

**UNIVERSIDADE FEDERAL DE PELOTAS**

**Instituto de Informática  
Ciência da Computação**



**Trabalho acadêmico**

**MODELAGEM MULTIDIMENSIONAL COMO SOLUÇÃO PARA  
TRATAMENTO DE AGLOMERADOS DE DADOS NÃO ESTRUTURADOS:  
o caso do Banco de Dados para atração de investimentos da Prefeitura  
Municipal de Pelotas**

**Rita Cristina Galarraga Berardi**

Pelotas, 2006

**RITA CRISTINA GALARRAGA BERARDI**

**MODELAGEM MULTIDIMENSIONAL COMO SOLUÇÃO PARA  
TRATAMENTO DE AGLOMERADOS DE DADOS NÃO ESTRUTURADOS:  
o caso do Banco de Dados para atração de investimentos da Prefeitura  
Municipal de Pelotas**

Trabalho acadêmico apresentado ao curso de Bacharelado em Ciência da Computação do Instituto de Informática, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof<sup>a</sup>. MSc. Flávia Braga de Azambuja

Co-orientador: Prof<sup>a</sup>. MSc. Ana Marilza Pernas Fleischmann

Pelotas, 2006

Dados de catalogação na fonte:  
Ubirajara Buddin Cruz – CRB-10/901  
Biblioteca de Ciência & Tecnologia - UFPel

Berardi, Rita Cristina Galarraga

Modelagem multidimensional como solução para tratamento de aglomerados de dados não estruturados : o caso do banco de dados para atração de investimentos da Prefeitura Municipal de Pelotas / Rita Cristina Galarraga Berardi ; orientador Flávia Braga de Azambuja ; co-orientador Ana Marilza Pernas Fleischmann. – Pelotas, 2006. – 149f. ; il. color. – Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2006.

1.Informática. 2.Banco de Dados Multidimensionais. 3.Data warehouse. 4.Sistemas de informação. 5.Sistemas de apoio a decisão. I.Azambuja, flávia Braga. II.Fleischmann, Ana Marilza Pernas. III.Título.

CDD: 005.74

**Banca examinadora:**

.....  
Prof<sup>a</sup>. MSc. Flávia Braga de Azambuja

.....  
Prof<sup>a</sup>. MSc. Ana Marilza Pernas Fleischmann

.....  
Prof. MSc. Rafael Krolow

Dedico esta conquista à minha mãe, e ao meu pai, sim, a ele também, pois sei que de algum lugar, de alguma maneira deve estar vendo que obedeci ao conselho dado pouco antes de partir : priorizar os estudos. É por vocês que cheguei aqui e será por vocês que seguirei em frente.

## AGRADECIMENTOS

Primeiramente a Deus, que me deu a vida e todos os sentimentos que me impulsionam a lutar e ser alguém. Obrigada por sempre colocar no meu caminho as pessoas certas nos momentos certos.

O maior agradecimento é para minha mãe que é um exemplo de vida para mim. Obrigada mãe por seres sempre a estrela que ilumina a minha vida, pelos teus ensinamentos, conselhos e por estares sempre comigo! Agradeço por toda a tua incondicional e constante dedicação para que nunca me faltasse nada, principalmente o estudo.

Ao meu namorado Rodrigo Barros, que mesmo quando ainda não era meu namorado sempre me ofereceu apoio, respeito e tentou me fazer uma pessoa feliz, e sempre conseguiu. Agradeço pela paciência e compreensão principalmente durante o desenvolvimento deste trabalho final. Obrigada pela valiosa ajuda!

Ao meu amigo Eduardo Gastal. Agradeço pela amizade mais verdadeira, sincera, leal e talvez a única que conheci nesta etapa da minha vida. A ti Ed, desejo muito sucesso!

Agradeço aos meus colegas de aula pela construtiva convivência, principalmente àqueles que algum dia me receberam em sua casa, dispostos a ajudar quando tinha dúvidas da matéria. Mesmo que minha vida tenha tomado rumo diferente da de vocês, reconheço que foram importantes na minha formação, e por isso agradeço verdadeiramente e desculpem o incômodo.

Às minhas orientadoras, Flávia Azambuja e Ana Marilza, com as quais aprendi muito na construção deste trabalho. À Professora Flávia que além de orientadora tornou-se uma grande amiga, que me ajudou em momentos difíceis no curso, sem eu nunca precisar pedir. Obrigada por ter apostado em mim e eu a perdão pelas vezes que me bloqueou no MSN.

Agradeço também ao Professor Ricardo Silveira por ter acreditado em mim e confiado a mim importantes responsabilidades enquanto era meu orientador no grupo de pesquisa.

Obrigada a todos que torcem por mim e que direta ou indiretamente contribuíram com a minha formação tanto profissional quanto pessoal. Peço perdão

pelas minhas ausências, falta de tempo, e meus momentos estressantes, mas saibam que nada foi em vão, porque eu consegui!

## Resumo

BERARDI, Rita Cristina Galarraga. **Modelagem multidimensional como solução para tratamento de aglomerados de dados não estruturados: o caso do Banco de Dados para atração de investimentos da Prefeitura Municipal de Pelotas** 2006. 149f. Trabalho acadêmico (Graduação) – Instituto de Informática. Universidade Federal de Pelotas, Pelotas.

O tratamento estratégico de dados é uma prática cada vez mais comum e necessária para as organizações tomarem suas decisões a fim de manterem a competitividade e a qualidade dos serviços que oferecem. Os Sistemas de Informação contribuem com esta prática, pois a partir de uma base de dados adequada é possível proporcionar aos gerentes de negócios a análise crítica dos dados para o auxílio na tomada de decisão. Mas para que o Sistema de Informação cumpra seu papel é extremamente importante que este utilize um Banco de Dados apropriado que agregue valor aos dados da organização, transformando-os em informações úteis. A administração da Prefeitura Municipal de Pelotas identificou a necessidade de obter dados sobre os segmentos sociais para disponibilizá-los a empresários atraindo seus interesses em investir na região. Apesar da Prefeitura ter construído um material muito rico em informações, este apresentava problemas que comprometiam a sua funcionalidade. Este trabalho é destinado ao desenvolvimento de um modelo de Banco de Dados que atenda os requisitos de informação da Prefeitura Municipal de Pelotas, a fim de implementar um sistema de informação em ambiente web. O estudo realizado sobre os dados coletados, bem como o tipo de Sistema de Informação apropriado e as técnicas de Banco de Dados para a modelagem está descrito neste trabalho. Com base nos estudos realizados foi desenvolvido o modelo da base de dados para o sistema da Prefeitura, utilizando a técnica de Banco de Dados Multidimensionais agregada à técnica de Data Warehousing, por estas possuírem as características que um Sistema de Apoio a Decisão exige. Os resultados foram considerados positivos. Os testes foram realizados com simulações de consultas criadas por investidores no Data Warehouse através das quais foi comprovada a funcionalidade da base de dados modelada, pois eliminou os problemas que o Banco de Dados da Prefeitura apresentava, podendo então esta base ser implementada em um ambiente Data Warehouse.

**Palavras-chave:** Banco de Dados Multidimensionais. Data Warehouse. Sistemas de Informação. Sistemas de Apoio a Decisão.



## Abstract

BERARDI, Rita Cristina Galarraga. **Modelagem multidimensional como solução para tratamento de aglomerados de dados não estruturados: o caso do Banco de Dados para atração de investimentos da Prefeitura Municipal de Pelotas** 2006. 149f. Trabalho acadêmico (Graduação) – Instituto de Informática. Universidade Federal de Pelotas, Pelotas.

To deal with strategic data has become an ordinary and important task where the institutions make decisions in order to increase the competitiveness and quality of the services they provide. The Information Systems make this task possible because they allow critical analysis of the data through a well-constructed database, supporting the business manager in their decision taking. So the Information System can fulfill its part, it is extremely important that it makes use of an appropriated database, that can gather value to the institution's data, transforming them in useful information. The administration sector of Pelotas City Hall felt the need of gathering data from several social segments and make them available in order to attract investments to the region. Though the City Hall built a very detailed collection of data, the problems that it held were compromising its functionality. This monograph's goal is the development of a Database model that fulfills the needs of the City Hall, aiming the implementation of a web-based Information System. The monograph contains all the study made about the collected data, as well as the suitable Information System type and database techniques chosen. Based on the research made, the City Hall database model was developed by using the Data Warehouse database technique allied to the Data Warehousing one, as their characteristics matched the ones needed in a Decision Support System. The results were considered positive. The tests were realized through simulations of queries made by business men in the Data Warehouse, which proved the functionality of the modeled database, due to the elimination of the problems that the City Hall database presented, making it possible to implement this database in a Data Warehouse environment.

**Keywords:** Multidimensional Database. Data Warehouse. Information System. Decision Support System.

## LISTA DE FIGURAS

Figura 1 - Principais características entre sistemas operacionais e dimensionais.....	18
Figura 2 - Esquema da Arquitetura <i>top-down</i> .....	22
Figura 3 - Esquema da Arquitetura <i>bottom-up</i> .....	24
Figura 4 - Modelo de implementação combinada das arquiteturas.....	25
Figura 5 - Granularidade de dados.....	26
Figura 6 - Esquema estrela da tabela de Fatos Matrícula.....	28
Figura 7 - Esquema floco de neve da Dimensão Região.....	29
Figura 8 - O processo de <i>Data Warehousing</i> .....	32
Figura 9 - Tipos de metadados.....	35
Figura 10 - Ferramentas de acesso aos dados.....	38
Figura 11 - Principais diferenças entre OLAP e OLTP.....	39
Figura 12 - Operações <i>Drill Down</i> e <i>Drill Up</i> ou <i>Roll Up</i> .....	41
Figura 13-- Operação <i>Slice and Dice</i> .....	42
Figura 14 - Operação <i>Pivoting</i> .....	43
Figura 15 - Tela do Allfusion ERwin.....	44
Figura 16 - Tela do Banco de Dados da Prefeitura.....	48
Figura 17 - Esquema Estrela do Fato Matrícula.....	54
Figura 18 - Carregamento dos dados com o uso de Enterprise Manager.....	56
Figura 19 - Organização dos cubos nos <i>Data Marts</i> no Analysis Manager.....	57
Figura 20 - Descrição da estrutura do cubo Matrículas na ferramenta <i>Cube Editor</i> . 57	
Figura 21 - Representação da disposição dos dados no DW.....	58
Figura 22 - Fatia do cubo com o resultado da consulta.....	58
Figura 23 - Disposição dos membros da dimensão Tempo.....	59
Figura 24 - Resultado da consulta no cubo Matrículas.....	62
Figura 25 - Resultado da consulta no cubo Matrículas.....	63

## LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
CASE	<i>Computer Aided Software Engineering</i>
DDL	<i>Data Definition Language</i>
DM	<i>Data Mart</i>
DOLAP	<i>Desktop On-Line Analytic Processing</i>
DW	<i>Data Warehouse</i>
ER	Modelo Entidade-Relacionamento
ETL	Extração, Transformação e Carga de dados
FIERGS	Federação das Indústrias do Estado do Rio Grande do Sul
HOLAP	<i>Hybrid On-Line Analytic Processing</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDH	Índice de Desenvolvimento Humano
ITEPA	Instituto Técnico de Pesquisa e Assessoria da UCPEL
MOLAP	<i>Multidimensional On-Line Analytic Processing</i>
OLAP	<i>On-Line Analytic Processing</i>
OLTP	<i>On-Line Transaction Processing</i>
ROLAP	<i>Relational On-Line Analytic Processing</i>
SAD	Sistemas de Apoio à Decisão
SEDAI	Secretaria do Desenvolvimento e dos Assuntos Internacionais do Rio Grande do Sul
SGBD	Sistema Gerenciador de Banco de Dados
SIE	Sistema de Informação Executivo
SQL	<i>Structured Query Language</i>
UCPEL	Universidade Católica de Pelotas

## SUMÁRIO

1 INTRODUÇÃO .....	13
2 FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 Sistemas de Apoio à Decisão (SADs).....	15
2.1.1 Estruturas de SADs.....	16
2.1.2 Armazenamento de dados .....	17
2.1.2.1 Banco de Dados Relacional x Banco de Dados Multidimensional .....	18
2.2 Data Warehouse (DW).....	19
2.2.1 Características do Data Warehouse.....	21
2.2.2 Arquiteturas de DW .....	22
2.2.3 Granularidade dos dados .....	27
2.2.4 Modelagem Multidimensional .....	28
2.2.5 O Processo de Desenvolvimento do Data Warehouse .....	31
2.2.6 Extração, transformação e carga dos dados .....	33
2.2.7 Metadados.....	35
2.2.8 Ferramentas de Data Warehouse .....	38
2.2.8.1 Ferramentas OLAP.....	39
2.2.8.2 Ferramentas de Projeto de DW.....	44
2.2.9 Ambiente de Data Webhouse.....	46
2.2.9.1 Data Warehouse na web .....	46
3 ESTUDO DE CASO: PREFEITURA MUNICIPAL DE PELOTAS.....	48
3.1 O Banco de Dados da Prefeitura Municipal de Pelotas.....	48
3.1.1 Coleta dos dados .....	49
3.2 Definição do Problema .....	50
3.3 Alternativa de solução .....	51
3.4 Modelagem do Data Warehouse.....	52
3.4.1 Análise dos arquivos .....	52
3.4.2 Modelagem e dimensionalidade dos dados .....	53

3.4.3 Protótipo da modelagem desenvolvida .....	55
3.4.3.1 Metadados considerados no sistema .....	60
4 ANÁLISE DOS RESULTADOS OBTIDOS .....	62
5 CONCLUSÕES E TRABALHOS FUTUROS .....	65
5.1 Conclusões.....	65
5.2 Trabalhos futuros .....	66
REFERÊNCIAS.....	67
APÊNDICES.....	70

## INTRODUÇÃO

A informação aliada aos recursos da tecnologia é uma necessidade primária e elementar para a funcionalidade tática, estratégica e operacional das empresas. Portanto, para as organizações que pretendem se destacar no mercado - que é tão competitivo - é essencial saber usar a informação como ferramenta estratégica nas suas decisões. Na "era da informação", saber como encontrar as informações, como apresentá-las e utilizá-las é tão importante quanto conhecê-las. Já prevendo o excesso da informação on line e a dificuldade para se conhecer fontes de qualidade, afirmava-se de forma peculiar que no contexto do gerenciamento de informações, 20% das informações eram responsáveis por 80% das decisões de uma organização, no objetivo de demonstrar a importância da informação de qualidade, a qual é vital hoje em dia, em plena era da Internet (KANAZAWA, 2006)

A equipe de desenvolvimento regional da Prefeitura Municipal de Pelotas identificou esta necessidade de obter dados suficientes tanto em quantidade quanto em qualidade para apresentar aos investidores auxiliando-os na tomada de decisão sobre o investimento na cidade e região sul. Para suprir esta necessidade, a equipe preparou um conjunto de dados, na forma de apresentações em slides, para oferecer aos empresários internos e externos à cidade. A partir destes dados reunidos, os empresários poderiam decidir se investiriam ou não na cidade, dependendo dos índices apresentados, referentes a sua respectiva área. Para construção deste material de apresentação, a Secretaria de Desenvolvimento buscou informações a partir de bases de dados já existentes, on-line ou não, automatizadas ou não, coletando ainda dados de forma empírica, a fim de agrupar aqueles que não estavam organizados, mas que a administração da cidade julgou pertinente sua apresentação aos interessados.

Apesar de o material ser rico em informações sobre a cidade de Pelotas, apresentou sérios problemas que acabaram prejudicando a funcionalidade que os dados deveriam proporcionar. O que se gerou na realidade um grande aglomerado de dados, sem formatação própria e adequada, que permitisse aos empresários e possíveis investidores o acesso rápido às informações. Este aglomerado de dados

foi organizado em sua forma final sem nenhum tipo de tratamento específico, constituindo-se apenas de arquivos texto formatados em apresentações de slides, tabelas e gráficos. Esta forma de organização impossibilitava a automação de uma base de dados propriamente dita bem como o cruzamento de informações com o objetivo de descoberta de conhecimento. Então a equipe da Secretaria motivou-se a obter uma solução para reunir dados suficientes e relevantes, capazes de traçar um abrangente e real perfil da cidade e da região, e que possibilitasse a demonstração clara e objetiva da situação dos vários segmentos da sociedade a fim de estimular investimentos. Com este sistema, os investidores teriam acesso a dados estatísticos e históricos da cidade de Pelotas e região sul, que seriam atualizados de forma dinâmica, facilitando então a tomada de decisão com relação ao investimento.

Para se obter uma base de dados que abranja o máximo de informação possível sobre a cidade de Pelotas, utilizando bases legadas e com alta capacidade de análise sobre essas informações este trabalho propõe-se a analisar os dados da Prefeitura e, com base nos conceitos de Banco de Dados (BDs), definir qual é a técnica mais adequada para o armazenamento e tratamento destes tipos de dados que a Prefeitura têm em sua coletânea. A partir desta definição objetiva-se a criação de um modelo de Banco de Dados automatizado em um Sistema de Informação *web*, que atenda os requisitos de informação da Prefeitura.

O presente trabalho é composto por cinco capítulos e três anexos. No capítulo 1 é apresentada a justificativa de se abordar o tratamento estratégico dos dados neste trabalho, juntamente com o problema abordado e como se espera solucioná-lo. O capítulo 2 desta monografia apresenta os conceitos de técnicas de Banco de Dados referentes aos Sistemas de Apoio a Decisão (SAD) e toda a fundamentação teórica em que se baseou o desenvolvimento da modelagem. No capítulo 3 é apresentado o estudo de caso da Prefeitura Municipal de Pelotas, a definição do problema e o processo utilizado para desenvolver a modelagem. O capítulo 4 apresenta os resultados obtidos com a modelagem através de um comparativo da organização inicial dos dados com a nova proposta no trabalho. No capítulo 5 é feita uma conclusão geral a respeito do trabalho desenvolvido e são sugeridos trabalhos futuros.

## **FUNDAMENTAÇÃO TEÓRICA**

Este capítulo apresenta a descrição dos elementos estudados para atingir os objetivos definidos. Dentre os quais estão a definição de Sistema de Apoio a Decisão, conceitos de *Data Warehouse* e a justificativa de se utilizar a técnica multidimensional para a base de dados deste Sistema de Informação. Por fim são apresentadas as ferramentas que possibilitam a manipulação de dados armazenados neste tipo de base, bem como aquelas ferramentas que auxiliam no desenvolvimento da base propriamente dita. A abordagem destes temas foi realizada por entender-se que se enquadram na resolução do problema abordado.

### **1.1 Sistemas de Apoio à Decisão (SADs)**

Hoje em dia admite-se que o conhecimento de sistemas de informação é essencial para os administradores, porque a maioria das organizações precisa deles para sobreviver e prosperar. Esses sistemas podem auxiliar as empresas a estender seu alcance a locais distantes, oferecer novos produtos e serviços, reorganizar fluxos de tarefas e trabalho e, talvez, transformar consideravelmente o modo como conduzem os negócios. Com o avanço dos sistemas de informação, tornou-se possível automatizar a realização de algumas tarefas de rotina de forma a melhorar a eficiência dessas, como coleta, processamento, armazenamento e distribuição de dados. A automação dessas ações, as quais são destinadas ao auxílio no processo de tomada de decisão é denominada Sistemas de Apoio a Decisão (SAD) (LAUDON, 2004).

O objetivo dos SAD é o apoio ao administrador, gerente ou gestor nas atividades de tomada de decisão e não substituição do tomador de decisões (ANNES, 2006).



### 2.1.1 Estruturas de SADs

Os processos administrativos são basicamente processos decisórios. As decisões gerenciais afetam diretamente a sobrevivência da empresa e a vida das pessoas que giram em torno dela, sejam elas empregados, acionistas, fornecedores ou clientes. Devido à importância desta prática decisória, as organizações sempre se utilizaram de recursos administrativos e gerenciais para melhor decidir por seus investimentos.

De acordo com Fisher (1998 apud BISPO,1998) os primeiros SADs surgiram nos anos 60 e 70, para dar suporte aos gerentes na solução de problemas gerenciais. Entretanto, estes eram muito caros, de uso muito específico e difíceis de se operar. O principal tipo de estrutura que sustentava estes SADs eram as bases de dados que não possuíam a arquitetura necessária para a realização de pesquisas típicas deste tipo de sistema. Devido à falta de dados históricos, existia muita dificuldade para a criação de relatórios e para se realizar as análises necessárias ao gerenciamento dos negócios. O enfoque não estava no processo de decisão, mas sim no suporte computacional para o desenvolvimento rápido das aplicações.

Nos anos 90 começaram a surgir outros recursos para melhor aproveitamento da estrutura SAD, como as ferramentas *CASE (Computer Aided Software Engineering)* e outras linguagens de programação, as quais prometiam resolver os problemas dos usuários finais que buscavam informações rápidas e que não disponibilizavam de tempo extra para o desenvolvimento de sistemas específicos. Com essas novas ferramentas, foi possível desenvolver sistemas de uma maneira mais rápida e simples, porém, ainda não suficientemente versáteis para atender a todas as necessidades gerenciais. Com o passar dos anos, as empresas foram crescendo, os negócios aumentando e também, conseqüentemente, o volume de dados armazenados. A análise dos dados tornou-se uma tarefa mais complexa e que necessitava de ferramentas mais eficientes no tratamento dos dados. Surgiram, então, ferramentas de Bancos de Dados (BD), Data Warehouse (DW), On-line Analytical Process (OLAP), entre outras, com as quais as consultas e os relatórios passaram a ser confeccionados pelos próprios usuários, sem um conhecimento profundo de tecnologias computacionais (BISPO, 1998).

### 2.1.2 Armazenamento de dados

Um sistema eficiente oferece aos usuários informação pertinente, exata e relevante. Para tanto, os dados precisam ser organizados e mantidos adequadamente, assim os usuários podem acessar e recuperar com facilidade a informação que precisam. Laudon (2004) defende que quando o sistema não utiliza uma abordagem como Banco de Dados (BD) para o gerenciamento das informações, podem surgir vários problemas, tais como:

- redundância e inconsistência dos dados;
- dependência programa/dados;
- falta de flexibilidade;
- baixo nível de segurança;
- falta de compartilhamento e disponibilidade de dados.

De acordo com Elmasri e Navathe (2000) o que torna o banco de dados disponível para todos os usuários além de proporcionar certas restrições de integridade de dados e de restrições de acesso é o Sistema de Gerenciamento de Banco de Dados (SGBD), sendo essa uma das maiores vantagens de se utilizar BD para armazenar os dados. Tão logo a atualização de um usuário seja aplicada ao BD, todos os outros usuários podem imediatamente visualizar essa atualização.

Essa disponibilidade de informações atualizadas é essencial para muitas aplicações de processamento de transações, ou até mesmo em sistemas de informação que não se utilizam de dados transacionais, como os próprios SADs.

Como as organizações precisam cada vez mais de análises rígidas e complexas sobre seus dados, os quais têm sido encontrados em grandes quantidades, a não utilização de BD como alternativa de armazenamento pode prejudicar consideravelmente o tempo e a precisão da tomada de decisão, e conseqüentemente, o uso estratégico dos dados. Um BD atende a uma comunidade mais ampla de usuários do que os sistemas de arquivos tradicionais.

As tecnologias de Banco de Dados são cada vez mais utilizadas em Sistemas de Informação como uma forte e essencial ferramenta, pois com a evolução dos SADs, as organizações podem utilizar dados já armazenados em

bases automatizadas para contribuir com a recuperação de informações e assim obter maior retorno no investimento.

Para possibilitar a utilização integrada de bases de dados surgiu a técnica de *Data Warehouse* (DW), que consiste de um grande depósito de informações, encontrado no cerne do processamento SAD, visando efetiva integração de bases de dados operacionais em um ambiente que permita o uso estratégico de dados (INMON, 1997). Tendo em vista as especificidades das bases dos SAD é importante realizar um comparativo entre as técnicas de modelagem de bases de dados, para que os dados sejam armazenados da forma mais adequada ao seu tipo de manipulação.

#### 2.1.2.1 Banco de Dados Relacional x Banco de Dados Multidimensional

A maioria das técnicas de modelagem concorda que a aplicação completa da teoria relacional não é apropriada para DW. É um erro comum projetistas modelarem o DW de acordo com a teoria relacional e seus princípios. A modelagem relacional é uma das mais utilizadas em sistemas que manipulam transações diárias ou aplicações OLTP (*On-Line Transaction Processing*) com dados operacionais, sendo muito eficiente por suas importantes características de controle de restrições de integridade e redundância de dados. Essa segurança com a manipulação dos dados é vista como parte essencial de sistemas que suportam ações de alteração, exclusão e atualização nos dados. Porém para um ambiente que se preocupa basicamente com duas ações, carga e acesso - como o DW -, a implementação de todas as características de um modelo relacional seria um investimento desnecessário (MACHADO, 2004).

Uma das vantagens da modelagem dimensional é sua simplicidade. Um projeto OLTP totalmente normalizado para um sistema de controle de pedidos, por exemplo, pode envolver dezenas de tabelas e tornar muito difícil o processo de obtenção de informações a partir delas. Isto se deve ao fato de que o modelo relacional foi desenvolvido para atender aos sistemas operacionais e a normalização evita redundâncias de tal forma que não haja preocupação com o sincronismo de dados nas operações de atualização, facilitando a manutenção da integridade desses dados nas tabelas (UNICAMP, 1998).

A figura 1 mostra as principais diferenças entre os sistemas tipicamente relacionais (bases de dados operacionais) e os sistemas dimensionais.

<b>Características</b>	<b>Base de dados Operacionais</b>	<b>Data Warehouse</b>
Objetivo	Operações diárias do negócio	Analisar o negócio
Uso	Operacional	Informativo
Tipo de processamento	OLTP	OLAP
Unidade de trabalho	Inclusão, alteração, exclusão	Carga e consulta
Número de utilizadores	Milhares	Centenas
Tipo de utilizador	Operadores	Comunidade gerencial
Interação do utilizador	Somente pré-definida	Pré-definida e ad-hoc
Condições dos dados	Dados operacionais	Dados analíticos
Volume	Megabytes-gigabytes	Gigabytes-terabytes
Histórico	60 a 90 dias	5 a 10 anos
Granularidade	Detalhados	Detalhado e resumidos
Redundância	Não ocorre	Ocorre
Estrutura	Estática	Variável
Manutenção desejada	Mínima	Constante
Acesso a registos	Dezenas	Milhares
Atualização	Contínua (tempo real)	Periódica (em batch)
Integridade	Transação	A cada atualização
Número de índices	Poucos/simples	Muitos/complexos
Intenção dos índices	Localizar um registo	Aperfeiçoar consultas

Figura 1 – Principais características entre sistemas operacionais e dimensionais

Fonte: Inmon e Kimbal (1997) apud Costa e Santos (2004).

Bancos de dados relacionais são sabidamente mais flexíveis quando são usados com uma estrutura de dados normalizada. Uma típica consulta OLAP (*On-Line Analytical Processing*), no entanto, "atravessa" diversas relações e requer diversas operações de junção para reunir estes dados. O desempenho dos sistemas de banco de dados relacionais tradicionais é melhor para consultas baseadas em chaves do que consultas baseadas em conteúdo, como acontece em DW.

## 1.2 Data Warehouse (DW)

A técnica de *Data Warehousing*, que consiste no processo de construção do DW, é considerada por muitos autores a evolução natural do Ambiente de Apoio à Decisão. Na bibliografia existem muitos conceitos, porém neste trabalho é considerado Inmon (1997) que conceitua DW como uma coleção de dados orientados por assunto, integrados, variáveis em relação ao tempo e não voláteis,

para dar suporte ao processo de tomada de decisão. Um DW tem a função de proporcionar uma integração histórica, sólida e concisa dos dados de uma organização, com respostas rápidas aos seus usuários. Em outras palavras, visa integrar e consolidar as informações de fontes internas, na maioria das vezes heterogêneas, e fontes externas, sumarizando, filtrando e limpando esses dados, preparando-os para análise e suporte à decisão. Provê um BD especializado, criado para armazenar estruturalmente dados vindos de diferentes fontes, e proporcionar aos usuários respostas com aspectos históricos.

De acordo com Domenico (2001) os dados mantidos por uma empresa são chamados de “operacionais” ou “primitivos”, referentes às informações necessárias para viabilizar operações diárias. Já os dados de DW são tidos como “analíticos”, destinados às necessidades de gerência no processo de tomada de decisão, podendo ser acessados em grande quantidade, se necessário, a fim de responder consultas complexas. Permitindo o armazenamento de informações históricas (de anos, meses, semanas...), o que implica em uma grande capacidade de processamento e armazenamento dos dados que se encontram detalhados ou resumidos. Existem várias razões que justificam a construção de um DW em uma organização, dentre as quais podem ser citadas :

- dificuldade acentuada na recuperação de dados históricos em períodos superiores ao ano atual de operações;
- falta de padronização e integração dos dados existentes nos diversos sistemas;
- o fato de que os tomadores de decisão são especialistas em atividades analíticas de negócios e não em computação;
- a necessidade de se tomar decisões de forma rápida, correta e clara, a partir de todos os dados disponíveis.

Neste contexto, é de extrema importância a consolidação dos conceitos de DW para que, entendendo suas características e objetivos, se determine as melhores alternativas em um projeto eficaz de DW.

### 2.2.1 Características do Data Warehouse

Segundo Inmon (1997), o DW pode ser caracterizado como segue: orientado por assunto, integrado, variável em relação ao tempo e não-volátil. Orientado por assunto: de maneira diferente dos sistemas que visam aplicações transacionais, o DW visa aplicações que buscam atender um determinado assunto, conforme a necessidade de um grupo específico de usuários. Esses assuntos são referentes a áreas estratégicas para as organizações. Por exemplo, em uma empresa, as vendas são foco de maior atenção dos gerenciadores, e um DW modelado para ela teria os dados das vendas como um assunto a ser focado.

*Integrado:* como os dados do DW são oriundos de diversas bases, locais ou não, estes normalmente não seguem uma padronização de estrutura. Cada projetista adota uma convenção de atribuição de nomes, unidades de medidas e outros aspectos, conforme sua necessidade. Por essa razão é que os dados devem passar por uma padronização, tornando-se assim integrados, livres de inconsistências e independentes de sua aplicação de origem. Por exemplo, um projetista pode entender que para a atribuição do campo “sexo”, a melhor forma é “F” como feminino e “M” como masculino, e outro pode atribuir este campo com “H” para homem e “M” para mulher, e assim em outras situações;

*Variável em relação ao tempo:* o DW é um ambiente projetado de forma que os dados não possam ser atualizados, mas o fato de não mudar não significa que não possa ser variável em relação ao tempo. Por exemplo, o valor total das notas fiscais de uma empresa às 12 horas pode ser diferente do encontrado no processo às 18 horas. Como se pode notar, o DW mantém o histórico dos seus dados por um período de tempo muito superior ao dos sistemas de BD tradicionalmente transacionais. Isto proporciona ao usuário, uma análise de comportamento e tendência dos assuntos da organização ou empresa;

*Não-volátil:* os dados são devidamente tratados ao serem carregados no DW: filtrados, limpos e transformados para satisfazer as necessidades de informação. Depois de carregado, o DW somente possui operações de consulta, e não tem necessidade de nenhum tipo de bloqueio por concorrência de usuários no acesso. À luz destes fatores, a redundância de dados entre a aplicação fonte e o DW raramente ocorre, resultando em menos de um por cento de duplicações.

Outra característica importante a ser considerada no ambiente de DW é quanto à granularidade dos dados. A granularidade de dados refere-se ao nível de sumarização dos elementos e de detalhes disponíveis nos dados (MACHADO 2004). Quanto mais detalhe existir, mais baixo será o nível de granularidade. Quanto menos detalhe, mais alto o nível será. A granularidade é uma característica relevante no projeto, devido ao fato de afetar diretamente o volume de dados que reside no DW, e ao mesmo tempo, afetar o tipo de consulta que pode ser atendida. Lembrando que quanto maior o volume de dados, menor é o desempenho de consultas mais detalhadas. Por essa razão, Inmon (1997) defende que a granularidade é a principal questão de projeto, pois o volume de dados contidos no DW é balanceado de acordo com o nível de uma consulta.

### 2.2.2 Arquiteturas de DW

Um dos objetivos do DW é a capacidade de prover respostas completas e rápidas aos seus usuários, com a máxima precisão dos resultados quanto ao tempo e clareza da resposta. Para isto, é preciso projetar uma arquitetura de acordo com a necessidade da apresentação final das informações, de modo que não falte detalhe e que, ao mesmo tempo, não prejudique a velocidade do processo de busca. Existem dois tipos de arquiteturas de DW : *top down* e *bottom up*. A diferença entre elas é com relação às visões que se pode ter do DW. O DW pode ser projetado com uma visão geral, em que a base provê informações de todos os negócios (assuntos) da organização (arquitetura *top-down*), ou pode ser projetado a partir de pequenas subdivisões (*data marts*), que juntas formam o DW da organização (arquitetura *bottom-up*) (KIMBALL, 1998).

*Data Mart* é um subconjunto lógico de um DW completo. Os *Data Marts* muitas vezes são vistos como uma alternativa ao uso de DW, pois levam menos tempo para serem desenvolvidos e implementados.

Uma perspectiva *top-down* considera que um DW completo e centralizado deve ser desenvolvido antes que partes dele, sumarizadas, possam ser derivadas na forma de *Data Marts*. Inversamente, uma perspectiva *bottom-up* considera que um DW possa ser composto a partir de *Data Marts* previamente desenvolvidos, segundo Campos (2002 apud PERNAS, 2002).

A implementação *top-down* é conhecida como padrão inicial do conceito de DW. A integração entre o DW e a os *Data Marts* é automática, desde que se mantenha uma disciplina na construção, tomando-se os *Data Marts* como subconjuntos do DW. Apesar desta forma de desenvolvimento ter como grande vantagem a integração total entre todos os assuntos da organização, e a concentração de todos os negócios da empresa, sendo possível extrair dele níveis menores de informações, esta visão generalista pode ser exatamente o seu maior complicador de projeto. Esta arquitetura apresenta alto grau de dificuldade no planejamento e implementação de um modelo único e mestre a toda organização (KIMBALL, 1998). Outra característica que é tida como desvantagem de utilização da arquitetura *top-down* é o tempo de implementação, pois os DW são, normalmente, desenvolvidos de modo iterativo, por áreas de assunto. Mesmo assim, seria preciso uma média de tempo não muito pequena para que a primeira área de assunto entrasse em produção. Na figura 2 é ilustrada a arquitetura do tipo *top-down* segundo Firestone (2000 apud DOMENICO,2001).

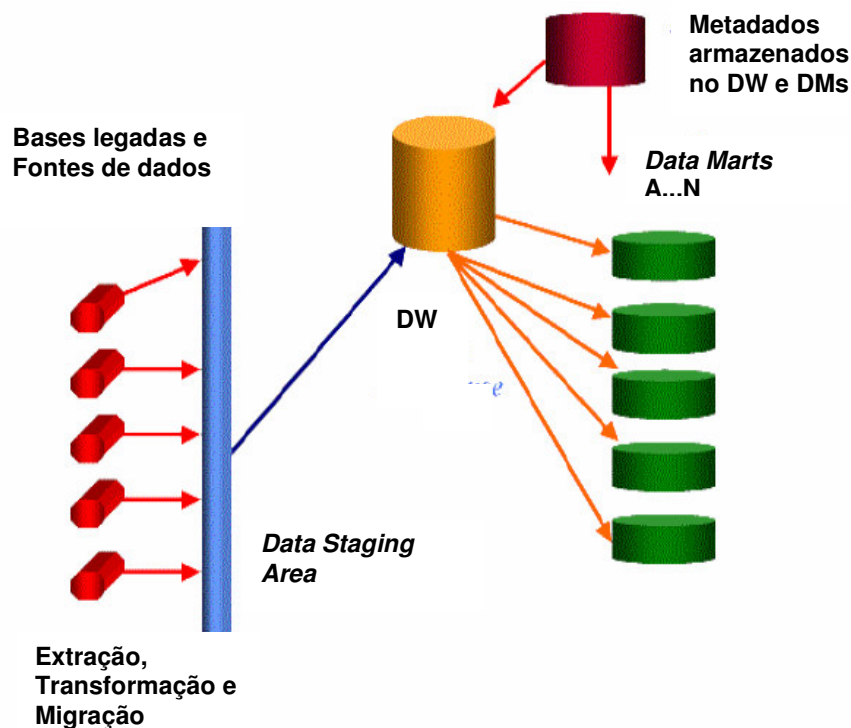


Figura 2 – Esquema da Arquitetura *top-down*

Fonte: Firestone (2000) apud Domenico (2001)



Pode ser observado tanto na abordagem *top-down* quanto *bottom-up* a utilização de uma etapa chamada *Data Staging Area*. O principal objetivo de uma *Area Staging* é criar um ambiente intermediário de armazenamento e processamento dos dados oriundos de aplicações OLTP e outras fontes, para o processo de extração e transformação e carga (ETL), possibilitando o seu tratamento, e permitindo sua posterior integração em formato e no tempo, evitando problemas após a criação do DW (MACHADO, 2004). O processo ETL será abordado na seção 2.2.6.

A arquitetura *bottom-up* é uma alternativa de construção e desenvolvimento rápido do DW. Esse tipo de implementação permite que o planejamento e o desenho dos *Data Marts* (DM) possam ser realizados sem esperar que seja definida uma infra-estrutura corporativa para DW na empresa. Não significa que essa infra-estrutura não existirá, mas sim que ela poderá ser implementada incrementalmente conforme forem sendo construídos os *Data Marts*. A idéia desta arquitetura é construir o DW de forma incremental, onde o processo inicia com a construção de um ou mais *Data Marts* independentes.

Um dos grandes problemas dessa implementação é a falta de um gerenciador que garanta padrões únicos de metadados, devido à independência dos *Data Marts*. A dificuldade em garantir essa padronização é responsável pela falha na elaboração incremental do DW. Pode ocorrer redundância de dados e inconsistências entre os *Data Marts*, que podem ser minimizadas por meio de um planejamento, monitoração e estabelecimento de regras de desenvolvimento (metodologias) (MACHADO, 2004). Na figura 3 é ilustrada a arquitetura *bottom-up*.

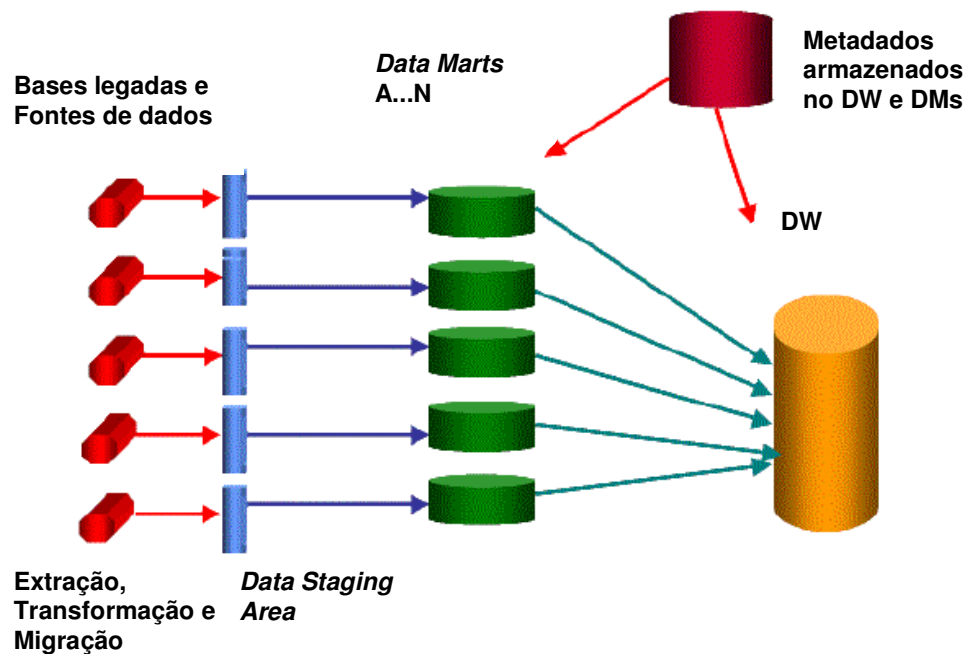


Figura 3 – Esquema da Arquitetura *bottom-up*

Fonte: Firestone (2000) apud Domenico (2001)

Alguns autores defendem a possibilidade de uma implementação híbrida, visando integrar a arquitetura *top-down* com a *bottom-up*. Nesta abordagem, primeiramente efetua-se a implementação de partes do DW e em seguida efetua-se a modelagem de dados do DW de visão macro. As partes são escolhidas por atividades ou processos relativos a assuntos de interesse, e constituem os *Data Marts*. A principal vantagem dessa abordagem é a garantia da consistência dos dados, já que o modelo de dados para os *Data Marts* é único, e possibilita realizar o mapeamento e o controle dos dados.

Características da implementação combinada, segundo Machado (2004):

- Implementação
  - Planejamento *Top Down*
    - Processos de Negócios
- Desenvolvimento *Bottom Up*
  - Um *Data Mart* de cada vez
  - Cada *Data Mart* é encarado de forma evolutiva
- Gestão de Metadados

A figura 4 ilustra a arquitetura combinada defendida por Machado (2004), onde se pode observar a modelagem de dados do DW em uma visão macro baseada em processos de negócios. Esta visão geral se utiliza do desenvolvimento de cada DM, sendo que estes podem ser gerados a partir do macromodelo de dados do DW também, constituindo assim uma forte coerência entre os vários DMs.

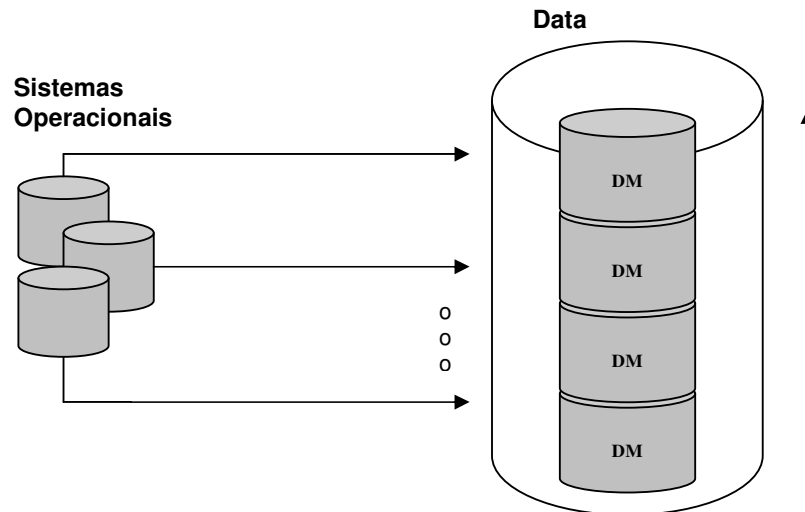


Figura 4 – Modelo de implementação combinada das arquiteturas

Fonte: Machado (2004)

A opção por um tipo de abordagem é influenciada por fatores como a infraestrutura de Tecnologia da Informação, o escopo de implementação, os recursos disponíveis, e principalmente pela necessidade ou não de acesso corporativo dos dados, assim como pelo retorno de investimento desejado e velocidade de implementação (MACHADO, 2004).

Outro aspecto importante quanto à arquitetura do DW é a existência dos metadados e a sua função no ambiente. Os metadados armazenam o significado de cada dado, isto é, eles são dados sobre dados. É fundamental que sejam previstos na aplicação, pois através deles o DW se torna mais produtivo, uma vez que proporciona, ao usuário final, navegação pelas possibilidades de manipulação dos dados. É importante que o DW possua uma variedade de metadados disponíveis, para que os usuários finais sejam capazes de acessar dados do DW sem a necessidade de saberem onde ou como os dados estão armazenados (INMON,

1997). Os metadados contêm, tipicamente, informações sobre a estrutura dos dados segundo a visão do programador e segundo a visão do analista do SAD, a fonte do dado e a transformação que o dado sofreu. Na seção 2.7.7 é apresentado um estudo mais detalhado sobre metadados.

### 2.2.3 Granularidade dos dados

A granularidade dos dados em um DW é um dos fatores mais importantes a ser analisados para a modelagem física dos dados, independente da arquitetura e implementação escolhida. A granularidade dos dados refere-se ao nível de sumarização dos elementos e de detalhe disponíveis nos dados. Quanto mais detalhe existir, mais baixo será o nível de granularidade e quanto menos detalhe existir, mais alto será o nível de granularidade, como pode ser observado na figura 5 (MACHADO, 2004) .

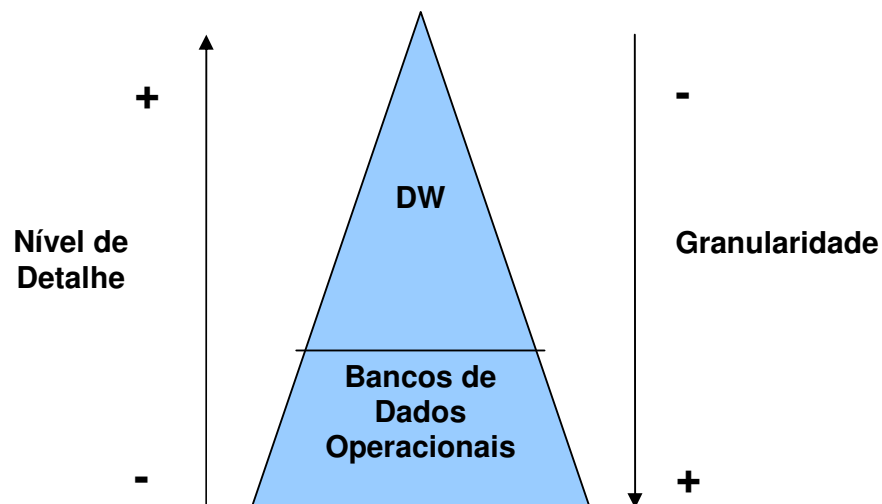


Figura 5 – Granularidade de dados

O nível de detalhamento dos dados é um ponto crítico porque isto afeta o volume de dados que reside no DW e ao mesmo tempo, afeta consideravelmente o tipo de consulta que pode ser executada. Portanto é importante que a informação armazenada no DW tenha um nível de detalhamento apropriado ao objetivo do usuário, ou seja, quanto mais dados a respeito da informação, mais consultas

podem ser realizadas. Por outro lado, quanto maior o volume de dados menor a performance do sistema (MACHADO , 2004).

Existem diferentes níveis de detalhe no DW. Uma das abordagens para o projeto de estrutura de dados em um DW foi proposta por Inmon (1997, apud DOMENICO, 2001), que contempla 3 níveis de armazenamento para os dados: dados de detalhe corrente, que seriam os dados de bases operacionais; dados levemente resumidos, armazenados por departamentos ou *Data Marts* e dados altamente resumidos, que seriam os dados individuais.

#### 2.2.4 Modelagem Multidimensional

A modelagem de dados para DW é diferente da utilizada para sistemas de transações diárias. Se uma organização que possui uma base de dados operacionais, projetada de forma relacional, tentar utilizar esta mesma base em um DW com uma simples inserção de dados históricos não obterá desempenho satisfatório nos resultados de busca de informações. Isto acontece por duas principais razões: a complexidade dos dados seria muito alta para os usuários realizarem as consultas e, conseqüentemente, a velocidade de respostas de apoio à decisão não seria a ideal para um sistema deste tipo.

Com o advento do DW, foi necessária a criação de uma técnica que suportasse um ambiente de análise multidimensional de dados, a técnica de modelagem multidimensional. Sendo assim, a modelagem Entidade – Relacionamento (ER), tipicamente utilizada em ambientes operacionais, não pode ser considerada uma opção para esta base.

O tempo gasto para consulta e apresentação dos dados são pontos críticos em um DW. A modelagem multidimensional consiste de uma técnica estruturada desenvolvida para a obtenção de modelos de dados de simples entendimento e alto desempenho, altamente otimizada para minimizar o tempo de consulta e apresentação (SCHLÖTTGEN, 2006).

Embora haja mais de um modelo para construir um DW com sucesso, este trabalho é baseado no modelo dimensional de dados estudado por Ralph Kimball (1998).

De maneira diferente do modelo relacional, o modelo dimensional é muito assimétrico. Existe uma grande tabela dominante no centro do esquema. Esta tabela

central é a única no esquema que se comunica com as outras tabelas, através de múltiplas junções. As outras tabelas se conectam à tabela central através de uma única junção. A tabela central é denominada tabela de fatos (*fact table*) e as outras são as tabelas de dimensões (*dimension tables*) (KIMBALL, 1996).

A tabela de fatos é a responsável pelo armazenamento das medidas numéricas do negócio. Cada uma dessas medidas é obtida da intersecção de todas as dimensões. Cada fato representa um item, uma transação ou um evento de negócio que é utilizado na análise de uma organização.

As tabelas dimensões são os elementos que participam de um fato, assunto ou negócio. As dimensões determinam um contexto de um assunto de negócios. Por exemplo um banco de dados que analisa as vendas de produtos: as dimensões que participam desse fato (vendas) geralmente são Tempo, Localização, Cliente, Vendedores. Dimensões não possuem valores numéricos, pois são descrições textuais dos elementos que participam de um fato. Uma visão prática e interessante, defendida por Machado (2004), é que as dimensões são possíveis formas de visualização dos dados, ou seja, são os “por” dos dados: “por mês”, “por país”, “por produto”, “por região”, etc.

O modelo dimensional também pode ser chamado de esquema estrela (*star schema*) devido à forma com que as tabelas ficam dispostas, uma tabela central de fatos ligada às dimensões, como pode ser notado na figura 6.

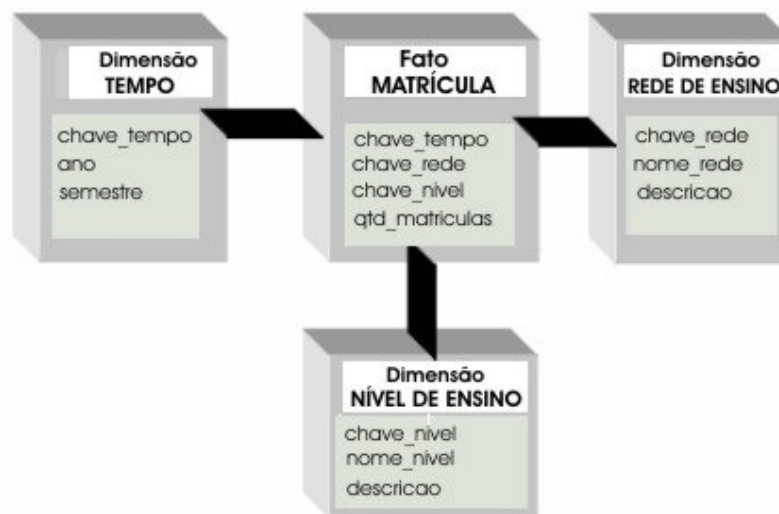


Figura 6 – Esquema estrela da tabela de Fatos Matrícula

Normalmente, a tabela de dimensão contém uma única chave primária (*primary key*) e vários atributos que descrevem a dimensão com detalhes. Enquanto que nas tabelas de fatos, a chave primária é composta pelas chaves primárias das tabelas de dimensão, ou seja, constituindo-se de várias chaves estrangeiras.

O esquema estrela pode sofrer variações, onde cada “ponta da estrela” passa a ser o centro de outras estrelas, constituindo assim um outro tipo de esquema chamado floco de neve (*snowflake schema*). Isto ocorre devido à necessidade de normalização das tabelas de dimensão, pela construção de hierarquias nas colunas das tabelas, como por exemplo uma dimensão Região que se divide em Estado, e Estado se divide em Cidade, como é ilustrado na figura 7.

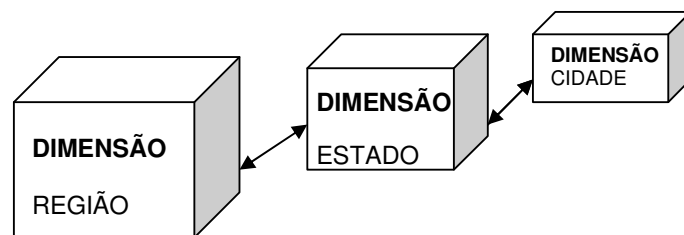


Figura 7 – Esquema floco de neve da Dimensão Região

No esquema floco de neve, cada um dos relacionamentos “muitos-para-muitos” é dividido em tabelas distintas. Pode-se dizer que o modelo floco de neve é o resultado da aplicação da terceira forma normal sobre as entidades dimensão. Os desenvolvedores freqüentemente elegem a utilização deste modelo pelo fato de preservar a utilização de meios de armazenamento.

De acordo com Kimball (1996), os projetistas devem resistir à tentação de transformar um modelo estrela em modelos floco de neve, devido ao impacto da complexidade deste tipo de estrutura sobre o usuário final. O ganho em termos de espaços de armazenamento também seria pouco relevante.

É importante lembrar que um DW não possui inclusão de dados por digitação. Exemplificando, não necessita de aplicação de normalização para garantir unicidade de valores textuais (campos de descrições) nem tampouco deve ser

preocupação do desenvolvedor a economia de espaço em meio magnético, e sim garantir a respeitabilidade do preceito de informação rápida (MACHADO, 2004).

### 2.2.5 O Processo de Desenvolvimento do Data Warehouse

Desenvolver um *Data Warehouse* é uma questão de aglutinar as necessidades dos seus usuários com a realidade dos dados disponíveis. Este processo consiste em extrair informações de BDs, na maioria das vezes distintos, integrar e armazenar estes dados de uma forma íntegra e possibilitar consultas rápidas e informações inteligentes sobre a base de dados gerada.

De acordo com Schlöttgen (2006), para um melhor esclarecimento do processo de *Data Warehousing*, pode-se dividi-lo em quatro grandes atividades:

- **Extração de dados:** realização da coleta dos dados mais relevantes da organização, e transformação destes dados a fim de tomarem uma única forma integrada. Grande parte do tempo de projeto do DW é despendida nesta atividade.
- **Armazenamento dos dados:** nesta etapa, os dados já limpos e devidamente tratados são armazenados em estruturas de um BD Multidimensional.
- **Consulta aos dados:** compreende todo o ambiente que o DW disponibiliza de acesso aos dados por parte do usuário, desde a base de dados até a interface amigável de pesquisa. É importante salientar que uma consulta no DW não se trata apenas de uma seleção de dados segundo alguns critérios, como em bases tradicionais. A consulta em um ambiente de DW realiza ações mais complexas. Além de varrer os registros da base de dados, ela executa um pré-processamento com cálculos matemáticos antes de entregar à interface de visualização as respostas.
- **Apresentação dos dados:** o foco principal desta atividade é proporcionar a melhor visualização de respostas aos usuários, de fácil compreensão e interpretação dos dados. Para isto, as simples estruturas de textos não são suficientes, sendo necessários recursos gráficos, planilhas para simulação e até mesmo animações. É essencial ao usuário que esta



etapa seja eficiente, pois uma simples alteração nos eixos de um gráfico pode fazer muita diferença para uma decisão.

Abaixo podemos analisar um conjunto de cinco etapas básicas no projeto de um *Data Warehouse* para que as quatro atividades anteriormente citadas sejam executadas com a máxima eficiência. Segundo a revista da unicamp (edição 04/05/98), as etapas são :

- construir um modelo lógico das informações disponíveis nas aplicações fontes;
- trabalhar com os gerentes e analistas de negócio para determinar qual o conjunto de informações que deve ser levado para o DW; este conjunto é o que será utilizado para a tomada de decisões;
- construir um modelo de transição que identifique dimensões e fatos;
- obter dos analistas de negócios que fornecem informações aos gerentes, o nível de agregação que deve ser utilizado, a frequência e periodicidade de cara de dados no DW;
- construir o projeto piloto e determinar a sua aderência aos requisitos exigidos.

A evolução de um DW é interativa por natureza. A medida que o DW é utilizado, são descobertas novas informações pelos analistas e gerentes de negócio. Esta utilização pode trazer impactos nos sistemas operacionais, porque pode-se precisar de dados que não constam no sistema, ou pode-se concluir que a qualidade dos dados não é boa.

DW não é um produto pronto que se compra, mas sim um projeto que envolve vários processos de análise e implementação, sendo que a observância de passos, citados anteriormente, pode aumentar a probabilidade de sucesso do projeto. Na figura 8 pode ser vista a seqüência de um processo, conforme proposto por Machado (2000 apud FAVARETTO, 2006).

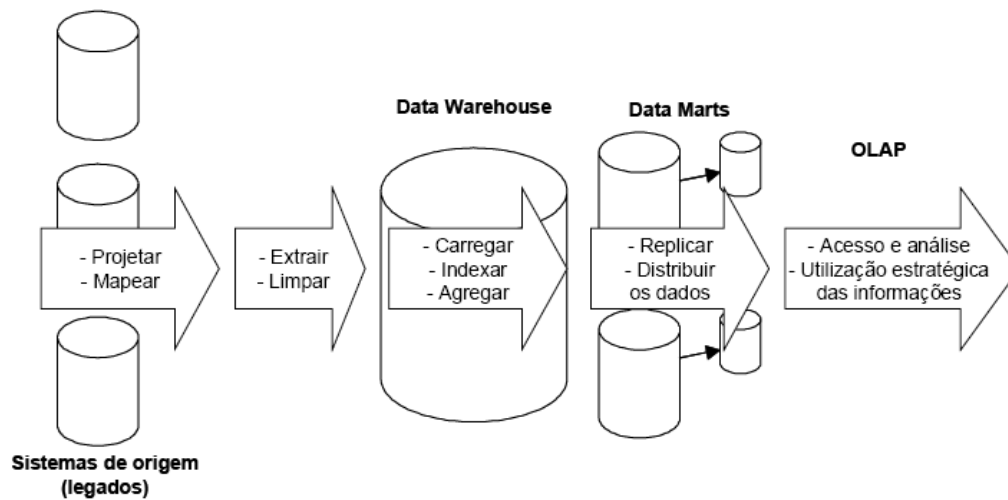


Figura 8 – O processo de *Data Warehousing*

Fonte: Machado (2000) apud Favaretto (2006)

### 2.2.6 Extração, transformação e carga dos dados

Esta etapa é uma das mais críticas de um DW, pois é a etapa que envolve a movimentação dos dados. A mesma se dá basicamente por três passos, conhecidos por ETL: Extração, Transformação e Carga dos dados. A fase ETL é caracterizada por um conjunto de rotinas e operações aplicadas às bases de dados dos sistemas que contêm os dados operacionais, para produzir os conteúdos do modelo dimensional. Este conjunto de operações pode ser construído manualmente ou utilizando-se ferramentas. O primeiro passo a ser tomado no processo de ETL é a definição das fontes de dados e sua extração. As fontes podem ser várias e de diferentes formatos, podendo-se encontrar desde sistemas transacionais de organizações ou empresas, até planilhas, arquivos textos e outros tipos. No segundo passo, o de transformação, são alteradas as características dos dados extraídos das bases operacionais, podendo incluir uma ou mais das seguintes operações: limpeza dos dados, integração dos tipos de dados das diversas fontes, alteração de códigos e seleção dos dados pertinentes. A operação de limpeza se torna necessária devido ao fato de que quando se obtém dados de fontes desconhecidas do projetista do DW, estes podem conter lixo em sua estrutura ou inconsistências, que podem comprometer a qualidade das consultas no DW. Um exemplo claro de lixo é quando na base operacional algum campo foi preenchido com valores irrealistas, apenas por

sua obrigatoriedade de preenchimento, como um número de telefone registrado como “99999999”, obtendo-se então um dado inválido dentro do sistema. Além da limpeza, na maioria das vezes, também é importante realizar uma transformação dos dados, pois como eles vêm de sistemas distintos, uma mesma informação pode ter formatos diferentes, e em um DW é preciso que os dados sigam uma padronização de apresentação dos dados, integrados em uma única forma. Outra razão para a transformação dos dados é o tratamento de valores que, ao serem carregados no DW, podem estar incompletos ou contendo valores que não podem ser manipulados corretamente. Então, é importante que nesta fase, sejam atribuídos valores padrão para dados perdidos ou corrompidos (CIELO, 2006).

É possível automatizar a transformação dos dados com ferramentas a partir de fontes heterogêneas (bases internas e externas), mapeamento da origem (*source data*) no dado destino (*target data*), criando as DDL (Data Definition Language) e gerando o código para transformação, manipulação e carga do dado na base destino (NAVARRO, 1996).

O terceiro passo a ser executado é a carga dos dados. Esta parte do processo possui uma grande complexidade, a qual se deve principalmente a heterogeneidade dos formatos dos dados nos sistemas “alimentadores” (UNICAMP, 1996). No momento da carga é necessário observar os campos que são chaves estrangeiras com suas respectivas tabelas para certificar-se de que os dados existentes na tabela da chave estrangeira estão de acordo com a tabela da chave primária e, assim, identificar se a tabela deve receber uma carga incremental ou uma carga por cima dos dados. A carga incremental é normalmente feita em tabelas fatos, e a carga por cima dos dados é feita em tabelas dimensões, onde o analista terá que excluir os dados existentes e incluí-los novamente (FUJIWARA, 2006).

De acordo com Inmon (1997), existem três tipos de carga que podem ser feitos do ambiente operacional para o DW:

- o carregamento de dados históricos, o qual, como regra, representa um desafio menor por não ser feito com frequência.
- o carregamento de dados de valor corrente no ambiente operacional, que também não constitui grande desafio por precisar ser feito apenas uma vez.

- o carregamento de alterações do DW a partir de atualizações que tenham ocorrido no ambiente operacional desde a última atualização do DW. Este constitui o maior desafio ao arquiteto de dados, pois o rastreamento eficiente e o tratamento dessas alterações não são tarefas fáceis de serem realizadas.

Para esta fase do projeto do DW, uma ferramenta de ETL tem grande valia, principalmente se os sistemas OLTP (transacionais) são muitos, pois elas são uma poderosa fonte de geração de metadados, que contribuirão muito para a produtividade da equipe de projeto. Apesar disto, é importante saber escolher a ferramenta. Minuciosidade e grande número de testes da ferramenta possibilitam maior chance de encontrar a mais adequada sendo que em alguns casos é interessante o auxílio de profissionais externos para a escolha. Com a utilização de uma ferramenta, a mais adequada possível, os benefícios são bastante vistosos e a produtividade pode aumentar consideravelmente.

### 2.2.7 Metadados

De acordo com Tronchin (1998 apud COME, 2006) a definição mais comum que se encontra na literatura sobre metadados é que eles representam "dados sobre dados". De uma forma um pouco mais completa pode-se dizer que o metadado é a "descrição do dado, do ambiente onde ele reside, como ele é manipulado e para onde é distribuído". Uma outra forma, mais concisa e direta, é definir metadado como "documentação".

Estes dados contêm informações sobre os dados que compõem a estrutura do DW. Eles não só descrevem o conteúdo como também fornecem ao usuário informações úteis para o julgamento da qualidade deste conteúdo. Mantém ainda informações sobre o modelo de dados, especificação dos arquivos (chaves e atributos), histórico de extrações, controle de acesso, entre outras informações. Os metadados podem ser vistos como dados de mais alto nível que descrevem dados em um nível inferior. Pela importância que o metadado tem no entendimento dos dados, para qualquer projeto de DW, é pertinente que se planeje a sua criação tendo em vista o grupo de usuários a que se destina. Por esta especificidade de implementação, metadados podem ser classificados em dois tipos: os metadados

técnicos, usados pelos desenvolvedores e analistas, e os metadados de negócio, usados pelos executivos e analistas de negócios (DOMENICO, 2001).

Os metadados técnicos fornecem aos desenvolvedores e administradores de bancos de dados, a confiança de que os dados estão corretos, através de descrições técnicas dos dados e suas operações. Este tipo é importante para a manutenção e o crescimento contínuo do DW. Especificam nomes de atributos, tipos de dados, fontes a partir dos quais os dados são extraídos, regras de transformação, etc. Os metadados de negócio são o elo entre os usuários de negócios e o DW. Eles mostram quais relatórios, consultas e dados estão no DW, a localização destes dados, confiabilidade, contexto, regras de transformação que foram aplicadas e as origens destes dados, podendo também especificar o cálculo de um valor, o cálculo de custo de um produto, etc. (MACHADO, 2004).

A figura 9 ilustra os tipos de metadados.

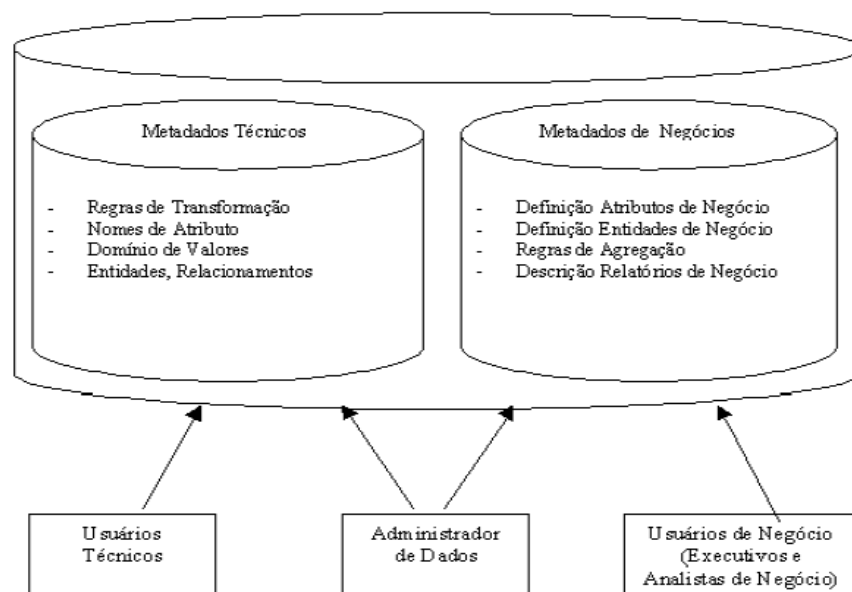


Figura 9 – Tipos de metadados

Fonte: David (1999) apud Domenico (2001)

Eles proporcionam maior credibilidade às informações recuperadas através do DW, uma vez que, informando ao usuário a origem, a forma inicial e alterações que o dado sofreu, a informação fica mais completa e permite uma geração efetiva de conhecimento e vantagens competitivas.

Os metadados podem surgir de vários locais no decorrer do projeto. Desde o material originado das entrevistas com os usuários, até a documentação dos sistemas operacionais.

Todo o elemento de dado precisa ter sua origem ou o processo que o gera identificado. Esta informação é importante para o caso da necessidade do usuário saber a fonte geradora do dado, que deve ser única.

Existem dois tipos de fontes de dados: formal e informal. Estas fontes são usadas tanto para os metadados técnicos, como para os metadados de negócios (DAVID, 1999 apud DOMENICO, 2001).

As fontes formais são aquelas em que os dados foram amplamente discutidos pelos tomadores de decisão da organização, documentados e formalizados. Geralmente são armazenados em ferramentas ou documentos que são mantidos e distribuídos dentro da corporação. Podem fazer parte dos metadados formais tanto os técnicos quanto os de negócios.

Metadados informais são os que estão na consciência dos profissionais da companhia, não tendo nenhum tipo de documentação a respeito. Quando estes metadados são relevantes ao processo, devem ser incluídos no projeto do DW e devidamente documentados, transformando-os imediatamente em metadados formais.

Abaixo são listadas algumas fontes de metadados, segundo Machado (2004):

- repositórios de ferramentas CASE
- códigos-fonte de sistemas
- objetos de banco de dados
- entrevistas com especialistas da empresa que entendam do negócio estratégico
- programas construídos para carga de dados no DW.

Devido à grande diversidade de fontes de metadados, é desejável que a sua extração ocorra automaticamente, assim como também o acompanhamento de suas alterações através de um repositório. Apesar de todas as vantagens e funcionalidades dos metadados, a tarefa de obtê-los e mantê-los eficientemente não

pode ser considerada uma tarefa de fácil solução. Alguns problemas são identificados na criação dos metadados, como a carência de bons metadados, o que muitas vezes limita a utilização absoluta dos recursos dos dados, e leva à criação de novos dados discrepantes, os quais não são adequadamente documentados, criando assim uma tendência de um ciclo de criação de dados discrepantes na organização (DOMENICO, 2001).

A qualidade dos dados obtidos no DW representa um dos maiores riscos que devem ser gerenciados quando são projetados os metadados.

### 2.2.8 Ferramentas de Data Warehouse

Como visto na seção 2.2.6, normalmente faz-se necessário o desenvolvimento de sistemas ou avaliação de ferramentas para extração de dados e atualização do DW. Esses sistemas ou aplicações são responsáveis pela filtragem, limpeza, sumarização e concentração dos dados espalhados pelas fontes externas e nos sistemas operacionais. A maioria das ferramentas possui interfaces gráficas e interativas que facilitam o mapeamento dos dados e a automação do processo ETL, fortemente baseadas no conhecimento de linguagem SQL (*Structured Query Language*), possuindo funções pré-determinadas para sua utilização.

No DW, as ferramentas devem permitir um acesso intuitivo aos dados, possibilitando a análise daqueles mais significativos. Elas não podem ser muito complexas porque não serão utilizadas por profissionais da área técnica, mas precisam ser robustas o suficiente para dar agilidade no acesso às informações estratégicas.

Existem várias maneiras de recuperar informações de um DW. As ferramentas mais utilizadas são os relatórios, as consultas, os Sistemas de Informação Executivo (SIE), OLAP (*On-Line Analytic Processing*) e Mineração de Dados (*Data Mining*). Algumas características destas ferramentas estão descritas na figura 10:

Tipo de ferramenta	Questão básica	Exemplo de resposta	Usuário típico e suas necessidades
Pesquisa e Relatórios	"O que aconteceu?"	Relatórios mensais de vendas, histórico do inventário.	Dados históricos, habilidade técnica limitada.
OLAP	"O que aconteceu e por quê?"	Vendas mensais versus mudança de preço dos competidores.	Visões estáticas da informação para uma visão multidimensional; tecnicamente astuto.
SIE	"O que eu preciso saber agora?"	Memorandos, centros de comando.	Informações de alto nível ou resumidas; pode não ser tecnicamente astuto.
<i>Data Mining</i>	"O que é interessante?" "O que pode acontecer?"	Modelos de previsão.	Tendências e relações obscuras entre os dados; tecnicamente astuto.

Figura 10 – Ferramentas de acesso aos dados

Fonte: Revista Byte Brasil (1997) apud Campos (2006)

Analisando-se a figura 10, identifica-se a utilização de ferramentas do tipo OLAP e Mineração de dados para o suporte a perguntas mais estruturadas e complexas. De acordo com Machado (2004) a ferramenta de Mineração de dados, *Data Mining*, é uma ferramenta de análise denominada *open-end*. Permite ao usuário avaliar tendências e padrões não conhecidos entre os dados. Esse tipo de ferramenta utiliza-se das mais modernas técnicas de computação, como redes neurais, algoritmos genéticos e lógica nebulosa. As ferramentas OLAP serão abordadas na seção seguinte.

#### 2.2.8.1 Ferramentas OLAP

A ferramenta "*On-line Analytical Processing*" – OLAP é o conjunto de tecnologias que possibilita explorar os dados de um DW, extraíndo-os de suas bases e construindo relatórios capazes de responder às questões gerenciais. O uso das ferramentas OLAP permite aos usuários analisar os dados em dimensões múltiplas de um empresa, como por exemplo região, produto, tempo e vendedor. Cada dimensão pode conter hierarquias, por exemplo, a dimensão tempo pode conter a hierarquia ano, trimestre, mês, semana ou dia. Os dados nessas dimensões são agregados, ou seja, são resumidos, mas pode-se navegar livremente de uma



hierarquia para outra, até chegar-se na máxima granularidade dos dados (BISPO, 1998).

Figueiredo (1998 apud BISPO, 1998) defende que a principal característica das ferramentas OLAP é permitir uma visão conceitual multidimensional dos dados de uma organização, com a apresentação dos dados em termos de dimensões. Sendo esta, uma visão muito mais útil para os usuários tomadores de decisão, se comparada à tradicional visão baseada em tabelas (modelo entidade-relacionamento) utilizada nos sistemas OLTP.

Considerando que o interesse de usuários de sistemas OLTP tipicamente é criar, atualizar e recuperar informações sobre registros individuais e que o interesse de usuários de OLAP é analisar globalmente os dados com diversas visões, com relação a aspectos temporais e desempenho de alguma função da organização, é que se pode identificar as diferenças funcionais entre sistemas OLTP e OLAP. Obter resposta a uma consulta mais complexa utilizando ferramentas relacionais padrão, não fornecerá a solução requerida. Como se pode observar, as aplicações OLAP diferem das aplicações OLTP no que se refere aos requisitos funcionais e de desempenho, conforme mostra a figura 11.

<b>CARACTERÍSTICAS</b>	<b>OLTP</b>	<b>OLAP</b>
Operação Típica	Atualização	Análise
Telas	Imutável	Definida pelo usuário
Nível de Dados	Atomizado	Altamente sumarizado
Idade dos Dados	Presente	Histórico, atual e projetado
Recuperação	Poucos registros	Muitos registros
Orientação	Registro	Arrays
Modelagem	Processo	Assunto

Figura 11 – Principais diferenças entre OLAP e OLTP

Fonte: Anzanello (2006)

As ferramentas OLAP permitem ao usuário analisar o porquê dos resultados obtidos. Atualmente existe uma variedade dessas ferramentas com diferentes tipos de abordagens, cujas arquiteturas dependem do método que é utilizado para o armazenamento dos dados. Os métodos de armazenamento de dados são: MOLAP, ROLAP, DOLAP e HOLAP. Cada um deles tem uma função específica, e deve ser utilizado quando melhor atender às necessidades de análise pela ferramenta OLAP.

- MOLAP (OLAP Multidimensional): os dados são armazenados de forma multidimensional e acessados por meio de cubos e hipercubos.
- ROLAP (OLAP Relacional): os dados são armazenados em bancos relacionais e suas consultas são processadas pelo gerenciador do banco relacional.
- DOLAP (OLAP Desktop): é uma variação que existe para fornecer portabilidade dos dados, geralmente voltadas para computadores pessoais. Este tipo de ferramenta vem sendo mais empregado nos bancos de dados individuais para análises mais específicas. A vantagem que oferece é a redução do tráfego na rede.
- HOLAP (OLAP Híbrida): é a arquitetura mais recente, na qual ocorre uma combinação entre ROLAP e MOLAP. Este tipo oferece a vantagem de extrair o que há de melhor em cada uma: o alto desempenho do MOLAP e a escalabilidade do ROLAP e misturá-las em uma única tecnologia (MACHADO, 2004).

De acordo com Colliat (1996), ferramentas OLAP permitem analistas, administradores e executivos obter detalhadas informações de uma organização, através do interativo e rápido acesso à variedade de visões dos dados. Estes são organizados por aspectos dimensionais, e para possibilitar estas funções, alguns requisitos fundamentais são necessários nas ferramenta deste tipo. São eles:

- o nível básico de dados é o de dados sumarizados (total de vendas de um produto em uma região em um determinado período);
- dados históricos, correntes e projetados;
- agregação de dados e possibilidade de navegação de forma interativa com os vários níveis de agregação (*drill down*);
- dados derivados, computados a partir de dados de entrada (variação de valores, calculados, dentro de um determinado período);
- visões multidimensionais dos dados (vendas por produto, por região, por período);
- consultas rápidas e interativas (resposta em segundos);
- de médio a alto grau de armazenamento (de 1 a 500 Gigabytes);

Além de todas estas características das ferramentas OLAP, também precisam ser consideradas as operações básicas, assim como para um modelo ER que tem as suas operações básicas de álgebra relacional. Quatro tipos de operações são utilizadas em OLAP para analisar os dados, de acordo com Machado (2004).

As operações *drill*, que utilizam a navegação nos dados, modificando o nível de granularidade de consulta. *Drill-down* ocorre quando o usuário aumenta o nível de detalhe da informação, diminuindo o nível de granularidade, para solicitar uma visão mais detalhada em um conjunto de dados. Já a *drill-up*, ou *roll-up*, é o oposto de *drill-down*, e ocorre quando o usuário aumenta o nível de granularidade, diminuindo o nível de detalhamento da informação. Com a capacidade de *drill-down*, o analista pode navegar do mais alto nível ao menor de detalhamento. Já com a capacidade de *drill-up*, o analista pode navegar do maior nível de detalhamento ao mais alto nível de sumarização dos dados. A figura 12 ilustra o efeito das ações *drill-down* e *drill-up*, utilizando o exemplo de um Estado em um nível mais alto de abstração e com suas respectivas cidades em um nível menor de abstração, ou seja, maior detalhamento.

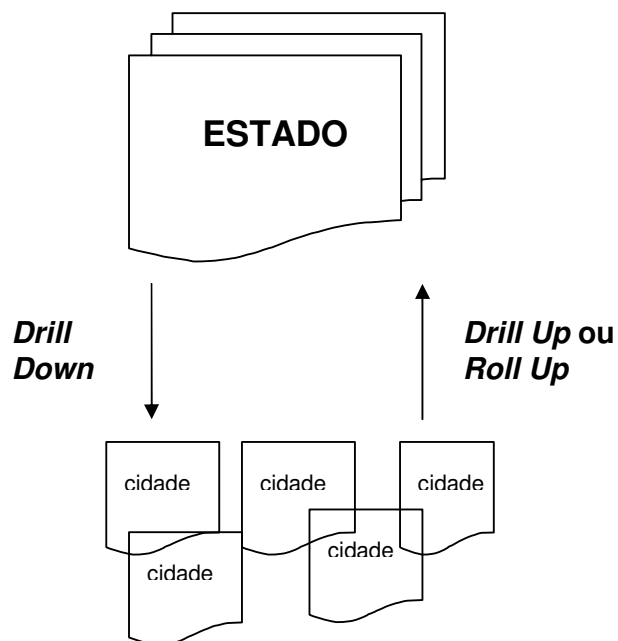


Figura 12 – Operações *Drill Down* e *Drill Up* ou *Roll Up*

Também existe a operação *drill-across*, que ocorre quando o usuário pula um nível intermediário dentro de uma mesma dimensão. Por exemplo, a dimensão tempo é composta por ano, semestre, trimestre, mês e dia. Se o usuário passar de ano direto para mês e dia a operação então ele está executando um *drill across*.

A última operação *drill* é a *drill-through*, que é quando o usuário passa de uma informação contida em uma dimensão para uma outra. Por exemplo, o usuário está realizando uma consulta por tempo e no próximo passo começa a analisar por região (MACHADO, 2004).

Para navegar nas dimensões, utilizam-se as operações de *slice and dice*. Significa uma descrição padrão para a habilidade de acessar os dados do DW através de qualquer uma das dimensões de forma igual. Ela permite a análise de informações de diferentes prismas. *Slice and dice* é o mesmo que filtrar. Enquanto que *slice* corta o cubo, mantendo a mesma perspectiva de visualização de dados, *dice* muda esta perspectiva de visão inicial, podendo ser interpretada como a extração de um “subcubo” ou a intersecção de vários *slices*. Esta operação pode ser vista na figura 13.

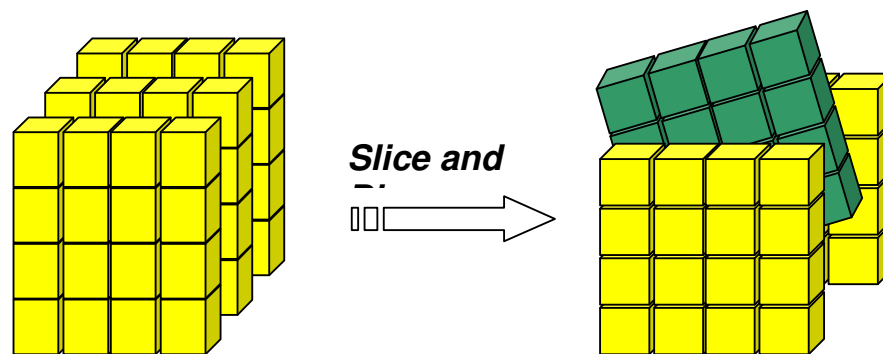


Figura 13 – Operação *Slice and Dice*

As operações OLAP podem ser utilizadas combinadas, ou seja, realizar um *slice* e um *dice* ao mesmo tempo junto com operações *drill-down*, ou *drill-up*.

Outras operações são a *pivoting* (pivoteamento) e tabelas cruzadas. Pivoteamento é o ângulo pelo qual os dados são vistos ou trocados, ou a mudança do arranjo de linhas e colunas em um relatório tabular (inversão dos eixos das

dimensões para obter-se novas visões de consultas). A figura 14 ilustra a operação *pivoting*.

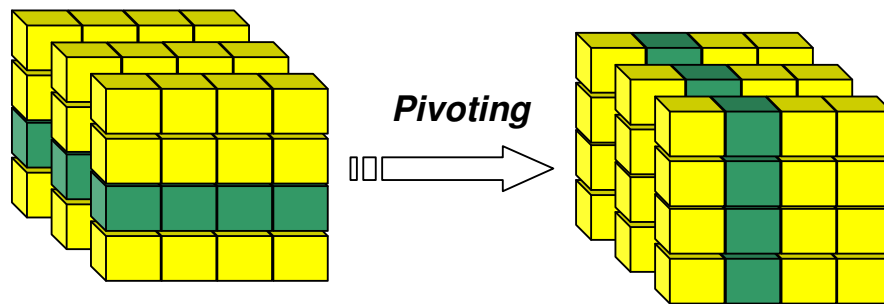


Figura 14 – Operação *Pivoting*

Tabelas cruzadas são as tradicionais tabelas eletrônicas, e se diferenciam pelo fato de que os dados são apresentados em planilhas com mais de duas dimensões, normalmente quatro ou mais (PERNAS, 2003).

#### 2.2.8.2 Ferramentas de Projeto de DW

Como já foi dito na seção 2.2.5, o Projeto de DW possui várias etapas, entre elas a modelagem da base de dados a partir das quais os usuários buscarão informações. As etapas de carga de dados, transformação e outras, já foram citadas anteriormente em outras seções, e a etapa de modelagem dos dados será abordada por esta seção.

Além das ferramentas necessárias para a execução das funções durante a utilização de um DW, é importante também a utilização de ferramentas adequadas ao projeto do DW, principalmente na etapa de modelagem da base de dados, a modelagem conceitual.

Uma poderosa ferramenta para o planejamento de banco de dados de fácil utilização, é a ferramenta CASE (*Computer Aided Software Engineering*) *Allfusion Erwin Data Modeler* da empresa *Computer Associates* ([www.ca.com](http://www.ca.com)), versão 4.0. Ela confere grande produtividade ao planejamento, geração e manutenção de

aplicativos de bancos de dados de qualidade (MACHADO, 2004). A Figura 15 mostra uma tela do Erwin:

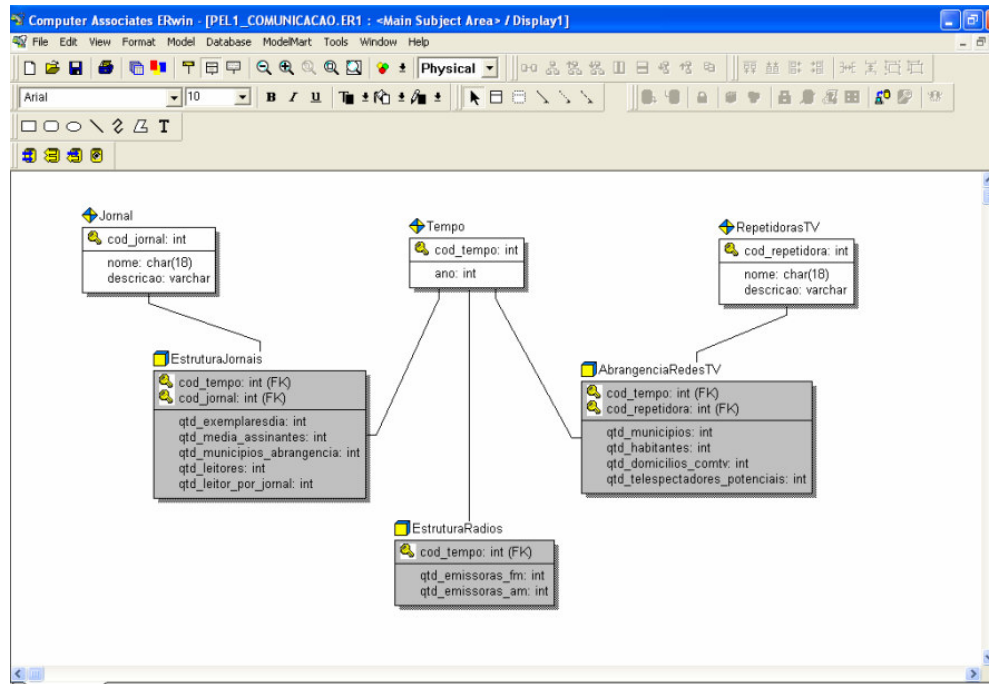


Figura 15 – Tela do Allfusion ERwin

O Erwin facilita a criação de um projeto de banco de dados com um modelo gráfico ER (Entidades-Relacionamento) de todos os requisitos de dados, capturando as regras de negócio em um modelo lógico, exibindo todas as entidades, atributos, relações e grupos principais.

Mais que uma ferramenta de “desenho”, o Erwin automatiza o processo do projeto de forma inteligente. Por exemplo, o Erwin permite a criação de um dicionário de atributos reutilizáveis, garantindo nomes e definições uniformes em todo o projeto. Também permite o controle de modelos de DW corporativos, dividindo-os em subconjuntos menores, mais fáceis de gerenciar. Isso permite que, modelando *Data Marts* individualmente, o analista concentre-se em áreas ou assuntos específicos.

O Erwin suporta as técnicas de modelagem multidimensional esquema estrela (*star schema*), e floco de neve (*snowflake*), comentadas na seção 2.2.4, permitindo o projeto de todos os sistemas transacionais, *Data Marts* e DWs em um

ambiente integrado. Utilizando as regras de modelagem multidimensional do Erwin para projetar tabelas de fato e dimensão, o modelo captura e documenta um vasto conjunto de informações sobre o *Warehouse*, incluindo fontes de dados, lógica de transformação e regras de gerenciamento de dados. Esse produto de modelagem pode ajudar e simplificar o projeto de DW. Esta ferramenta é um software proprietário, portanto exige custos para a sua utilização ou é possível obter uma versão para testes a partir do site da empresa [www.ca.com](http://www.ca.com).

### 2.2.9 Ambiente de Data Webhouse

A proposta deste trabalho é que o DW seja modelado para a utilização via *Internet*, se tornando não somente a modelagem de um DW, mas de um *Data Webhouse*.

Segundo Kimball (2000), nos anos 90, DWs tiveram muito sucesso, mas um número considerável de falhas foram detectadas. Estas falhas ocorreram principalmente por:

- não responderem aos requisitos urgentes de negócio da empresa;
- não construírem um sistema que poderia ser utilizado pelos usuários finais;
- embarcarem em projetos extremamente centralizados, caros, ambiciosos e que não poderiam ser concluídos.

A chegada da *web* foi extremamente benéfica para o movimento de DW. As facilidades que a *web* proporciona, oferecem muitos benefícios que a indústria do DW não poderia construir, tratando-se então de uma nova perspectiva do DW.

#### 2.2.9.1 Data Warehouse na web

No contexto de divulgação e movimentação que o DW busca, é praticamente impossível não se pensar na utilização da *internet* com todas suas vantagens. Ela permite que qualquer pessoa tenha acesso imediato, de qualquer lugar do mundo, sobre os mais variados assuntos a que o DW se propõe.

Para a abrangência que o DW idealmente deve alcançar, a *web* vem facilitar de forma considerável para a divulgação e a movimentação dos dados. Por exemplo, no caso de uma empresa que tem seus dados organizados em um DW, a divulgação pela *web* é essencial pelo fato de seus clientes, em grande parte, estarem conectados à *web*, podendo, a qualquer hora e de qualquer localidade, interagirem com o DW, mais especificamente, *Data Webhouse* (PERNAS, 2003).

Kimball (2003) prevê algumas expectativas por parte do usuário do *Data Webhouse*, e que podem justificar o uso da internet para a divulgação do DW:

- a *web* é fácil de ser utilizada. Qualquer pessoa pode pressionar botões e preencher pequenos formulários.
- espera-se que a *web* seja rápida – não importa que, às vezes, ela não o seja. Essa expectativa é inegociável.
- espera-se que os sites de *web* estejam funcionando 24 horas por dia, 7 dias por semana.
- pode-se encontrar o *status* de quase todas as coisas na *web*, e espera-se que tal *status* esteja atualizado até o último minuto.
- todos os tipos de coisas surpreendentes acontecem na tela. As interfaces de *web* são intensamente gráficas e dinâmicas atualmente.
- a capacidade de comunicar-se com alguém é colocada em quase todas as páginas *on-line*.
- uma página da *web* deve ser personalizada para o usuário, de fácil utilização.
- o que o usuário quer é um portal de informações corporativas.
- a segurança é uma enorme e relevante questão que deve ser resolvida antes de o usuário poder confiar na *web*.

Para que o *Data Webhouse* atenda a todas essas expectativas, é necessário um projeto voltado a atender questões relativas à *web*, como segurança e interface do site além das resoluções com relação à modelagem do DW. Supondo uma adequação do DW para o *site*, a utilização da internet agrega valor à utilidade do DW devido sua abrangência e flexibilidade.



## **ESTUDO DE CASO: PREFEITURA MUNICIPAL DE PELOTAS**

### **3.1 O Banco de Dados da Prefeitura Municipal de Pelotas**

O tratamento estratégico de informações, hoje em dia, pode ser visto como uma prática poderosa no mundo dos negócios. No entanto, além de conhecer as informações, é preciso saber a melhor forma de obtê-las, tratá-las e disponibilizá-las. Ter acesso ao máximo possível de informações não determina que se deva ter um grande aglomerado de dados.

Tendo em vista esta necessidade, a Coordenadoria para Desenvolvimento Local e Regional, implantada no ano de 2005 pela Prefeitura de Pelotas, empreendeu esforços propondo debates com a comunidade no sentido de:

- a) adequadamente identificar as causas da perda de importância e de dinamismo da economia local e regional, bem como as potencialidades existentes e os entraves a superar para que a situação fosse revertida;
- b) reunir dados capazes de traçar um perfil realista e abrangente da realidade atual, que permitissem quantificar e qualificar as diferentes dimensões da vida comunitária, com vistas a corretamente subsidiar as ações a serem encetadas;
- c) organizar estes dados em forma de apresentação capaz de estimular investidores internos e externos;
- d) examinar, discutir e priorizar estratégias capazes de possibilitar a desejada aceleração no processo de desenvolvimento.

Através destas atividades realizadas, a reunião destas informações deveria ter uma apresentação real e ao mesmo tempo otimista da cidade de Pelotas, para que, através da amostragem dos dados aos investidores, estes se sentissem motivados a instalar seus empreendimentos na região.

Através da reunião dos dados, a equipe da Prefeitura construiu um Banco de Dados, organizando-os em formato de slides, como mostra a figura 16:

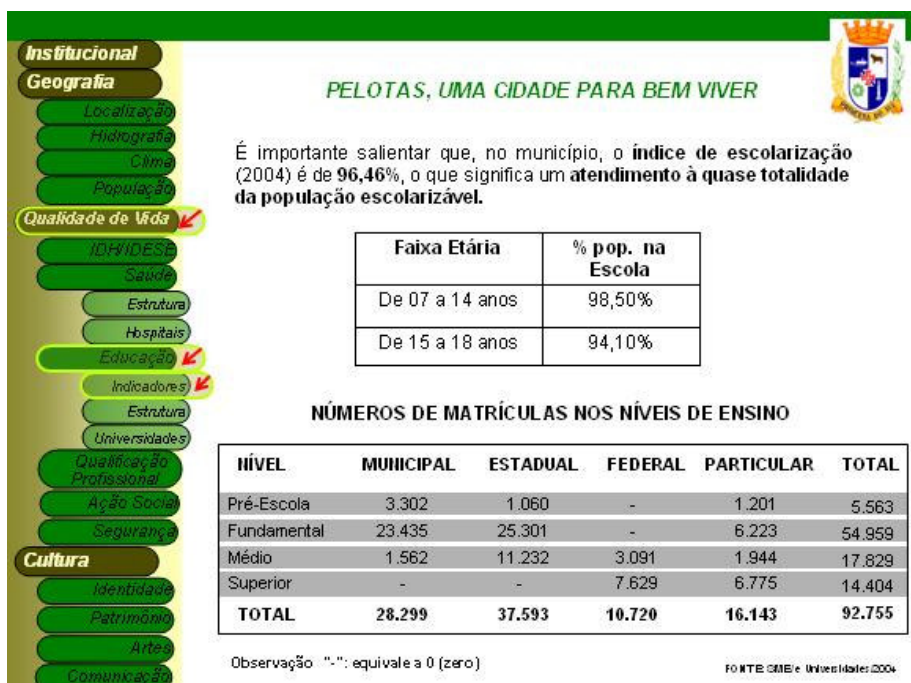


Figura 16 – Tela do Banco de Dados da Prefeitura

Os slides são compostos por um menu não-interativo, com a divisão dos segmentos da sociedade, mostrando à esquerda os dados relativos ao segmento escolhido. Os dados não contêm um padrão de armazenamento e estão dispostos em vários tipos de visualização como em tabelas, imagens, textos e gráficos.

### 3.1.1 Coleta dos dados

Os dados foram coletados de fontes consideradas fidedignas pela administração da cidade, sendo que alguns foram coletados de bases automatizadas. Outros foram obtidos através de relatórios em forma de texto ou planilhas no formato Excel, software proprietário da Microsoft. Também foi utilizado material multimídia para a reunião dos dados.

A Prefeitura realizou extensiva pesquisa nos diversos órgãos e entidades públicas e privadas com atuação no município, a fim de reunir o máximo de informação possível, nas seguintes fontes:

- levantamentos realizados pelo IBGE (2004), pelo ITEPA/UCPEL (2003-2005) e outros estudos e pesquisas que continham dados sobre o município de Pelotas e Zona Sul também foram considerados;
- análises resumidas elaboradas pela FIERGS sobre indicadores econômicos e sociais do RS e de suas diversas regiões;
- relatórios desenvolvidos por consultoria contratada pela Secretaria de Coordenação do RS para desenvolver ações que combatam as desigualdades regionais do Estado
- material promocional elaborado pelo Governo do Estado (SEDAI) para atração de investimentos e outros materiais deste tipo.

### **3.2 Definição do Problema**

Apesar do material elaborado pela equipe da Prefeitura ser muito rico em dados, e possibilitar o acesso a informações completas projetando assim um perfil realista da região, apresenta sérios problemas, sendo eles:

- a) armazenamento: os dados estão dispostos em forma de apresentação de slides, construídos no programa Power Point, proprietário da empresa Microsoft ([www.microsoft.com.br](http://www.microsoft.com.br)), constituindo no total 3 arquivos, um independente do outro. Possuem muitos gráficos e imagens, exigindo assim uma considerável capacidade dos computadores para visualizá-los;
- b) portabilidade: o tamanho dos arquivos que contêm os dados limita a portabilidade por serem consideravelmente grandes.
- c) disseminação da informação: devido a portabilidade limitada dos arquivos, a divulgação das informações fica prejudicada, o que leva a uma dificuldade de acesso ao material por parte dos investidores. Em certos casos, este problema faz com que o material não seja nem mesmo analisado.
- d) consulta aos dados: caso um investidor queira tomar conhecimento sobre dados de determinada área, gasta um tempo relativamente grande para encontrar a informação referente à sua pesquisa. Como o material em slides provido pela Prefeitura Municipal de Pelotas consiste de 3 arquivos

independentes, é necessário que o investidor saiba em qual destes está a informação buscada. Em seguida, exige-se a navegação por todo arquivo de apresentação, para se chegar à informação questionada, sendo que esta ainda pode estar inserida em um texto, exigindo assim mais o tempo de leitura para a descoberta do dado. Todo este tempo gasto na procura faz com que a consulta ao material não seja uma prática dinâmica e rápida.

- e) estrutura dos dados: o material é composto por uma quantidade muito grande de dados, inclusive por aqueles que não necessitam estar armazenados. Alguns deles podem ser automaticamente calculados se estivessem em um sistema computacional, economizando assim espaço de armazenamento e proporcionando mais clareza à informação.

Conforme estudado no capítulo 2, um sistema de informação deve proporcionar a comunicação e o poder de análise de que as empresas necessitam para administrar negócios em escala global e que, além disso, o sistema deve se utilizar de um Banco de Dados adequado à especificidade de cada organização, para proporcionar estas características e oferecer um melhor mecanismo de busca, tratamento e disponibilidade das informações.

### **3.3 Alternativa de solução**

Neste trabalho procurou-se estudar qual a melhor forma de disponibilizar os dados relativos aos segmentos da sociedade em um DW, tendo em vista a otimização da análise de grandes volumes de dados com aspecto histórico que este tipo de sistema oferece, para que venha a contribuir com a qualidade do material de apresentação elaborado pela equipe da Prefeitura e assim erradicar os problemas identificados anteriormente. Além deste enfoque, foi considerada a problemática de que os usuários deste sistema podem estar localizados tanto na cidade de Pelotas quanto fora dela, portanto buscou-se uma forma de modelar os dados buscando otimização no tempo de acesso, disponibilizando-os na *Internet*, na forma de um *Data Webhouse*.

A partir do entendimento do problema a ser solucionado, é possível definir que objetiva-se, através de estudos e análises sobre as técnicas de BD, a criação de

um modelo de BD que atenda os requisitos de informação da Prefeitura Municipal de Pelotas, a fim de implementar um Sistema de Informação em ambiente *web*. Para tanto, é preciso definir metodologias mais adequadas ao tratamento destes aglomerados de dados, que são incapazes de oferecer informações interligadas e relevantes ao sistema. Objetiva-se que, com a modelagem apresentada por este trabalho, seja facilitada a tomada de decisão por parte dos investidores, tanto no fator de disponibilidade de dados sobre a região quanto no fator de facilidade de acesso destes pela *Internet*.

Acredita-se que a definição de uma metodologia de Banco de Dados para o tratamento de informações expostas desta forma possa facilitar o desenvolvimento de Sistemas de Informação que utilizam dados deste tipo. Pretende-se que o sistema seja robusto o suficiente para suprir a insuficiência de informações sobre a cidade, disponibilizando respostas rápidas, organizadas e da forma mais clara possível para os usuários.

### **3.4 Modelagem do Data Warehouse**

Nesta subseção são descritas as etapas realizadas para a modelagem da base de dados propriamente dita. Partiu-se da análise dos arquivos coletados, desenvolvendo em seguida a modelagem de acordo com a dimensionalidade dos dados analisados e conseqüentemente construindo o protótipo da modelagem, por fim a definição dos metadados.

#### **3.4.1 Análise dos arquivos**

Como o BD da Prefeitura possuía um aglomerado de dados, cujos formatos eram os mais variados foi necessário fazer uma análise sobre os arquivos a fim de identificar os dados que estariam no sistema computacional eliminando aqueles que não deveriam ser armazenados, os quais seriam gerados/calculados automaticamente pelo sistema, como valores totais e gráficos. No entanto, a necessidade de se realizar a análise de requisitos do sistema anteriormente à análise dos dados nos arquivos também foi identificada, pois assim a identificação dos dados para o sistema ocorreria com base nas definições dos requisitos.

A técnica de análise de requisitos é inicialmente utilizada para especificar as necessidades a serem supridas pelo sistema, onde deve-se primeiro identificar todas as partes que interagirão com o sistema de informação. Na definição dos usuários envolvidos é possível prever a forma final de apresentação dos dados, pois disto depende a quantidade de acessos que um determinado grupo de dados terá. Também é possível determinar, com esta técnica, a melhor forma de apresentação dos dados para que a informação seja mais clara e objetiva possível.

Através da análise de requisitos, observou-se que os usuários que interagirão com o sistema são empresários, possíveis empreendedores internos e externos à cidade e assim, foram definidos como parte do escopo do sistema. Também foram estabelecidas prioridades, por exemplo, as áreas mais visualizadas pelos investidores foram priorizadas na modelagem, de forma a atender ao máximo as suas expectativas desde o momento de planejamento do Sistema de Informação, pois segundo os conceitos estudados no capítulo 2, entende-se que usuários de Sistemas de Apoio a Decisão (SAD), estão tipicamente interessados em identificar tendências, comportamentos e não pretendem analisar individualmente cada registro no sistema.

Após estas definições, os arquivos em formato de apresentação de slides passaram por uma fase de análise crítica. Cada slide foi lido e cuidadosamente foram sendo extraídos os dados úteis ao sistema com o máximo cuidado para não deixar algum dado importante fora do escopo da modelagem. Esta filtragem foi realizada juntamente com a equipe da Prefeitura através de reuniões, pois esta equipe era a responsável pela qualidade e quantidade de informações que o Sistema deveria suprir. Com esta análise pronta, já estariam especificados os dados do Sistema, podendo então ser iniciada a modelagem conceitual.

#### 3.4.2 Modelagem e dimensionalidade dos dados

A modelagem do DW proposto foi feita de acordo com as diretrizes apresentadas na fundamentação teórica deste trabalho. Uma etapa inicial à modelagem do DW é a decisão por qual arquitetura utilizar. Partindo dos estudos realizados previamente, foi definida a arquitetura *bottom-up* como a arquitetura que mais se adequava à proposta do Sistema, pois esta arquitetura se caracteriza pela divisão por áreas dentro da organização do sistema, tendo o DW completo a partir

da modelagem destas pequenas áreas, denominadas *Data Marts*. Esta arquitetura torna objetiva a recuperação de informações e otimiza consideravelmente o tempo de consulta uma vez que diminui o escopo da consulta e objetiva a resposta da pergunta. Entende-se que um empresário que está interessado em investir em uma determinada área social, não necessita de dados sobre outra área não relacionada. Sendo assim, foram identificadas as seguintes áreas:

1. Comunicação
2. Educação
3. Geografia e Qualidade de vida
4. Qualificação e Ação Social
5. Saúde
6. Segurança
7. Drenagem
8. Economia
9. Economia do setor primário
10. Indústria, comércio e serviços
11. Infra-estrutura
12. Telecomunicações
13. Transporte
14. Apoio ao investidor
15. Estrutura do turismo

Em cada uma das áreas identificadas como importantes na análise das potencialidades do município e região foram definidas as tabelas Fato (*fact table*) e as tabelas Dimensão (*dimensional table*) seguindo o esquema estrela, cujo conceito foi tratado no capítulo 2. Para identificar as tabelas Fato foi necessário analisar os dados e verificar os eventos ou acontecimentos representados por valores aditivos e que tivessem um caráter evolutivo, podendo ser sempre questionado sobre essa evolução relativa a um período de tempo, como exemplo a quantidade de matrículas em uma rede de ensino, o índice de desenvolvimento humano (IDH) entre outros. Para identificar as dimensões do Fato foi feita uma análise respondendo as perguntas “Quando ocorreu o fato?”, “Onde ocorreu o fato?”, “Quem fez o fato acontecer?” e “O que ocorreu com o fato?”. A resposta a estas perguntas fornece

uma maneira prática e rápida de definir as Dimensões envolvidas no Fato. Não é obrigatório, que exista resposta para todas estas questões, mas as que existirem, se caracterizam como Dimensão. Por exemplo, para o Fato *Matricula*, a Dimensão *Tempo* responde a pergunta “Quando ocorreu o fato?”, as Dimensões *Nível de Ensino* e *Rede de Ensino* respondem à pergunta “Onde ocorreu o fato?”.

Na figura 17 é ilustrado um dos fatos e suas respectivas dimensões, no esquema estrela. O fato ilustrado é *Matricula*, modelado para o *Data Mart* Educação, com as dimensões *Tempo*, *Rede de Ensino* e *Nível de Ensino*.

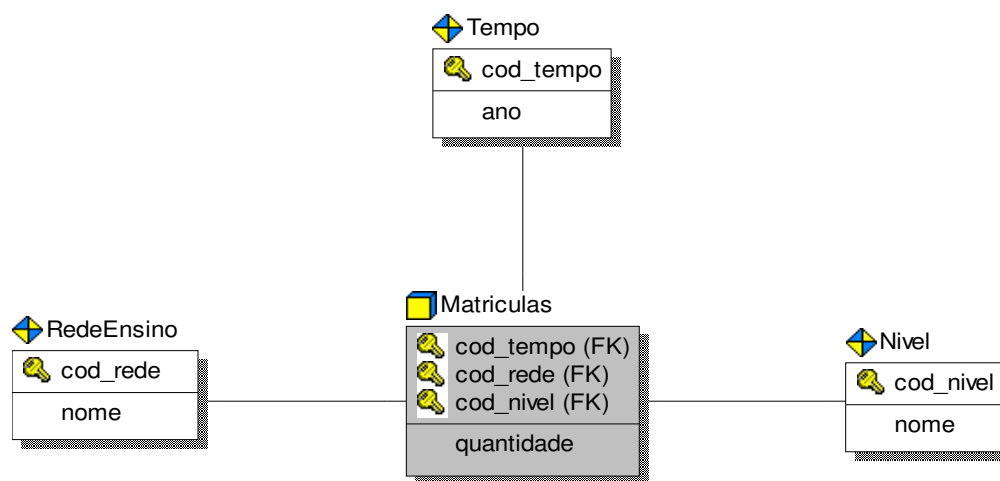


Figura 17 – Esquema Estrela do Fato Matrícula

### 3.4.3 Protótipo da modelagem desenvolvida

Para a primeira etapa da modelagem foi utilizada a ferramenta CASE (*Computer Aided Software Engineering*) Allfusion *Erwin Data Modeler* da empresa Computer Associates ([www.ca.com](http://www.ca.com)), em uma versão para testes, válida pelo período de 30 dias. Através desta ferramenta foi possível projetar conceitualmente a base de dados, com a visualização das tabelas Fato e das tabelas Dimensão. A decisão pela utilização desta ferramenta foi por ela oferecer diversas facilidades com relação ao projeto de Banco de Dados, uma vez que ela “interage” com o SGBD que se pretende manipular as tabelas. O SGBD utilizado no desenvolvimento do protótipo do DW foi o Microsoft SQL Server 2000, em uma versão também para testes válida pelo período de 120 dias. Optou-se pela utilização deste SGBD por oferecer recursos para criação de DWs e por possuir uma plataforma baseada na

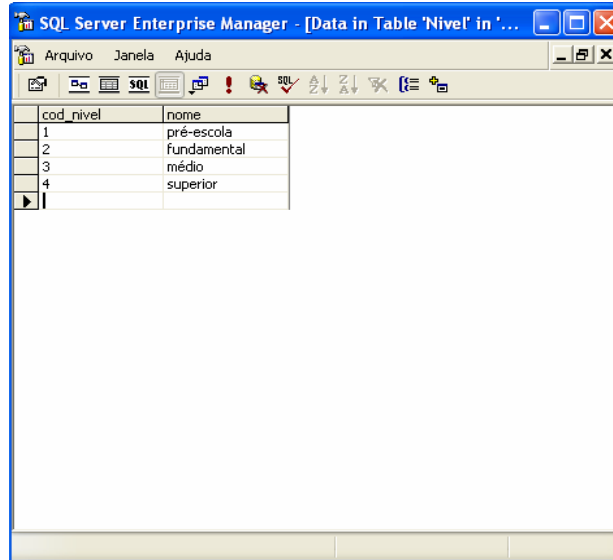


metodologia defendida por Ralph Kimball (1996), a qual também foi seguida neste trabalho. Esta metodologia apresenta o desenvolvimento de DWs a partir da criação de cubos de dados, que são originados pelas tabelas de Dimensão e de Fatos, para a análise dos dados.

Com a utilização da ferramenta Erwin foram modelados conceitualmente as bases de dados dos *Data Marts* do DW, sendo que os modelos podem ser visualizados no Apêndice A, incluindo a visualização das tabelas Fato com suas respectivas medidas e tabelas Dimensão. As tabelas Fato são identificadas pelo ícone de um cubo e as Dimensão por um ícone losango.

Para a construção física da base de dados foi utilizado o recurso que a ferramenta CASE ERwin possui de conectar-se automaticamente a uma base, no caso o SQL Server foi a escolhida, gerando assim os scripts SQL de cada modelo conceitual projetado. Este recurso otimizou consideravelmente o tempo de projeto, uma vez que não foi necessário escrever manualmente os scripts para a criação das tabelas. Os scripts podem ser visualizados no Apêndice B.

Após a modelagem conceitual e física, a etapa seguinte foi a de criação e processamento dos cubos multidimensionais, cuja tarefa foi realizada com o uso da ferramenta Microsoft SQL Server Analysis Services. Para que os cubos fossem criados era necessário que as tabelas de Fatos e Dimensão já estivessem com seus respectivos dados armazenados. Esta etapa de extração, limpeza e carregamento dos dados para a base de dados do DW é uma etapa muito complexa e importante, porém não foi tratada neste trabalho. Então, devido a esta particularidade os dados passaram por limpeza e carregamento manuais para a base de dados proposta neste estudo. Para carregar os dados foi utilizado o recurso da ferramenta Microsoft Enterprise Manager, o qual permite digitar cada registro da tabela, como mostra a figura 18.

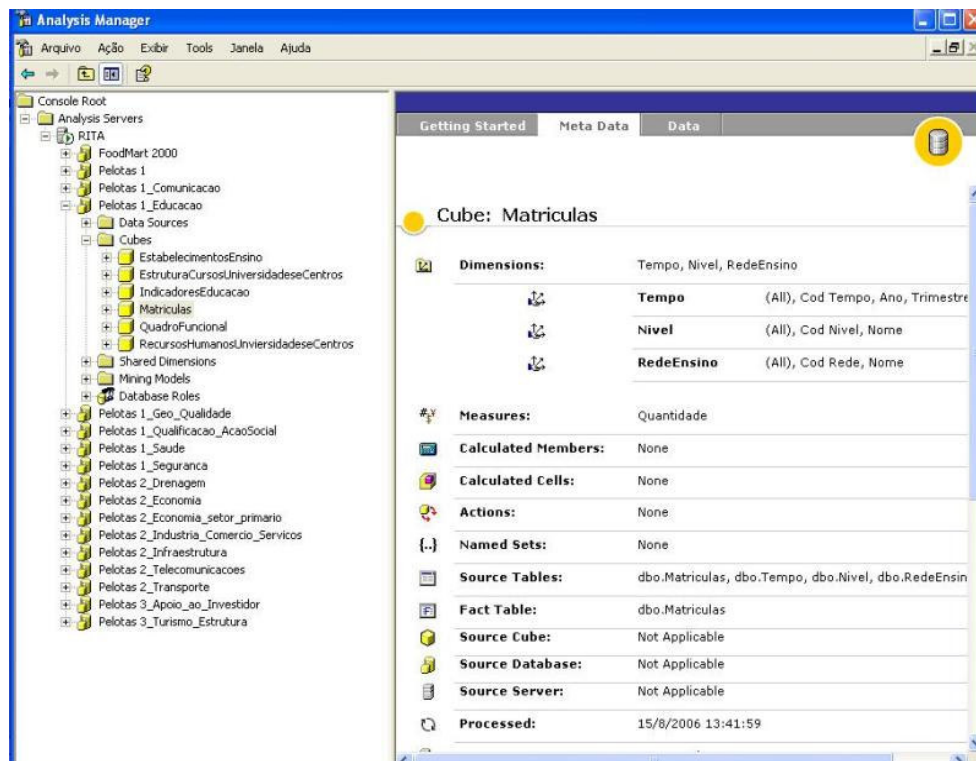


The screenshot shows the SQL Server Enterprise Manager interface. The main window displays a table with two columns: 'cod\_nivel' and 'nome'. The table contains four rows of data representing different educational levels.

cod_nivel	nome
1	pré-escola
2	fundamental
3	médio
4	superior

Figura 18 – Carregamento dos dados com o uso de Enterprise Manager

Com a base carregada, foi possível utilizar a ferramenta Microsoft Analysis Manager para a criação dos cubos. Na figura 19 é possível visualizar a forma de organização dos *Data Marts* e seus cubos.



The screenshot shows the Microsoft Analysis Manager interface. The left pane displays a tree view of the 'Console Root' with various Data Marts and Cubes. The right pane shows the details for a cube named 'Matriculas'.

Cube: Matriculas	
<b>Dimensions:</b>	Tempo, Nivel, RedeEnsino
<b>Tempo</b>	(All), Cod Tempo, Ano, Trimestre
<b>Nivel</b>	(All), Cod Nivel, Nome
<b>RedeEnsino</b>	(All), Cod Rede, Nome
<b>Measures:</b>	Quantidade
<b>Calculated Members:</b>	None
<b>Calculated Cells:</b>	None
<b>Actions:</b>	None
<b>Named Sets:</b>	None
<b>Source Tables:</b>	dbo.Matriculas, dbo.Tempo, dbo.Nivel, dbo.RedEnsino
<b>Fact Table:</b>	dbo.Matriculas
<b>Source Cube:</b>	Not Applicable
<b>Source Database:</b>	Not Applicable
<b>Source Server:</b>	Not Applicable
<b>Processed:</b>	15/8/2006 13:41:59

Figura 19 – Organização dos cubos nos Data Marts no Analysis Manager

Para a edição e montagem dos cubos, foi utilizada a ferramenta Analysis Service a qual possui um recurso de edição de cubos, que pode ser visualizado na figura 20 onde está sendo demonstrado como um exemplo o cubo Matrículas.

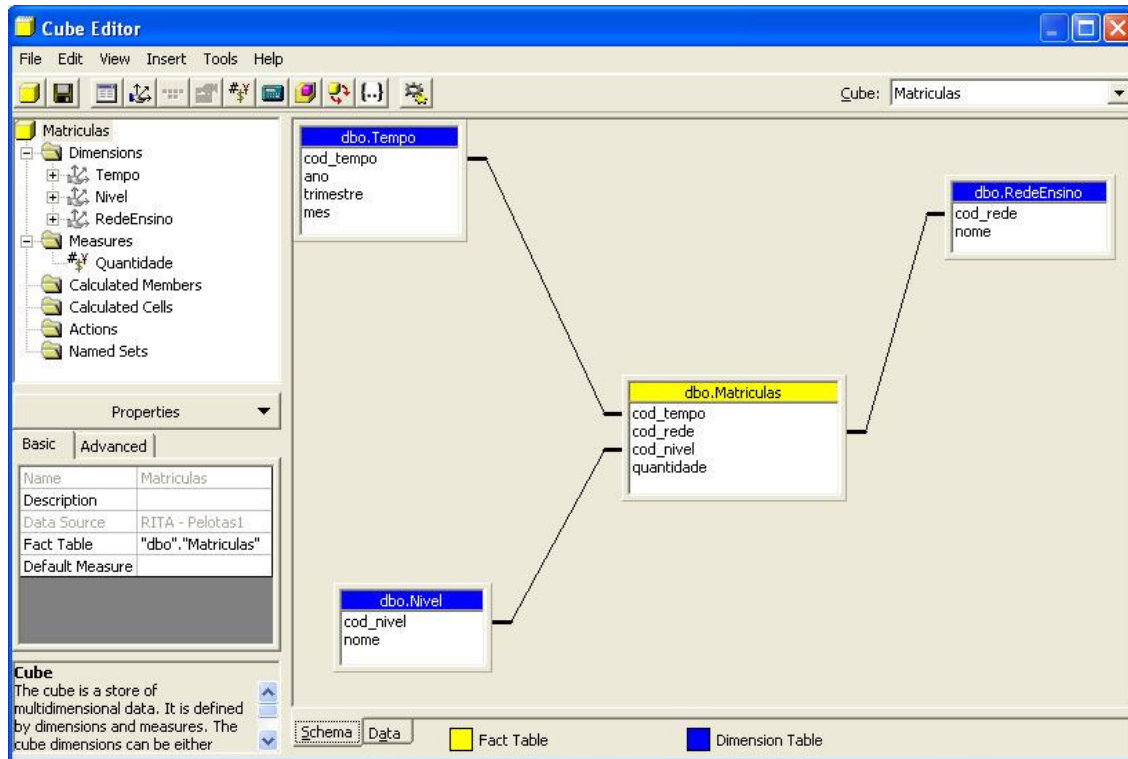


Figura 20 – Descrição da estrutura do cubo Matrículas na ferramenta Cube Editor

Para facilitar o entendimento da disposição dos dados em cubos, a figura 21 demonstra a modelagem do cubo processado a partir da combinação dos dados do fato Matricula com cada uma das suas dimensões Tempo, Rede de Ensino e Nível de Ensino.

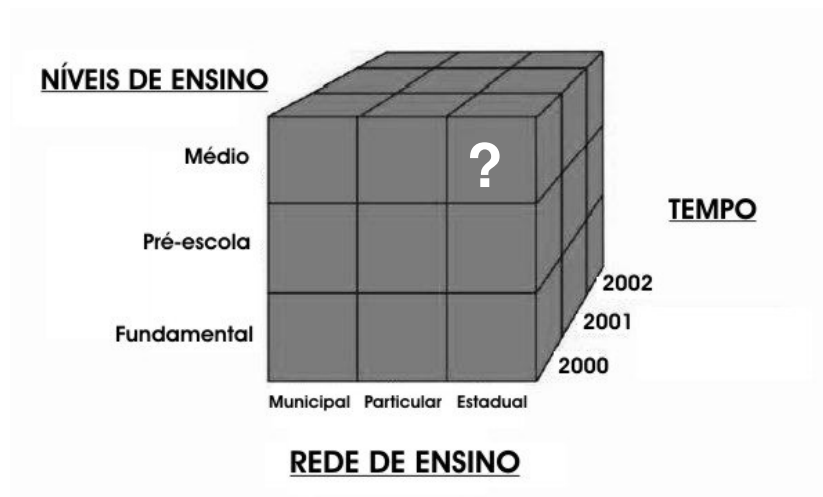


Figura 21 – Representação da disposição dos dados no DW

Em cada fatia deste cubo Matrículas, existe a quantidade de matrículas efetuadas em um determinado ano, em determinada rede de ensino e em determinado nível de ensino. A figura 22 ilustra a recuperação da informação referente à fatia com o sinal de interrogação (?) da figura 21.

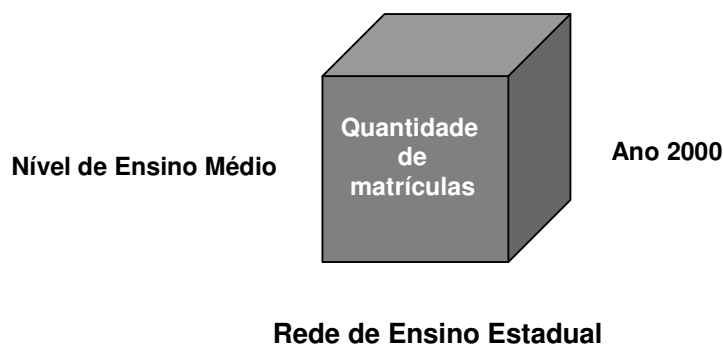


Figura 22 – Fatia do cubo com o resultado da consulta

Neste processo de criação dos cubos foi realizada a especificação da tabela Fato na própria ferramenta, as Dimensões e as medidas incluindo o tipo de armazenamento dos dados, no caso o tipo utilizado foi a HOLAP por esta ser uma abordagem híbrida em que faz uma combinação entre ROLAP e MOLAP, extraindo o melhor de cada uma delas demonstrando bom desempenho com grandes bases de dados. O registro dos membros das dimensões pode ser visualizada na figura 23,

onde é possível observar na Dimensão Tempo o registro de ano, e os meses do respectivo ano, padronizado 1 como janeiro, 2 como fevereiro e assim sucessivamente.

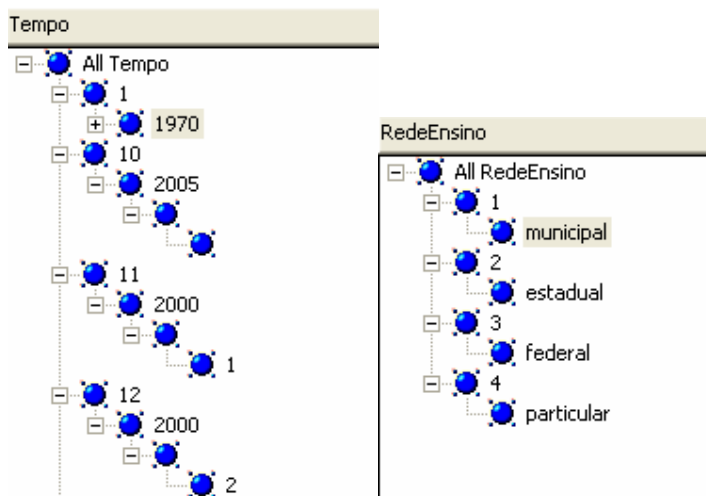


Figura 23 – Disposição dos membros da dimensão Tempo

Como é possível observar na figura 23, o ano de 2005 não está registrado com seus respectivos meses, diferente do ano de 2000, ou seja, no ano de 2000 é possível consultar dados específicos de meses o que não é possível no ano de 2005, onde as consultas devem ser generalistas. Esta dimensão Tempo é sempre uma dimensão considerada, pois é através dela que o DW apresenta o aspecto analítico temporal dos dados.

Posteriormente à edição dos cubos foi realizado o processamento de cada um deles, pois não basta o cubo ser especificado e criado, é necessário que seja processado para tornar possível a consulta sobre os seus dados. No processamento a ferramenta procura agregações entre os dados, e os relaciona automaticamente para otimizar o tempo de consulta ao cubo. Todos os cubos foram processados com sucesso, tornando possível então a consulta aos dados. Exemplos de consultas podem ser visualizadas no capítulo 4.

#### 3.4.3.1 Metadados considerados no sistema

Com base nos estudos sobre os metadados na seção 2.2.7, entende-se que eles possuem extrema importância no entendimento dos dados e a sua criação deve

ser planejada visando o grupo de usuários a que se destina, para determinar o tipo de linguagem que deve ser usada. Devido à grande quantidade de tabelas de fatos e dimensões, não é viável criar metadados para cada um destes elementos do DW, porém observou-se a necessidade de descrever os elementos dos cubos de cada área modelada. As informações sobre cada cubo são:

- nome do cubo
- medidas do cubo
- nome da tabela Fato
- nome das tabelas Dimensão
- fonte dos dados carregados no cubo

Estes metadados foram criados focalizando não os usuários finais do sistema, mas sim em termos de documentação para os técnicos que realizarão a manutenção da base de dados ou até mesmo melhorias futuras realizadas na base de dados. Os metadados utilizados na base podem ser visualizados no Apêndice C.

## ANÁLISE DOS RESULTADOS OBTIDOS

Buscou-se desenvolver a modelagem da base de dados do Sistema de Informação da Prefeitura de Pelotas de modo a diminuir ou eliminar os problemas apresentados pelo aglomerado de dados dispostos em apresentação de slides.

Os resultados obtidos quanto à problemática do armazenamento foram relevantemente positivos. O espaço em disco que o material em formato de slides ocupava somente na língua portuguesa era de 72.8 Megabytes. Enquanto que o espaço ocupado pela base do DW é somente de 11.58 Megabytes (contando com os registros armazenados por cada tabela, o esquema do BD em si e o espaço ocupado pelos cubos). Desta forma, pode-se concluir que foi obtido êxito na questão relacionada ao armazenamento dos dados. O grande êxito obtido no armazenamento se deve em parte pela análise crítica realizada nos dados para extrair apenas informação útil e armazenável. Além disso foi notada a relação do espaço armazenado com o nível de granularidade dos dados e detalhamento das consultas, o que não foi uma característica encontrada nos dados da Prefeitura.

Foram modeladas 146 tabelas contando as Dimensões e Fatos, 89 cubos em 15 *Data Marts*. A divisão dos dados por áreas, possibilitando a modelagem de *Data Marts* diminuiu a questão do tempo de recuperação de informações uma vez que, com a utilização de *Data Marts*, a procura por determinado dado é mais direta e conseqüentemente mais rápida, otimizando assim o tempo de recuperação de informação além do tempo de consulta realizado pela ferramenta OLAP.

O protótipo foi criado e visando a estruturação dos dados, foi bem sucedido. Os cubos foram processados utilizando a ferramenta Microsoft SQL Server Analysis Services e apresentaram bom desempenho, tendo o cubo "Abate" no *Data Mart* Economia do setor primário, o cubo com maior número de agregações, totalizando em 7. As agregações facilitam as consultas realizadas na medida em que as informações já ficam automaticamente cruzadas precisando apenas utilizar poucos índices no processo de busca do dado. Os testes foram realizados em um aplicativo do próprio Microsoft SQL Server Analysis Services, chamada MDX Sample Application, o qual permite que as consultas sejam criadas, editadas e aplicadas aos

cubos e os resultados sejam visualizados no mesmo ambiente do aplicativo. Foram feitas consultas aos cubos simulando consultas que seriam realizadas pelos usuários do DW estando ele localizado em um servidor real. Para a edição das consultas foi utilizada a linguagem SQL padrão, com as modificações exigidas pelo aplicativo utilizado.

Os resultados apresentados com a realização dos testes foram positivos pois tendo em vista a estruturação dos dados na base, as respostas às consultas apresentaram clareza e boa apresentação das informações.

Na figura 24 é ilustrado o resultado da consulta criada e aplicada ao cubo Matrículas no *Data Mart* Educação. Esta consulta diz respeito a uma pergunta simulada de um investidor : “Qual a quantidade de matrículas efetuadas na rede de ensino particular, do nível pré-escolar até o superior?”. O ano não foi especificado porque no momento a base não tem dados de outros anos, então o ano considerado foi o 2004, que é de quando se tem registros.

#### **Consulta:**

```
select
{{Measures}.[Quantidade]} on columns,
{{Nivel}.[nome].[pré-escola],[Nivel].[nome].[fundamental],
[Nivel].[nome].[médio],[Nivel].[nome].[superior]} on rows
from Matrículas
where ([RedeEnsino].[particular])
```

#### **Resultado:**

	Quantidade
pré-escola	1.201,00
fundamental	6.223,00
médio	1.944,00
superior	6.775,00

Figura 24 – Resultado da consulta no cubo Matrículas

Na figura 25 é possível visualizar o resultado de outra consulta executada ao cubo QuadroFuncional no *Data Mart* Educação. Esta consulta diz respeito à pergunta “Qual o quadro de professores e funcionários na rede de ensino federal, em todos os níveis de ensino?”.



**Consulta:**

```
select
```

```
    {[Measures].[Qtdade Professores], [Measures].[Qtdade Funcionarios]}
```

```
on columns,
```

```
    {[Nivel].[nome].members} on rows
```

```
from QuadroFuncional
```

```
where ([RedeEnsino].[federal])
```

**Resultado:**

	Qtdade Professores	Qtdade Funcionarios
pré-escola		
Fundamental		
médio	789	417
superior	894	1.215

Figura 25 – Resultado da consulta no cubo Matrículas

Pela falta da infra-estrutura exigida ao ambiente para o funcionamento do DW como a exigência que este esteja localizado em um servidor, não foram realizados testes reais. Por esse motivo foi desenvolvida a modelagem de um DW viável de implementação em um ambiente real, o qual não foi desenvolvido pelos motivos já citados.

## CONCLUSÕES E TRABALHOS FUTUROS

### 5.1 Conclusões

Concluiu-se neste trabalho que cada vez mais as organizações buscam obter dados relevantes à tomada de decisão utilizando como uma ferramenta poderosa o tratamento estratégico de dados para descoberta de conhecimento sobre eventos ocorridos dentro da organização em um determinado período de tempo.

Através dos estudos e pesquisa realizados, notou-se que a modelagem de dados precisa ser específica e apropriada ao objetivo do usuário final e, tendo em vista que os usuários em questão são investidores em busca de informações integradas e rápidas sobre a cidade de Pelotas e região Sul, o uso da modelagem multidimensional foi bem empregada para o Sistema de Apoio à Decisão proposto, junto com a técnica de *Data Warehousing*. Com o uso destas técnicas, as dificuldades encontradas no Banco de Dados inicial da Prefeitura foram eliminadas com êxito, podendo contribuir efetivamente com o que acredita-se ser um material poderoso na atração de investimentos em Pelotas.

Concluiu-se também que o principal objetivo proposto neste trabalho - criar um modelo de BD que atendesse os requisitos de informação da Prefeitura Municipal de Pelotas, viável de ser implementado em um Sistema de Informação *web*, facilitando assim a tomada de decisão por parte dos investidores - foi alcançado visto que foram realizados testes nos cubos modelados e os resultados foram satisfatórios em vários aspectos, comprovando a possibilidade de sua implementação definitiva. Infelizmente, por não dispor de recursos suficientes não foi possível a sua implementação propriamente dita. Espera-se que este trabalho tenha a continuidade que merece por sua utilidade à comunidade da região sul do estado.

## 5.2 Trabalhos futuros

A execução de um projeto completo de modelagem de dados, além da criação de uma base para um *Data Warehouse* é uma tarefa extensa, complexa e que para atingir o sucesso exige um período de tempo não tão pequeno por parte do desenvolvedor. A etapa de modelagem de um BD para o *Data Warehouse* que se objetivava para a Prefeitura, foi executada com sucesso. Porém, ainda estão faltando etapas a serem realizadas para que o sistema seja concluído, as quais estão abaixo listadas como sugestões de trabalhos futuros:

- automatização do processo de ETL dos dados na base do DW, através de desenvolvimento de metodologias para ferramentas de extração de dados de fontes heterogêneas e distribuídas, visando assim incentivar a continuidade do sistema por um longo período de tempo. A automatização proporciona um uso contínuo e abrangente de dados de fontes externas à Prefeitura, como utilizando-se como fonte de dados as bases do IBGE, por exemplo;
- desenvolvimento de aplicativos que executem o direcionamento das transformações que os dados devem sofrer após serem extraídos de uma base operacional e ser carregado no DW. Tornar os dados totalmente prontos para a recarga é uma tarefa bem complexa para se executada em um único trabalho porém seria um auxiliador muito importante no processo de ETL;
- a implementação definitiva de aplicativos que tornem funcional o *Data Warehouse* e disponibilização destes dados na *web*, caracterizando assim um *Data Webhouse* para os investidores da cidade de Pelotas e região sul.

## REFERÊNCIAS

- ANNES, R. **Sistemas de apoio a decisão**. Disponível em: <[http://www.pucrs.campus2.br/~annes/pos\\_sad.html](http://www.pucrs.campus2.br/~annes/pos_sad.html)>. Acesso em: 17 de junho de 2006.
- ANZANELLO, C.A. **OLAP: conceitos e utilização**. Disponível em: <[http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo\\_cynthia.pdf](http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_cynthia.pdf)>. Acesso em: 17 jun. 2006.
- BISPO, C.A.F. **Uma análise da nova geração de sistemas de apoio a decisão**. 1998. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18140/tde-04042004-152849/>>. Acesso em: 17 jun. 2006.
- BORTOLIN JUNIOR, A.M. **Sistemas de apoio à decisão**. Disponível em: <<http://ibirocay.al.urcamp.tche.br/infocamp/edicoes/nov05/Apoio%20a%20Decisao.pdf>>. Acesso em: 19 jun. 2006.
- CAMPOS, M. L.; ROCHA FILHO, A.V. **Tutorial de Data Warehouse**. Disponível em: <<http://genesis.nce.ufrj.br/dataware/tutorial/home.html>>. Acesso em: 19 jun. 2006.
- CIELO, I. **Data Mining**. Disponível em: <<http://www.datawarehouse.inf.br>>. Acesso em: 1 ago. 2006.
- COME, G. **Os metadados no ambiente de Data Warehouse**. Publicado em IV SemeAD. Disponível em: <<http://www.ead.fea.usp.br/semead/4semead/artigos/mqi/come.pdf>>. Acesso em: 28 ago. 2006.
- COLLIAT, G. **OLAP, relational, and multidimensional Database Systems**. SIGMOD Record, Vol. 25, No3, September 1996.
- COSTA, G.R.; SANTOS, V.A. Data Warehouse. In: CONGRESSO DE INICIAÇÃO CIENTÍFICA, 2004, Pelotas. **Anais do...** Pelotas: Universidade Federal de Pelotas, 2004.
- DAVID, Marco. **Managing Metadata Today and into the 21st Century**. Disponível em: <<http://www.dbmsmag.com/9708d16.html>>. Acesso em: 22 jun. 2006.
- DOMENICO, J.A. **Definição de um ambiente Data Warehouse em uma instituição de ensino superior**. 2001. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

ELMASRI, R.; NAVATHE, S.B. **Sistemas de Bancos de Dados**. 4.ed. Addison Wesley Longman, 2005.

FAVARETTO, F.; RHODEN, C.A. Considerações sobre atividades de identificação, localização e tratamento de dados na construção de um Data Warehouse. In: CONGRESSO DE TECNOLOGIAS PARA GESTÃO DE DADOS. 2003. **Anais do...** Disponível em: <[conged.deinfo.uepg.br/~iconged/Artigos/Artigo\\_05.pdf](http://conged.deinfo.uepg.br/~iconged/Artigos/Artigo_05.pdf)>. Acesso em: 10 ago. 2006.

FUJIWARA, D. **Bancos MOLAP**. Disponível em: <[www.dwbrasil.com.br/html/arto.molap.html](http://www.dwbrasil.com.br/html/arto.molap.html)>. Acesso em: 19 jun. 2006.

INMON, W.H. **Como construir o Data Warehouse**. Rio de Janeiro: Campus, 1997.

KANAZAWA, K.N. **A importância da informação e suas principais classificações**. Disponível em: <<http://www2.prudente.unesp.br/posti/download/time2.pdf>>. Acesso em: 15 abr. 2006.

KIMBALL, R. **The Data Warehouse Toolkit**. John Wiley & Sons, New York, 1996.

KIMBALL, R.; MERZ, R. **Data Webhouse: construindo o Data Warehouse para a Web**. Rio de Janeiro: Campus, 2000.

KIMBALL, R.; REEVES, L.; ROSS, M; THORNTHWAITE, W. **The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses**. John Wiley & Sons, New York, 1998.

LAUDON, K.C.; LAUDON, J.P. **Sistemas de informações gerenciais**. 5.ed. São Paulo: [s.n.], 2004.

MACHADO, F.N.R. **Tecnologia e projeto de Data Warehouse**. São Paulo: [s.n.], 2004.

NAVARRO, M.C. de A. **O que é Data Warehouse?** Disponível em: <[www.serpro.gov.br/publicacao/tematec/1996/ttec27](http://www.serpro.gov.br/publicacao/tematec/1996/ttec27)>. Acesso em: 19 jun. 2006.

PERNAS, A.M. **Modelagem de um Data Webhouse voltado a produção e comercialização de sementes**. 2003. Monografia (Graduação em Ciência da Computação) Instituto de Informática, Universidade Federal de Pelotas, Pelotas.

Revista da UNICAMP. **Análise Multidimensional. Informativo Técnico no. 54** (16/09/98), Disponível em: <<http://www.revista.unicamp.br/infotec/informacao/inf54.htm>>. Acesso em: 14 jun. 2006.

Revista da UNICAMP. **Data Warehouse. Informativo técnico no.46** (04/05/98), Disponível em: <<http://www.revista.unicamp.br/infotec/informacao/inf46.htm>>. Acesso em: 19 jun. 2006.

SCHLÖTTGEN, A. **Aspectos temporais de um Data Warehouse.** Disponível em:  
<[http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo\\_alexandre.doc](http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_alexandre.doc)>.  
Acesso em: 18 jun. 2006.

## APÉNDICES

## Apêndice A - Modelos Conceituais dos Data Marts

### Área Data Mart: Comunicação

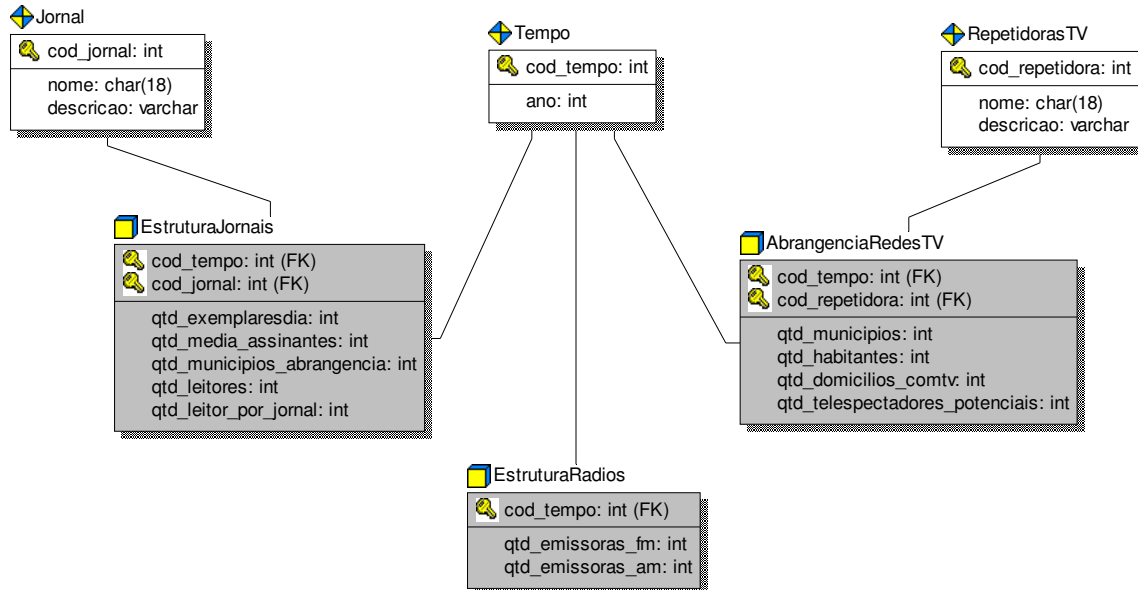


Figura A.1 – Modelo Conceitual Comunicação

### Área Data Mart: Educação

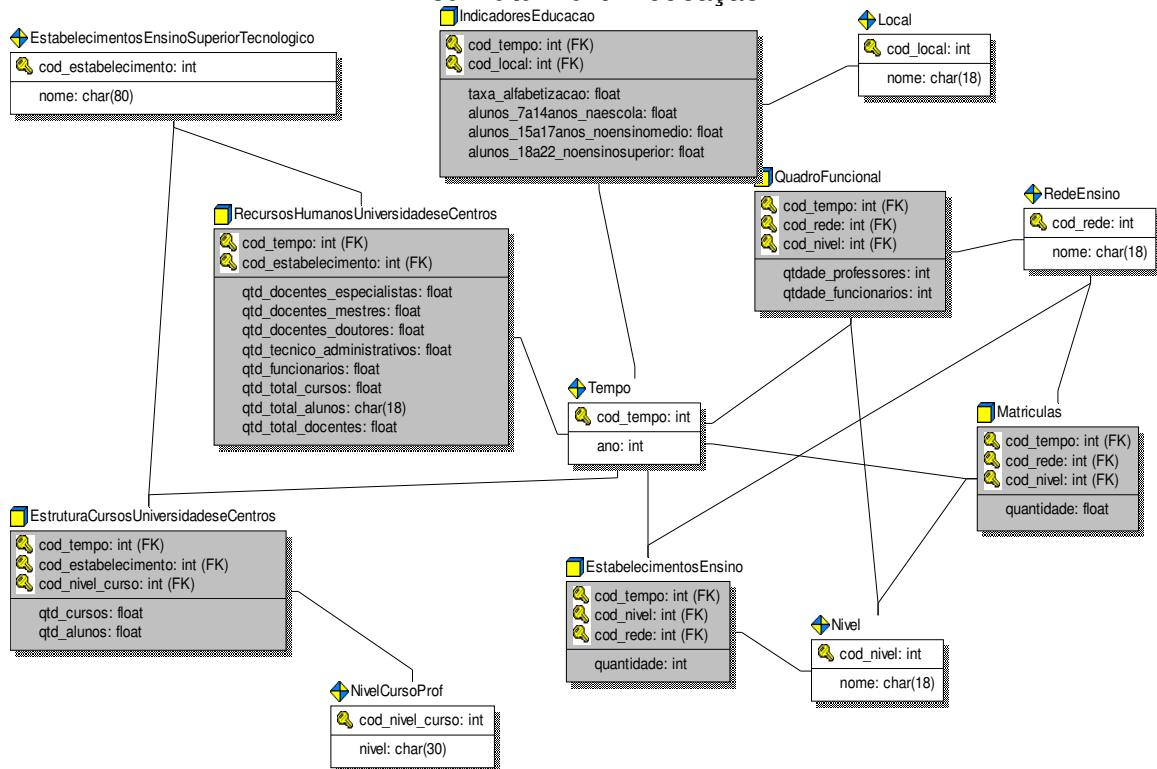


Figura A.2 – Modelo Conceitual Educação



### Área Data Mart: Geografia e Qualidade de vida

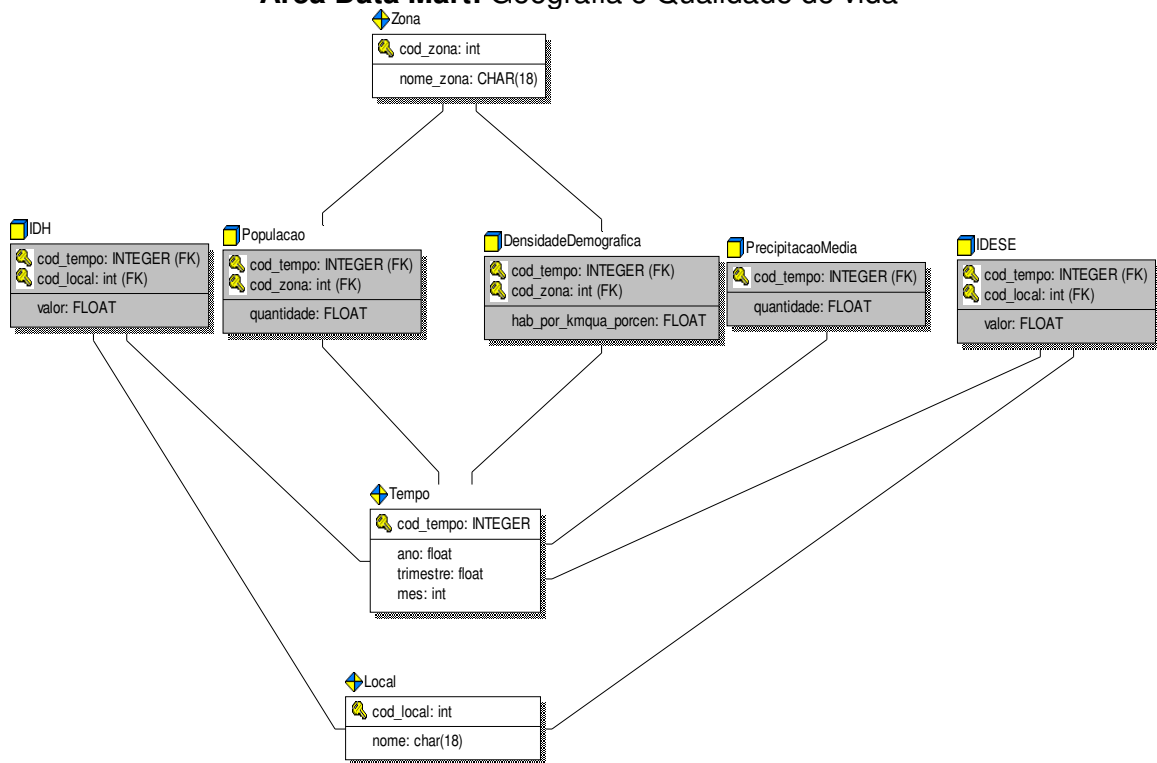


Figura A.3 – Modelo Conceitual Geografia e Qualidade de vida

### Área Data Mart: Qualificação Profissional e Ação Social

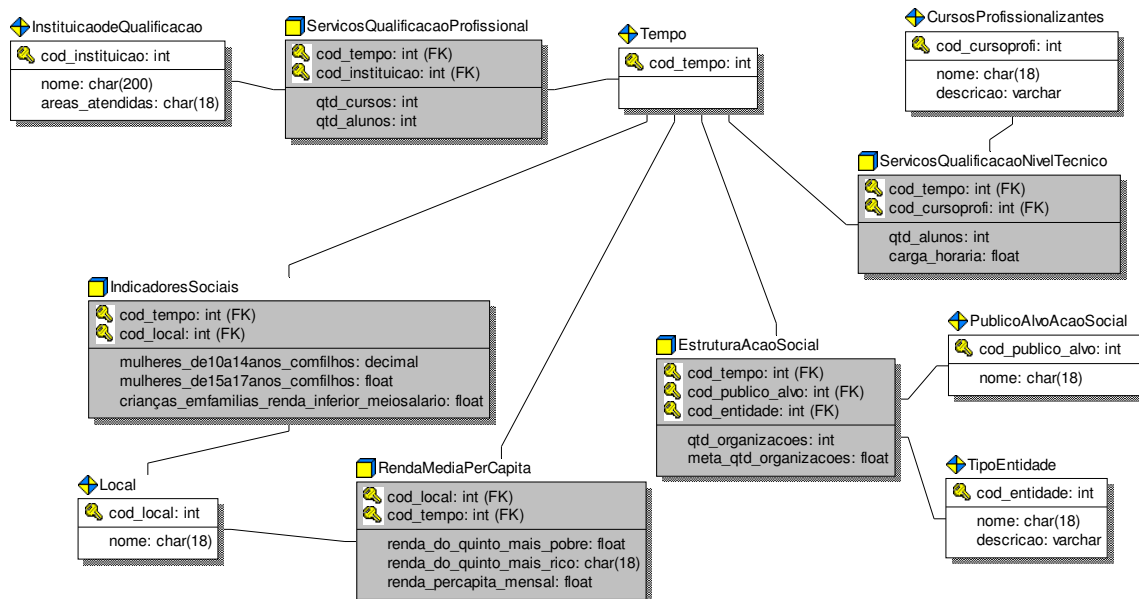


Figura A.4 – Modelo Conceitual Qualificação Profissional e Ação Social

## Área Data Mart: Saúde

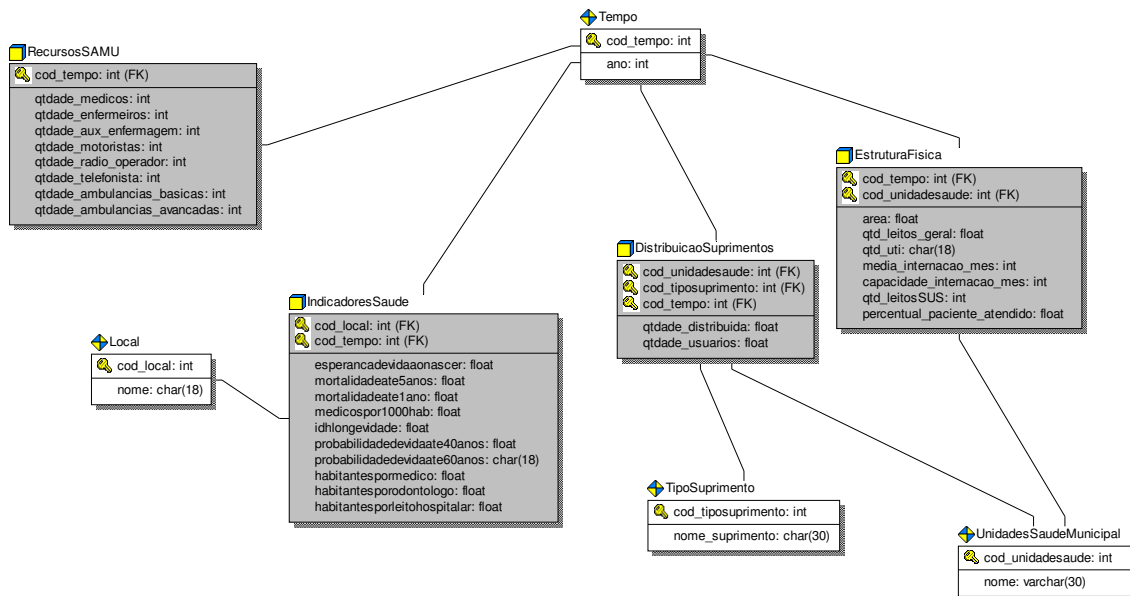


Figura A.5 – Modelo Conceitual Qualificação Profissional e Ação Social

## Área Data Mart: Segurança

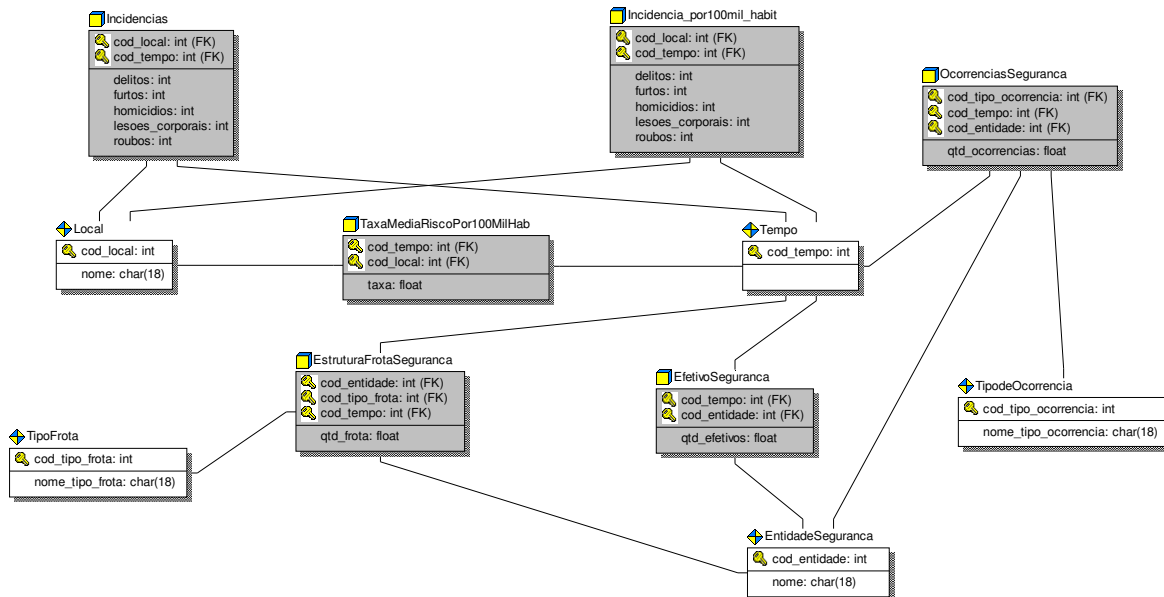


Figura A.6 – Modelo Conceitual Qualificação Profissional e Ação Social

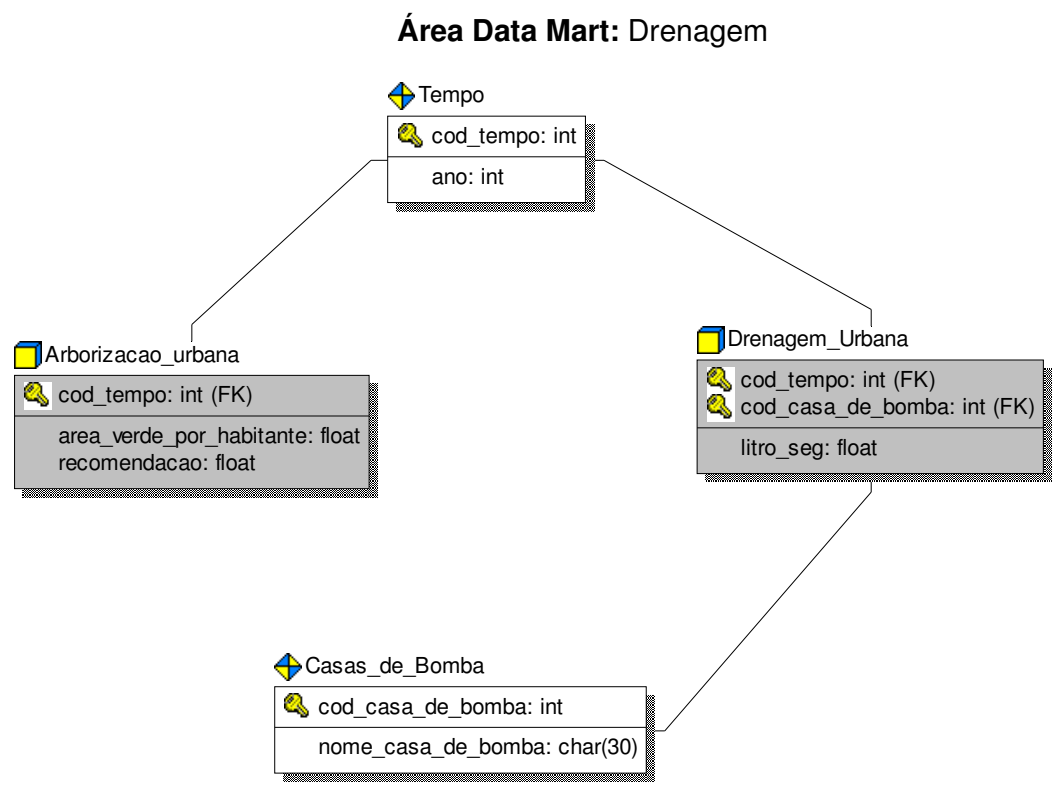


Figura A.7 – Modelo Conceitual Qualificação Profissional e Ação Social

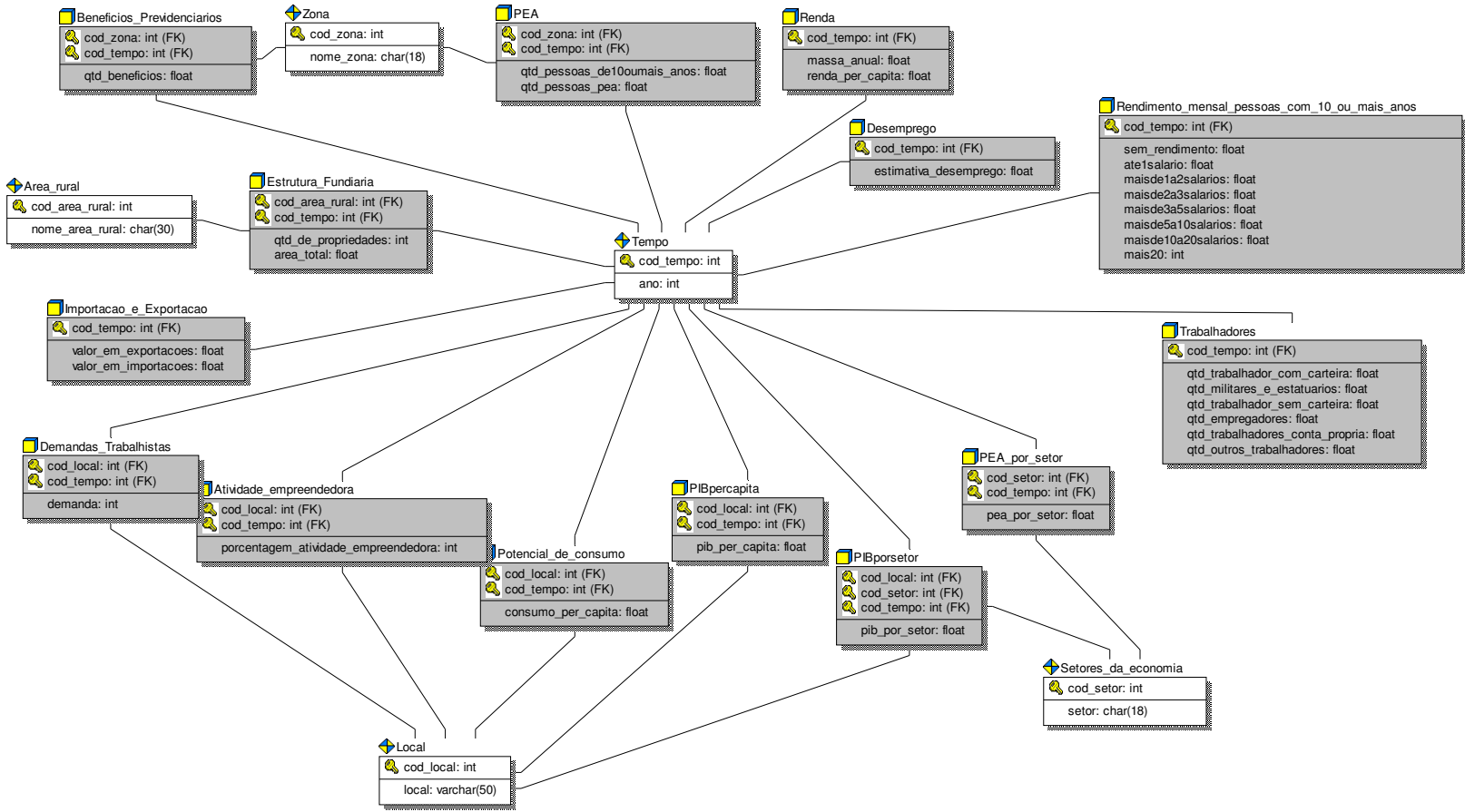


Figura A.8 – Modelo Conceitual Economia

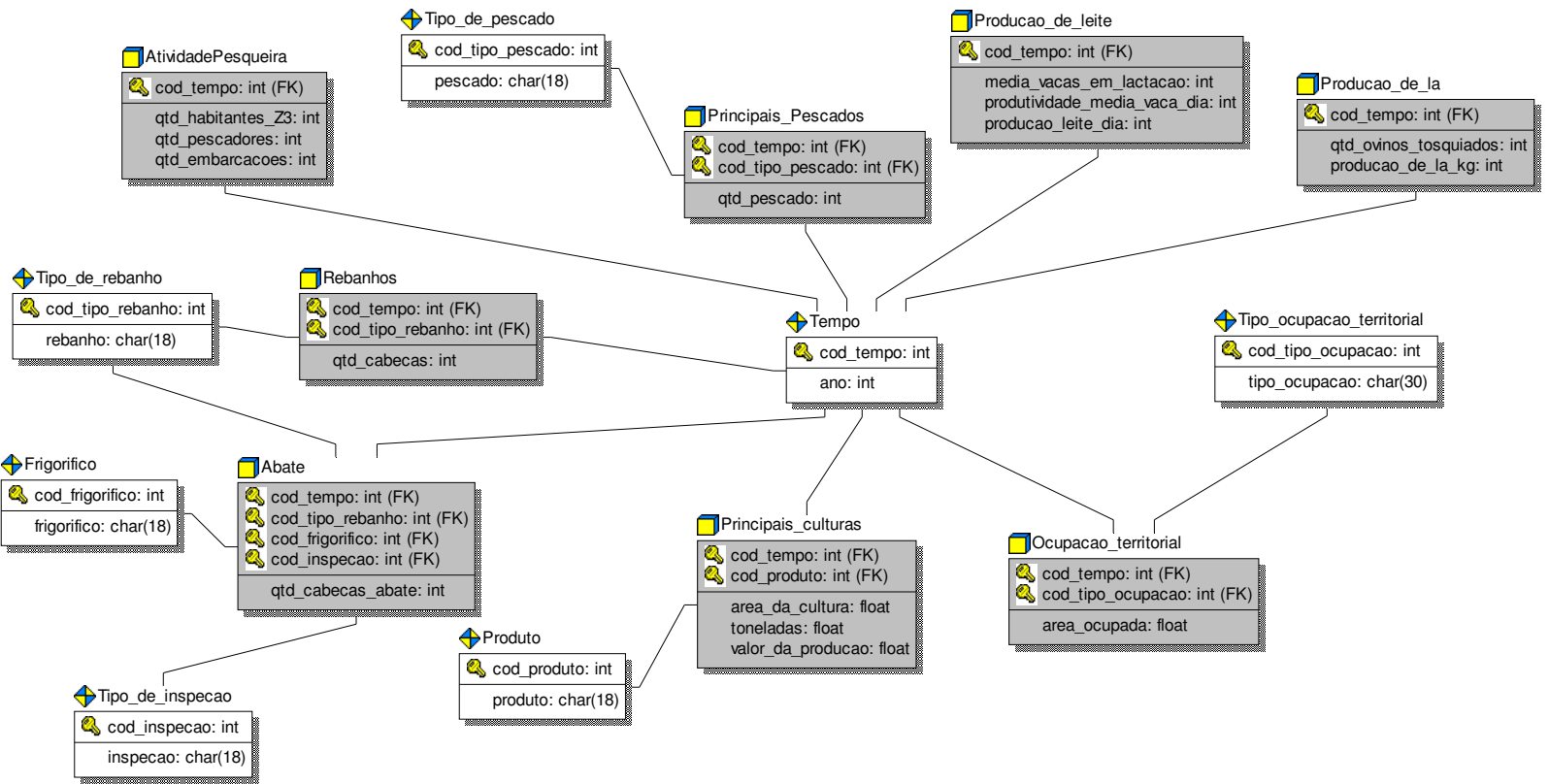


Figura A.9 – Modelo Conceitual Economia do Setor Primário

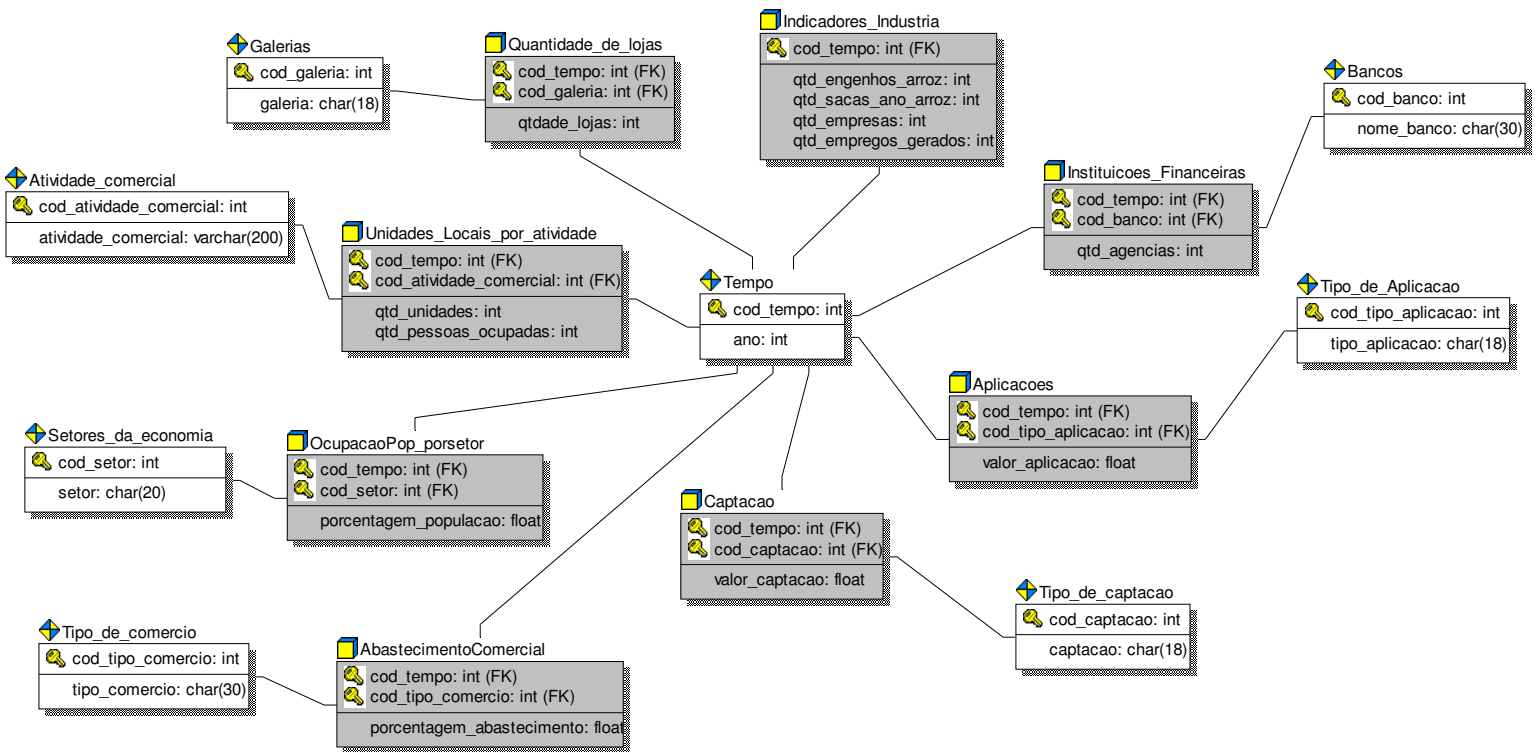


Figura A.10 – Modelo Conceitual Indústria, comércio e serviços

## Área Data Mart: Infra-estrutura

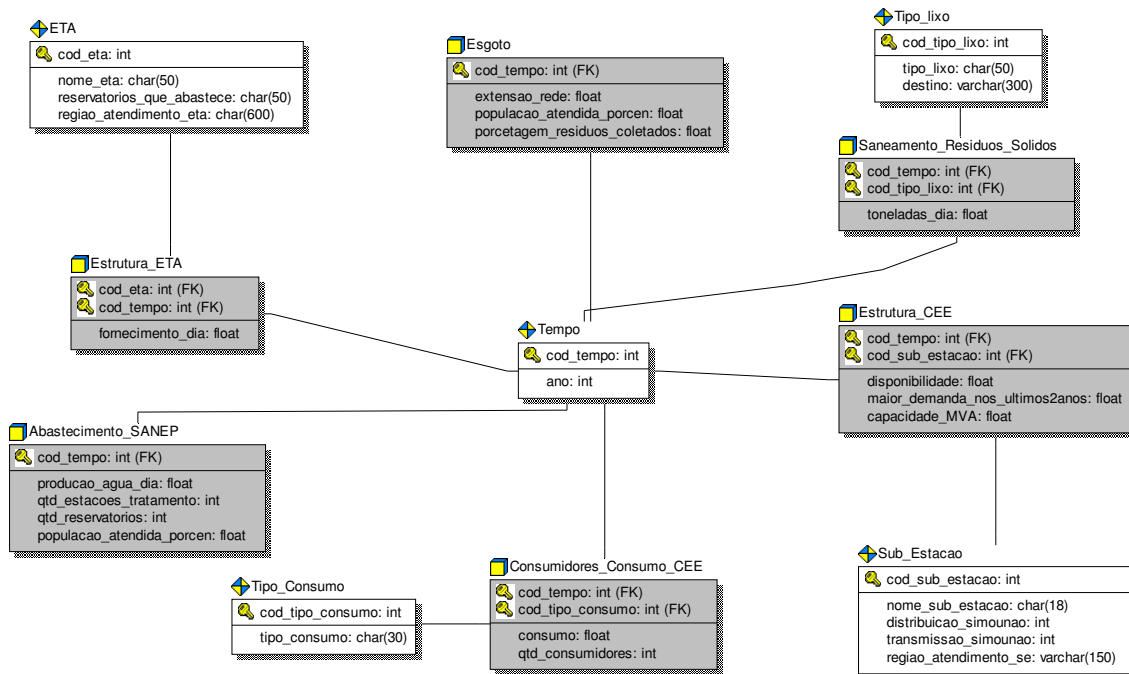


Figura A.11 – Modelo Conceitual Infra-estrutura

## Área Data Mart: Telecomunicações

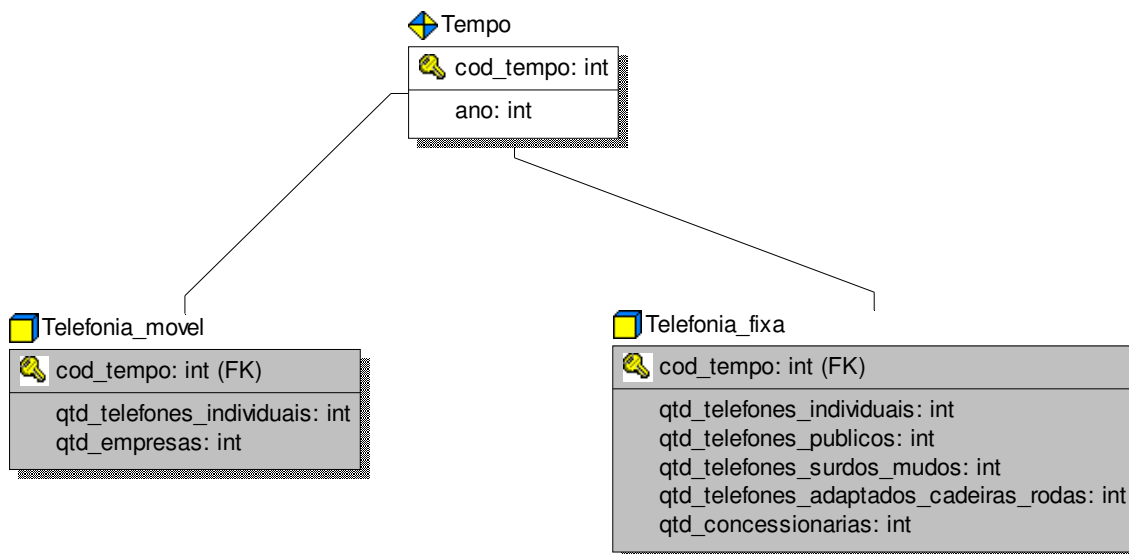
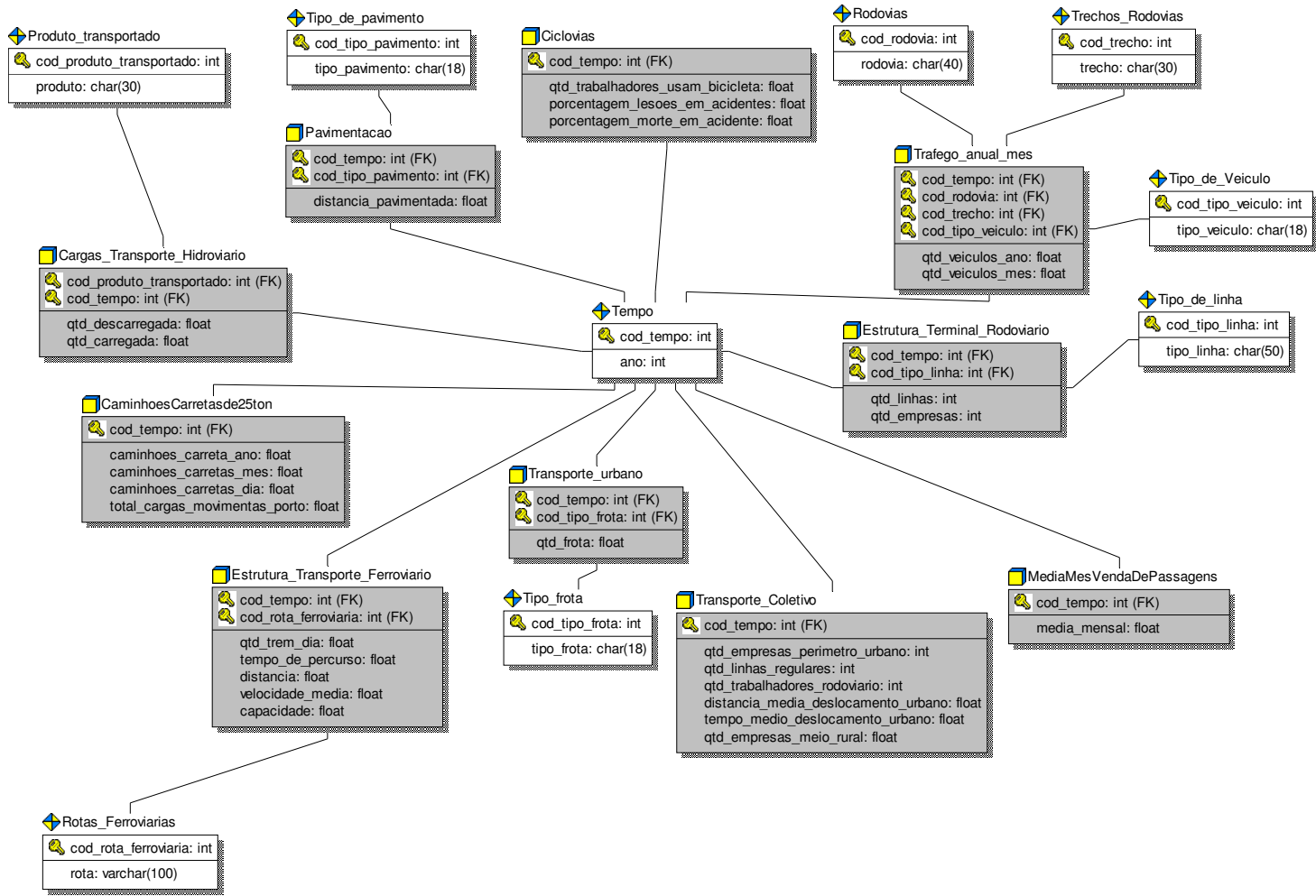


Figura A.12 – Modelo Conceitual Telecomunicações

Figura A.13 – Modelo Conceitual Transporte



Área Data Mart: Transporte



## Área Data Mart: Apoio ao investidor

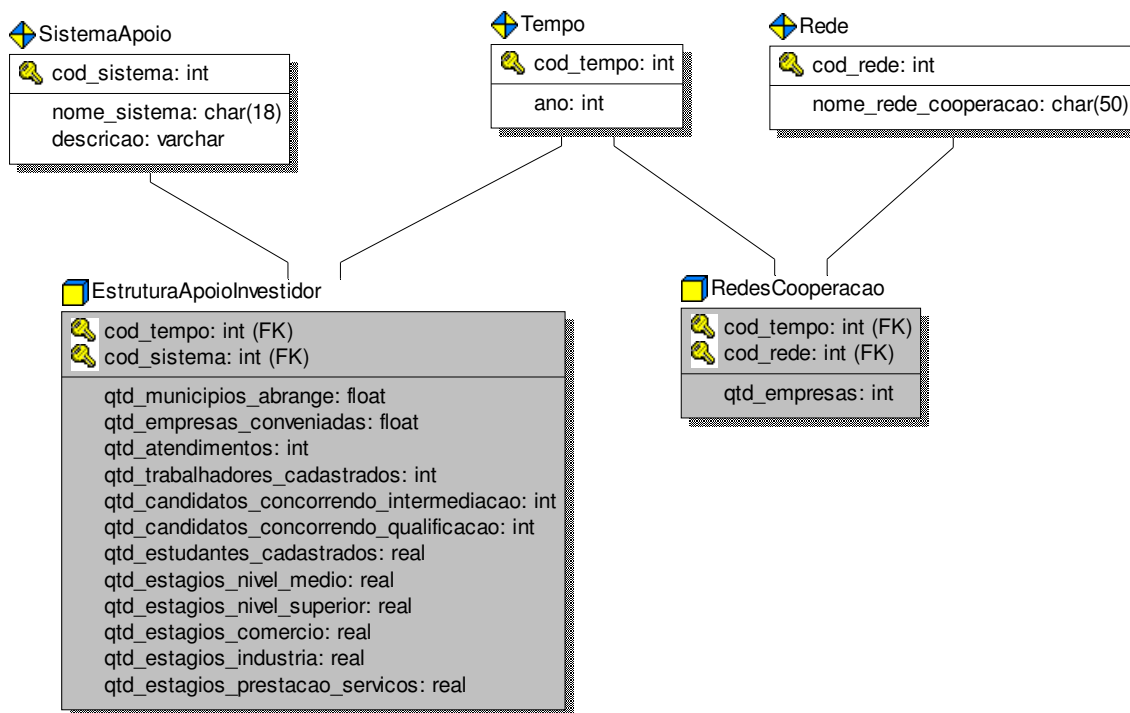


Figura A.14 – Modelo Conceitual Apoio ao investidor

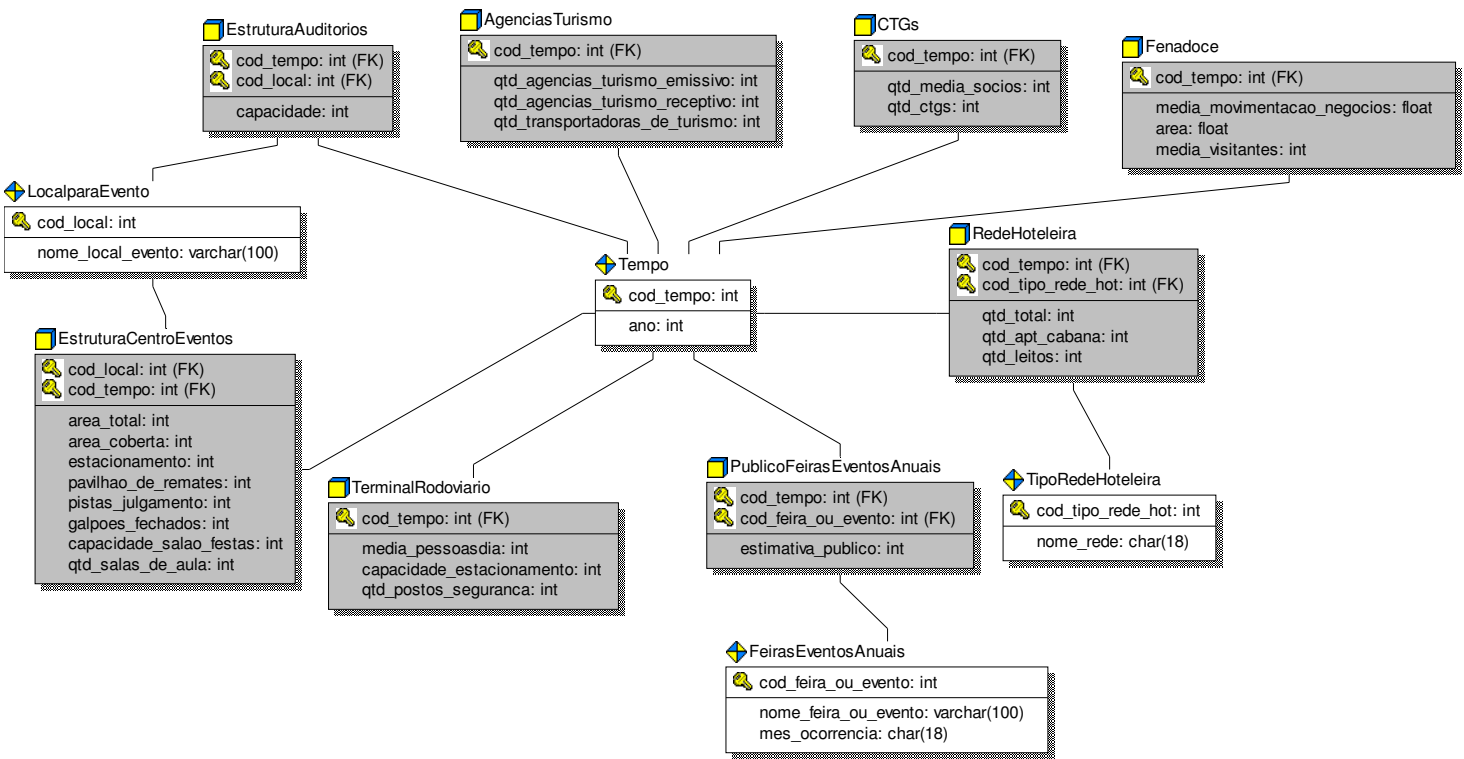


Figura A.15 – Modelo Conceitual Estrutura do turismo

## Apêndice B - Scripts SQL das tabelas gerados pela ferramenta ERwin

### Área Data Mart: Comunicação

#### Script: comunicacao.sql

```

/*
ACTION is CREATE Table RepetidorasTV
*/
CREATE TABLE RepetidorasTV (
    cod_repetidora    int NOT NULL,
    nome              char(18) NULL,
    descricao         varchar NULL
)
go
ALTER TABLE RepetidorasTV
    ADD PRIMARY KEY (cod_repetidora)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo         int NOT NULL,
    ano               int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table AbrangenciaRedesTV
*/
CREATE TABLE AbrangenciaRedesTV (
    cod_tempo         int NOT NULL,
    cod_repetidora    int NOT NULL,
    qtd_municipios    int NULL,
    qtd_habitantes    int NULL,
    qtd_domicilios_comtv int NULL,
    qtd telespectadores_potenciais int NULL
)
go
ALTER TABLE AbrangenciaRedesTV
    ADD PRIMARY KEY (cod_tempo, cod_repetidora)
go
/*
ACTION is CREATE Table Jornal
*/
CREATE TABLE Jornal (
    cod_jornal        int NOT NULL,
    nome              char(18) NULL,
    descricao         varchar NULL
)
go
ALTER TABLE Jornal
    ADD PRIMARY KEY (cod_jornal)
go
/*

```

```

ACTION is CREATE Table EstruturaJornais
*/
CREATE TABLE EstruturaJornais (
    cod_tempo      int NOT NULL,
    cod_jornal     int NOT NULL,
    qtd_exemplaresdia int NULL,
    qtd_media_assinantes int NULL,
    qtd_municipios_abrangencia int NULL,
    qtd_leitores   int NULL,
    qtd_leitor_por_jornal int NULL
)
go
ALTER TABLE EstruturaJornais
    ADD PRIMARY KEY (cod_tempo, cod_jornal)
go
/*
ACTION is CREATE Table EstruturaRadios
*/
CREATE TABLE EstruturaRadios (
    cod_tempo      int NOT NULL,
    qtd_emissoras_fm int NULL,
    qtd_emissoras_am int NULL
)
go
ALTER TABLE EstruturaRadios
    ADD PRIMARY KEY (cod_tempo)
go
ALTER TABLE AbrangenciaRedesTV
    ADD FOREIGN KEY (cod_repetidora)
        REFERENCES RepetidorasTV
go
ALTER TABLE AbrangenciaRedesTV
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaJornais
    ADD FOREIGN KEY (cod_jornal)
        REFERENCES Jornal
go
ALTER TABLE EstruturaJornais
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaRadios
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
Go

Área Data Mart: Educação
Script: educacao.sql

/*
ACTION is CREATE Table RedeEnsino
*/
CREATE TABLE RedeEnsino (
    cod_rede      int NOT NULL,
    nome          char(18) NULL

```

```

)
go
ALTER TABLE RedeEnsino
    ADD PRIMARY KEY (cod_rede)
go
/*
ACTION is CREATE Table Nivel
*/
CREATE TABLE Nivel (
    cod_nivel      int NOT NULL,
    nome           char(18) NULL
)
go
ALTER TABLE Nivel
    ADD PRIMARY KEY (cod_nivel)
go
/*
ACTION is CREATE Table Tempo
*/

CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
Go
/*
ACTION is CREATE Table EstabelecimentosEnsino
*/
CREATE TABLE EstabelecimentosEnsino (
    cod_tempo      int NOT NULL,
    cod_nivel      int NOT NULL,
    cod_rede       int NOT NULL,
    quantidade     int NULL
)
go
ALTER TABLE EstabelecimentosEnsino
    ADD PRIMARY KEY (cod_tempo, cod_nivel, cod_rede)
go
/*
ACTION is CREATE Table EstabelecimentosEnsinoSuperiorTecnologico
*/
CREATE TABLE EstabelecimentosEnsinoSuperiorTecnologico (
    cod_estabelecimento int NOT NULL,
    nome                char(80) NULL
)
go
ALTER TABLE EstabelecimentosEnsinoSuperiorTecnologico
    ADD PRIMARY KEY (cod_estabelecimento)
go
/*
ACTION is CREATE Table NivelCursoProf
*/

```

```

CREATE TABLE NivelCursoProf (
    cod_nivel_curso    int NOT NULL,
    nivel              char(30) NULL
)
go
ALTER TABLE NivelCursoProf
    ADD PRIMARY KEY (cod_nivel_curso)
go
/*
    ACTION is CREATE Table EstruturaCursosUniversidadeCentros
*/
CREATE TABLE EstruturaCursosUniversidadeCentros (
    qtd_cursos        float NULL,
    qtd_alunos        float NULL,
    cod_tempo         int NOT NULL,
    cod_estabelecimento int NOT NULL,
    cod_nivel_curso   int NOT NULL
)
go

ALTER TABLE EstruturaCursosUniversidadeCentros
    ADD PRIMARY KEY (cod_tempo, cod_estabelecimento,
                    cod_nivel_curso)
go
/*
    ACTION is CREATE Table Local
*/
CREATE TABLE Local (
    cod_local         int NOT NULL,
    nome              char(18) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go
/*
    ACTION is CREATE Table IndicadoresEducacao
*/
CREATE TABLE IndicadoresEducacao (
    cod_tempo         int NOT NULL,
    taxa_alfabetizacao float NULL,
    alunos_7a14anos_naescola float NULL,
    cod_local         int NOT NULL,
    alunos_15a17anos_noensinomedio float NULL,
    alunos_18a22_noensinosuperior float NULL,
    alunos_25anosoumais_nivelsuperior float NULL,
    alunos_25anosoumais_commenos8anosdeestudo float NULL,
    prof_ensinofundamental_comcursosuperior float NULL
)
go
ALTER TABLE IndicadoresEducacao
    ADD PRIMARY KEY (cod_tempo, cod_local)
go
/*

```

```

ACTION is CREATE Table Matriculas
*/
CREATE TABLE Matriculas (
    cod_tempo      int NOT NULL,
    cod_rede       int NOT NULL,
    cod_nivel      int NOT NULL,
    quantidade     float NULL
)
go
ALTER TABLE Matriculas
    ADD PRIMARY KEY (cod_tempo, cod_rede, cod_nivel)
go
/*
ACTION is CREATE Table QuadroFuncional
*/
CREATE TABLE QuadroFuncional (
    cod_tempo      int NOT NULL,
    cod_rede       int NOT NULL,
    cod_nivel      int NOT NULL,
    qtdade_professores int NULL,
    qtdade_funcionarios int NULL
)
go
ALTER TABLE QuadroFuncional
    ADD PRIMARY KEY (cod_tempo, cod_rede, cod_nivel)
go
/*
ACTION is CREATE Table RecursosHumanosUniversidadeCentros
*/
CREATE TABLE RecursosHumanosUniversidadeCentros (
    qtd_docentes_especialistas float NULL,
    qtd_docentes_mestres float NULL,
    qtd_docentes_doutores float NULL,
    qtd_tecnico_administrativos float NULL,
    qtd_funcionarios float NULL,
    qtd_total_cursos float NULL,
    qtd_total_alunos char(18) NULL,
    qtd_total_docentes float NULL,
    cod_tempo      int NOT NULL,
    cod_estabelecimento int NOT NULL
)
go
ALTER TABLE RecursosHumanosUniversidadeCentros
    ADD PRIMARY KEY (cod_tempo, cod_estabelecimento)
go
ALTER TABLE EstabelecimentosEnsino
    ADD FOREIGN KEY (cod_rede)
        REFERENCES RedeEnsino
go

ALTER TABLE EstabelecimentosEnsino
    ADD FOREIGN KEY (cod_nivel)
        REFERENCES Nivel
go

```

```
ALTER TABLE EstabelecimentosEnsino
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaCursosUniversidadeCentros
    ADD FOREIGN KEY (cod_nivel_curso)
        REFERENCES NivelCursoProf
go
ALTER TABLE EstruturaCursosUniversidadeCentros
    ADD FOREIGN KEY (cod_estabelecimento)
        REFERENCES EstabelecimentosEnsinoSuperiorTecnologico
go
ALTER TABLE EstruturaCursosUniversidadeCentros
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE IndicadoresEducacao
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local
go
ALTER TABLE IndicadoresEducacao
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Matriculas
    ADD FOREIGN KEY (cod_nivel)
        REFERENCES Nivel
go
ALTER TABLE Matriculas
    ADD FOREIGN KEY (cod_rede)
        REFERENCES RedeEnsino
go
ALTER TABLE Matriculas
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE QuadroFuncional
    ADD FOREIGN KEY (cod_nivel)
        REFERENCES Nivel
go
ALTER TABLE QuadroFuncional
    ADD FOREIGN KEY (cod_rede)
        REFERENCES RedeEnsino
go
ALTER TABLE QuadroFuncional
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE RecursosHumanosUniversidadeCentros
    ADD FOREIGN KEY (cod_estabelecimento)
        REFERENCES EstabelecimentosEnsinoSuperiorTecnologico
go
ALTER TABLE RecursosHumanosUniversidadeCentros
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
```



```

go
                Área Data Mart: Geografia e Qualidade de vida
                Script: geo_qualidade.sql
/*
ACTION is CREATE Table Zona
*/
CREATE TABLE Zona (
    cod_zona      int NOT NULL,
    nome_zona     CHAR(18) NULL
)
go
ALTER TABLE Zona
    ADD PRIMARY KEY (cod_zona)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo     INTEGER NOT NULL,
    ano           float NULL,
    trimestre     float NULL,
    mes           int NULL
)
go

ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table DensidadeDemografica
*/
CREATE TABLE DensidadeDemografica (
    cod_tempo     INTEGER NOT NULL,
    hab_por_kmqua_porcen FLOAT NULL,
    cod_zona      int NOT NULL
)
go
ALTER TABLE DensidadeDemografica
    ADD PRIMARY KEY (cod_tempo, cod_zona)
go
/*
ACTION is CREATE Table Local
*/
CREATE TABLE Local (
    cod_local     int NOT NULL,
    nome         char(18) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go

/*

```

```

ACTION is CREATE Table IDESE
*/
CREATE TABLE IDESE (
    cod_tempo      INTEGER NOT NULL,
    valor          FLOAT NULL,
    cod_local      int NOT NULL
)
go
ALTER TABLE IDESE
    ADD PRIMARY KEY (cod_tempo, cod_local)
Go
/*
ACTION is CREATE Table IDH
*/
CREATE TABLE IDH (
    cod_tempo      INTEGER NOT NULL,
    valor          FLOAT NULL,
    cod_local      int NOT NULL
)
go
ALTER TABLE IDH
    ADD PRIMARY KEY (cod_tempo, cod_local)
go
/*
ACTION is CREATE Table Populacao
*/
CREATE TABLE Populacao (
    cod_tempo      INTEGER NOT NULL,
    quantidade     FLOAT NULL,
    cod_zona       int NOT NULL
)
go
ALTER TABLE Populacao
    ADD PRIMARY KEY (cod_tempo, cod_zona)
go
/*
ACTION is CREATE Table PrecipitacaoMedia
*/
CREATE TABLE PrecipitacaoMedia (
    cod_tempo      INTEGER NOT NULL,
    quantidade     FLOAT NULL
)
go
ALTER TABLE PrecipitacaoMedia
    ADD PRIMARY KEY (cod_tempo)
go
ALTER TABLE DensidadeDemografica
    ADD FOREIGN KEY (cod_zona)
        REFERENCES Zona
go
ALTER TABLE DensidadeDemografica
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE IDESE

```

```

        ADD FOREIGN KEY (cod_local)
            REFERENCÉS Local
go
ALTER TABLE IDESE
    ADD FOREIGN KEY (cod_tempo)
        REFERENCÉS Tempo
go
ALTER TABLE IDH
    ADD FOREIGN KEY (cod_local)
        REFERENCÉS Local
go
ALTER TABLE IDH
    ADD FOREIGN KEY (cod_tempo)
        REFERENCÉS Tempo
go
ALTER TABLE Populacao
    ADD FOREIGN KEY (cod_zona)
        REFERENCÉS Zona
go
ALTER TABLE Populacao
    ADD FOREIGN KEY (cod_tempo)
        REFERENCÉS Tempo
go
ALTER TABLE PrecipitacaoMedia
    ADD FOREIGN KEY (cod_tempo)
        REFERENCÉS Tempo
go

```

**Área Data Mart: Qualificação Profissional e Ação Social**  
**Script: qualificacao\_acaosocial.sql**

```

/*
ACTION is CREATE Table CursosProfissionalizantes*/
CREATE TABLE CursosProfissionalizantes (
    cod_cursoprofi    int NOT NULL,
    nome              char(18) NULL,
    descricao         varchar NULL
)
go
ALTER TABLE CursosProfissionalizantes
    ADD PRIMARY KEY (cod_cursoprofi)
go
/*
ACTION is CREATE Table TipoEntidade
*/
CREATE TABLE TipoEntidade (
    cod_entidade     int NOT NULL,
    nome              char(18) NULL,
    descricao         varchar NULL
)
go
ALTER TABLE TipoEntidade
    ADD PRIMARY KEY (cod_entidade)
go
/*
ACTION is CREATE Table PublicoAlvoAcaoSocial
*/

```

```

CREATE TABLE PublicoAlvoAcaoSocial (
    cod_publico_alvo int NOT NULL,
    nome char(18) NULL
)
go

ALTER TABLE PublicoAlvoAcaoSocial
    ADD PRIMARY KEY (cod_publico_alvo)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo int NOT NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table EstruturaAcaoSocial
*/
CREATE TABLE EstruturaAcaoSocial (
    cod_tempo int NOT NULL,
    cod_publico_alvo int NOT NULL,
    cod_entidade int NOT NULL,
    qtd_organizacoes int NULL,
    meta_qtd_organizacoes float NULL
)
go
ALTER TABLE EstruturaAcaoSocial
    ADD PRIMARY KEY (cod_tempo, cod_publico_alvo, cod_entidade)
go
/*
ACTION is CREATE Table Local
*/
CREATE TABLE Local (
    cod_local int NOT NULL,
    nome char(18) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go
/*
ACTION is CREATE Table IndicadoresSociais
*/
CREATE TABLE IndicadoresSociais (
    cod_tempo int NOT NULL,
    cod_local int NOT NULL,
    mulheres_de10a14anos_comfilhos decimal NULL,
    mulheres_de15a17anos_comfilhos float NULL,
    crianças_emfamilias_renda_inferior_meiosalario float NULL
)
go

```

```

ALTER TABLE IndicadoresSociais
    ADD PRIMARY KEY (cod_tempo, cod_local)
go
/*
ACTION is CREATE Table InstituicaoQualificacao
*/

CREATE TABLE InstituicaoQualificacao (
    cod_instituicao int NOT NULL,
    nome char(200) NULL,
    areas_atendidas char(18) NULL
)
go
ALTER TABLE InstituicaoQualificacao
    ADD PRIMARY KEY (cod_instituicao)
go
/*
ACTION is CREATE Table RendaMediaPerCapita
*/
CREATE TABLE RendaMediaPerCapita (
    cod_local int NOT NULL,
    cod_tempo int NOT NULL,
    renda_do_quinto_mais_pobre float NULL,
    renda_do_quinto_mais_rico char(18) NULL,
    renda_percapita_mensal float NULL
)
go
ALTER TABLE RendaMediaPerCapita
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table ServicosQualificacaoNivelTecnico
*/
CREATE TABLE ServicosQualificacaoNivelTecnico (
    cod_tempo int NOT NULL,
    qtd_alunos int NULL,
    cod_cursoprofi int NOT NULL,
    carga_horaria float NULL
)
go
ALTER TABLE ServicosQualificacaoNivelTecnico
    ADD PRIMARY KEY (cod_tempo, cod_cursoprofi)
go
/*
ACTION is CREATE Table ServicosQualificacaoProfissional
*/
CREATE TABLE ServicosQualificacaoProfissional (
    cod_tempo int NOT NULL,
    cod_instituicao int NOT NULL,
    qtd_cursos int NULL,
    qtd_alunos int NULL
)
go
ALTER TABLE ServicosQualificacaoProfissional
    ADD PRIMARY KEY (cod_tempo, cod_instituicao)

```

```

go
ALTER TABLE EstruturaAcaoSocial
    ADD FOREIGN KEY (cod_entidade)
        REFERENCES TipoEntidade

go
ALTER TABLE EstruturaAcaoSocial
    ADD FOREIGN KEY (cod_publico_alvo)
        REFERENCES PublicoAlvoAcaoSocial

go

ALTER TABLE EstruturaAcaoSocial
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

go
ALTER TABLE IndicadoresSociais
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local

go
ALTER TABLE IndicadoresSociais
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

go
ALTER TABLE RendaMediaPerCapita
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

go
ALTER TABLE RendaMediaPerCapita
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local

go
ALTER TABLE ServicosQualificacaoNivelTecnico
    ADD FOREIGN KEY (cod_cursoprofi)
        REFERENCES CursosProfissionalizantes

go
ALTER TABLE ServicosQualificacaoNivelTecnico
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

go

ALTER TABLE ServicosQualificacaoProfissional
    ADD FOREIGN KEY (cod_instituicao)
        REFERENCES InstituicaodeQualificacao

go
ALTER TABLE ServicosQualificacaoProfissional
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

go


```

**Área Data Mart: Saúde**  
**Script: saude.sql**

```

/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL

```

```

)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table TipoSuprimento
*/
CREATE TABLE TipoSuprimento (
    cod_tiposuprimento int NOT NULL,
    nome_suprimento char(30) NULL
)
go
ALTER TABLE TipoSuprimento
    ADD PRIMARY KEY (cod_tiposuprimento)
go
/*
ACTION is CREATE Table UnidadesSaudeMunicipal
*/
CREATE TABLE UnidadesSaudeMunicipal (
    cod_unidadesaude int NOT NULL,
    nome varchar(30) NULL
)
go
ALTER TABLE UnidadesSaudeMunicipal
    ADD PRIMARY KEY (cod_unidadesaude)
go
/*
ACTION is CREATE Table DistribuicaoSuprimentos
*/
CREATE TABLE DistribuicaoSuprimentos (
    cod_unidadesaude int NOT NULL,
    cod_tiposuprimento int NOT NULL,
    cod_tempo int NOT NULL,
    qtdade_distribuida float NULL,
    qtdade_usuarios float NULL
)
go
ALTER TABLE DistribuicaoSuprimentos
    ADD PRIMARY KEY (cod_unidadesaude, cod_tiposuprimento,
        cod_tempo)
go
/*
ACTION is CREATE Table EstruturaFisica
*/
CREATE TABLE EstruturaFisica (
    cod_tempo int NOT NULL,
    cod_unidadesaude int NOT NULL,
    area float NULL,
    qtd_leitos_geral float NULL,
    qtd_uti char(18) NULL,
    media_internacao_mes int NULL,
    capacidade_internacao_mes int NULL,
    qtd_leitosSUS int NULL,
    percentual_paciente_atendido float NULL

```

```

)
go
ALTER TABLE EstruturaFisica
    ADD PRIMARY KEY (cod_tempo, cod_unidadesaude)
go
/*
ACTION is CREATE Table EstruturaServicos
*/
CREATE TABLE EstruturaServicos (
    cod_tempo          int NOT NULL,
    cod_unidadesaude  int NOT NULL,
    especialidades_medicas varchar(200) NULL,
    servicos_especializados varchar(200) NULL,
    descricao          varchar(200) NULL
)
go
ALTER TABLE EstruturaServicos
    ADD PRIMARY KEY (cod_tempo, cod_unidadesaude)
go
/*
ACTION is CREATE Table Local
*/
CREATE TABLE Local (
    cod_local          int NOT NULL,
    nome              char(18) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go
/*
ACTION is CREATE Table IndicadoresSaude
*/
CREATE TABLE IndicadoresSaude (
    esperancadevidaao nascer float NULL,
    mortalidadeate5anos float NULL,
    cod_local          int NOT NULL,
    mortalidadeate1ano float NULL,
    cod_tempo          int NOT NULL,
    medicospor1000hab float NULL,
    idhlongevidade     float NULL,
    probabilidadedevidaate40anos float NULL,
    probabilidadedevidaate60anos char(18) NULL,
    habitantespormedico float NULL,
    habitantesporodontologo float NULL,
    habitantesporleitohospitalar float NULL
)
go
ALTER TABLE IndicadoresSaude
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table RecursosSAMU
*/
CREATE TABLE RecursosSAMU (

```



```
        cod_tempo          int NOT NULL,
        qtidade_medicos    int NULL,
        qtidade_enfermeiros int NULL,
        qtidade_aux_enfermagem int NULL,
        qtidade_motoristas int NULL,
        qtidade_radio_operador int NULL,
        qtidade_telefonista int NULL,
        qtidade_ambulancias_basicas int NULL,
        qtidade_ambulancias_avancadas int NULL
    )
go
ALTER TABLE RecursosSAMU
    ADD PRIMARY KEY (cod_tempo)
go
ALTER TABLE DistribuicaoSuprimentos
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE DistribuicaoSuprimentos
    ADD FOREIGN KEY (cod_tiposuprimento)
        REFERENCES TipoSuprimento
go
ALTER TABLE DistribuicaoSuprimentos
    ADD FOREIGN KEY (cod_unidadesaude)
        REFERENCES UnidadesSaudeMunicipal
go
ALTER TABLE EstruturaFisica
    ADD FOREIGN KEY (cod_unidadesaude)
        REFERENCES UnidadesSaudeMunicipal
go
ALTER TABLE EstruturaFisica
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaServicos
    ADD FOREIGN KEY (cod_unidadesaude)
        REFERENCES UnidadesSaudeMunicipal
go
ALTER TABLE EstruturaServicos
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE IndicadoresSaude
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE IndicadoresSaude
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local
go
ALTER TABLE RecursosSAMU
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
```

**Área Data Mart: Segurança**  
**Script: seguranca.sql**

```
/*
ACTION is CREATE Table EntidadeSeguranca
*/
CREATE TABLE EntidadeSeguranca (
    cod_entidade    int NOT NULL,
    nome            char(18) NULL
)
go
ALTER TABLE EntidadeSeguranca
    ADD PRIMARY KEY (cod_entidade)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo       int NOT NULL
)
go

ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table EfetivoSeguranca
*/
CREATE TABLE EfetivoSeguranca (
    cod_tempo       int NOT NULL,
    cod_entidade    int NOT NULL,
    qtd_efetivos    float NULL
)
go
ALTER TABLE EfetivoSeguranca
    ADD PRIMARY KEY (cod_tempo, cod_entidade)
go
/*
ACTION is CREATE Table TipoFrota
*/
CREATE TABLE TipoFrota (
    cod_tipo_frota  int NOT NULL,
    nome_tipo_frota char(18) NULL
)
go
ALTER TABLE TipoFrota
    ADD PRIMARY KEY (cod_tipo_frota)
go
/*
ACTION is CREATE Table EstruturaFrotaSeguranca
*/
CREATE TABLE EstruturaFrotaSeguranca (
    cod_entidade    int NOT NULL,
    cod_tipo_frota  int NOT NULL,
    cod_tempo       int NOT NULL,
    qtd_frota       float NULL
)
```

```

)
go
ALTER TABLE EstruturaFrotaSeguranca
    ADD PRIMARY KEY (cod_entidade, cod_tipo_frota, cod_tempo)
go
/*
ACTION is CREATE Table Local
*/
CREATE TABLE Local (
    cod_local      int NOT NULL,
    nome           char(18) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go
/*
ACTION is CREATE Table Incidencia_por100mil_habit
*/
CREATE TABLE Incidencia_por100mil_habit (
    cod_local      int NOT NULL,
    cod_tempo      int NOT NULL,
    delitos        int NULL,
    furtos         int NULL,
    homicidios     int NULL,
    lesoes_corporais int NULL,
    roubos        int NULL
)
go
ALTER TABLE Incidencia_por100mil_habit
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table Incidencias
*/
CREATE TABLE Incidencias (
    cod_local      int NOT NULL,
    cod_tempo      int NOT NULL,
    delitos        int NULL,
    furtos         int NULL,
    homicidios     int NULL,
    lesoes_corporais int NULL,
    roubos        int NULL
)
go
ALTER TABLE Incidencias
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table TipodeOcorrencia
*/
CREATE TABLE TipodeOcorrencia (
    cod_tipo_ocorrencia int NOT NULL,
    nome_tipo_ocorrencia char(18) NULL
)

```

```

go
ALTER TABLE TipodeOcorrencia
    ADD PRIMARY KEY (cod_tipo_ocorrencia)
go
/*
ACTION is CREATE Table OcorrenciasSeguranca
*/
CREATE TABLE OcorrenciasSeguranca (
    cod_tipo_ocorrencia int NOT NULL,
    cod_tempo           int NOT NULL,
    cod_entidade        int NOT NULL,
    qtd_ocorrencias    float NULL
)
go
ALTER TABLE OcorrenciasSeguranca
    ADD PRIMARY KEY (cod_tipo_ocorrencia, cod_tempo, cod_entidade)
go
/*
ACTION is CREATE Table TaxaMediaRiscoPor100MilHab
*/
CREATE TABLE TaxaMediaRiscoPor100MilHab (
    cod_tempo           int NOT NULL,
    cod_local           int NOT NULL,
    taxa                float NULL
)
go
ALTER TABLE TaxaMediaRiscoPor100MilHab
    ADD PRIMARY KEY (cod_tempo, cod_local)
go
ALTER TABLE EfetivoSeguranca
    ADD FOREIGN KEY (cod_entidade)
        REFERENCES EntidadeSeguranca
go
ALTER TABLE EfetivoSeguranca
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaFrotaSeguranca
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaFrotaSeguranca
    ADD FOREIGN KEY (cod_tipo_frota)
        REFERENCES TipoFrota
go
ALTER TABLE EstruturaFrotaSeguranca
    ADD FOREIGN KEY (cod_entidade)
        REFERENCES EntidadeSeguranca
go
ALTER TABLE Incidencia_por100mil_habit
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Incidencia_por100mil_habit

```

```

        ADD FOREIGN KEY (cod_local)
            REFERENCES Local
go
ALTER TABLE Incidencias
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Incidencias
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local
go
ALTER TABLE OcorrenciasSeguranca
    ADD FOREIGN KEY (cod_entidade)
        REFERENCES EntidadeSeguranca
go
ALTER TABLE OcorrenciasSeguranca
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE OcorrenciasSeguranca
    ADD FOREIGN KEY (cod_tipo_ocorrencia)
        REFERENCES TipodeOcorrencia
go
ALTER TABLE TaxaMediaRiscoPor100MilHab
    ADD FOREIGN KEY (cod_local)
        REFERENCES Local
go

ALTER TABLE TaxaMediaRiscoPor100MilHab
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go

Área Data Mart: Drenagem
Script: drenagem. Sql

/*
    ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
    ACTION is CREATE Table Arborizacao_urbana
*/
CREATE TABLE Arborizacao_urbana (
    cod_tempo      int NOT NULL,
    area_verde_por_habitante float NULL,
    recomendacao   float NULL
)
go

```

```

ALTER TABLE Arborizacao_urbana
  ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Casas_de_Bomba
*/
CREATE TABLE Casas_de_Bomba (
  cod_casa_de_bomba int NOT NULL,
  nome_casa_de_bomba char(30) NULL
)
go
ALTER TABLE Casas_de_Bomba
  ADD PRIMARY KEY (cod_casa_de_bomba)
go
/*
ACTION is CREATE Table Drenagem_Urbana
*/
CREATE TABLE Drenagem_Urbana (
  cod_tempo int NOT NULL,
  cod_casa_de_bomba int NOT NULL,
  litro_seg float NULL
)
go
ALTER TABLE Drenagem_Urbana
  ADD PRIMARY KEY (cod_tempo, cod_casa_de_bomba)
go
ALTER TABLE Arborizacao_urbana
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE Drenagem_Urbana
  ADD FOREIGN KEY (cod_casa_de_bomba)
    REFERENCES Casas_de_Bomba
go
ALTER TABLE Drenagem_Urbana
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go

```

**Área Data Mart: Economia**  
**Script: economia.sql**

```

/*
ACTION is CREATE Table Area_rural
*/
CREATE TABLE Area_rural (
  cod_area_rural int NOT NULL,
  nome_area_rural char(30) NULL
)
go
ALTER TABLE Area_rural
  ADD PRIMARY KEY (cod_area_rural)
go
/*
ACTION is CREATE Table Local
*/

```

```
CREATE TABLE Local (
    cod_local      int NOT NULL,
    local          varchar(50) NULL
)
go
ALTER TABLE Local
    ADD PRIMARY KEY (cod_local)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Atividade_empreededora
*/
CREATE TABLE Atividade_empreededora (
    cod_local      int NOT NULL,
    porcentagem_atividade_empreededora int NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE Atividade_empreededora
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table Zona
*/
CREATE TABLE Zona (
    cod_zona       int NOT NULL,
    nome_zona      char(18) NULL
)
go
ALTER TABLE Zona
    ADD PRIMARY KEY (cod_zona)
go
/*
ACTION is CREATE Table Beneficios_Previdenciarios
*/
CREATE TABLE Beneficios_Previdenciarios (
    cod_zona       int NOT NULL,
    qtd_beneficios float NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE Beneficios_Previdenciarios
    ADD PRIMARY KEY (cod_zona, cod_tempo)
go
/*
```

```

ACTION is CREATE Table Demandas_Trabalhistas
*/
CREATE TABLE Demandas_Trabalhistas (
    cod_local      int NOT NULL,
    demanda        int NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE Demandas_Trabalhistas
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table Desemprego
*/
CREATE TABLE Desemprego (
    estimativa_desemprego float NULL,
    cod_tempo            int NOT NULL
)
go
ALTER TABLE Desemprego
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Estrutura_Fundiaria
*/
CREATE TABLE Estrutura_Fundiaria (
    cod_area_rural    int NOT NULL,
    qtd_de_propriedades int NULL,
    area_total        float NULL,
    cod_tempo         int NOT NULL
)
go
ALTER TABLE Estrutura_Fundiaria
    ADD PRIMARY KEY (cod_area_rural, cod_tempo)
go
/*
ACTION is CREATE Table Importacao_e_Exportacao*/
CREATE TABLE Importacao_e_Exportacao (
    valor_em_exportacoes float NULL,
    valor_em_importacoes float NULL,
    cod_tempo            int NOT NULL
)
go
ALTER TABLE Importacao_e_Exportacao
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table PEA
*/
CREATE TABLE PEA (
    cod_zona          int NOT NULL,
    qtd_pessoas_de10oumais_anos float NULL,
    qtd_pessoas_pea   float NULL,
    cod_tempo         int NOT NULL
)

```



```

go
ALTER TABLE PEA
    ADD PRIMARY KEY (cod_zona, cod_tempo)
go
/*
ACTION is CREATE Table Setores_da_economia
*/
CREATE TABLE Setores_da_economia (
    cod_setor      int NOT NULL,
    setor          char(18) NULL
)
go
ALTER TABLE Setores_da_economia
    ADD PRIMARY KEY (cod_setor)
go
/*
ACTION is CREATE Table PEA_por_setor
*/
CREATE TABLE PEA_por_setor (
    cod_setor      int NOT NULL,
    pea_por_setor  float NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE PEA_por_setor
    ADD PRIMARY KEY (cod_setor, cod_tempo)
go
/*
ACTION is CREATE Table PIBpercapita
*/
CREATE TABLE PIBpercapita (
    cod_local      int NOT NULL,
    pib_per_capita float NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE PIBpercapita
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table PIBporsetor
*/
CREATE TABLE PIBporsetor (
    cod_local      int NOT NULL,
    cod_setor      int NOT NULL,
    pib_por_setor  float NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE PIBporsetor
    ADD PRIMARY KEY (cod_local, cod_setor, cod_tempo)
go
/*
ACTION is CREATE Table Potencial_de_consumo
*/

```

```

CREATE TABLE Potencial_de_consumo (
    cod_local      int NOT NULL,
    consumo_per_capita float NULL,
    cod_tempo      int NOT NULL
)
go
ALTER TABLE Potencial_de_consumo
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table Renda
*/
CREATE TABLE Renda (
    massa_anual      float NULL,
    renda_per_capita float NULL,
    cod_tempo        int NOT NULL
)
go
ALTER TABLE Renda
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Rendimento_mensal_pessoas_com_10_ou_mais_anos
*/
CREATE TABLE Rendimento_mensal_pessoas_com_10_ou_mais_anos (
    sem_rendimento float NULL,
    ate1salario    float NULL,
    maisde1a2salarios float NULL,
    maisde2a3salarios float NULL,
    maisde3a5salarios float NULL,
    maisde5a10salarios float NULL,
    maisde10a20salarios float NULL,
    mais20          int NULL,
    cod_tempo       int NOT NULL
)
go
ALTER TABLE Rendimento_mensal_pessoas_com_10_ou_mais_anos
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Trabalhadores
*/
CREATE TABLE Trabalhadores (
    qtd_trabalhador_com_carteira float NULL,
    qtd_militares_e_estatuarios float NULL,
    qtd_trabalhador_sem_carteira float NULL,
    qtd_empregadores float NULL,
    qtd_trabalhadores_conta_propria float NULL,
    qtd_outros_trabalhadores float NULL,
    cod_tempo          int NOT NULL
)
go
ALTER TABLE Trabalhadores
    ADD PRIMARY KEY (cod_tempo)

```

```
go
ALTER TABLE Atividade_empreadedora
  ADD FOREIGN KEY (cod_local)
    REFERENCES Local

go
ALTER TABLE Atividade_empreadedora
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go
ALTER TABLE Beneficios_Previdenciarios
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go

ALTER TABLE Beneficios_Previdenciarios
  ADD FOREIGN KEY (cod_zona)
    REFERENCES Zona

go
ALTER TABLE Demandas_Trabalhistas
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go
ALTER TABLE Demandas_Trabalhistas
  ADD FOREIGN KEY (cod_local)
    REFERENCES Local

go
ALTER TABLE Desemprego
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go
ALTER TABLE Estrutura_Fundiaria
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go
ALTER TABLE Estrutura_Fundiaria
  ADD FOREIGN KEY (cod_area_rural)
    REFERENCES Area_rural

go
ALTER TABLE Importacao_e_Exportacao
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go

ALTER TABLE PEA
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo

go
ALTER TABLE PEA
  ADD FOREIGN KEY (cod_zona)
    REFERENCES Zona

go
ALTER TABLE PEA_por_setor
  ADD FOREIGN KEY (cod_setor)
    REFERENCES Setores_da_economia

go
```

```

ALTER TABLE PEA_por_setor
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE PIBpercapita
  ADD FOREIGN KEY (cod_local)
    REFERENCES Local
go
ALTER TABLE PIBpercapita
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE PIBporsetor
  ADD FOREIGN KEY (cod_local)
    REFERENCES Local
go

ALTER TABLE PIBporsetor
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE PIBporsetor
  ADD FOREIGN KEY (cod_setor)
    REFERENCES Setores_da_economia
go
ALTER TABLE Potencial_de_consumo
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE Potencial_de_consumo
  ADD FOREIGN KEY (cod_local)
    REFERENCES Local
go
ALTER TABLE Renda
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE
  Rendimento_mensal_pessoas_com_10_ou_mais_anos
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
ALTER TABLE Trabalhadores
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go

Área Data Mart: Economia do setor primário
Script: economia_setor_primario.sql

/*
ACTION is CREATE Table Tipo_de_inspecao
*/
CREATE TABLE Tipo_de_inspecao (
  cod_inspecao      int NOT NULL,
  inspecao          char(18) NULL

```

```
)
go
ALTER TABLE Tipo_de_inspecao
    ADD PRIMARY KEY (cod_inspecao)
go
/*
    ACTION is CREATE Table Frigorifico
*/
CREATE TABLE Frigorifico (
    cod_frigorifico int NOT NULL,
    frigorifico char(18) NULL
)
go
ALTER TABLE Frigorifico
    ADD PRIMARY KEY (cod_frigorifico)
go
/*
    ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo int NOT NULL,
    ano int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
    ACTION is CREATE Table Tipo_de_rebanho
*/
CREATE TABLE Tipo_de_rebanho (
    cod_tipo_rebanho int NOT NULL,
    rebanho char(18) NULL
)
go
ALTER TABLE Tipo_de_rebanho
    ADD PRIMARY KEY (cod_tipo_rebanho)
go
/*
    ACTION is CREATE Table Abate
*/
CREATE TABLE Abate (
    cod_tempo int NOT NULL,
    cod_tipo_rebanho int NOT NULL,
    cod_frigorifico int NOT NULL,
    cod_inspecao int NOT NULL,
    qtd_cabecas_abate int NULL
)
go
ALTER TABLE Abate
    ADD PRIMARY KEY (cod_tempo, cod_tipo_rebanho, cod_frigorifico,
        cod_inspecao)
go
/*
    ACTION is CREATE Table AtividadePesqueira
```

```

*/
CREATE TABLE AtividadePesqueira (
    cod_tempo      int NOT NULL,
    qtd_habitantes_Z3  int NULL,
    qtd_pescadores   int NULL,
    qtd_embarcacoes  int NULL
)
go
ALTER TABLE AtividadePesqueira
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Tipo_ocupacao_territorial
*/
CREATE TABLE Tipo_ocupacao_territorial (
    cod_tipo_ocupacao  int NOT NULL,
    tipo_ocupacao     char(30) NULL
)
go
ALTER TABLE Tipo_ocupacao_territorial
    ADD PRIMARY KEY (cod_tipo_ocupacao)
go
/*
ACTION is CREATE Table Ocupacao_territorial
*/
CREATE TABLE Ocupacao_territorial (
    cod_tempo      int NOT NULL,
    cod_tipo_ocupacao  int NOT NULL,
    area_ocupada    float NULL
)
go
ALTER TABLE Ocupacao_territorial
    ADD PRIMARY KEY (cod_tempo, cod_tipo_ocupacao)
go
/*
ACTION is CREATE Table Produto
*/
CREATE TABLE Produto (
    cod_produto      int NOT NULL,
    produto          char(18) NULL
)
go
ALTER TABLE Produto
    ADD PRIMARY KEY (cod_produto)
go
/*
ACTION is CREATE Table Principais_culturas
*/
CREATE TABLE Principais_culturas (
    cod_tempo      int NOT NULL,
    cod_produto      int NOT NULL,
    area_da_cultura  float NULL,
    toneladas       float NULL,
    valor_da_producao float NULL

```

```

)
go
ALTER TABLE Principais_culturas
    ADD PRIMARY KEY (cod_tempo, cod_produto)
go
/*
ACTION is CREATE Table Tipo_de_pescado
*/
CREATE TABLE Tipo_de_pescado (
    cod_tipo_pescado int NOT NULL,
    pescado          char(18) NULL
)
go
ALTER TABLE Tipo_de_pescado
    ADD PRIMARY KEY (cod_tipo_pescado)
go
/*
ACTION is CREATE Table Principais_Pescados
*/
CREATE TABLE Principais_Pescados (
    cod_tempo          int NOT NULL,
    cod_tipo_pescado  int NOT NULL,
    qtd_pescado       int NULL
)
go
ALTER TABLE Principais_Pescados
    ADD PRIMARY KEY (cod_tempo, cod_tipo_pescado)
go
/*
ACTION is CREATE Table Producao_de_la
*/

CREATE TABLE Producao_de_la (
    qtd_ovinos_tosquiados int NULL,
    cod_tempo             int NOT NULL,
    producao_de_la_kg     int NULL
)
go
ALTER TABLE Producao_de_la
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Producao_de_leite
*/
CREATE TABLE Producao_de_leite (
    cod_tempo          int NOT NULL,
    media_vacas_em_lactacao int NULL,
    produtividade_media_vaca_dia int NULL,
    producao_leite_dia int NULL
)
go
ALTER TABLE Producao_de_leite
    ADD PRIMARY KEY (cod_tempo)
go
/*

```

```
        ACTION is CREATE Table Rebanhos
    */
CREATE TABLE Rebanhos (
    cod_tempo          int NOT NULL,
    cod_tipo_rebanho  int NOT NULL,
    qtd_cabecas       int NULL
)
go
ALTER TABLE Rebanhos
    ADD PRIMARY KEY (cod_tempo, cod_tipo_rebanho)
go
ALTER TABLE Abate
    ADD FOREIGN KEY (cod_inspecao)
        REFERENCES Tipo_de_inspecao
go
ALTER TABLE Abate
    ADD FOREIGN KEY (cod_frigorifico)
        REFERENCES Frigorifico
go
ALTER TABLE Abate
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Abate
    ADD FOREIGN KEY (cod_tipo_rebanho)
        REFERENCES Tipo_de_rebanho
go
ALTER TABLE AtividadePesqueira
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Ocupacao_territorial
    ADD FOREIGN KEY (cod_tipo_ocupacao)
        REFERENCES Tipo_ocupacao_territorial
go
ALTER TABLE Ocupacao_territorial
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Principais_culturas
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Principais_culturas
    ADD FOREIGN KEY (cod_produto)
        REFERENCES Produto
go

ALTER TABLE Principais_Pescados
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Principais_Pescados
    ADD FOREIGN KEY (cod_tipo_pescado)
```



```

REFERENCES Tipo_de_pescado
go
ALTER TABLE Producao_de_la
ADD FOREIGN KEY (cod_tempo)
REFERENCES Tempo
go
ALTER TABLE Producao_de_leite
ADD FOREIGN KEY (cod_tempo)
REFERENCES Tempo
go
ALTER TABLE Rebanhos
ADD FOREIGN KEY (cod_tempo)
REFERENCES Tempo
go
ALTER TABLE Rebanhos
ADD FOREIGN KEY (cod_tipo_rebanho)
REFERENCES Tipo_de_rebanho
go

```

**Área Data Mart: Indústria, comércio e serviços**  
**Script: industria\_comercio\_servicos.sql**

```

/*
ACTION is CREATE Table Tipo_de_comercio
*/
CREATE TABLE Tipo_de_comercio (
cod_tipo_comercio int NOT NULL,
tipo_comercio char(30) NULL
)
go
ALTER TABLE Tipo_de_comercio
ADD PRIMARY KEY (cod_tipo_comercio)
go
/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
cod_tempo int NOT NULL,
ano int NULL
)
go
ALTER TABLE Tempo
ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table AbastecimentoComercial
*/
CREATE TABLE AbastecimentoComercial (
cod_tempo int NOT NULL,
cod_tipo_comercio int NOT NULL,
porcentagem_abastecimento float NULL
)
go
ALTER TABLE AbastecimentoComercial

```

```

        ADD PRIMARY KEY (cod_tempo, cod_tipo_comercio)
go
/*
ACTION is CREATE Table Tipo_de_Aplicacao
*/

CREATE TABLE Tipo_de_Aplicacao (
    cod_tipo_aplicacao int NOT NULL,
    tipo_aplicacao char(18) NULL
)
go
ALTER TABLE Tipo_de_Aplicacao
    ADD PRIMARY KEY (cod_tipo_aplicacao)
go
/*
ACTION is CREATE Table Aplicacoes
*/
CREATE TABLE Aplicacoes (
    cod_tempo int NOT NULL,
    cod_tipo_aplicacao int NOT NULL,
    valor_aplicacao float NULL
)
go
ALTER TABLE Aplicacoes
    ADD PRIMARY KEY (cod_tempo, cod_tipo_aplicacao)
go
/*
ACTION is CREATE Table Atividade_comercial
*/

CREATE TABLE Atividade_comercial (
    cod_atividade_comercial int NOT NULL,
    atividade_comercial varchar(200) NULL
)
go
ALTER TABLE Atividade_comercial
    ADD PRIMARY KEY (cod_atividade_comercial)
go
/*
ACTION is CREATE Table Bancos
*/

CREATE TABLE Bancos (
    cod_banco int NOT NULL,
    nome_banco char(30) NULL
)
go

ALTER TABLE Bancos
    ADD PRIMARY KEY (cod_banco)
go
/*
ACTION is CREATE Table Tipo_de_captacao
*/

```

```
CREATE TABLE Tipo_de_captacao (  
    cod_captacao    int NOT NULL,  
    captacao       char(18) NULL  
)  
go  
ALTER TABLE Tipo_de_captacao  
    ADD PRIMARY KEY (cod_captacao)  
go  
/*  
    ACTION is CREATE Table Captacao  
*/  
  
CREATE TABLE Captacao (  
    cod_tempo       int NOT NULL,  
    cod_captacao    int NOT NULL,  
    valor_captacao  float NULL  
)  
go  
ALTER TABLE Captacao  
    ADD PRIMARY KEY (cod_tempo, cod_captacao)  
go  
/*  
    ACTION is CREATE Table Galerias  
*/  
  
CREATE TABLE Galerias (  
    cod_galeria     int NOT NULL,  
    galeria         char(18) NULL  
)  
go  
  
ALTER TABLE Galerias  
    ADD PRIMARY KEY (cod_galeria)  
go  
  
/*  
    ACTION is CREATE Table Indicadores_Industria  
*/  
  
CREATE TABLE Indicadores_Industria (  
    cod_tempo       int NOT NULL,  
    qtd_engenhos_arroz int NULL,  
    qtd_sacas_ano_arroz int NULL,  
    qtd_empresas    int NULL,  
    qtd_empregos_gerados int NULL  
)  
go  
ALTER TABLE Indicadores_Industria  
    ADD PRIMARY KEY (cod_tempo)  
go  
/*  
    ACTION is CREATE Table Instituicoes_Financeiras
```

```

*/

CREATE TABLE Instituicoes_Financeiras (
    cod_tempo      int NOT NULL,
    cod_banco      int NOT NULL,
    qtd_agencias   int NULL
)
go
ALTER TABLE Instituicoes_Financeiras
    ADD PRIMARY KEY (cod_tempo, cod_banco)
go
/*
ACTION is CREATE Table Setores_da_economia
*/
CREATE TABLE Setores_da_economia (
    cod_setor      int NOT NULL,
    setor          char(20) NOT NULL
)
go
ALTER TABLE Setores_da_economia
    ADD PRIMARY KEY (cod_setor)
go
/*
ACTION is CREATE Table OcupacaoPop_porsetor
*/

CREATE TABLE OcupacaoPop_porsetor (
    cod_tempo      int NOT NULL,
    cod_setor      int NOT NULL,
    porcentagem_populacao float NULL
)
go
ALTER TABLE OcupacaoPop_porsetor
    ADD PRIMARY KEY (cod_tempo, cod_setor)
go
/*
ACTION is CREATE Table Quantidade_de_lojas
*/

CREATE TABLE Quantidade_de_lojas (
    cod_tempo      int NOT NULL,
    cod_galeria    int NOT NULL,
    qtdade_lojas   int NULL
)
go
ALTER TABLE Quantidade_de_lojas
    ADD PRIMARY KEY (cod_tempo, cod_galeria)
go
/*
ACTION is CREATE Table Unidades_Locais_por_atividade
*/
CREATE TABLE Unidades_Locais_por_atividade (
    cod_tempo      int NOT NULL,
    cod_atividade_comercial int NOT NULL,
    qtd_unidades   int NULL,

```

```
    qtd_pessoas_ocupadas int NULL
)
go
ALTER TABLE Unidades_Locais_por_atividade
    ADD PRIMARY KEY (cod_tempo, cod_atividade_comercial)
go
ALTER TABLE AbastecimentoComercial
    ADD FOREIGN KEY (cod_tipo_comercio)
        REFERENCES Tipo_de_comercio
go
ALTER TABLE AbastecimentoComercial
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Aplicacoes
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Aplicacoes
    ADD FOREIGN KEY (cod_tipo_aplicacao)
        REFERENCES Tipo_de_Aplicacao
go
ALTER TABLE Captacao
    ADD FOREIGN KEY (cod_captacao)
        REFERENCES Tipo_de_captacao
go
ALTER TABLE Captacao
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Indicadores_Industria
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Instituicoes_Financeiras
    ADD FOREIGN KEY (cod_banco)
        REFERENCES Bancos
go
ALTER TABLE Instituicoes_Financeiras
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE OcupacaoPop_porsetor
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE OcupacaoPop_porsetor
    ADD FOREIGN KEY (cod_setor)
        REFERENCES Setores_da_economia
go
ALTER TABLE Quantidade_de_lojas
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Quantidade_de_lojas
```

```

        ADD FOREIGN KEY (cod_galeria)
            REFERENCÉS Galerias
go
ALTER TABLE Unidades_Locais_por_atividade
    ADD FOREIGN KEY (cod_atividade_comercial)
        REFERENCÉS Atividade_comercial
go
ALTER TABLE Unidades_Locais_por_atividade
    ADD FOREIGN KEY (cod_tempo)
        REFERENCÉS Tempo
go

Área Data Mart: Infra-estrutura
Script: infraestrutura.sql

/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Abastecimento_SANEP
*/
CREATE TABLE Abastecimento_SANEP (
    cod_tempo      int NOT NULL,
    producao_agua_dia float NULL,
    qtd_estacoes_tratamento int NULL,
    qtd_reservatorios int NULL,
    populacao_atendida_porcen float NULL
)
go
ALTER TABLE Abastecimento_SANEP
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Tipo_Consumo
*/

CREATE TABLE Tipo_Consumo (
    cod_tipo_consumo int NOT NULL,
    tipo_consumo     char(30) NULL
)
go
ALTER TABLE Tipo_Consumo
    ADD PRIMARY KEY (cod_tipo_consumo)
go
/*
ACTION is CREATE Table Consumidores_Consumo_CEE
*/
CREATE TABLE Consumidores_Consumo_CEE (

```

```

        cod_tempo          int NOT NULL,
        cod_tipo_consumo   int NOT NULL,
        consumo            float NULL,
        qtd_consumidores   int NULL
    )
go
ALTER TABLE Consumidores_Consumo_CEE
    ADD PRIMARY KEY (cod_tempo, cod_tipo_consumo)
go
/*
ACTION is CREATE Table Esgoto
*/
CREATE TABLE Esgoto (
    cod_tempo          int NOT NULL,
    extensao_rede      float NULL,
    populacao_atendida_porcen float NULL,
    porcetagem_residuos_coletados float NULL
)
go
ALTER TABLE Esgoto
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Sub_Estacao
*/
CREATE TABLE Sub_Estacao (
    cod_sub_estacao     int NOT NULL,
    nome_sub_estacao    char(18) NULL,
    distribuicao_simounao int NULL,
    transmissao_simounao int NULL,
    regio_atendimento_se varchar(150) NULL
)
go
ALTER TABLE Sub_Estacao
    ADD PRIMARY KEY (cod_sub_estacao)
go
/*
ACTION is CREATE Table Estrutura_CEE
*/
CREATE TABLE Estrutura_CEE (
    cod_tempo          int NOT NULL,
    cod_sub_estacao     int NOT NULL,
    disponibilidade     float NULL,
    maior_demanda_nos_ultimos2anos_MVA float NULL,
    capacidade_MVA      float NULL
)
go
ALTER TABLE Estrutura_CEE
    ADD PRIMARY KEY (cod_tempo, cod_sub_estacao)
go
/*
ACTION is CREATE Table ETA
*/
CREATE TABLE ETA (

```

```

        cod_eta          int NOT NULL,
        nome_eta         char(50) NULL,
        reservatorios_que_abastece char(50) NULL,
        regio_atendimento_eta char(600) NULL
    )
go
ALTER TABLE ETA
    ADD PRIMARY KEY (cod_eta)
go
/*
ACTION is CREATE Table Estrutura_ETA
*/

CREATE TABLE Estrutura_ETA (
    cod_eta          int NOT NULL,
    cod_tempo        int NOT NULL,
    fornecimento_dia float NULL
)
go
ALTER TABLE Estrutura_ETA
    ADD PRIMARY KEY (cod_eta, cod_tempo)
go
/*
ACTION is CREATE Table Tipo_lixo
*/

CREATE TABLE Tipo_lixo (
    cod_tipo_lixo     int NOT NULL,
    tipo_lixo         char(50) NULL,
    destino            varchar(300) NULL
)
go
ALTER TABLE Tipo_lixo
    ADD PRIMARY KEY (cod_tipo_lixo)
go
/*
ACTION is CREATE Table Saneamento_Residuos_Solidos
*/

CREATE TABLE Saneamento_Residuos_Solidos (
    cod_tempo         int NOT NULL,
    cod_tipo_lixo     int NOT NULL,
    toneladas_dia     float NULL
)
go
ALTER TABLE Saneamento_Residuos_Solidos
    ADD PRIMARY KEY (cod_tempo, cod_tipo_lixo)
go

ALTER TABLE Abastecimento_SANEP
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Consumidores_Consumo_CEE
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

```



```

go
ALTER TABLE Consumidores_Consumo_CEE
    ADD FOREIGN KEY (cod_tipo_consumo)
        REFERENCES Tipo_Consumo

```

```

go
ALTER TABLE Esgoto
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

```

```

go
ALTER TABLE Estrutura_CEE
    ADD FOREIGN KEY (cod_sub_estacao)
        REFERENCES Sub_Estacao

```

```

go
ALTER TABLE Estrutura_CEE
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

```

```

go
ALTER TABLE Estrutura_ETA
    ADD FOREIGN KEY (cod_eta)
        REFERENCES ETA

```

```

go
ALTER TABLE Estrutura_ETA
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

```

```

go
ALTER TABLE Saneamento_Residuos_Solidos
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo

```

```

go
ALTER TABLE Saneamento_Residuos_Solidos
    ADD FOREIGN KEY (cod_tipo_lixo)
        REFERENCES Tipo_lixo

```

```

go
Área Data Mart: Telecomunicações
Script: telecomunicacoes.sql

```

```

/*
ACTION is CREATE Table Tempo
*/

```

```

CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)

```

```

go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)

```

```

go
/*
ACTION is CREATE Table Telefonias_fixa
*/

```

```

CREATE TABLE Telefonias_fixa (
    cod_tempo      int NOT NULL,
    qtd_telefones_individuais int NULL,
    qtd_telefones_publicos int NULL,

```

```

        qtd_telefones_surdos_mudos int NULL,
        qtd_telefones_adaptados_cadeiras_rodas int NULL,
        qtd_concessionarias int NULL
    )
go
ALTER TABLE Telefonica_fixa
    ADD PRIMARY KEY (cod_tempo)
go
/*
    ACTION is CREATE Table Telefonica_movel
*/
CREATE TABLE Telefonica_movel (
    cod_tempo int NOT NULL,
    qtd_telefones_individuais int NULL,
    qtd_empresas int NULL
)
go
ALTER TABLE Telefonica_movel
    ADD PRIMARY KEY (cod_tempo)
go
ALTER TABLE Telefonica_fixa
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Telefonica_movel
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go

```

**Área Data Mart: Transporte**  
**Script: transporte.sql**

```

/*
    ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo int NOT NULL,
    ano int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go
/*
    ACTION is CREATE Table CaminhoesCarretasde25ton
*/
CREATE TABLE CaminhoesCarretasde25ton (
    caminhoes_carreta_ano float NULL,
    caminhoes_carretas_mes float NULL,
    caminhoes_carretas_dia float NULL,
    total_cargas_movimentas_porto float NULL,
    cod_tempo int NOT NULL
)
go
ALTER TABLE CaminhoesCarretasde25ton

```

```

        ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Produto_transportado
*/
CREATE TABLE Produto_transportado (
    cod_produto_transportado int NOT NULL,
    produto                char(30) NULL
)
go
ALTER TABLE Produto_transportado
    ADD PRIMARY KEY (cod_produto_transportado)
go
/*
ACTION is CREATE Table Cargas_Transporte_Hidroviario
*/
CREATE TABLE Cargas_Transporte_Hidroviario (
    cod_produto_transportado int NOT NULL,
    qtd_descarregada        float NULL,
    cod_tempo                int NOT NULL,
    qtd_carregada           float NULL
)
go
ALTER TABLE Cargas_Transporte_Hidroviario
    ADD PRIMARY KEY (cod_produto_transportado, cod_tempo)
go
/*
ACTION is CREATE Table Ciclovias
*/
CREATE TABLE Ciclovias (
    cod_tempo                int NOT NULL,
    qtd_trabalhadores_usam_bicicleta float NULL,
    porcentagem_lesoes_em_acidentes float NULL,
    porcentagem_morte_em_acidente float NULL
)
go
ALTER TABLE Ciclovias
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Tipo_de_linha
*/
CREATE TABLE Tipo_de_linha (
    cod_tipo_linha          int NOT NULL,
    tipo_linha              char(50) NULL
)
go
ALTER TABLE Tipo_de_linha
    ADD PRIMARY KEY (cod_tipo_linha)
go
/*
ACTION is CREATE Table Estrutura_Terminal_Rodoviario
*/
CREATE TABLE Estrutura_Terminal_Rodoviario (
    cod_tempo                int NOT NULL,

```

```

        cod_tipo_linha    int NOT NULL,
        qtd_linhas       int NULL,
        qtd_empresas     int NULL
    )
go
ALTER TABLE Estrutura_Terminal_Rodoviario
    ADD PRIMARY KEY (cod_tempo, cod_tipo_linha)
go
/*
ACTION is CREATE Table Rotas_Ferrovias
*/
CREATE TABLE Rotas_Ferrovias (
    cod_rota_ferroviaria int NOT NULL,
    rota                 varchar(100) NULL
)
go
ALTER TABLE Rotas_Ferrovias
    ADD PRIMARY KEY (cod_rota_ferroviaria)
go
/*
ACTION is CREATE Table Estrutura_Transporte_Ferrovio
*/
CREATE TABLE Estrutura_Transporte_Ferrovio (
    cod_tempo            int NOT NULL,
    cod_rota_ferroviaria int NOT NULL,
    qtd_trem_dia         float NULL,
    tempo_de_percurso    float NULL,
    distancia            float NULL,
    velocidade_media     float NULL,
    capacidade           float NULL
)
go
ALTER TABLE Estrutura_Transporte_Ferrovio
    ADD PRIMARY KEY (cod_tempo, cod_rota_ferroviaria)
go
/*
ACTION is CREATE Table MediaMesVendaDePassagens
*/
CREATE TABLE MediaMesVendaDePassagens (
    media_mensal         float NULL,
    cod_tempo            int NOT NULL
)
go
ALTER TABLE MediaMesVendaDePassagens
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Tipo_de_pavimento
*/
CREATE TABLE Tipo_de_pavimento (
    cod_tipo_pavimento   int NOT NULL,
    tipo_pavimento       char(18) NULL
)
go

```

```

ALTER TABLE Tipo_de_pavimento
  ADD PRIMARY KEY (cod_tipo_pavimento)
go
/*
  ACTION is CREATE Table Pavimentacao
*/
CREATE TABLE Pavimentacao (
  cod_tempo      int NOT NULL,
  cod_tipo_pavimento int NOT NULL,
  distancia_pavimentada float NULL
)
go
ALTER TABLE Pavimentacao
  ADD PRIMARY KEY (cod_tempo, cod_tipo_pavimento)
go
/*
  ACTION is CREATE Table Rodovias
*/

CREATE TABLE Rodovias (
  cod_rodovia      int NOT NULL,
  rodovia          char(40) NULL
)
go
ALTER TABLE Rodovias
  ADD PRIMARY KEY (cod_rodovia)
go
/*
  ACTION is CREATE Table Tipo_de_Veiculo
*/
CREATE TABLE Tipo_de_Veiculo (
  cod_tipo_veiculo int NOT NULL,
  tipo_veiculo     char(18) NULL
)
go
ALTER TABLE Tipo_de_Veiculo
  ADD PRIMARY KEY (cod_tipo_veiculo)
go
/*
  ACTION is CREATE Table Tipo_frota
*/
CREATE TABLE Tipo_frota (
  cod_tipo_frota  int NOT NULL,
  tipo_frota     char(18) NULL
)
go
ALTER TABLE Tipo_frota
  ADD PRIMARY KEY (cod_tipo_frota)
go
/*
  ACTION is CREATE Table Trechos_Rodovias
*/
CREATE TABLE Trechos_Rodovias (
  cod_trecho      int NOT NULL,
  trecho          char(30) NULL

```

```

)
go
ALTER TABLE Trechos_Rodovias
    ADD PRIMARY KEY (cod_trecho)
go
/*
ACTION is CREATE Table Trafego_anual_mes
*/
CREATE TABLE Trafego_anual_mes (
    cod_tempo          int NOT NULL,
    cod_rodovia        int NOT NULL,
    cod_trecho         int NOT NULL,
    cod_tipo_veiculo   int NOT NULL,
    qtd_veiculos_ano   float NULL,
    qtd_veiculos_mes   float NULL
)
go
ALTER TABLE Trafego_anual_mes
    ADD PRIMARY KEY (cod_tempo, cod_rodovia, cod_trecho,
                    cod_tipo_veiculo)
go
/*
ACTION is CREATE Table Transporte_Coletivo
*/
CREATE TABLE Transporte_Coletivo (
    cod_tempo          int NOT NULL,
    qtd_empresas_perimetro_urbano int NULL,
    qtd_linhas_regulares int NULL,
    qtd_trabalhadores_rodoviario int NULL,
    qtd_empresas_meio_rural float NULL,
    distancia_media_deslocamento_urbano float NULL,
    tempo_medio_deslocamento_urbano float NULL
)
go
ALTER TABLE Transporte_Coletivo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table Transporte_urbano
*/
CREATE TABLE Transporte_urbano (
    cod_tempo          int NOT NULL,
    cod_tipo_frota     int NOT NULL,
    qtd_frota          float NULL
)
go
ALTER TABLE Transporte_urbano
    ADD PRIMARY KEY (cod_tempo, cod_tipo_frota)
go
ALTER TABLE CaminhoesCarretasde25ton
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE Cargas_Transporte_Hidroviario
    ADD FOREIGN KEY (cod_produto_transportado)

```

```
REFERENCES Produto_transportado
go
ALTER TABLE Cargas_Transporte_Hidroviario
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go
ALTER TABLE Ciclovias
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go
ALTER TABLE Estrutura_Terminal_Rodoviario
  ADD FOREIGN KEY (cod_tipo_linha)
  REFERENCES Tipo_de_linha
go
ALTER TABLE Estrutura_Terminal_Rodoviario
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go

ALTER TABLE Estrutura_Transporte_Ferrovuario
  ADD FOREIGN KEY (cod_rota_ferrovuaria)
  REFERENCES Rotas_Ferrovuarias
go

ALTER TABLE Estrutura_Transporte_Ferrovuario
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go

ALTER TABLE MediaMesVendaDePassagens
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go

ALTER TABLE Pavimentacao
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go

ALTER TABLE Pavimentacao
  ADD FOREIGN KEY (cod_tipo_pavimento)
  REFERENCES Tipo_de_pavimento
go

ALTER TABLE Trafego_anual_mes
  ADD FOREIGN KEY (cod_tempo)
  REFERENCES Tempo
go
```

```
ALTER TABLE Trafego_anual_mes
  ADD FOREIGN KEY (cod_tipo_veiculo)
    REFERENCES Tipo_de_Veiculo
go
```

```
ALTER TABLE Trafego_anual_mes
  ADD FOREIGN KEY (cod_trecho)
    REFERENCES Trechos_Rodovias
go
```

```
ALTER TABLE Trafego_anual_mes
  ADD FOREIGN KEY (cod_rodovia)
    REFERENCES Rodovias
go
```

```
ALTER TABLE Transporte_Coletivo
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
```

```
ALTER TABLE Transporte_urbano
  ADD FOREIGN KEY (cod_tipo_frota)
    REFERENCES Tipo_frota
go
```

```
ALTER TABLE Transporte_urbano
  ADD FOREIGN KEY (cod_tempo)
    REFERENCES Tempo
go
```

**Área Data Mart: Apoio ao investidor**  
**Script: apoio\_ao\_investidor.sql**

```
/*
ACTION is CREATE Table SistemaApoio
*/
```

```
CREATE TABLE SistemaApoio (
  cod_sistema      int NOT NULL,
  nome_sistema     char(18) NULL,
  descricao        varchar NULL
)
go
```

```
ALTER TABLE SistemaApoio
  ADD PRIMARY KEY (cod_sistema)
go
```

```
/*
```



```
ACTION is CREATE Table Tempo
*/

CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go

ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)
go

/*
ACTION is CREATE Table EstruturaApoioInvestidor
*/

CREATE TABLE EstruturaApoioInvestidor (
    cod_tempo      int NOT NULL,
    cod_sistema    int NOT NULL,
    qtd_municipios_abrange float NULL,
    qtd_empresas_conveniadas float NULL,
    qtd_atendimentos int NULL,
    qtd_trabalhadores_cadastrados int NULL,
    qtd_candidatos_concorrendo_intermediacao int NULL,
    qtd_candidatos_concorrendo_qualificacao int NULL,
    qtd_estudantes_cadastrados real NULL,
    qtd_estagios_nivel_medio real NULL,
    qtd_estagios_nivel_superior real NULL,
    qtd_estagios_comercio real NULL,
    qtd_estagios_industria real NULL,
    qtd_estagios_prestacao_servicos real NULL
)
go

ALTER TABLE EstruturaApoioInvestidor
    ADD PRIMARY KEY (cod_tempo, cod_sistema)
go

/*
ACTION is CREATE Table Rede
*/

CREATE TABLE Rede (
    cod_rede      int NOT NULL,
    nome_rede_cooperacao char(50) NULL
)
go

ALTER TABLE Rede
```

```

    ADD PRIMARY KEY (cod_rede)
go

/*
ACTION is CREATE Table RedesCooperacao
*/

CREATE TABLE RedesCooperacao (
    cod_tempo      int NOT NULL,
    cod_rede       int NOT NULL,
    qtd_empresas   int NULL
)
go

ALTER TABLE RedesCooperacao
    ADD PRIMARY KEY (cod_tempo, cod_rede)
go

ALTER TABLE EstruturaApoioInvestidor
    ADD FOREIGN KEY (cod_sistema)
        REFERENCES SistemaApoio
go

ALTER TABLE EstruturaApoioInvestidor
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go

ALTER TABLE RedesCooperacao
    ADD FOREIGN KEY (cod_rede)
        REFERENCES Rede
go

ALTER TABLE RedesCooperacao
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go

Área Data Mart: Estrutura do turismo
Script: turismo_estrutura.sql

/*
ACTION is CREATE Table Tempo
*/
CREATE TABLE Tempo (
    cod_tempo      int NOT NULL,
    ano            int NULL
)
go
ALTER TABLE Tempo
    ADD PRIMARY KEY (cod_tempo)

```

```
go
/*
ACTION is CREATE Table AgenciasTurismo
*/
CREATE TABLE AgenciasTurismo (
    cod_tempo          int NOT NULL,
    qtd_agencias_turismo_emissivo int NULL,
    qtd_agencias_turismo_receptivo int NULL,
    qtd_transportadoras_de_turismo int NULL
)
go
ALTER TABLE AgenciasTurismo
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table CTGs
*/
CREATE TABLE CTGs (
    cod_tempo          int NOT NULL,
    qtd_media_socios   int NULL,
    qtd_ctgs           int NULL
)
go
ALTER TABLE CTGs
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table LocalparaEvento
*/
CREATE TABLE LocalparaEvento (
    cod_local          int NOT NULL,
    nome_local_evento  varchar(100) NULL
)
go
ALTER TABLE LocalparaEvento
    ADD PRIMARY KEY (cod_local)
go
/*
ACTION is CREATE Table EstruturaAuditorios
*/
CREATE TABLE EstruturaAuditorios (
    cod_tempo          int NOT NULL,
    cod_local          int NOT NULL,
    capacidade         int NULL
)
go
ALTER TABLE EstruturaAuditorios
    ADD PRIMARY KEY (cod_tempo, cod_local)
go
/*
ACTION is CREATE Table EstruturaCentroEventos
*/
CREATE TABLE EstruturaCentroEventos (
    cod_local          int NOT NULL,
```

```

        cod_tempo      int NOT NULL,
        area_total    int NULL,
        area_coberta  int NULL,
        estacionamento int NULL,
        pavilhao_de_remates int NULL,
        pistas_julgamento int NULL,
        galpoes_fechados int NULL,
        capacidade_salao_festas int NULL,
        qtd_salas_de_aula int NULL
    )
go
ALTER TABLE EstruturaCentroEventos
    ADD PRIMARY KEY (cod_local, cod_tempo)
go
/*
ACTION is CREATE Table FeirasEventosAnuais
*/
CREATE TABLE FeirasEventosAnuais (
    cod_feira_ou_evento int NOT NULL,
    nome_feira_ou_evento varchar(100) NULL,
    mes_ocorrencia char(18) NULL
)
go
ALTER TABLE FeirasEventosAnuais
    ADD PRIMARY KEY (cod_feira_ou_evento)
go
/*
ACTION is CREATE Table Fenadoce
*/
CREATE TABLE Fenadoce (
    cod_tempo      int NOT NULL,
    media_movimentacao_negocios float NULL,
    area           float NULL,
    media_visitantes int NULL
)
go
ALTER TABLE Fenadoce
    ADD PRIMARY KEY (cod_tempo)
go
/*
ACTION is CREATE Table PublicoFeirasEventosAnuais
*/
CREATE TABLE PublicoFeirasEventosAnuais (
    cod_tempo      int NOT NULL,
    cod_feira_ou_evento int NOT NULL,
    estimativa_publico int NULL
)
go
ALTER TABLE PublicoFeirasEventosAnuais
    ADD PRIMARY KEY (cod_tempo, cod_feira_ou_evento)
go
/*
ACTION is CREATE Table TipoRedeHoteleira
*/
CREATE TABLE TipoRedeHoteleira (

```

```

        cod_tipo_rede_hot int NOT NULL,
        nome_rede         char(18) NULL
    )
go
ALTER TABLE TipoRedeHoteleira
    ADD PRIMARY KEY (cod_tipo_rede_hot)
go
/*
ACTION is CREATE Table RedeHoteleira
*/
CREATE TABLE RedeHoteleira (
    cod_tempo          int NOT NULL,
    cod_tipo_rede_hot int NOT NULL,
    qtd_total          int NULL,
    qtd_apto_cabana    int NULL,
    qtd_leitos         int NULL
)
go
ALTER TABLE RedeHoteleira
    ADD PRIMARY KEY (cod_tempo, cod_tipo_rede_hot)
go
/*
ACTION is CREATE Table TerminalRodoviario
*/
CREATE TABLE TerminalRodoviario (
    cod_tempo          int NOT NULL,
    media_pessoasdia  int NULL,
    capacidade_estacionamento int NULL,
    qtd_postos_seguranca int NULL
)
go
ALTER TABLE TerminalRodoviario
    ADD PRIMARY KEY (cod_tempo)
go
ALTER TABLE AgenciasTurismo
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE CTGs
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaAuditorios
    ADD FOREIGN KEY (cod_local)
        REFERENCES LocalparaEvento
go
ALTER TABLE EstruturaAuditorios
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaCentroEventos
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE EstruturaCentroEventos

```

```
        ADD FOREIGN KEY (cod_local)
            REFERENCES LocalparaEvento
go
ALTER TABLE Fenadoce
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE PublicoFeirasEventosAnuais
    ADD FOREIGN KEY (cod_feira_ou_evento)
        REFERENCES FeirasEventosAnuais
go
ALTER TABLE PublicoFeirasEventosAnuais
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE RedeHoteleira
    ADD FOREIGN KEY (cod_tipo_rede_hot)
        REFERENCES TipoRedeHoteleira
go
ALTER TABLE RedeHoteleira
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
ALTER TABLE TerminalRodoviario
    ADD FOREIGN KEY (cod_tempo)
        REFERENCES Tempo
go
```

## Apêndice C – Metadados dos cubos

### Área Data Mart: Comunicação

Cubo: EstruturaJornais

Medidas do cubo: qtd\_exemplaresdia (int)  
 qtd\_media\_assinantes (int)  
 qtd\_municipios\_abrangencia (int)  
 qtd\_leitores (int)  
 qtd\_leitor\_por\_jornal (int)

Tabela Fato: EstruturaJornais

Tabela Dimensão: Tempo  
 Jornais

Fonte dos dados: CDLR - 2005

--

Cubo: EstruturaRadios

Medidas do cubo: qtd\_emissoras\_am (int)  
 qtd\_emissoras\_fm (int)

Tabela Fato: EstruturaRadios

Tabela Dimensão: Tempo

Fonte dos dados: CDLR - 2005

--

Cubo: AbrangenciaRedesTV

Medidas do cubo: qtd\_municipios (int)  
 qtd\_habitantes (int)  
 qtd\_domicilios\_comtv (int)  
 qtd\_telespectadores\_potenciais (int)

Tabela Fato: AbrangenciaRedesTV

Tabela Dimensão: Tempo  
 RepetidorasTV

Fonte dos dados: RBS TV/PELOTAS – 2005  
 TV NATIVA PELOTAS – 2005  
 CDLR - 2005

### Área Data Mart: Educação

Cubo: IndicadoresEducacao

Medidas do cubo: taxa\_alfabetizacao (float)  
 alunos\_7a14anos\_naescola (float)  
 alunos\_15a17anos\_noensinomedio (float)  
 alunos\_18 a22anos\_noensinosuperior (float)  
 alunos\_25anosoumais\_nivelsuperior (float)  
 alunos\_25anosoumais\_commenos8anosdeestudo (float)  
 prof\_ensinofundamental\_comcursosuperior (float)

Tabela Fato: IndicadoresEducacao

Tabela Dimensão: Tempo  
 Local

Fonte dos dados: PNUD - 2003

--

Cubo: QuadroFuncional

Medidas do cubo: qtdade\_professores (int)  
 qtdade\_funcionarios (int)

Tabela Fato: QuadroFuncional  
 Tabela Dimensão: Tempo  
 RedeEnsino  
 Nivel

Fonte dos dados: ITEPA – 2002 e SME- 2005

--

Cubo: Matriculas  
 Medidas do cubo: quantidade (float)

Tabela Fato: Matriculas  
 Tabela Dimensão: Tempo  
 RedeEnsino  
 Nivel

Fonte dos dados: SME e Universidades - 2004

--

Cubo: EstabelecimentosEnsino  
 Medidas do cubo: quantidade (int)

Tabela Fato: EstabelecimentosEnsino  
 Tabela Dimensão: Tempo  
 RedeEnsino  
 Nivel

Fonte dos dados: IBGE – 2002 e SME - 2005

--

Cubo: EstruturaCursosUniversidadeseCentros  
 Medidas do cubo: qtd\_cursos (float)  
 qtd\_alunos (float)

Tabela Fato: EstabelecimentosEnsino  
 Tabela Dimensão: Tempo  
 NivelCursoProf  
 EstabelecimentosEnsinoSuperiorTecnologico

Fonte dos dados: CEFET – 2005  
 UFPEL – 2004/2005  
 UCPEL – 2004  
 ATLÂNTICO SUL – 2005  
 SENAC - 2005  
 5° Coordenadoria Regional de Educação - 2004

--

Cubo: RecursosHumanosUniversidadeseCentros  
 Medidas do cubo: qtd\_docentes\_especialistas (float)  
 qtd\_docentes\_mestres (float)  
 qtd\_docentes\_doutores (float)  
 qtd\_tecnico\_administrativos (float)  
 qtd\_funcionarios (float)  
 qtd\_total\_cursos (float)  
 qtd\_total\_alunos (float)  
 qtd\_total\_docentes (float)

Tabela Fato: RecursosHumanosUniversidadeseCentros  
 Tabela Dimensão: Tempo  
 NivelCursoProf  
 EstabelecimentosEnsinoSuperiorTecnologico

Fonte dos dados: CEFET – 2005



UFPEL – 2004/2005  
 UCPEL – 2004  
 ATLÂNTICO SUL – 2005  
 SENAC - 2005  
 5° Coordenadoria Regional de Educação - 2004

### Área Data Mart: Geografia e Qualidade de vida

Cubo: IDH

Medidas do cubo: valor (float)

Tabela Fato: IDH

Tabela Dimensão: Tempo  
Local

Fonte dos dados: PNUD - 2003

--

Cubo: Populacao

Medidas do cubo: quantidade (float)

Tabela Fato: Populacao

Tabela Dimensão: Tempo  
Zona

Fonte dos dados: IBGE - 2004

--

Cubo: DensidadeDemografica

Medidas do cubo: hab\_por\_kmqua\_porcen (float)

Tabela Fato: Populacao

Tabela Dimensão: Tempo  
Zona

Fonte dos dados: ITEPA e CDLR

--

Cubo: PrecipitacaoMedia

Medidas do cubo: quantidade (float)

Tabela Fato: PrecipitacaoMedia

Tabela Dimensão: Tempo

Fonte dos dados: Estação Agroclimatológica de Pelotas/EMBRAPA - 2005

--

Cubo: IDESE

Medidas do cubo: valor (float)

Tabela Fato: IDESE

Tabela Dimensão: Tempo  
Local

Fonte dos dados: PNUD – 2003

### Área Data Mart: Qualificação Profissional e Ação Social

Cubo: ServicosQualificacaoProfissional

Medidas do cubo: qtd\_cursos (int)  
qtd\_alunos (int)

Tabela Fato: ServicosQualificacaoProfissional

Tabela Dimensão: Tempo

InstituicaoQualificacao  
 Fonte dos dados: SENAI e SENAC -2004  
 --  
 Cubo: IndicadoresSociais  
 Medidas do cubo: mulheres\_de10a14anos\_comfilhos (float)  
 mulheres\_de15a17anos\_comfilhos (float)  
 crianças\_emfamilias\_renda\_inferior\_meiosalario (float)

Tabela Fato: IndicadoresSociais  
 Tabela Dimensão: Tempo  
 Local  
 Fonte dos dados: PNUD - 2003  
 --  
 Cubo: RendaMediaPerCapita  
 Medidas do cubo: renda\_do\_quinto\_mais\_pobre (float)  
 renda\_do\_quinto\_mais\_rico (float)  
 renda\_percapita\_mensal (float)

Tabela Fato: RendaMediaPerCapita  
 Tabela Dimensão: Tempo  
 Local  
 Fonte dos dados: PNUD- 2003  
 --  
 Cubo: EstruturaAcaoSocial  
 Medidas do cubo: qtd\_organizacoes (float)  
 meta\_qtd\_organizacoes (float)

Tabela Fato: EstruturaAcaoSocial  
 Tabela Dimensão: Tempo  
 PublicoAlvoAcaoSocial  
 TipoEntidade  
 Fonte dos dados: SMC -2005  
 --  
 Cubo: ServicosQualificacaoNivelTecnico  
 Medidas do cubo: qtd\_alunos (int)  
 carga\_horaria (float)

Tabela Fato: ServicosQualificacaoNivelTecnico  
 Tabela Dimensão: Tempo  
 CursosProfissionalizantes  
 Fonte dos dados: 5° Coordenadoria Regional de Educação – 2005

### Área Data Mart: Saúde

Cubo: RecursosSAMU  
 Medidas do cubo: qtdade\_medicos (int)  
 qtdade\_enfermeiros (int)  
 qtdade\_aux\_enfermagem (int)  
 qtdade\_motoristas (int)  
 qtdade\_radio\_operador (int)  
 qtdade\_telefonista (int)  
 qtdade\_ambulancias\_basicas (int)  
 qtdade\_ambulancias\_avancadas (int)

Tabela Fato: RecursosSAMU

Tabela Dimensão: Tempo  
 Fonte dos dados: PSM e www.pelotas.com.br2005

--

Cubo: IndicadoresSaude

Medidas do cubo: esperancadevidaao nascer (float)  
 mortalidadeate5anos (float)  
 mortalidadeate1ano (float)  
 medicospor1000hab (float)  
 idhlongevidade (float)  
 probabilidadedevidaate40anos (float)  
 probabilidadedevidaate60anos (float)  
 habitantespor medico (float)  
 habitantesporodontologo (float)  
 habitantesporleito hospitalar (float)

Tabela Fato: IndicadoresSaude

Tabela Dimensão: Tempo  
 Local

Fonte dos dados: ITEPA e SMS – 2004 e PNUD - 2003

--

Cubo: DistribuicaoSuprimentos

Medidas do cubo: qtdade\_distribuida (float)  
 qtdade\_usuarios (float)

Tabela Fato: DistribuicaoSuprimentos

Tabela Dimensão: Tempo  
 TipoSuprimento  
 UnidadesSaudeMunicipal

Fonte dos dados: Secretaria Municipal de Saúde

--

Cubo: EstruturaFisica

Medidas do cubo: area (float)  
 qtd\_leitos\_geral (float)  
 qtd\_uti (float)  
 media\_internacao\_mes (int)  
 capacidade\_internacao\_mes (int)  
 qtd\_leitosSUS (int)  
 percentual\_paciente\_atendido (float)

Tabela Fato: EstruturaFisica

Tabela Dimensão: Tempo  
 UnidadesSaudeMunicipal

Fonte dos dados: ITEPA e UCPEL – 2005

### Área Data Mart: Segurança

Cubo: Incidencias

Medidas do cubo: delitos (int)  
 furtos (int)  
 homicidios (int)  
 lesoes\_corporais (int)  
 roubos (int)

Tabela Fato: Incidencias

Tabela Dimensão: Tempo  
 Local

Fonte dos dados: Secretaria da Justiça e de Segurança Pública –RS - 2004

--

Cubo: TaxaMediaRiscoPor100MilHab

Medidas do cubo: taxa (float)

Tabela Fato: TaxaMediaRiscoPor100MilHab

Tabela Dimensão: Tempo

Local

Fonte dos dados: Secretaria da Justiça e de Segurança Pública –RS - 2004

--

Cubo: EstruturaFrotaSeguranca

Medidas do cubo: qtd\_frota (float)

Tabela Fato: EstruturaFrotaSeguranca

Tabela Dimensão: Tempo

TipoFrota

Fonte dos dados: POLICIA CIVIL DE PELOTAS – 2005

CORPO DE BOMBEIROS DE PELOTAS - 2005

BRIGADA MILITAR - 2005

--

Cubo: EfetivoSeguranca

Medidas do cubo: qtd\_efetivos (float)

Tabela Fato: EfetivoSeguranca

Tabela Dimensão: Tempo

EntidadeSeguranca

Fonte dos dados: POLICIA CIVIL DE PELOTAS – 2005

CORPO DE BOMBEIROS DE PELOTAS - 2005

BRIGADA MILITAR - 2005

--

Cubo: OcorrenciasSeguranca

Medidas do cubo: qtd\_ocorrencias (float)

Tabela Fato: OcorrenciasSeguranca

Tabela Dimensão: Tempo

EntidadeSeguranca

TipodeOcorrencia

Fonte dos dados: SMTT - 2005

--

Cubo: Incidencia\_por100mil\_habit

Medidas do cubo: delitos (float)

furtos (int)

homicidios (int)

lesoes\_corporais (int)

roubos (int)

Tabela Fato: Incidencia\_por100mil\_habit

Tabela Dimensão: Tempo

Local

Fonte dos dados: Secretaria da Justiça e de Segurança Pública – RS - 2004

### Área Data Mart: Drenagem

Cubo: Arborização urbana

Medidas do cubo: area\_verde\_por\_habitante (float)

recomendacao (float)

Tabela Fato: Arborizacao\_urbana

Tabela Dimensão: Tempo  
 Fonte dos dados: SANEP - 2005

--

Cubo: Drenagem Urbana  
 Medidas do cubo: litro\_seg (float)  
 Tabela Fato: Drenagem\_urbana  
 Tabela dimensão: Tempo  
                   Casas\_de\_Bomba  
 Fonte dos dados: SANEP – 2005

### Área Data Mart: Economia

Cubo: Renda  
 Medidas do cubo: massa\_anual (float)  
                   renda\_per\_capita (float)

Tabela Fato: Renda  
 Tabela Dimensão: Tempo

Fonte dos dados: ITEPA

--

Cubo: Desemprego  
 Medidas do cubo: estimativa\_desemprego (float)  
 Tabela Fato: Desemprego  
 Tabela Dimensão: Tempo  
 Fonte dos dados: SINE - 2005

--

Cubo: Rendimento mensal de pessoas com 10 anos ou mais  
 Medidas do cubo: sem\_rendimento  
                   ate1salario  
                   maisde1a2salarios (float)  
                   maisde2a3salarios (float)  
                   maisde3a5salarios (float)  
                   maisde5a10salarios (float)  
                   maisde10a20salarios (float)  
                   maisde20 (int)  
 Tabela Fato: Rendimento\_mensal\_pessoas\_com\_10\_ou\_mais\_anos  
 Tabela Dimensão: Tempo  
 Fonte dos dados: IBGE - 2000

--

Cubo: Trabalhadores  
 Medidas do cubo: qtd\_trabalhador\_com\_carteira (float)  
                   qtd\_militares\_e\_estatuarios (float)  
                   qtd\_trabalhador\_sem\_carteira (float)  
                   qtd\_empregadores (float)  
                   qtd\_trabalhadores\_conta\_propria (float)  
                   qtd\_outros\_trabalhadores (float)  
 Tabela Fato: Trabalhadores  
 Tabela Dimensão: Tempo  
 Fonte dos dados: IBGE - 2000

--

Cubo: População ativa por setor  
 Medidas do cubo: pea\_por\_setor (float)  
 Tabela Fato: PEA\_por\_setor

Tabela Dimensão: Tempo  
Setores\_da\_economia  
Fonte dos dados: RAIS - 2003

--

Cubo: PIB por setor  
Medidas do cubo: pib\_por\_setor (float)  
Tabela Fato: PIBporsetor  
Tabela Dimensão: Tempo  
Setores\_da\_economia  
Local  
Fonte dos dados: IBGE - 2002

--

Cubo: PIB per capita  
Medidas do cubo: pib\_per\_capita(float)  
Tabela Fato: PIBpercapita  
Tabela Dimensão: Tempo  
Local  
Fonte dos dados: IBGE - 2002

--

Cubo: Potencial de consumo  
Medidas do cubo: consumo\_per\_capita (float)  
Tabela Fato: Potencial\_de\_consumo  
Tabela Dimensão: Tempo  
Local  
Fonte dos dados: FEE- IPEADATA- ITEPA - 2002

--

Cubo: Atividade Empreendedora  
Medidas do cubo: porcentagem\_atividade\_empreendedora (int)  
Tabela Fato: Atividade\_empreendedora  
Tabela Dimensão: Tempo  
Local  
Fonte dos dados: JORNAL ZERO HORA (EDIÇÃO 05/10/2005)

--

Cubo: Demandas Trabalhistas  
Medidas do cubo: demanda (int)  
Tabela Fato: Demandas\_Trabalhistas  
Tabela Dimensão: Tempo  
Local  
Fonte dos dados: TRIBUNAL REGIONAL DO TRABALHO – 4º REGIÃO - 2005

--

Cubo: Importação e Exportação  
Medidas do cubo: valor\_em\_exportacoes (float)  
valor\_em\_importacao (float)  
Tabela Fato: Importacao\_e\_Exportacao  
Tabela Dimensão: Tempo  
Fonte dos dados: AGÊNCIA DE DESENVOLVIMENTO DA LAGOA MIRIM/UFPEL-2005

--

Cubo: Estrutura Fundiaria  
Medidas do cubo: qtd\_de\_propriedades (int)  
area\_total (float)  
Tabela Fato: Estrutura\_Fundiaria  
Tabela Dimensão: Tempo  
Area\_rural  
Fonte dos dados: INCRA - 1997

--

Cubo: População Economicamente Ativa

Medidas do cubo: qtd\_pessoas\_de10oumais\_anos (float)  
qtd\_pessoas\_pea (float)

Tabela Fato: PEA

Tabela Dimensão: Tempo  
Zona

Fonte dos dados: IBGE/SENSO - 2000

--

Cubo: Benefícios Previdenciários

Medidas do cubo: qtd\_beneficios (float)

Tabela Fato: Beneficios\_Previdenciarios

Tabela Dimensão: Tempo  
Zona

Fonte dos dados: INSS – 2005

**Área Data Mart: Economia do setor primário**

Cubo: Produção de leite

Medidas do cubo: media\_vacas\_em\_lactacao (int)  
Produtividade\_media\_vaca\_dia (int)  
Producao\_leite\_dia (int)

Tabela Fato: Producao\_de\_leite

Tabela Dimensão: Tempo

Fonte dos dados: EMATER – 2005 e SMDR - 2004

--

Cubo: Produção de lã

Medidas do cubo: qtd\_ovinos\_tosquiados (int)  
producao\_de\_la\_kg (int)

Tabela Fato: Producao\_de\_la

Tabela Dimensão: Tempo

Fonte dos dados: EMATER – 2005 e SMDR - 2004

--

Cubo: Ocupação territorial

Medidas do cubo: area\_ocupada (float)

Tabela Fato: Ocupacao\_territorial

Tabela Dimensão: Tempo  
Tipo\_ocupacao\_territorial

Fonte dos dados: EMATER - 2005

--

Cubo: Principais culturas

Medidas do cubo: area\_da\_cultura (float)  
toneladas (float)

Tabela Fato: Principais\_culturas

Tabela Dimensão: Tempo  
Produto

Fonte dos dados: EMATER – 2005 e ITEPA - 2004

--

Cubo: Abate

Medidas do cubo: qtd\_cabecas\_abate (int)

Tabela Fato: Abate  
Tabela Dimensão: Tempo  
Tipo\_de\_inspecao  
Tipo\_de\_rebanho  
Frigorífico

Fonte dos dados: SMDR – 2005  
Coordenadoria Regional de Pelotas da Secretaria Estadual da Agricultura  
– 2005  
Ministério da Agricultura, Unidade de Pelotas - 2005

--

Cubo: Rebanhos  
Medidas do cubo: qtd\_cabecas (int)

Tabela Fato: Rebanhos  
Tabela Dimensão: Tempo  
Tipo\_de\_rebanho

Fonte dos dados: EMATER – 2005 e SMDR - 2004

--

Cubo: Atividade Pesqueira  
Medidas do cubo: qtd\_habitantes\_Z3 (int)  
qtd\_pescadores ( int)  
qtd\_embarcacoes (int)

Tabela Fato: AtividadePesqueira  
Tabela Dimensão: Tempo  
Tipo\_de\_rebanho

Fonte dos dados: ITEPA e SMDR - 2005

--

Cubo: Principais pescados  
Medidas do cubo: qtd\_pescado (int)

Tabela Fato: Principais\_Pescados  
Tabela Dimensão: Tempo  
Tipo\_de\_pescado

Fonte dos dados: ITEPA e SMDR - 2005

### **Área Data Mart: Indústria, comércio e serviços**

Cubo: Indicadores Industria  
Medidas do cubo: qtd\_engenhos\_arroz (int)  
qtd\_sacas\_ano\_arroz (int)  
qtd\_empresas (int)  
qtd\_empregos\_gerados (int)

Tabela Fato: Indicadores\_Industria  
Tabela Dimensão: Tempo

Fonte dos dados: ITEPA – 2002  
SINDAPEL  
IBGE - 2002



--

Cubo: Instituicoes Financeiras  
 Medidas do cubo: qtd\_agencias (int)

Tabela Fato: Instituicoes\_Financeiras  
 Tabela Dimensão: Tempo  
 Fonte dos dados: BANCO DO BRASIL

--

Cubo: Aplicacoes  
 Medidas do cubo: valor\_aplicacao (int)

Tabela Fato: Aplicacoes  
 Tabela Dimensão: Tempo  
 Tipo\_de\_aplicacao  
 Fonte dos dados: SISBACEN – MAIO 2005

--

Cubo: Captacao  
 Medidas do cubo: valor\_captacao (int)

Tabela Fato: Captacao  
 Tabela Dimensão: Tempo  
 Tipo\_de\_captacao  
 Fonte dos dados: SISBACEN – Maio 2005

--

Cubo: Abastecimento Comercial  
 Medidas do cubo: porcentagem\_abastecimento (float)

Tabela Fato: Abastecimento\_Comercial  
 Tabela Dimensão: Tempo  
 Tipo\_de\_comercio  
 Fonte dos dados:

--

Cubo: Ocupacao da Populacao por setor  
 Medidas do cubo: porcentagem\_populacao (float)

Tabela Fato: OcupacaoPop\_porsetor  
 Tabela Dimensão: Tempo  
 Setores\_da\_economia  
 Fonte dos dados:

--

Cubo: Unidades locais por atividade  
 Medidas do cubo: qtd\_unidades (int)  
 qtd\_pessoas\_ocupadas (int)  
 Tabela Fato: Unidades\_Locais\_por\_atividade  
 Tabela Dimensão: Tempo  
 Atividade\_comercial  
 Fonte dos dados:

--

Cubo: Quantidade de lojas  
 Medidas do cubo: qtdade\_lojas (int)

Tabela Fato: Quantidade\_de\_lojas  
 Tabela Dimensão: Tempo  
 Galerias  
 Fonte dos dados: SINDILOJAS - 2005

## Área Data Mart: Infra-estrutura

Cubo: Abastecimento SANEP

Medidas do cubo: producao\_agua\_dia (float)  
 qtd\_estacoes\_tratamento (int)  
 qtd\_reservatorios (int)  
 populacao\_atendida\_porcen (float)

Tabela Fato: Abastecimento\_SANEP

Tabela Dimensão: Tempo  
 Fonte dos dados: SANEP - 2005

--

Cubo: Esgoto

Medidas do cubo: extensao\_rede\_km (float)  
 populacao\_atendida\_porcen (float)  
 porcentagem\_residuos\_coletados(float)

Tabela Fato: Esgoto

Tabela Dimensão: Tempo  
 Fonte dos dados: SANEP - 2005

--

Cubo: Consumidores e Consumo CEE

Medidas do cubo: consumo (float)  
 qtd\_consumidores (int)

Tabela Fato: Consumidores\_Consumo\_CEE

Tabela Dimensão: Tempo  
 Tipo\_Consumo

Fonte dos dados: CEEE - 2005

--

Cubo: Estrutura CEEE

Medidas do cubo: disponibilidade (float)  
 maior\_demanda\_nos\_ultimos2anos (float)  
 capacidade (float)

Tabela Fato: Estrutura\_CEEE

Tabela Dimensão: Tempo  
 Sub\_estacao

Fonte dos dados: CEEE - 2005

--

Cubo: Estrutura ETA

Medidas do cubo: fornecimento\_dia (float)

Tabela Fato: Estrutura\_ETA

Tabela Dimensão: Tempo  
 ETA

Fonte dos dados: SANEP - 2005

--

Cubo: Saneamento Resíduos Sólidos

Medidas do cubo: toneladas\_dia (float)

Tabela Fato: Saneamento\_Residuos\_Solidos

Tabela Dimensão: Tempo  
 Tipo\_lixo

Fonte dos dados: SANEP - 2005

### Área Data Mart: Telecomunicações

Cubo: Telefonia móvel

Medidas do cubo: qtd\_telefones\_individuais (int)  
qtd\_empresas (int)

Tabela Fato: Telefonia\_movel

Tabela dimensão: Tempo

Fonte dos dados: ANATEL - 2005

--

Cubo: Telefonia fixa

Medidas do cubo: qtd\_telefones\_individuais (int)  
qtd\_telefones\_publicos (int)  
qtd\_telefones\_surdos\_mudos (int)  
qtd\_telefones\_adaptados\_cadeiras\_rodas (int)  
qtd\_concessionarias (int)

Tabela Fato: Telefonia\_fixa

Tabela dimensão: Tempo

Fonte dos dados: ANATEL – 2005

### Área Data Mart: Transporte

Cubo: Ciclovias

Medidas do cubo: qtd\_trabalhadores\_usam\_bicicleta (float)  
porcentagem\_lesoes\_em\_acidentes (float)  
porcentagem\_mortes\_em\_acidentes (float)

Tabela Fato: Ciclovias

Tabela dimensão: Tempo

Fonte dos dados: SMTT - 2005

--

Cubo: Pavimentação

Medidas do cubo: distancia\_pavimentada(float)

Tabela Fato: Ciclovias

Tabela Dimensão: Tempo  
Tipo\_de\_pavimento

Fonte dos dados: Secretaria Municipal de Habitação e Obras - 2005

--

Cubo: Cargas Transporte Hidroviário

Medidas do cubo: qtd\_descarregada (float)  
qtd\_carregada (float)

Tabela Fato: Cargas\_Transporte\_Hidroviario

Tabela Dimensão: Tempo  
Produto\_transportado

Fonte dos dados: ST/RS – SPH/Pel - 2005

--

Cubo: Caminhões Carretas de 25 ton

Medidas do cubo: caminhoes\_carreta\_ano (float)  
caminhoes\_carretas\_mes (float)  
caminhoes\_carretas\_dia (float)  
total\_cargas\_movimentadas\_porto (float)

Tabela Fato: CaminhoesCarretasde25ton

Tabela Dimensão: Tempo

Fonte dos dados: ST/RS – SPH/Pel - 2005

--

Cubo: Estrutura Transporte Ferroviário

Medidas do cubo: qtd\_trem\_dia (float)  
tempo\_de\_percurso (float)  
distancia(float)  
velocidade\_media (float)  
capacidade (float)

Tabela Fato: Estrutura\_Transporte\_Ferrovuario

Tabela Dimensão: Tempo  
Rotas\_Ferrovitarias

Fonte dos dados: América Latina Logística - 2004

--

Cubo: Transporte urbano

Medidas do cubo: qtd\_frota (float)  
Tabela Fato: Transporte\_urbano  
Tabela Dimensão: Tempo  
Tipo\_frota

Fonte dos dados: ETERPEL-2005

IBGE-2003

STT-2005

Sindicato dos Trabalhadores em Transportes Rodoviários de Pelotas

--

Cubo: Transporte Coletivo

Medidas do cubo: qtd\_empresas\_perimetro\_urbano (int)  
qtd\_linhas\_regulares (int)  
qtd\_trabalhadores\_rodoviario (int)  
distancia\_media\_deslocamento\_urbano (float)  
tempo\_medio\_deslocamento\_urbano (float)  
qtd\_empresas\_meio\_rural (float)

Tabela Fato: Transporte\_coletivo

Fonte dos dados: ETERPEL-2005

IBGE-2003

STT-2005

Sindicato dos Trabalhadores em Transportes Rodoviários de Pelotas

--

Cubo Media Mês Venda de Passagens

Medidas do cubo: media\_mensal (float)  
Tabela Fato: MediaMesVendaDePassagens  
Tabela Dimensão: Tempo  
Fonte dos dados: ETERPEL

--

Cubo: Estrutura Terminal Rodoviario

Medidas do cubo: qtd\_linhas (int)  
qtd\_empresas (int)  
Tabela Fato: Estrutura\_Terminal\_Rodoviario  
Tabela Dimensão: Tempo  
Tipo\_de\_linha

Fonte dos dados: ETERPEL

### Área Data Mart: Apoio ao investidor

Cubo: Redes de cooperacao

Medidas do cubo: qtd\_empresas (int)

Tabela Fato: RedesCooperacao  
 Tabela Dimensão: Tempo  
 Rede

Fonte dos dados: EDR, UCPEL - 2005

--

Cubo: Estrutura Apoio Investidor

Medidas do cubo: qtd\_municipios\_abrange (int)  
 qtd\_empresas\_conveniadas (int)  
 qtd\_atendimentos (int)  
 qtd\_trabalhadores\_cadastrados (int)  
 qtd\_candidatos\_concorrendo\_intermediacao (int)  
 qtd\_candidatos\_concorrendo\_qualificacao (int)  
 qtd\_estudantes\_cadastrados (real)  
 qtd\_estagios\_nivel\_medio (real)  
 qtd\_estagios\_nivel\_superior (real)  
 qtd\_estagios\_comercio (real)  
 qtd\_estagios\_industria (real)  
 qtd\_estagios\_prestacao\_servicos (real)

Tabela Fato: EstruturaApoioInvestidor

Tabela Dimensão: Tempo  
 SistemaApoio

Fonte dos dados: SEBRAE – 2005  
 SINE PELOTAS – 2005  
 CIEE PELOTAS – 2005

### Área Data Mart: Estrutura do turismo

Cubo: Agencias Turismo

Medidas do cubo: qtidade\_lojas (int)

Tabela Fato: AgenciasTurismo

Tabela Dimensão: Tempo

Fonte dos dados:

--

Cubo: CTGs

Medidas do cubo: qtd\_media\_socios (int)  
 qtd\_ctgs (int)

Tabela Fato: CTGs

Tabela Dimensão: Tempo

Fonte dos dados: STE e 26º Reg. Tradicionalista - 2005

--

Cubo: Fenadoce

Medidas do cubo: media\_movimentacao\_negocios (float)  
 Area (float)  
 Media\_visitantes (int)

Tabela Fato: Fenadoce

Tabela Dimensão: Tempo

Fonte dos dados: www.fenadoce.com.br2005

--

Cubo: Rede Hoteleira

Medidas do cubo: qtd\_total (int)  
 qtd\_apto\_cabana (int)  
 qtd\_leitos (int)

Tabela Fato: RedeHoteleira  
 Tabela Dimensão: Tempo  
 TipoRedeHoteleira

Fonte dos dados: STE - 2005  
 --

Cubo: Publico nas feiras ou eventos anuais  
 Medidas do cubo: estimativa\_publico (int)  
 Tabela Fato: PublicoFeirasEventosAnuais  
 Tabela Dimensão: Tempo  
 FeirasEventosAnuais

Fonte dos dados: STE- 2005  
 --

Cubo: Terminal Rodoviario  
 Medidas do cubo: media\_pessoasdia (int)  
 capacidade\_estacionamento (int)  
 qtd\_postos\_seguranca (int)

Tabela Fato: TerminalRodoviario  
 Tabela Dimensão: Tempo

Fonte dos dados: ETERPEL - 2005  
 --

Cubo: Estrutura Centro Eventos  
 Medidas do cubo: area\_total (int)  
 area\_coberta (int)  
 estacionamento (int)  
 pavilhao\_de\_remates (int)  
 pistas\_julgamento (int)  
 galpoes\_fechados (int)  
 capacidade\_salao\_festas (int)  
 qtd\_salas\_de\_aula (int)

Tabela Fato: EstruturaCentroEventos  
 Tabela Dimensão: Tempo  
 LocalparaEvento

Fonte dos dados: STE- 2005  
 --

Cubo: Estrutura Auditorios  
 Medidas do cubo: capacidade (int)

Tabela Fato: EstruturaAuditorios  
 Tabela Dimensão: Tempo  
 LocalparaEvento

Fonte dos dados: STE e CDLR - 2005