



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JAILSON CAMARGO DE LIMA

BANCO DE DADOS MULTIDIMENSIONAL

Assis
2011



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JAILSON CAMARGO DE LIMA

BANCO DE DADOS MULTIDIMENSIONAL

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora.

Orientador: Profº Domingos de Carvalho Villela Junior

Área de Concentração: Informática

Assis
2011

FICHA CATALOGRÁFICA

LIMA, Jailson Camargo.

Banco de Dados Multidimensional / Jailson Camargo de Lima. Fundação Educacional do Município de Assis – FEMA – Assis, 2011.

52 p.

Orientador: Profº Domingos de Carvalho Villela Junior.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Modelagem de Dados. 2. Dados. 3. Formatação de Estruturas

CDD: 001.6
Biblioteca da FEMA

BANCO DE DADOS MULTIDIMENSIONAL

JAILSON CAMARGO DE LIMA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof^o Domingos de Carvalho Villela Junior

Analisador: Prof^o Dr^o Luiz Ricardo Begosso

Assis
2011

DEDICATÓRIA

Dedico este trabalho aos meus pais, **Jair Ribeiro de Lima e Vilma Camargo Pereira** por sempre me apoiarem e incentivarem. A razão da minha existência.

AGRADECIMENTOS

Primeiramente agradeço a **Deus** por tudo que tem feito por mim, pois sem ele nada seria possível.

Ao meu Orientador **Profº Domingos de Carvalho Villela Junior**, pela orientação durante a realização deste trabalho. Muito obrigado pelo apoio, disponibilidade e conhecimento a mim ensinado.

A **Fundação Educacional do Município de Assis – FEMA**, por me capacitar durante o curso para desenvolver este trabalho.

A todos os professores do Curso de Ciência da Computação da FEMA, pelos valiosos ensinamentos passados durante a minha jornada acadêmica.

Ao Sr. Carlos Eugênio Canton Tozoni, por me ajudar e me apoiar durante minha caminhada, tanto no aspecto profissional quanto nos ensinamentos e conselhos a mim passados.

Aos meus Pais pela paciência, estímulo, entusiasmo e compreensão nas inúmeras vezes que me apoiaram nas horas mais difíceis e complicadas.

Aos meus **Amigos** que tanto prezo, pelo apoio e amizade e companheirismo.

A todos que direta ou indiretamente, contribuíram para a realização deste trabalho.

“Wo ein Wille ist, ist auch ein Weg [...]”
Deutsch Sprichwort

RESUMO

A Modelagem de Dados é um instrumento de grande importância, na formatação de estruturas capazes de serem implantadas e compreendidas pelos Sistemas Gerenciadores de Banco de Dados. Constitui a camada fundamental de processos operacionais das empresas. Com a crescente evolução e competitividade em buscar diferentes áreas de negócios e conseqüentemente tomada de decisões, as informações e estruturas são diferenciadas. Desta forma é necessário adotar outra estrutura de Modelagem de Dados. A proposta do trabalho foi pesquisar e compreender os fundamentos conceituais de Banco de Dados Multidimensional que utiliza os procedimentos e comportamentos de tal Modelo, de maneira que possa exportar para ferramentas de consultas, como o Microsoft Office Excel.

Palavras-Chave: Modelagem de Dados; Dados; Formatação de Estruturas.

ABSTRACT

The Data modeling is an important instrument, in the formatting structures can be implemented and understood by Data Base Management Systems. Constitute the fundamental layer of companies processes. Increasing development and competitiveness in searching different business areas and consequently decision-making, information and structures are different. So, it is necessary to adopt another structure of Data Modeling. The purpose of this, was search and understand fundamental concepts of Multidimensional Database using behavior of this model, it can be export to tools and queries, as Microsoft Office Excel.

Keywords: Data Modeling; Data; Formatting Structures.

LISTA DE ILUSTRAÇÕES

Figura 1 – Instrução Select.....	18
Figura 2 – Integridade Referencial.....	20
Figura 3 – Instrução Integridade Referencial entre as tabelas Produto e Venda.....	20
Figura 4 - Exemplo Chave Primária da tabela Produto.....	22
Figura 5 - Exemplo Criação da chave primária da tabela Produto.....	22
Figura 6 - Exemplo Chave Primária Composta das tabelas Funcionário e Dependente.....	23
Figura 7 - Exemplo Criação da chave primária composta da tabela Produto e Dependente.....	23
Figura 8 - Exemplo Chave Estrangeira – Conjunto da entidade Funcionário relacionado com a entidade Dependente.....	24
Figura 9 - Exemplo Criação da chave estrangeira do conjunto da entidade Funcionário relacionada com a entidade Dependente.....	24
Figura 10 - Representação de um Sistema Gerenciador de Banco de Dados.....	27
Figura 11 – Representação de um Cubo de Dados.....	30
Figura 12 – Representação de Dimensões.....	30
Figura 13 - Representação Dimensional – Valor Pontual.....	32
Figura 14 - Representação Dimensional – Valores em Planos Fatias Slicing.....	32
Figura 15 - Representação Dimensional – Valores em Cubos de Dados Dicing.	33
Figura 16 - Representação Dimensional – Rotação de Planos.....	33
Figura 17 - Representação dos Principais Comandos Dimensionais.....	34

Figura 18 - Representação dos Principais Comandos Dimensionais, Drill-Down, Drill-Up e Drill-Across.....	35
Figura 19 - Representação dos principais comandos dimensionais, Drill Through.....	36
Figura 20 - Elementos Básicos do Data Warehouse.....	41
Figura 21 - Representação do Esquema Estrela.....	43
Figura 22 - Modelo Relacional Sistema de Vendas da rede de lojas.....	46
Figura 23 – Modelo Dimensional Sistema de Vendas da rede de lojas.....	47

LISTA DE TABELAS

Tabela 1 – Comparação dos Modelos Relacional e Dimensional29

Tabela 2 – Comparação dos dados de natureza Operacional e Informacional37

LISTA DE ABREVIATURAS E SIGLAS

DDL	Data Definition Language
DER	Diagrama Entidade Relacionamento
DM	Data Mart
DML	Data Manipulation Language
DW	Data Warehouse
ETL	Extract, Transformation and Load
OLAP	On Line Analytical Processing
OLTP	On Line Transaction Processing
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structure Query Language

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVOS.....	15
1.2 JUSTIFICATIVA.....	16
1.3 ESTRUTURA DO TRABALHO.....	16
2 FUNDAMENTAÇÃO CONCEITUAL.....	17
2.1 SISTEMA DE BANCO DE DADOS.....	17
2.1.1 Linguagem e manipulação de dados.....	18
2.1.2 A álgebra relacional.....	18
2.1.3 Cálculo relacional de tupla.....	19
2.1.4 Integridade referencial.....	20
3 MODELO RELACIONAL	21
4 SISTEMA GERENCIADOR DE BANCO DE DADOS.....	26
5 MODELO DIMENSIONAL.....	28
5.1 CUBO.....	29
5.1.2 DIMENSÕES.....	30
5.1.3 MEDIDAS.....	31
5.1.4 OPERAÇÕES DIMENSIONAIS.....	31
5.1.5 Operadores Dimensionais Drill-Down e Drill-up.....	33
5.1.6 Operadores Dimensionais Drill-Across.....	34
5.1.7 Operadores Dimensionais Drill-Through.....	35
6 DIFERENÇA ENTRE DADOS OPERACIONAIS E DADOS INFORMACIONAIS.....	37
7 DATA WAREHOUSE E DATA MART.....	38
8 GRANULARIDADE.....	42
9 ESQUEMA DE ESTRELA (STAR SCHEMA).....	43
10 BANCO DE DADOS ORACLE.....	44
11 DESENVOLVIMENTO DO PROJETO.....	45
12 CONSIDERAÇÕES FINAIS.....	48
REFERENCIAS.....	49
ANEXO.....	51

1 – INTRODUÇÃO

A utilização de Sistemas Gerenciadores de Banco de Dados Relacional (SGBDR) é fundamental e de grande importância. Os Dados precisam ser armazenados e recuperados em um curto intervalo de tempo, em situações que exigem alto desempenho, acessos concorrentes e simultâneos diretamente baseados em dados históricos. As estruturas de dados e o próprio mecanismo utilizado pelo SGBDR voltado as aplicações analíticas que são grandes volumes de dados com baixa taxa de atualização, requer outro modelo de Dados, sendo mais eficiente em relação às situações acima citadas, desta forma é necessário utilizar a Modelagem Multidimensional. Sistemas de ambiente operacional que apoiam os usuários nas funções cotidianas são chamados de OLTP (On Line Transaction Processing), o principal objetivo é o maior número de transações com menor tempo de processamento. Os sistemas OLTP não são flexíveis, em relação às quantidades de relatórios e consultas, limitação esta por seu modelo de dados. Sistemas de grande volume de dados e com suporte a decisão, onde as consultas são complexas é necessário adotar os denominados sistemas OLAP (On Line Analytical Processing), ou seja, permite aos usuários de alto nível, tais como, gerentes e alta gerencia navegar entre os dados com maior facilidade, visualizando os dados multidimensionalmente.

1.1 - OBJETIVO

O objetivo deste trabalho é pesquisar e compreender os fundamentos conceituais de Banco de Dados Multidimensional que utiliza procedimentos e comportamentos distintos do Modelo Relacional. Bases de dados Multidimensionais fornecem subsídio para a realização de análises e consultas sobre os dados, permitindo realizar cruzamentos e agrupamentos de informações, resultando em consultas mais rápidas do que o modelo Relacional.

1.2 - JUSTIFICATIVA

Mostrar a diferença do Modelo Relacional, utilizando o Modelo de Banco de Dados Multidimensional, o quão satisfatório será o desempenho das consultas e a maneira de se analisar através de Dados Históricos, a partir das dimensões dos Dados e a forma de Modelagem para com o Banco de Dados.

1.3 - ESTRUTURA DO TRABALHO

- Introdução
- Fundamentação conceitual
- Desenvolvimento do projeto
- Conclusão
- Referencias

2 - FUNDAMENTAÇÃO CONCEITUAL

A elaboração do estudo de caso será responsável pela aplicação dos conceitos obtidos, para tal, é necessária uma pesquisa com a finalidade de adquirir toda a fundamentação conceitual que o trabalho requer.

2.1 - SISTEMA DE BANCO DE DADOS

Um Sistema de Banco de Dados permite armazenar dados de uma organização, sendo o controle destes dados, é essencial para a organização. Um Sistema de banco de dados fornece uma linguagem denominada definição, que especifica o esquema do banco de dados e outra linguagem de manipulação para as consultas e atualizações.

- Linguagem de Definição de Dados (DDL): permite ser utilizada para especificar uma determinada propriedade adicional de dados. Onde é especificada a estrutura de armazenamento e métodos pelo sistema de banco de dados por uma série de instruções em um tipo especial DDL denotada armazenamento e definição de dados. Estas instruções definem os detalhes de implementação dos esquemas de banco de dados que geralmente é oculto para o usuário. Os valores dos dados armazenados precisam satisfazer as restrições de consistência, ou seja, são validações que especificam e verificam as restrições toda vez que o banco de dados é atualizado desta forma mantendo a integridade dos dados.

- Linguagem de Manipulação (DML): permite aos usuários acessar ou manipular os dados conforme são organizados. Sendo possível efetuar as instruções de (Select, Insert, Update e Delete), ou seja, Seleção, Inserção, Modificação e Deleção.

Os sistemas de banco de dados são projetos para gerenciar grandes blocos de informações. Esses blocos de informações não existem isolados. Eles são parte da operação de alguma empresa cujo produto final pode ser informações do banco de dados ou pode ser algum dispositivo ou serviço para o qual o banco de dados desempenha apenas um papel de apoio (SILBERSCHATZ et al, 2006).

2.1.1 – Linguagem de manipulação de dados

A Linguagem de manipulação é uma linguagem na qual um determinado usuário obtém informações do banco de dados, ou seja, realmente efetua a manipulação dos dados. Esta linguagem é de alto nível, sendo formada por dois tipos: procedurais e não procedurais. Na Linguagem procedural o usuário deve informar ao sistema uma sequência de operações no banco para obter o resultado. Na linguagem não procedural o usuário descreve a informação desejada sem fornecer um determinado procedimento para a obtenção do resultado. A Álgebra relacional é procedural, enquanto o cálculo relacional de uma tupla e o cálculo relacional de um domínio não são procedurais (SILBERSCHATZ et al, 1999). Podemos dizer que a Linguagem SQL não é procedural, pois como entrada são várias tabelas ou podendo também ser apenas uma, retornando uma única tabela. A operação select seleciona tuplas que satisfazem um determinado predicado, a letra grega minúscula sigma (σ) determina a seleção. O predicado aparece subscrito a sigma, o argumento da relação é dado entre parênteses, seguido a sigma. Desta forma, para selecionar tuplas da relação cliente, cujo cliente é o 'Jailson', podemos escrever:

```
SELECT NOME
FROM CLIENTE
WHERE NOME = 'JAILSON';
```

Figura 1 Instrução Select (In: Feito por Jailson Camargo de Lima)

σ NOME = "Jailson" (CLIENTE)

2.1.2 – A álgebra relacional

A álgebra relacional é uma linguagem de consulta procedural, consistindo em conjuntos de operações tendo como entrada uma ou duas relações e resultando em

uma nova relação. As operações na álgebra relacional são select, project, union, set difference, cartesian product e rename. Existindo entre outras namely, set intersection, natural join, division e assignment. Os termos das operações fundamentais são as operações select, project e rename são denominadas operações primárias operando em uma única relação, logo as três relações operam em um par de relação e são chamadas de operações binárias.

Exemplo operação project, a operação project é primária e retorna o argumento da relação, deixando de lado certos atributos. Sendo que a relação é um conjunto quaisquer de linhas, as duplicidades são eliminadas. A projeção é denotada pela letra grega pi (π). No subscrito π , listamos os atributos que deseja trazer no resultado. O argumento da relação é entre parênteses.

π VENDA_COD_VENDA, QTDE ^(VENDA).

2.1.3 – Cálculo relacional de tupla

A álgebra relacional define uma sequencia de procedimentos que geram respostas para nossas consultas. O cálculo de uma tupla relacional por outro é uma linguagem de consulta não procedural, permitindo a descrição da consulta desejada sem especificar os procedimentos para a obtenção dessas informações.

A expressão é:

$$\{ t \mid P(t) \}$$

Significado -> O conjunto de todas as tuplas t tal que o predicado P é verdadeiro para t.

Seguindo a notação de t[A] para denotar os valores das tuplas t sobre os atributos de A, usamos t e r para denotar que a tupla t é uma relação r.

Exemplo Consulta:

$$\{ t \mid \epsilon \text{ PRODUTO } t[\text{PRECO_VENDA}] > 2000 \}$$

A expressão acima é para consultar a DESCRICAO, o COD_PRODUTO e o PRECO_VENDA do produto acima > 2000 reais.

2.1.4 – Integridade referencial

A Integridade referencial permite que em um banco de dados não possa conter nenhum valor de chave estrangeira que não corresponda ao valor da chave primária nelas relacionadas. A integridade referencial pode ser especificada durante a construção das estruturas de dados de uma tabela, por meio da instrução “CREATE TABLE”, onde são acrescentadas as cláusulas “PRIMARY KEY” (chave primária), “FOREIGN KEY” (chave estrangeira) e “UNIQUE KEY” (chave candidata). As modificações no banco de dados ocorrem por meio de inserções, atualizações e deleções podendo causar violações de integridade referencial e certamente a inconsistência dos dados.

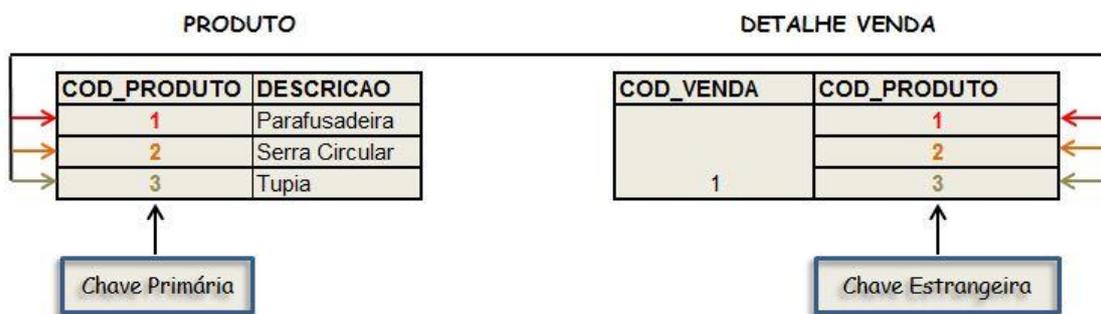


Figura 2 Integridade Referencial (In: Feito por Jailson Camargo de Lima)

A Figura 2 representa a seguinte instrução:

```
CREATE TABLE PRODUTO (
  COD_PRODUTO INTEGER NOT NULL,
  DESCRICAO VARCHAR2(40),
  PRIMARY KEY(COD_PRODUTO));

CREATE TABLE DET_VENDA (
  COD_VENDA INTEGER NOT NULL,
  COD_PRODUTO INTEGER NOT NULL,
  PRIMARY KEY(COD_VENDA, COD_PRODUTO),
  FOREIGN KEY(COD_PRODUTO)
  REFERENCES PRODUTO(COD_PRODUTO));
```

Figura 3 Instrução Integridade Referencial entre as tabelas Produto e Detalhe Venda (In: Feito por Jailson Camargo de Lima)

3 - MODELO RELACIONAL

O Modelo Relacional foi criado por Edgar Frank Codd, no ano de 1970. Segundo SILBERSCHATZ et al (2006), embora academicamente interessante, inicialmente o modelo relacional não era usado na prática, devido as suas desvantagens de desempenho; os bancos de dados relacionais não podiam se igualar aos bancos de dados de rede e hierárquicos existentes. Isto mudou com o System R, um projeto inovador da IBM Research que desenvolvia técnicas para a construção de um sistema de banco de dados relacional eficiente. O protótipo totalmente funcional do System R levou ao primeiro produto de banco de dados relacional da IBM, o SQL/DS. Os primeiros sistemas de banco de dados relacionais comerciais, como o IBM DB2, Oracle, Ingres e DEC Rdb, desempenharam um papel fundamental no desenvolvimento de técnicas para o processamento eficiente de consultas. No Início da década 1980, os bancos de dados relacionais haviam se tornado competitivos com os sistemas de banco de dados hierárquicos e de rede. A Abordagem relacional está baseada no principio de que as informações em uma base de dados podem ser consideradas como relações matemáticas e que estão representadas de maneira uniforme, através do uso de tabelas bidimensionais. O modelo relacional é o conjunto que representa os dados, visto segundo um conjunto de tabelas e suas operações sobre elas executadas por linguagens que manipulam a álgebra relacional (teoria dos conjuntos), não sendo procedurais, ou seja, os conjuntos são manipulados de uma só vez (MACHADO e ABREU, 2004).

O modelo relacional é também definido, como um conjunto de tuplas (linhas), que por sua vez são listas de valores, todas as tuplas de uma relação tem o mesmo numero de componentes. Cabeçalho de colunas é chamado de atributos, que representa o significado de cada componente das tuplas (MOLINA et al, 2001).

O modelo relacional é composto pelos seguintes conceitos:

- Chave Primária – Primary Key (PK): Designa o conceito de item de busca, ou seja, como as entidades de um dado conjunto de entidades e os relacionamentos dentro de um conjunto de relacionamentos podem ser identificadas. Segundo SILBERSCHATZ et al (2006) a

chave primária é um conjunto de um ou mais atributos que, tomados coletivamente, nos permitem identificar de maneira única uma entidade em um conjunto de entidades. A chave primária é uma propriedade do conjunto de entidades e não de uma propriedade individual.

COD_PRODUTO	DESCRICAO
1	Parafusadeira
2	Serra Circular
3	Tupia

↑

Chave Primária

Figura 4 Exemplo Chave Primária da tabela Produto (In: Feito por Jailson Camargo de Lima)

Na Figura 4, a coluna “código” é uma chave primária, que não pode ser repetida, ou seja, a chave deve – ser única não tendo os mesmos valores em seu atributo chave.

A Figura 4 representa a seguinte instrução:

```
CREATE TABLE PRODUTO (
  COD_PRODUTO INTEGER NOT NULL,
  DESCRICAO VARCHAR2(40),
  PRIMARY KEY(COD_PRODUTO));
```

Figura 5 Exemplo Criação da chave primária da tabela Produto (In: Feito por Jailson Camargo de Lima)

- Chave Primária Composta: As chaves primárias compostas servem para a utilização de conjuntos de entidades que servirão para a

movimentação, com o mapeamento de cardinalidade muitos para muitos (M: N).

COD_FUNCIONARIO	COD_DEPEN	NOME	DT_NASCIMENTO
1	1	Mariana	05/06/1988
1	2	Marcela	10/09/1990
2	1	Lais	03/10/1991
2	2	Aline	08/11/1995

Chave Primária Composta

Figura 6 Exemplo Chave Primária Composta das tabelas Funcionário e Dependente
(In: Feito por Jailson Camargo de Lima)

Na Figura 6, a chave primária composta permitirá a inclusão de “n” dependentes quanto for necessário. A chave primária composta identifica que o Funcionário “1” possui dois filhos, o “1” (Mariana) e o “2” (Marcela), bem como o Funcionário “2” possui também dois filhos o “1” (Lais) e o “2” (Aline). Mesmo com códigos de dependentes similares como a Mariana e a Laís que estão representados pelo COD_DEPEN = ‘1’ não irá ter duplicidade de entidades, sendo que cada um dos dependentes pertencem a funcionários com códigos diferentes.

A Figura 6 representa a seguinte instrução:

```
CREATE TABLE FUNCIONARIO (
  COD_FUNCIONARIO INTEGER NOT NULL,
  COD_DEPEN INTEGER NOT NULL,
  NOME VARCHAR2(40),
  DT_NASCIMENTO DATE,
  PRIMARY KEY(COD_FUNCIONARIO, COD_DEPEN));
```

Figura 7 Exemplo Criação da chave primária composta da tabela Produto e Dependente (In: Feito por Jailson Camargo de Lima)

- Chave Estrangeira – Foreign Key (FK): As chaves estrangeiras são de vital importância no modelo relacional, pois são elos entre as tabelas. Quando dizemos que duas tabelas estão relacionadas através de atributos comuns, provavelmente esta coluna em uma das

tabelas é chave primária. Na outra tabela, este atributo irá caracterizar o que é denominado de chave estrangeira, desta maneira, formando um relacionamento de ligação lógica (MACHADO e ABREU, 2004). Com isso, é garantida a regra de Integridade Referencial, em que um banco de dados não pode conter valores de chaves estrangeiras que não correspondam com os valores das chaves primárias, assim referenciadas.

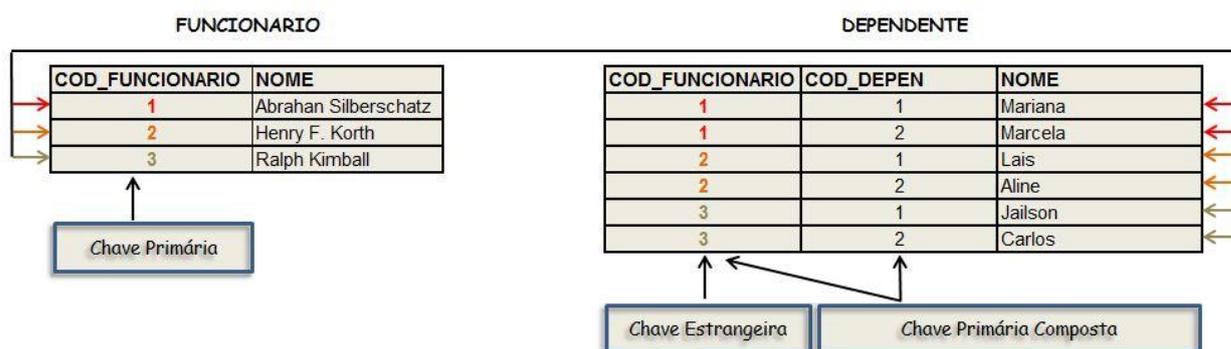


Figura 8 Exemplo Chave Estrangeira – Conjunto da entidade Funcionário relacionado com a entidade Dependente. (In: Feito por Jailson Camargo de Lima)

Na Figura 8, mostra a criação da chave estrangeira da entidade Dependente que estará sendo referenciada a chave primária da entidade Funcionário. A chave estrangeira deverá ser definida como uma restrição através da Foreign Key que, no entanto referenciará a chave primária do conjunto de entidades a qual está se relacionando.

A Figura 8 é representada pela seguinte instrução:

```
CREATE TABLE FUNCIONARIO (
  COD_FUNCIONARIO INTEGER NOT NULL,
  NOME VARCHAR2(40),
  PRIMARY KEY(COD_FUNCIONARIO));

CREATE TABLE DEPENDENTE (
  COD_FUNCIONARIO INTEGER NOT NULL,
  COD_DEPEN INTEGER NOT NULL,
  NOME VARCHAR2(40),
  PRIMARY KEY (COD_FUNCIONARIO,COD_DEPEN),
  FOREIGN KEY(COD_FUNCIONARIO)
  REFERENCES FUNCIONARIO(COD_FUNCIONARIO));
```

Figura 9 Exemplo Criação da chave estrangeira do conjunto da entidade funcionário relacionada com a entidade Dependente. (In: Feito por Jailson Camargo de Lima)

- Índice: Visa otimizar a recuperação de uma informação, via um método de acesso. Seu objetivo principal está relacionado com o desempenho. Uma chave pode ser utilizada como índice, mas um índice não é uma chave. A forma da criação do índice depende do ambiente relacional, onde são armazenados os ponteiros em uma tabela de modo ascendente ou descendente.

Normalização é o processo que substitui um conjunto de entidades e relacionamentos por outro, o qual se apresenta “purificado” em relação aos problemas de atualização (inclusão, alteração e exclusão), causando certos problemas, tais como, grupos repetitivos (atributos multivalorados) de dados, dependências parciais em relação a uma chave concatenada, redundâncias de dados desnecessários, perdas acidentais de informações, dificuldade na apresentação de fatos da realidade (MACHADO e ABREU, 2004).

Segundo SILBERSCHATZ et al (2006), normalização tem como principal objetivo gerar um conjunto de esquemas de relação que permita armazenar informações sem redundância desnecessária, ao mesmo tempo permitindo recuperar informações facilmente. A necessidade da normalização é a Repetição de Informações e Incapacidade de representar certas informações.

4 - SISTEMA GERENCIADOR DE BANCO DE DADOS – SGBD

Um Sistema Gerenciador de Banco de Dados cria e gerencia grandes quantidades de dados de forma eficiente, permitindo que esses dados persistam longos espaços de tempo com segurança, ou seja, um gerenciador de armazenamento, recuperação e exclusão, propiciando segurança e integridade dos dados.

O SGBD fornece os seguintes recursos:

- Armazenamento persistente: Admite o armazenamento de quantidades muito grandes de dados que existem independentemente de processos que estejam utilizando esses dados.
- Proporciona flexibilidade, como estruturas de dados que permitem o acesso eficiente a grandes quantidades de dados (MOLINA et al, 2001).
- Interface de programação: Permite ao usuário acessar e modificar dados através de uma linguagem de consulta, manipulando dados de forma mais complexas, que a leitura e gravação de arquivos e um sistema de arquivos (MOLINA et al, 2001).
- Gerenciamento de Transações: Admite acessos concorrentes a dados, ou seja, acessos simultâneos por muitos processos de forma distinta ao mesmo tempo. Para evitar consequências indesejáveis de acesso simultâneo, o SGBD admite:

O isolamento de dados: a aparência de que as transações são executadas uma de cada vez.

A atomicidade, o requisito de que as transações sejam executadas completamente ou não sejam executadas.

A resiliência, a capacidade de recuperar de muitos tipos de falhas ou erros (MOLINA et al, 2001). Desta forma um Sistema Gerenciador de Banco de Dados deve fornecer os recursos indispensáveis, mantendo a integridade dos dados, restringindo acessos não autorizados, caso ocorra erros inesperados a recuperação

e backup dos dados. Propiciar o controle de concorrência, suportar múltiplas interfaces e manter a representação de relações complexas entre os dados. Estes recursos são de fundamental importância a um Sistema Gerenciador de Banco de Dados.

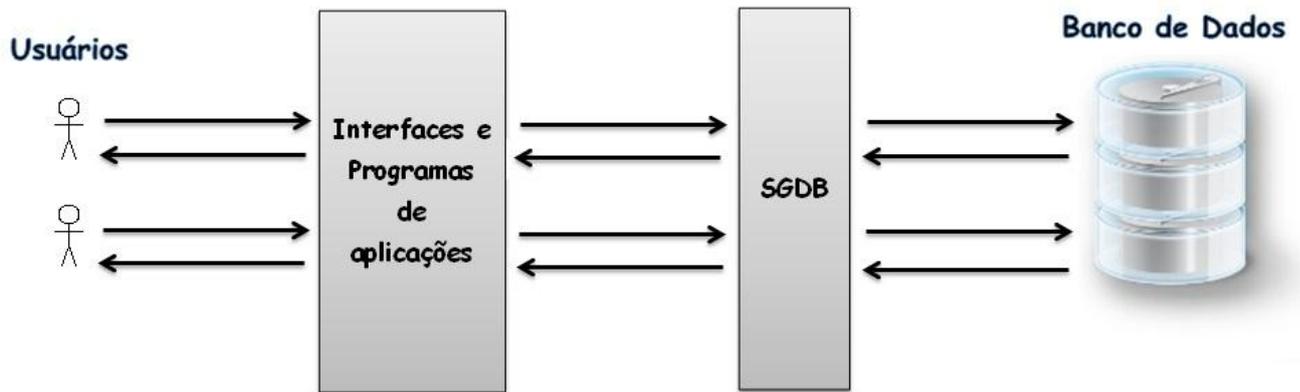


Figura 10 Representação de um Sistema Gerenciador de Banco de Dados (In: Feito por Jailson Camargo de Lima)

5 - MODELO DIMENSIONAL

A estrutura dimensional modifica a ordem de distribuição de campos, por entre as tabelas, permitindo uma formatação estrutural voltada para os muitos pontos de entradas específicos, as denotadas dimensões e para os dados granulares os chamados fatos. De tal forma, a estrutura dimensional dos dados estará em uma forma estelar, onde varias tabelas de entradas estarão se relacionando com algumas poucas tabelas de informações, criando – se uma notação mais sintética, legível e objetiva (BARBIERI, 2001).

No Modelo Dimensional é claro e direto os elementos que se precisa para buscar as informações sobre fatos via dimensões de referências, ao contrário da malha relacional, próprias do modelo anterior, onde não existe estrutura especifica de entrada.

Embora o modelo Dimensional seja mais claro e direto do que o modelo Relacional, o mesmo pode se tornar mais complexo, na medida em que novas extensões forem sendo criadas. Relacionamentos M:N entre tabelas Fato e Dimensão, estruturas recursivas nas dimensões poderão tornar o modelo mais complexo.

Abaixo, as diferenças entre os modelos, Dimensional e o Relacional.

Modelo Dimensional	Modelo Relacional
O Padrão de Estrutura é mais Fácil	Modelo Complexo
Anterior ao MER, ano de 1960.	Ênfase nos Bancos Relacionais, ano de 1970.
Tabelas Fato e tabelas Dimensão	Tabelas que representam Dados e relacionamentos
Tabela Fato é núcleo – normalizada	Tabelas são comumente normalizadas
Tabela Dimensão é ponto de entrada	Tabelas são indistintamente acessadas e de filtro inicial
Tabela Dimensão opcionalmente normalizada	Tabelas são comumente normalizadas
Modelo mais fácil para fazer “Join”	Maior dificuldade de “Join” por possuir maior número de tabelas
Fácil Entendimento para usuários não especializados	Maior dificuldade para usuários não especializados

Tabela 1 Comparação de Modelos, Dimensional e Relacional (BARBIERI, 2001).

5.1 – Cubo

O cubo é uma estrutura de dados multidimensional para analisar e recuperar dados. As hierarquias em dimensões e fórmulas são de primeira classe, ou seja, um cubo não duplica suas definições, o cubo pode ter qualquer n dimensões, também denotado de hipercubo ao invés de cubo, sendo que uma coleção de cubos é relacionada ao banco de dados multidimensional ou armazém de dados (Data Warehouse) multidimensional. O cubo é composto de células que são identificáveis em cada uma das suas interseções de dimensões.

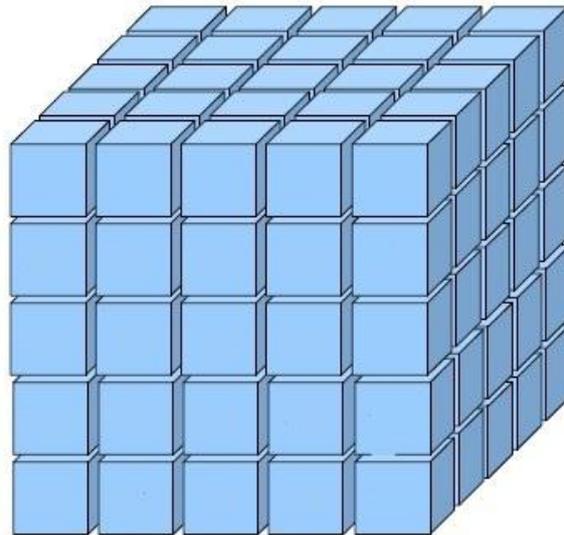


Figura 11 Representação de um Cubo de Dados (In: Feito por Jailson Camargo de Lima)

5.1.2 - Dimensões

A notação de dimensão é essencial para distinguir o conceito de banco de dados multidimensional que são a seleção de dados e o agrupamento de dados para o desejado nível de detalhamento. Uma dimensão é organizada em composição de números de níveis, representando um nível de detalhamento que seja o interessante para a análise. As instâncias são valores de dimensão ou também membros de dimensão, pertencendo a um nível em particular.

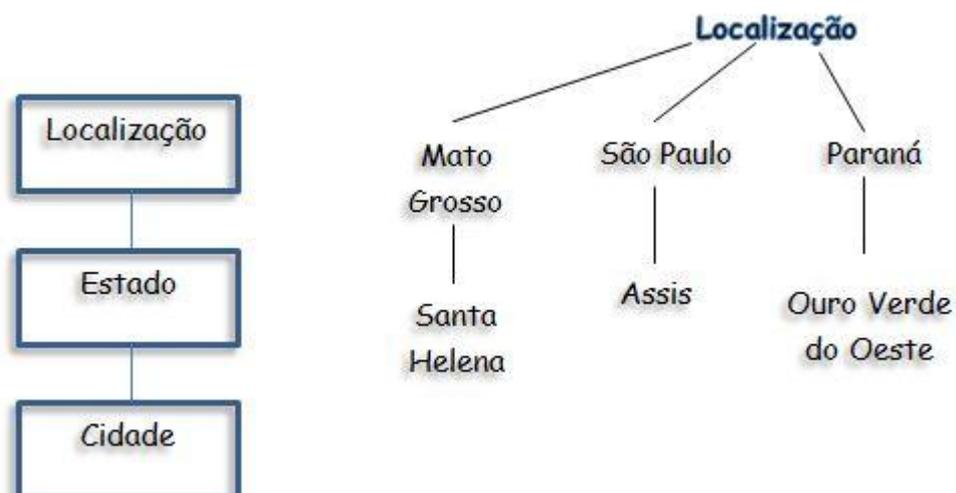


Figura 12 Representação de dimensões (In: Feito por Jailson Camargo de Lima)

5.1.3 – Medidas

As medidas têm duas componentes uma chamada numérica e a outra fórmula que são utilizadas para efetuar combinações e medições. Assumindo valores distintos para diferentes combinações de dimensão. E estas medições estão localizadas na tabela fato.

5.1.4 - Operadores dimensionais

As dimensões Produto, Tempo e Região, o valor pontual chamado de ponto representa a interseção de valores fato aos três eixos chamados de dimensões.

Na Figura 13, segue as três dimensões, porém, podendo um modelo ser visto com representação de n dimensões. O Plano chamado slicing é o conceito de plano ou fatia, evidenciando os Produtos PR1 a PRn que venderam na Região Rx, nos períodos entre P1 e Pn (BARBIERI, 2001).

No Plano Dicing, é o conceito de Rotação estando relacionado com a mudança dos eixos das dimensões, permitindo as transformações de visualizações dos respectivos dados. Lembrando que a visualização dos planos exigirá pivotamento (rotação) para que as diversas dimensões sejam exibidas. De início, em uma planilha bidimensional, os valores relativos à Produto e Região, com dimensão Tempo sendo fixa. Logo, por pivotamento, fixando a Região a mostrar os valores dos Produtos ao longo do Tempo, permitindo a realizar os pivotamentos conforme necessários a dinâmica das informações. O conceito de cubo é usado em Data Mart, ou de parte, do Data Warehouse conforme extração a certas necessidades de aplicações, representando assim, o conjunto de dados de dimensões e de fatos, certamente extraídos de um universo maior, de tal objetivo a cada necessidade (BARBIERI, 2001).

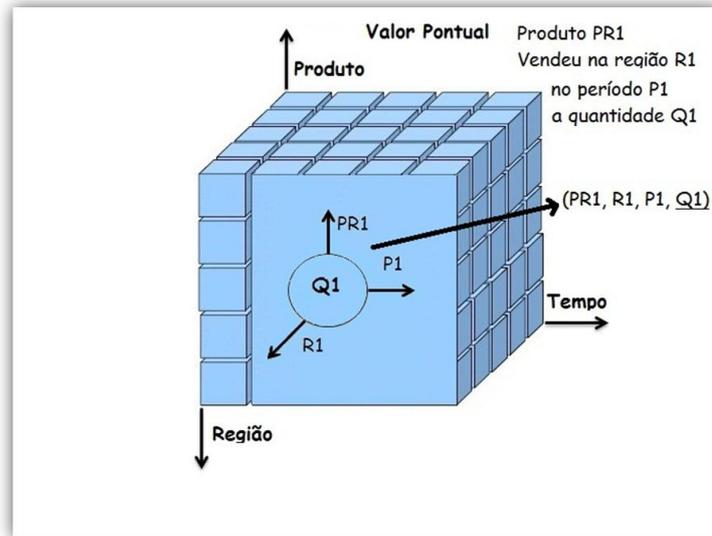


Figura 13 Representação Dimensional – Valor Pontual (In: BARBIERI, 2001).

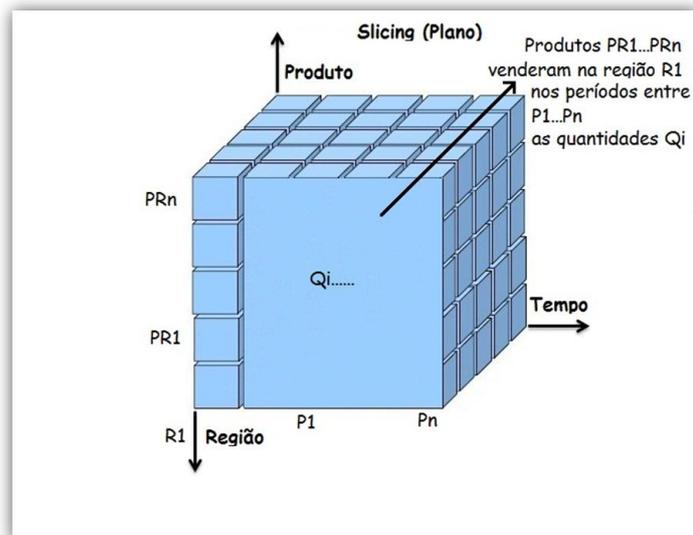


Figura 14 Representação Dimensional – Valores em Planos Fatias Slicing (In: BARBIERI, 2001).

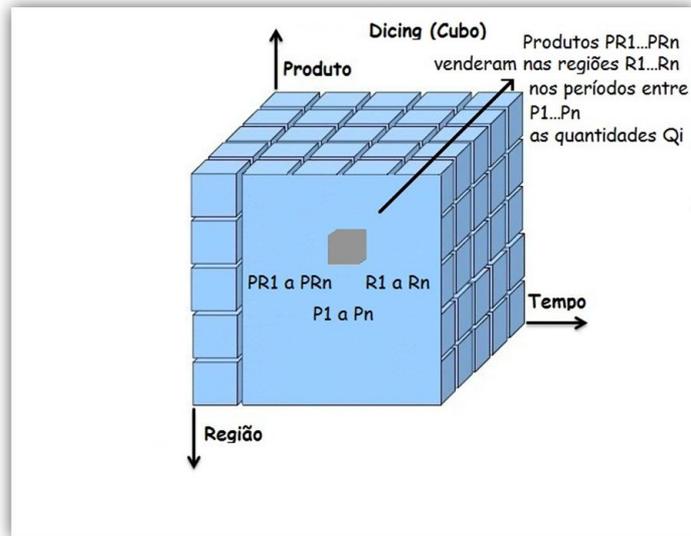


Figura 15 Representação Dimensional – Valores em Cubos (Dicing) (In: BARBIERI, 2001).

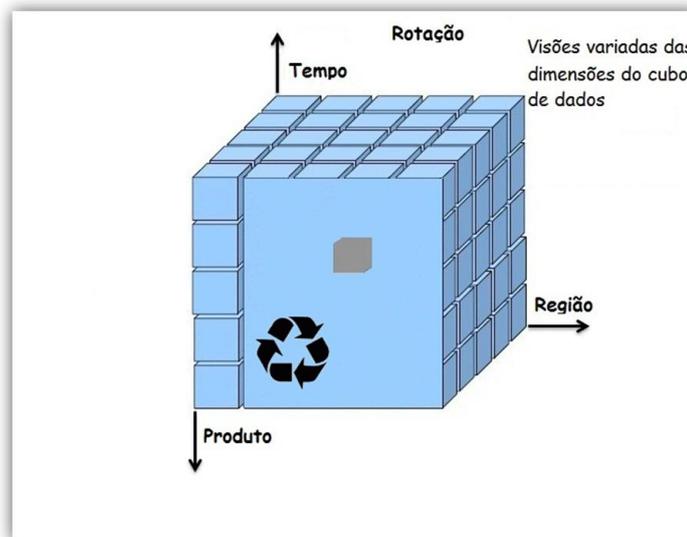


Figura 16 Representação Dimensional – Rotação de Planos (In: BARBIERI, 2001).

5.1.5 - Operadores dimensionais – Drill - Down E Drill – Up

Drill – Down e Drill – Up, também conhecido como roll – up estão relacionados com o nível de granularidade dos dados armazenados. Como exemplo, as informações relacionadas à dimensão Geografia estão estruturadas na seguinte hierarquia, PAÍS → REGIÃO → ESTADO → CIDADE → LOJA.

Na dimensão Tempo estão estruturadas na seguinte hierarquia:

SEMESTRE → MÊS → DIA.

O Conceito Drill – Down é relacionado com o fato de sair de um nível mais alto da hierarquia para buscar informações mais detalhadas em níveis menores. Por exemplo, descer da dimensão em nível (hierarquia) ESTADO, sendo efetuada a consulta das informações de vendas em nível de estado, desejando agora ter o detalhe por CIDADES, este processo se caracteriza por Drill – Down. Já o conceito de Drill – up é o inverso, por exemplo, está sendo efetuada uma análise em nível de Dia, porém deseja ter uma posição consolidada em nível de Mês, ou, seja, está se subindo em um nível, sendo caracterizado por Drill-Up (BARBIERI, 2001).

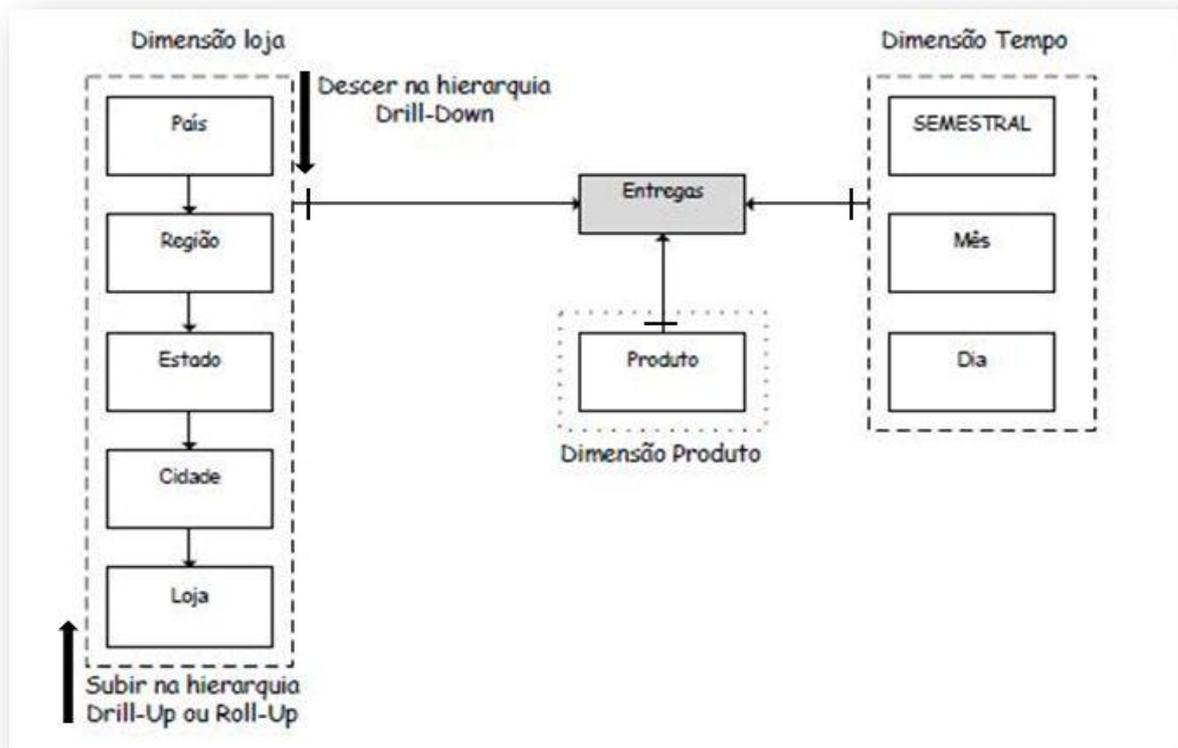


Figura 17 Representação dos principais comandos Dimensionais (In: BARBIERI, 2001).

5.1.6 - Operadores dimensionais – Drill – Across

O Conceito de Drill – Across está relacionado com o fato de poder pular de esquema para outro, desde que tenha algumas dimensões em conformidade, ou seja, as

dimensões estejam compartilhadas. Como segue o exemplo, do anterior, somente tendo a diferença que o Fato (informações centrais) é de ENTREGAS, ao invés de VENDAS. Desta forma, está sendo comparadas informações de VENDAS e ENTREGAS, de forma coerente, pois são dimensões comuns. Permitindo assim, o tratamento destas informações, correlacionadas e estruturadas separadas, sendo unidas por algumas dimensões coerentes, fazendo um “join” dimensional (BARBIERI, 2001).

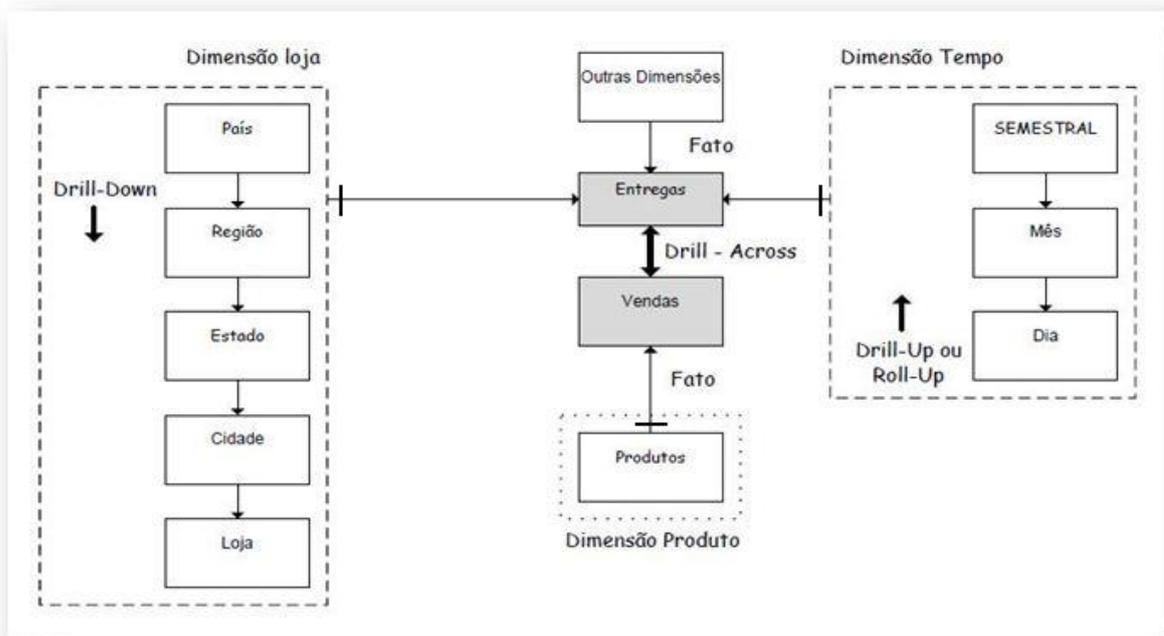


Figura 18 Representação dos principais comandos dimensionais, Drill-Down, Drill-Up e Drill-Across (In: BARBIERI, 2001).

5.1.7 - Operadores dimensionais – Drill – Through

O Conceito de Drill – Through está relacionado com o fato de desejar uma informação em um nível de detalhe menor do que o colocado na tabela Fato, permitindo assim a granularidade. Por exemplo, na tabela Fato VENDAS, as informações foram armazenadas em um nível de granularidade de PRODUTO por DIA e por LOJA, sendo que a menor informação que se pode alcançar é sobre o PRODUTO, tendo também as informações em nível de NOTA FISCAL, que foram totalizadas em PRODUTO, DIA e LOJA, dando origem à granularidade com que foi

armazenada na tabela Fato em outro ambiente diferente do DW / DM (Data Warehouse / Data Mart). Se tendo compatibilidade de software entre os dois ambientes, o próprio sistema transacional de origem, através do Drill – Through permitiria a operação em busca de informações além do nível de granularidade existente nas estruturas dimensionais. Estaria acessando as NOTAS FISCAIS que deram origem (por agregação) armazenada em cada linha da tabela Fato, como se fosse um Drill – Down, com diferença de propriedade de buscar o detalhe em outra estrutura (BARBIERI, 2001).

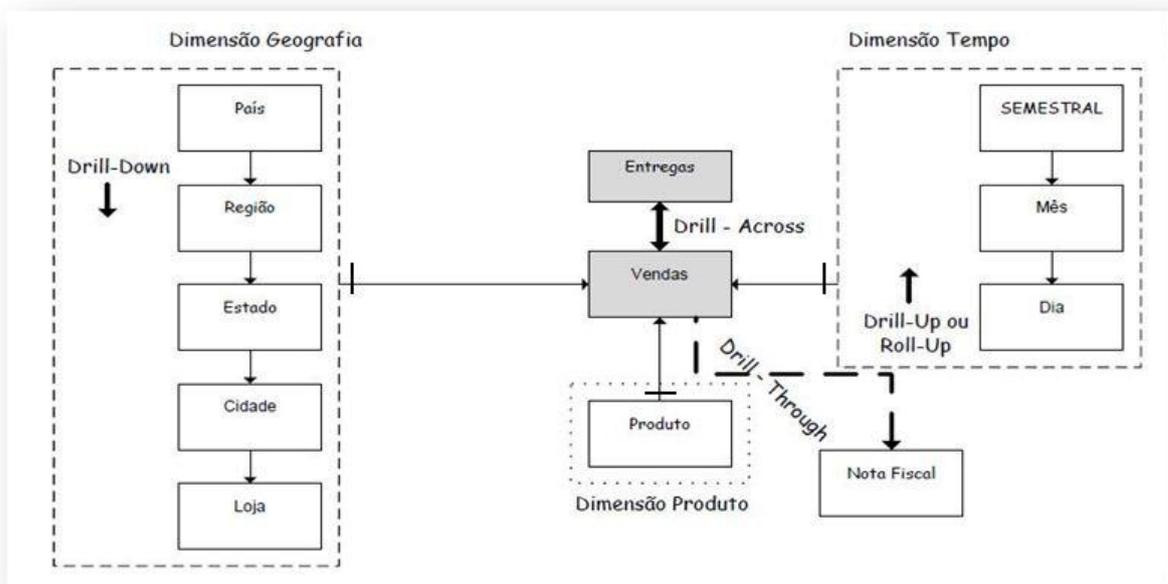


Figura 19 Representação dos principais comandos dimensionais, Drill-Through (In: BARBIERI, 2001).

6 – DIFERENÇA ENTRE DADOS OPERACIONAIS E DADOS INFORMACIONAIS

Estes dois tipos de dados possuem objetivos diferentes e estão relacionados aos sistemas tradicionais de informações, implementados sobre base de dados. Já os sistemas de informações executivas, implementados sobre Data Warehouse ou Data Marts.

Características	Dados Operacionais	Dados Informacionais
Conteúdo	Valores Correntes	Valores Sumarizados, Calculados, Integrados de várias fontes.
Organização dos Dados	Por aplicação de informação / Sistema	Por Assuntos / Negócios
Natureza dos Dados	Dinâmica	Estática
Formato das Estruturas	Relacional, próprio para processos transacionais.	Dimensional, simplificado, próprio para atividades analíticas.
Atualização dos Dados	Atualização campo a campo	Acesso sem update.
Uso	Estruturado, processamento repetitivo.	Desestruturado, com processamento analítico / heurístico.
Tempo de resposta	Otimizado para 2 a 3 segundos	Analises mais complexa, com tempos de respostas maiores.

Tabela 2 Comparação dos dados de natureza Operacional e Informacional (In: BARBIERI, 2001).

7 - DATA WAREHOUSE E DATA MART

Data Warehouse, cujo significado literal é Armazém de Dados, pode ser definido como um banco de dados, destinado a sistemas de apoio a decisão, cujos dados foram armazenados em estruturas lógicas dimensionais, possibilitando o seu processamento por ferramentas OLAP (Processamento Analítico On Line). Um data warehouse é um conjunto de programas que extraem e tratam dados do ambiente operacional da empresa, dando suporte a consultas ad-hoc (consulta de acesso casual único e com tratamento dos dados segundo parâmetros nunca utilizados, ou seja, calcula com antecedência todos os agregados possíveis de forma sistemática). O Data Warehouse deve ser adaptável e flexível a mudanças, ainda mais que as mudanças não podem ser evitadas. As necessidades dos usuários, as condições comerciais, os dados e a tecnologia, todos estes fatores estão sujeitos às mudanças decorrentes no tempo. Desta forma o Data Warehouse deve ser seguro de tal forma que proteja as informações controlando de modo eficaz o acesso às estas informações.

Um Data Warehouse é um Banco de Dados reunido a partir de muitos sistemas destinados a suportar a produção de relatórios gerenciais e a tomada de decisão. De uma forma geral, quando as pessoas falam a respeito de armazenamento de dados, na verdade elas estão se referindo aos armazenamentos de dados e às ferramentas que, juntos, formam os modernos ambientes de produção de relatórios (COREY, 2001).

Os componentes de um Data Warehouse são:

- **Sistemas Operacionais de Origem:** Estes sistemas capturam as transações da empresa, são considerados externo ao data warehouse, pois se tem pouco ou nenhum controle sobre o conteúdo e o formato dos dados nestes sistemas. Estes Sistemas têm como prioridade, desempenho e disponibilidade de Processamento. Consultas efetuadas nestes sistemas são limitadas, feitas em um registro por vez, que fazem parte do fluxo normal de transações e com uma demanda no sistema operacional limitado. Os Sistemas de origem mantem um volume pequeno de dados históricos, se

haver um data warehouse adequado, a responsabilidade dos sistemas operacionais representarem o passado será significante reduzido. Cada sistema de origem é uma aplicação independente (KIMBALL, 2002).

- **Data Staging Area:** É uma área de armazenamento com um conjunto de processos chamado de ETL (Extract, Transformation and Load), cujos significados são: Extração, Transformação e Carga. A Data Staging área, abrange os sistemas operacionais de origem e a área de apresentação dos dados. No Data warehouse os dados operacionais brutos, são transformados em um formato de warehouse pronto para serem consultados (KIMBALL, 2002).

Extração é a primeira etapa do processo, onde obtém – se os dados no ambiente Data Warehouse, sendo que o processo de extração envolve a leitura e a compreensão de dados de origem e cópia de dados necessários ao data warehouse para a Staging área a serem manipulados.

Transformação, logo que os dados são extraídos, ocorrem muitas transformações, como filtragem dos dados (correções de erros de digitação, solução de conflitos de domínio, tratamentos de elementos ausentes ou divisão em formatos padrão), combinação de dados de varias origens, cancelamento de dados com duplicidade e atribuição de chaves de warehouse. Estas transformações são para carregar os dados na área de apresentação. Porém existem casos que os dados chegam para a criação de um banco de dados físico concreto em um formato relacional (3NF, third-normal form), ou seja, é uma técnica de design que tenta remover redundâncias de dados, para isso realiza-se a filtragem e transformação usando conjunto de estruturas normalizadas. Porém há algumas restrições, significa que os dados são extraídos, transformados e carregados duas vezes, uma no banco de dados normalizado e outra quando é carregado no modelo dimensional, desta maneira, este processo consome mais tempo para carga periódica ou atualização de dados e capacidade de armazenamento dos dados (KIMBALL, 2002).

Carregar os dados assume a forma de um processo de apresentação de tabelas dimensionais garantidas por controle de qualidade para os vários recursos de carga de cada data mart. O Data Mart de destino precisa indexar

os dados que acabaram de chegar para execução da consulta (KIMBALL, 2002).

- Apresentação dos Dados: É o local em que os dados ficam organizados, armazenados e tornam se disponíveis para serem consultados diretamente pelos usuários (KIMBALL, 2002).

Os Data Marts (Mercados de Dados), referem – se à área de apresentação de dados, como uma serie de data marts integrados. Um Data Mart é uma parte que compõe a área de apresentação, ou seja, um data mart representa os dados de um único processo de negócio, tais processos de negócio cruzam os limites das funções da empresa, é o depósito de dados que atende certas áreas específicas da empresa, voltada para o processo gerencial.

- Ferramentas de acesso aos dados: as ferramentas de acesso aos dados consultam os dados na área de apresentação do data warehouse, sendo que a consulta é o ponto principal de um data warehouse.

Os metadados se referem a todas as informações no ambiente de data warehouse que não são os dados propriamente ditos. Os metadados são uma enciclopédia para o data warehouse, e podem estar presentes em uma variedade de formas e formatos para suportar as necessidades desiguais dos grupos de usuários técnicos, administrativos e de negócio do data warehouse. Quando os dados estão na Data Staging área, encontram-se os processos de transformação e carga. Os metadados são arquivos de testes, layouts de tabelas de destino, regras de transformação e filtragem, definições de dimensões e fatos em conformidade, códigos de personalização que estão na staging área. O objetivo dos metadados é cercar, catalogar, integrar e melhorar essas diversas variedades de metadados, da mesma forma que os recursos de uma biblioteca (KIMBALL, 2002).

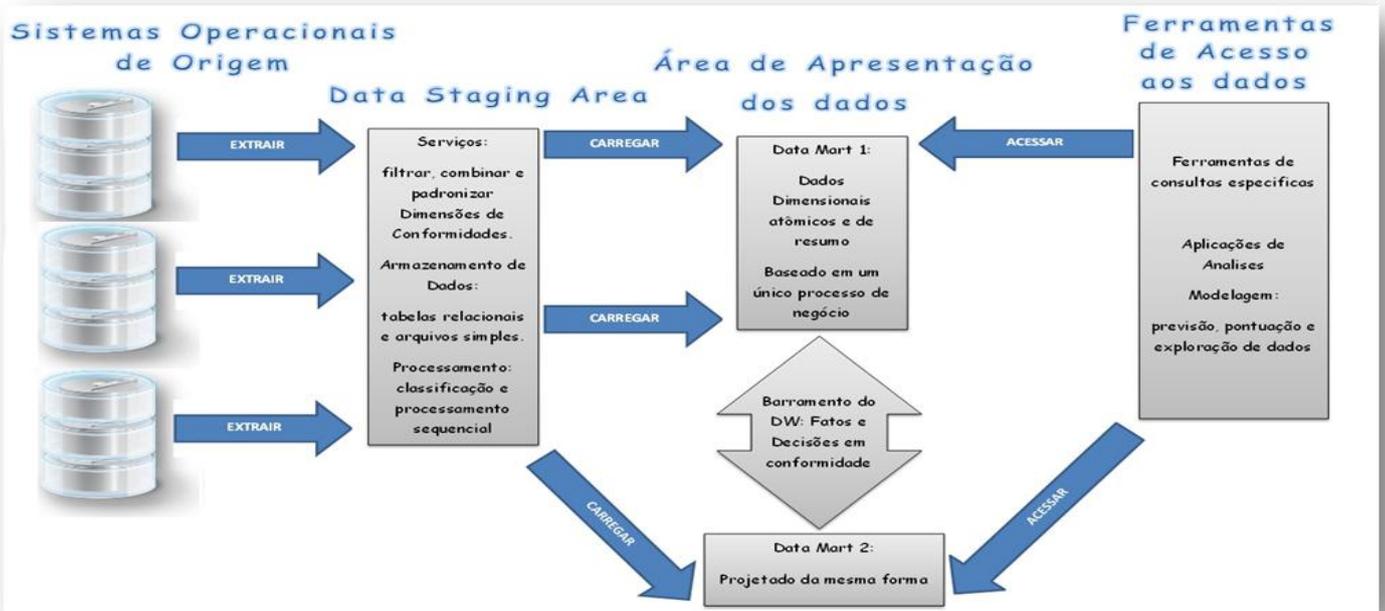


Figura 20 Elementos Básicos do Data Warehouse (In: adaptado KIMBALL, 2002, p.09).

8 – GRANULARIDADE

Granularidade refere-se ao nível de detalhamento ou de resumo de dados, que pode ser encontrada em um Ambiente data warehouse. No entanto, a definição do nível de granularidade mais baixo possível depende do processo de negócio que está sendo modelado (KIMBALL, 2002). Quanto menor o nível de granularidade maior será o nível de detalhamento, porém será necessário mais espaço para os dados serem armazenados e quanto maior o nível de granularidade menor será o nível de detalhamento. Quando se consulta dados em um Data Warehouse, geralmente os dados estão resumidos em diferentes níveis, é possível mudar o nível específico de detalhe e por fim a consulta será finalizada. Uma transação estaria em um nível de granularidade mais baixo e o resumo de todas as transações poderia estar em um nível mais alto de granularidade. A granularidade trás a vantagem de maior flexibilidade contendo dados históricos e eventos de uma organização.

9 - ESQUEMA DE ESTRELA (STAR SCHEMA)

O esquema de estrela transforma os dados em tabelas Fatos, onde se concentram os dados de interesse de manipulação numérica e estatística. Nas tabelas Dimensão, possuem as chaves de entrada do modelo, além das informações descritivas de cada dimensão (BARBIERI, 2001). A Tabela de Fatos está no centro da “estrela”, cujos pontos são as tabelas de dimensões. O esquema de estrela otimiza o armazenamento no banco de dados, em relação a acessos aos dados

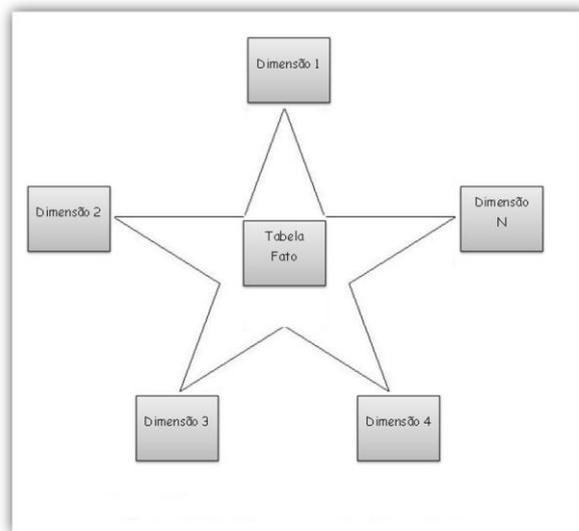


Figura 21 Representação do Esquema Estrela (In: Feito por Jailson Camargo de Lima)

10 - BANCO DE DADOS ORACLE

Produzido pela Oracle Corporation de Belmont, Califórnia, no final da década de 1970, o Sistema Gerenciador de Banco de Dados Oracle, idealizado por Lawrence J. Ellison (Larry Ellison). Inclui extensões orientadas a objetos significativas, e tem um inovador modelo de relacionamento de dados e linguagem de programação servidora, que é o PL/SQL. Uma rede (LAN) é suficiente para comportar um Banco de Dados Relacional de tecnologia Cliente/Servidor (C/S). Esta tecnologia tem como característica principal a divisão de tarefas entre o cliente, a estação de trabalho que ordena através das aplicações o acesso ao Banco de Dados, e o servidor, que executa tarefas.

11- DESENVOLVIMENTO DO PROJETO

A Modelagem é de um sistema de vendas de uma rede de lojas. Essa rede possui diversas lojas distribuídas em diferentes regiões. O modelo relacional utilizado será mostrado na Figura 22, possui uma tabela de Venda que contém todos os dados relacionados a uma venda, com seus itens vendidos sendo armazenados na tabela Det_Venda, uma tabela Pedido de Compra para requisitar novos produtos, com seus itens requisitados sendo armazenados na tabela Det_Compra e a condição de pagamento armazenada na tabela Cond_Pagto. Para gerenciar as vendas, possui também um cadastro de Cliente, Dependente, Loja Filial, Funcionário, Produto, Categoria, Marca, Fornecedor e Contas a pagar e receber.

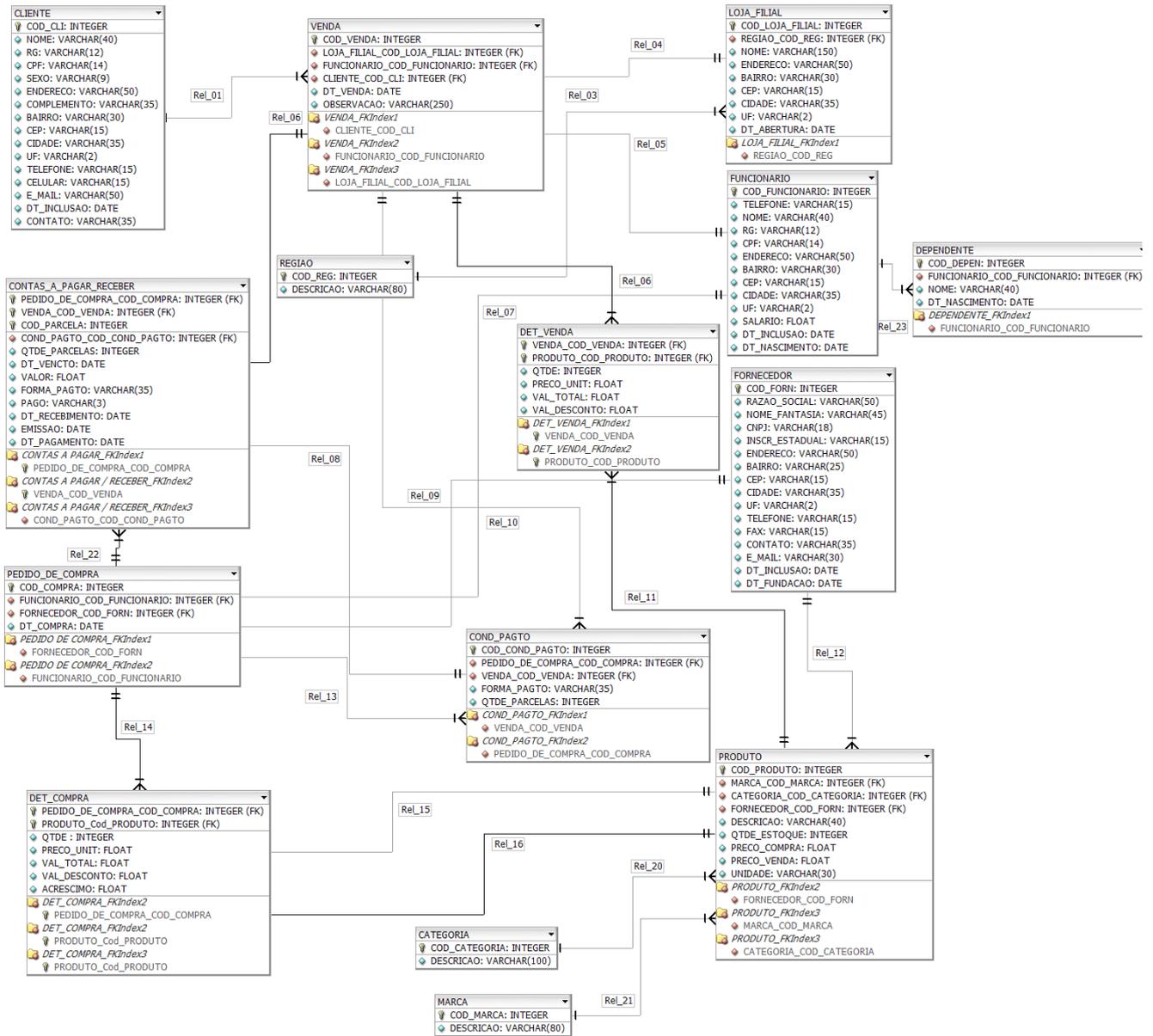


Figura 22 Modelo Relacional do Sistema de Vendas da rede de lojas (In: Feito por Jailson Camargo de Lima)

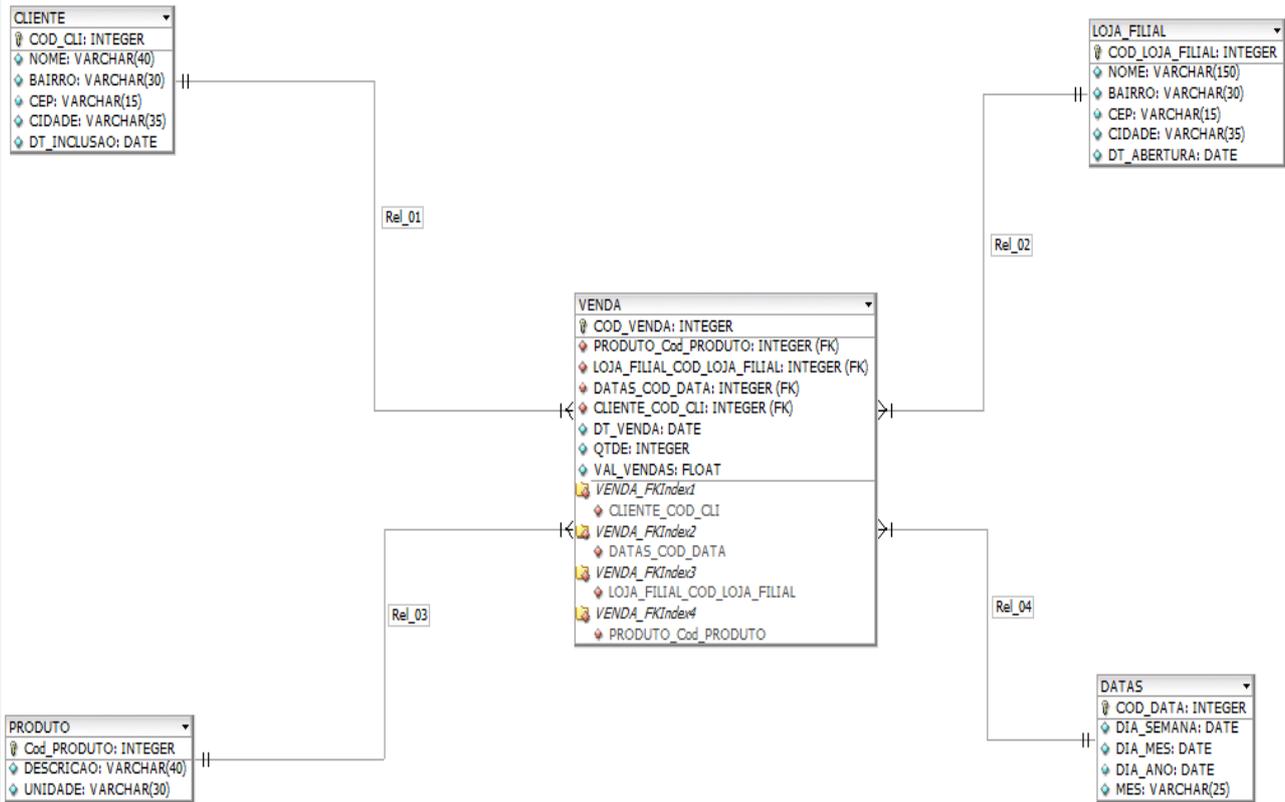


Figura 23 Modelo Dimensional do Sistema de Vendas da rede de Lojas (In: Feito por Jailson Camargo de Lima)

12 – CONSIDERAÇÕES FINAIS

O Modelo Relacional é fundamental, pois nele são desempenhadas as operações transacionais, que é o seu objetivo principal, porém não atende a grande quantidade de informações, desta maneira seu desempenho é baixo para executar consultas mais complexas. Com o Modelo Multidimensional é possível realizar consultas complexas, em grandes volumes de dados, com alto desempenho. As informações dependem de ser analisadas e compreendidas, separadas do sistema de apoio às operações e visto como uma solução de inteligência de negócios viabilizando a tomada de decisões claramente extraídas dos Sistemas OLTP.

REFERÊNCIAS

BARBIERI, Carlos, BI – **Business Intelligence, Modelagem & Tecnologia**, Axcel. Books do Brasil Editora Ltda., 2001.

COREY, Michael, ABBEY, Michael, ABRAMSON, Ian; TAUB, Ben. **Oracle 8i Data Warehouse**, Editora Campus Ltda. (Autorizado por Oracle Press), 2001.

DAVIES, Paul Beynon, **Database Systems**, Palgrave Macmillan, 2004.

INMON, William H., **Building the Data Warehouse**, Wiley Publishing, Inc., 2005.

JENSEN, Christian S., PEDERSEN, Torben Bach, THOMSEN, Christian, **Multidimensional Databases and Data Warehousing**, Morgan & Claypool, 2010.

KIMBALL, Ralph, ROSS, Margy. **The Data Warehouse Toolkit**, Editora Campus Ltda., 2002.

MACHADO, Felipe, ABREU, Mauricio, **PROJETO DE BANCO DE DADOS – Uma Visão Prática**, Editora Érica Ltda., 2004.

MOLINA, Hector Garcia, ULLMAN, Jeffrey D., WIDOM, Jennifer, **Implementação de Sistemas de Banco de Dados**, Editora Campus Ltda, 2001.

NARDI, Alexandre Ricardo, **Fundamentos e Modelagem de Bancos de Dados Multidimensionais, Microsoft Brasil**, Disponível em: <http://msdn.microsoft.com/pt-br/library/cc518031.aspx>. Acessado em Abril 2011.

PONNIAH, PAULRAJ, **Data Warehousing Fundamentals, A Comprehensive Guide for IT Professionals**, A Wiley-Interscience Publication John Wiley & Sons, Inc., 2001.

SILBERSCHATZ, Abraham, KORTH, Henry F., SUDARSHAN, S., **Sistema de Banco de Dados**, Pearson Makron Books, 1999.

SILBERSCHATZ, Abraham, KORTH, Henry F., SUDARSHAN, S., **Sistema de Banco de Dados**, Elsevier Editora Ltda, 2006.

THOMSEN, Erik, **Building Multidimensional Information Systems**, Wiley Computer Publishing, John Wiley & Sons, Inc., 2002.

ANEXO

O Microsoft Excel fará a comunicação entre o Banco de Dados Multidimensional. Conforme mostrado abaixo.

The screenshot shows a PivotTable in Microsoft Excel with the following data:

REGIOI	CITY	Beverage	Game Rental	Laser Disc Rent	Snacks	Video Rental	Video Sale	Total geral
Central	Chicago	63.43	4533.58	3616.87	99.74	13177.65	28856.23	50347.5
	Cincinnati	810.05	19355.59	14272.32	567.66	61710.98	107448.35	204164.95
	Dallas	167.76	3550.98	3034.54	138.99	14480.42	18335.11	39707.8
	Louisville	519.66	15309.48	12292.17	551.9	58639.66	86799.7	174112.57
	Minneapolis	119.52	3160.63	3911.52	129.81	14177.66	25125.66	46624.8
	Nashville	32.71	4081.77	3693.7	171.35	13874.53	18549.83	40403.89
	St. Louis	415.1	8640.25	6929.95	278.91	32011.62	57246.3	105522.13
Central Total		2128,23	58632,28	47751,07	1938,36	208072,52	342361,18	660883,64
East	Atlanta	449.03	8756.08	7731.61	206.93	27229.69	49103.38	93476.72
	Boston	359.34	8697.9	6439.94	220.69	27705.45	48609.47	92032.79
	Miami	210.69	3903.55	3720.21	140.62	13014.4	20613.86	41603.33
	New Orleans	99.35	3991.55	3425.9	140.72	15162.3	25297.61	48117.43
	New York	1402.14	36707.83	30246.58	1152.48	126113.82	200785.41	396408.26
	Philadelphia	552.48	8904.68	8246.05	268.22	32249.18	58687.49	108908.1
	Pittsburgh	318.56	8109.73	6520.66	240.13	27293.73	45160.37	87643.18
	Washington	730.32	17424.27	12905.29	672.87	57548.68	86509.26	175790.69
East Total		4121,91	96495,59	79236,24	3042,66	326317,25	534766,85	1043980,5
West	Denver	296.65	8715.9	6580.41	234.88	28007.17	46839.13	90674.14
	Los Angeles	189.82	4208.95	3802.74	93.39	11986.96	14500.75	34782.61
	Phoenix	307.94	3738.55	3351.02	122.58	14193.25	20650.4	42363.74
	San Francisco	635.88	15261.94	12642.95	522.61	56706.32	97039.63	182809.33
	Seattle	439.38	14598.69	14362.73	588.17	55352.22	94444.97	179786.16
West Total		1869,67	45504,03	40720,05	1039,03	107945,92	172424,28	324445,00



Foi efetuado um Drill-Down de Região (Region) para Cidade (City), ou seja, desceu da hierarquia região no nível de detalhamento cidade.

Lista de campos da tabela dinâmica

Escolha os campos para adicionar ao relatório:

- Soma de COST_SUM
- Soma de PROFIT_SUM
- Soma de SALES_SUM
- CAL_DATE_YR
- CAL_DATE_YR
- CITY
- COST_COUNT
- COST_COUNT

Arraste os campos entre as áreas abaixo:

Filtro de Relat... Rótulos de Col...

DEPARTMENT

Rótulos de Linha Σ Valores

REGION Soma de SAL...

CITY

Adiar Atualização do L... Atualizar

Soltar Campos de Filtros de Relatório Aqui								
Soma de SALES SUM		DEPARTMENT						
REGION	CITY	Beverage	Game Rental	Laser Disc Rent	Snacks	Vídeo Rental	Vídeo Sale	Total Geral
East	Atlanta	449,03	8756,08	7731,61	206,93	27229,69	49103,38	93476,72
East Total		449,03	8756,08	7731,61	206,93	27229,69	49103,38	93476,72
Total Geral		449,03	8756,08	7731,61	206,93	27229,69	49103,38	93476,72

Abaixo, foi feito um Drill-up de Cidade (City) para Região (Region) sendo consultado somente o departamento (Department) bebida.

Lista de campos da tabela dinâmica

Escolha os campos para adicionar ao relatório:

- Soma de COST_SUM
- Soma de PROFIT_SUM
- Soma de SALES_SUM
- CAL_DATE_YR
- CAL_DATE_YR
- CITY
- COST_COUNT
- COST_COUNT

Arraste os campos entre as áreas abaixo:

Filtro de Relat... Rótulos de Col...

DEPARTMENT

Rótulos de Linha Σ Valores

REGION Soma de SAL...

CITY

Adiar Atualização do L... Atualizar

Soltar Campos de Filtros de Relatório Aqui			
Soma de SALES SUM		DEPARTMENT	
REGION	CITY	Beverage	Total Geral
Central	Chicago	63,43	63,43
	Cincinnati	810,05	810,05
	Dallas	167,76	167,76
	Louisville	519,66	519,66
	Minneapolis	119,52	119,52
	Nashville	32,71	32,71
	St. Louis	415,1	415,1
Central Total		2128,23	2128,23
East	Atlanta	449,03	449,03
	Boston	359,34	359,34
	Miami	210,69	210,69
	New Orleans	99,35	99,35
	New York	1402,14	1402,14
	Philadelphia	552,48	552,48
	Pittsburgh	318,56	318,56
	Washington	730,32	730,32
East Total		4121,91	4121,91
West	Denver	296,65	296,65
	Los Angeles	189,82	189,82
	Phoenix	307,94	307,94
	San Francisco	635,88	635,88
	Seattle	439,38	439,38