

Ferramentas de análise para exploração destes dados georreferenciados são capazes de manipular este conjunto de várias formas, como exemplo alterando as perspectivas de análise através de operações *drill-up* e *drill-down*. Para tanto, agregações sobre as medidas espaciais devem ser realizadas, as quais podem ser categorizadas em distributivas, algébricas e holísticas.

Entre as operações de agregação de medidas numéricas estão a média, variância, desvio padrão, soma, contagem, mediana, maior frequência e ranking. Já entre as operações de agregação de medidas geométricas estão a união geométrica, intercecção geométrica, centro de gravidade, índice *Nearest Neighbor* e centro de n pontos geométricos (MALINOWSKI; ZIMÁNYI, 2004). Na implementação do modelo, caso nenhuma operação de agregação seja especificada, a soma é definida por padrão.

2.2.2.2 DIMENSÕES

Na tecnologia de SDW, além de dimensões com dados textuais e numéricos, dimensões com dados espaciais estão presentes. Estas são chamadas dimensões espaciais, e de acordo com seus dados, são categorizadas em três tipos (BÉDARD et al., 2001).

O primeiro armazena valores quantitativos, como referências espaciais em um sistema de coordenadas X, Y (BÉDARD et al., 2001). Nestas dimensões são armazenados dados geométricos como pontos, linhas ou polígonos, representando fenômenos no espaço.

No segundo tipo de dimensão espacial são armazenados apenas dados qualitativos que descrevem fenômenos, como o nome de cidades ou rodovias. Neste tipo de dimensão, dados de baixa granularidade e também suas agregações são textuais (BÉDARD et al., 2001).

O terceiro tipo armazena em uma mesma estrutura dados quantitativos que representam a localização exata do fenômeno, e dados qualitativos utilizados para identificar o fenômeno (BÉDARD et al., 2001). Outra prática para esta dimensão é representar níveis de menor granularidade com dados geométricos, e os níveis mais altos da hierarquia com dados nominais.

Ao se adotar o esquema *Snowflake* em SDWs, caso o encadeamento aconteça entre tabelas que possuam dados espaciais, a união destas passa a se chamar hierarquia espacial. Malinowski e Zimányi (2004) define hierarquia espacial como sendo uma estrutura de pelo menos um nível espacial.

Malinowski e Zimányi (2004) define um nível espacial como sendo componente de uma hierarquia espacial. Este pode ser representado fisicamente por uma tabela em um *Spatial Data Warehouse*. O nível espacial contém dados que representam geometrias, sendo elas pontos, linhas, polígonos ou conjuntos das formas mencionadas.

Entre as relações topológicas existentes entre os elementos espaciais destes níveis estão: um objeto ”contém o outro”, ”é igual ao outro”, ”há intercecção entre eles”e ”um objeto sobrepõem o outro”(MALINOWSKI; ZIMÁNYI, 2004).

A união entre duas dimensões espaciais é feita atravéz da tabela fato. Segundo Malinowski e Zimányi (2004), esta união também conhecida como *spatial fact relationship*, só é possível caso haja relacionamento topológico entre as dimensões.

2.2.3 ESTRATEGIAS DE CARGA

Segundo Bédard et al. (2001), de acordo com os tipos de consultas que serão realizadas sobre os dados do *Spatial Data Warehouse*, diferentes estratégias podem ser adotadas na carga das tabelas fato. Tais estratégias estão relacionadas ao pré-processamento das medidas espaciais, e caso a carga dos dados seja feita de maneira equivocada, o resultado virá em consultas com baixa performance.

A primeira alternativa é atribuir às medidas espaciais um conjunto de ponteiros, os quais referenciam dados espaciais. Desta forma, o custo computacional para a carga de dados no *Spatial Data Warehouse* é baixo, pois não são realizadas pré-computações destas medidas. Esta estratégia de carga de dados não é adequada quando operações de mineração de dados ou consultas OLAP serão realizadas. Isto ocorre pois as agregações necessárias serão feitas em tempo real, influenciando negativamente nos tempos de resposta (BÉDARD et al., 2001).

A segunda alternativa é realizar pré-computações sobre os dados espaciais afim de obter valores que representem cada conjunto. Os valores obtivos geralmente exigem pouca capacidade de armazenamento, uma vez que não visam representar com detalhes os valores agregados. Em situações em que maior qualidade é exigida nesta medida espacial, novos cálculos de agregação podem ser feitos antes da apresentação dos resultados (BÉDARD et al., 2001).

2.3 TECNOLOGIAS PARA DESENVOLVIMENTO

Segundo Inmon (2000), na escolha das tecnologias de hardware e softwares que serão utilizados para construção do *Data Warehouse*, cuidados quanto a robustez e riqueza em funcionalidades devem ser tomados. Comenta que as tecnologias escolhidas devem comportar o grande conjunto de dados acumulados ao longo dos anos.

Devida a importância do *Spatial Data Warehouse* em ambientes estratégicos, em sua construção as tecnologias utilizadas também devem ser cuidadosamente escolhidas. Bédard et al. (2001) comentam que as escolhas serão de acordo com as necessidades da empresa, e alguns pontos a serem avaliados para tais decisões são:

- Quanto a estrutura de rede sobre a qual o sistema será montado;
- Quanto a distribuição do *Data Warehouse*, podendo ser centralizado ou distribuído;
- Quanto aos típos de cliente (*thin-client* ou *thick-client*[6]).

Já para Inmon (2000), os fatores que influenciam na escolha das tecnologias a serem utilizadas na construção do *Data Warehouse* são:

- O volume de dados a serem acomodados;
- A velocidade necessária na captura dos dados;
- A história da organização;
- Quantos usuários usarão a solução;
- Quais tipos de análises serão realizadas sobre este produto;
- O custo da tecnologia.

2.3.1 SISTEMA GERENCIADOR DE BANCO DE DADOS

Segundo Inmon (2000), o sistema sobre o qual é construído o *Data Warehouse* tipicamente é o Sistema Gerenciador de Banco de Dados - SGBD ou um sistema otimizado para este fim.

Unilins (2012) define o SGBD como sendo um conjunto de *softwares* que facilitam a construção de estruturas de dados, a definição e a manipulaçao dos dados alí armazenados. Alguns dos benefícios que este sistema oferece são:

- Meios para controle de redundância dos dados. Quando necessário, estruturas normalizadas são utilizadas para evitar redundância dos dados;

[6]Em uma arquitetura cliente/servidor, o *thin-client* repassa os dados pela rede para que o processamento seja feito por um servidor. O *thick-client* em contrapartida realiza o processamento na própria máquina, repassando os resultados para o servidor.

- Facilidade de acesso aos dados, disponibilizando-os através de interfaces simples e intuitivas. *Softwares* que padronizam e facilitam seu acesso, denominados drivers, normalmente são disponibilizados pelos fabricantes da tecnologia viabilizando ainda mais o seu uso;
- Controle de consistência de seus dados, gerenciando regras de integridade referencial e unicidade de chaves identificadoras;
- Suporte de transações atômicas. Transações são tratadas como procedimentos indivisíveis, havendo assim garantia quanto a completeza das operações;
- Gerencia de acesso concorrente a seus registros, garantindo consistência em um ambiente acessado por múltiplos usuários;
- Controle de restrição de acesso aos dados, garantindo que usuários só tenham acesso aquilo que lhe dizem respeito.

Segundo Bédard et al. (2001), extensões espaciais podem ser utilizadas para complementar as funcionalidades dos SGBDs, permitindo que estes sirvam de base para construção de *Spatial Data Warehouses*. Tais sistemas quando utilizados para esta finalidade, podem adotar estruturas que sigam os paradigmas relacional, orientado a objetos ou hibrido, embora o primeiro seja o mais adequado.

O profissional responsável pela administração do SGBD é o *Database Administrator* (DBA), e entre suas funções encontram-se definir modelos e esquemas, permissões de acesso e privilégios, regras de integridade, coordenar migrações de tecnologias, criar testes de *backup* e estratégias para melhoria de desempenho em consultas (UNILINS, 2011).

2.3.2 POSTGRESQL

PostgreSQL é um Sistema Gerenciador de Bancos de Dados de código aberto, desenvolvido pela comunidade há mais de 15 anos. Segundo PostgreSQL (2013), possui uma arquitetura capaz de garantir a confiabilidade e integridade de seus dados. Atualmente é compatível com os principais sistemas operacionais, como Microsoft Windows, GNU/Linux e Unix AIX, BSD, HP-UX, Tru64, SGI IRIX, Solaris e Mac OS X.

Permite que arquivos binários como imagens, vídeos e sons sejam persistidos, além de suportar os formatos de dados integer, numeric, boolean, char, varchar, date, interval e timestamp, sendo estes padrões ISO SQL:1999 (POSTGRESQL, 2013).

Procedimentos podem ser implementados em várias linguagens de programação e então persistidos para futuro uso. Entre as linguagens suportadas por esta tecnologia estão o Java, C, C++, .Net, Perl, Python, Ruby, Tcl e ODBC (POSTGRESQL, 2013).

Segundo PostgreSQL (2013), este SGBD oferece controle de concorrência multiversionado, recuperação em um ponto no tempo, tablespaces, replicação assíncrona, transações agrupadas, cópias de segurança, otimizador de consultas e registrador de transações sequencial para tolerância a falhas.

O PostgreSQL mostra-se uma solução livre capaz de suportar grandes *Data Warehouses*. Segundo PostgreSQL (2013), é altamente escalável tendo como limíte máximo por tabela 32 terabytes de armazenamento. Seus bancos de dados possuem tamanho ilimitado, assim como o número de linhas e índices de suas tabelas. As limitações deste sistema não representam impedimentos para seu uso em ambientes de *Data Warehouse*, sendo elas:

- Tamanho máximo de 1GB para cada campo.
- Tamanho máximo de 1.6TB por tupla.
- Máximo de 250 ou 1600 colunas por tabela, dependendo do tipo utilizado.

2.3.3 POSTGIS

PostGIS é um *software* de código aberto disponível para *download* em sua página oficial. Seu projeto é suportado por uma comunidade ativa, a qual oferece serviços de resolução de problemas de novas versões e ajuda de uso. Em seu *website* há referências para serviços diferenciais de consultoria, os quais são pagos.

Seu objetivo é servir como um extensor espacial para o *software* de código aberto PostgreSQL. Funciona como um *plugin* para este SGBD, adicionando poderosas funcionalidades para tratamento de espacialidade dos dados (POSTGIS, 2013).

Com sua utilização como complemento, o PostgreSQL passa a suportar novos tipos de dados, como *geometry*, *geography* e *raster*, utilizados para representar espacialidade. Outro

benefício que traz é a inclusão de índices para objetos espaciais, permitindo que o SGBD atenda ao requisito de baixo tempo de resposta (POSTGIS, 2013).

O PostGIS permite que operações como *splicing*, *dicing*, *morphing*, reclassificação, coleção e união sejam feitas com comandos SQL para manipular dados vetoriais e do tipo raster. Para estes dados, oferece funções já implementadas para reprojeção, as quais também podem ser chamadas com comandos SQL. De acordo com PostGIS (2013), outros benefícios oferecidos incluem:

- Suporte a redes topológicas, objetos tridimensionais e índices espaciais.
- Importação e renderização de dados *raster* em formatos GeoTIFF, PNG, JPG e NetCDF.
- Interfaces de linha de comando e gráfica para importação e exportação de arquivos *shapefile*.

2.3.4 FERRAMENTAS ETL

Segundo Inmon (2002), devido a primeiras impressões que acusam simplicidade na movimentação dos dados operacionais para o *Data Warehouse*, algumas empresas negligenciam o uso de ferramentas especializadas em ETL e optam pela codificação manual das transformações.

O processo de ETL quando feito de forma manual torna-se complexo, tedioso e repetitivo, podendo assim resultar em baixa qualidade dos dados implantados no *Data Warehouse* (INMON, 2002). Ao utilizar-se de tal metodologia, todas regras de transformação são implementadas sob demanda, através de linguagens de programação. Entre as dificuldades encontradas nesta prática se encontram:

- O difícil acesso a sistemas legados, os quais detém os dados operacionais a serem tratados, bem como a necessidade de fornecer acesso a diversos SGBDs, passíveis de serem utilizados como base de *Data Warehouses*;
- A necessidade de restruturação e reformatação dos dados, bem como substituição de valores com base em padrões a serem definidos;

- A existência de diversas fontes de dados, sendo estas heterogêneas, portanto possuindo estruturas de dados completamente diferentes;
- A necessidade de implantar os dados transformados em diversos *Data Warehouses* ou *Data Marts*, diferindo entre eles em níveis de sumarização.

Inmon (2000) afirma que uma alternativa viável ao desenvolvimento manual é o uso de ferramentas ETL. Para Aguiar (2010), a principal vantagem que estes softwares oferecem é de não precisar implementar regras de transformação. Tais regras já estão construídas e disponíveis, sendo assim necessário apenas configurações que as adaptem às necessidades de transformação. Outras vantagens do uso de tais ferramentas incluem (AGUIAR, 2010):

- O processo de recuperação dos dados em ambientes operacionais é facilitado com o uso de ferramentas ETL. Os softwares desta categoria geralmente oferecem fácil comunicação com os principais SGBDs do mercado, além de suportarem acesso a fontes de dados alternativas, como arquivos de texto plano ou planilhas eletrônicas;
- A performance do processamento de grandes quantidades de dados por ferramentas ETL costuma ser maior que de programas implementados sob demanda. O ganho de desempenho muitas vezes é resultante de estratégias como o processamento paralelo, que possibilita um melhor uso do hardware utilizado;
- Projetos de ETL podem ser parcialmente reutilizados quando feitos com auxílio de ferramentas especializadas. Neste cenário, regras de carga e transformações são apenas adaptadas para as novas regras de negócio, desta forma poupando custos de implementação;
- Os custos de treinamento de equipes para operar estes softwares muitas vezes é reduzido por conta da documentação disponível pelo fabricante. Outro fator que diminui a curva de aprendizado é a usabilidade das ferramentas, resultado de interfaces simples e autoexplicativas;
- As principais soluções hoje disponíveis permitem que um processo seja retomado de onde parou. Sendo assim, em caso de falha nas transformações, não há a necessidade de retrabalho;
- Ferramentas especializadas permitem auditorar os procedimentos realizados sobre os dados, oferecendo assim um maior controle sobre todo ETL.

2.3.5 GEOKETTLE

Geokettle é uma ferramenta ETL gratuita, produzida pela empresa Spatialytics e disponibilizada em seu *website* oficial. Baseado no *software open source* Pentaho Data Integration (ou Kettle) da Pentaho Corporation, é apresentado como uma poderosa solução livre, contendo as principais funcionalidades necessárias para desenvolvimento de projetos ETL. Segundo GeoKettle (2013), diversas organizações no mundo, como agências governamentais, agências de seguro e bancos já o adotaram como solução oficial em projetos de *Spatial Data Warehouse*.

A interface gráfica deste *software* é simples e intuitiva, sendo dividida em duas áreas principais. A primeira encontra-se ao lado esquerdo e disponibiliza os componentes para ETL, bem como atalhos para conexão com bancos de dados, particionamento de esquemas e conexão com servidores remotos. A segunda encontra-se ao lado direito e consiste na área para desenvolvimento do projeto. Os componentes podem ser acessados com facilidade, bastando selecioná-los e arrastá-los para a área de desenvolvimento.

Segundo GeoKettle (2013), a ferramenta ETL GeoKettle é uma solução rápida, estável e construída com base em padrões. Possui integração com trinta e sete SGBDs, estando entre eles o PostgreSQL, MySQL, Oracle, DB2 e MS SQL Server.

Diversos formatos de arquivo são suportados para operações de leitura e escrita. Entre eles estão DBF, XML, JSON, CVS, CSW, texto plano, properties e formatos proprietários como dos *softwares* Microsoft Office Excel e Access. Entre os arquivos de dados espaciais suportados encontram-se o *Shapefile*, GML 3.1.1, KML 2.2 e todos formatos OGR (GEOKETTLE, 2013).

Entre as operações de transformação de dados encontram-se seleções, particionamento, filtros, junções, duplicidade de dados, clusterização, pivoting, divisões e diversos cálculos. Transformações livres podem ser feitas com *scrips* de linguagem como SQL, Javascript e expressões regulares. As operações específicas para dados espaciais incluem agregações espaciais, visualizações cartográficas, geoprocessamentos com recortes, simplificação de geometrias, divisões espaciais, análises espaciais como bufferização, centróides, distâncias, intercecções e uniões de formas (GEOKETTLE, 2013).

Alguns dos recursos oferecidos pelo GeoKettle são a automação de processos com uso de *Jobs*, otimização de tempo de execução através de execuções paralelas ou através de distribuição de trabalho para servidores remotos, acesso a dados de serviços WEB e sensores de observação, agendamento de execuções e transferência de arquivos via protocolos FTP, SFTP e

SSH2.

Este *software* trabalha com conceito de *Transformation*, definido por GeoKettle (2013) como um conjunto de operações a serem feitas sobre os dados operacionais. As operações são realizadas por *steps*, os quais são componentes dispostos na área de trabalho da ferramenta e são executados de acordo com o fluxo tomado pelos dados.

Para as ligações entre os componentes é dado o nome de *hops*. Estes são categorizados em *copy*, *distribute* e *conditional output*. Os dois primeiros dizem respeito a forma que os dados serão repassados aos componentes seguintes. Em ligações do tipo *copy*, o conjunto de dados é repassado por inteiro como objeto de entrada ao próximo componente. Já em ligações do tipo *distribute*, o conjunto de dados é repartido em partes iguais e então distribuído entre todos os próximos *steps*.

As ligações do tipo *conditional output* ditam o fluxo a ser tomado pelos dados. São utilizadas por componentes que realizam operações condicionais, como o *Table exists* que verifica se uma determinada tabela existe em um banco de dados ou não.

2.4 SISTEMA DE SUPORTE A DECISÃO

Sauter (2011) define Sistema de Suporte a Decisão como um sistema baseado em computadores cujo objetivo é fornecer informações de negócio, úteis e em um formato adequado. Este *software* fornece meios flexiveis de recuperação a análise das informações, bem como ferramentas que auxiliam no entendimento do problema, detecção de oportunidades e possíveis soluções.

Sistemas de Suporte a Decisões baseados em SDW possuem entre outras funcionalidades, a análise espacial generalizada, múltiplas representações de mapas multi-escalares, interfaces cartográficas navegacionais para operações de SOLAP, métodos de indexação espaço-temporal multi-escala e fusão automática de dados espaciais (BÉDARD et al., 2001).

Para Sauter (2011), SSDs são úteis quando não está claro qual escolha deve ser tomada para solução de um problema. Diferentemente de sistemas heurísticos especializados, os quais utilizam de lógicas *fuzzy* e redes neurais para simular lógicas humanas, SSDs não apontam qual solução deve ser tomada, mas sim fornecem informações que serão interpretadas por um

especialista no negócio.

Crossland e Wynne (1994) puderam comprovar através de experimentos que decisões embasadas em SSDs são tomadas em menores espaços de tempo, além de apresentarem uma menor taxa de erros. Vale observar que softwares de SIG quando utilizados em ambientes estratégicos, também aumentam a performance nas tomadas de decisões onde informações geográficas estão presentes.

Segundo Bédard et al. (2001), entre as exigências dos usuários destes sistemas, estão interfaces simplificadas e grande usabilidade. Outros requisitos são respostas rápidas a consultas, suporte a consultas ad-hoc[7] em diferentes níveis de agregação, disponibilidade de dados de diferentes épocas e capacidade de análises automáticas para descoberta de padrões. Kimball e Ross (2002) complementa a lista ao dizer que a usabilidade é um requisito sempre presente em sistemas desta categoria.

2.4.1 USUÁRIOS

Segundo Power (2012), os usuários dos SSDs levam como título de cargo analista de suporte a decisão. Estes têm como principal função realizar análises sobre dados do negócio, e então entregar os resultados obtidos de forma resumida aos tomadores de decisões. Análises financeiras, de custo benefício e estudos de viabilidade são exemplos de trabalhos executados por estes profissionais.

Analistas de suporte a decisão têm como ferramentas de trabalho *softwares* OLAP ou SOLAP, planilhas eletrônicas, pacotes estatísticos, *softwares* para mineração de dados, sistemas baseados em SIG, entre outros (POWER, 2012).

Para GreatSampleResume (2013), entre as responsabilidades deste profissional estão a condução de treinamentos dos membros da equipe para uso do SSD e levantamento de requisitos de informações. Outras funções que lhe dizem respeito são:

- Gerenciar a integridade e precisão do sistema;
- Resolver problemas de cunho técnico relacionados as ferramentas;
- Planejar o desenvolvimento da solução, bem como recomendar melhorias;

[7] Consultas construídas e executadas sob demanda, em tempo real, sem que haja a necessidade de defini-las previamente.

- Gerenciar o repositório de dados para o sistema;
- Testar o sistema a cada inclusão de funcionalidades em busca de erros.

2.4.2 OLAP

Online Analytical Processing - OLAP é uma categoria de SSD que realizam consultas rápidas, interativas e fáceis através de uma interface multidimensional (BÉDARD et al., 2001). Ferramentas OLAP operam sobre *Data Warehouses* ou *Data Marts* e oferecem funções que permitem a visualização de dados devidamente estruturados sob várias perspectivas (KIMBALL; ROSS, 2002).

As ferramentas OLAP auxiliam analistas de suporte a decisões em suas tarefas com consultas ad-hoc, apresentando informações de negócio em tabelas, gráficos de pizza, barras, histogramas, gráficos estatísticos tri-dimensionais e gerando relatórios (BÉDARD et al., 2001).

Mecanismos que provêem inteligibilidade dos resultados são requisitos para estes *softwares*. Uma função comumente disponibilizada é a de configurar diferentes formas de apresentação para valores que satisfaçam algumas condições (RIVEST et al., 2003). Em um exemplo de uso deste artifício, o total em vendas de uma filial em um determinado período seria apresentado em vermelho caso não atingisse um valor mínimo estipulado.

Outras possibilidades quanto a apresentação incluem agrupar resultados de diferentes consultas em uma mesma tela, dividi-los em várias páginas, ordená-los, filtrá-los de acordo com vários critérios e listá-los em forma de *ranking* (ANZANELLO, 2009).

As operações oferecidas para manipulação das consultas ad-hoc são o *drill-down*, *drill-up*, *slice*, *dice*, *drill-across*, *drill-through* e *pivoting*. O *drill-up* permite que a consulta antes feita sobre um conjunto mais detalhado de dados, passe a ser feita sobre dados mais agregados (RIVEST et al., 2003). Por exemplo, visualizar dados relacionados a unidades federativas ao invés de microrregiões.

A operação *drill-down* em contrapartida possibilita consultas sobre dados de menor granularidade (RIVEST et al., 2003). Um exemplo seria alterar o período de trimestre para mês em uma consulta que buscasse o total em vendas para um cliente.

Na operação *drill-across*, o nível de dados sobre o qual é feita a consulta é alterado.

áA particularidade desta operação é que um ou mais níveis intermediarios em uma hierarquia podem ser ”pulados”. Em uma dimensão temporal composta por ano, mês e semana, o *drill-across* acontece ao se passar diretamente de ano para semana.

Ao se fazer um *drill-through*, a análise é feita sobre o mesmo conjunto de dados, porém alternando entre dimensões. Já com *slice* e *dice* a análise pode ser limitada a um subconjunto de dados de interesse, observando-o sob várias perspectivas.

2.4.3 SOLAP

Segundo Rivest et al. (2003), ferramentas de OLAP, *Data Mining* e construtores de relatórios utilizados para exploração de DWs, não são otimizados para explorar e analisar dados espaciais. Sistemas de Informação Geográfica oferecem funcionalidades que facilitam a visualização destes dados, porém não são preparados para servir informações em ambientes estratégicos.

Ferramentas SOLAP são o resultado da união dos conceitos de OLAP com SIG. Com boa performance para consultas complexas e formas adequadas de apresentação de dados espaciais, o *software* SOLAP é uma alternativa viável para análises em *Spatial Data Warehouses* (RIVEST et al., 2003).

Bédard et al. (2001) define SOLAP como ”uma plataforma visual especialmente construída para exploração e análise espaço-temporal fácil e rápida dos dados, seguindo uma abordagem multidimensional composta por níveis de agregação apresentados em displays cartográficos, tabulares e de diagramas”.

Ferramentas SOLAP são capacitadas para explorar os dados disponíveis nos três tipos exclusivos de dimensões dos *Spatial Data Warehouses*. Além disso, são capazes de apresentar dois tipos de medidas espaciais (RIVEST et al., 2003). A primeira consiste em uma forma geométrica ou um conjunto destas. Ao apresentar o conjunto das formas, operações de agregação são realizados automaticamente pelo aplicativo. O segundo tipo de medida são valores numéricos que representam características do conjunto de dados espaciais. Um exemplo é a soma da área de todos os objetos.

Uma das duas partes básicas que formam a interface das ferramentas SOLAP é o espaço para apresentação dos resultados das consultas ad-hoc. Neste espaço, os resultados são apresentados através de múltiplos mapas, diagramas e tabelas (RIVEST et al., 2003).

A segunda parte da interface é o painel de navegação, que tem como função apresentar dimensões e medidas para construção das consultas (RIVEST et al., 2003). Nesta área, ambos os elementos são apresentados em listas ou estruturas de árvore para maior organização.

Ferramentas SOLAP oferecem formas de exploração dos dados que extendem as operações das aplicações OLAP. Algumas delas são o *spatial drill-down*, *spatial drill-up* e *spatial drill-across*, responsáveis pela navegação entre os níveis de hierarquias espaciais. Rivest et al. (2003) explica que conforme estas operações são executadas, os resultados das consultas são automaticamente atualizados na área de apresentação.

3 MATERIAL E MÉTODOS

Neste estudo de caso foram realizados trabalhos de extração, transformação e carga dos dados apresentados na sessão 3.1. O projeto de ETL foi dividído em 4 etapas (descritas no capítulo 4), sendo todas desenvolvidas através do *software* GeoKettle. Cada etapa corresponde a uma ou mais *Transformations*, conceito definido por GeoKettle (2013) como ”um conjunto de operações a serem realizadas sobre os dados extraídos”. No universo deste *software* o nome *Step* é dado a uma operação.

O produto de cada *Transformation* é o ponto de entrada da subsequente, tornando assim necessário que cada uma seja realizada por inteiro para que a próxima possa ser iniciada. A ferramenta utilizada oferece ao seu operador total controle das operações feitas sobre os dados. Caso algum *Step* em qualquer ponto do projeto falhe, o erro pode ser interpretado através do *logging* apresentado.

O conjunto de dados a ser processado é proveniente de um arquivo *Shapefile*. Sua carga se feita diretamente para um banco de dados, resultaria em uma estrutura denormalizada composta por uma única tabela de 53 colunas, muitas das quais não são de interesse neste projeto. A existência de colunas desnecessárias somada aos ruídos identificados nos dados remete a algumas das necessidades de transformações descritas na sessão 3.2.

Foram avaliados *softwares* livres especializados em ETL e gerenciamento de dados com o intuito de julgar sua aptidão para uso em construções de *Spatial Data Warehouses*. Os *softwares* escolhidos foram o GeoKettle e o SGBDs PostgreSQL com seu complemento PostGIS. No estudo de caso estas soluções foram utilizadas para desenvolvimento de um projeto experimental, no qual foram levantados requisitos de ETL para um conjunto de dados espaciais.

3.1 CONJUNTO DE DADOS

O conjunto de dados utilizado nos projetos ETL do estudo de caso, pode ser obtido gratuitamente em GADM (2013) nos formatos *Shapefile*, ESRI, RData e KMZ. Estes são dados espaciais de formato vetorial, os quais representam limites administrativos de países por todo o mundo, bem como suas divisões administrativas internas. Vale notar que o nível de

detalhamento das divisões territoriais não é uniforme.

Na Figura 1 pode-se observar os limites territoriais internos do Brasil representados por dados espaciais de formato vetorial.

Neste conjunto de dados, atributos descritivos estão relacionados aos objetos espaciais. Entre eles se encontram os valores identificadores de cada território, sigla, nome, tipo do território (como província, país ou unidade federativa), data de oficialização, entre outros.

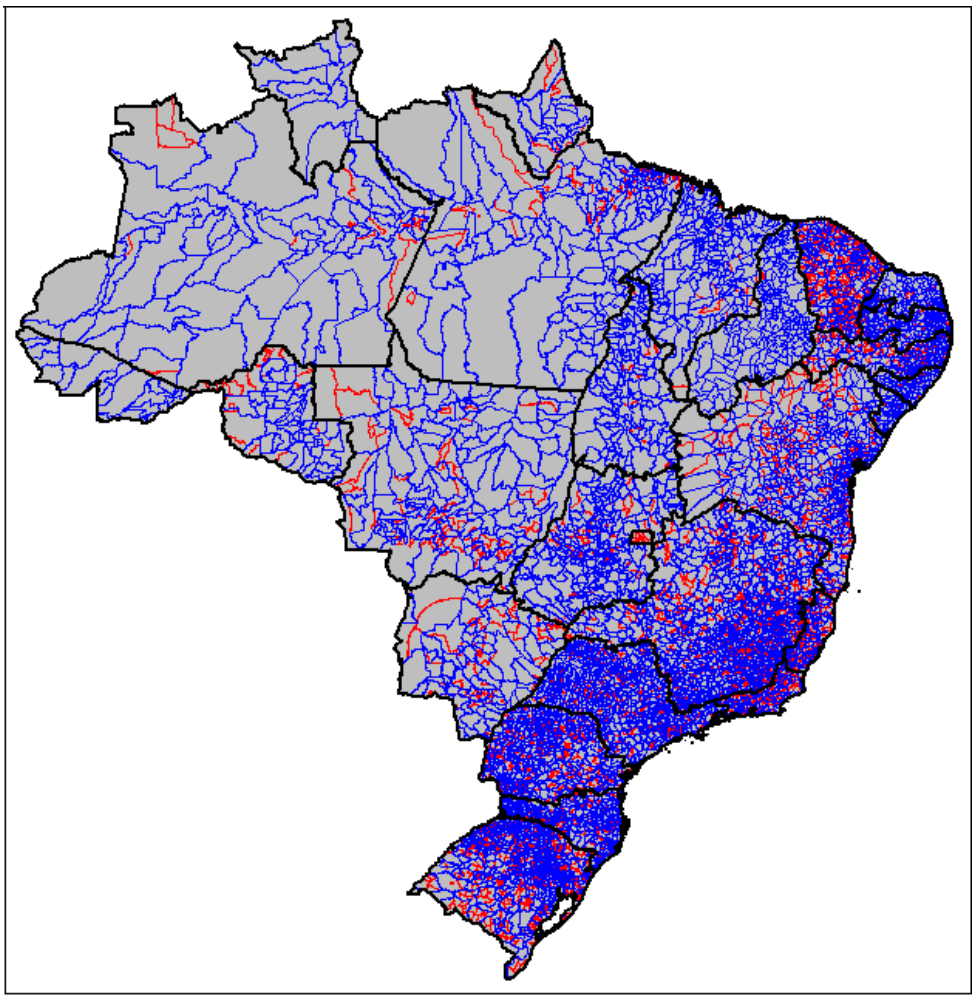

Figura 1: Limites territoriais do Brasil representados por dados vetoriais.
Fonte: (GADM, 2013).

3.2 REQUISITOS DE ETL

O primeiro requisito de ETL é a manipulação das colunas das tabelas. Entre as operações relacionadas encontram-se a seleção, remoção e renomeação das tabelas, bem como alteração do tipo, comprimento, precisão, formato e número de casas decimais de seus dados. Outra necessidade é de alteração da ordem que são apresentadas.

O segundo requisito é a limpeza de ruídos nos dados. Entre as operações relacionadas encontram-se a alteração dos valores identificadores dos dados de cada nível espacial. Tal transformação é necessária, pois neste conjunto de dados, o valor padrão 0 (zero) é atribuído ao atributo identificador do nível espacial, o que ocorre caso este não seja representado por um objeto geométrico. Neste projeto, tal valor é considerado um ruído, portanto deve ser eliminado.

Outra operação de limpeza será sobre as datas de oficialização dos territórios. Em casos onde a data não é conhecida, o valor padrão ”Unknown”foi atribuído, levando a necessidade de eliminá-lo por ser considerado um ruído.

O terceiro requisito diz respeito a manutenção da integridade dos dados. Deve-se eliminar a duplicidade dos valores identificadores dos dados de todos os níveis espaciais, afim de representar corretamente os relacionamentos topológicos dos objetos espaciais. A manutenção cuidadosa dos valores identificadores permitirá que uma consistente estrutura hierárquica seja construída, em um processo de migração do esquema original *Star* para o esquema *Snowflake*.

O quarto requisito é a criação de uma estrutura normalizada, sobre a qual os dados serão implantados. Esta estrutura correponde a dimensão espacial, composta por uma hierarquia de seis níveis espaciais. Na prática, o principal benefício conseguido seria uma melhor representação das relações topológicas entre os níveis espaciais. Em contrapartida, haveria perda de desempenho em consultas sobre dados espaciais de maior granularidade.

O quinto requisito é a geração de dados espaciais para níveis hierárquicos de maior granularidade, a partir de operações de agregação sobre os dados de menor granularidade. Dessa forma, poderia ser gerada uma forma geométrica que representaria o território de um país a partir da união dos dados espaciais de seus estados.

O sexto requisito é a geração de novos atributos cujos valores serão calculados a partir das geometrias de cada nível ou através de simples operações de contagem. Os atributos gerados serão a área, perímetro e número de componentes de cada território. Com exceção dos dados de menor granularidade, os dois últimos requisitos de ETL estão relacionados com os dados de

todo conjunto.

O sétimo e último requisito de ETL para este projeto é a limpeza da *Staging Area*. No decorrer das operações, os resultados das *Transformations* (exceto da última) são armazenados em arquivos *Shapefile*, os quais deverão ser removidos do sistema liberando espaço em disco.

A estrutura da dimensão espacial que se pretende obter no final do desenvolvimento deste estudo de caso pode ser observada na Figura 2. A tabela *dim_nvl5* corresponde a folha da hierarquia contendo os dados de menor granularidade, enquanto que a tabela *dim_nvl0* corresponde a raiz.

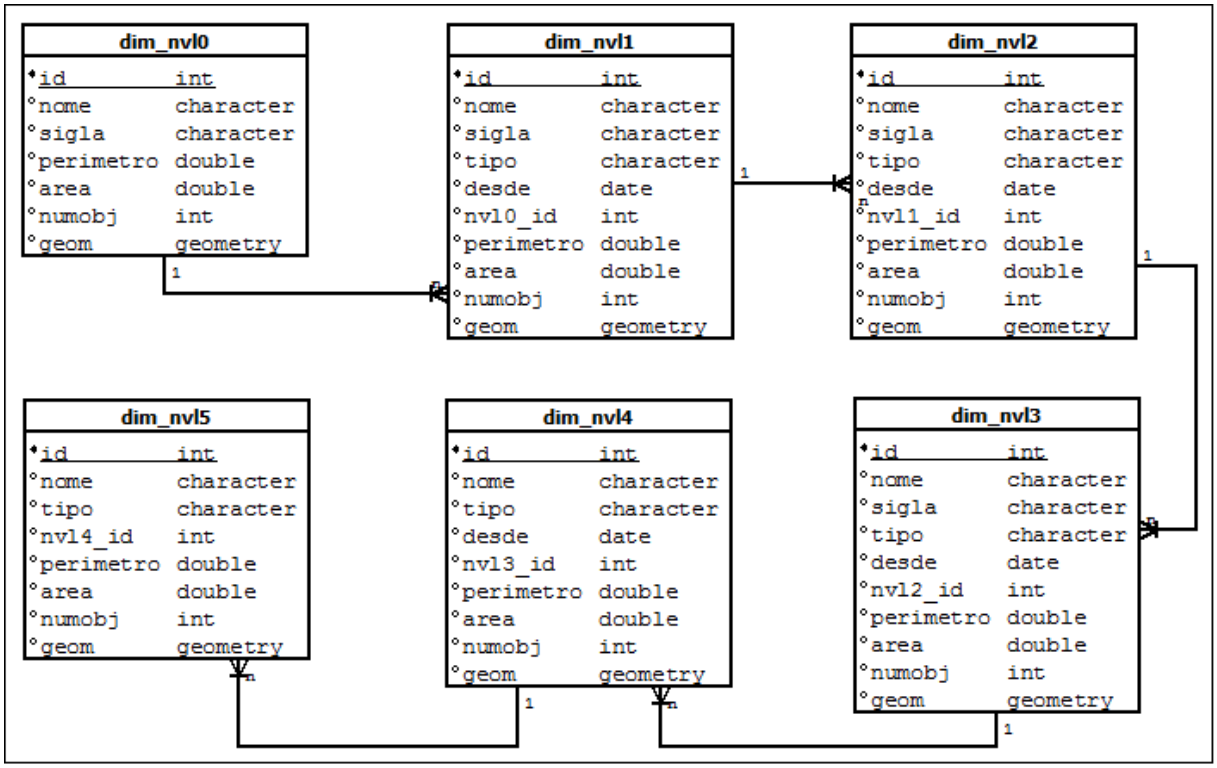

Figura 2: Modelo hierárquico da dimensão espacial.

Fonte: Autoria própria.

3.3 GEOKETTLE

O *software* para ETL que será utilizado neste trabalho é o GeoKettle em sua versão 2.5. Este será usado para extrair os dados do arquivo *shapefile*, transformar os dados espaciais de acordo com os requisitos identificados e carregá-los em um Sistema Gerenciador de Banco de Dados.

3.4 POSTGRESQL

O Sistema Gerenciador de Banco de Dados que será utilizado é o PostgreSQL em sua versão 9.2. Neste será construída uma base de dados que conterá um conjunto de tabelas que representarão uma dimensão espacial do modelo multidimensional.

Este sistema deverá suportar dados do tipo geométrico, os quais conterão as informações espaciais para representação dos objetos espacialmente referenciados.

3.5 POSTGIS

Como complemente do sistema PostgreSQL será utilizado o *software* PostGIS em sua versão 2. Este terá como função adicionar suporte ao SGBD a dados do tipo vetorial. Também deverá fornecer suporte a funções espaciais, como criação de colunas geométricas sobre uma estrutura de tabela já criada.

4 ESTUDO DE CASO

4.1 EXTRAÇÃO E LIMPEZA DOS DADOS

O processo de extração dos dados espaciais ocorre na primeira *Transformation* com auxílio do componente *Shapefile file input*. Dados espaciais armazenados em fontes de outros formatos podem ser recuperados através de outros componentes fornecidos pela ferramenta. Entre eles estão o *GML file input*, *KML file input* e *OGR Input*, além dos componentes para acesso a dados de sensores de observação e serviços WEB. A configuração do componente utilizado, assim como nos outros mencionados pode ser feita com facilidade através de uma interface simples e autoexplicativa. A tela de configuração é apresentada na Figura 3.

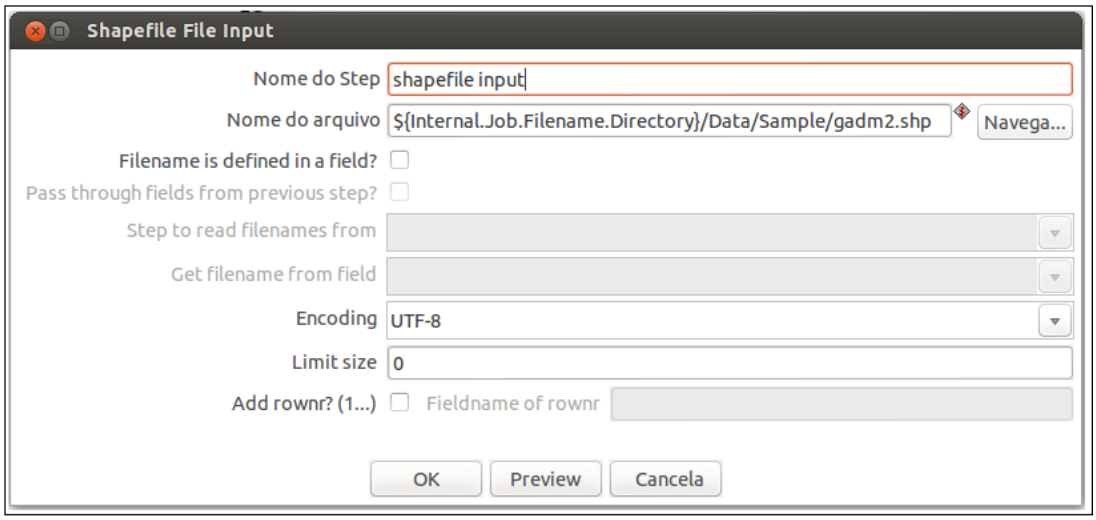

Figura 3: Configuração do *Step* responsável pela extração dos dados.

Fonte: Autoria própria.

A disposição dos *Steps* que compõem esta *Transformation* pode ser visualizada na Figura 4. Nesta imagem pode-se observar a documentação sobre a *Transformation* apresentada em uma caixa amarela. A prática de documentar cada etapa do projeto de ETL é recomendada por GeoKettle (2013), e auxilia o entendimento do projeto por novos membros da equipe. Neste *software* também é possível a documentação dos componentes de forma individual.

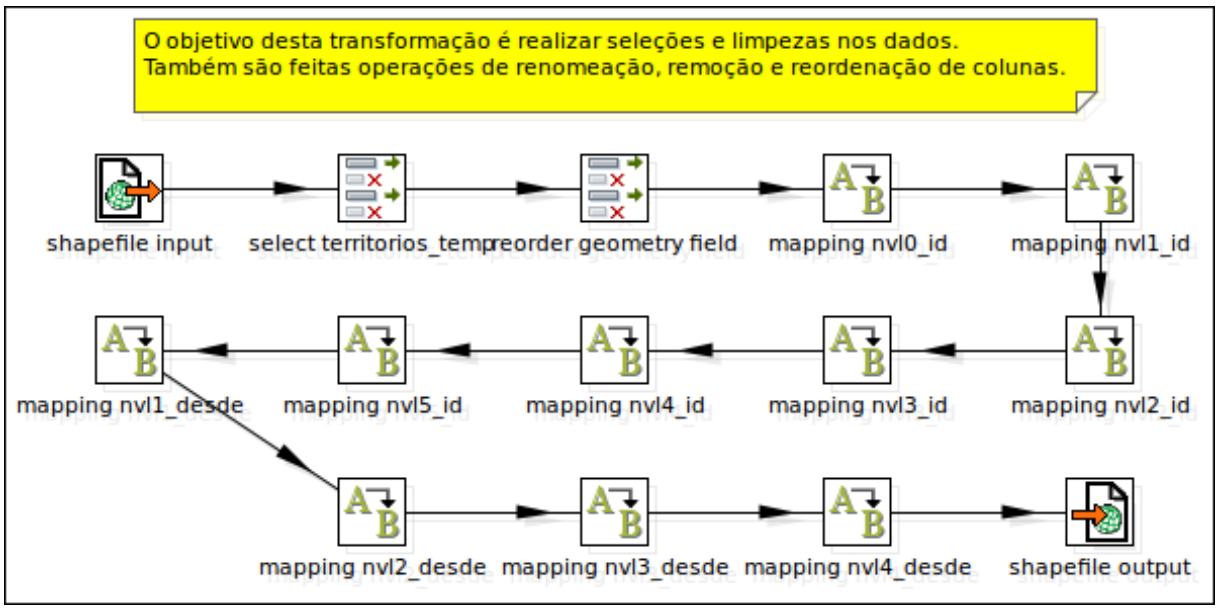

Figura 4: ***Steps*** **da primeira** ***Transformation*****.**

Fonte: Autoria própria.

Nesta etapa também foram feitas remoções dos atributos desnecessários e configurações dos de interesse. Entre as alterações encontram-se a renomeação do nome dos atributos, a reordenação destes, bem como a alteração dos tipos de dados dos atributos identificadores de cada nível espacial. A última operação mencionada foi necessária pois o tipo *number* foi assumido por padrão pela ferramenta para valores numéricos. Neste caso, o tipo de dado *integer* foi explicitamente atribuído. O componente utilizado para tais operações foi o *Select values*, e sua tela de configuração pode ser visualizada na Figura 5.

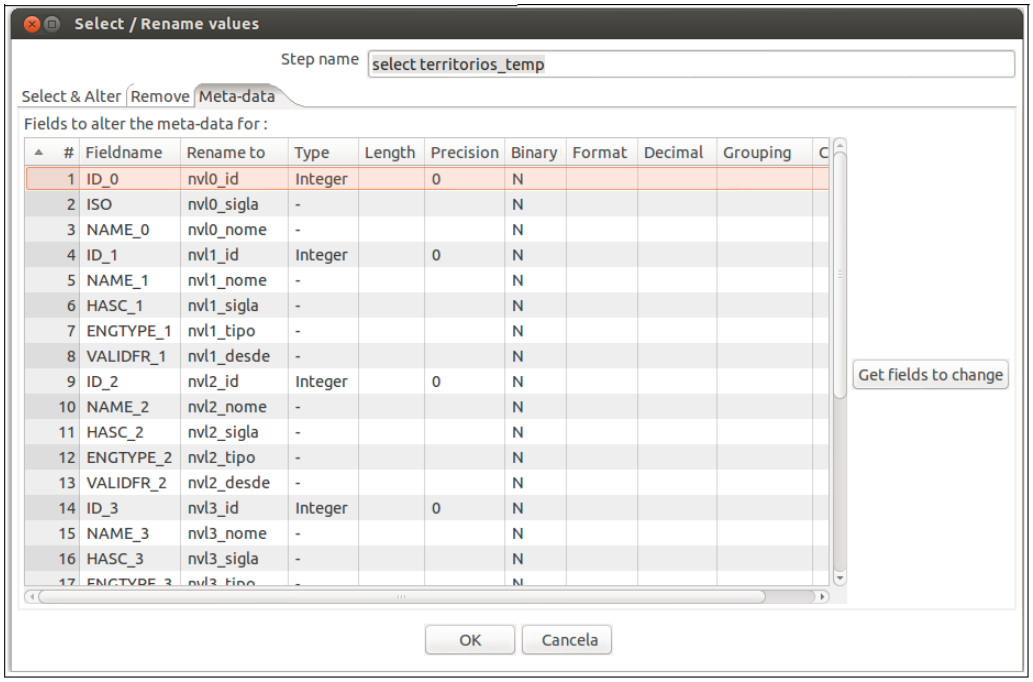

Figura 5: Tela para manipulação de atributos do componente *Select values*.

Fonte: Autoria própria.

As últimas transformações desta etapa foram o mapeamento de valores, servindo como limpeza de ruídos nos dados. Os ruídos removidos foram valores numéricos automaticamente atribuídos para os atributos identificadores de cada nível. Esta operação de limpeza foi realizada com auxílio do componente *Value Mapper*, e sua tela de configuração é apresentada na Figura 6. Operações de limpeza também ocorreram nos valores das datas de oficialização dos territórios. Em situações onde a data não era conhecida, o valor ”Unknown”havia sido atribuído automaticamente, resultando em um acrescimo desnecessário no volume do conjunto de dados.

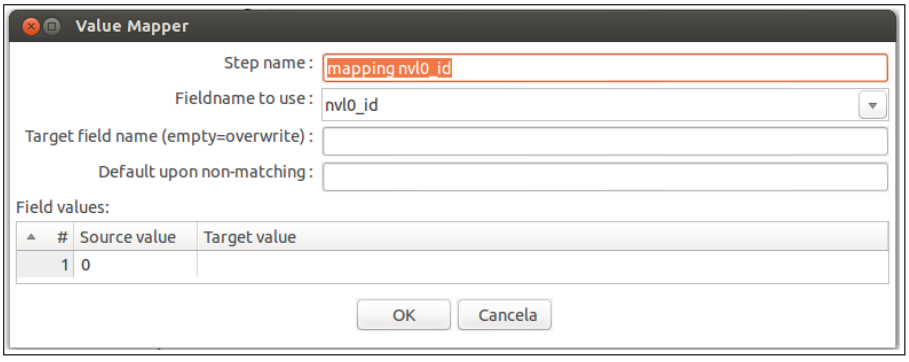

Figura 6: Limpeza de ruídos nos identificadores dos dados espaciais.

Fonte: Autoria própria.

A última operação desta etapa é a persistência dos dados transformados em arquivos *Shapefile*, através do componente *Shapefile File output*. Este arquivo será objeto de entrada para a etapa subsequente.

4.2 MANUTENÇÃO DE INTEGRIDADE

A segunda etapa está relacionada a segunda *Transformation* do projeto de ETL, e corresponde a manutenção da integridade dos dados. Para isto, são eliminadas as duplicidades dos valores identificadores, permitindo que nas próximas etapas seja feita a migração do esquema *Star* para o esquema normalizado *Snowflake*.

Os dados transformados nesta etapa são provenientes de um arquivo *Shapefile*, este resultante da etapa anterior. A disposição e o fluxo das transformações podem ser visualizados na Figura 7.

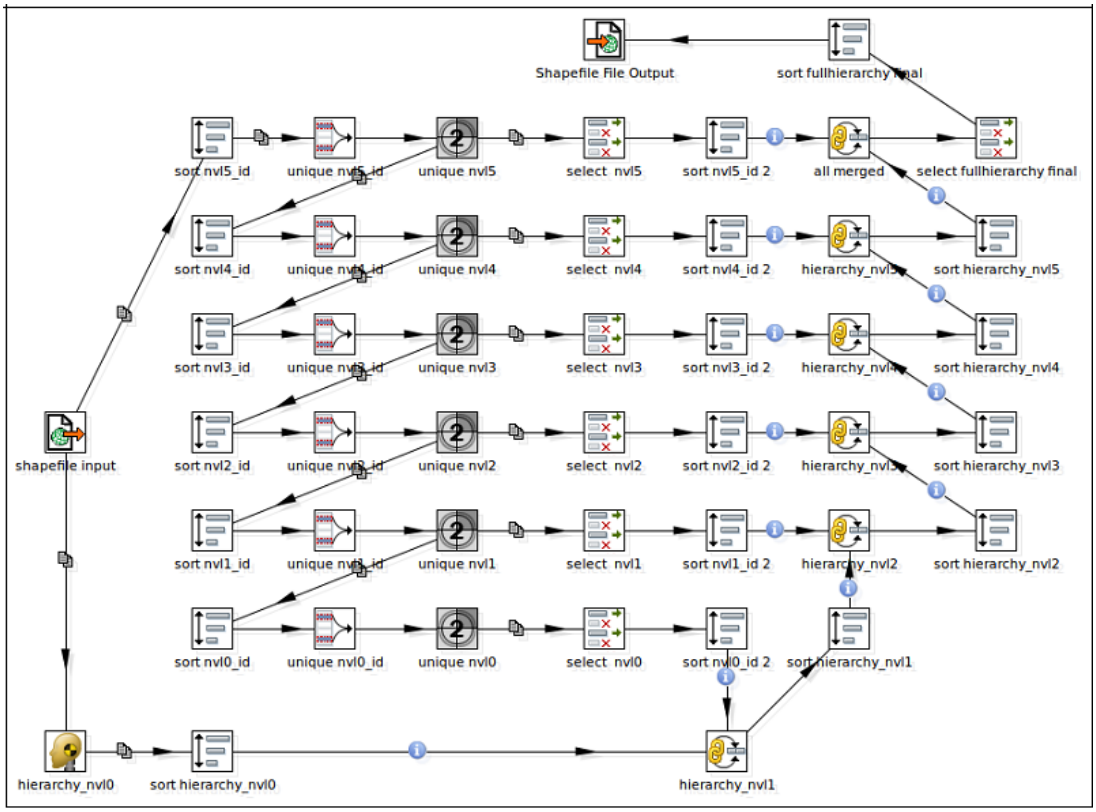

Figura 7: *Steps* da segunda *Transformation*.

Fonte: Autoria própria.

Nesta etapa, houve a necessidade de tratamento dos dados de diferentes granularidades separadamente. Os dados que antes se encontravam em uma só estrutura, foram divididos conforme seu nível de detalhamento em um processo de *snowflaking*. Após feita a manutenção de seus valores identificadores, os dados foram novamente integrados gradativamente em um processo de junção que iniciou nos níveis de maior granularidade para terminar na menor granularidade.

A atribuição dos novos valores aos atributos indentificadores foi realizada através do componente *Add sequence*, cuja função é atribuir valores numéricos sequenciais à uma coluna, a qual deve ser definida pelo usuário da ferramenta. As instâncias deste componente apresentadas na Figura 7 são o *unique nvl0*, *unique nvl1*, *unique nvl2*, *unique nvl3*, *unique nvl4* e *unique nvl5*, estando o último relacionado com os dados de menor granularidade.

Para que os novos valores identificadores fossem corretamente distribuídos entre os membros da dimensão espacial, foi necessária a captura dos valores únicos de cada nível. Esta filtragem nos dados foi realizada pelas instâncias do componente *Unique rows*, que obtiveram como resultado apenas valores não redundantes. Para tais instâncias foram definidos os nomes *unique nvl5_id*, *unique nvl4_id*, *unique nvl3_id*, *unique nvl2_id*, *unique nvl1_id* e *unique nvl0_id*. Na Figura 8 pode-se observar a configuração necessária para obtenção dos dados de menor granularidade não redundantes.

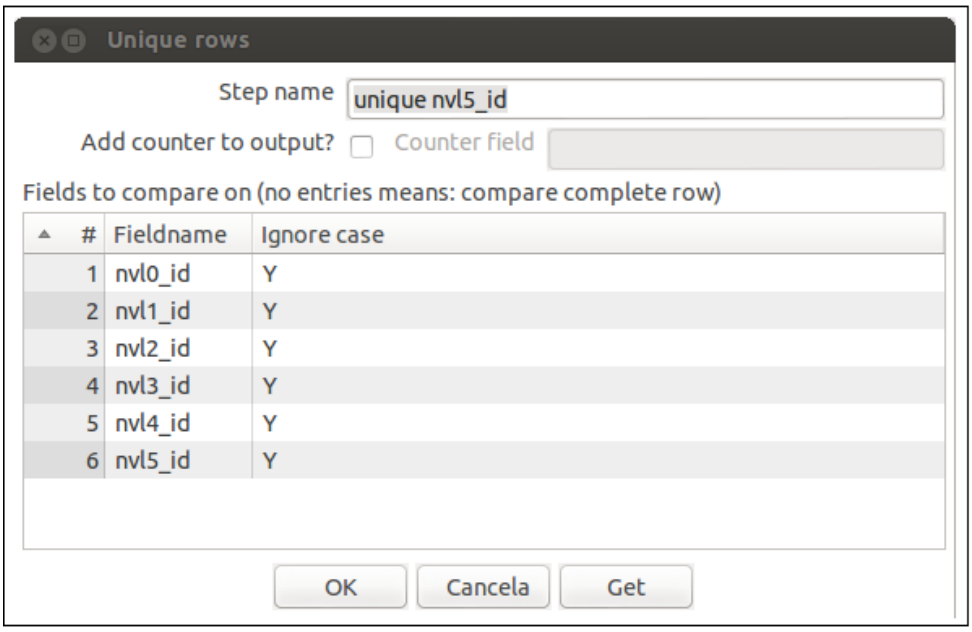

Figura 8: Configuração do componente *Unique rows*

Fonte: Autoria própria.

Um requisito para que as instâncias de *Unique rows* não apresentem em seus resultados dados duplicados, é que seus conjuntos de dados de entrada estejam devidamente ordenados. Neste projeto, o *Step* utilizado para ordenação é o *Sort rows*, embora a ferramenta ETL utilizada disponibilize outros componentes para mesma finalidade.

Após a atribuição das novas chaves de identificação, são iniciados os procedimentos que precedem a união dos dados dos diversos níveis espaciais. O primeiro procedimento é a seleção dos atributos utilizados como critério para integração, e este é realizado pelas instâncias do *Step Select values*. Tais instâncias são apresentadas na Figura 7 com o nome de *select nvl5*, *select nvl4*, *select nvl3*, *select nvl2*, *select nvl1* e *select nvl0*.

O segundo procedimento é a reordenação dos atributos já citados. A configuração dos componentes responsáveis devem coincidir quando estes forem utilizados em uma mesma operação. Por exemplo, os componentes denominados *sort hierarchy_nvl0* e *sort nvl0_id 2* devem ordenar seus dados seguindo um mesmo critério, pois seus resultados serão utilizados por um mesmo componente.

Neste momento os dados são integrados pelas instâncias do *Step Merge Join*. Este processo ocorre de forma gradativa, iniciando pela união dos dados de maior granularidade em direção aos mais detalhados. Na Figura 9 é apresentada a configuração do componente *all merged*.

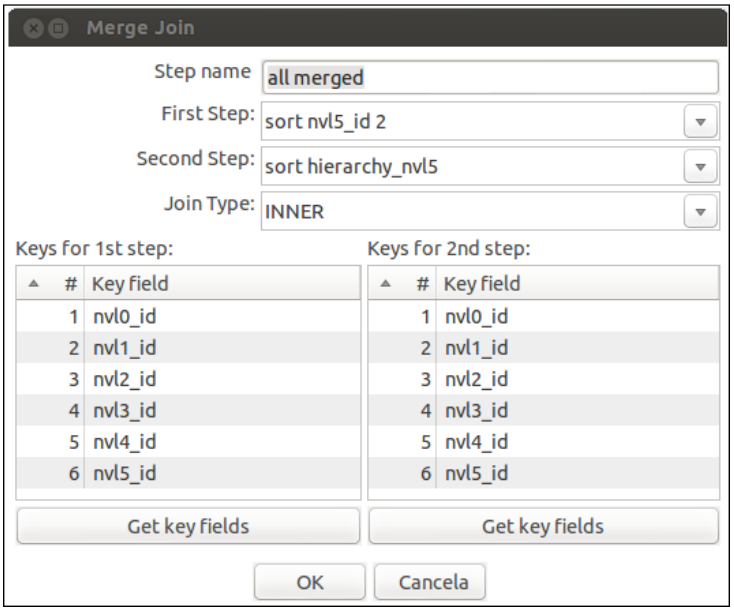

Figura 9: Integração dos níveis hierárquicos de menor granularidade.

Fonte: Autoria própria.

No final desta etapa, atributos definitivos são selecionados, reconfigurados, ordenados e em alguns casos seus tipos de dados são modificados. Os resultados obtidos nesta transformação é persistida em um arquivo *Shapefile* através do componente *Shapefile File Output*.

4.3 CRIAÇÃO DA DIMENSÃO ESPACIAL

A criação das tabelas que compõem a estrutura hierárquica da dimensão espacial é feita na terceira etapa deste projeto de ETL. Esta etapa é formada pela terceira até a oitava *Transformation*, as quais são formadas por um único componente cada.

O único *Step* presente nestas transformações é o *Execute SQL script*, que como o nome sugere, possibilita que instruções SQL sejam executadas no SGBD a partir da ferramenta de ETL. Segundo GeoKettle (2013), devido a natureza dinâmica deste componente, este não deve ser utilizado quando são necessárias instruções com performance otimizada. Para esta situação os componentes *Table Output*, *Table Input*, *Table Output*, *Insert*, *Update* e *Delete* atendem melhor as espectativas.

O componente utilizado permite que suas instruções SQL sejam executadas uma única vez ou uma vez para cada linha afetada pela instrução. Também é possível através de configuração, executar várias instruções como se fosse uma única instrução atômica. Outra possibilidade com o uso desta ferramenta é a utilização de parâmetros e variáveis de substituição para instruções dinâmicas.

Como mencionado, ao longo destas transformações são executados scripts para criação das tabelas que correspondem aos níveis hierárquicos da dimensão espacial. A adição da coluna geométrica em cada tabela é feita através da função do PostGIS ”AddGeometryColumn”. Seus parâmetros incluem o nome da tabela a ser alterada, o nome da columa geométrica, o identificador de sistema de referência espacial, o tipo de dado geométrico e o número de dimensões deste dado. Os scripts comentados são mostrados no apêndice A.

Para que software GeoKettle consiga realizar as instruções configuradas nos componentes destas transformações, faz-se necessária uma conexão com um banco de dados. Esta é configurada através de um assistente de conexões, o qual pode ser acessado a partir de cada componente que utilize deste recurso, ou a partir de um atalho na tela principal da ferramenta. A ferramenta permite que conexões sejam configuradas uma única vez e

compartilhadas entre os *Steps* que as utilizem, facilitando aínda mais o desenvolvimento do projeto. A tela do assistente mostrada na Figura 10.

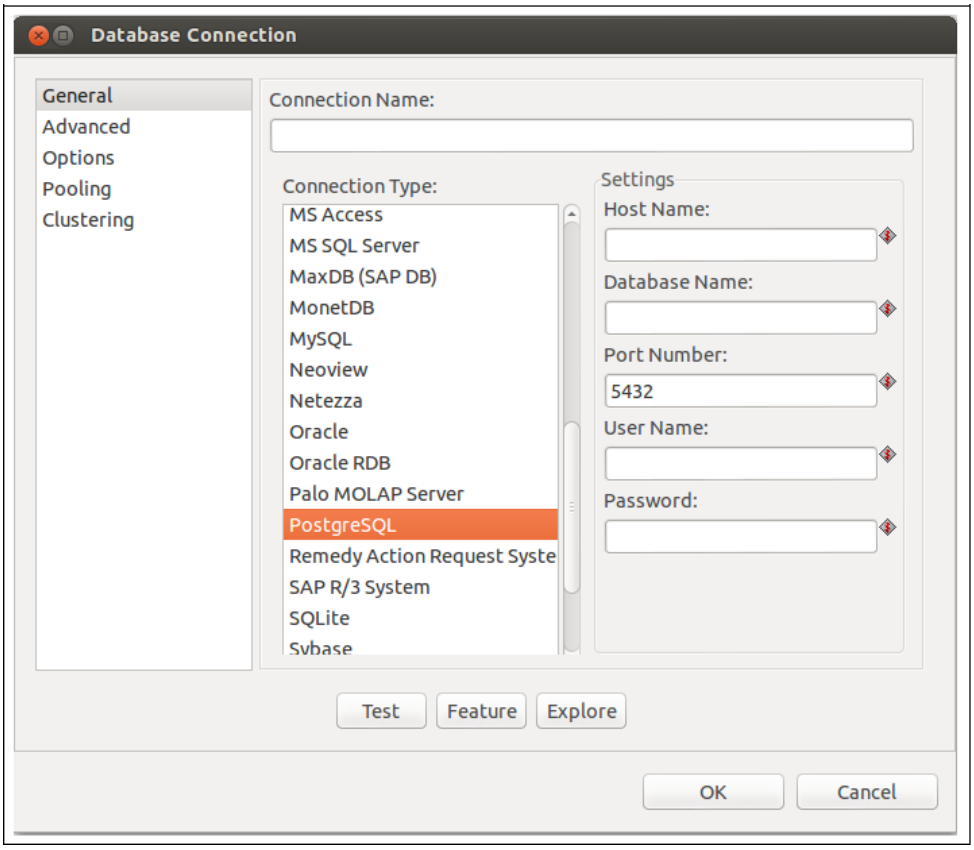

Figura 10: Tela para configuração de conexões com bancos de dados.

Fonte: Autoria própria.

4.4 POPULAÇÃO DA DIMENSÃO ESPACIAL

A quarta etapa do projeto de ETL corresponde a nona *Transformation*. Neste momento a estrutura hierárquica da dimensão espacial já se encontra criada e pronta para receber os dados espaciais. Os dados de entrada desta transformação são provenientes de um arquivo *Shapefile*, o qual foi gerado na segunda etapa do projeto.

O primeiro objetivo desta *Transformation* é realizar o processo de *snowflaking*, permitindo que os dados espaciais sejam depositados na estrutura hierárquica já existente no banco de dados. O segundo objetivo é gerar as colunas *perimetro*, *area*, *numobj* e *geom* para tabelas de todos os níveis da hierarquia. Os dados destas colunas são calculados a partir de operações de soma, contagem e união geométrica. O terceiro objetivo é realizar a carga dos

dados no SGBD através de componentes especializados.

A disposição dos componentes e o fluxo das transformações podem ser visualizadas nas Figuras 11 e 12. Ambas as Figuras mostram *Steps* da nona *Transformation*.

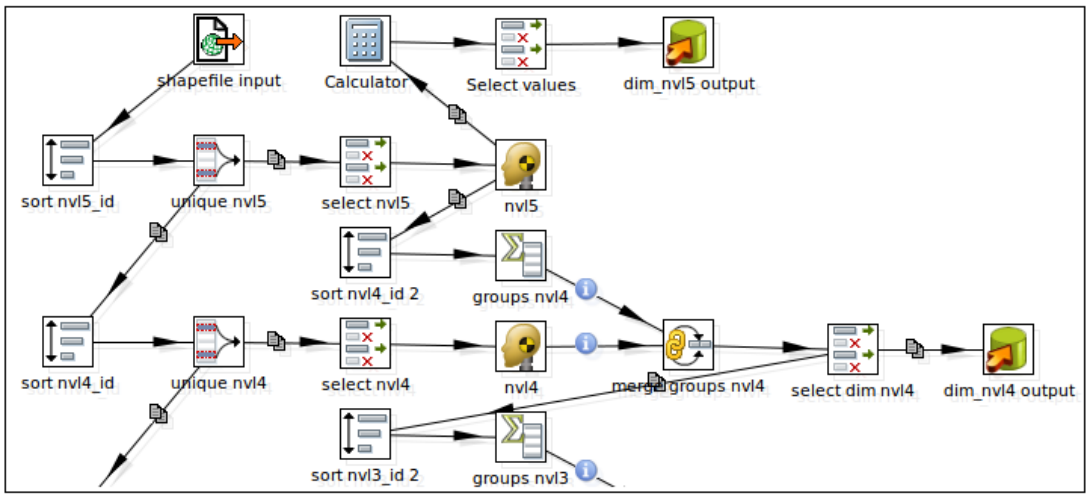

Figura 11: Primeiros componentes do fluxo de transformações da nona *Transformation*.

Fonte: Autoria própria.

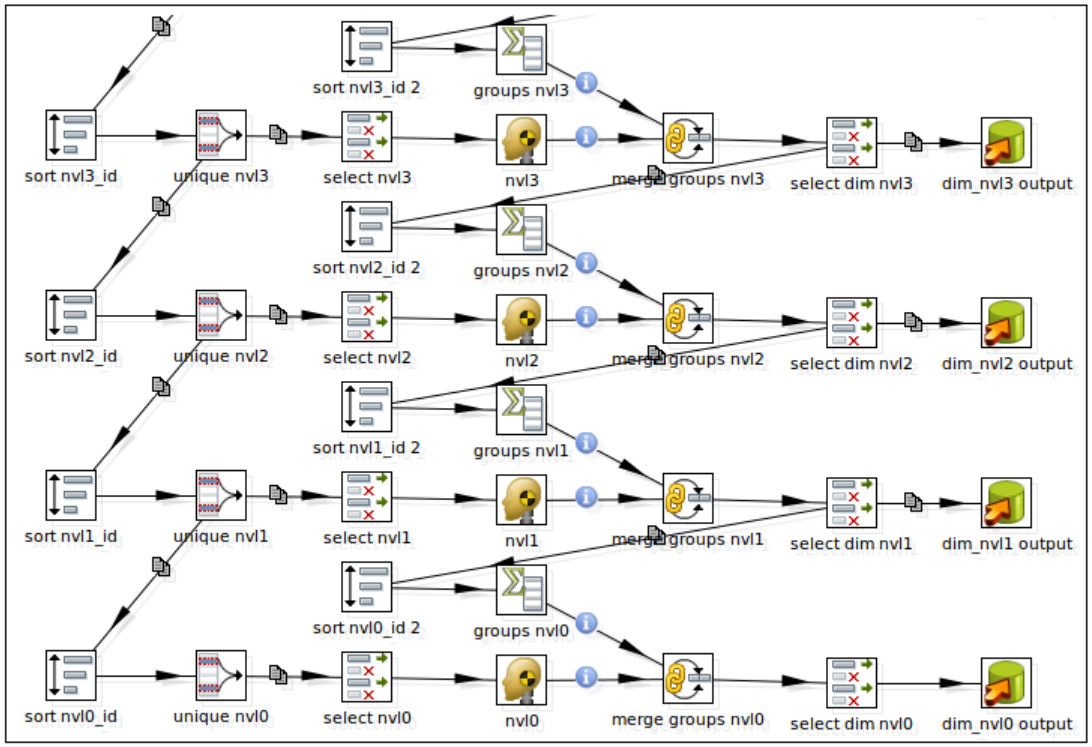

Figura 12: Outros componentes do fluxo de transformações da nona *Transformation*.

Fonte: Autoria própria.

Nesta transformação, a distribuição dos dados espaciais entre os níveis hierárquicos é feita com a combinação das instâncias de *Unique rows* com os *Sort rows*. O processo é iniciado com ordenação dos valores identificadores dos dados geométricos de menor granularidade. Em seguida uma consulta para captura de valores não redundantes é realizada com auxílio de um componente *Unique rows*. Os resultados obtidos pelo último componente mencionado são distribuídos como cópias exatas para outros dois componentes, sendo eles instâncias de *Select values* e *Sort rows*. Este procedimento é repetido outras cinco vezes, resultando na total separação dos dados espaciais de acordo com seus níveis de granularidade.

Após a normalização da estrutura dos dados, os componentes *Select values* são utilizados para configuração das colunas. As operações realizadas por estes *Steps* são a remoção, renomeação e alteração do tipo da coluna, bem como alteração do cumprimento e precisão de seus valores.

Neste momento é iniciado o processo para geração de valores agrupados. As operações de união ocorrem após a ordenação dos valores identificadores, os quais são utilizados como critério de agrupamento. Tais operações são realizadas por *Steps* do tipo *Group By* e têm como resultado a criação dos valores numéricos *perimetro*, *area* e *numobj*, sendo o último utilizado para apresentar o número de elementos que compõem o objeto espacial.

Uma quarta coluna com o nome *geom* é gerada pelo componente *Group By*. Os dados suportados por esta coluna são do tipo geométrico, e são utilizados para representar formas de objetos espaciais. Os dados gerados pelo componente neste projeto são resultados de uniões geométricas realizadas sobre dados de menor granularidade. Além da união geométrica, as operações sobre dados espaciais coleção geométrica e *bounding box* também são suportadas. Na Figura 13 é mostrada a tela de configuração do componente em questão.

Após a geração das novas geometrias a partir de uniões geométricas, os valores são devidamente agrupados em seus níveis espaciais através de componentes *Merge Join* e suas colunas são configuradas através de *Select values*.

A última operação do projeto de ETL é realizada por componentes *Table output*, e diz respeito a carga dos dados na dimensão espacial. Para tanto, é feita a configuração do componente atribuíndo a este uma conexão com o SGBD do *Spatial Data Warehouse*, além do nome da tabela sobre a qual serão depositados os dados.

4.5 ENCADEAMENTO DE TRANSFORMAÇÕES

O *software* GeoKettle permite que transformações e vários outros processos sejam encadeados em projetos de ETL. Isto é possível com o uso de *Jobs*, que além de um aprimorado controle do fluxo dos dados, facilita o envio de informações relacionadas a transformações aos interessados. O fluxo dos dados e a disposição dos componentes do *Job* deste projeto são mostrados na Figura 14.

Outra possibilidade é o uso de variáveis de ambiente ou locais para configuração de caminhos de diretórios. O uso deste recurso garante a flexibilidade do projeto, permitindo que este seja implantado em outros ambientes sem que haja a necessidade de reconfiguração dos componentes. Este projeto de ETL foi totalmente construído com utilização de variáveis para atribuição de caminhos de diretórios.

Todo *Job* é iniciado a partir de um componente *START*, que além de servir como um ponto de partida, permite que a execução automática do projeto seja agendada para intervalos diários, semanais ou mensais. Este *Step* é apresentado na Figura 14 como um triângulo verde.

A primeira operação efetiva deste *Job* é a verificação se um diretório está vazio ou não. Este teste é feito através do componente *Check if a folder is empty* e sua tela de configuração é mostrada na Figura 15. Pode-se observar que este *Step* está relacionado com outros dois através de flechas verde e vermelha. Na ferramenta GeoKettle o fluxo das transformações seguem as flechas verdes caso sejam atendidas todas as condições impostas no componente, caso contrário o próximo componente ligado a flecha vermelha será executado.

No *Job* mostrado na Figura 14, os componentes *Delete folders* que têm como função remover diretórios de forma recursiva, são executados caso a condição dos componentes *Check if a folder is empty* não seja satisfeita. Após sua execução, o componente *Create a folder* é acionado recriando o diretório. Caso o componente *Check if a folder is empty* tenha verificado que o diretório está vazio e preparado para receber arquivos resultantes de transformações, o fluxo segue em direção do componente *Transformation*, que executa uma transformação.

Na Figura 16 é mostrado a tela de configuração do *Step Transformation*, o qual tem como função iniciar a execução de uma transformação. Como se pode observar na Figura, a variável *Internal.Job.Filename.Directory* é utilizada para apontar dinamicamente o caminho do arquivo a ser executado pelo componente. Uma vez que esta variável aponta para o diretório no qual o próprio *Job* está presente, mostra-se uma boa prática armazenar o *Job* no diretório

raiz do projeto ETL, pois desta forma o uso desta variável será possível em qualquer ponto do projeto.

A ferramenta GeoKettle permite que seja feito balanceamento de carga em projetos ETL. Dessa forma, transformações ou mesmo *Jobs* podem ser distribuídos em diversos servidores remotos e então executados por um servidor mestre. Outra recurso oferecido é a geração de *logs* com vários níveis de detalhamento para monitoramento de transformações. A configuração de ambos os recursos citados pode ser feita através de assistentes ou mesmo nos próprios componentes.

A automação neste *Job* é completada com auxílio dos *Steps Table exists*, *Truncate table* e *Delete folders*. Os dois primeiros garantem a continuidade da execução das transformações tratando insconsistências no banco de dados. O último componente realiza limpezas no sistema de arquivos, eliminando arquivos temporários gerados pelas transformações.

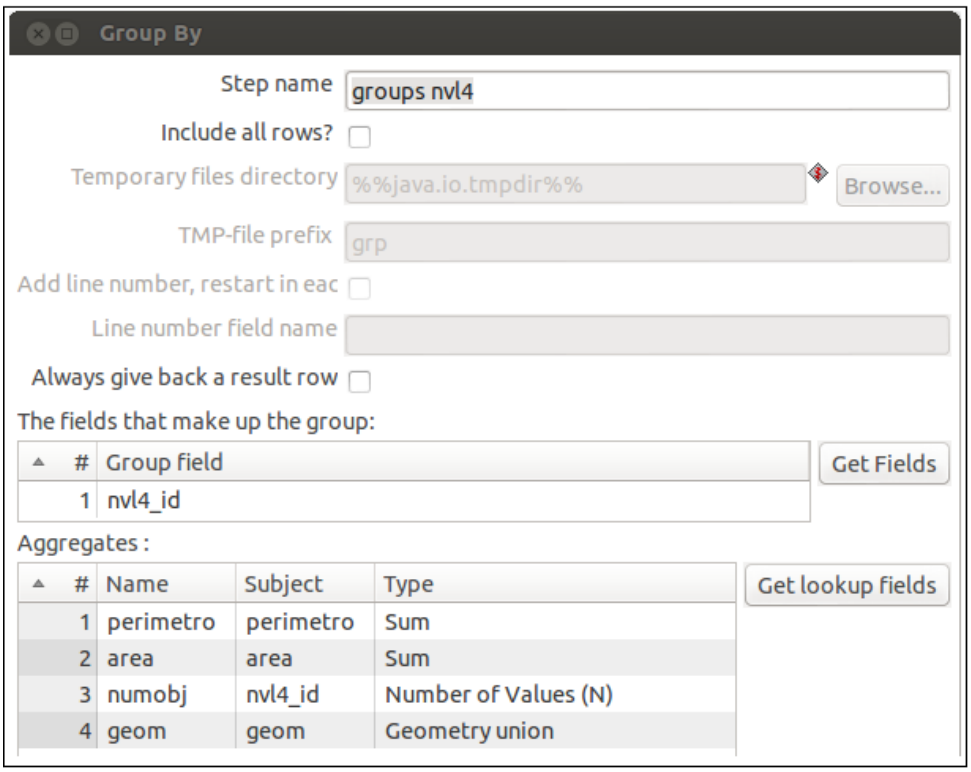

Figura 13: Operações de agrupamento com o componente *Group By*.

Fonte: Autoria própria.

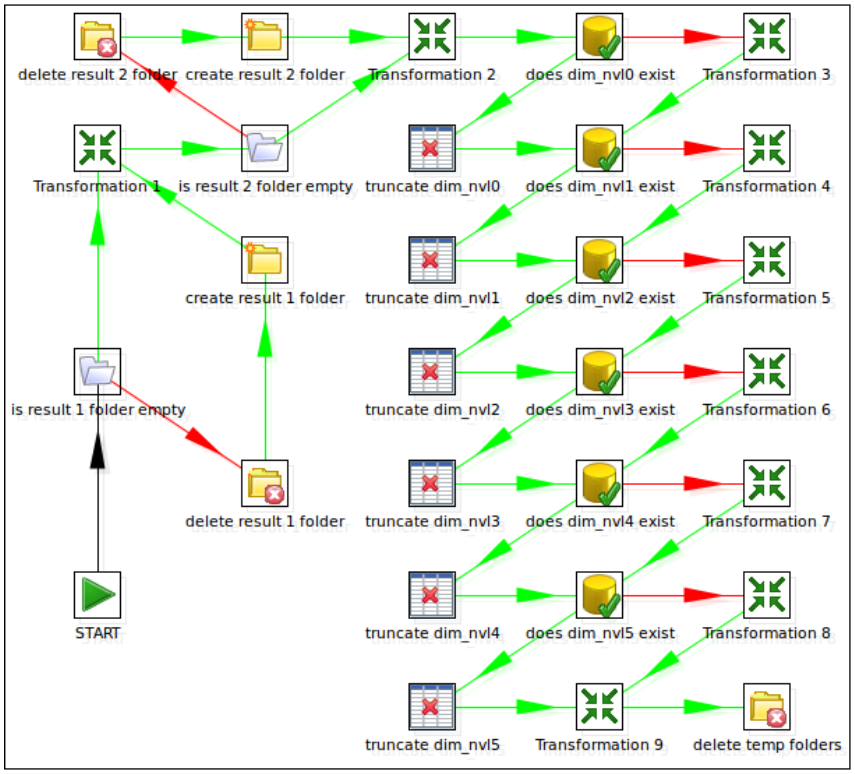

Figura 14: Componentes do *Job* deste projeto de ETL.

Fonte: Autoria própria.

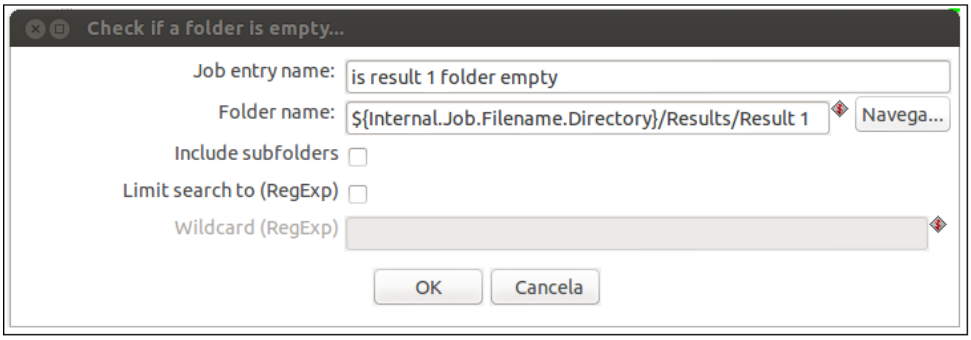

Figura 15: Verificação quanto a existência de um diretório.

Fonte: Autoria própria.

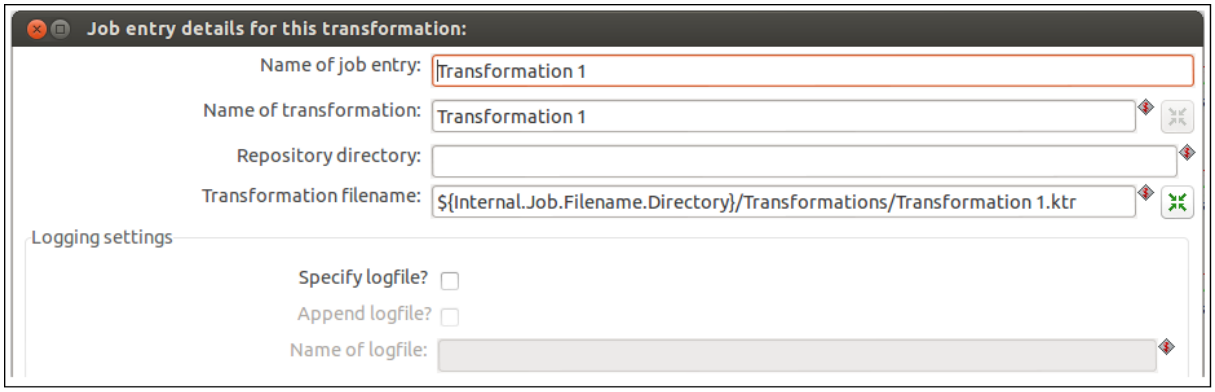

Figura 16: Tela de configuração do *Step Transformation*.

Fonte: Autoria própria.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÃO

A ferramenta GeoKettle mostrou-se de fácil manuseio e rica em recursos, podendo satisfazer todos os requisitos identificados, os quais foram a reconfiguração dos atributos dos dados, limpeza de ruídos, manutenção de integridade, reestruração dos dados, geração de dados espaciais por operações de agrupamento, geração de valores numéricos por meio de cálculos, implantação dos resultados em SGBDs e limpeza da *Staging Area* para liberação de recursos do sistema.

A interface amigável, simplicidade de uso e alta qualidade do material de aprendizado disponibilizado na WEB e na própria ferramenta, permitiram que as funcionalidades básicas do *software* GeoKettle fossem dominadas em pouco tempo.

O SGBD PostgreSQL comportou o conjunto inteiro dos dados espaciais resultantes das transformações. O complemento PostGIS permitiu que fosse criado em cada nível hierárquico da dimensão espacial a coluna geométrica necessária para depósito dos dados espaciais. Portanto os requisitos do sistema responsável pelo armazenamento dos resultados foram atendidos em sua totalidade.

Por conta da facilidade de uso, consistência dos resultados, alta qualidade dos materiais oficiais de aprendizado, riqueza de funcionalidades e principalmente por terem satisfeitos todos os requisitos apresentados no estudo de caso, as ferramentas utilizadas atenderam as espectativas e levaram a conclusão de que é possível construir *Spatial Data Warehouses* utilizando apenas *softwares* livres.

5.2 TRABALHOS FUTUROS

A ferramenta GeoKettle permite que *Transformations* e *Jobs* sejam executados remotamente enviando-os em formato XML a um servidor. Isto é possível através do *software* Carte, o qual vem instalado com a ferramenta de ETL e pode ser acessado via serviços Web.

Para continuidade deste trabalho é proposto um comparativo de desempenho da execução de *Jobs* onde um só computador está presente e em arquiteturas onde diversos servidores somam poder de processamento.

REFERÊNCIAS

AGUIAR, Gustavo Maia. [S.I.]: **Por que utilizar uma ferramenta de ETL?**, 2010. Disponível em: <http://gustavomaiaaguiar.wordpress.com/2010/05/10/por-que-utilizar-uma-ferramenta-de-etl/>. Acesso em: 25 jul. 2010, 22:00.

ANZANELLO, Cynthia Aurora. **OLAP**: Conceitos e Utilização, Porto Alegre, 2009.

BÉDARD, Yvan; MERRETT, Tim; HAN, Jiawei. **Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery, Geographic Data Mining and Knowledge Discovery**, 2001, p.53-73.

CROSSLAND, Martin D.; WYNNE, Bayard E. **Measuring and testing the effectiveness of a Spatial Decision Support System**. 1994.

GADM. Disponível em: <http://www.gadm.org/>. Acesso em: 15 jun. 2013, 18:00.

GARG, Nipun; MITHAL, Surabhi. **Spatial Data Warehouses**: A survey. 2010.

GeoKettle. Disponível em: <http://www.spatialytics.org/projects/geokettle/>. Acesso em: 15 jun. 2013, 20:00.

GreatSampleResume. Disponível em: <http://www.greatsampleresume.com/Job-Responsibilities/Decision-Support-Analyst-Responsibilities.html>. Acesso em: 28 jul. 2013. 21:00.

INMON, William H. **Building the Data Warehouse**: Getting started. Nova Iorque. 2000.

INMON, William H. **Building the Data Warehouse**. 3. ed. Nova Iorque, 2000.

KARABEGOVIC, Almir; PONJAVIC, Mirza. **Geoportal as Decision Support System with Spatial Data Warehouse**. 2012. P. 915-918.

KIMBALL, Ralph et all. **The Data Warehouse Lifecycle Toolkit**: Expert Methods for Designing, Developing and Deploying Data Warehouses. 1998. 771p.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit**: The complete guide to dimensional modeling. 2. ed. 2002. 447p.

MALINOWSKI, Elzbieta; ZIMÁNYI, Esteban. **Representing Spatiality in a Conceptual Multimensional Model**. Nova Iorque, 2004.

PostGIS. Disponível em: <http://postgis.net/>. Acesso em: 29 jun. 2013. 17:30.

PostgreSQL. Disponível em: < http://www.postgresql.org.br/sobre>. Acesso em: 29 jun. 2013. 15:30.

POWER, Dan. Disponível em: <http://dssresources.com/faq/index.php?action=artikel&id=9>. Acesso em: 15 jun. 2013. 21:20.

RIVEST, Sonia et all. Solap: **A new type of user interface to support spatio-temporal multidimensional data exploration and analysis**. 2003. 8p.

SAUTER, Vicki L. **Decision Support Systems for Business Intelligence**. Wiley, 2011.

SHEKHAR, Shashi et all. **Trends in Spatial Data Mining**. Em: Data Mining: Next Generation Challenges and Future Directions. Minneapolis, 2003. 24p.

SHNEIDER, Markus. **Spatial Data Types for Database Systems**. 1997. 282p.

SODRÉ, Patricia. Disponível em: <http://businessintelligencebrasil.com.br/devo-utilizar-uma-ferramenta-de-etl/ >. Acesso em: 20 jun. 2013.

UNILINS. Disponível em: <ftp://ftp.unilins.edu.br/tuca/Banco%20de%20Dados/Modelagem%20de%20Banco%20de%20dados.pdf >. Acesso em: 21 jun. 2013.

UNILINS. Disponível em: <ftp://ftp.unilins.edu.br/wagner/Projeto%20de%20Banco%20de%20Dados/Projeto%20de%20Banco%20de%20Dados.pdf>. Acesso em: 21 jun. 2013.

ZIMÁNYI, Esteban; MALINOWSKI, Elzbieta. **OLAP Hierarchies**: A Conceptual Perspective. Brussels, 2004. 15p.

APÊNDICES

APÊNDICE A – SCRIPTS SQL DA DIMENSÃO ESPACIAL

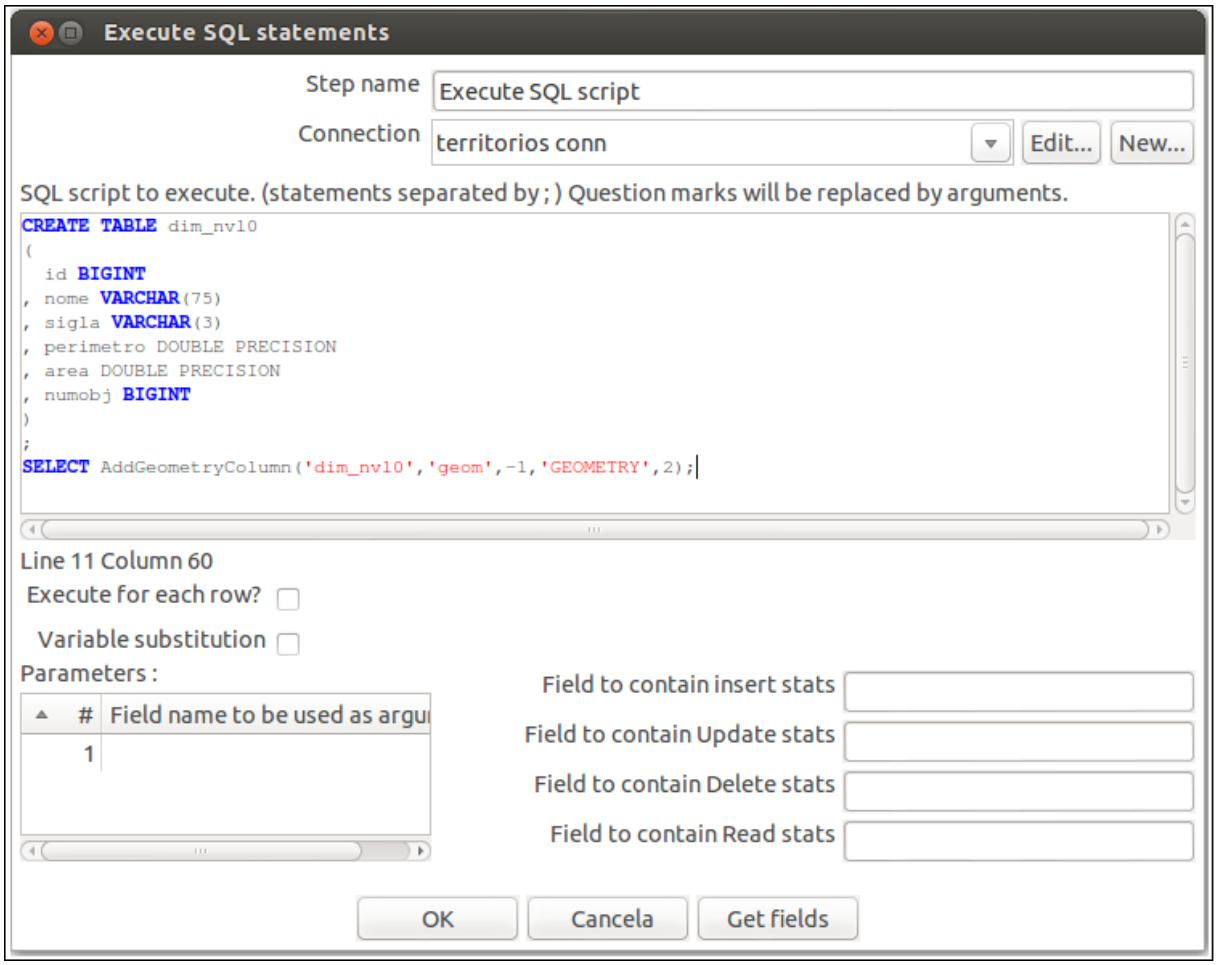

Figura 17: Script de criação da tabela dim_nvl0 e coluna geométrica.

Fonte: Autoria própria.

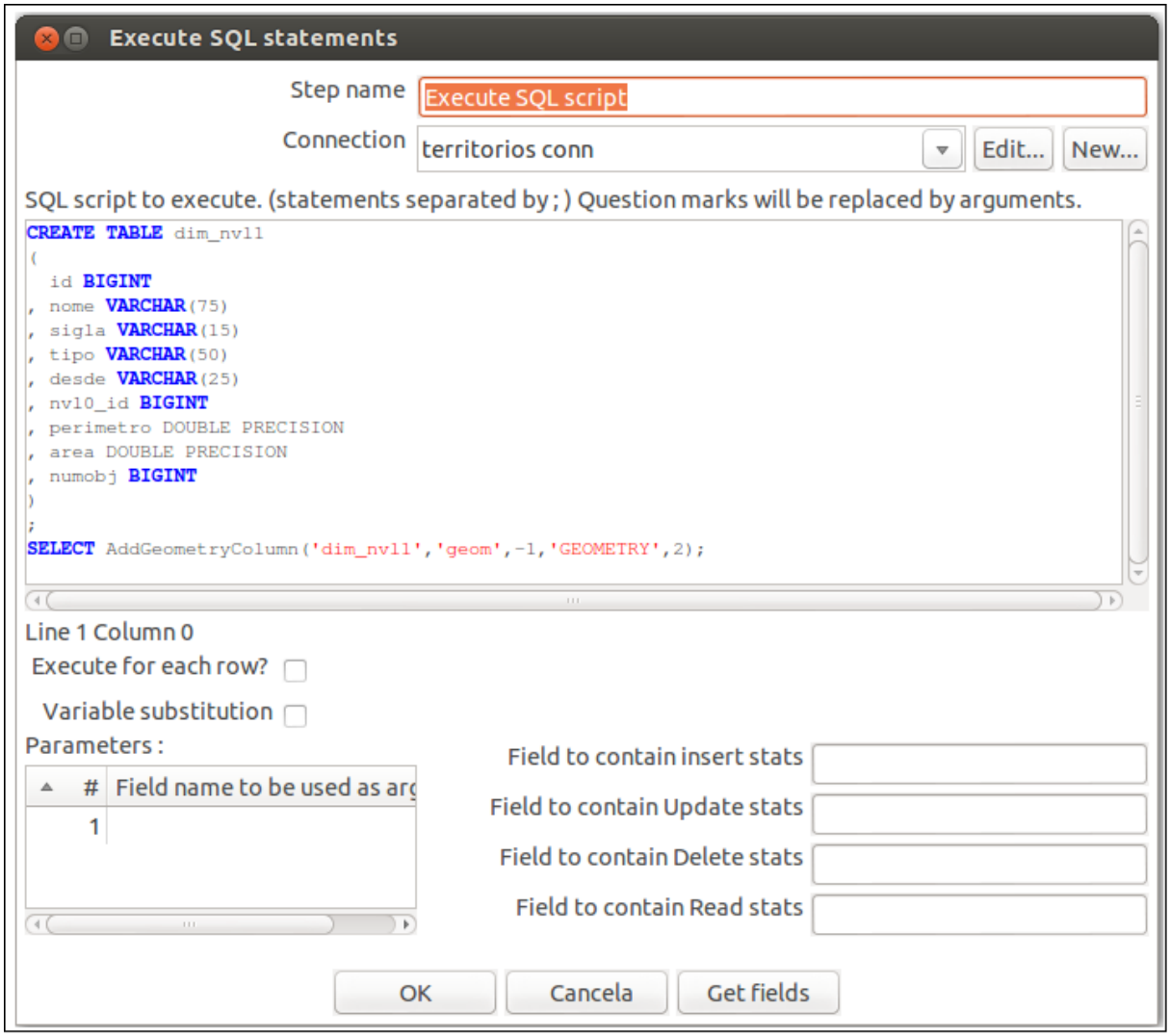

Figura 18: Script de criação da tabela dim_nvl1 e coluna geométrica.

Fonte: Autoria própria.

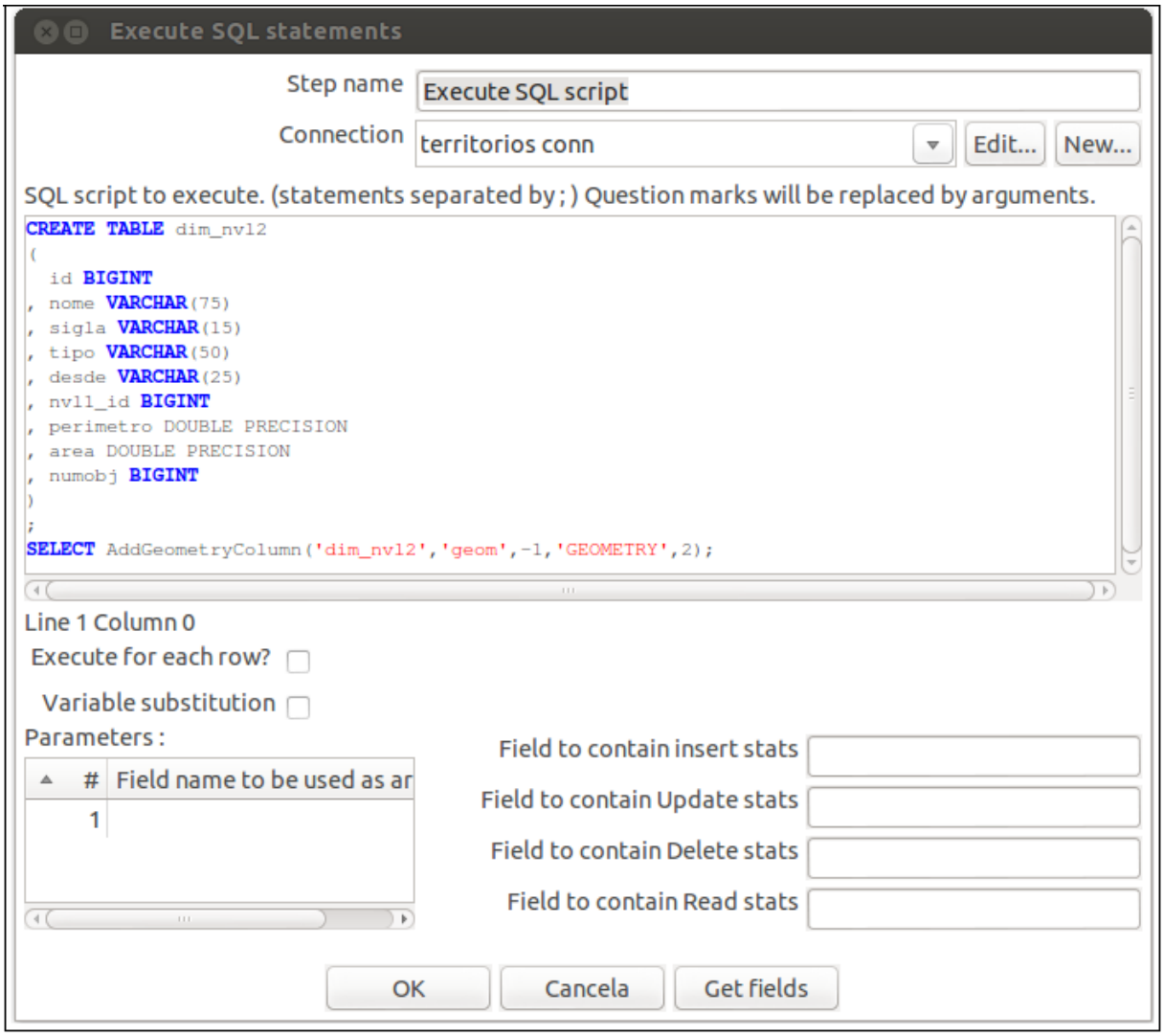

Figura 19: Script de criação da tabela dim_nvl2 e coluna geométrica.

Fonte: Autoria própria.

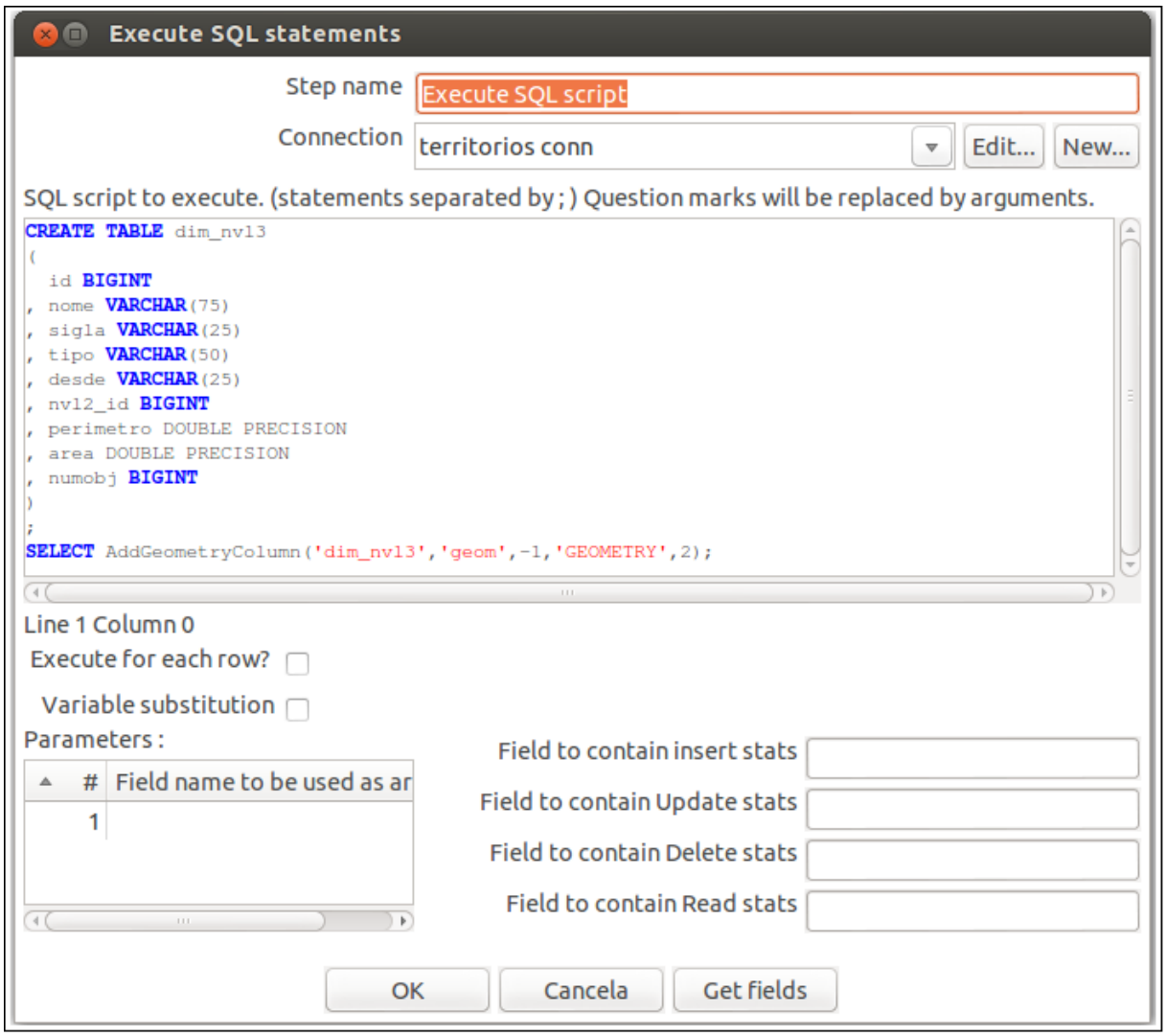

Figura 20: Script de criação da tabela dim_nvl3 e coluna geométrica.

Fonte: Autoria própria.

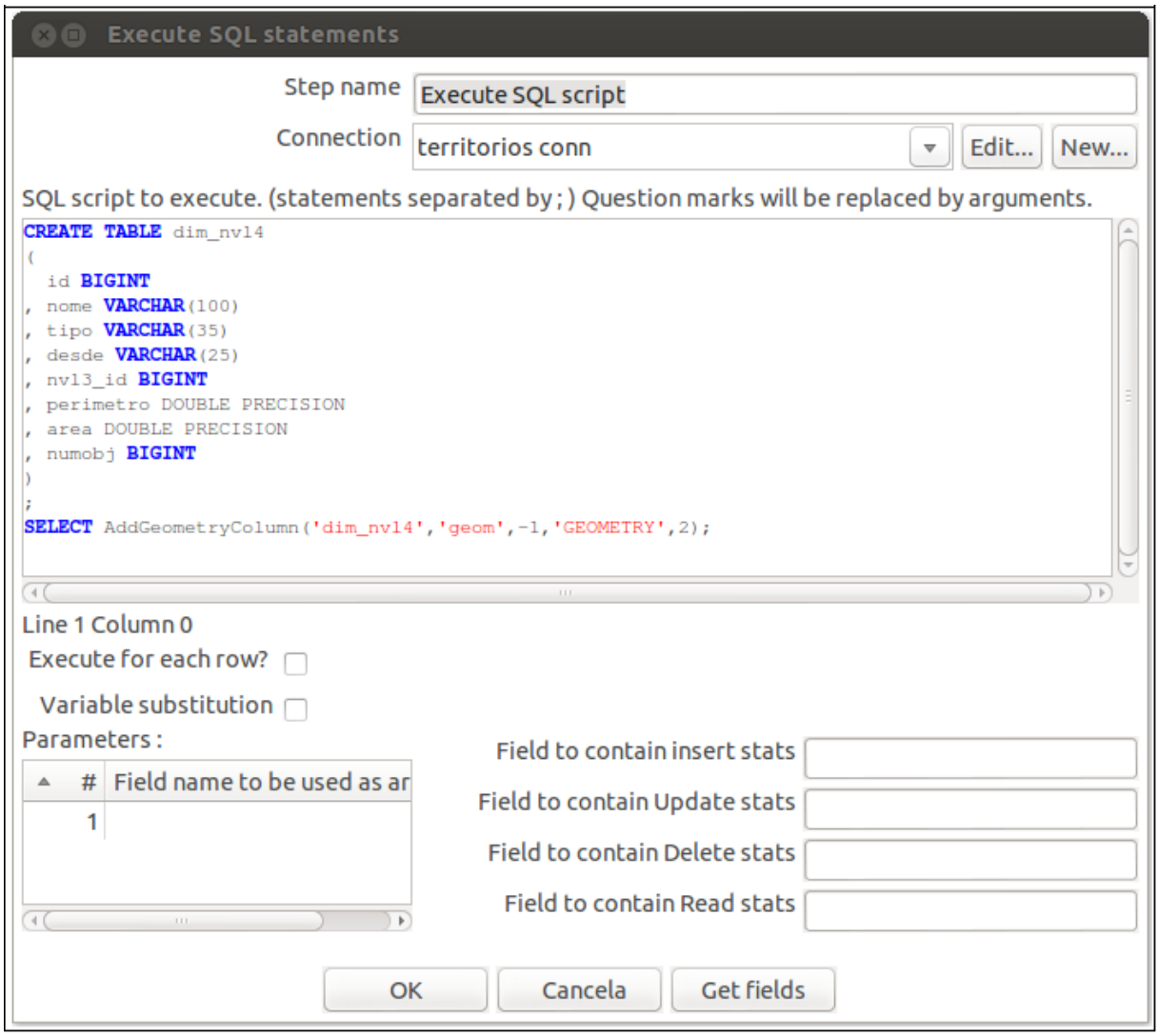

Figura 21: Script de criação da tabela dim_nvl4 e coluna geométrica.

Fonte: Autoria própria.

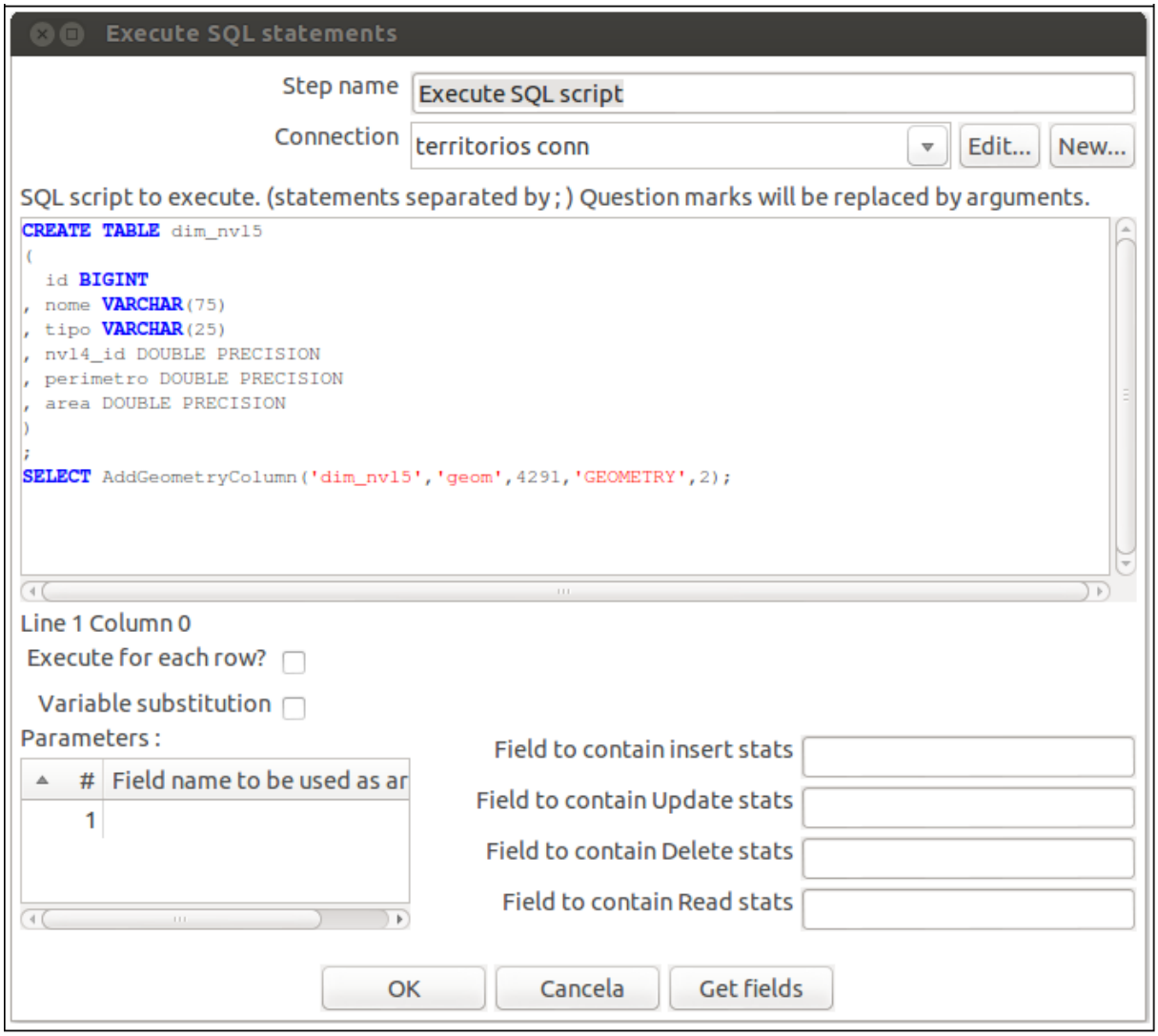

Figura 22: Script de criação da tabela dim_nvl5 e coluna geométrica.

Fonte: Autoria própria.