

UNIVERSIDADE FEEVALE

FELIPE ENGELMANN

PROPOSTA DE UMA SOLUÇÃO DE BI PARA O ERP
SAFETECH

Novo Hamburgo
2012

FELIPE ENGELMANN

PROPOSTA DE UMA SOLUÇÃO DE BI PARA O ERP
SAFETECH

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Ciência da Computação pela
Universidade Feevale

Orientador: Me. Edvar Bergmann Araujo

Novo Hamburgo
2012

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A minha família pelo apoio nos momentos mais difíceis do trabalho e principalmente pela educação que me foi dada. A minha namorada Marla, pelo carinho, apoio e compreensão durante esse período. Ao meu orientador Edvar Bergmann Araujo que acrescentou seu conhecimento para o enriquecimento desse trabalho.

RESUMO

Com a crescente expansão da informação pelo mundo, existem dados espalhados por toda a parte. Nas empresas funciona da mesma forma, a demanda por informação é enorme sendo que muitas vezes as empresas não conseguem aproveitar todo o potencial dessa informação. Geralmente os sistemas ERP mostram de uma forma muito específica a informação, cabível para a situação necessária no momento. Nesses casos, o *Business Intelligence* (BI) surge para auxiliar em soluções mais abrangentes para as áreas de negócio, fornecendo subsídios para uma melhor tomada de decisão pelos gestores da empresa. Sendo assim, esse trabalho tem como objetivo propor uma solução de BI para atender as necessidades do ERP Safetech. Objetiva-se criar uma arquitetura que permita a uma equipe treinada reunir e organizar as informações da empresa conforme sua regra de negócio, em repositórios de dados conhecidos como *Data Mart* (DM). Também está prevista a concepção das demais etapas da construção de uma arquitetura completa de BI, sendo elas o ETL e as consultas OLAP. Como o ERP desenvolvido e comercializado pela Safetech é construído com soluções *Oracle*, de onde serão aplicadas ferramentas da *Oracle* para compor a solução que será concebida neste trabalho.

Palavras-chave: *Business Intelligence*. ERP. ETL. *Data Mart*. OLAP.

ABSTRACT

With the increasing spread of information around the world, data is scattered everywhere. In companies work the same way, the demand for information is enormous and that companies often fail to realize the full potential of your information. ERP systems generally show very specific information, appropriate for the situation when necessary. In such cases, Business Intelligence (BI) appears to assist in more comprehensive solutions in the areas of business, supporting better decision making by managers. Thus, this paper aims to propose a BI solution to meet the needs of ERP Safetech. The objective is to create an architecture that enables a trained team gather and organize information according to your company's business rule in data repositories known as Data Mart (DM). Also provided is the design of the other stages of building a complete architecture of BI, they being the ETL and OLAP queries. As the ERP developed and marketed by Safetech is constructed with Oracle solutions, is intended to evaluate, and if possible to apply Oracle tools to form a solution which is designed in this work.

Key words: Business Intelligence. ERP. ETL. Data Warehouse. OLAP.

LISTA DE FIGURAS

Figura 1.1 – Estrutura do ambiente BI _____	16
Figura 1.2 – Data Mart _____	18
Figura 1.3 – Granularidade dos dados em um DW _____	21
Figura 1.4 – Tabela fato e tabelas dimensões _____	23
Figura 1.5 – Modelo em Estrela _____	24
Figura 1.6 – Modelo Floco de Neve _____	25
Figura 1.7 – Modelo ETL _____	28
Figura 1.8 – Modelo OLAP _____	29
Figura 1.9 – Tipos de Arquitetura OLAP _____	32
Figura 2.1 – Distribuição da carteira de clientes por atividade _____	34
Figura 2.2 – Módulos do ERP Safetech _____	36
Figura 3.1 – Opções de ETL _____	40
Figura 3.2 – Integração do Produto _____	41
Figura 3.3 – Propagando alterações no sistema _____	42
Figura 3.4 – Geração de Perfis _____	42
Figura 3.5 – Ciclo do BI _____	44
Figura 3.6 – ODBC para OLAP _____	44
Figura 3.7 – Tela principal do Oracle BI Administration Tool _____	45
Figura 3.8 – OLAP visualizado _____	46
Figura 3.9 – Relatório Web OLAP _____	46
Figura 3.10 – Relatório Web OLAP _____	47
Figura 4.1 – Visão geral da solução de BI construída _____	49
Figura 4.2 – Cubo de Dados no Oracle Warehouse Builder _____	51
Figura 4.3 – Cubo de Dados pronto para o Export _____	52
Figura 4.4 – Fluxo da solução de BI _____	53
Figura 4.5 – Configuração do <i>Oracle Warehouse Builder</i> _____	55
Figura 4.6 – Configuração do <i>Oracle Warehouse Builder</i> - Parâmetros _____	56
Figura 4.7 – Tela de desenvolvimento do <i>Oracle Warehouse Builder</i> _____	57
Figura 4.8 – Tela de desenvolvimento do <i>Oracle BI Administration Tool</i> _____	58
Figura 4.9 – Geração do relatório <i>Oracle BI Publisher</i> - Geral _____	59
Figura 4.10 – Geração do relatório <i>Oracle BI Publisher</i> - Filtro _____	60

Figura 4.11 – Geração do relatório OLAP pelo Excel	61
Figura 4.12 – Carregando Cubo de dados	61
Figura 4.13 – Escolhendo opção de Dados	62
Figura 4.14 – Tela inicial da estrutura desenvolvida	63
Figura 5.1 – ER do modelo usado para estudo de caso	66
Figura 5.2 – Rotina de carga de dados entre ERP e DM	70
Figura 5.3 – Estrutura de dados carregados no DM	71
Figura 5.4 – ER do Modelo <i>Star</i>	72
Figura 5.5 – Cubo de dados Modelo <i>Star</i>	72
Figura 5.6 – ER do Modelo <i>Snowflake</i>	73
Figura 5.7 – Cubo de dados Modelo <i>Snowflake</i>	74
Figura 5.8 – Plano de execução do Modelo <i>Star</i>	75
Figura 5.9 – Plano de execução do Modelo <i>Snowflake</i>	76
Figura 5.10 – Solução proposta com <i>Oracle BI Publisher</i>	77
Figura 5.11 – Tela de parametrização das consultas OLAP	78
Figura 5.12 – Tela de carga das consultas OLAP	78
Figura 5.13 – Tela para consulta da carga das consultas OLAP	79
Figura 5.14 – Tela para geração do arquivo Excel	79
Figura 5.15 – Relatório Excel formatado no módulo OLAP	80

LISTA DE QUADROS

Quadro 1.1 – Diferenças entre bancos de dados operacionais e DW _____	19
Quadro 5.1 – Descrição dos campos da tabela SFT_FATO_TITULO_CLIENTE _____	66
Quadro 5.2 – Descrição dos campos da tabela SFT_AGRUP_BANCO _____	67
Quadro 5.3 – Descrição dos campos da tabela SFT_AGRUP_CLIENTE _____	67
Quadro 5.4 – Descrição dos campos da tabela SFT_MOVTO_VENC_TITULO_CLIENTE _____	68
Quadro 5.5 – Descrição dos campos da tabela FUNCAO_DATA _____	68
Quadro 5.6 – Descrição dos campos da tabela SFT_LOCALIZA _____	69

LISTA DE ABREVIATURAS E SIGLAS

<i>BI</i>	<i>Business Intelligence</i>
<i>CRM</i>	<i>Customer Relationship Management</i>
<i>CSV</i>	<i>Comma-Separated Values</i>
<i>DM</i>	<i>Data Mart</i>
<i>DOLAP</i>	<i>Desktop On-line Analytical Processing</i>
<i>DSS</i>	<i>Operacional Source Store</i>
<i>DW</i>	<i>Data Warehouse</i>
<i>ER</i>	<i>Entidade Relacionamento</i>
<i>ERP</i>	<i>Enterprise Resource Planning</i>
<i>ETL</i>	<i>Extract Transform Load</i>
<i>EIS</i>	<i>Executive Information Systems</i>
<i>HOLAP</i>	<i>Hybrid On-Line Analytical Processing</i>
<i>HTML</i>	<i>Hyper Text Markup Language</i>
<i>JDBC</i>	<i>Java Database Connectivity</i>
<i>MIT</i>	<i>Massachusetts Institute of Technology – EUA</i>
<i>MOLAP</i>	<i>Multidimensional On-Line Analytical Processing</i>
<i>MRP</i>	<i>Material Requirement Planning</i>
<i>ODBC</i>	<i>Open Data Base Connectivity</i>
<i>ODS</i>	<i>Operacional Data Store</i>
<i>OLAP</i>	<i>On-Line Analytical Processing</i>
<i>OLTP</i>	<i>On-Line Transaction Processing</i>
<i>ROLAP</i>	<i>Relation On-Line Analytical Processing</i>
<i>SAD</i>	<i>Sistemas de Apoio a Decisão</i>
<i>SGBDR</i>	<i>Sistema Gerenciador de Banco de Dados Relacional</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>TI</i>	<i>Tecnologia da Informação</i>
<i>XML</i>	<i>Extensible Markup Language</i>

SUMÁRIO

INTRODUÇÃO	12
1 BUSINESS INTELLIGENCE	15
1.1 <i>Data Warehouse</i> (DW)	16
1.1.1 Data Marts (DM)	17
1.1.2 Arquitetura DM e DW	18
1.1.3 Diferenças entre ambiente operacional e ambiente DW	19
1.1.4 Granularidade	20
1.1.5 Métodos de implementação do DW	21
1.2 Modelagem dimensional dos dados	22
1.2.1 Tabela fato	23
1.2.2 Tabela dimensão	23
1.2.3 Modelo Star	24
1.2.4 Modelo Snowflake	24
1.2.5 Metadados	25
1.3 Extração, transformação e carga (ETL)	26
1.4 On-line analytic processing (OLAP)	29
1.4.1 Arquiteturas OLAP	31
2 ERP SAFETECH	34
2.1 SAFETECH INFORMÁTICA	34
2.2 Gestão SFT	35
2.2.1 Principais Recursos	36
2.3 Origem dos dados para análise	37
3 ANÁLISE DE SOLUÇÕES DE BI	39
3.1 <i>ETL</i>	39
3.1.1 <i>Oracle Warehouse Builder</i>	39
3.2 <i>OLAP</i>	43
3.2.1 <i>Oracle BI Administration Tool</i>	43
3.2.2 <i>Oracle BI Publisher</i>	46
3.3 Considerações finais	48
4 ARQUITETURA PROPOSTA	49
4.1 Definição das ferramentas	50
4.1.1 ETL	50
4.1.2 Cubo de Dados	51
4.1.3 OLAP	52
4.2 Visão geral	53
4.3 Criação do <i>Data Mart</i>	54
4.4 ETL	55
4.5 Cubo de dados	57
4.6 OLAP	58
4.6.1 <i>Oracle BI Publisher</i>	59
4.6.2 Excel	60
4.6.3 Modelagem da Arquitetura proposta	62
5 ESTUDO DE CASO	65
5.1 Construção do DM	65
5.1.1 Tabela fato SFT_FATO_TITULO_CLIENTE	66

5.1.2	Tabela dimensão SFT_AGRUP_BANCO	67
5.1.3	Tabela dimensão SFT_AGRUP_CLIENTE	67
5.1.4	Tabela dimensão SFT_MOVTO_VENC_TITULO_CLIENTE	68
5.1.5	Tabela dimensão FUNCAO_DATA	68
5.1.6	Tabela dimensão SFT_LOCALIZA	69
5.2	Carga de dados	69
5.3	Criação dos cubos de dados	71
5.3.1	Modelo <i>Star</i>	71
5.3.2	Modelo <i>SnowFlake</i>	73
5.4	Análise dos cubos de dados	74
5.4.1	Modelo <i>Star</i>	74
5.4.2	Modelo <i>SnowFlake</i>	75
5.4.3	Considerações Finais	75
5.5	Consulta dos dados	76
5.5.1	Oracle BI Publisher	76
5.5.2	Módulo OLAP	77
CONCLUSÃO		81
REFERÊNCIAS BIBLIOGRÁFICAS		82
APÊNDICE		84
APÊNDICE A – CONFIGURAÇÃO DA TABLESPACE DO DM		84
APÊNDICE B – CRIAÇÃO DAS TABELAS PARA O OLAP		85

INTRODUÇÃO

Com a crescente expansão da informação pelo mundo, existem muitos dados distribuídos por toda a parte. Nas empresas funciona da mesma forma, a demanda por informação é enorme, sendo que muitas vezes não se consegue tirar todo o proveito destas informações, obtendo possíveis tendências ou problemas que poderiam ser administrados de uma forma mais simplificada. Nos dias de hoje, quem possui a informação certa pode ganhar uma boa fatia do mercado e se sobrepôr sobre seus concorrentes.

O BI (*Business Intelligence*) tem como propósito auxiliar a tomada de decisão pelos gestores da empresa, pois permite visualizar as informações de uma forma simples e flexível. O BI possui consultas dinâmicas com a facilidade de modificá-las com um clique do mouse (COREY, 2001). Empresas de todos os portes estão passando a utilizar soluções de BI para auxiliarem em diferentes situações na tomada de decisão como: otimizar trabalho, reduzir custos, eliminar duplicidade de tarefas, prever crescimento da empresa e planejar o controle estratégico. (PRIMAK, 2008)

A história do *Business Intelligence* também está conectada diretamente ao ERP (*Enterprise Resource Planning*), sigla que representa os sistemas interligados de gestão empresarial, cuja função é facilitar os aspectos operacionais das empresas. Esses sistemas registram, processam e documentam cada fato novo na engrenagem corporativa e distribuem a informação de maneira clara e segura, em tempo real. No entanto, as informações geradas pelo ERP podem contribuir de forma mais efetiva para o processo decisório com o suporte de uma ferramenta BI. (PRIMAK, 2008)

Geralmente os sistemas ERP mais tradicionais, são focados nos processos individuais para que não falte nenhuma informação para o usuário. Os processos são complexos, muito bem constituídos, e relevantes a área e atuação para o qual foram desenvolvidos. Os módulos de um ERP são geralmente separados, existindo interligações entre eles para consistir as informações. Além de possuírem módulos separados, existem casos onde a informação está armazenada em estruturas totalmente diferentes (fisicamente ou não) havendo então a preocupação em interligar estas informações para que se tenha uma visão completa.

Resumindo, os sistemas ERP são muito importantes para as empresas. Sem eles, as empresas tendem a perder competitividade e podem ter problemas para se manterem legalizadas perante o governo. Com tanta competição de mercado, tentar controlar processos de forma manual pode gerar uma grande perda para a empresa (INMON, 1999). Por outro

lado, estas organizações precisam de soluções que permitam consolidar as informações de forma dinâmica e eficiente. Neste ponto, as soluções de BI apresentam grande contribuição. Desta forma, é possível afirmar que uma solução de BI pode complementar os sistemas ERP fornecendo aos gestores informações valiosas ao processo decisório.

Com o objetivo de contribuir efetivamente para o processo de tomada de decisão, é importante criar um repositório de dados único, que irá agrupar os dados que poderão ser provenientes de diversos sistemas e bases de dados espalhados pela empresa. Com isto, gera-se um DW (*Data Warehouse*) completo que irá contribuir melhor com as decisões gerenciais. Segundo Larson (2006), o DW é um repositório de dados atuais e históricos que servirá para a tomada de decisão dos gestores da organização sobre tendências, métricas, causas, sintomas, possibilidades e outras avaliações importantes para a empresa.

A etapa de extração, transformação e carga é responsável por copiar e organizar os dados das bases de dados de origem para o DW. O DW pode armazenar múltiplas linhas de informação, com várias dimensões e tipos de dados totalmente diferentes (CHAKRABARTI, 2009, p. 119-120). A informação contida no DW é modelada de forma dimensional. Modelagem dimensional é o nome dado a uma técnica antiga usada para criar bancos de dados simples, relacionais e compreensíveis.

Muitas pessoas consideram fácil imaginar a área de negócio como um cubo de dados, com legendas em cada uma das arestas do cubo, visualizando graficamente a informação contida. Com um simples clique do mouse é possível agrupar essas informações da forma mais cabível ao usuário no momento. (KIMBALL, 1998)

Quando o DW está completamente estruturado, é utilizada uma ferramenta OLAP (*Online Analytical Processing*) que possibilite a consulta e análise de dados para se adquirir informações importantes para várias tomadas de decisão. Podendo trabalhar com a informação das formas mais diversas existentes, sem ter a posição fixa que os relatórios ERP geram.

Muitas vezes as soluções de BI desenvolvidas são focadas em determinados departamentos ou áreas de negócio. Um exemplo disto é o trabalho desenvolvido por Felber (2006), que criou um módulo de BI para o ERP Safetech com o foco na área comercial. Tais soluções são interessantes e atendem aos propósitos previstos. No entanto, a estrutura do DW e o processo de ETL não podem ser modificados sem a interferência dos técnicos da Safetech.

Desta forma a proposta do trabalho é desenvolver uma solução de BI para o ERP Safetech que permita aos consultores criarem soluções de BI de acordo com a área de atuação e necessidades da empresa em destaque. A ferramenta proposta permitirá que o consultor da empresa ou uma equipe treinada possa projetar o DM, o processo de ETL e a montagem dos cubos de dados, conforme regras de negócio e necessidades do usuário final. Os gestores, que serão usuários finais, terão a disposição consultas OLAP de simples utilização, onde poderão fazer as análises pertinentes ao negócio.

Esse projeto irá contemplar estruturas de negócios existentes na empresa, sem que haja a preocupação com que área de negócio é proveniente ou que formas de dados são utilizadas por tal. Para a validação da proposta, será feito um estudo de caso para uma amostragem de um processo específico utilizado.

Com o auxílio de ferramentas da *Oracle*, tal como “*Oracle BI Administration Tool*” será possível dimensionar de uma forma mais simplificada toda a estrutura do BI. Essa ferramenta auxilia na estruturação do BI como um todo, simplificando a construção da solução proposta. Essa ferramenta será implementada no ERP Safetech para com que a empresa obtenha ganhos perante o mercado de *software*. A estruturação do BI irá trazer ganhos financeiros e motivacionais com os usuários por estar permitindo mais flexibilidade nas consultas de informações requeridas por eles.

O trabalho está dividido em cinco capítulos. O primeiro capítulo aborda o conteúdo teórico sobre os assuntos envolvidos neste trabalho. O segundo capítulo apresenta o ERP Safetech e sua estrutura. O terceiro capítulo trata sobre as análises feitas em algumas ferramentas de BI, ETL e OLAP existentes, a fim de verificar quais poderiam ser utilizadas na solução proposta. O quarto capítulo detalha a solução proposta e todas as suas funcionalidades. O quinto e último capítulo inclui um estudo de caso que visa demonstrar todo o processo de criação da solução de BI utilizando a solução proposta e, desta forma, apresentar os resultados obtidos.

1 BUSINESS INTELLIGENCE

O atual interesse pelo BI vem crescendo assustadoramente na medida em que seu emprego possibilita às organizações realizarem uma série de análises e projeções, de forma a agilizar os processos relacionados às tomadas e decisão. É o que defende Howard Dresner, vice-presidente da empresa Gartner Group. (PRIMAK, 2008, p. 4)

O termo *Business Intelligence* surgiu na década de 80 no Gartner Group e faz referência ao processo inteligente de coleta, organização, análise, comportamento e monitoração de dados contidos em *Data Warehouse / Data Mart*, gerando informações para o suporte à tomada de decisões no ambiente de Negócios (PRIMAK, 2008, p. 5). Desde muito tempo as áreas de TI existentes estão atrás de meios para melhorar a forma como utilizar a informação existente nos ambientes organizacionais.

O setor corporativo passou a se interessar pelas soluções de BI de forma mais expressiva, principalmente no final de 1996, quando o conceito começou a ser espalhado como um processo de evolução do EIS (*Executive Information Systems*) – um sistema criado no final da década de 70, a partir dos trabalhos desenvolvidos pelos pesquisadores do MIT (*Massachusetts Institute of Technology* - EUA). (PRIMAK, 2008, p. 5)

Para obter as informações utilizadas pelo BI, pode ser necessário captar essas informações de diversas bases de dados diferentes como: ERP (*Enterprise Resource Planning*), planilhas eletrônicas, arquivos XML (*Extensible Markup Language*), base de dados consolidadas, etc.

A figura 1.1 apresenta a estrutura de uma solução de BI, destacando sistemas de origem, passando pelo processo de extração e transformação, onde após esse processo, os dados são carregados no repositório, para ser feita a exploração dos dados, através de ferramentas OLAP e/ou *Data Mining*.

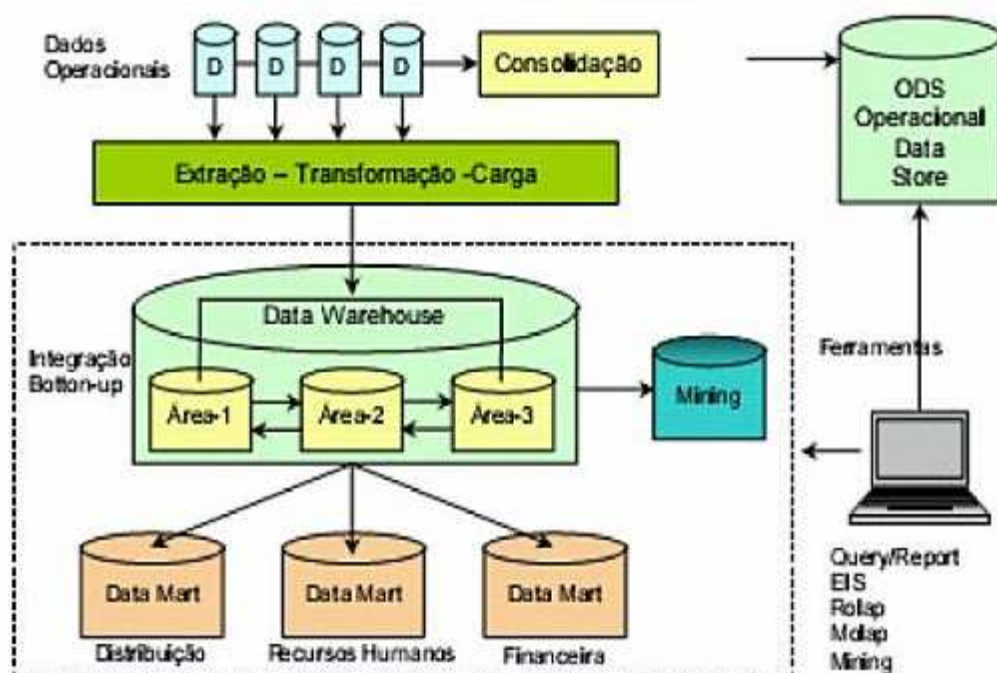


Figura 1.1 – Estrutura do ambiente BI

Fonte: Primak (2008)

1.1 Data Warehouse (DW)

Segundo Inmon (1999, p.13), o DW é um local onde as pessoas podem acessar suas informações de forma confiável e facilitada. As metas fundamentais de um DW podem ser desenvolvidas andando-se pelos corredores de uma organização de grande porte e ouvindo as conversas nas gerências. Os temas que se repetem em todos os diálogos são:

- Possuímos montanhas de dados nesta empresa, mas não conseguimos acessá-los;
- Nada deixa o chefe mais enfurecido do que duas pessoas apresentando o mesmo resultado do negócio, mas com números diferentes;
- Queremos acessar os dados de todas as formas;
- Mostre-me apenas o que é importante;
- Todos sabem que alguns dos dados não estão bons.

Para tratar de muitos desses problemas de dados, Bill Inmon fala do DW. Na verdade, Inmon é frequentemente chamado de o “pai do armazenamento de dados”. Em seu livro, *Building the data Warehouse* (John Wiley & Sons Inc., 1996), Inmon descreve o DW como um “conjunto de dados orientado para o assunto, integrado, não volátil, variante no tempo, no apoio de decisões gerenciais”. Em termos amigáveis, um DW é um conjunto de

instantâneos de dados, extraídos de sistemas de processamento de transação, em determinados intervalos (COREY, 2001). Explicando melhor cada um dos itens relacionados:

- **Orientado para o assunto:** As informações armazenadas no DW são aglomeradas por assunto de interesse do usuário ou da empresa do qual trabalha;
- **Integrado:** Todos os dados devem ser codificados da mesma forma, para não ocorrer divergências no seu processamento pelo DW;
- **Não volátil:** Os dados não são alterados com o passar do tempo. Eles são armazenados historicamente e são preservados dessa forma;
- **Variante do tempo:** As informações são retornadas conforme resultado de tempo solicitado pelo usuário.

Os DW são bancos de dados usados para a produção de informações relevantes de onde serão vista em relatórios. Isso contrasta com os bancos de dados subjacentes aos sistemas operacionais / captura de dados e sistemas de processamento de transação on-line (OLTP) tradicionais. (COREY, 2001)

1.1.1 Data Marts (DM)

O DM é um subconjunto do DW. O papel do DM é focar diretamente em setores departamentais de uma empresa como: estoque, PCP, vendas, compras, contabilidade. Segundo Primak (2008, p. 47), algumas organizações são atraídas aos DM não apenas por causa do custo mais baixo e um tempo de implementação menor, mas também por causa dos correntes avanços tecnológicos. São elas que fornecem um SAD customizado para grupos pequenos, de tal modo que um sistema centralizado pode não estar apto a fornecer. DM podem servir como veículo de teste para companhias que desejam explorar os benefícios do DW. A figura 1.2 exemplifica o funcionamento um DM.

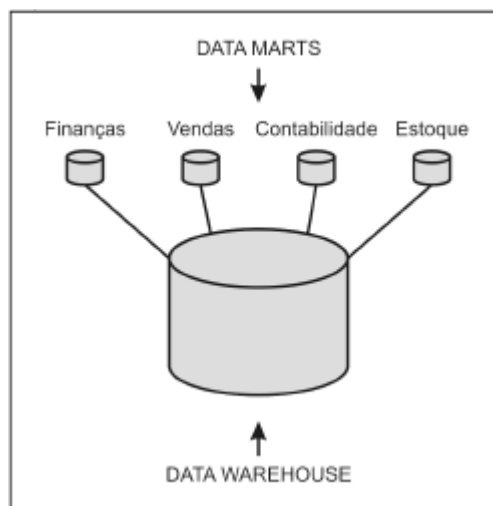


Figura 1.2 – Data Mart

Fonte: Inmon, Welch, Glassey (1999 apud FELTER, 2010, p.16)

1.1.2 Arquitetura DM e DW

Para efetuar a escolha certa no momento da implementação, é necessário saber qual é a amplitude da necessidade da empresa. É possível implementar um DM para testar o novo conceito e após sua utilização, é transformado o DM em DW.

Segundo Machado (2000 apud FELTES, 2010, p. 18), existem três tipos de arquiteturas para a plataforma: global, independente e integrada. A seguir a definição de cada uma das arquiteturas:

- **Arquitetura global:** é a plataforma que suporta por completo a utilização de um DW integrado, onde possa ser utilizado por qualquer departamento da empresa. Esse DW pode estar centralizado ou distribuído, conforme arquitetura implementada;
- **Arquitetura independente:** é a plataforma utilizada para atender a necessidade de um ou alguns departamentos específicos, nesse tipo de arquitetura é recomendado a utilização de um DM;
- **Arquitetura Integrada:** nessa arquitetura, existe a interligação de vários DM entre si de onde as informações são disponibilizadas entre departamentos da empresa onde a informação é necessária para ambos.

1.1.3 Diferenças entre ambiente operacional e ambiente DW

Segundo Machado (2000 apud FELTES, 2010, p. 20), conforme os tipos de ambiente de informação existentes nas empresas podem dividir a informação em duas categorias:

- **Dado operacional:** Esse tipo de informação é a utilizada para as funções de negócio propriamente ditas pela empresa;
- **Dado informativo (ambiente DW):** Essa informação é utilizada para a tomada de decisão perante os gestores da empresa.

O quadro 1.1 demonstra uma comparação de algumas características e as diferenças entre os dados operacionais e os dados informacionais.

Características	Banco de dados Operacionais	<i>Data Warehouse</i>
Objetivo	Operações diárias do negocio	Analisar o negócio
Tipo de processamento	OLTP	OLAP
Unidade de trabalho	Inclusão, alteração e exclusão	Carga e consulta
Número de usuários	Milhares	Centenas
Tipo de usuário	Operadores	Comunidade gerencial
Interação do usuário	Somente pré-definida	Pré-definida e <i>ad-hoc</i>
Volume	Megabytes – Gigabytes	Gigabytes - Terabytes
Histórico	60 a 90 dias	5 a 10 anos
Granularidade	Detalhados	Detalhados e resumidos
Redundância	Não ocorre	Ocorre
Estrutura	Estática	Variável
Manutenção desejada	Mínima	Constante
Acesso a registros	Dezenas	Milhares

Atualização	Contínua (tempo real)	Periódica (em <i>batch</i>)
Integridade	Transação	A cada atualização
Número de índices	Poucos / simples	Muitos / complexos
Intenção dos índices	Localizar um registro	Aperfeiçoar consultas

Quadro 1.1 – Diferenças entre bancos de dados operacionais e DW

Fonte: Primak (2008)

1.1.4 Granularidade

Conforme Inmon (1997 apud FELBER, 2006, p. 21), a mais importante questão de projeto que o desenvolvedor do DW precisa enfrentar, refere-se à definição da granularidade do DW, ou seja, o nível de detalhe ou de resumo dos dados existentes no DW. Quando a granularidade de um DW é apropriadamente estabelecida, os demais aspectos do projeto e implementação fluem tranquilamente; quando ela não é estabelecida, todos os outros aspectos se complicam.

À medida que o nível de granularidade aumenta, o número de consultas que podem ser atendidas diminui, sendo que uma granularidade mínima as consultas detalhadas podem ser respondidas. Portanto, é necessário encontrar um ponto de equilíbrio. O nível adequado de granularidade deve ser definido de tal forma que atenda as necessidades do usuário, tendo como limitação os recursos disponíveis.

A definição de granularidade de dados é a etapa mais importante do projeto de um DW, porque ela afeta profundamente o volume de dados que reside do DW, ao mesmo tempo, afeta o tipo de consulta que pode ser atendida. Devem-se definir níveis adequados de granularidade, de acordo com as necessidades do usuário.

A granularidade diz respeito ao nível de detalhe ou de resumo contido nas unidades de dados existentes no DW. Quanto mais detalhe, mais baixo o nível de granularidade. Quanto menos detalhe, mais alto o nível de granularidade. (INMON, 1999)

Existe a possibilidade de utilizar um nível duplo de granularidade (níveis duais de granularidade). Essa técnica se enquadra nos requisitos da maioria das empresas. São criadas duas camadas: uma camada para os dados levemente resumidos e outra para os dados históricos. Com a criação de dois níveis de granularidade, é possível atender a todos os tipos

de consultas, com um melhor desempenho, visto que a maior parte do processamento analítico dirige-se aos dados levemente resumidos, que são compactos e de fácil acesso. Para as ocasiões em que um maior nível de detalhe deve ser analisado, existe o nível de dados históricos, o qual é complexo e de alto custo. A figura 1.3 ilustra essa granularidade.

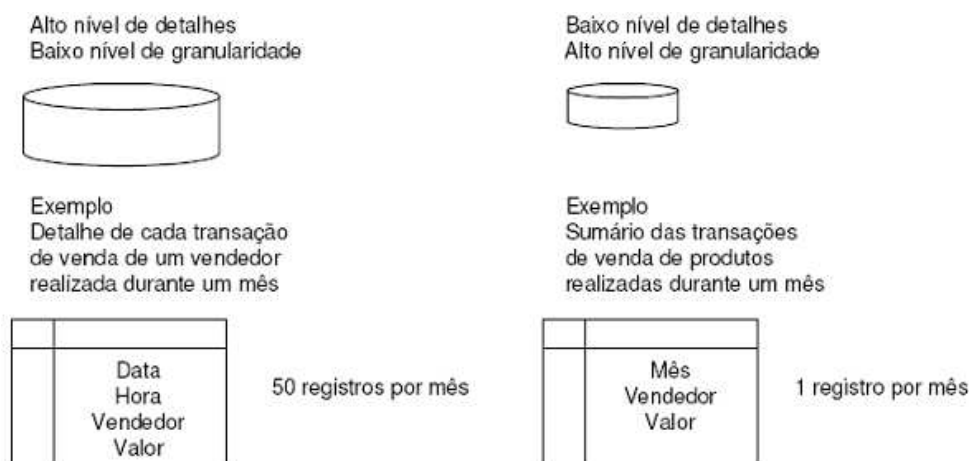


Figura 1.3 – Granularidade dos dados em um DW

Fonte: Machado (2000 apud FELBER, 2006, p.22)

1.1.5 Métodos de implementação do DW

Segundo Machado (2000 apud SANTOS, 2009, p. 18), a escolha por um tipo de implementação é influenciada por diversos fatores, como a infra-estrutura de TI, arquitetura escolhida, recursos disponíveis, velocidade de implementação, investimento desejado e pela necessidade ou não do acesso corporativo dos dados. Os principais tipos de implementação do DW são:

- **Top-Down:** nesse tipo de implementação, deve-se primeiro tomar as decisões sobre as fontes de dados, segurança, estrutura, qualidade e padrões de dados significando um trabalho inicial oneroso pelo planejamento e definições conceituais para a construção do DW. Após construído o DW, os dados e metadados são extraídos para os DM. Esse tipo de implementação tem como vantagem a visão geral dos dados da empresa, facilitando a manutenção de repositório de dados centralizado. Porém, como desvantagem, requer um longo período de implementação;
- **Bottom-Up:** nessa abordagem primeiramente são construídos os DMs departamentais, que vão sendo gradativamente incrementados ao DW. Esse processo tem como vantagem uma rápida implementação e consequentemente

um rápido retorno e a opção de escolher os principais departamentos da empresa para a criação dos DMs. Tem como desvantagem a falta de padronização dos DMs que pode vir a gerar redundância e inconsistência de dados;

- **Implementação combinada:** essa forma de implementação tem o propósito de integrar as duas arquiteturas, realizando a modelagem dos dados do DW com uma visão macro e gerar os DMs a partir do macro modelo de dados do DW. Essa abordagem tem como vantagem a garantia da consistência dos dados, pelo fato de se utilizar um modelo de dados únicos para os DMs.

1.2 Modelagem dimensional dos dados

Conforme Kimball (1998, p.9), a modelagem dimensional permite visualizar os dados de uma forma concreta e tangível. Esse é o segredo da compreensibilidade. Na qualidade de projetistas de DW, “Vendemos Produtos em vários Mercados e avaliamos nosso desempenho ao longo do Tempo”.

Muitas pessoas consideram fácil imaginar a informação como um cubo, onde cada dado do cubo esteja em suas arestas e com um simples clique mover essas informações da forma que lhe for mais adequado.

O ponto forte em um modelo dimensional é sua simplicidade. Simplicidade é a chave fundamental que permite aos usuários entender o banco de dados e permite que o software navegue pelo banco de dados com eficiência. De várias formas, o processo de modelagem dimensional visa “proteger o forte” e simplificar o conceito proposto. Retornando continuamente à perspectiva *top-down* e recusando comprometer as metas de um projeto coerente de banco de dados que atenderá as necessidades do DW.

A figura 1.4 ilustra uma tabela fato e suas dimensões para um melhor entendimento do conceito. Essa imagem mostra a distribuição da tabela fato sobre a modelagem dimensional.

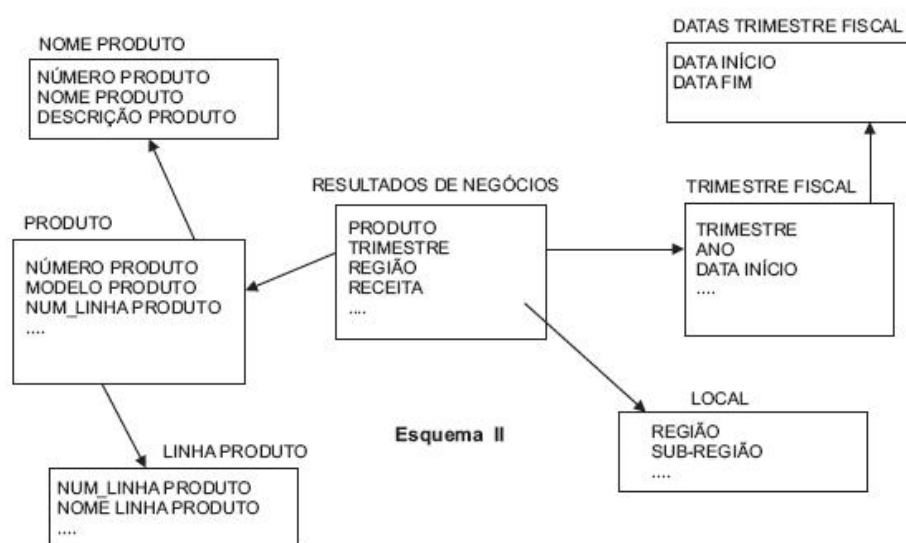


Figura 1.4 – Tabela fato e tabelas dimensões
Fonte: Adaptado conforme autor Kimball (2008)

1.2.1 Tabela fato

De acordo com Kimball (2008), as tabelas fato correspondem as medidas numéricas referentes a situação de negócio realizada sobre ela. Ela armazena as chaves de informação de várias tabelas dimensionais de onde essa informação cruzada é fundamental para a área de negócio decorrente da empresa. Geralmente essas tabelas correspondem ao processo de negócio da empresa, de onde são utilizadas para a emissão de relatórios com a finalidade de conferir ou até mesmo gerar procedimentos vitais para o funcionamento da área de negócio.

Tabelas fato podem ter milhões ou bilhões de linhas de informação, com muita eficiência de performance porque a maioria das tabelas fato utilizam em torno de 90% de informações vindas das tabelas dimensionais, fazendo com que essas informações não fiquem redundantes.

1.2.2 Tabela dimensão

De acordo com Kimball (2008, p.241), as tabelas dimensionais são responsáveis por armazenar as informações de uma forma mais detalhada onde possam existir N informações nela, tanto numéricas como alfanuméricas, informações recentes ou passadas. Sua estrutura é (1:N) possibilitando um melhor aproveitamento do processamento das informações para as tabelas fato.

Tabelas dimensão possuem dezenas de atributos de todos os tipos, podendo armazenar informações das mais diversas desde informações chave para o negócio da empresa, até informações relativamente menos importantes onde não interfiram diretamente no processo de negócio da empresa.

1.2.3 Modelo *Star*

De acordo com Singh (2001 apud FELTES, 2010, p. 22), a principal característica do Modelo *Star*, também chamado de Modelo Estrela, são suas dimensões desnormalizadas. As vantagens na utilização desse modelo são a redução de *joins* físicos entre as tabelas dimensões no momento da consulta, melhorando muito o desempenho e a simplicidade da visualização do modelo de dados, permitindo a expansão e evolução do DW com pouca manutenção.

O Modelo *Star* é a estrutura básica de um modelo de dados dimensional. Sua composição possui uma grande entidade central chamada de fato e um conjunto de entidades menores chamadas de dimensões, organizadas ao redor da tabela fato formando uma estrela. O relacionamento entre a entidade fato e as dimensões é uma ligação entre as duas entidades com um relacionamento de um para muitos no sentido da dimensão para o fato. A figura 1.5 ilustra esse modelo.

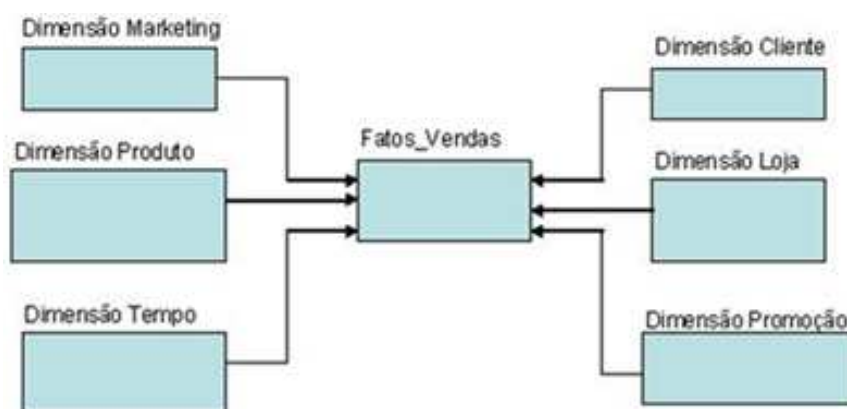


Figura 1.5 – Modelo em Estrela
Fonte: Machado (2000 apud FELTER, 2010)

1.2.4 Modelo *Snowflake*

Conforme Singh (2001 apud FELTES, 2010, p. 23), o modelo *Snowflake*, também chamado de Modelo Floco de Neve, é o resultado da terceira forma normal sobre as tabelas dimensões do Modelo *Star*. Cada tabela dimensional armazena uma chave para cada elemento da dimensão, formando uma hierarquia. Tem como vantagem a normalização, evitando a redundância de valores textuais na tabela dimensão.

Esse modelo tem como desvantagem sua complexidade da estrutura de dados normalizada, dificultando os usuários na navegação das consultas e os programas de carga e manutenção podem ficar mais difíceis de administrar de acordo com o aumento da complexidade. A figura 1.6 ilustra esse modelo.

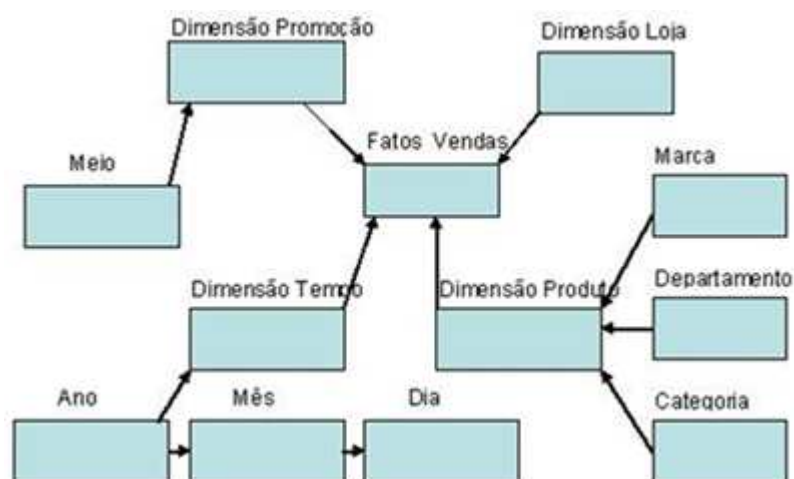


Figura 1.6 – Modelo Floco de Neve
Fonte: Machado (2000 apud FELTER, 2010)

1.2.5 Metadados

Conforme Inmom (1999, p.73), os metadados compõem o centro nervoso do DW. Sem os metadados, o DW e seus componentes associados no ambiente projetado são meramente componentes soltos funcionando independentemente e com objetivos separados. A fim de alcançar a harmonia e unidade dos diferentes componentes do ambiente projetado, deve haver uma abordagem bem definida e disciplinada para metadados.

É pelo metadado que é possível localizar todas as informações existentes no DW e organizá-las de uma forma simplificada, organizada, sintetizada e coerente podendo facilitar a vida do usuário para a construção de suas consultas. É organizado o ambiente operacional, a camada de integração e transformação, o ODS (*Operacional Data Store*), o nível de detalhe corrente do DW, o DM, o ambiente de desenvolvimento e a modelagem de negócios abrangida.

O ambiente operacional em geral contém grande quantidade de processamento de transações de alta *performance*. O ambiente ODS contém uma mistura de processamento sequencial DSS e atualizações de alta *performance*. O nível corrente de detalhe dos DW baseia-se em softwares que atuam sobre grandes quantidades de dados e suportam explorações de dados. O ambiente DM suporta acesso e exploração de dados imediatos. Do

ponto de vista de software, há uma grande diversidade de tecnologias que atendem os diversos componentes do ambiente projetado.

Podem ser distribuídos por toda a rede de objetos que compõem a infra-estrutura e serem tecnologicamente compatíveis e disponíveis para todo o sistema local abrangente da estrutura desenvolvida.

Os metadados são certamente úteis no mundo operacional. Contudo, tendo como ênfase o mundo operacional do ponto de vista do usuário, a comunidade de funcionários operacionais da empresa. Essa comunidade executa as mesmas transações repetidas vezes, dia após dia. Já o analista de DSS enxerga essa mesma informação de uma forma totalmente diferente, como um todo. A grande diferença entre ver os metadados de usuário para o analista DSS é o quanto dessa informação é vista, o usuário vê um processo e executa-o diversas vezes ao dia, já o analista tem como conhecimento toda a sua estrutura.

1.3 Extração, transformação e carga (ETL)

Conforme Corey (2001, p.226), ETL é um processo de extração, transformação e carregamento de informações da base de dados original para o DW. Quando e onde deve-se usar a tecnologia ETL? Essa pergunta tem sido feita frequentemente por muitos profissionais de DW e, da mesma forma, por gerentes e desenvolvedores. A resposta é: sempre. Independente do que é feito, deve-se extrair, transformar e carregar dados de alguma maneira. ETL não é um conceito novo. Nada mais é do que uma série de lotes de interface entre sistemas, e as interfaces de lote já existem há bastante tempo.

O que torna ETL difícil atualmente é a maior ênfase na captura e consolidação da inteligência empresarial, em conjunto com o processamento em lote de dados. Os objetivos do DW podem ser consideravelmente mais complexos do que aqueles do processamento em lote tradicional, graças a mais essa dimensão. Então, a pergunta real não é apenas “quando você usa tecnologia ETL”, mas também “como consolidar a inteligência empresarial?”.

Segundo a experiência do autor, bons programadores podem escrever bons processos de ETL. E geralmente podem fazer melhor do que qualquer ferramenta ETL. Mas aos processos resultantes falta um repositório central para armazenar os metadados técnicos necessários para descrever seu processamento ou torná-lo reutilizáveis. Por outro lado, uma ferramenta ETL reúne dados sobre os processos ETL, os torna reutilizáveis, é mais fácil de

gerenciar e transferir conhecimento, mas pode não ter parte do poder e da complexidade necessários para processar as transformações eficientemente.

De acordo com Kimball (2008, p.127), o processo ETL não se restringe somente a extrair, transformar e carregar as informações, é necessário também outros processos para que a rotina não gere problemas no DW como:

- **Limpeza e conformidade dos dados:** é necessário que algum algoritmo trabalhe os dados, não deixando com que sujeiras de informação passem pelo ETL e também se as informações estão em conformidade com a proposta do DW;
- **Entrega dos dados:** é necessário que rotinas internas verifiquem se todas as informações geradas pela fonte de dados dos ambientes operacionais (ERP, planilhas, etc), sejam processadas pelo ETL e posteriormente carregadas ao DW;
- **Tipos de fontes de dados:** importante avaliar as fontes de dados na carga. Quando os dados forem transformados, é necessário que toda a informação esteja padronizada para não ocorrer conflitos;
- **Valores *default*:** há condições onde o dado não possua informação no processamento de entrada, onde é necessário definir uma informação padrão para esses casos;
- **Resumir dados:** todas as informações geradas pelos ambientes operacionais são necessárias, mas é necessário resumir essas informações de uma forma que seja melhor processá-las internamente;
- **Tabela fato e tabela dimensão:** é necessário saber trabalhar corretamente com as informações dessas duas estruturas, por serem de vital importância na consistência da informação do DW;
- **Múltiplos valores:** há situações onde tabelas possuem diversos valores, é necessário agrupar essas informações e saber tratar se realmente são necessárias no DW.

A figura 1.7 apresenta um modelo ETL com todas suas etapas.

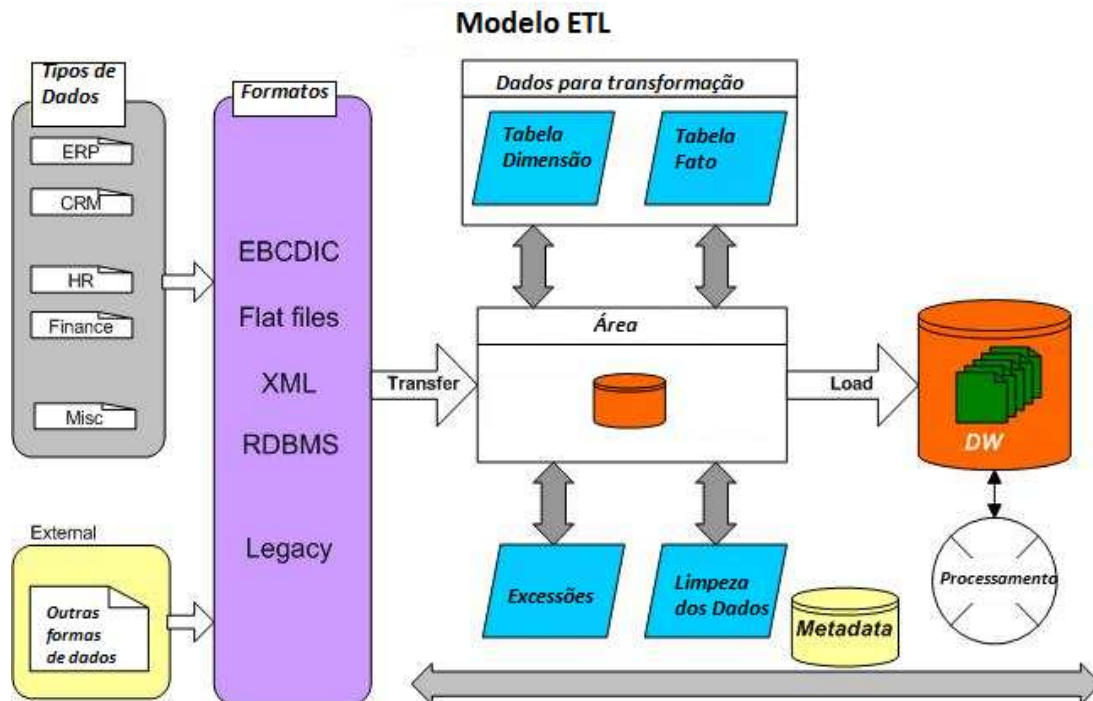


Figura 1.7 – Modelo ETL

Fonte: Adaptado pelo autor segundo Corey (2001)

Conforme Corey (2001, p.229), quando se planeja na estrutura do ETL, alguns fatores devem ser avaliados antes da utilização ou desenvolvimento da ferramenta como:

- Armazenados em esquemas diferentes;
- Armazenados em tipos de arquivo ou banco de dados diferentes;
- Fisicamente espalhados por vários centros de dados, em vários países;
- Armazenados em diferentes plataformas de hardware, executando diferentes sistemas operacionais.

E quando essas informações forem todas processadas pelo ETL, elas irão ser armazenadas da seguinte forma:

- Estão armazenados em uma estrutura comum;
- Estão armazenados no mesmo tipo de banco de dados;
- Estão localizados no mesmo centro de dados e possivelmente no mesmo banco de dados.
- Compartilham a mesma plataforma de hardware e o mesmo sistema operacional.

O desenvolvimento do ETL é algo muito complexo por se tratar de vários fatores para sua estruturação. Existem casos onde o processo de trabalhar os metadados e a

conectividade ao sistema é mais simples com ferramentas ETL prontas do que as desenvolver, mas há casos onde transformações de dados e agregações como o ETL manual possui uma média de ganho melhor que a ferramenta pronta. Sempre que se tratar da utilização do ETL, é necessário avaliar bem que situação se enquadra sua base de dados para a utilização da forma que melhor se adéqua a situação.

1.4 On-line analytic processing (OLAP)

Conforme Thomsen (2002), os conceitos de OLAP incluem a noção ou ideia de múltiplas dimensões hierárquicas e podem ser usados por qualquer um para que se pense mais claramente a respeito do mundo. Podendo esclarecer de uma forma simplificada uma situação decorrente. Em outras palavras, mesmo sem qualquer tipo de linguagem formal, é útil apenas conseguir ter capacidade de pensar em um mundo multidimensional e com múltiplos níveis, independente de sua posição na vida. A figura 1.8 ilustra um modelo de uma consulta OLAP.



Figura 1.8 – Modelo OLAP

Fonte: Thomsen (2002)

Para conseguir ter acesso a múltiplos dados de várias bases diferentes, é necessário processar essas informações de uma forma única sem que haja a preocupação do usuário de

saber de onde vêm esses dados, se o banco de dados encontra-se perto de onde se está ou não, ou se existe mais de um banco de dados para as informações consultadas por mim, isso sim é o significado de OLAP. (KIMBALL, 2008, p. 167)

De acordo com Inmon (1999, p. 177), um conjunto de regras de OLAP acabou surgindo de uma variedade de fontes. Muitas dessas supostas regras representam na verdade um guia para fornecedores de ferramentas de análise e outros produtos, em geral chamados de servidores OLAP, e não são específicos a um tipo de dado. Essas regras que podem ser aplicadas a uma arquitetura de dados fornecida pelo DW. Como a arquitetura do DW suporta essas regras de OLAP? Os itens a seguir demonstram isso:

- **Visão conceitual dimensional:** o nível atômico do DW que possibilita capturar todos os atributos de qualquer área de assunto que seja adequada e necessária para dar suporte ao processamento DSS para a organização;
- **Transparência:** acesso a qualquer nível do DW, incluindo metadados e acesso a qualquer tipo de atributo disponível pelos dados do DW pelo usuário;
- **Acessibilidade:** poder unir todos os dados de diversas bases fisicamente separados ou não e poder estruturá-las de uma forma que satisfaça as necessidades de informação da organização ao longo dos diferentes níveis de arquitetura;
- **Performance de relatório consistente:** consistência das informações extraídas pelo DW podendo auxiliar na manutenção da precisão de dados históricos;
- **Arquitetura cliente-servidor:** dados são adquiridos, acessados e processados através da arquitetura cliente-servidor de onde é independente da plataforma utilizada;
- **Dimensionalidade genérica:** onde todos os atributos que poderão ser agrupados logicamente para serem vistos dimensionalmente, são representados como genéricos;
- **Operação dimensional cruzada irrestrita:** também conhecida como *drill-across*, é a forma de navegação da tabela dimensão para outro nível de tabela sem passar pelos níveis intermediárias;

- **Manipulação de dados intuitiva:** é a manipulação dos metadados de uma forma intuitiva pelo usuário que detém informação suficiente sobre a área de negócio de maior entendimento;
- **Flexibilidade quanto a relatórios:** flexibilidade na forma pelo qual pretende-se utilizar ou agrupar as informações da base pelo seu entendimento;
- **Dimensão e níveis de agregamento ilimitados:** poder modificar de todas as formas possíveis para satisfazer as necessidades de dados e informações da organização;
- **Pesquisa de detalhes (*drill down*):** permite a visão dos dados ao longo de todos os níveis de uma dimensão, permitindo navegar-se do nível mais alto até o nível mais detalhado;
- **Atualização incremental de banco de dados:** suportar ao longo do tempo do DW que os dados sejam periodicamente adicionados, carregados, anexados, inseridos ou atualizados, dependendo das características da informação e não ser feita uma carga completa da base cada vez que houver uma alteração de informação efetuada;
- **Arrays múltiplos:** poder satisfazer as mais diversas necessidades dos usuários com informações redundantes do qual se julga necessária;
- **Seleção de subconjuntos:** poder selecionar subconjuntos de informação da forma que for mais adequada para o usuário final;
- **Suporte a dados locais:** suporte a implementação descentralizada de informações pelo qual são julgadas importantes para o usuário, podendo ser gerenciadas pelos metadados desenvolvidos.

1.4.1 Arquiteturas OLAP

De acordo com Thomsen (2002, p. 7), o OLAP possui várias formas de arquitetura diferente para compor toda a sua estrutura. Abaixo, algumas de suas formas mais tradicionais:

- **ROLAP (*Relation On-Line Analytical Processing*):** é a arquitetura OLAP relacional que utiliza a própria estrutura do SGBDR como as tabelas construídas para estruturar os relacionamentos mais tradicionais. Oferece como vantagens toda a estrutura do SGBDR sobre a *performance* das análises efetuadas;

- **MOLAP (Multidimensional On-Line Analytical Processing)**: essa arquitetura utiliza dos princípios da arquitetura multidimensional dos bancos de dados proprietários;
- **HOLAP (Hybrid On-Line Analytical Processing)**: essa arquitetura utiliza princípio híbrido entre o MOLAP e o ROLAP, onde a estrutura gerencia os dados para melhor os processá-los. Quando a granularidade é maior, daí se usa a estrutura ROLAP, já quando a estrutura tem granularidade menor é utilizado a arquitetura MOLAP;
- **DOLAP (Desktop On-line Analytical Processing)**: essa arquitetura desfruta dos princípios de todas as arquiteturas mencionadas anteriormente, como relacional e dimensional, que devem ser transferidas para a estação de trabalho do cliente, podendo ter melhor *performance* para consultas e redução no tráfego de rede e processamento de servidor de dados.

Em sequencia, uma imagem de como funciona essas arquiteturas do OLAP.

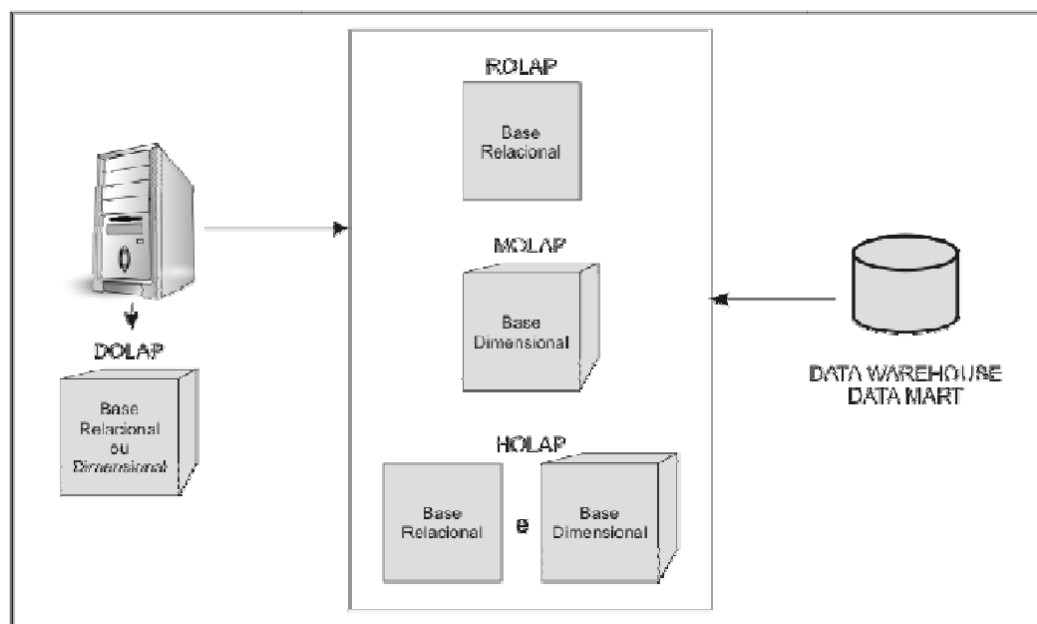


Figura 1.9 – Tipo de Arquitetura OLAP
Fonte: Barbieri (2001 apud FELTER, 2010, p. 27)

Um conceito muito necessário para ser avaliado é como esses dados serão visualizados pelo usuário final, já que a preocupação principal do OLAP é repassar as informações corretas para os usuários finais onde se tenha algum ganho de necessidade como isso. Existem duas formas de visualizar as informações: visualização análoga e visualização simbólica.

- **Visualização Análoga:** forma de visualizar as informações da forma como realmente são dispostas, sem se reter a imagens ou representações visuais de itens. Essa forma captura o máximo de informações para que possa ser visto, podendo ser chamadas de realistas;
- **Visualização Simbólica:** definição se enquadra em visualizar informações em formas de gráficos, linhas, tabelas, da forma que comercialmente melhor se demonstre a situação ocorrida no momento.

Os casos do OLAP são situações de visualização simbólica onde há necessidade do seu uso para melhor encontrar informações importantes para o ramo de negócio da empresa.

2 ERP SAFETECH

O Gestão SFT é um sistema ERP criado pela empresa Safetech Informática LTDA, com o intuito de proporcionar um controle administrativo e gerencial simplificado e abrangente. Ao implantar o Gestão SFT são agregados aos negócios do cliente conceitos de gestão, integração da empresa, velocidade e flexibilidade na análise de informações. Concebido através da plataforma de desenvolvimento *Oracle*, possui uma enorme flexibilidade, permitindo seu funcionamento em diversos ambientes operacionais, o que garante o acompanhamento da evolução tecnológica sem perda do investimento inicial.

O Gestão SFT trabalha com uma estrutura multicamadas, permitindo assim a independência entre a aplicação, banco de dados, sistema operacional e rede. Pode ser considerado multi-plataforma, pois pode ser utilizado em ambientes *Windows* ou *Linux*.

2.1 SAFETECH

A empresa possui clientes nos estados do Rio Grande do Sul, Paraná, São Paulo, Bahia, Mato Grosso do Sul e há operações também na Argentina. Seu nicho de mercado está mais especificadamente no Vale dos Sinos, Vale do Paranhana e região metropolitana de Porto Alegre. Atuando com cerca de 211 clientes em atividades diversas de negócio, totalizando mais de 1300 usuários que utilizam o Gestão SFT no seu dia-a-dia. A figura 2.1 demonstra a distribuição da carteira de clientes por atividade.

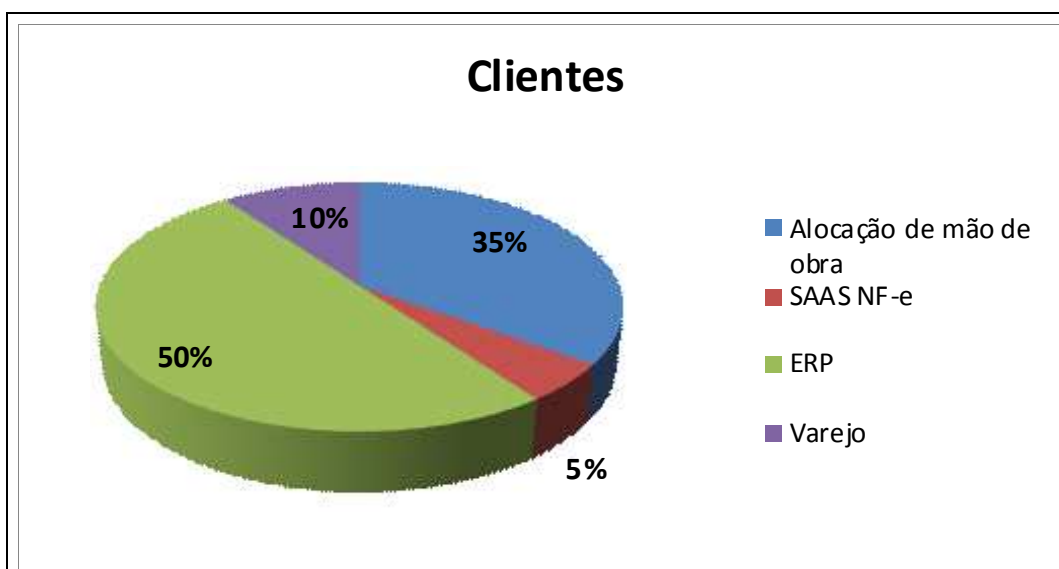


Figura 2.1 – Distribuição da carteira de clientes por atividade

Fonte: Safetech Informática (2012)

Atualmente a estrutura organizacional da empresa está dividida em cinco grandes áreas (Direção, Controladoria, Suporte, Consultoria e Desenvolvimento) sendo a direção responsável pela estratégia, visão e administração da Safetech Informática e a controladoria responsável pelo monitoramento funcional e operacional da empresa e seu relacionamento com o mercado.

O suporte é responsável pelo canal direto e aberto permanentemente nos relacionamentos com os clientes e mercado e a manutenção do uso do sistema. A consultoria é responsável por reconhecer o processo de gestão das empresas e implantar o ERP. O desenvolvimento é a divisão responsável pelo planejamento e execução dos projetos de desenvolvimento de sistema, relacionando demanda e novas tecnologias.

2.2 Gestão SFT

Concebido em uma arquitetura modular composta por trinta e oito módulos com funções organizacionais bem distintas, o Gestão SFT permite sua utilização por clientes de diversos ramos de atividade, atuando em áreas como: comércio, indústria e prestação de serviço. Como exemplos destes clientes, podem-se citar lojas, indústrias, manutenção de computadores, escritórios, entre outros. O cliente Safetech pode adquirir pacotes de módulos de acordo com a sua evolução e complexidade organizacional.

O fato de ser composto por módulos permite o Gestão SFT ser implementado por etapas, de acordo com a capacidade de investimento de cada cliente, apresentando os resultados de forma gradual. Desta maneira, mesmo as empresas que iniciam apenas com os módulos básicos podem ter acesso às informações gerenciais, tirar proveito dos conceitos de gestão que automaticamente são implementados junto com o ERP. Apesar de ser composto por módulos individuais, o Gestão SFT possui uma única interface, o que permite uma fácil navegação entre os módulos e uma redução no tempo de aprendizado e nos custos de implementação. A figura 2.2 ilustra os módulos do ERP Safetech e seus relacionamentos.



Figura 2.2 – Módulos do ERP Safetech
Fonte: ERP Safetech (2012)

2.2.1 Principais Recursos

As sofisticadas técnicas do Gestão SFT permitem personalizar regras e funções de acordo com as necessidades de cada negócio. Alguns dos recursos oferecidos são:

- **Financeiro:** gerencia o setor financeiro da empresa com visões em contas do tipo financeiras, contábeis e gerenciais, permitindo acompanhamento histórico com síntese financeira ou previsão com fluxo financeiro em longo prazo, automatizando as operações e decisões do dia-a-dia, e muitas outras informações gerenciais importantes no processo da tomada de decisão;
- **Comercial:** gerencia as vendas, validando datas de entrega, previsão financeira, limites de crédito, saldos por pedido e cliente, garantindo o processo e a administração da carteira de vendas. Automatiza o processo de expedição, carga, emissão de etiquetas, separação dos produtos, rota de entrega, gerando romaneio e as notas fiscais automaticamente de acordo com o planejamento de entrega, otimizando toda logística da empresa. Também otimiza o processo de embalagem, montagem de carga e entrega, resultando na emissão das notas fiscais e no manifesto de carga no momento do embarque;
- **Industrial:** gerencia as solicitações, cotações, ordens de compras e notas de entradas de forma prática e otimizada, evitando retrabalhos no processo. Permite ao comprador um acesso ágil das informações gerenciais que necessita para o bom planejamento de compras da empresa. Planejar a produção, realiza análise

de demandas de pedidos e parâmetros de reposição, gerando ordens de compra e produção em função das regras e parâmetros universais do MRP (*Material Requirement Planning*). Agiliza o processo de análise de necessidade e liberação de ordens. Sua velocidade no planejamento simplifica e automatiza o processo produtivo, eliminando falhas e atrasos no processo produtivo, integrando os departamentos de produção, compras e suprimentos. Gerencia o estoque físico e financeiro da empresa, desde a matéria-prima até o produto pronto. O controle e a classificação do estoque podem ser totalmente parametrizados de acordo com os critérios previamente estabelecidos. Este processo possibilita uma administração de materiais dinâmica e otimizada;

- **Administrativo:** atua como um instrumento de abrangência e flexibilidade na escrituração e gerenciamento das informações contábeis das empresas. Oferece recursos múltiplos para maior produtividade dos usuários, e eficiência na administração e gerenciamento contábil. Gerencia e faz acompanhamento da complexa legislação fiscal brasileira, permitindo a geração das obrigações fiscais, relatórios e arquivos legais. Permite também o controle patrimonial da empresa, atuando como uma ferramenta abrangente e flexível na administração e gerenciamento do patrimônio da empresa. Agregando valor aos investimentos e potencializando os resultados dos ativos operacionais.

2.3 Origem dos dados para análise

Para atender as análises mencionadas anteriormente, será realizada a carga do *Data Mart* de cobrança de clientes com informações do ERP Safetech. A estrutura de tabelas do Gestão SFT foi construída visando evitar a redundância de informações. Por este motivo, mesmo se tratando de um DM de faturamento, serão necessárias informações provenientes de diferentes módulos do sistema, tais como: Administrativo, Industrial, Financeiro, Comercial e tabelas gerais.

Através do módulo de Faturamento é possível realizar a emissão de Notas Fiscais de Saída, Cupons Fiscais e Pedidos de Venda. Além de permitir o cadastro de mercadorias e múltiplas comissões para as notas fiscais. Quando uma nota fiscal é cadastrada, os impostos que incidem sobre a mesma são calculados automaticamente pelo sistema, de acordo com as regras de cálculo previamente definidas pelo usuário. Diretamente do cadastro de notas fiscais, é possível realizar a geração das duplicatas que serão enviadas aos clientes. Assim

sendo, a partir deste módulo serão obtidas informações referentes aos Pedidos de Venda, Notas Fiscais de Saída, comissões e regiões de mercado.

As duplicatas geradas através da nota fiscal ficam armazenadas no módulo de Gestão Financeira. Este módulo permite um completo controle financeiro da empresa, pois verifica as disponibilidades financeiras através do fluxo financeiro, controla contas correntes e caixa, contas a pagar e a receber, emite relatórios em quaisquer moedas, boletos bancários, emissão de cheques e liquidação dos títulos, além de outras inúmeras funcionalidades. Desta forma, a partir deste módulo serão obtidas informações sobre as duplicatas geradas através das notas fiscais, limites de crédito e histórico financeiro dos clientes.

Os produtos vendidos aos clientes e as matérias-primas utilizadas na produção dos mesmos são cadastrados através do módulo de Gestão Industrial, o qual é responsável pelo controle do estoque, desde a matéria-prima até o produto pronto. Este módulo possui inúmeras funcionalidades, como por exemplo, cadastro de tabelas de preços para os materiais, variando por cliente. Também é possível cadastrar movimentos de estoque, ou seja, entradas e saídas para o material, a fim de, ajustar a quantidade disponível em estoque. Por isso, deste módulo serão obtidas informações referentes aos materiais, tais como: características, saldo em estoque, movimentações por cliente, principais fornecedores e preços de tabela.

A Safetech trabalha com a filosofia de Sistema Integrado, por isso muitos módulos utilizam a mesma tela de cadastro para entidades comuns, utilizadas por mais de um módulo do sistema. Estes cadastros estão centralizados em um módulo chamado Genéricos. Dentre estas tabelas comuns, podem ser citadas: Pessoas, Cep, Estados, Municípios, Tipos de operação, dentre outros. O cadastro de pessoas merece atenção especial, pois no Gestão SFT, uma pessoa (física ou jurídica) pode ser um cliente, um fornecedor, um funcionário, uma transportadora, um representante, ou outra entidade qualquer com essas características. Desta forma, o Gestão SFT permite que uma mesma pessoa seja, ora cliente, ora fornecedor. Sendo assim a partir do módulo Genéricos serão obtidas informações referentes às empresas, estados, municípios, regiões, dentre outras.

3 ANÁLISE DE SOLUÇÕES DE BI

Para a análise da solução proposta, houve diversas pesquisas sobre várias ferramentas para construção de soluções de BI existente no mercado, tais como: *Business Process Analysis Suite*, *Business Process Management* e *BI Spreadsheet Add-in*. Muitas dessas ferramentas são *free* e outras são de propriedade comercial, de onde muitas delas atendem a necessidades existentes no mercado para esse tipo de solução.

Já essa solução foi construída sobre o princípio de alocar recursos disponíveis pela *Oracle* por ser a plataforma pelo qual a empresa dispõem e tem licença para seu uso integral. Para a escolha das ferramentas, escolheram-se as ferramentas a seguir por serem as opções recomendadas pela *Oracle* como um todo, desde os recursos de ETL, DM até o OLAP utilizado como referência.

Todos os testes foram realizados utilizando um DM com diversos módulos do ERP Safetech para testar suas funcionalidades, tanto das ferramentas de análise do processo ETL, como os recursos do módulo OLAP.

Entre as várias soluções de BI existentes no mercado, o *Oracle Warehouse Builder* e o *Oracle BI Administration Tool* foram escolhidos para a análise, a fim de identificar determinadas funcionalidades, que vão servir como base para a construção da arquitetura proposta.

As ferramentas pesquisadas dos quais não foram utilizadas para a demonstração desse trabalho não serão citadas por não existir a necessidade de compara-las diretamente com os recursos utilizados.

3.1 ETL

A ferramenta *Oracle Warehouse Builder* foi escolhida para a análise, por agregar ao conjunto de soluções propostas no projeto, uma forma mais simplificada e de fácil interação com a arquitetura proposta.

3.1.1 Oracle Warehouse Builder

A ferramenta *Oracle Warehouse Builder* é um dos componentes da plataforma *Oracle* mais completo em gerenciar processos de ETL. A ferramenta pode ser utilizada tanto

como parte do processo de BI, como para outras finalidades, como a migração de dados entre servidores, exportação de dados e integração de aplicativos. (ORACLE, 2012)

A instalação da ferramenta é simples, de onde é necessário descompactar o arquivo e efetuar seu processo de instalação do qual é constituído por poucas informações para o usuário que a utiliza. A ferramenta possui versões de instalação tanto para plataforma Windows como Linux. É necessário configurar as conexões JDBC do banco de dados para efetuar corretamente a conexão da ferramenta ao sistema.

O *Oracle Warehouse Builder* possui vários recursos para serem utilizados em rotinas de ETL, conforme identificados na figura 3.1.



Figura 3.1 – Opções do ETL

Fonte: Oracle (2012)

Com o *Oracle Warehouse Builder* é possível importar dados de diversas fontes, como por exemplo: arquivos CSV, arquivos texto, tabelas de banco, XML, planilhas eletrônicas, etc.

Existem muitos recursos do produto que permitem a integração de dados e a modelagem para a maioria dos ambientes. São entrada do mundo do *Oracle Warehouse Builder* porque permitem utilizá-lo sem custos adicionais. A seguir, alguns desses recursos:

- **Opcional Enterprise ETL:** este complemento é voltado especificamente para aumentar a *performance* (permitindo extrações de alta performance) e produtividade, com grande capacidade de reutilização. Alguns recursos

avançados de gerenciamento de metadados também são incorporados a esse opcional;

- **Opcional *Data Quality***: sendo parte integrante do produto oferecendo um grande diferencial em comparação com as ferramentas da concorrência, o complemento *Data Quality* oferece suporte a geração de perfis de dados, regras de dados (basicamente, regras de negócios) e recursos de *compliance* das informações. Devido à integração, as informações coletadas durante a geração de perfis de dados podem ser usadas para gerar automaticamente as correções de dados;
- **Conectores**: Os conectores permitem melhor acesso e maior integração com os pacotes de aplicativos do ERP e CRM.

A figura 3.2 irá ilustrar a integração do produto disponibilizado pela ferramenta.



Figura 3.2 – Integração do produto
Fonte: Oracle (2012)

Sua utilização é simples e intuitiva, pois utiliza serviços *Warehouse Builder* *Dependency Management* permitindo revelar o conteúdo em efeito dominó das alterações dos metadados. A figura 3.3 mostra a interface para montar o mapeamento do sistema ETL.

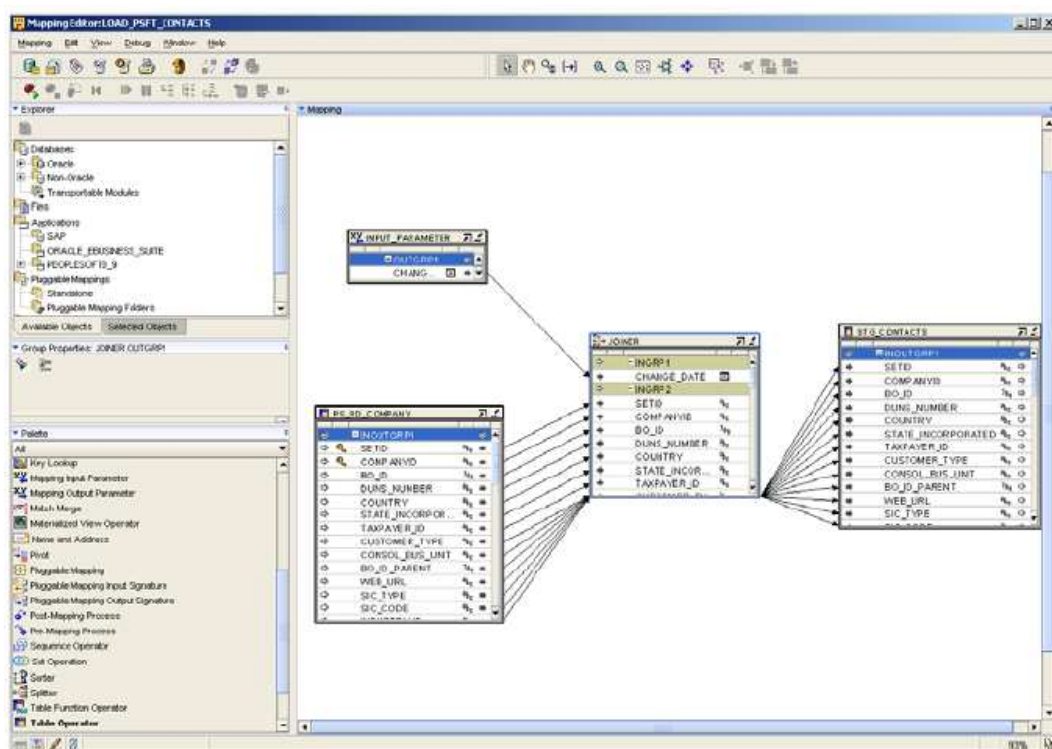


Figura 3.3 – Propagando alterações no sistema
Fonte: Oracle (2012)

Outro componente de vital importância para tornar as informações do DM em informações necessárias ao gestor que as utiliza é garantir os níveis de qualidade dos dados. O componente para a qualidade das informações no *WareHouse Builder* é a capacidade de avaliar a qualidade dos dados no próprio *WareHouse Builder*, em geral conhecida como geração de perfis de dados. Os resultados gerais da geração de perfis de dados são mostrados na figura 3.4.

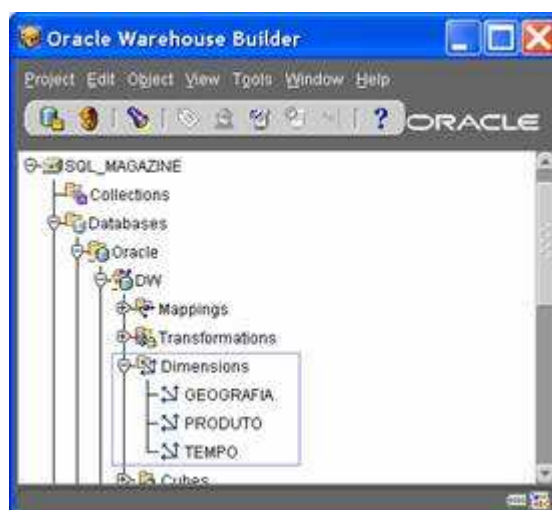


Figura 3.4 – Geração de Perfis Dimensionais
Fonte: Oracle (2012)

O atual conjunto de recursos é melhorado utilizando tecnologia central de banco de dados. Os algoritmos de qualidade de dados agora podem ser executados em paralelo para alcançar uma melhor taxa de transferência. Os algoritmos são melhorados e aprimorados constantemente para garantir a produção de resultados melhores e mais rápidos. Tudo isso é fornecido com o *WareHouse Builder* e está disponível nos componentes centrais do ETL.

O *Warehouse Builder* foi configurado numa base de teste, onde se conseguiu de forma plena atender a suas necessidades como o processo de ETL a ele dado. Existem além de recursos ETL, opção para montar cubo de dados na própria ferramenta, mas esse recurso não foi proposto nessa solução por possuir um nível de complexibilidade na sua solução do qual se tornou inviável sua utilização.

3.2 OLAP

A ferramenta *Oracle BI Administration Tool* se trata de uma ferramenta agregada a plataforma *Oracle*, da qual se faz bastante ênfase no processo de consultas de informações OLAP. Junto a ela, o *Oracle BI Publisher* disponibiliza relatórios dinâmicos para sua utilização através de qualquer navegador *Web* existente.

3.2.1 Oracle BI Administration Tool

A ferramenta *Oracle BI Administration Tool* é uma plataforma de BI completa, com grande flexibilidade e que funciona em diversas plataformas sobre sistemas operacionais. A solução oferece ferramentas de relatórios, análises, medidores de indicadores (*dashboards*), integração de dados e mineração de dados (*Data Mining*), como mostra a figura 3.5 que ilustra as ferramentas disponíveis.



Figura 3.5 – Ciclo do BI
Fonte: Oracle (2012)

Com o uso do *Oracle BI Administration Tool* o usuário pode analisar uma grande quantidade de dados no banco de dados. Sua vantagem é possuir uma interface amigável para o desenvolvimento dos cubos de dados do qual serão utilizados posteriormente na solução proposta pela ferramenta.

O *Oracle BI Administration Tool* possui estruturas para conectar as informações para o OLAP de diversas formas como: banco de dados, arquivos texto, Excel, etc. Sua estrutura de conexão depende da estruturação da ferramenta OLAP junto ao ODBC da base do DM. A figura 3.6 mostra a configuração do ODBC junto ao OLAP.

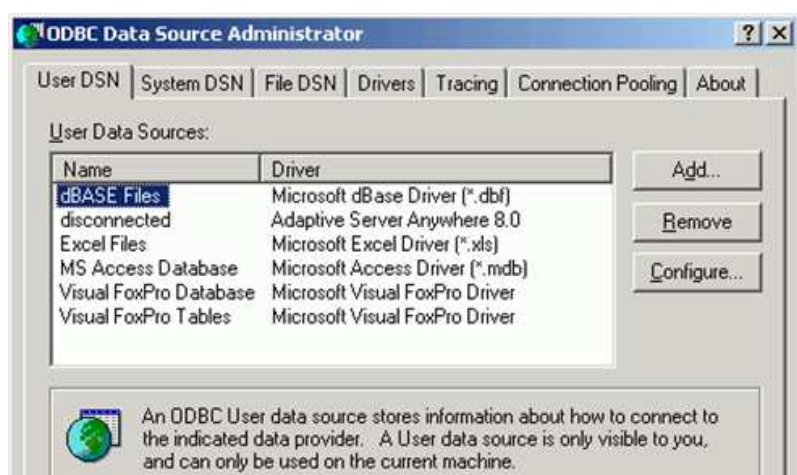


Figura 3.6 – ODBC para OLAP
Fonte: Oracle (2012)

Quando o OLAP é desenvolvido, suas informações podem ser montadas em três camadas de informação: Física, Modelo de Negócio e Apresentação.

- **Modelo Físico:** é onde as informações do *Data Source* estão fisicamente armazenadas;
- **Modelo de Negócio:** é onde os termos de negócio são desenvolvidos após a camada física ser carregada. Ex: chaves primárias, chaves estrangeiras, regras de negócio entre tabelas, etc;
- **Modelo de Apresentação:** é onde o modelo de negócio já finalizado, o usuário irá enxergar exatamente a informação como disposta no Modelo de Apresentação.

A figura 3.7 ilustra a tela principal de configuração do *Oracle BI Administration Tool*.

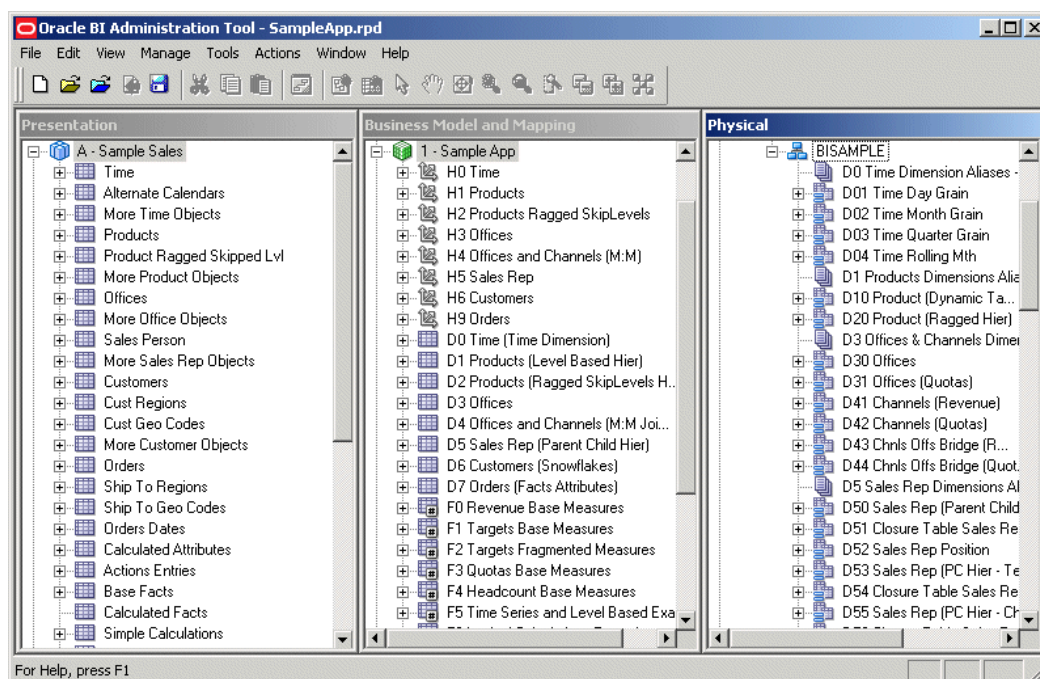


Figura 3.7 – Tela principal do Oracle BI Administration Tool

Fonte: Oracle (2012)

E com o resultado do cubo desenvolvido, é apresentado a informação da seguinte forma conforme figura 3.8.

Calendar Year	Amount Sold	Unit Cost	Gross Profit
1998	23,806,489	17,954,557	5,851,931
1999	22,219,948	17,685,654	4,534,294
2000	23,676,378	19,974,952	3,701,426
2001	28,136,462	22,102,118	6,034,344

Figura 3.8 – OLAP visualizado

Fonte: Oracle (2012)

3.2.2 Oracle BI Publisher

A ferramenta *Oracle BI Publisher* é uma plataforma de BI *Oracle*, própria para construção de relatórios dinâmicos, com grande flexibilidade e que funciona em diversas plataformas *Web*. A solução oferece ferramentas de relatórios dispostas em vários formatos, como mostra a figura 3.9 e 3.10 que ilustra alguns exemplos disponíveis.



Figure 1: Retail Sales by Product Category

2004 Sales

March Sales

Category	Sales (in thousand USD)
Motor vehicle & parts dealers	73,458
Furniture & home furnishings stores	8,655
Electronics & appliance stores	7,762
Building material & garden eq. & supplies dealers	25,385
Food & beverage stores	40,854
Health & personal care stores	16,977
Gasoline stations	25,167
Clothing & clothing accessories stores	15,917
Sporting goods, hobby, book & music stores	6,592
General merchandise stores	41,482
Miscellaneous store retailers	9,781
Nonstore retailers	18,353
Food services & drinking places	31,364

Figura 3.9 – Relatório Web OLAP

Fonte: Oracle (2012)

ORACLE General Ledger
Vision Operations (USA)

Posted Journals
For Jan-99

Report Date: 7/29/2005
Page 13 of 50

Currency: USD
Balance Type: Actual

Journal Total	18,395.83	18,395.83	0.00
Batch Total	18,395.83	18,395.83	0.00

Batch: Manual 10466 20-OCT-99 09:18:01				Posted Date: 20-OCT-99			
Journal: Manual 10165 20-OCT-99 09:18:01				Category: Depreciation Adjustment		Effective Date: 31-JAN-99	
Reference:				Currency: USD			
Line	Account	Transaction Date	Description	Line Item	Debit	Credit	Unit
10	01-520-7320-0000-000	31-JAN-99	Jan-99 Dep Adj for F/S Presentation		147,556.00		0.00
20	01-000-1620-0000-000	31-JAN-99	Jan-99 Dep Adj for F/S Presentation			147,556.00	0.00
30	01-540-7330-0000-000	31-JAN-99	Jan-99 Dep Adj for F/S Presentation		15,000.00		0.00
40	01-000-1630-0000-000	31-JAN-99	Jan-99 Dep Adj for F/S Presentation			15,000.00	0.00

Journal Total	162,556.00	162,556.00	0.00
Batch Total	162,556.00	162,556.00	0.00

Batch: Reverse "test" 21-OCT-99 14:31:37 - 349600				Posted Date: 21-OCT-99			
Journal: Reverse "test" 21-OCT-99 14:31:37				Category: Adjustment		Effective Date: 31-JAN-99	
Reference:				Currency: USD			
Line	Account	Transaction Date	Description	Line Item	Debit	Credit	Unit
10	01-000-1110-0000-000	31-JAN-99				10,000.00	0.00
20	01-000-1110-0000-000	31-JAN-99			10,000.00		0.00

Journal Total	10,000.00	10,000.00	0.00
Batch Total	10,000.00	10,000.00	0.00

Batch: mm#1 01-JUN-98 14:46:09				Posted Date: 01-JUN-98			
Journal: mm#1				Category: Adjustment		Effective Date: 01-JAN-99	
Reference:				Currency: USD			
Line	Account	Transaction Date	Description	Line Item	Debit	Credit	Unit
1	01-430-7110-0000-000	01-JAN-99			100.00		0.00
2	01-430-7120-0000-000	01-JAN-99				100.00	0.00

Private and Confidential

Private and Confidential

Figura 3.10 – Relatório Web OLAP

Fonte: Oracle (2012)

O *Oracle Business Intelligence Publisher* (BI) (anteriormente *XML Publisher*) é uma solução de relatórios corporativos para a criação, gerenciamento e entrega de todos os seus documentos altamente formatados, como relatórios operacionais, documentos eletrônicos de transferência de fundos, formatos para o governo em PDF, etiquetas de transporte, cheques, vendas e cartas de marketing, e muito mais.

Construído em padrões abertos, *Oracle BI Publisher* também permite a equipe e os desenvolvedores criem os modelos de dados com qualquer fonte de dados existente no mercado e com isso construir relatórios personalizados. O *Oracle BI Publisher* pode gerar dezenas de milhares de documentos por hora, com o mínimo de impacto aos sistemas operacionais. Os relatórios podem ser projetados usando produtos desktop familiares e visualizado com algum navegador *Web* onde há vários formatos de entrega para uma grande variedade de destinos.

Alguns benefícios da ferramenta.

- Reduzir a complexidade e custos de implementação mais baixos, eliminando soluções pontuais caras e simplificar a sua arquitetura de informação com um ambiente de relatório único para todos os seus documentos;
- Utiliza dados de qualquer lugar, cria relatórios a partir de praticamente qualquer fonte de dados que fornece uma conexão JDBC, incluindo serviços *Web*, XML, HTTP, fonte de dados de arquivos, e muito mais;
- Usa as ferramentas que você sabe usar em produtos de desktop conhecidos ou ferramentas de desenvolvimento para criar e manter layouts de relatório;
- *Hot-Pluggable-Oracle BI Publisher* integra com sua infra-estrutura de TI, suporte para uma vasta gama de modelos de segurança e integração com aplicativos personalizados através de serviços *Web* ou APIs Java;
- Pré-integrada com o *Oracle*, funciona perfeitamente com o *Oracle Business Intelligence Suite Enterprise Edition Plus*, *E-Business* da *Oracle Suite*, *PeopleSoft Enterprise*, *JD Edwards Enterprise One*, *Hyperion Planning*, *Oracle Application Express (APEX)* e *Oracle Discoverer BI*.

3.3 Considerações finais

Na análise realizada com o *Oracle Warehouse Builder* destacou-se a funcionalidade na qual é possível construir um ETL de uma forma mais abrangente e sucinta com a proposta de BI estabelecida no projeto. Sua interface simplificada ajuda no processo de formulação das regras de carga das informações para o DM.

O *Oracle BI Administration Tool* possui ótimos recursos OLAP para montar cubos de dados, com facilidade para a aplicação em outras ferramentas da plataforma *Oracle*. Com sua interface dividida em camadas é possível estruturar o cubo de dados da forma em que essa informação será disponibilizada para o usuário final, do qual será utilizada da maneira mais interessante e relevante às regras de negócio utilizadas na empresa.

Já o *Oracle BI Publisher* disponibiliza excelentes recursos OLAP para a construção de relatórios dinâmicos para sua utilização pelos gestores da empresa. Sua flexibilidade na interação de sua consulta e a disponibilidade para sua utilização através de qualquer navegador *Web* facilita como um todo a sua forma de obter os resultados necessários para a consulta solicitada.

4 ARQUITETURA PROPOSTA

A proposta deste trabalho é propor uma solução de BI para o ERP SFT concebida com a utilização de ferramentas *Oracle*, objetiva com isto estabelecer uma arquitetura ou conjunto de ferramentas que passem a ser utilizadas na elaboração das futuras soluções de BI a serem criadas para atender as necessidades dos clientes.

Considera-se a premissa que esta forma de conceber soluções de BI possa ser aplicada para atender diversas áreas e ramos de atividade, de fácil utilização e que permita a construção de novos DMs para que seja possível criá-los de acordo com a área de negócio dos clientes. Foi desenvolvida uma solução integrada para atender a todos esses requisitos de forma simples e intuitiva sem a necessidade do usuário possuir conhecimento em várias ferramentas para a construção de uma solução de BI.

Várias ferramentas foram avaliadas com o objetivo de escolher a melhor solução possível para a construção do BI. Entre tantas possíveis ferramentas pesquisadas, foram optadas as que realmente possuem uma melhor flexibilidade na sua implementação, uma interface de simples utilização pelo usuário final e métricas da própria *Oracle* sobre *performance* e capacidade de utilização com grandes massas de dados.

Conforme pode ser visto na figura 4.1 a solução proposta contempla a construção do DM, o processo de ETL e a criação dos cubos de dados para futuras consultas e análises.

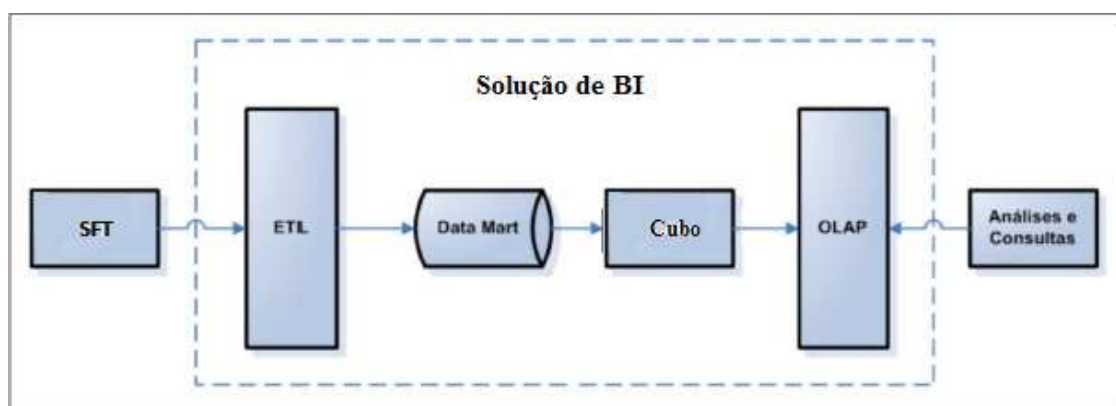


Figura 4.1– Visão geral da solução de BI construída
Fonte: Do próprio autor

Segundo Prodanov (2009), ao se analisar um fato, o conhecimento científico não apenas trata de explicá-lo, mas também busca descobrir e explicar suas relações com outros fatos, conhecendo a realidade além de suas aparências. O conhecimento científico é considerado como:

- Acumulativo, por oferecer um processo de acumulação seletiva, em que novos conhecimentos substituem outros antigos, ou somam-se aos anteriores;
- Útil para a melhoria da condição da vida humana;
- Analítico, pois procura compreender uma situação ou um fenômeno global por meio de seus componentes;
- Comunicável, já que a comunicabilidade é um meio de promover o reconhecimento de um trabalho como científico. A divulgação do conhecimento é responsável pelo progresso da ciência;
- Preditivo, pois a partir de investigação dos fatos e do acúmulo de experiências, o conhecimento científico pode dizer o que foi passado e prever o que será futuro.

4.1 Definição das ferramentas

O uso das ferramentas se deve pelo foco desenvolvido para este trabalho propor o uso de recursos disponíveis pela *Oracle*, do qual possui vários softwares destinados a diversas áreas existentes, onde há muitos recursos existentes para diversas soluções de BI.

Os principais motivos para a escolha dessas ferramentas foram: (i) a quantidade e variedade dos recursos disponíveis nestas ferramentas, (ii) a possibilidade de serem utilizadas pela plataforma existente pelo ERP da Safetech, (iii) pela facilidade de integração com a solução proposta, visto que o funcionamento destas ferramentas possui amplitude para diversos tipos de dados a serem processados e seus resultados podem ser vistos de forma WEB por um navegador próprio da *Oracle*, criando uma solução integrada e de fácil uso por parte dos usuários finais da solução de BI do Gestão SFT.

Foram efetuadas várias pesquisas para encontrar os melhores recursos da *Oracle* para a construção do BI para o ERP Safetech. Procurou-se utilizar recursos de simples manuseio e fácil implementação para conseguir o resultado esperado.

4.1.1 ETL

Para a construção do processo de ETL, foi efetuadas várias pesquisa para encontrar ferramentas *Oracle* compatíveis com essa necessidade. Fóruns e tutoriais da *Oracle* sugerem que a ferramenta *Oracle Warehouse Builder* seja utilizada. Foram efetuadas várias pesquisas

para encontrar outras possibilidades de ferramentas de ETL, mas a *Oracle* somente sugere que essa seja a opção escolhida para essa necessidade.

O armazenamento dos dados na forma de um DM será realizado na versão mais atual do banco de dados que é *Oracle 11g Standard Edition Release 11.2.0.1.0*.

4.1.2 Cubo de Dados

Para a construção do cubo de dados, são disponibilizadas várias ferramentas pela *Oracle*. Tanto o *Oracle Warehouse Builder* como o *Oracle BI Administration Tool* possuem opções para desenvolver cubos de dados para as consultas OLAP. Pelo grau de dificuldade na interação e desenvolvimento da etapa pelo *Oracle Warehouse Builder*, optou-se em utilizar o *Oracle BI Administration Tool* por ser uma ferramenta muito flexível, compacta, de simples interação e que possui um *help* com muito mais informações quando se trata no desenvolvimento do cubo.

A ferramenta *Oracle Warehouse Builder* possui várias possibilidades quando se fala de métodos para o desenvolvimento do ETL, tanto que existem várias soluções para a construção do cubo de dados. A figura 4.2 ilustra o método de criação de cubo de dados no *Oracle Warehouse Builder*.

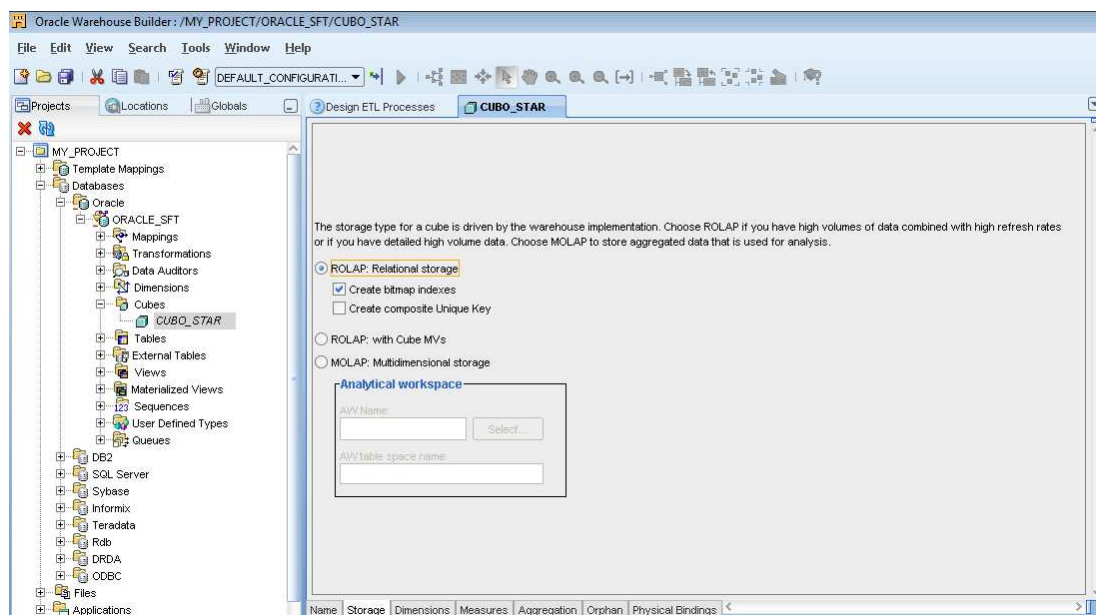


Figura 4.2– Cubo de Dados no Oracle Warehouse Builder

Fonte: Do próprio autor

Conforme dito, essa opção para criação do cubo de dados existe, mas pela grande quantidade de problemas na sua utilização, essa opção foi descartada. Houve vários problemas na criação do cubo de dados de forma simplificada, também houve a situação que ao conseguir criar um cubo de dados para amostragem, o *Export* do cubo criado somente gerou erros em tela no processo para carrega-lo, chegando ao ponto da própria ferramenta encerrar-se automaticamente deixando de salvar tudo o que havia sido feito, casos onde o *help* da ferramenta não conseguiu auxiliar nas dúvidas ocorridas, problemas de lentidão e não conseguir-se em nenhum caso montar o cubo de dados ao ponto de executa-lo.

A figura 4.3 mostra a forma do cubo de dados criado, aguardando para ser feito o *Export* da consulta para assim ser carregado para uma ferramenta OLAP.

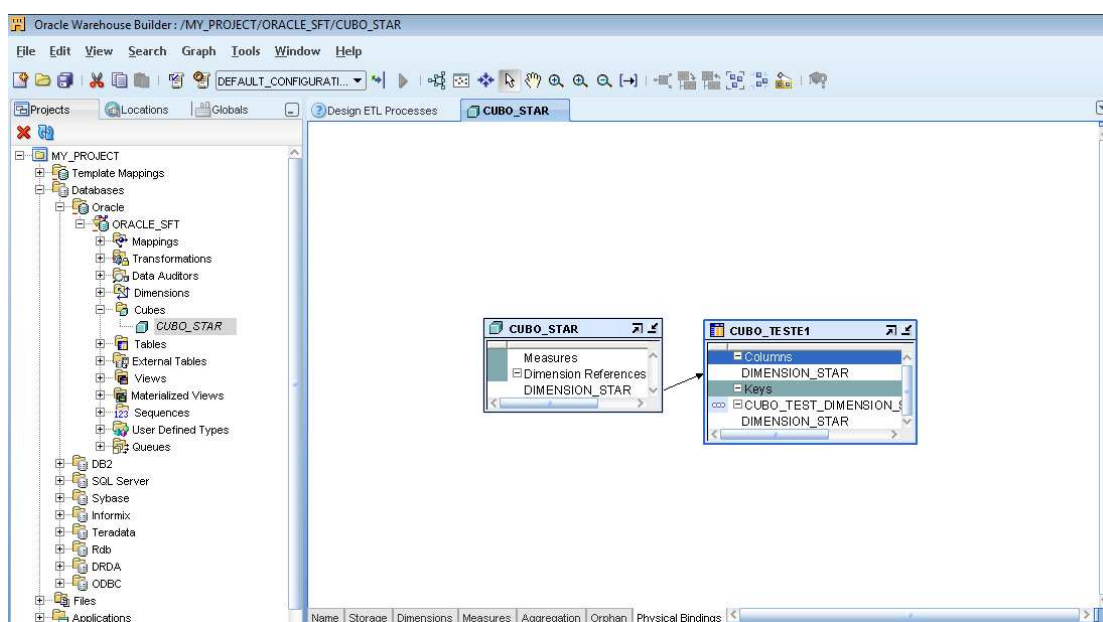


Figura 4.3– Cubo de Dados pronto para o Export

Fonte: Do próprio autor

Após vários testes serem feitos e nenhum com êxito, resolveu-se escolher o *Oracle BI Administration Tool* como a ferramenta *Oracle* para montar os cubos de dados para a solução BI.

4.1.3 OLAP

Quando se trata da utilização de uma ferramenta OLAP para as consultas e relatórios, a *Oracle* sugere que seja utilizado o *Oracle BI Publisher* por se tratar de uma simples interface para a interação do usuário, sua modelagem para utilização em ambiente Web e por

possuir vários recursos, desde transformar o relatório em PDF, Excel, Word, HTML, entre tantos outros formatos. Foram efetuadas várias pesquisas, e em todas elas a *Oracle* é clara que se deve utilizar essa ferramenta para desenvolvimento OLAP. Para ser possível ter outra forma de gerar o relatório OLAP sem o uso dessa ferramenta, foi desenvolvido um módulo onde é possível gerar arquivos Excel com formato OLAP para o usuário, essas informações são disponíveis em *Pivot Tables*, dos quais há recursos de interação diretamente com o usuário.

Para a criação do módulo OLAP optou-se em utilizar duas formas diferenciadas de geração da consulta OLAP por questões de acessos e privilégios de rede. Existem muitos casos onde empresas não disponibilizam o uso de seus funcionários da internet, e com isso são criadas várias políticas de segurança nas redes empresariais para com que isso ocorra corretamente. Para não obterem-se problemas com esses tipos de seguranças internas, resolveu-se criar além da opção da consulta OLAP através de uma ferramenta *Web*, um recurso do qual possa ser utilizado por um modulo criado no sistema SFT que, além de servir para parametrização das configurações, possa ser usado para emissão desses relatórios.

Com isso, a opção do módulo OLAP criado poderá abranger vários tipos de usuários diferentes, não só gestores, mas sim todo e qualquer usuário que necessita de uma consulta de informações mais filtrada para facilitar seu operacional diário.

4.2 Visão geral

O modelo de proposta para uma solução de BI foi dividido em algumas etapas conforme a necessidade de cada uma delas perante a arquitetura construída.

A figura 4.4 demonstra o fluxo da solução proposta como um todo, onde está localizado cada processo e os dados do sistema. Como base em toda a solução está o sistema SFT onde o usuário final estará efetuando lançamentos de dados nos diversos módulos lá existentes, de onde serão gravados num banco de dados.

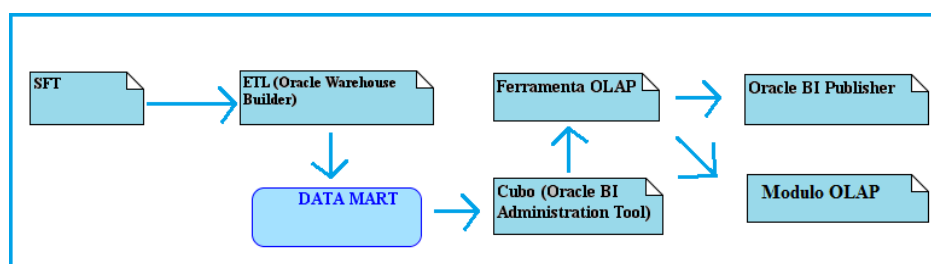


Figura 4.4– Fluxo da solução de BI

Fonte: Do próprio autor

Como o sistema de Gestão SFT é constituído da arquitetura de dados da *Oracle*, é conveniente que os processos de ETL e OLAP sejam constituídos também por ferramentas *Oracle* para facilitar o consultor no desenvolvimento do processo de BI.

O fluxo de dados demonstra toda a estruturação da solução, onde o ERP da Safetech é utilizado para extração da sua base de informações para o ETL (*Oracle Warehouse Builder*), de onde é efetuado manipulações dos dados, limpeza e extração do que é existente para o DM. Após isso, uma ferramenta OLAP (*Oracle BI Administration Tool*) irá montar o cubo de dados das consultas para que posteriormente, possa ser utilizado por alguma das duas ferramentas de OLAP dispostas na estrutura (*Oracle BI Publisher* ou Excel) com que o processo de BI esteja estruturado por completo.

A proposta de interface OLAP desenvolvida é fornecida através do ERP SFT, com o intuito de simples utilização para o usuário final a fim de optar-se por quais cubos de dados deseja utilizar e através do módulo OLAP disponibilizado, efetuar as consultas da forma que achar mais conveniente para sua situação.

4.3 Criação do *Data Mart*

Uma das necessidades para a construção da proposta é o DM, sem ele não há como armazenar as informações. O estudo de caso utilizado foi um DM do módulo de Faturamento que irá implicar em informações vindas de outros módulos do sistema para que possa atender a todas as necessidades existentes do usuário. Conforme o DM vai se expandindo com várias informações, é possível reestruturar sua base e adaptar o DM para virar um DW com vários módulos de vários sistemas existentes.

Para cada DM existente, é possível definir suas tabelas fato e dimensão, sendo possível adicionar novas tabelas fato e dimensão ou alterá-las conforme a necessidade proposta. Sua flexibilidade já está disposta na montagem da configuração do banco de dados selecionado para a proposta, banco *Oracle 11g Stantard Edition Release 11.2.0.1.0*. O banco de dados disposto permite efetuar várias cargas de dados e construção de vários cubos de dados, que depois serão disponibilizados para o usuário final efetuar as consultas necessárias.

Serão construídos dois modelos dimensionais para validar o desempenho em sua execução, onde um utiliza o modelo *Star* e o outro utiliza o modelo *SnowFlake*.

4.4 ETL

Com o DM totalmente parametrizado, é possível efetuar as configurações do ETL para montar a transformação, limpeza e carga das informações ao DM. Há necessidade de efetuar as configurações do cubo de dados e por último, desenvolver as configurações OLAP.

A ferramenta *Oracle Warehouse Builder* necessita de várias parametrizações para ser possível efetuar a conexão entre a base de dados que será extraída e o DM no qual estão as informações transformadas para melhor se adequar a necessidade do BI. A figura 4.5 mostra a tela de configuração do *Oracle Warehouse Builder*.

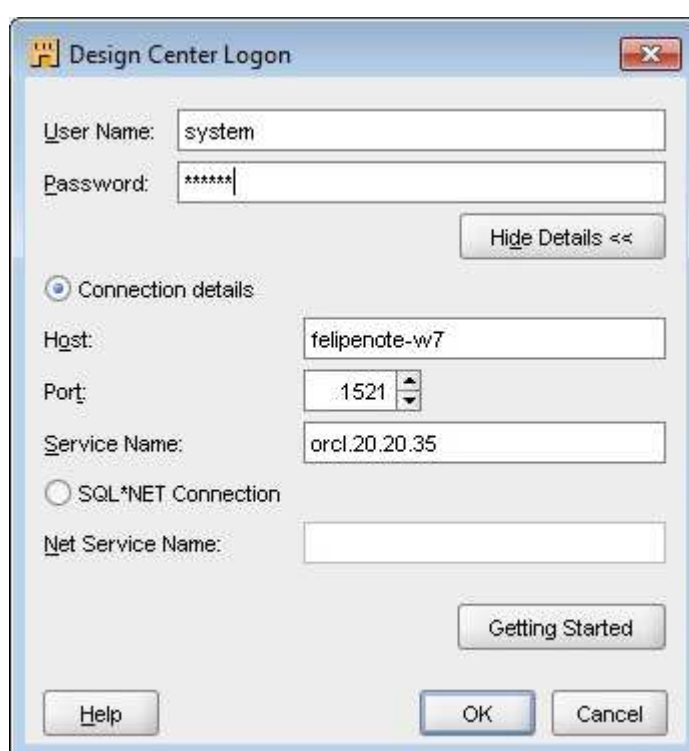


Figura 4.5– Configuração do *Oracle Warehouse Builder*

Fonte: Do próprio autor

Conforme as configurações são efetuadas, é possível parametrizar a ferramenta para qual tipo de ETL deseja que seja efetuada a carga. É possível que seja montada uma carga entre bancos de dados distintos ou efetuar cargas mais refinadas como carregar informações diretamente de outros ERPs já existentes ou até mesmo carregar informações de outras redes distintas da qual está conectada esta estação. A figura 4.6 ilustra as configurações do ETL sobre qual forma será feita a busca dessas informações.

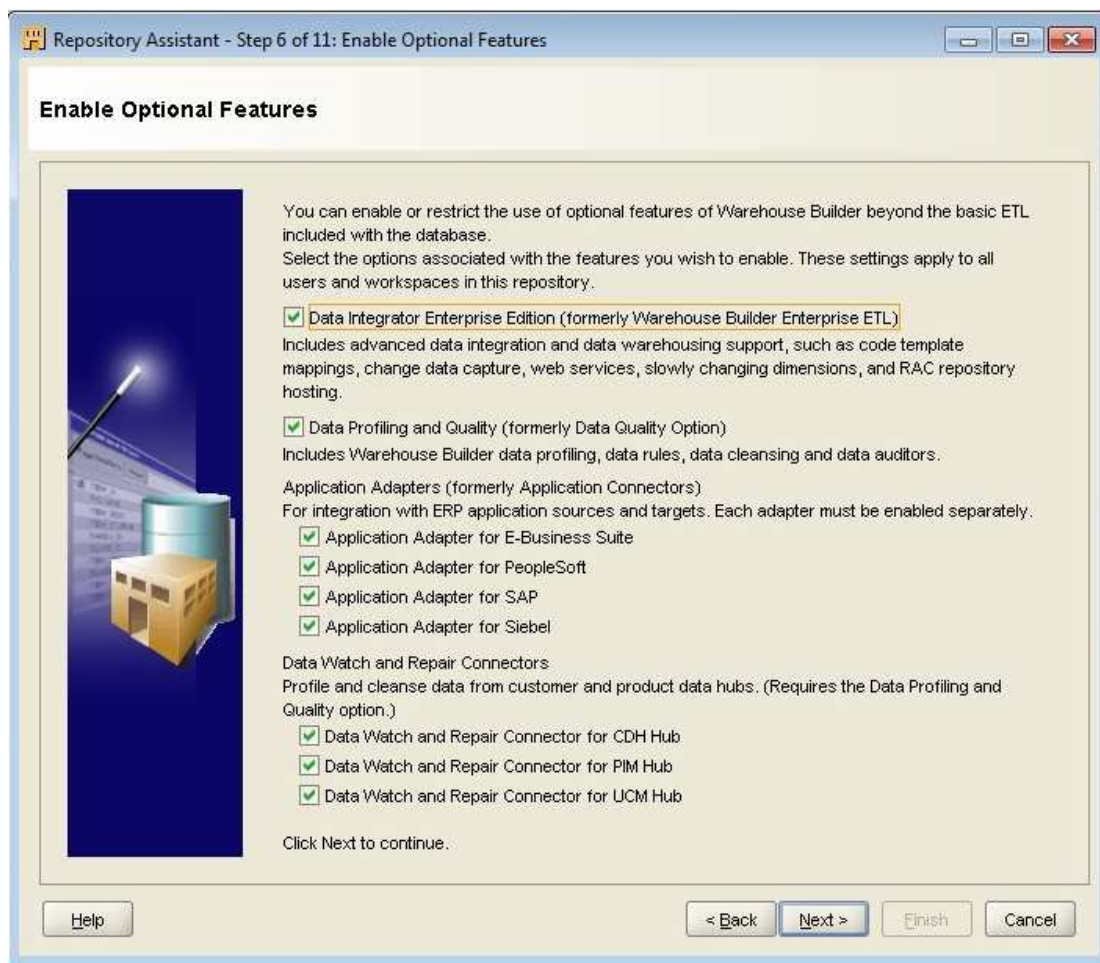


Figura 4.6– Configuração do *Oracle Warehouse Builder* - Parâmetros
Fonte: Do próprio autor

Com a parametrização concluída, é possível efetuar o filtro das informações como: limpar lixos de dados existentes, desenvolver funções para maximizar o resultado da consulta desejada, desenvolver agregações de informações como somas ou distinções de informações repetidas, montar mapeamentos de unidades lógicas, montar as tabelas fato e dimensão para serem utilizadas pelo cubo, montar o próprio cubo de dados, entre outras tantas opções existentes nessa ferramenta. Em sequência, será disposto na tela de configurações das informações do modelo ETL – *Oracle Warehouse Builder*. A figura 4.7 mostra a tela de desenvolvimento do *Oracle Warehouse Builder*.

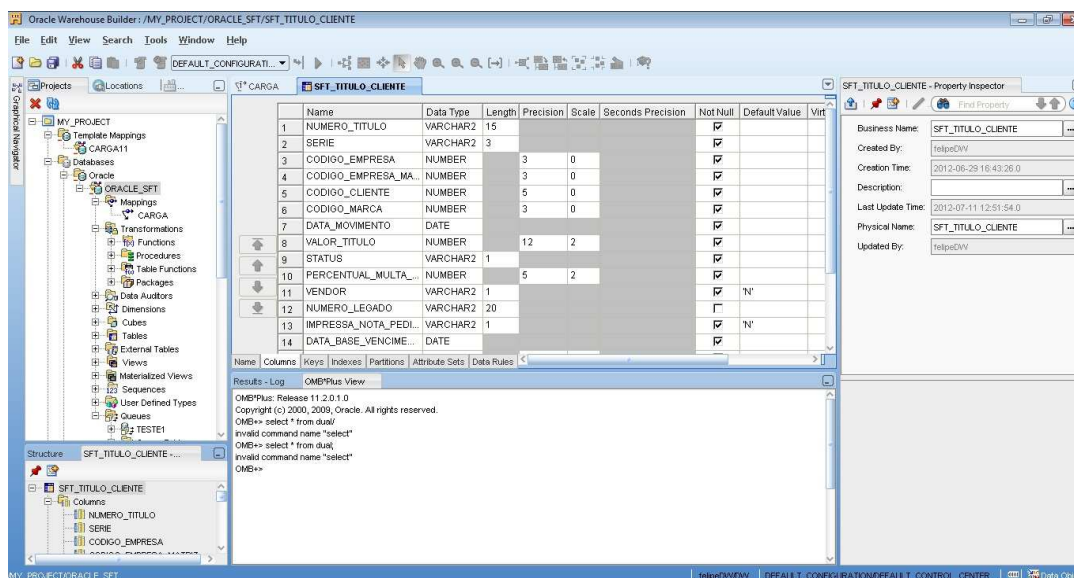


Figura 4.7– Tela de desenvolvimento do *Oracle Warehouse Builder*

Fonte: Do próprio autor

A partir dessa tela, é possível monitorar todo o andamento e programação do processo de ETL. Nela é feita a verificação de como está a sincronização da exportação das informações para o DM e de quanto em quanto tempo é efetuada essa sincronização, as cargas podem ser programadas de hora a hora, ou até mesmo dia a dia.

Se houver necessidades, podem ser feita várias cargas de dados manuais através do programa, se houver a necessidade. Há vários recursos disponíveis como software de integração entre vários bancos de dados, migrações entre ERPs, sincronização de dados através de várias fontes, entre tantos outros recursos dos quais não foram vistos nesse trabalho por não terem sido focados.

4.5 Cubo de dados

O processo de criação dos cubos de dados é necessário para viabilizar a visualização dos dados contidos no DM sobre várias perspectivas. Esse processo é importante, pois com ele são criadas as visões de dados que o usuário, o tomador da decisão, vai utilizar para realizar as análises com as informações que foram fornecidas para cada situação.

Desta forma, foi utilizado um aplicativo da *Oracle* para montagem do cubos de dados que é o *Oracle BI Administration Tool*, esse software interage perfeitamente com a estrutura de desenvolvimento do cubo de dados e a interface do qual seja utilizada para a consulta do OLAP.

A ferramenta *Oracle BI Administration Tool* necessita de algumas configurações desde o meio físico onde se encontram as informações, o mapeamento da unidade e a estruturação do cubo propriamente dito.

Para a estruturação do cubo, muitos dados serão compartilhados entre o ETL e a ferramenta OLAP porque a parte da transformação de muitos dados foi disposta pela ferramenta ETL e essas informações serão utilizadas pela ferramenta OLAP que montará esses cubos. A sincronia entre ferramentas *Oracle* ocorre sem muitas dificuldades, já que sua preocupação é em ter o máximo de interação entre toda sua plataforma. A figura 4.8 é relacionada à tela de importação dos metadados do *Oracle BI Administration Tool*.

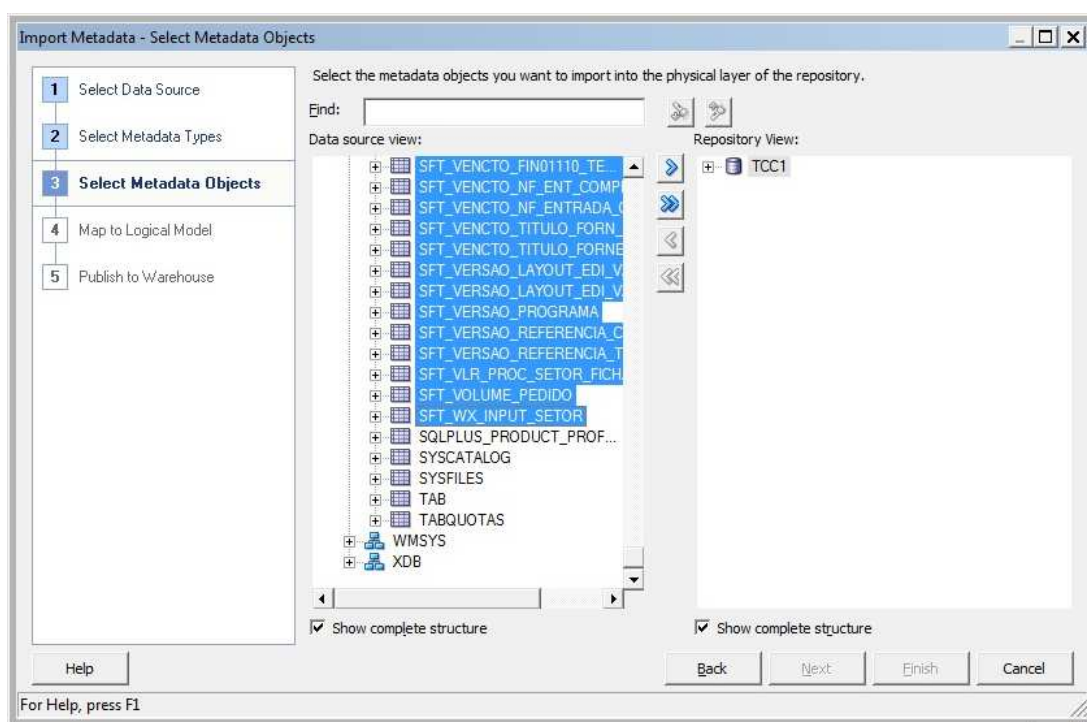


Figura 4.8– Tela de desenvolvimento do *Oracle BI Administration Tool*

Fonte: Do próprio autor

4.6 OLAP

Para efetuar a geração das consultas OLAP, será disponibilizado uma solução pronta que a *Oracle* sugere para as implementações BI de sua arquitetura, junto a isso também foi desenvolvido um módulo onde é possível emitir planilhas pelo uso de um dos módulos do ERP Safetech.

4.6.1 Oracle BI Publisher

A ferramenta *Oracle BI Publisher* possui algumas configurações relacionadas aos usuários que irão possuir acesso para gerar relatórios pelo navegador *web*. Suas opções de relatórios são imensas, podendo ser filtradas da forma que o usuário desejar e criadas no modelo que for mais conveniente para o usuário. A figura 4.9 ilustra uma visão de geração do *Oracle BI Publisher*.

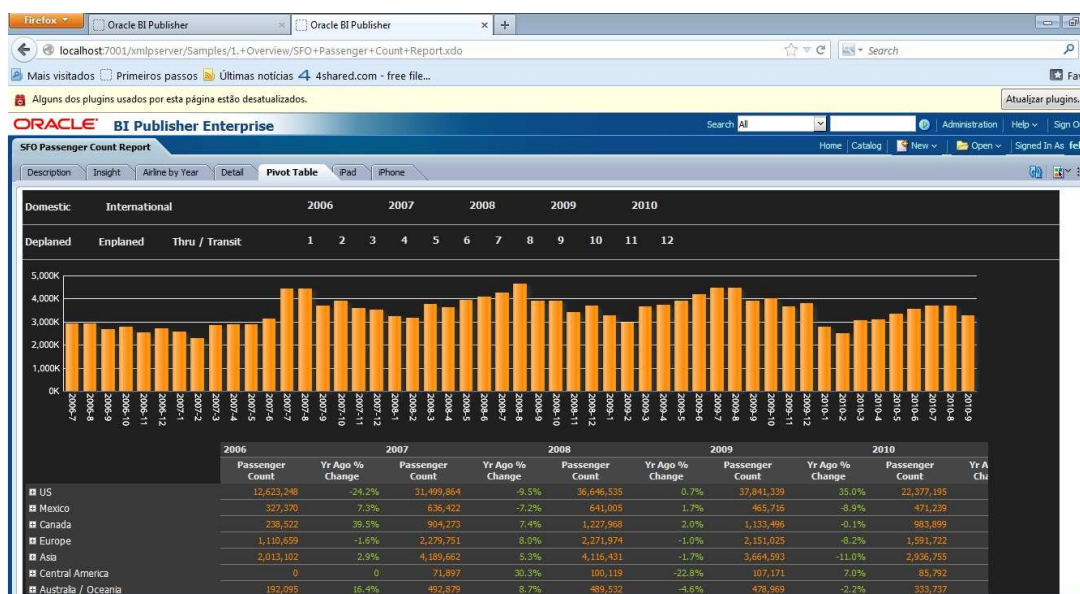


Figura 4.9– Geração do relatório *Oracle BI Publisher* - Geral

Fonte: Do próprio autor

É possível emitir relacionamentos sobre as consultas para as mais diversas necessidades, ocasiões, solicitações e abrangências de informações, das quais não são necessários possuir grandes conhecimentos na área vista, para entender do que se trata. Existe também opções de módulos para a geração do relatório OLAP para dispositivos móveis como IPads e IPHones. A figura 4.10 ilustra uma das formas flexíveis de usufruir da informação existente.

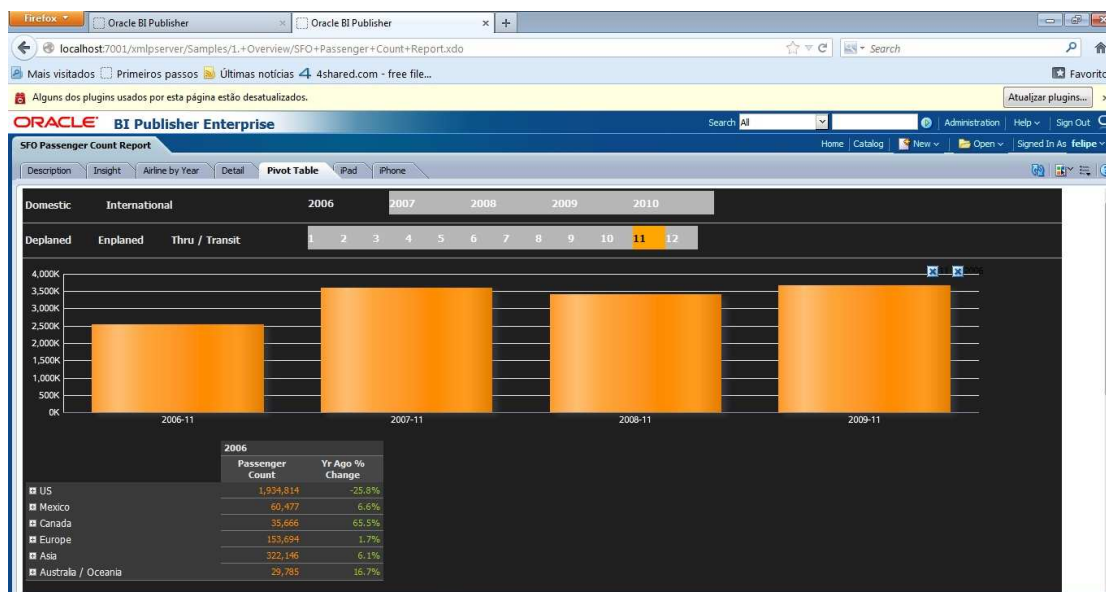


Figura 4.10– Geração do relatório *Oracle BI Publisher* - Filtro

Fonte: Do próprio autor

Os exemplos dispostos acima são exemplos sobre a base de dados padrão de teste da *Oracle*, base SCOTT/TIGER.

4.6.2 Excel

Foi efetuado um estudo para a utilização da ferramenta Excel para a geração das consultas OLAP perante os usuários. Optou-se em utilizá-lo pelo alto grau de entendimento que os usuários chave do sistema possuem em utilizar este recurso com mais facilidade por se tratar de um recurso rotineiro em Excel do que com uma ferramenta *web* do qual nunca tiveram o contato direto.

O Excel possui classes para conectar o banco de dados e montar cubos de dados perfeitamente como se fosse uma ferramenta OLAP, mas o módulo *Pivot Table* atribuído no Excel se trata de conexões diretas ao banco de dados SQL Server, dificultando a geração. Mas com APIs específicas, é possível montar o cubo de dados e consultar suas informações perfeitamente perante a estrutura. As figuras 4.11, 4.12 e 4.13 ilustram a parametrização necessária para configurar a geração do arquivo Excel para o módulo OLAP.

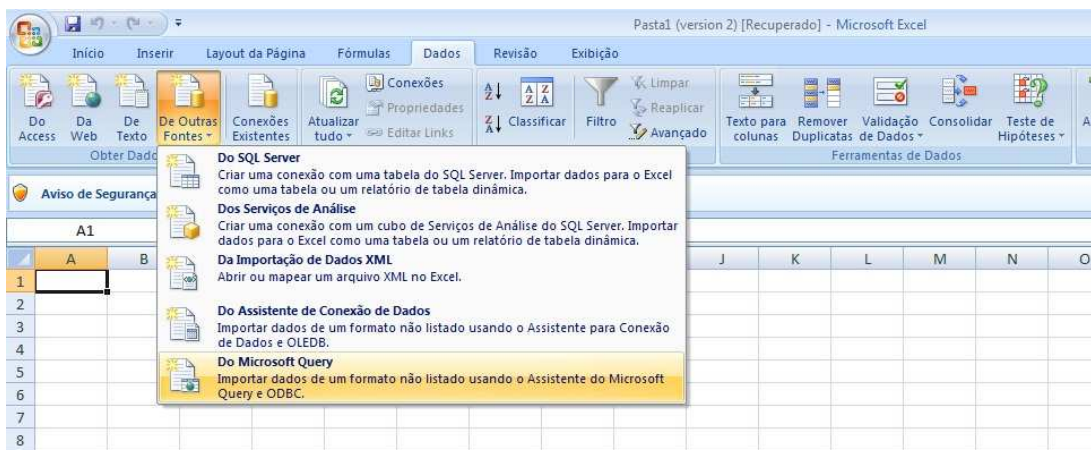


Figura 4.11– Geração do relatório OLAP pelo Excel

Fonte: Do próprio autor

A figura 4.12 mostra a ferramenta *Microsoft Query* da arquitetura Office Excel, onde é possível carregar o cubo de dados e utilizá-lo no Excel como ferramenta OLAP.

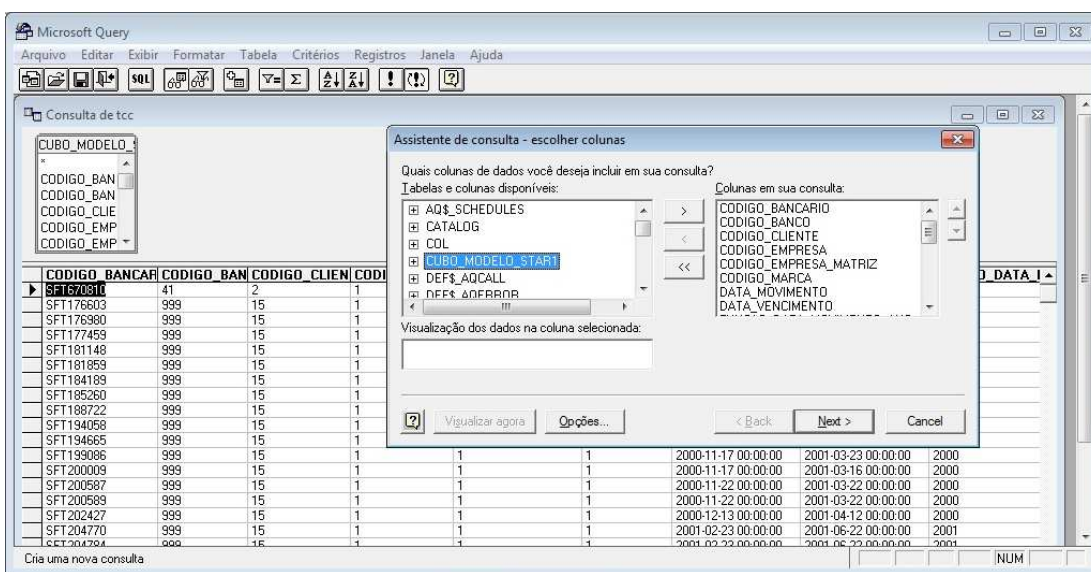


Figura 4.12– Carregando Cubo de dados

Fonte: Do próprio autor

Já a figura 4.13 mostra a carga do modelo *Star* sendo feita para a estrutura OLAP existente no Excel.

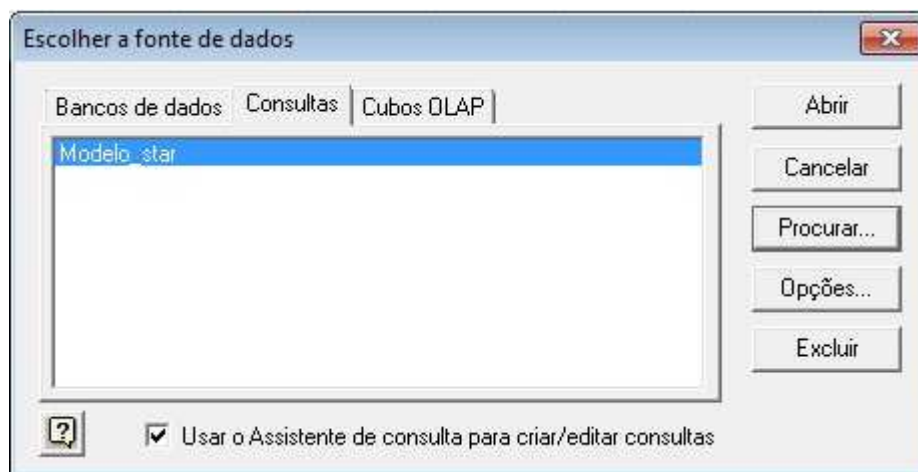


Figura 4.13– Escolhendo opção de Dados

Fonte: Do próprio autor

A opção de geração Excel, será disponibilizada num módulo desenvolvido junto aos módulos do ERP SFT, que terá opções para parametrizar e configurar as consultas OLAP que o usuário irá utilizar.

4.6.3 Modelagem da Arquitetura proposta

O módulo desenvolvido para o ERP da Safetech constitui das opções OLAP que os usuários poderão efetuar a emissão dos relatórios parametrizados para seu nível de sistema, conforme a utilidade do qual necessitam. A figura 4.14 mostra a tela inicial para geração do arquivo Excel sobre a consulta OLAP.

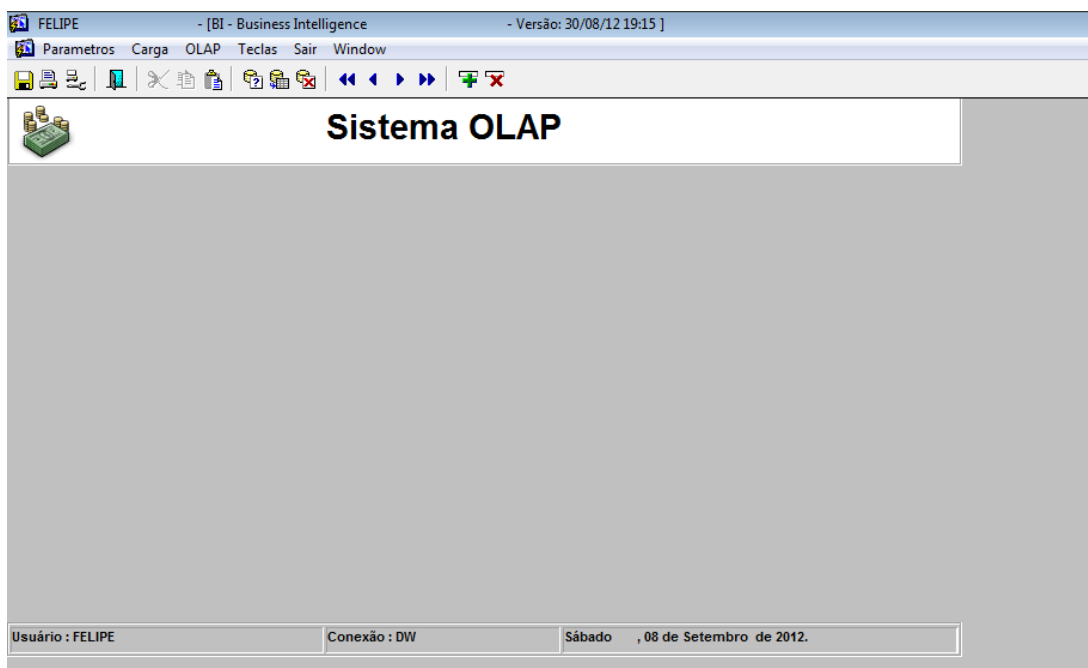


Figura 4.14– Tela inicial da estrutura desenvolvida

Fonte: Do próprio autor

O módulo foi dividido em três itens: Parâmetros, Carga e OLAP. Cada módulo possui informações cadastrais perante usuários para acesso, carga do cubo de dados e as consultas OLAP.

Os parâmetros são responsáveis pelas informações cadastrais de cada usuário, esse item somente será disponibilizado para o gestor de cada área de negócio da empresa para efetuar o cadastro das empresas que o usuário terá acesso, cadastro de usuários para acessar o módulo e alteração de senha no sistema.

A estrutura de carga terá a opção de carregar o cubo de dados que foi montado pelo *Oracle BI Administration Tool* para poder ser visualizado suas informações num módulo OLAP gerado pelo Excel. Também será possível consultar quais cubos de dados estão carregados no sistema e quais não, esse nível de acesso será disponibilizado para usuários chave do processo BI.

Já o item OLAP é onde está disponibilizada a geração das informações para o Excel do usuário, de onde todo o usuário que possui acesso ao módulo OLAP terá acesso a esse item. Será disponibilizado no programa o local onde o usuário deseja que o arquivo seja gerado, e após sua exportação, ele poderá abrir o arquivo e efetuar o filtro de consulta das informações mais convenientes para a situação necessária no momento.

Na seção 5.5.2 será apresentado à ferramenta desenvolvida para geração das consultas OLAP sobre a estrutura Excel. Nessa ferramenta serão apresentadas suas telas de parametrização, o módulo de carga do cubo de dados e a estrutura desenvolvida para exportar o cubo para a estrutura OLAP Excel.

5 ESTUDO DE CASO

Segundo Simon (2012), o estudo de caso é uma técnica de estudo, em que é realizada uma pesquisa sobre um caso particular, para tirar conclusões sobre princípios gerais daquele caso específico.

Conforme Yin (2001 apud FELBER, 2006, p. 57), um estudo de caso é um questionamento empírico que investiga um fenômeno contemporâneo com seus contextos da vida real, quando as fronteiras entre fenômeno e contexto não são claramente evidentes, e nos quais fontes múltiplas de evidências são usadas.

Com objetivo de validar a arquitetura proposta com um caso prático, foi desenvolvido um estudo de caso com uma base de dados real. Alguns dados foram modificados para preservar seu conteúdo perante a empresa.

Neste capítulo será organizado de acordo com as etapas de todo o processo de criação de uma solução de BI utilizando a arquitetura proposta nesse trabalho. Inicia-se construindo o DM, criando as tabelas dimensões com seus respectivos campos e a tabela fato com seus respectivos campos. Posteriormente será demonstrar o processo ETL do ERP para o DM utilizado a ferramenta *Oracle Warehouse Builder*. Na sequência a criação de dois cubos de dados com modelos dimensionais diferentes para efetuar um comparativo entre as formas. Por fim, a análise de alguns dados com a ferramenta *Oracle BI Publisher* ou através de um programa desenvolvido para exportar o cubo de dados diretamente para um arquivo Excel.

5.1 Construção do DM

Para a construção da proposta de uma solução de BI, foram modelados dois cubos de dados e como consequência disso, foram construídos dois DMs distintos. Para o estudo de caso, será abordado o DM do qual possui a modelagem *Star* como referência.

Para o estudo de caso será criada uma tabela fato chamada “SFT_FATO_TITULO_CLIENTE” e cinco tabelas dimensões, sendo uma para os clientes “SFT_AGRUP_CLIENTE”, outra para movimentações bancárias “SFT_AGRUP_BANCO”, outra para as movimentações dos vencimentos “SFT_MOVTO_VENC_TITULO_CLIENTE”, outra para localizar o cliente “SFT_LOCALIZA” e uma função para calcular variações de tempo nas datas processadas “FUNCAO_DATA”. Os itens na sequência abordam estas tabelas e seus campos correspondentes.

A figura 5.1 ilustra o ER utilizado para o estudo de caso da construção do DM.

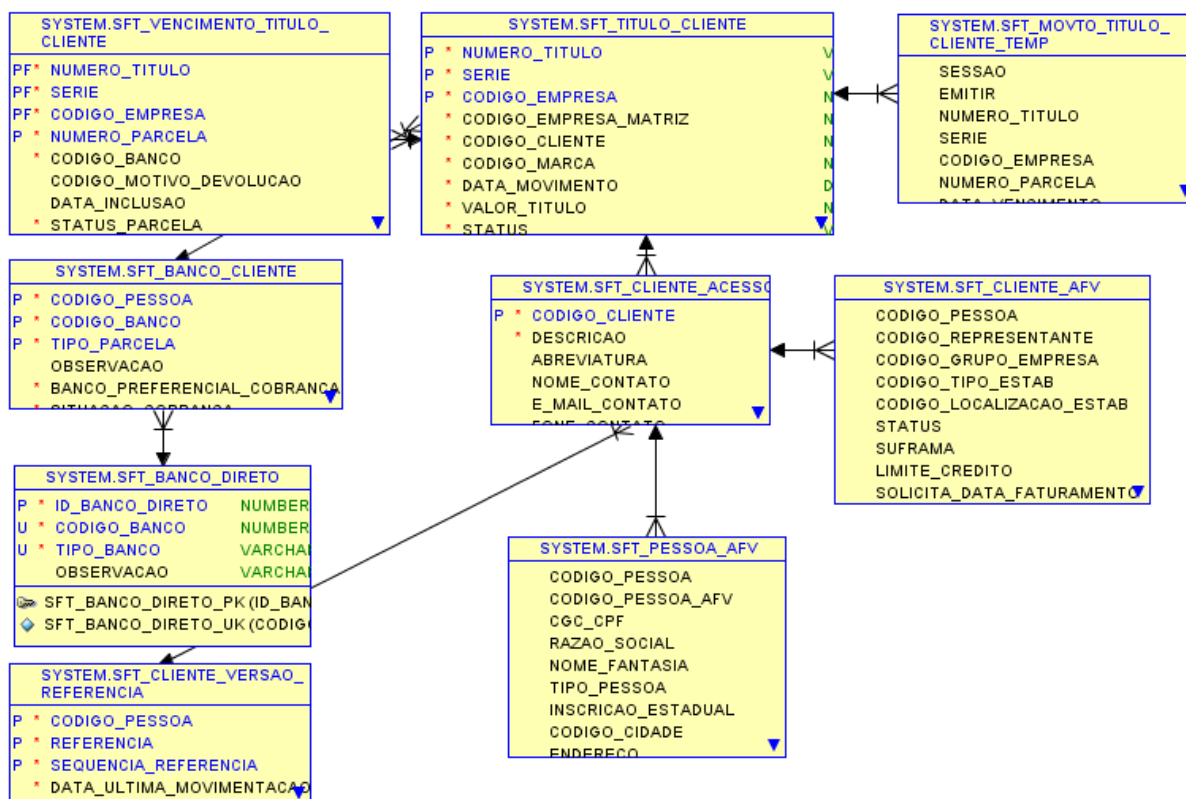


Figura 5.1– ER do modelo usado para estudo de caso

Fonte: Do próprio autor

5.1.1 Tabela fato SFT_FATO_TITULO_CLIENTE

A tabela fato SFT_FATO_TITULO_CLIENTE contém informações sobre a cobrança do faturamento emitido pela empresa e é composta pelas chaves estrangeiras para as dimensões descritas na próxima sessão e pelas suas medidas. O quadro 5.1 apresenta uma explicação sobre os atributos que compõem esta tabela fato:

Nome da Coluna	Descrição
CODIGO_CLIENTE	Código cliente sobre o qual o título foi efetuado
CODIGO_BANCO	Código do banco vinculado à parcela
DATA_MOVIMENTO	Data da ultima movimentação que o título sofreu
NUMERO_TITULOS	Número do título perante a nota fiscal
SERIE	Série perante a nota fiscal
CODIGO_EMPRESA	Código da empresa perante a nota fiscal

NUMERO_PARCELA	Número da parcela dos títulos perante a nota fiscal
DATA_VENCIMENTO	Data de vencimento da parcela do título
VALOR_PARCELA	Valor da parcela do título
VALOR_ABATIMENTO	Valor do abatimento que a parcela sofreu sobre o título
VALOR_DESCONTO	Valor do desconto que a parcela sofreu sobre o título
VALOR_PAGO	Valor pago da parcela perante o título

Quadro 5.1 – Descrição dos campos da tabela SFT_FATO_TITULO_CLIENTE

Fonte: Do próprio autor

5.1.2 Tabela dimensão SFT_AGRUP_BANCO

A tabela dimensão SFT_AGRUP_BANCO contém informações sobre os bancos que cada cliente utiliza para efetuar suas movimentações perante a empresa. O quadro 5.2 apresenta uma explicação sobre os atributos que compõem esta tabela dimensão:

Nome da Coluna	Descrição
CODIGO_BANCO	Código do banco do qual a empresa trabalha
NOME_BANCO	Nome do banco
TIPO_BANCO	Tipo de banco existente para a movimentação

Quadro 5.2 – Descrição dos campos da tabela SFT_AGRUP_BANCO

Fonte: Do próprio autor

5.1.3 Tabela dimensão SFT_AGRUP_CLIENTE

A tabela dimensão SFT_AGRUP_CLIENTE contém informações sobre cada cliente existente perante a empresa. O quadro 5.3 apresenta uma explicação sobre os atributos que compõem esta tabela dimensão:

Nome da Coluna	Descrição
CODIGO_CLIENTE	Código do cliente perante a empresa
DESCRICAO	Nome desse cliente
ABREVIATURA	Abreviatura ou sigla utilizada pela empresa

NOME_CONTATO	Nome do contato utilizado para vincular transações financeiras com o cliente
E_MAIL_CONTATO	Email do contato do cliente
SITUAÇÃO	Situação do cliente perante a empresa

Quadro 5.3 – Descrição dos campos da tabela SFT_AGRUP_CLIENTE

Fonte: Do próprio autor

5.1.4 Tabela dimensão SFT_MOVTO_VENC_TITULO_CLIENTE

A tabela dimensão SFT_MOVTO_VENC_TITULO_CLIENTE contém informações sobre as movimentações ocorridas sobre os vencimentos das parcelas. O quadro 5.4 apresenta uma explicação sobre os atributos que compõem esta tabela dimensão:

Nome da Coluna	Descrição
CODIGO_MOVIMENTO	Código do movimento o qual a parcel sofreu
CODIGO_BANCO	Código do banco que a parcela foi vinculada
CODIGO_FORMA_PAGTO_DOCUMENTO	Forma de pagamento vinculado à parcela
DATA_MOVIMENTO	Data de movimentação pelo qual a parcela passou
VALOR	Valor da parcela
NUMERO_TITULOS	Número do titulo perante a nota fiscal
SERIE	Série perante a nota fiscal
CODIGO_EMPRESA	Código da empresa perante a nota fiscal
NUMERO_PARCELA	Número da parcela dos títulos perante a nota fiscal

Quadro 5.4 – Descrição dos campos da tabela SFT_MOVTO_VENC_TITULO_CLIENTE

Fonte: Do próprio autor

5.1.5 Tabela dimensão FUNCAO_DATA

A tabela dimensão FUNCAO_DATA contém uma fórmula para agrupar informações referentes a datas das consultas processadas. O quadro 5.5 apresenta uma explicação sobre os atributos que compõem esta tabela dimensão:

Nome da Coluna	Descrição
----------------	-----------

DATA_MOVIMENTO (IN)	Data de movimentação do título
DATA_VENCIMENTO (IN)	Data de vencimento da parcela
BIMESTRE (OUT)	Bimestre que a data se encontra
TRIMESTRE (OUT)	Trimestre que a data se encontra
SEMESTRE (OUT)	Semestre que a data se encontra
ANO (OUT)	Ano que a data se encontra

Quadro 5.5 – Descrição dos campos da tabela FUNCAO_DATA

Fonte: Do próprio autor

5.1.6 Tabela dimensão SFT_LOCALIZA

A tabela dimensão SFT_LOCALIZA contém as informações referentes à localização geograficamente de cada cliente para possíveis agrupamentos por região. O quadro 5.6 apresenta uma explicação sobre os atributos que compõem esta tabela dimensão:

Nome da Coluna	Descrição
COD_LOCALIZACAO	Código da cidade
NOME_CIDADE	Nome da cidade
UF	Estado

Quadro 5.6 – Descrição dos campos da tabela SFT_LOCALIZA

Fonte: Do próprio autor

5.2 Carga de dados

O processo de carga dos dados é automatizado pela ferramenta com base nas informações cadastradas no processo de criação do DM descrito anteriormente. Neste estudo de caso, todas as informações virão diretamente de tabelas do ERP SFT, a partir de informações utilizadas pela empresa.

As informações são carregadas conforme o modelo dimensional utilizado, onde a ferramenta verifica as informações existentes entre o ERP e o DM, faz um comparativo verificando quais informações obterão algum tipo de alteração e posterior essa análise estar concluída, é efetuado a carga de somente as informações que estão diferentes entre as bases. Este processo é simples, automático e funciona sem interações com o usuário no decorrer da

execução. Pode ser executado através de ferramenta ou por um agendamento efetuado na própria ferramenta. Os testes de cargas efetuados foram montados sobre o funcionamento manual da ferramenta, somente para fins de testar a estrutura.

Para cada nova carga montada, é efetuado testes sobre a base do qual serão dispostas essas informações e o status que cada tabela carregada teve no processo. A figura 5.2 demonstra o *job* pelo qual irá montar a carga automática da base de dados já à figura 5.3 demonstra após a carga estar concluída, as informações oriundas desse processo.

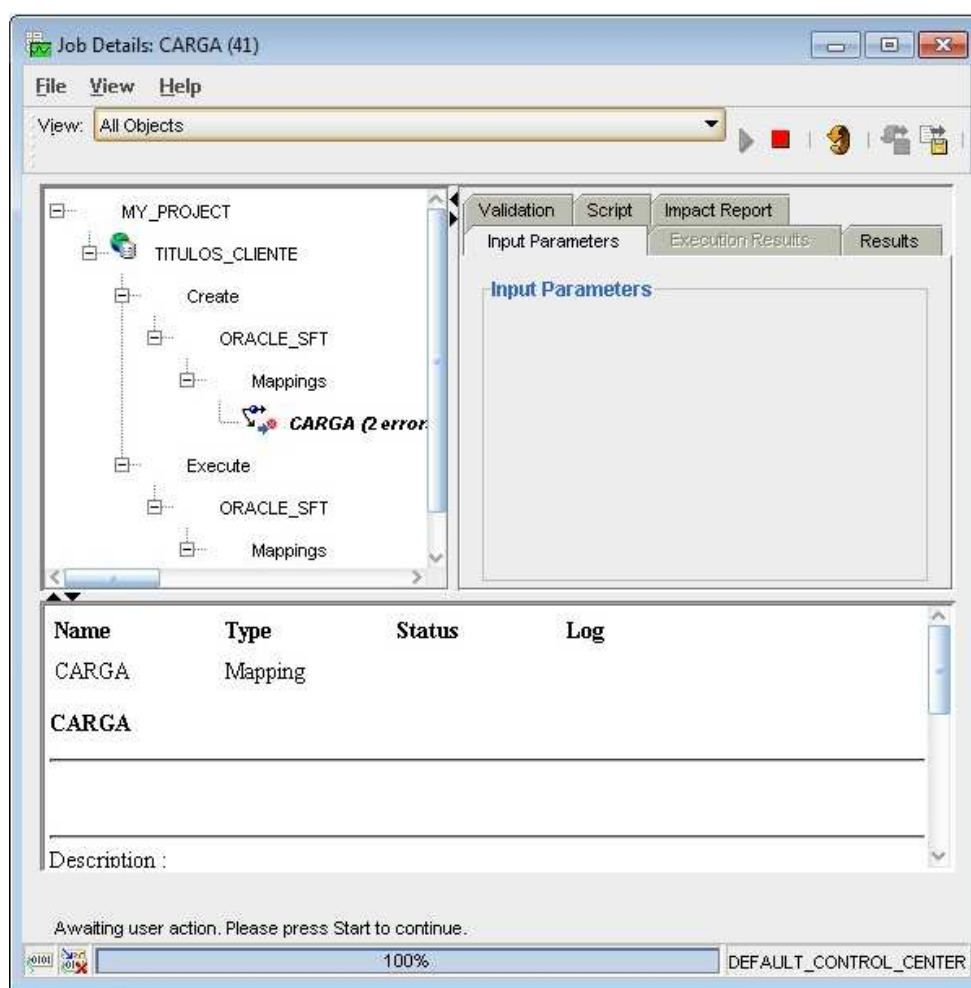


Figura 5.2– Rotina de carga de dados entre ERP e DM

Fonte: Do próprio autor

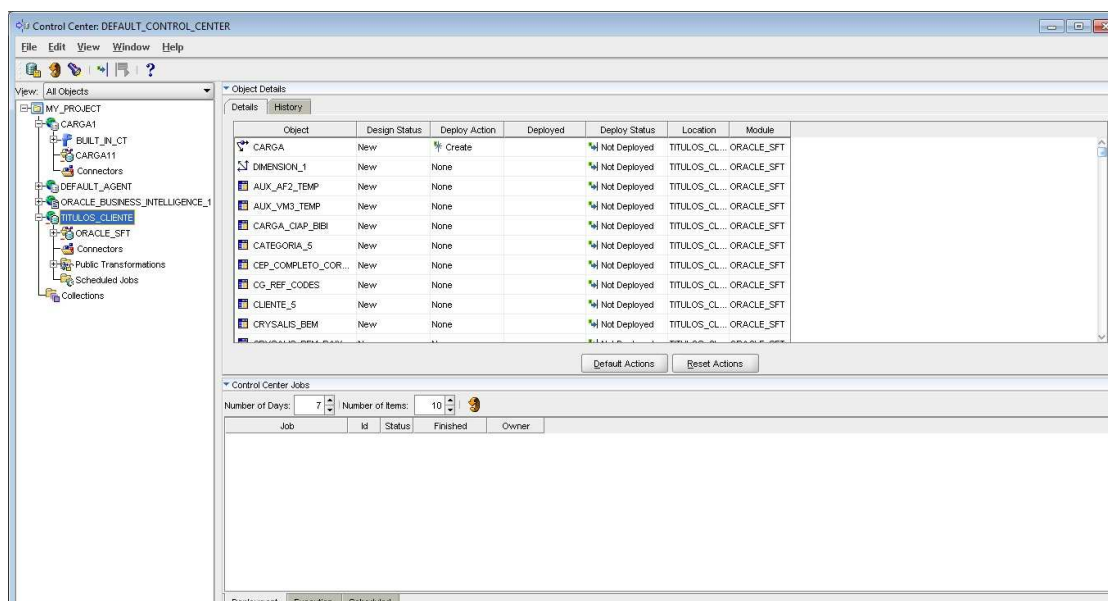


Figura 5.3– Estrutura de dados carregados no DM

Fonte: Do próprio autor

5.3 Criação dos cubos de dados

Conforme exposto anteriormente, será criado um cubo de dados para cada um dos DMs construídos. A fim de demonstrar este recurso, optou-se pela construção de dois cubos de dados para trabalhar com diferentes visões sobre os DMs construídos. As duas visões trabalham com a tabela fato `SFT_FATO_TITUTLO_CLIENTE` e as dimensões `SFT_AGRUP_CLIENTE`, `SFT_LOCALIZA`, `FUNÇÃO_DATA`, `SFT_AGRUP_BANCO`, `SFT_MOVTO_VENC_TITULO_CLIENTE`, diferenças de nomenclatura foram postas pelos modelos utilizados para cada caso. A diferença entre eles consiste nos modelos dimensionais utilizados para cada modelagem do cubo, onde um utiliza o modelo *Star* e o outro utiliza o modelo *SnowFlake*.

5.3.1 Modelo *Star*

O primeiro cubo construído tem como finalidade agrupar as informações das parcelas pagas e a serem pagas pelos clientes agrupando por banco do qual a parcela está vinculado, informações referente ao cliente que possui movimentações sobre as parcelas sofridas, agrupar por períodos de data e localização geográfica, e isso tudo utilizando o modelo *Star*. A figura 5.4 ilustra esse modelo relacionado.

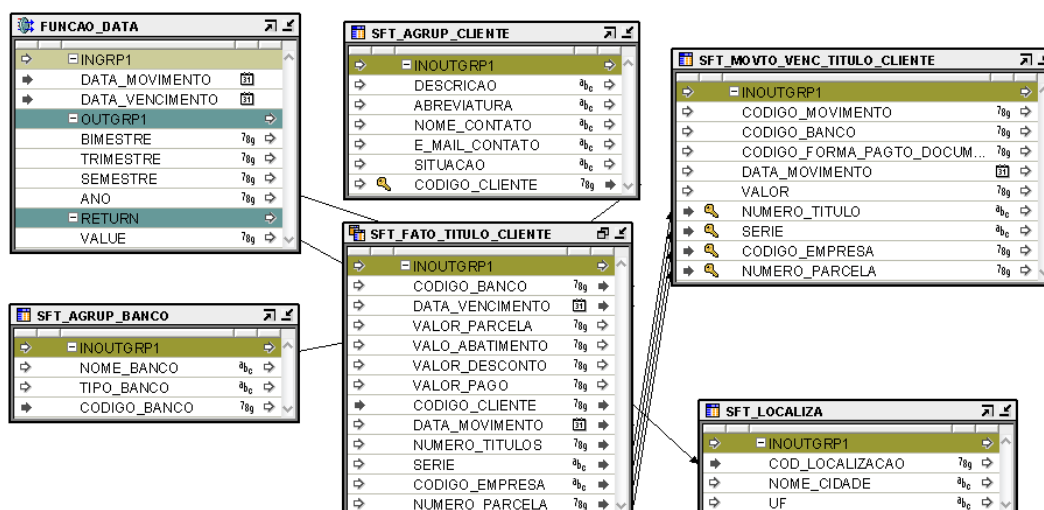


Figura 5.4– ER do Modelo Star

Fonte: Do próprio autor

Inicializa-se a criação do cubo de dados informando um nome para o mesmo do qual esse nome seja bem claro a situação podendo facilitar no momento de alguma manutenção. Após deve ser definido qual a tabela fato desse cubo e suas dimensões que serão utilizadas na visão dos dados. A figura 5.5 representa a criação desse cubo, e a escolha das tabelas. As tabelas apresentadas na descrição do cubo estão abertas definindo a origem de cada informação perante o cubo.

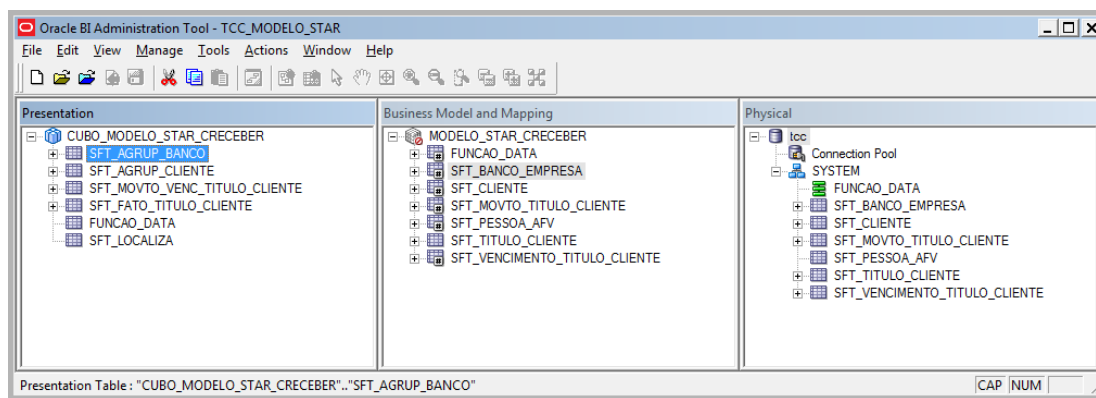


Figura 5.5– Cubo de dados Modelo Star

Fonte: Do próprio autor

Na construção desse modelo, é necessário efetuar o vínculo das tabelas físicas a ferramenta *Oracle BI Administration Tool* para que tenha conexão entre as estruturas. Após isso é efetuado o mapeamento do modelo de negócio que será utilizado, onde as tabelas fato e dimensão são montadas internamente na estrutura e aparecem no modelo as informações físicas utilizadas para a rotina. Essa estrutura até o módulo de mapeamento, as informações

vem carregadas pela ferramenta *Oracle Warehouse Builder*, de onde o *Oracle BI Administration Tool* somente faz alguns refinamentos de informação de onde o *Oracle Warehouse Builder* não proporciona para a montagem do cubo.

Esse modelo dimensional utilizado proporciona maior facilidade para se trabalhar com as informações contidas no cubo, facilita na manutenção do cubo para fins de acréscimo ou deleção de algum campo não mais necessário, possibilitando o fácil entendimento do que se necessita e facilitando a utilização como um todo.

5.3.2 Modelo Snowflake

O segundo cubo construído tem como finalidade agrupar as informações das parcelas pagas e a serem pagas pelos clientes agrupando por banco do qual a parcela está vinculado, informações referente ao cliente que possui movimentações sobre as parcelas sofridas, agrupar por períodos de data e localização geográfica, e isso tudo utilizando o modelo *Snowflake*. A figura 5.6 ilustra esse modelo relacionado.

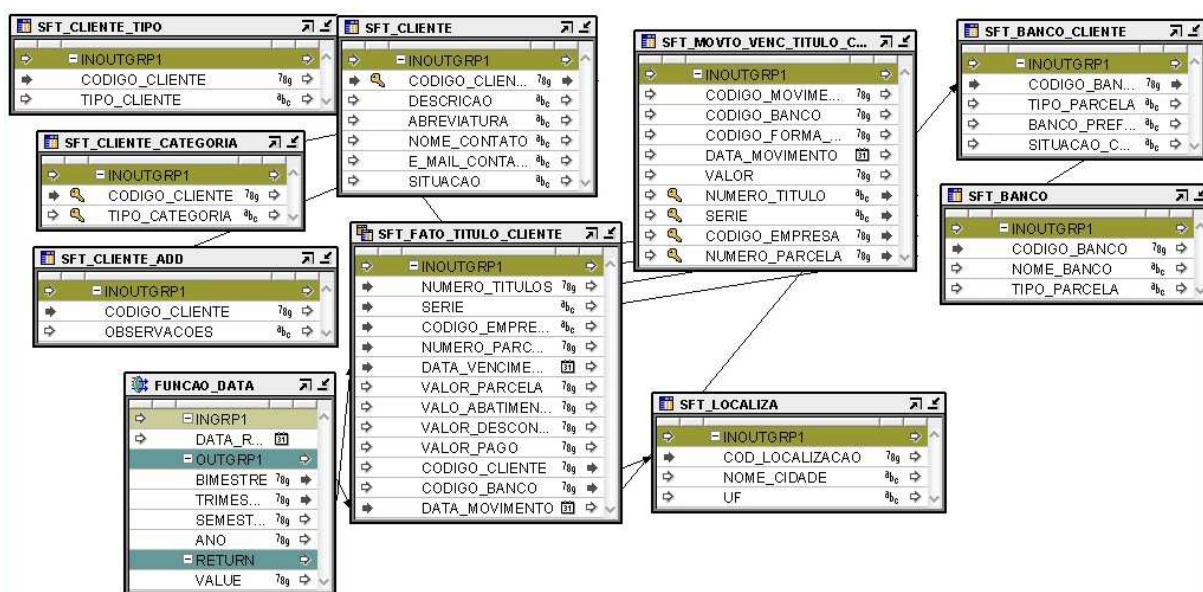


Figura 5.6— ER do Modelo Snowflake

Fonte: Do próprio autor

Inicializa-se a criação do cubo de dados informando um nome para o mesmo do qual esse nome seja bem claro a situação podendo facilitar no momento de alguma manutenção. Após deve ser definido qual a tabela fato desse cubo e suas dimensões que serão utilizadas na visão dos dados. A figura 5.7 representa a criação desse cubo, e a escolha das tabelas. As tabelas apresentadas na descrição do cubo estão abertas definindo a origem de cada informação perante o cubo.

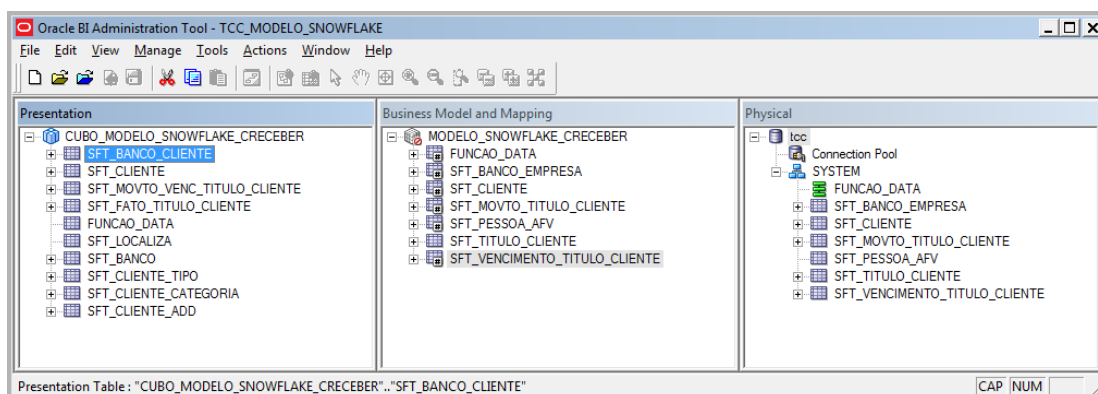


Figura 5.7– Cubo de dados Modelo *Snowflake*

Fonte: Do próprio autor

Esse modelo dimensional utilizado proporciona um modelo mais detalhado de todas as informações contidas no cubo, detalha as movimentações que as tabelas sofrem para retornar alguma informação solicitada, expande as consultas no seu nível máximo, de fácil compreensão para possíveis manutenções.

5.4 Análise dos cubos de dados

Após a construção do DM, carga dos dados e criação dos cubos de dados, está tudo pronto para que os gestores ou responsáveis pela análise das informações oriundas da base possam ser utilizados por uma ferramenta OLAP.

Para melhor avaliar o desempenho de cada cubo, será efetuada uma leitura do custo de processamento necessário para efetuar a consulta das informações. Esse custo aponta todos os locais por onde foi efetuada a consulta, e quanto maior o custo para efetuar consultas, maior o tempo de processamento da consulta.

5.4.1 Modelo *Star*

O modelo desenvolvido para este caso proporcionou uma quantidade inferior de custo de processamento, menor taxa de cardinalidade entre a consulta das informações utilizadas e uma quantidade muito menor de bytes utilizados para efetuar a consulta. Na sequência, a figura 5.8 irá listar a quantidade de bytes necessários para efetuar a consulta no modelo *Star*.

```
select *
from cubo_modelo_star1
```

Optimizer goal: All rows

Description	Object own...	Object name	Cost	Cardinality	Bytes
SELECT STATEMENT, GOAL = ALL_ROWS			971	1	279
VIEW	SYS	CUBO_MODEL0_STAR1	971	1	279
HASH UNIQUE			971	1	158
HASH JOIN OUTER			970	1	158
NESTED LOOPS			967	1	120
HASH JOIN OUTER			966	1	117
NESTED LOOPS			964	1	110
NESTED LOOPS			963	1	106
NESTED LOOPS			960	1	50
NESTED LOOPS			960	8	376
HASH JOIN			959	1326	57018
TABLE ACCESS FULL	SYSTEM	SFT_BANCO_CLIENTE	2	68	748
TABLE ACCESS FULL	SYSTEM	SFT_TITULO_CLIENTE	955	314364	10059648
INDEX UNIQUE SCAN	SYSTEM	CLI_ACCESS_PK	0	1	4
INDEX RANGE SCAN	SYSTEM	CLI_ACCESS_SLI_ACCESS_FK_I	0	1	3
TABLE ACCESS BY INDEX ROWID	SYSTEM	SFT_VENCIMENTO_TITULO_CLIENTE	3	1	56
INDEX RANGE SCAN	SYSTEM	SFT_VETICL_IND_3	2	1	
INDEX RANGE SCAN	SYSTEM	CLI_ACCESS_RELEASE_CLI_ACCESS_FK_I	1	23	92
TABLE ACCESS FULL	SYSTEM	SFT_BANCO_EMPRESA	2	4	28
INDEX RANGE SCAN	SYSTEM	SFT_CLIENTE_ACESSO_MODULO_PK	1	9	27
TABLE ACCESS FULL	SYSTEM	SFT_MOVTO_TITULO_CLIENTE_TEMP	2	1	38

Figura 5.8– Plano de execução do Modelo *Star*

Fonte: Do próprio autor

5.4.2 Modelo *Snowflake*

O modelo desenvolvido para este caso proporcionou uma quantidade superior de custo de processamento, maior taxa de cardinalidade entre a consulta das informações utilizadas e uma quantidade maior de bytes utilizados para efetuar a consulta. Na sequência, a figura 5.9 irá listar a quantidade de bytes necessários para efetuar a consulta no modelo *StarowFlake*.

```
select *
from cubo_modelo_snowflake1
```

Optimizer goal: All rows

Description	Object own...	Object name	Cost	Cardinality	Bytes
SELECT STATEMENT, GOAL = ALL_ROWS			1845	5240	1461960
VIEW	SYS	CUBO_MODEL0_SNOWFLAKE1	1845	5240	1461960
HASH UNIQUE			1845	5240	806960
HASH JOIN			1663	5240	806960
INDEX FAST FULL SCAN	SYSTEM	CLI_ACCESS_RELEASE_CLI_ACCESS_FK_I	3	1171	4684
HASH JOIN			1660	4306	645900
INDEX FULL SCAN	SYSTEM	SFT_CLIENTE_ACESSO_MODULO_PK	1	113	339
HASH JOIN RIGHT OUTER			1658	2439	358533
TABLE ACCESS FULL	SYSTEM	SFT_BANCO_EMPRESA	2	4	28
HASH JOIN			1656	2439	341460
TABLE ACCESS FULL	SYSTEM	SFT_BANCO_CLIENTE	2	68	748
HASH JOIN RIGHT OUTER			1653	825	106425
TABLE ACCESS FULL	SYSTEM	SFT_MOVTO_TITULO_CLIENTE_TEMP	2	1	38
NESTED LOOPS			1651	825	75075
NESTED LOOPS			423	409	14315
INDEX FULL SCAN	SYSTEM	CLI_ACCESS_SLI_ACCESS_FK_I	1	21	63
TABLE ACCESS BY INDEX ROWID	SYSTEM	SFT_TITULO_CLIENTE	21	19	608
INDEX RANGE SCAN	SYSTEM	SFT_TICL_IND_1	2	19	
INDEX RANGE SCAN	SYSTEM	SFT_VETICL_IND_3	2	1	
TABLE ACCESS BY INDEX ROWID	SYSTEM	SFT_VENCIMENTO_TITULO_CLIENTE	3	2	112

Figura 5.9– Plano de execução do Modelo *Snowflake*

Fonte: Do próprio autor

5.4.3 Considerações Finais

Os modelos apresentados possuem resultados de pesquisas utilizando DMs diferentes e com tabelas modeladas conforme a estrutura vista, todos os atributos e campos *primary key* e *foreign key* foram utilizados conforme a arquitetura utilizada para cada necessidade. Foi visto que há um melhor aproveitamento dos recursos do banco de dados para cubos montados sobre modelos *Star* onde a quantidade de recursos necessários para suas consultas é muito inferior ao modelo *Snowflake* mostrado no trabalho.

Pela visualização de dados utilizada, foi visto que recursos do modelo *Star* possuem um melhor desempenho no seu todo, para casos onde há pouca informação sendo processada, essa diferença de tempo quase não é percebida, mas para casos com um montante muito maior de dados, é possível verificar a real diferença pela escolha da modelagem dimensional dos cubos de dados.

O intuito dessa comparação era montar um paradigma entre dois modelos clássicos da construção de DMs dos quais, geralmente se atribuía o modelo *Star* por ser um modelo de mais simples construção e mais rápido desenvolvimento. Para cada caso é necessário avaliar muito bem qual modelagem foi utilizada porque a regra não cabe a todos os DMs desenvolvidos, onde possa haver situações onde o modelo *Snowflake* possa ser de um melhor desempenho que o *Star*.

5.5 Consulta dos dados

Após a construção do DM, a carga de dados e a criação dos cubos de dados, está na hora em que o gestor possa analisar os dados usando uma ferramenta OLAP, para que este possa interpretar os dados e transformá-los em informações para auxiliar o processo de decisão da empresa. Conforme indicado anteriormente, as consultas OLAP podem ser realizadas de várias formas propostas no trabalho, desde soluções prontas da *Oracle*, até pela utilização de um módulo OLAP desenvolvido para abranger essa necessidade. No estudo de caso será utilizada a ferramenta *Oracle BI Publisher* para efetuar consultas dinâmicas interativas de dados do ERP SFT e também será posto o módulo desenvolvido para consultas na ferramenta em Excel.

5.5.1 Oracle BI Publisher

Com a possibilidade de desenvolver relatórios dinâmicos e totalmente interativos com os usuários, essa proposta de utilização é a mais recomendada para uso de recursos

OLAP para soluções *Oracle*. Seu conceito interativo, possível de ser gerado para qualquer tipo de aplicativo e de fácil manuseio faz com que seu uso seja mais utilizado.

A figura 5.10 mostra a flexibilidade de seu uso.

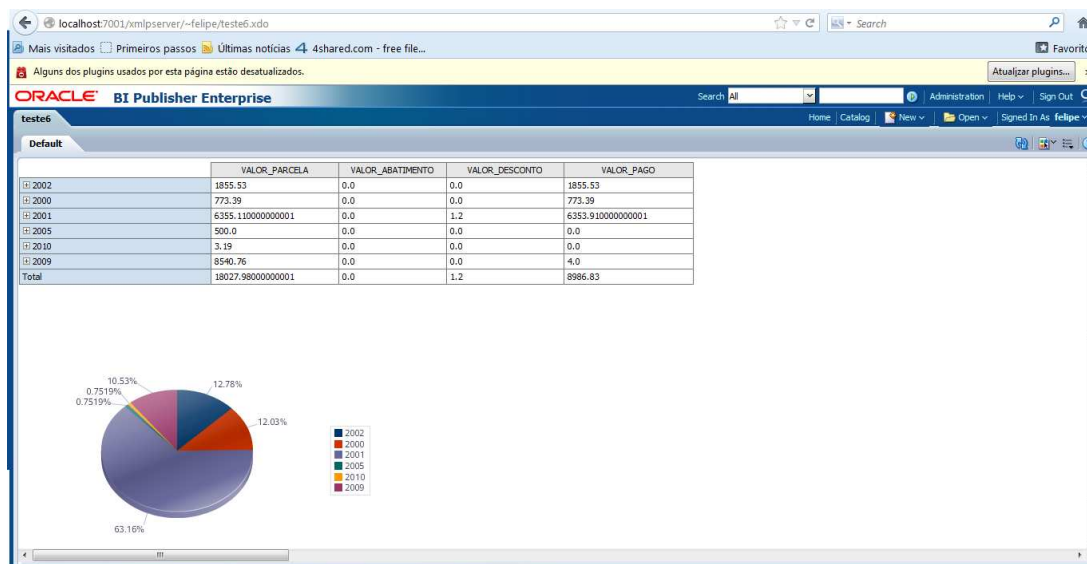


Figura 5.10– Solução proposta com *Oracle BI Publisher*

Fonte: Do próprio autor

5.5.2 Módulo OLAP

Como há a possibilidade que nem todos os usuários possuam acesso à utilização da ferramenta proposta pela *Oracle*, foi desenvolvido um módulo OLAP do qual será utilizado para criar os relatórios dinâmicos para os usuários.

Junto a isso, foi desenvolvida toda a estrutura, configurações e parametrizações para a geração desses relatórios no sistema através desse módulo, que interage com todas as informações que serão utilizadas tanto na ferramenta da *Oracle*, como no arquivo Excel gerado por ele.

A figura 5.11 irá mostrar como funciona a parametrização da ferramenta.

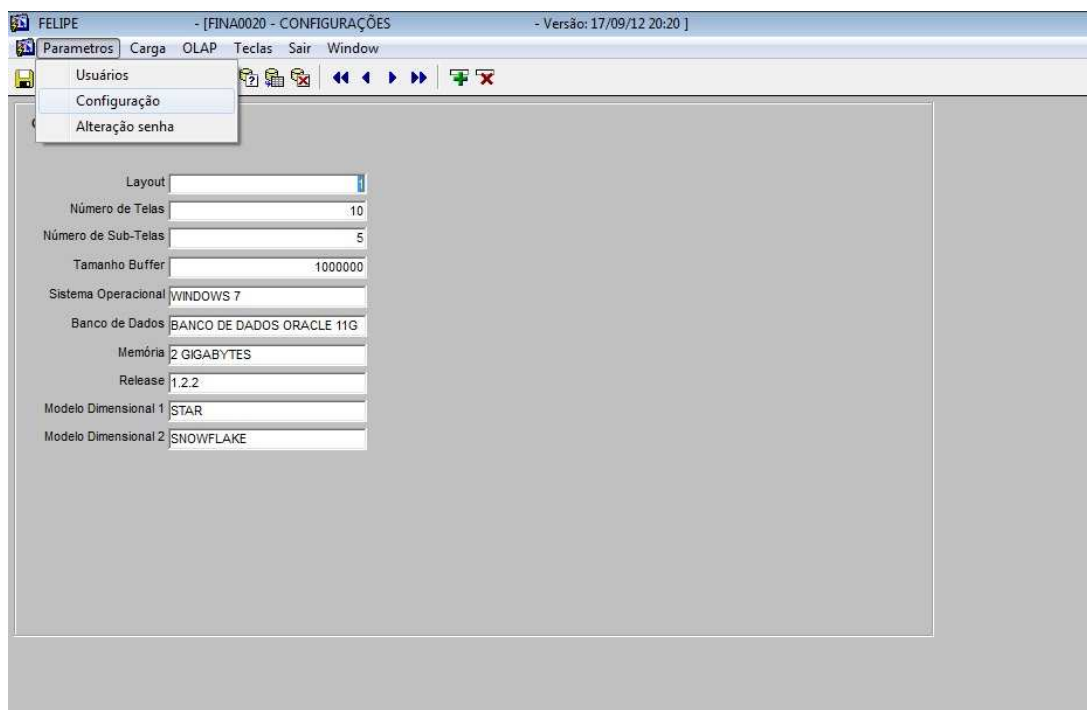


Figura 5.11– Tela de parametrização das consultas OLAP

Fonte: Do próprio autor

Para efetuar a carga do cubo de dados sobre a estrutura, o módulo possui um programa que carrega o arquivo com formato RPD (*Oracle BI Repository File*) para a estrutura de formatação do sistema e após essa etapa concluída, pode gerar qualquer planilha OLAP com as informações extraídas do arquivo.

As figuras 5.12 e 5.13 relatam essa carga de informação e a forma como essa informação ficam dispostas para um uso posterior.



Figura 5.12– Tela de carga das consultas OLAP

Fonte: Do próprio autor

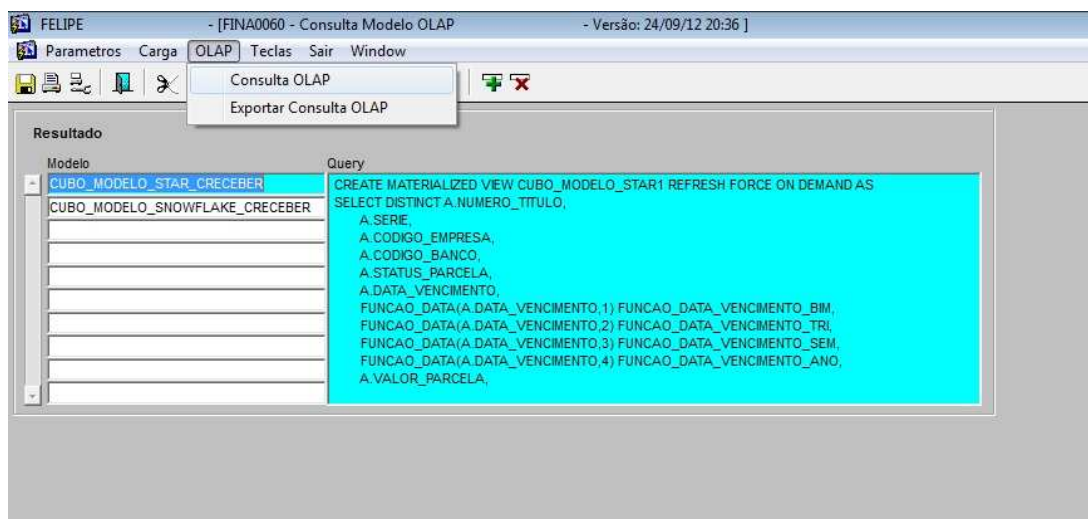


Figura 5.13– Tela para consulta da carga das consultas OLAP

Fonte: Do próprio autor

Para poder efetuar a geração do arquivo Excel com os recursos OLAP, é necessário utilizar o programa FINA0070.fmb (figura 5.14) de onde são exportados todas as informações coletadas do cubo de dados importado via sistema.

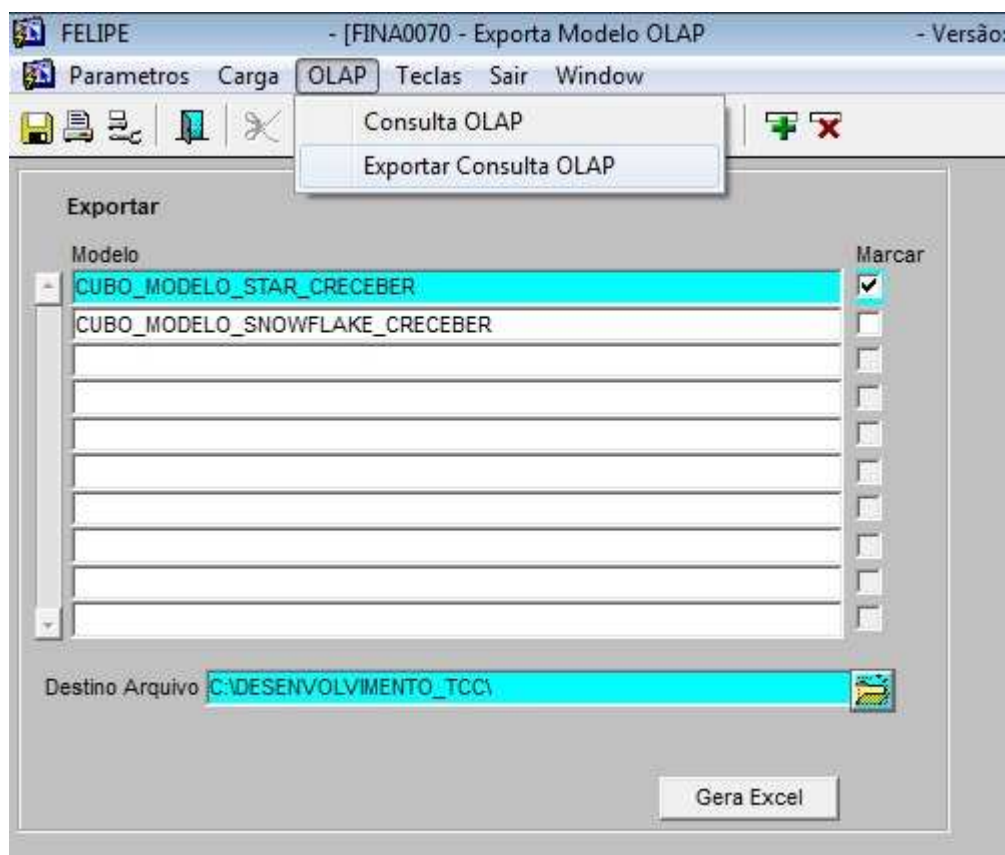


Figura 5.14– Tela para geração do arquivo Excel

Fonte: Do próprio autor

O resultado dessa geração de arquivo Excel, é a figura 5.15 relacionada no modelo abaixo.

NUMERO_TITULO	SERIE	CODIGO_EMPRESA	CODIGO_BANCO	STATUS_PARCELA	DATA_VENCIMENTO	FUNCAO
209245	1	1	999	P	9/7/2001	
209245 Total						
213790	1	1	999	P	9/27/2001	
213790 Total						
753	1	9	999	A	8/4/2009	
753 Total						
160397	1	1	999	P	11/7/2000	
160397 Total						
210085	1	1	999	P	9/28/2001	
210085 Total						
186545	1	1	999	P	3/16/2001	
186545 Total						
160784	1	1	999	P	11/17/2000	
160784 Total						
194993	1	1	999	P	8/28/2001	

Figura 5.15– Relatório Excel formatado no módulo OLAP

Fonte: Do próprio autor

O modelo visualizado no Excel pode ser redimensionado conforme a necessidade do gestor que o utilize, tendo opções para poder ser consultada essas informações da forma que for mais conveniente para a situação no momento. O processo de utilização do módulo OLAP para Excel é algo bem intuitivo para usuários que já possuem por prática o seu uso nas diversas atividades do dia-a-dia de uma empresa.

CONCLUSÃO

A partir do estudo realizado e considerando o mercado de TI, pode-se observar que os sistemas ERP exercem um papel importante dentro de uma organização. Afinal, eles são responsáveis pela unificação das informações operacionais da empresa, solucionando problemas de disponibilidade e confiabilidade de informações. Considerando estas características, as empresas estão investindo cada vez mais em sistemas ERP, visando aumentar a sua competitividade no mercado.

Os sistemas de BI são reconhecidos no mercado pela sua capacidade de reunir e consolidar informações oriundas dos diferentes sistemas de uma empresa. Desta forma o BI torna-se uma importante ferramenta para os gestores que desejam obter vantagens competitivas através da tomada de decisões rápidas e acertadas.

Visando reunir as funcionalidades dos sistemas ERP, com os benefícios de um sistema de BI, este trabalho propôs a construção de uma solução de BI integrada ao ERP Safetech. A solução proposta é composta pelo *Data Mart* e por ferramentas com consultas analíticas, visando fornecer aos gestores subsídios para a tomada de decisões, baseadas em informações consolidadas pertinentes ao processo de faturamento.

Desta forma, facilmente podem ser construídos os cubos de dados para atender os diversos ramos de atividades que utilizam o Gestão SFT. Sendo assim, a informação estará disponível para os gestores das empresas no momento e no formato em que forem necessárias, facilitando e agilizando a tomada de decisão, e evitando assim uma provável perda de mercado.

Com uma estrutura desenvolvida para a geração de consultas OLAP em ambiente *Web* e pela geração de arquivos Excel pelo módulo desenvolvido, sua flexibilidade é seu grande atrativo e diferencial perante a proposta desenvolvida.

O estudo de caso realizado demonstra que a solução atende a diversas situações, tendo flexibilidade para obtenção de dados contidos no Gestão SFT, onde puderam ser analisados a fim de obter informações importantes para o processo decisório.

REFERÊNCIAS BIBLIOGRÁFICAS

CHAKRABARTI, Soumen. **Data Mining: Know it all**, 2009. 460 p.

COREY, Michael. **Oracle 8i Data Warehouse**. São Paulo, SP: Editora Campus, 2001, 9 p.

FELBER, Edmilson J. W. **Proposta de Uma Ferramenta OLAP em um Data Mart Comercial**: Uma Aplicação prática na Indústria Calçadista. Novo Hamburgo: 2006. 21 p. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo.

FELTES, Luiz Henrique. **Desenvolvimento de uma solução de BI para o ERP Siger**. Novo Hamburgo: 2010. 18 p. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo.

INMON, W.H. Wetch, J.D. Glassey, Katherine L. **Gerenciador Data Warehouse**. São Paulo, SP: Makron Books, 1999. 375 p.

KIMBALL, Ralph. **Data Warehouse Toolkit**. São Paulo, SP: Makron Books, 1998. 8 p.

KIMBALL, Ralph. **The Data Warehouse Lifecycle Toolkit**. São Paulo, SP: Makron Books, 2008. 238 p.

LARSON, B. e Argarwall, S. (2006), **Delivery Business Intelligence with Microsoft SQL Server 2005**, McGraw-Hill. 792 p.

ORACLE. **Oracle**. Disponível em: <http://www.oracle.com/technetwork/indexes/downloads/index.html>. Acesso em: 05 de Novembro de 2012.

PRIMAK, Fábio Vinícius. **Decisões de BI**, 2008. 4-5p.152 p.

PRODANOV, C.C. Freitas, E.C. **Metodologia do Trabalho Científico**. Novo Hamburgo, RS: Editora Feevale, 2009. 31 p.

SANTOS, Rodrigo Schermer. **Solução de Business Intelligence aplicada ao módulo de Vendas do CIGAM**. Novo Hamburgo: 2009. 18 p. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo.

SIMON, Inre. **Um Estudo de Caso: A Produção e Disseminação da Literatura Acadêmica**. Disponível em: <http://www.ime.usp.br/~is/ddt/mac339-01/aulas/www.linux>.

ime.usp.br/hvila/mac339/tema8.html. Acesso em: 10 de setembro de 2012.

THOMSEN, Erik. **OLAP : Construindo Sistemas de Informações Multidimensionais**, 2002. 5-6 p.

YIN, R. **Estudo de caso: Planejamento e Métodos**. 2^a ed. Porto Alegre: Bookman, 2001.

APÊNDICE

APÊNDICE A – CONFIGURAÇÃO DA TABLESPACE DO DM

```
create tablespace tcc datafile 'C:\Oracle\app\oradata\orcl\tcc.dbf'  
size 10M  
autoextend on maxsize 500M  
extent management local uniform size 64K;
```

```
CREATE USER felipe IDENTIFIED BY felipe DEFAULT TABLESPACE tcc QUOTA UNLIMITED ON tcc;
```

```
GRANT create session, alter session, select_catalog_role, execute_catalog_role, create  
table, create procedure, create view,  
create materialized view, create trigger, create sequence, create any directory,  
create type, create synonym, administer database trigger TO felipe;
```

```
--
```

```
create role r_tcc;
```

```
--
```

```
-- Grant/Revoke role privileges
```

```
grant connect to FELIPE;
```

```
grant r_tcc to FELIPE;
```

```
grant debug connect session to FELIPE;
```

```
grant select any table to FELIPE;
```

APÊNDICE B – CRIAÇÃO DAS TABELAS PARA O OLAP

```

create table TRADUCAO_PROGRAMA_ITEM
(
  CODIGO_IDIOMA          NUMBER(3) not null,
  NOME_PROGRAMA          VARCHAR2(50) not null,
  NOME_BLOCO             VARCHAR2(100) not null,
  NOME_ITEM              VARCHAR2(100) not null,
  TIPO_ITEM              VARCHAR2(50) not null,
  PROMPT                 VARCHAR2(1000),
  LABEL                  VARCHAR2(1000),
  HINT                   VARCHAR2(1000),
  TOOLTIP                VARCHAR2(1000),
  VERSAO_PROGRAMA        VARCHAR2(20) not null,
  DATA_HORA_ATUALIZACAO DATE not null
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table TRADUCAO_PROGRAMA_ITEM
  add constraint TRADUCAO_PROGRAMA_ITEM_PK primary key (CODIGO_IDIOMA, NOME_PROGRAMA,
  NOME_BLOCO, NOME_ITEM)
  using index
  tablespace PLANEJAMENTO_INDICES
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
-- Create/Recreate indexes
create index TRAD_PROG_ITE_IDIOMA_PROG_IDX on TRADUCAO_PROGRAMA_ITEM (CODIGO_IDIOMA,
  NOME_PROGRAMA)
  tablespace PLANEJAMENTO_INDICES
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
--
create public synonym TRADUCAO_PROGRAMA_ITEM for TRADUCAO_PROGRAMA_ITEM;
--
grant select, insert, update, delete on TRADUCAO_PROGRAMA_ITEM to r_tcc;
--
create table TRADUCAO_PROGRAMA_OBJETO

```

```

(
  CODIGO_IDIOMA          NUMBER(3) not null,
  NOME_PROGRAMA          VARCHAR2(50) not null,
  NOME_OBJETO            VARCHAR2(100) not null,
  TIPO_OBJETO            VARCHAR2(100) not null,
  TEXTO                  VARCHAR2(1000),
  DATA_HORA_ATUALIZACAO DATE not null
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table PTRADUCAO_PROGRAMA_OBJETO
  add constraint TRADUCAO_PROGRAMA_OBJETO_PK primary key (CODIGO_IDIOMA,
  NOME_PROGRAMA, NOME_OBJETO, TIPO_OBJETO)
  using index
  tablespace PLANEJAMENTO_INDICES
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
-- Create/Recreate indexes
create index TRAD_PROG_OBJ_IDIOMA_PROG_IDX on TRADUCAO_PROGRAMA_OBJETO (CODIGO_IDIOMA,
  NOME_PROGRAMA)
  tablespace PLANEJAMENTO_INDICES
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 64K
    minextents 1
    maxextents unlimited
  );
--
create public synonym TRADUCAO_PROGRAMA_OBJETO for TRADUCAO_PROGRAMA_OBJETO;
--
grant select, insert, update, delete on TRADUCAO_PROGRAMA_OBJETO to r_tcc;
--
create table TRADUCAO_PROGRAMA_LISTA
(
  CODIGO_IDIOMA          NUMBER(3) not null,
  NOME_PROGRAMA          VARCHAR2(50) not null,
  NOME_BLOCO             VARCHAR2(100) not null,
  NOME_ITEM              VARCHAR2(100) not null,
  INDICE                 NUMBER(3) not null,
  VALOR                  VARCHAR2(100),
  LABEL                  VARCHAR2(100),
  VERSAO_PROGRAMA        VARCHAR2(20) not null,
  DATA_HORA_ATUALIZACAO DATE not null

```

```

)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table TRADUCAO_PROGRAMA_LISTA
add constraint TRADUCAO_PROGRAMA_LISTA_PK primary key (CODIGO_IDIOMA, NOME_PROGRAMA,
NOME_BLOCO, NOME_ITEM, INDICE)
using index
tablespace PLANEJAMENTO_INDICES
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create index TRAD_PROG_LIS_IDIOMA_PROG_IDX on TRADUCAO_PROGRAMA_LISTA (CODIGO_IDIOMA,
NOME_PROGRAMA)
tablespace PLANEJAMENTO_INDICES
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
--
create public synonym TRADUCAO_PROGRAMA_LISTA for TRADUCAO_PROGRAMA_LISTA;
--
grant select, insert, update, delete on TRADUCAO_PROGRAMA_LISTA to r_tcc;
--
create table USUARIO_INFORMATICA
(
    SERVIDOR                VARCHAR2(30) not null,
    LOGIN_USUARIO           VARCHAR2(30) not null,
    NOME_USUARIO            VARCHAR2(50) not null,
    SETOR                   VARCHAR2(30) not null,
    RAMAL                   VARCHAR2(30),
    E_MAIL                  VARCHAR2(300),
    SERVICOS                 VARCHAR2(1),
    MENU_NOVO               VARCHAR2(1),
    IDENTIFICADOR_FUNCIONARIO NUMBER(6),
    LICENCA_USO             VARCHAR2(1) not null,
    CODIGO_AGRUPAMENTO_MARCA NUMBER(3),
    BLOQUEADO               VARCHAR2(1) not null,
    CODIGO_IDIOMA           NUMBER(3) default 1 not null,
    CODIGO_LOCAL_CHAMADO    NUMBER(4)
)
tablespace TCC

```

```

pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 200K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table USUARIO_INFORMATICA
add constraint PK_USUARIO_INFORMATICA primary key (LOGIN_USUARIO)
using index
tablespace CHAMADO_INDICES
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 104K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate check constraints
alter table USUARIO_INFORMATICA
add constraint CK_USU_INFORMATICA_SERVICO
check (SERVICO IN ('S','N'));
--
create public synonym USUARIO_INFORMATICA for USUARIO_INFORMATICA;
--
grant select, insert, update, delete on USUARIO_INFORMATICA to r_tcc;
--
create table TRADUCAO_PROGRAMA_OUTROS
(
    CODIGO_IDIOMA          NUMBER(3) not null,
    NOME_PROGRAMA          VARCHAR2(50) not null,
    NOME_OBJETO_PAIS       VARCHAR2(100) not null,
    TIPO_OBJETO_PAIS       VARCHAR2(100) not null,
    NOME_OBJETO_FILHO      VARCHAR2(100) not null,
    TIPO_OBJETO_FILHO      VARCHAR2(100) not null,
    PROMPT                 VARCHAR2(1000),
    LABEL                  VARCHAR2(1000),
    TITLE                  VARCHAR2(1000),
    DATA_HORA_ATUALIZACAO DATE not null
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table TRADUCAO_PROGRAMA_OUTROS
add constraint TRADUCAO_PROGRAMA_OUTROS_PK primary key (CODIGO_IDIOMA,
NOME_PROGRAMA, NOME_OBJETO_PAIS, TIPO_OBJETO_PAIS, NOME_OBJETO_FILHO, TIPO_OBJETO_FILHO)
using index
tablespace PLANEJAMENTO_INDICES
pctfree 10

```



```

initrans 2
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create index TRAD_PROG_OUT_IDIOMA_PROG_IDX on TRADUCAO_PROGRAMA_OUTROS (CODIGO_IDIOMA,
NOME_PROGRAMA)
    tablespace PLANEJAMENTO_INDICES
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 64K
        minextents 1
        maxextents unlimited
    );
--
create public synonym TRADUCAO_PROGRAMA_OUTROS for TRADUCAO_PROGRAMA_OUTROS;
--
grant select, insert, update, delete on TRADUCAO_PROGRAMA_OUTROS to r_tcc;
--
create table USUARIOS
(
    IDENTIFICACAO_USUARIO VARCHAR2(1000),
    NOME_USUARIO           VARCHAR2(1000),
    OBSERVACAO             VARCHAR2(1000),
    ACESSO                 VARCHAR2(1000)
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
--
create public synonym USUARIOS for USUARIOS;
--
grant select, insert, update, delete on USUARIOS to r_tcc;
--
create table USUARIOS_EMPRESAS_FINANCEIRO
(
    IDENTIFICACAO_USUARIO VARCHAR2(1000),
    COD_EMPRESA            VARCHAR2(1000),
    altera_forma_pagamento VARCHAR2(1000)
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1

```

```

        maxextents unlimited
    );
--
create public synonym USUARIOS_EMPRESAS_FINANCEIRO for USUARIOS_EMPRESAS_FINANCEIRO;
--
grant select, insert, update, delete on USUARIOS_EMPRESAS_FINANCEIRO to r_tcc;
--
create table CARGA_OLAP
(
    PROGRAMA_CARREGADO    VARCHAR2(1000),
    CARGA_COMPLETA        VARCHAR2(1) DEFAULT 'S'
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
--
create public synonym CARGA_OLAP for CARGA_OLAP;
--
grant select, insert, update, delete on CARGA_OLAP to r_tcc;
--
create table MODELO_OLAP
(
    MODELO_CARREGADO      VARCHAR2(200),
    QUERY_CARREGADA       LONG
)
tablespace TCC
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
--
create public synonym MODELO_OLAP for MODELO_OLAP;
--
grant select, insert, update, delete on MODELO_OLAP to r_tcc;

```